Supplementary Material

Cheng Gu
TU Berlin
c.gu@campus.tu-berlin.de

Erik Learned-Miller
UMass Amherst
elm@cs.umass.edu

Daniel Sheldon
UMass Amherst
sheldon@cs.umass.edu

Guillermo Gallego
TU Berlin & Einstein Center Digital Future
guillermo.gallego@tu-berlin.de

Pia Bideau
TU Berlin
p.bideau@tu-berlin.de

Abstract

In the main paper we introduced a new probabilistic approach for event alignment. We show new state-of-the-art results based on several different error measures such as the RMS-error or the absolute angular error. Here we support this analysis by showing additional visual results indicating the high quality of our new proposed method. Secondly we provide a more theoretical background of our model justifying the likelihood function of observed event data. The observed event data is defined through the mapping $(\mathbf{x}, t) \mapsto (R^t_\omega \mathbf{x}, t)$*, where* R_ω *defines the mapping that is applied to event data. This mapping can be formally justified via the Poisson mapping theorem, which is discussed here. Along with this pdf-document we provide code of our algorithm including a demo that aligns a batch of 30k events.*

1. Visual results for velocity estimation

1.1. Angular Velocity

In Figure 2 we show the accuracy of our estimated velocity compared to the next best performing method (EMin [4]) and ground truth (IMU). Taking the sequence boxes_rotation as an example we show the estimated angular velocities over the entire sequence (60sec) as well as over a shorter duration (0.05sec) for each axis of rotation. Even during peak velocities with around 380 deg/sec estimates obtained by our method are robust and outperform previous results from [4] (Figure 2(d)).

Constant velocity assumption within a fixed event batch. Event alignment of a fixed batch of events (e.g., 30k) is typically done via assuming a constant velocity during the time span of the events. However, such a time span is a variable that depends on the amount of texture in the scene and motion of the camera. A possible fix to this issue is to use an adaptive number of events, depending on texture [2]. However this makes comparisons more difficult to interpret.

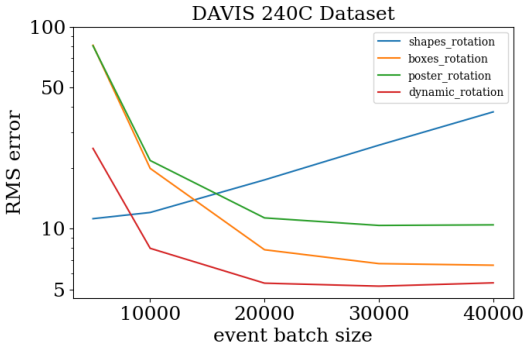


Figure 1. **Event batch size vs. accuracy.** Accuracy measured in terms of RMS error (deg/sec) for different event batch sizes to process the four rotational sequences from dataset [3].

In Figure 1 we show for each sequence the accuracy reached for a specific batch size of events. We vary the batch size between 5k and 40k events. While the three sequences boxes, poster and dynamic show relatively realistic scenes, the shapes sequence shows just a few black shapes posted onto a white background. Due to this rather simple texture much fewer events are generated within the same time interval. Conversely, a fixed number of 30k events spans over a much larger time interval, with possibly high variation in motion. If the batch size of events is too large, the constant velocity assumption leads to a significant drop in performance. As one can see in Figure 1 velocity estimations based on fewer events are significantly more suitable for the shape_sequence.

1.2. Linear Velocity

Figure 3 shows the accuracy of EMin [4], CMax [1] and ours compared to ground truth for the boxes_translation se-

quence. Ground truth is taken from a motion capture system. EMin trying to minimize the pairwaise entropy among *all event-event pairs* suffers a lot from global minima which are reached for large Z-motions. In this case all events are mapped onto a single point, which is the focus of expansion of the camera. Besides being more robust to outliers it is visible that our new proposed method is also more accurate (see Figure 3(b), 3(d) and 3(f)).

2. Model - Additional Theoretical Background

In this section, we elaborate on the definition of the likelihood $p_{\mathcal{O}}(\mathcal{O}|\omega) = p_{\mathcal{A}}(R_\omega(\mathcal{A}))$ as an instance of the Poisson mapping theorem. We use $q_{\mathcal{A}}$ and $q_{\mathcal{O}}$ to represent the density of point sets under Poisson processes, to distinguish from the notation $p_{\mathcal{A}}$ and $p_{\mathcal{O}}$ used in the main text for the probability of the resultant event counts.

2.1. Density of aligned events $\mathcal{A}$

Recall that the aligned events $\mathcal{A}$ are distributed according to a Poisson process on $\mathcal{X} \times [0, \Delta T]$ with intensity function $\lambda(\mathbf{x}, t) \doteq \Delta T^{-1} \lambda_{\mathbf{x}}$; the factor ΔT^{-1} adjusts for the time interval so that $k_{\mathbf{x}} \sim \text{Pois}(\lambda_{\mathbf{x}})$, where $k_{\mathbf{x}}$ is the number of events observed at pixel $\mathbf{x}$ over the interval $[0, \Delta T]$.

The density of the point set $\mathcal{A} = [(a_1^{\mathbf{x}}, a_1^t), \dots, (a_N^{\mathbf{x}}, a_N^t)]$ is [5]

$$q_{\mathcal{A}}(\mathcal{A}) = \exp\Big(-\sum_{\mathbf{x}\in\mathcal{X}} \int_0^{\Delta T} \lambda(\mathbf{x}, t) dt\Big) \prod_{i=1}^{N} \lambda(a_i^{\mathbf{x}}, a_i^t) \quad (1)$$

$$= \exp\Big(-\sum_{\mathbf{x}\in\mathcal{X}} \lambda_{\mathbf{x}}\Big) \prod_{i=1}^{N} \Delta T^{-1} \lambda_{a_i^{\mathbf{x}}} \quad (2)$$

$$= \Delta T^{-N} \prod_{\mathbf{x}\in\mathcal{X}} \lambda_{\mathbf{x}}^{k_{\mathbf{x}}} \exp(-\lambda_{\mathbf{x}}). \quad (3)$$

In the second line, we used $\int_0^{\Delta T} \lambda(\mathbf{x}, t) dt = \int_0^{\Delta T} \Delta T^{-1} \lambda_{\mathbf{x}} dt = \lambda_{\mathbf{x}}$. In the third line, we grouped events with $a_i^{\mathbf{x}} = \mathbf{x}$. These simplifications are possible because space is discrete and the intensity function is homogeneous with respect to time.

Observe that the probability $p_{\mathcal{A}}(\mathcal{A})$ of the pixel counts as defined in the main text is related to $q_{\mathcal{A}}(\mathcal{A})$ by

$$p_{\mathcal{A}}(\mathcal{A}) = \frac{\Delta T^N}{\prod_{\mathbf{x}} k_{\mathbf{x}}!} q_{\mathcal{A}}(\mathcal{A}) = \Big(\prod_{\mathbf{x}} \frac{\Delta T^{k_{\mathbf{x}}}}{k_{\mathbf{x}}!}\Big) q_{\mathcal{A}}(\mathcal{A}). \quad (4)$$

The extra factor of $\prod_{\mathbf{x}} \frac{\Delta T^{k_{\mathbf{x}}}}{k_{\mathbf{x}}!}$ comes from integrating over all possible ordered sets of time indices $t_1, \dots t_{k_{\mathbf{x}}} \in [0, \Delta T]$ for the $k_{\mathbf{x}}$ points for each pixel $\mathbf{x}$, and then dividing by $k_{\mathbf{x}}!$ to switch from an ordered tuple to an unordered set.

2.2. Density of observed events $\mathcal{O}$

The Poisson mapping theorem describes what happens when the points of a Poisson process are mapped by a deterministic mapping: the result is a new Poisson process with modified intensity function. Let f_t be the ground-truth mapping from reference coordinates to camera coordinates at time t. Assume for now that f_t is a bijection on $\mathcal{X}$ for all t, as is the case for rotations. We discuss relaxations of this assumption below. Let S be the joint mapping on space and time that sends $(\mathbf{x}, t)$ to $(f_t(\mathbf{x}), t)$, so the ith observed event is obtained from the ith aligned event as $o_i = S(a_i)$. By the Poisson mapping theorem, the observed point set $\mathcal{O} = S(\mathcal{A}) \doteq [S(a_1), \dots, S(a_n)]$ is distributed according to a Poisson process with intensity function

$$\lambda'(\mathbf{x}, t) = \lambda\big(S^{-1}(\mathbf{x}, t)\big) = \lambda\big(f_t^{-1}(\mathbf{x}), t\big) \quad (5)$$

$$= \Delta T^{-1} \lambda_{f_t^{-1}(\mathbf{x})}. \quad (6)$$

In more general settings, a Jacobian term is required to adjust for changes of volume. In our case, it is not needed because the spatial coordinate is discrete, and the time coordinate mapping is the identity, which has unit Jacobian.

The density of the mapped point set $\mathcal{O}$ is therefore

$$q_{\mathcal{O}}(\mathcal{O}) = \exp\Big(-\sum_{\mathcal{X}} \int_0^{\Delta T} \lambda'(\mathbf{x}, t) dt\Big) \prod_{i=1}^{N} \lambda'(o_i^{\mathbf{x}}, o_i^t) \quad (7)$$

$$= \exp\Big(\sum_{\mathbf{x}\in\mathcal{X}} \lambda_{\mathbf{x}}\Big) \prod_{i=1}^{N} \Delta T^{-1} \lambda_{f_t^{-1}(o_i^{\mathbf{x}})} \quad (8)$$

$$= q_{\mathcal{A}}\big(S^{-1}(\mathcal{O})\big) \quad (9)$$

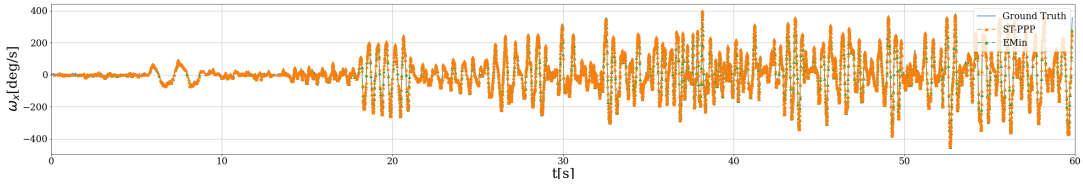
In the second line, we used $\sum_{\mathbf{x}\in\mathcal{X}} \lambda_{f_t^{-1}(\mathbf{x})} = \sum_{\mathbf{x}\in\mathcal{X}} \lambda_{\mathbf{x}}$, which follows because f_t is a bijection.

By aggregating to counts in the same manner described above, we obtain the result used in the main text:

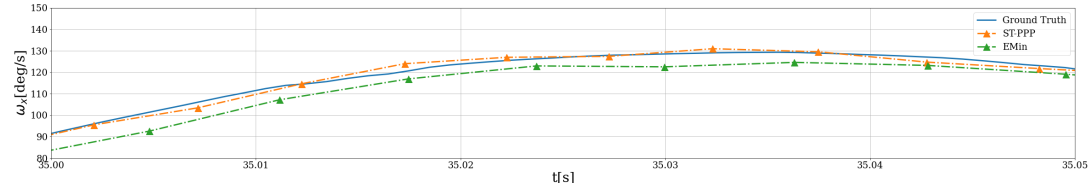
$$p_{\mathcal{O}}(\mathcal{O}) = p_{\mathcal{A}}\big(S^{-1}(\mathcal{O})\big). \quad (10)$$

Our method parameterizes the *inverse* mapping (i.e. from observed events to aligned ones) as $S^{-1} \approx R_\omega$, so that $p_{\mathcal{O}}(\mathcal{O}|\omega) = p_{\mathcal{A}}\big(R_\omega(\mathcal{O})\big)$.

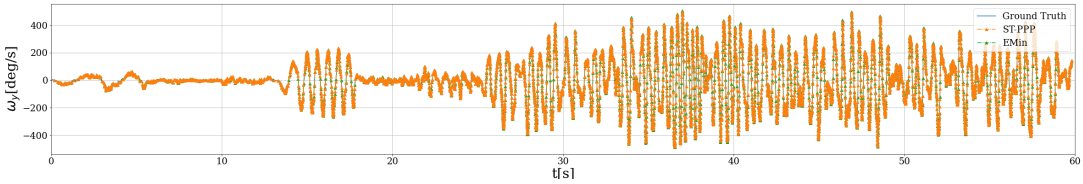
Changes of volume. When the camera movement is a rotation, it is true that the mapping f_t is a bijection on the discrete pixel set $\mathcal{X}$. For more general motions, even if the underlying continuous mapping is bijective, the discrete mapping may fail to be so because volume is not preserved, causing source pixels to stretch or compress so that many or no source pixels maps to a particular destination pixel $\mathbf{x}$. The derivation above then becomes ambiguous because the inverse image $f_t^{-1}(\mathbf{x})$ may be a set of any size, including zero. The ambiguity can be resolved cleanly whenever the underlying continuous mapping is bijective by describing the entire point process in continuous spatial coordinates and correcting for the change of volume by the usual



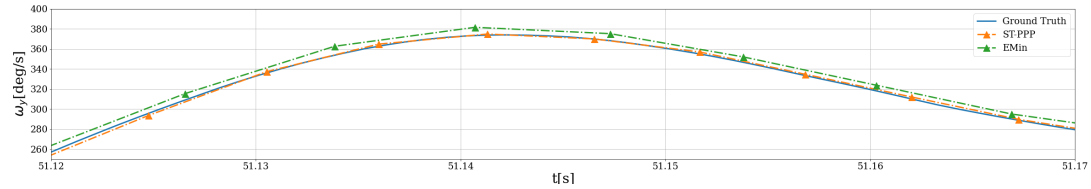
(a) Angular velocity around x-axis: full sequence (60sec)



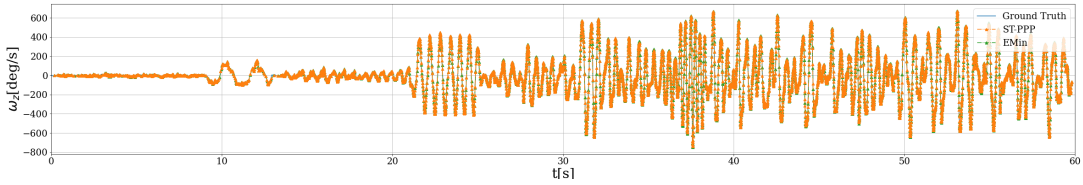
(b) Zoomed-in plots of corresponding bounded regions



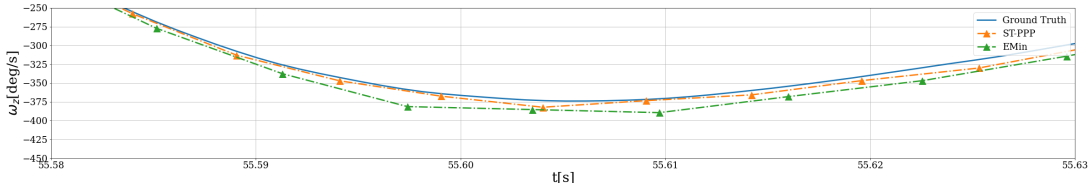
(c) Angular velocity around y-axis: full sequence (60sec)



(d) Zoomed-in plots of corresponding bounded regions

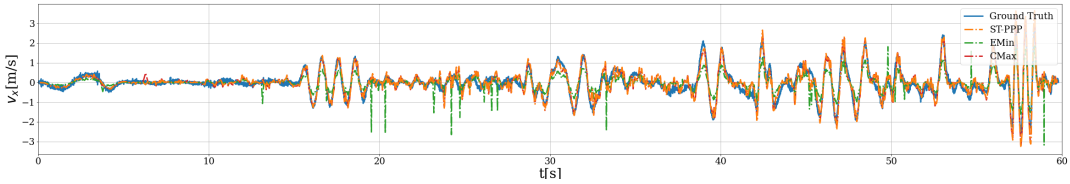


(e) Angular velocity around z-axis: full sequence (60sec)

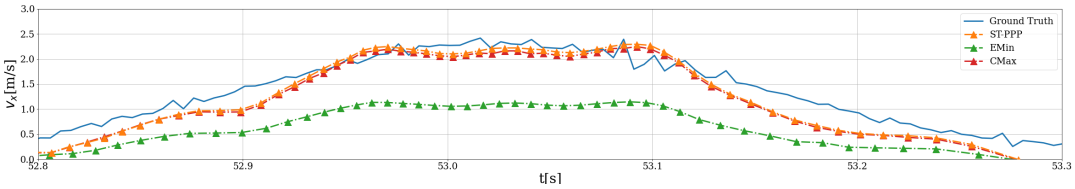


(f) Zoomed-in plots of corresponding bounded regions

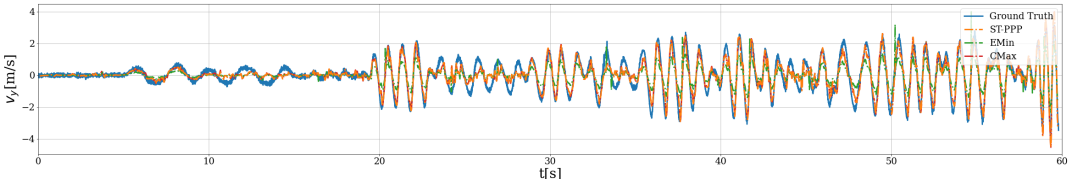
Figure 2. **Angular velocity estimates** measured in deg/sec plotted versus ground truth (IMU). Example *boxes_rotation*. Comparison to the next best performing method (EMin).



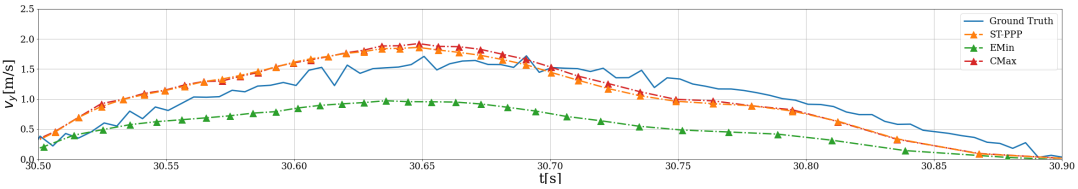
(a) Linear velocity around x-axis: full sequence (60sec)



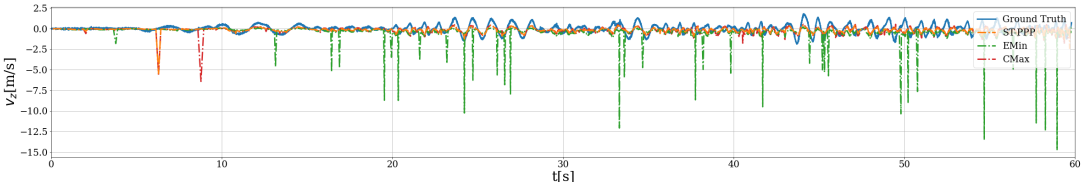
(b) Zoomed-in plots of corresponding bounded regions



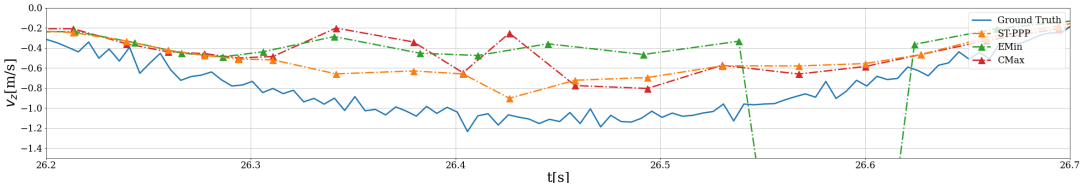
(c) Linear velocity around y-axis: full sequence (60sec)



(d) Zoomed-in plots of corresponding bounded regions



(e) Linear velocity around z-axis: full sequence (60sec)



(f) Zoomed-in plots of corresponding bounded regions

Figure 3. **Linear velocity estimates** measured in m/s plotted versus ground truth from motion capture system. Example sequence *boxes_translation*. Comparison to EMin [4], CMax [1] and ground truth.

change-of-variables formula involving the determinant of the Jacobian. We leave this direction for future work. In our experiments with translations (non-volume-preserving mapping) we simply transform pixel coordinates of events, ignoring the change of area due to translations along the Z camera axis (stretching, compressing transformations).

3. Code

Along with this supplementary material we provide code including a demo script, that runs the alignment on a small example sequence of 30k events. The code requires python3 and can be run optionally using GPU. More details regarding executing the code can be found in the README going with the code. Code: https://github.com/pbideau/Event-ST-PPP

References

[1] 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 *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 3867–3876, 2018. 1, 4

[2] Min Liu and Tobi Delbruck. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In *British Mach. Vis. Conf. (BMVC)*, 2018. 1

[3] 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. *Int. J. Robot. Research*, 36(2):142–149, 2017. 1

[4] Urbano Miguel Nunes and Yiannis Demiris. Entropy minimisation framework for event-based vision model estimation. In *Eur. Conf. Comput. Vis. (ECCV)*, 2020. 1, 4

[5] Roy L. Streit. *Poisson Point Processes*. Springer, 2010. 2