2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Continuous-Time Gaussian Process Motion-Compensation for Event-vision Pattern Tracking with Distance Fields

Cedric Le Gentil[1], Ignacio Alzugaray[2] and Teresa Vidal-Calleja[1]

Abstract— **This work addresses the issue of motion compensation and pattern tracking in event camera data. An event camera generates asynchronous streams of *events* triggered independently by each of the pixels upon changes in the observed intensity. Providing great advantages in low-light and rapid-motion scenarios, such unconventional data present significant research challenges as traditional vision algorithms are not directly applicable to this sensing modality. The proposed method decomposes the tracking problem into a local SE(2) motion-compensation step followed by a homography registration of small motion-compensated event batches. The first component relies on Gaussian Process (GP) theory to model the continuous occupancy field of the events in the image plane and embed the camera trajectory in the covariance kernel function. In doing so, estimating the trajectory is done similarly to GP hyperparameter learning by maximising the log marginal likelihood of the data. The continuous occupancy fields are turned into distance fields and used as templates for homography-based registration. By benchmarking the proposed method against other state-of-the-art techniques, we show that our open-source implementation performs high-accuracy motion compensation and produces high-quality tracks in real-world scenarios.**

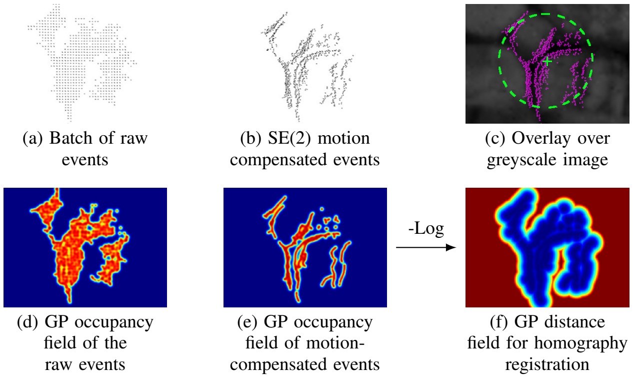


Fig. 1. The proposed pattern tracking method performs motion-compensation of event batches ((a) and (b)) using a Gaussian Process (GP) continuous representation of the events' occurrences in the image plane (occupancy fields in (d) and (e)). Applying the logarithm to the occupancy field yields a continuous distance field (f) later used for homography registration of different event batches. For visualisation only: (e) shows the compensated events over the corresponding greyscale image (green circle is the area of interest), and the colours in (f) are saturated above a certain threshold to show the details in the vicinity of the events.

I. Introduction

In the context of Simultaneous Localisation And Mapping (SLAM), robotic systems have been built upon a variety of exteroceptive sensors throughout the years. Vision-based SLAM systems generally consist of two main components that are the front-end, dealing with visual keypoints detection, tracking and/or matching; and the back-end, which estimates the system's pose and keypoints location in 3D. Recently, the robotics community started to investigate the use of *event cameras* for state estimation. Unlike standard "frame-based" cameras that trigger data acquisition of millions of pixels at specific, discrete timestamps, event cameras do not provide regular snapshots of the environment. They produce an asynchronous stream of *events* driven by changes of intensity observed independently by each of the cameras' pixels. Each event corresponds to a tuple of four values that represents the direction of the light intensity change compared to a given threshold (referred to as polarity), the timestamp of that change, and the x-y location of the pixel in the image plane. Note that, as event cameras only react to changes, events generated in a static environment with constant illumination are produced solely due to the motion of the camera and concentrate on highly salient scene regions projected on the image plane. Reacting only to small and incremental changes, event cameras exhibit an extended dynamic range and extreme resilience to motion blur in comparison with their frame-based counterparts. This unconventional sensing modality makes event-based state estimation an open research topic where mature techniques for frame-based vision cannot be directly applied, especially when it comes to the front-end side of SLAM. In this paper, we propose a novel method to address the issue of pattern tracking in the event stream.

As event generation and camera motion are correlated, the proposed method models the events-only pattern tracking as a motion estimation problem. This is also commonly denoted as motion compensation, or unwarping. Our method first corrects locally the flow of events in small batches assuming Special Euclidean group in 2D (SE(2)) motion in the image plane. By modelling the spatiotemporal (x-y-time) occupancy of the events with a Gaussian Process (GP) that embeds the camera motion in its covariance kernel, the method estimates the SE(2) trajectory by maximising the log marginal likelihood of the data similarly to the standard kernel hyperparameter learning. Fig. 1 shows the event occupancy field at given timestamps before and after motion estimation. Turning an occupancy field into a distance field (Fig. 1(f)) allows for the query of the distance to

This work was partially supported by the Australian Research Council Discovery Project under Grant DP210101336.

[1]C. Le Gentil and T. Vidal-Calleja are with the Robotics Institute at the University of Technology Sydney, Australia: {cedric.legentil ; teresa.vidalcalleja}@uts.edu.au

[2]I. Alzugaray is with the Department of Computing, Imperial College London, UK: i.alzugaray@imperial.ac.uk

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10160768

979-8-3503-2365-8/23/$31.00 ©2023 IEEE

the closest motion-compensated events at any location in the image plane. Long-term pattern tracking is achieved by estimating homography between batches of SE(2)-motion-compensated events employing the above-mentioned distance fields. The contributions associated with this work are:

- The formulation of the continuous-time motion-compensation of event data as a hyperparameter learning optimisation based on the GP theory.
- The introduction of a patch registration algorithm for event data using continuous GP distance fields.
- A set of experiments analysing the performance of the proposed, open-source method [1] compared to other open-source state-of-the-art alternatives.

II. Related Work

Monocular event-based SLAM and Visual Odometry (VO) have been active topics of research since the conception of event cameras. While early approaches on the topic tackle simplified scene geometry and camera motions [1, 2], first works achieving 6-Degree-of-Freedom (DoF) event-based monocular SLAM and VO [3, 4] opt for a direct approach where the definition of explicit visual features is avoided and thus the need for data association is circumvented. Event-based visual features are arguably more challenging to be detected and tracked than their traditional frame-based counterpart as motion and appearance are intertwined in the event stream [5]. Consequentially, many approaches to indirect, feature-based VO for event cameras rely on complementary sensing modalities such as traditional frame-based cameras in [6, 7, 8], where corners and edges are extracted from intensity images and used as templates to be tracked in the event stream. More complex approaches such as [9, 10] employ generative models [11] that correlate motion, intensity and event data generation, leading to the definition of feature-based VO systems such as the one described in [12].

Several works have devised the use of image-like event-frames or other time-invariant representations [13] on which traditional frame-based techniques to feature detection and tracking can be easily adapted. For instance, the method from [14] infers intensity gradient images from batches of events whereas the algorithm in [15] generates image-like edge-maps that are updated at each new event and on which traditional Harris corners [16] are detected. Similarly, the event-based VO systems described in [17, 18] use an Inertial Measurement Unit (IMU)-aided motion-compensation scheme to also generate event-frames where traditional frame-based FAST corners [19] are detected and tracked using KLT [20]. Initially explored in [21] and further formalised in [22, 23], motion-compensation is one of the most prevalent frameworks in the event-based literature as it correlates motion and event generation. Similarly to [17], the VO approach presented in [24] also relies on event-frames to detect [16] corners, but establishes data association based on a per-feature Expectation-Maximization (EM) motion-compensation scheme [25]. Approaches such as [26, 27] also apply motion-compensation techniques to track features in the event stream, but model the feature's motion using a continuous-time formulation in contrast to [24, 25, 17]. As event motion-compensation frameworks increase in popularity, other similar alternatives have been proposed emphasizing specific aspects such as robustness [28, 29], efficiency [30] or global optimality [31, 32].

Significant research effort has been also dedicated to the definition of event-driven features [5], i.e. features that can be detected and tracked by processing each event individually and without any form of event batching. While most of the literature on this topic is focused on corner-events, their detection [33, 34, 35, 36, 37, 38, 39] and tracking [40, 41, 42, 43, 44], corner-events are still to be successfully applied to VO due to their limited reliability. Nonetheless, emerging event-driven pattern-tracking techniques as the ones described in [45, 46] have found their way to VO systems as the ones described in [47, 48]. Another promising approach is the use of event-based line features [49, 50, 51, 52] which exploits the natural sensitivity of event cameras to edges and for which a significant number of event-based VO systems have been proposed in recent years [53, 54, 55].

Among all these event-vision works, the proposed method finds similarities in the continuous-time motion-compensation of [26], and the kernel-based optimisation in [30] where the kernel represents a "potential" and the entropy of the data is minimised. Our method leverages GPs regression [56] over SE(2) to characterise the continuous motion as opposed to a 2-DoFs (translation in the image plane) Bezier parameterisation in [26]. Various works use GPs to perform efficient long-term state estimation using motion priors to derive exactly sparse GP kernels as in [57, 58, 48]. In the proposed method, we opted for an approach similar to [59] where the inducing values of the GP state are estimated. Embedding these GP trajectories in a spatiotemporal kernel that characterises the covariance between events, we can optimise the image-plane motion by maximising the marginal likelihood of the events as for the traditional GP regression hyperparameter learning [56]. When employing the proposed motion-compensated scheme to pattern tracking, a key element of the process is the proposed homography registration algorithm that relies on continuous GP-based distance fields. Given the GP event occupancy field in the SE(2)-corrected patches, distance fields can be obtained by simply applying a negative logarithm as shown in [60] in the context of implicit surface [61, 62].

III. Method

The proposed method addresses the motion compensation of events and, specifically, applies it to the problem of pattern tracking. Starting from a given seed (x-y coordinates in the image plane at a given time t), our pipeline first performs motion compensation using a SE(2) motion model over a small batch of events in the spatio-temporal vicinity of the seed. The motion-compensated events are used to create a template

[1] https://github.com/UTS-CAS/gp_comet

that will be used for later batch-to-template registration. The seed location is updated based on the estimated motion and is used to collect the next batch of events. For the second batch and the subsequent ones, the motion compensation is performed followed by a homography registration with respect to the previous batch and the template. The template is dynamically updated with the last registered motion-compensated event batch. The data pipeline of the proposed method is shown in Fig. 2. This framework relies on the assumption that a continuous-time SE(2) motion in the image plane can explain the generation of events over small periods of time (i.e., locally), and that homography transformations are sufficient to model the appearance changes of patches over longer time periods (i.e., globally).

A. GP regression preliminaries

GP regression [56] is a kernel-based, probabilistic interpolation method that allows for data-driven inferences of an unknown signal given a set of noisy observations. The so-called kernel function represents the covariance between two instantiations of the signal and is essentially the only assumption about the signal. Let us consider $h(x)$ as a function of $x \in \mathbb{R}$, with noisy observations $y_i = h(x_i) + \eta$ where $\eta \sim \mathcal{N}(0, \sigma_y^2)$ and $i = 1, \cdots, N$. By modelling $h(x)$ with a GP as $h(x) \sim \mathcal{GP}\big(0, k_h(x, x')\big)$, with $k_h(x, x') = \mathrm{cov}\big(h(x), h(x')\big)$ being the covariance kernel, it is possible to infer any new point $h^*(x)$ and its variance $\mathrm{var}\big(h^*(x)\big)$ as

$$\begin{aligned} h^*(x) \sim \mathcal{N}\Big(&\mathbf{k}_h(x,\mathbf{x})\big[\mathbf{K}_h(\mathbf{x},\mathbf{x}) + \sigma_y^2\mathbf{I}\big]^{-1}\mathbf{y}, \qquad (1)\\ &k_h(x,x)\text{-}\ \mathbf{k}_h(x,\mathbf{x})\big[\mathbf{K}_h(\mathbf{x},\mathbf{x}) + \sigma_y^2\mathbf{I}\big]^{-1}\mathbf{k}_h(\mathbf{x},x)\Big), \end{aligned}$$

where $\mathbf{y} = \big[y_1 \ \cdots \ y_N)\big]^\top$, $\mathbf{x} = \big[x_1 \ \cdots \ x_N\big]^\top$, $\mathbf{k}_h(x,\mathbf{x}) = \big[k_h(x,x_1) \ \cdots \ k_h(x,x_N)\big]$, $\mathbf{k}_h(\mathbf{x},x) = \mathbf{k}_h(x,\mathbf{x})^\top$, and $\mathbf{K}_h(\mathbf{x},\mathbf{x}) = \big[\mathbf{k}_h(\mathbf{x},x_1) \ \cdots \ \mathbf{k}_h(\mathbf{x},x_N)\big]$. Note that in the rest of this paper, the vector $\mathbf{y}$ will be also referred to as the set of *inducing values*.

The kernel function generally depends on a set of *hyperparameters* that we denote Θ_h. For example, the square exponential kernel possesses two hyperparameters which are a scaling factor and a lengthscale. While the hyperparameters can be set arbitrarily with an educated guess, they can also be learnt directly from the data. An approach to hyperparameter learning is to maximise the log marginal likelihood of the observations of h with respect to the value of the hyperparameters, $\underset{\Theta_h}{\mathrm{argmax}} \ \log(P(\mathbf{x}|\mathbf{y},\Theta_h))$, with

$$\begin{aligned} \log(P(\mathbf{y}|\mathbf{x},\Theta_h)) = &-\frac{1}{2}\mathbf{y}^\top(\mathbf{K}_h(\mathbf{x},\mathbf{x}) + \sigma_y^2\mathbf{I})^{-1}\mathbf{y}\\ &-\frac{1}{2}\log(|\mathbf{K}_h(\mathbf{x},\mathbf{x}) + \sigma_y^2\mathbf{I}|) - \frac{N}{2}\log(2\pi). \end{aligned} \qquad (2)$$

For more details about GP regression and hyperparameter learning, we invite the reader to refer to [56].

B. System's description and definitions

Let us consider an event camera providing events at timestamps t_i ($i = 1, \ldots, N$). For each event, we denote $\mathbf{e}^i_{t_i}$ its x-y coordinate vector in the image plane at time t_i. The subscript denotes the timestamp in which the coordinates of the event are expressed. Locally assuming a SE(2) motion in the image plane, and given some knowledge of that motion, an event can be transformed to a different timestamp as $[\bar{\mathbf{e}}^i_{t_j}{}^\top \ 1]^\top = \mathbf{T}^{t_i}_{t_j}[\mathbf{e}^i_{t_i}{}^\top \ 1]^\top$, with $\bar{\mathbf{e}}^i_{t_j}$ the SE(2)-transformed event, and $\mathbf{T}^{t_i}_{t_j} \in \mathbb{R}^{3\times3}$ the homogeneous transformation matrix that characterises the SE(2) motion in the image plane between timestamps t_j and t_i. This transformation matrix is associated with the rotation $r^{t_i}_{t_j}$ and translation $\mathbf{p}^{t_i}_{t_j} = [{}_xp^{t_i}_{t_j} \ {}_yp^{t_i}_{t_j}]^\top$ as $\mathbf{T}^{t_i}_{t_j} = \begin{bmatrix} \mathbf{R}^{t_i}_{t_j} & \mathbf{p}^{t_i}_{t_j} \\ \mathbf{0} & 1 \end{bmatrix}$, with $\mathbf{R}^{t_i}_{t_j} = \begin{bmatrix} \cos(r^{t_i}_{t_j}) & -\sin(r^{t_i}_{t_j}) \\ \sin(r^{t_i}_{t_j}) & \cos(r^{t_i}_{t_j}) \end{bmatrix}$.

Similarly, an event $\mathbf{e}^i_{t_i}$ can be projected via homography on the image plane, $\hat{\mathbf{e}}^i_{t_j}$, at any timestamp t_j as $\hat{\mathbf{e}}^i_{t_j} = \pi_{\mathbf{H}^{t_i}_{t_j}}(\mathbf{e}^i_{t_i}) = [\frac{{}_xe^i_{t_j}}{{}_we^i_{t_j}} \ \frac{{}_ye^i_{t_j}}{{}_we^i_{t_j}}]^\top$, with $[{}_xe^i_{t_j} \ {}_ye^i_{t_j} \ {}_we^i_{t_j}]^\top = \mathbf{H}^{t_i}_{t_j}[\mathbf{e}^i_{t_i} \ 1]^\top$, and $\mathbf{H}^{t_i}_{t_j} \in \mathbb{R}^{3\times3}$ the homograhy matrix.

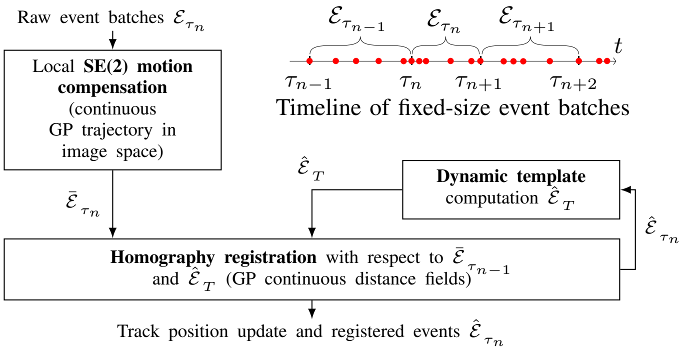

Raw event batches $\mathcal{E}_{\tau_n}$
Local **SE(2) motion compensation** (continuous GP trajectory in image space)
$\bar{\mathcal{E}}_{\tau_n}$
$\mathcal{E}_{\tau_{n-1}}$
$\mathcal{E}_{\tau_n}$
$\mathcal{E}_{\tau_{n+1}}$
t
τ_{n-1}
τ_n
τ_{n+1}
τ_{n+2}
Timeline of fixed-size event batches
$\hat{\mathcal{E}}_T$
Dynamic template computation $\hat{\mathcal{E}}_T$
$\hat{\mathcal{E}}_{\tau_n}$
Homography registration with respect to $\bar{\mathcal{E}}_{\tau_{n-1}}$ and $\hat{\mathcal{E}}_T$ (GP continuous distance fields)
Track position update and registered events $\hat{\mathcal{E}}_{\tau_n}$

Fig. 2. Overview of the proposed event-only pattern tracking method (event timeline in upper right with red dots as events' timestamps).

C. Continuous-time SE(2) event data motion-compensation

This subsection details the proposed SE(2) motion-compensation mechanism for a set of raw events $\mathcal{E}_{\tau_n}$. This set contains a fixed number of events occurring after time τ_n as illustrated in Fig. 2. Based on the assumption that the events in $\mathcal{E}_{\tau_n}$ are generated by a SE(2) camera motion parallel to the image plane, let us model the trajectory rotation $r^t_{\tau_n}$ and position $\mathbf{p}^t_{\tau_n}$ with three independent GPs

$$r^t_{\tau_n}(t) \sim \mathcal{GP}\left(0, k_r(t,t')\right) \qquad (3)$$

$${}_\bullet p^t_{\tau_n}(t) \sim \mathcal{GP}\left(0, k_{\bullet p}(t,t')\right), \qquad (4)$$

where k_r and k_p are the associated covariance kernels. The *inducing values* of these GPs (i.e. the set of "observations" y_i in (1) at arbitrarily chosen timestamps t_i with $i = 1, \cdots, Q$) are denoted $\mathcal{R}_n$ and $\mathcal{P}_n$. The associated continuous-time transformation matrix is denoted $\mathbf{T}^t_{\tau_n}(t)$, which is "parameterised" by $\mathcal{R}_n$ and $\mathcal{P}_n$.

Now, let us model the occurrence of events with another GP (in an implicit-surface-like manner) with

$$f\left(\begin{bmatrix}\mathbf{e}\\t\end{bmatrix}\right) \sim \mathcal{GP}\left(0, k_f\left(\begin{bmatrix}\mathbf{e}\\t\end{bmatrix}, \begin{bmatrix}\mathbf{e}'\\t'\end{bmatrix}\right)\right). \qquad (5)$$

The associated observations are the event coordinates and timestamps in $\mathcal{E}_{\tau_n}$ as inputs and the constant value *one* as outputs. Thus, f represents the continuous occupancy field of the image plane after motion-compensation (according to a continuous-time SE(2) model) of the event stream. This is illustrated in Fig. 1 (a,b,d,e). The kernel function k_f embeds the continuous-time SE(2) trajectory defined in (3) and (4):

$$k_f\left(\begin{bmatrix}\mathbf{e}\\t\end{bmatrix},\begin{bmatrix}\mathbf{e}'\\t'\end{bmatrix}\right)=\sigma_f \exp\left(-\frac{\|\mathbf{T}^{t}_{\tau_n}(t)\mathbf{e}-\mathbf{T}^{t'}_{\tau_n}(t')\mathbf{e}'\|^2}{2l_f^2}\right).$$

Intuitively the kernel k_f corresponds to a standard square exponential kernel over motion-compensated events. As this kernel relies on the knowledge of the SE(2) trajectory, it depends on the GP inducing values in $\mathcal{R}_n$ and $\mathcal{P}_n$. In other words, the inducing values in $\mathcal{R}_n$ and $\mathcal{P}_n$ can be considered as hyperparameters of the kernel function k_f. The proposed SE(2) compensation scheme optimises the trajectory as a hyperparameter-learning problem by maximising the marginal likelihood of the event data given the hyperparameters $\Theta_n = \{\mathcal{R}_n, \mathcal{P}_n\}$ as in (2):

$$\Theta_n^* = \underset{\Theta_n}{\operatorname{argmax}}\ \log(P(\mathbf{y}|\mathcal{E}, \mathbf{t}, \Theta_n)), \tag{6}$$

with $\mathbf{t}$ the vector of event timestamps in $\mathcal{E}_{\tau_n}$. In the rest of this paper, $\bar{\mathcal{E}}_{\tau_n}$ will refer to the set of events in $\mathcal{E}_{\tau_n}$ after SE(2) motion-compensation using the estimated trajectory parameters Θ_n^*, (3), and (4).

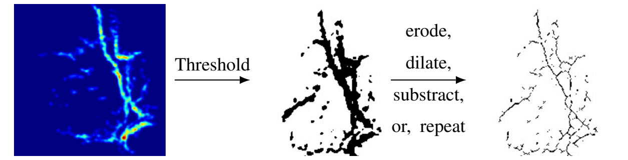
D. Homography registration

Let us consider two sets of locally-SE(2)-compensated events $\bar{\mathcal{E}}_{\tau_A}$ and $\bar{\mathcal{E}}_{\tau_B}$. Assuming that both sets contain observations of a single planar patch, the proposed method estimates the homography transformation $\mathbf{H}^{\tau_B}_{\tau_A}$ between the two sets using distance fields in the image space. Similarly to (5), we model the continuous occupancy field associated to $\bar{\mathcal{E}}_{\tau_A}$ as $g(\mathbf{e}) \sim \mathcal{GP}(0, k_g(\mathbf{e}, \mathbf{e}'))$ with k_g the corresponding spatial covariance kernel. It is possible to infer a continuous distance field d as the negative logarithm of the occupancy field g similarly to [60] or [63]: $d(\mathbf{e}) = -\log(g(\mathbf{e}))$. Note that the occupancy field $g(\mathbf{e})$ is providing values between zero (GP means) and one (observation values). As illustrated in Fig. 1 (e,f), taking the negative logarithm of such a field provides a distance-to-closest-event field that is equal to zero around events and growing monotonously when inferring further. By querying the distance field for the events in $\bar{\mathcal{E}}_{\tau_B}$ projected at time τ_A with $\mathbf{H}^{\tau_B}_{\tau_A}$, it is possible to build a non-linear least-square cost function C to register $\bar{\mathcal{E}}_{\tau_B}$ to $\bar{\mathcal{E}}_{\tau_A}$ by estimating $\mathbf{H}^{\tau_B}_{\tau_A}$:

$$C(\mathbf{H}^{\tau_B}_{\tau_A}) = \sum_{\bar{\mathbf{e}}^i_{\tau_B} \in \bar{\mathcal{E}}_{\tau_B}} \left(d(\pi_{\mathbf{H}^{\tau_B}_{\tau_A}}(\bar{\mathbf{e}}^i_{\tau_B}))^2\right). \tag{7}$$

E. Pattern Tracking

As illustrated in Fig. 2, the proposed method combines the SE(2) continuous-time motion-compensation to locally compensate event batches and then performs discrete homography registration between batches to track patterns over time. Performing solely sequential *batch-to-batch* homography registration between $\bar{\mathcal{E}}_{\tau_{n-1}}$ and $\bar{\mathcal{E}}_{\tau_n}$ inherently results in the accumulation of drift over time as different regions in the tracked pattern generate events depending on its motion direction in the image plane. The proposed method includes the use of a refined-on-the-fly template $\bar{\mathcal{E}}_T$ that is augmented as new batches are co-registered. The template is built upon an image-like histogram of the accumulation of all the motion-compensated events that are spatiotemporally close to the track similar to [45, 46]. By binarising and skeletonising it using morphological operations [64] as shown in Fig. 3, our method generates virtual events $\hat{\mathcal{E}}_T$ that can be used to generate a distance field $d(\mathbf{e})$.To robustify the homography estimation, both constraints from $\bar{\mathcal{E}}_{\tau_n}$ to $\bar{\mathcal{E}}_{\tau_{n-1}}$ using $\mathbf{H}^{\tau_n}_{\tau_{n-1}}$, and from $\bar{\mathcal{E}}_{\tau_{n-1}}$ to $\bar{\mathcal{E}}_{\tau_n}$ using ${\mathbf{H}^{\tau_n}_{\tau_{n-1}}}^{-1}$ are considered in the cost function (7). This prevents the collapse of events into a single location when the pattern texture provides weak constraints on some of the 8 DoFs of the homography.



(a) Event accumulation (b) Binarised image (c) Skeleton / template

Fig. 3. Creation process of the dynamic template using skeletonisation.

Starting from a given location $\mathbf{s}_0$ in the image plane, the position of the track is updated by combining the continuous local SE(2) model and the discrete homographies as $\mathbf{s}(t) = \mathbf{T}^{\tau_{n\text{-}1}}_{t}(t)[\pi_{\mathbf{H}^{\tau_0}_{\tau_{n\text{-}1}}}(\mathbf{s}_0)] + (t\text{-}\tau_{n\text{-}1})\alpha$ with $\alpha = (\pi_{\mathbf{H}^{\tau_0}_{\tau_n}}(\mathbf{s}_0) - \mathbf{T}^{\tau_{n\text{-}1}}_{\tau_n}[\pi_{\mathbf{H}^{\tau_0}_{\tau_{n-1}}}(\mathbf{s}_0)])/(\tau_n\text{-}\tau_{n\text{-}1})$. The proposed method stops tracking a pattern when the change of marginal likelihood of events after solving (6) is under a certain threshold (proxy for SE(2) motion-compensation failure), when the homography between two successive batches differs significantly from the SE(2) estimate (simple distance threshold between $\pi_{\mathbf{H}^{\tau_0}_{\tau_n}}(\mathbf{s}_0)$ and $\mathbf{T}^{\tau_{n\text{-}1}}_{\tau_n}[\pi_{\mathbf{H}^{\tau_0}_{\tau_{n-1}}}(\mathbf{s}_0)]$), or when the pattern is getting close to the image borders.

IV. Experiments

In this section, we analyse the accuracy of our motion compensation algorithm with simulated data before testing the full tracking pipeline (motion compensation and homography registration) with real-world data. The same set of parameters is used across all the experiments both in Section IV-A and IV-B. The SE(2) GP trajectories are modelled with inducing values every 250 events and a square exponential kernel (scaling factor equal to one and a lengthscale three times bigger than the distance between inducing values). Each batch used for SE(2) motion compensation contains 1250 events. The lengthscale and scaling factor of the occupancy fields are set to 0.25 and 1 respectively. The SE(2) local motion compensation is estimated using the BFGS solver and the homography registration is solved with the Levenberg-Marquardt algorithm and a Cauchy loss function.

A. Event-based motion-compensation

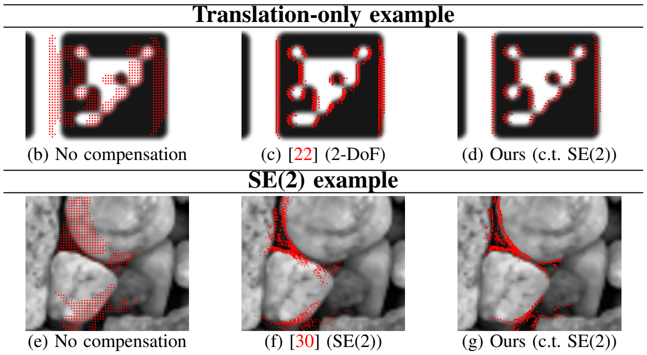
We provide a quantitative evaluation of the proposed continuous-time SE(2) motion compensation (later referred to as “c.t. SE(2)”) with a comparison against two other state-of-the-art motion-compensation approaches [22, 30]. We use simulated data due to the need for accurate groundtruth localisation and scene structure. The method in [22] relies on an improved version of the popular contrast-maximisation framework [65] by optimising over a combined objective function that rewards a large accumulation of events at a sparse set of locations. More closely related to ours, the technique described in [30] relies on a kernel to quantify a pair-wise “potential” between events and minimise the dispersion of the motion-compensated events. Both of these methods have open-source implementations to estimate the 2D translation (referred to as 2-DoF) in the image plane based on the assumption of constant velocity all along the event batch, but [30] also provides a SE(2) motion model. We use the ESIM simulator [66] to generate events in two planar environments (a lattice of April Tags and a picture of rocks as textures) with translation-only, constant-velocity trajectories (suitable for [22] and [30]-2-DoF) as well as randomly generated SE(2) trajectories.

To quantify the performance of the different methods, we use the reprojection error between the motion-compensated events and their groundtruth locations based on the known scene structure and camera poses. The results in Fig. 4 have been obtained by averaging the RMSE over the successful runs (out of 50) for each dataset. A run is considered successful if the RMSE of all events in the batch is inferior to 7 px. When successful, [22] and [30]-2-DoF provide similar levels of accuracy but [30] does not seem to always converge towards a good solution especially in the low-texture environment (lower success rate). When it comes to SE(2) motion, the accuracy of [22] and [30]-2-DoF expectedly decreases whereas [30]-SE2 retains certain level of accuracy. The proposed motion-compensation method consistently outperforms the alternatives by an order of magnitude for both trajectory types and across the different scenes. We believe that the difference in performance between our method and [30]-SE(2) can be mostly explained by the flexibility of our continuous-time motion model contrasting with the constant velocities assumption in [30]-SE(2).

While the proposed approach excels in motion-compensation accuracy compared to the alternatives, our method is computationally more demanding than [22] and [30] due to the cubic complexity of (2): around 25 s per batch with our method while [22] and [30] yield results in under a second. A simple way to reduce the complexity of the proposed motion-compensation algorithm is to reduce the number of events used in the optimisation. As an example, with straightforward uniform downsampling from 1250 to 400 events, our method leads to an error of 0.53 px in around 3 s for the April Tags SE(2) dataset. We believe that a more principled approach to downsampling and a multi-scale approach (coarse-to-fine) can significantly improve the computational efficiency of the proposed method. However, due to space limitation, this paper focuses on demonstrating the soundness of our novel GP-based approach to motion compensation and leaves as future work further developments to decrease the computational cost and the analysis of their analysis on the accuracy.

Dataset	[22] 2-DoF	[30] 2-DoF	[30] SE(2)	Ours c.t. SE(2)
April Tags 2D Translation	1.58 px (100%)	2.41 px (54%)	2.11 px (58%)	**0.25** px (100%)
April Tags SE(2)	4.21 px (100%)	2.76 px (94%)	2.53 px (98%)	**0.38** px (100%)
Rocks 2D Translation	2.05 px (100%)	1.66 px (78%)	1.76 px (78%)	**0.50** px (100%)
Rocks SE(2)	6.65 px (54%)	3.72 px (90%)	2.37 px (90%)	**0.71** px (100%)

(a) Motion-compensation accuracy and success rate in simulated data.



(b) No compensation (c) [22] (2-DoF) (d) Ours (c.t. SE(2))

(e) No compensation (f) [30] (SE(2)) (g) Ours (c.t. SE(2))

Fig. 4. Average RMS reprojection error and success rate (RMSE < 7 px) of event motion-compensation on simulated data with examples of motion-compensated event batches. (b) and (d) show the raw events (in red) over the corresponding grayscale images. (c,d,f,g) show motion-compensated events. Note that grayscale images are not used in any of the methods.

B. Event-based pattern tracking

We analyse the performance of the full tracking pipeline in real-world scenarios and compare our results with [24] and [46] on three 6-DoF datasets from [34] (`shapes_6dof`, `poster_6dof`, and `boxes_6dof`). Each dataset is one minute long and contains increasingly challenging camera motion. Both `shapes_6dof` and `poster_6dof` observe planar scenes with simple shape-based and realistic rock-like textures, respectively. The scene in `boxes_6dof` consists of a stack of textured boxes arranged in a pile. As a real 3D scene, this dataset contains numerous occlusions that can hinder the tracking process. In the absence of ground truth for pattern tracking evaluation in these datasets, we opt for the KLT-based tracking metric described in [10] that compares event-based feature tracks with tracks obtained in frame-based data using the KLT tracker[2]. Accordingly, we use only the first 25 seconds of each dataset to limit the KLT tracker performance degradation due to motion blur on intensity images. As we focus on tracking, we employ the event-based feature detections provided by [24] to initialise

[2] Note that, while this metric provides a reasonable ground for comparison across different approaches, the results are completely dependent on the quality of the intensity-based KLT tracks which are easily compromised during motion blur, in poorly illuminated scenarios, or in poorly textured areas. See [42, 45, 46, 39] for further discussion on the topic.

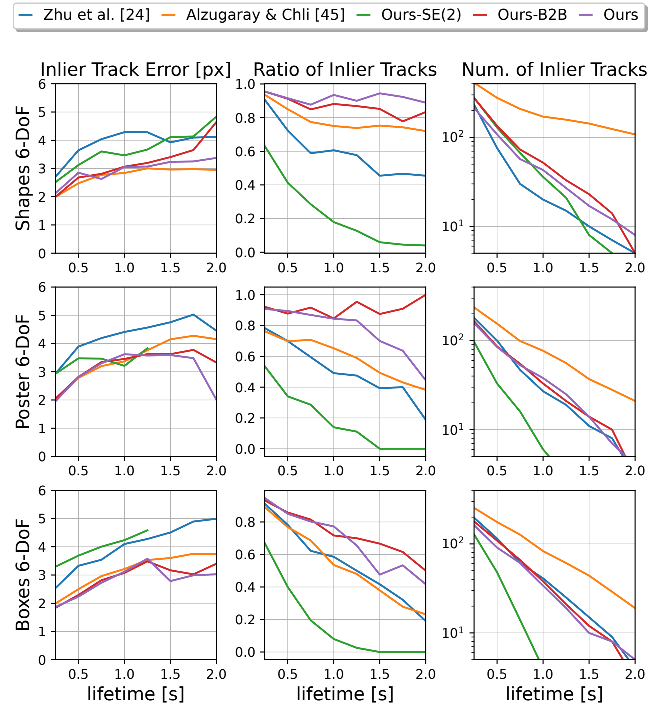


Fig. 5. Analysis of the proposed tracking pipeline using the KLT-based error metric in different datasets. All the plots are functions of the tracks' lifetime. The first column shows the average RMS error ↓ (in px) between the inlier tracks and KLT tracks. The second column is the ratio of inlier tracks ↑ and the third column is the actual number of inlier tracks ↑.

the tracks' starting point of all the compared methods for the sake of fairness. Atop the comparison with other methods, we conduct an ablation study by considering two additional versions of the proposed tracker that all perform local SE(2) motion compensation but differ in their batch registration strategies between motion-compensated batches of events. *Ours-SE(2)* uses directly the output of the SE(2) motion compensation to register the events from batch-to-batch (no homography estimation). *Ours-B2B* corresponds to the proposed pipeline as show in Fig. 2 but without the dynamic template (solely batch-to-batch homography). *Ours* is the full proposed pipeline with the refined-on-the-fly template.

While the proposed method and [24] are able to detect tracking failures on-the-go, thus, discarding unreliable feature tracks in an online fashion, the method from [46] does not implement such capability. Accordingly, we create a set of "inlier tracks" by cutting off tracks when the error goes above 10 px similarly to the evaluation in [46]. Fig. 5 reports the RMS tracking error of the inlier tracks (first column) as a function of their lifetime, as well as the ratio of inlier tracks among all the tracks (second column), and the actual number of inlier tracks (third column). In terms of tracking error, the proposed method and [46] offer similar levels of accuracy across the different scenes, and both significantly outperform [24]. *Ours* and [24] have approximately the same number of inlier tracks. While [46] retains the highest number of inlier tracks among all the methods, these inliers are only retrievable in this offline evaluation stage, as [46] does not implement any online outlier rejection policy. Both [24] and [46] display a ratio of inliers that rapidly decreases for longer track lifetimes. Our method keeps this ratio mostly above 80-90% in `shapes_6dof` and `poster_6dof`, and above 50% in `boxes_6dof`, indicating a smaller number of outliers in the tracks generated. This high ratio of inliers is key in downstream applications such as VO where only a minimal amount of outliers would have to be handled. Qualitative results are provided in the associated video. In summary, compared with existing methods, the proposed framework proved to be more robust (fewer outliers) in most scenarios while performing at a similar level of inlier accuracy.

In the ablation study, *Ours-SE(2)* significantly underperforms the other two versions. While the inlier tracks' accuracy is similar to those from [24], the ratio of inliers rapidly decreases. The decrease in accuracy of *Ours-SE(2)* compared to *Ours-B2B* and [46] shows the limits of the SE(2) model in the absence of a dynamic template ([46] also estimates a SE(2) motion but leverages a refined-on-the-fly template). The higher inlier ratio of *Ours-B2B* and *Ours* highlights the performance of our track-termination criteria that confronts the SE(2) and homography estimates. The use of the proposed refined-on-the-fly template does not seem to significantly affect the performance (*Ours-B2B* vs. *Ours*). We believe that the drift of our batch-to-batch homography registration is too small to observe any benefits from the refined-on-the-fly template for tracks under 2 s of lifetime.

V. Conclusion

In this paper, we presented both a novel event-based method for continuous-time motion-compensation built upon GP theory, and a tracking algorithm based on homography registration between motion-compensated batches. The first component models the occupancy of the events in the image plane with a GP that embeds the events' motion in its covariance kernel. The motion is estimated by maximising the marginal likelihood of the events with respect to the trajectory parameters. The homography registration leverages GP distance fields to minimise distances between motion-compensated events and a template. Via simulated and real-world experiments, we demonstrated state-of-the-art performance both for motion compensation and pattern tracking.

Future work will address the computational cost of the SE(2) motion-compensation (around 95% of the tracking computation time) by including the use of structured-kernel interpolation [67] which can significantly reduce the computational complexity with negligible impact on accuracy as demonstrated in [68].

References

[1] D. Weikersdorfer and J. Conradt, "Event-based particle filtering for robot self-localization," in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2012, pp. 866–870.

[2] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. Davison, "Simultaneous Mosaicing and Tracking with an Event Camera," in *Proceedings of the British*

Machine Vision Conference 2014. Nottingham: British Machine Vision Association, 2014, pp. 26.1–26.12.

[3] H. Kim, S. Leutenegger, and A. J. Davison, "Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 349–364, 2016.

[4] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza, "EVO: A Geometric Approach to Event-Based 6-DOF Parallel Tracking and Mapping in Real Time," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 593–600, 2017.

[5] I. Alzugaray, "Event-driven Feature Detection and Tracking for Visual SLAM," Doctoral Thesis, ETH Zurich, 2022, accepted: 2022-04-08T11:37:59Z. [Online]. Available: https://www.research-collection.ethz.ch/handle/20.500.11850/541700

[6] D. Tedaldi, G. Gallego, E. Mueggler, and D. Scaramuzza, "Feature detection and tracking with the dynamic and active-pixel vision sensor (DAVIS)," in *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*. Krakow, Poland: IEEE, June 2016, pp. 1–7.

[7] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, "Low-latency visual odometry using event-based feature tracks," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, South Korea: IEEE, Oct. 2016, pp. 16–23.

[8] K. Li, D. Shi, Y. Zhang, R. Li, W. Qin, and R. Li, "Feature Tracking Based on Line Segments With the Dynamic and Active-Pixel Vision Sensor (DAVIS)," *IEEE Access*, vol. 7, pp. 110 874–110 883, 2019.

[9] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "Asynchronous, Photometric Feature Tracking Using Events and Frames," in *European Conference on Computer Vision*, ser. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 766–781.

[10] ——, "EKLT: Asynchronous Photometric Feature Tracking Using Events and Frames," *International Journal of Computer Vision*, vol. 128, no. 3, pp. 601–618, Mar. 2020.

[11] G. Gallego, C. Forster, E. Mueggler, and D. Scaramuzza, "Event-based Camera Pose Tracking using a Generative Event Model," *arXiv:1510.01972 [cs]*, Oct. 2015.

[12] J. Hidalgo-Carrio, G. Gallego, and D. Scaramuzza, "Event-Aided Direct Sparse Odometry," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 10, 2022.

[13] J. Jiao, H. Huang, L. Li, Z. He, Y. Zhu, and M. Liu, "Comparing Representations in Tracking for Event Camera-Based SLAM," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021, pp. 1369–1376.

[14] P. Chiberre, E. Perot, A. Sironi, and V. Lepetit, "Detecting Stable Keypoints From Events Through Image Gradient Prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1387–1394.

[15] A. Glover, A. Dinale, L. D. S. Rosa, S. Bamford, and C. Bartolozzi, "luvharris: A practical corner detector for event-cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 10 087–10 098, 2022.

[16] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *In Proc. of Fourth Alvey Vision Conference*, pp. 147–152, 1988.

[17] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization," in *Procedings of the British Machine Vision Conference 2017*. London, UK: British Machine Vision Association, 2017, p. 16.

[18] A. Rosinol Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[19] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 430–443.

[20] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Aug. 1981, pp. 674–679.

[21] G. Gallego and D. Scaramuzza, "Accurate Angular Velocity Estimation With an Event Camera," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 632–639, Apr. 2017.

[22] T. Stoffregen and L. Kleeman, "Event Cameras, Contrast Maximization and Reward Functions: An Analysis," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 12 292–12 300.

[23] G. Gallego, T. Delbrruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based Vision : A Survey," pp. 1–30, 2019.

[24] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based feature tracking with probabilistic data association," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4465–4470, 2017.

[25] ——, "Event-based feature tracking with probabilistic data association," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4465–4470.

[26] H. Seok and J. Lim, "Robust Feature Tracking in DVS Event Stream using Bézier Mapping," in *2020 IEEE Winter Conference on Applications of Computer Vision*

(WACV). Snowmass Village, CO, USA: IEEE, Mar. 2020, pp. 1647–1656.

[27] J. Chui, S. Klenk, and D. Cremers, "Event-Based Feature Tracking in Continuous Time with Sliding Window Optimization," *arXiv:2107.04536 [cs]*, July 2021.

[28] U. M. Nunes and Y. Demiris, "Entropy Minimisation Framework for Event-Based Vision Model Estimation," in *European Conference on Computer Vision (ECCV)*, vol. 12350. Springer International Publishing, 2020, pp. 161–176.

[29] J. Xu, M. Jiang, L. Yu, W. Yang, and W. Wang, "Robust Motion Compensation for Event Cameras With Smooth Constraint," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 604–614, 2020, conference Name: IEEE Transactions on Computational Imaging.

[30] U. M. Nunes and Y. Demiris, "Robust Event-based Vision Model Estimation by Dispersion Minimisation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[31] D. Liu, Parra, and T.-J. Chin, "Globally optimal contrast maximisation for event-based motion estimation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6348–6357.

[32] X. Peng, Y. Wang, L. Gao, and L. Kneip, "Globally-Optimal Event Camera Motion Estimation," in *European Conference on Computer Vision (ECCV)*. Cham: Springer International Publishing, 2020, vol. 12371, pp. 51–67.

[33] V. Vasco, A. Glover, and C. Bartolozzi, "Fast event-based Harris corner detection exploiting the advantages of event-driven cameras," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 4144–4149, 2016.

[34] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *International Journal of Robotics Research*, vol. 36, no. 2, pp. 142–149, 2017.

[35] I. Alzugaray and M. Chli, "Asynchronous Corner Detection and Tracking for Event Cameras in Real Time," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3177–3184, 2018.

[36] R. Li, D. Shi, Y. Zhang, K. Li, and R. Li, "FA-Harris: A Fast and Asynchronous Corner Detector for Event Cameras," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 6223–6229.

[37] C. Scheerlinck, N. Barnes, and R. Mahony, "Asynchronous Spatial Image Convolutions for Event Cameras," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 816–822, Apr. 2019.

[38] J. Manderscheid, A. Sironi, N. Bourdis, D. Migliore, and V. Lepetit, "Speed invariant time surface for learning to detect corner points with event-based cameras," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 10 237–10 246, 2019.

[39] Ö. Yilmaz, C. Simon-Chane, and A. Histace, "Evaluation of Event-Based Corner Detectors," *Journal of Imaging*, vol. 7, no. 2, p. 25, Feb. 2021.

[40] X. Clady, S. H. Ieng, and R. Benosman, "Asynchronous event-based corner detection and matching," *Neural Networks*, vol. 66, pp. 91–106, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2015.02.013

[41] X. Clady, J.-M. Maro, S. Barré, and R. B. Benosman, "A Motion-Based Feature for Event-Based Pattern Recognition," *Frontiers in Neuroscience*, vol. 10, p. 594, 2017.

[42] I. Alzugaray and M. Chli, "ACE: An Efficient Asynchronous Corner Tracker for Event Cameras," in *2018 International Conference on 3D Vision (3DV)*, Sept. 2018, pp. 653–661.

[43] R. Li, D. Shi, Y. Zhang, R. Li, and M. Wang, "Asynchronous event feature generation and tracking based on gradient descriptor for event cameras," *International Journal of Advanced Robotic Systems*, vol. 18, no. 4, July 2021.

[44] B. Dai, C. Le Gentil, and T. Vidal-Calleja, "A tightly-coupled event-inertial odometry using exponential decay and linear preintegrated measurements," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 9475–9482.

[45] I. Alzugaray and M. Chli, "Asynchronous Multi-Hypothesis Tracking of Features with Event Cameras," *International Conference on 3D Vision (3DV 2019)*, pp. 0–6, 2019.

[46] ——, "HASTE: Multi-Hypothesis Asynchronous Speeded-up Tracking of Events," in *31st British Machine Vision Virtual Conference (BMVC 2020)*, Oct. 2020.

[47] F. Mahlknecht, D. Gehrig, J. Nash, F. M. Rockenbauer, B. Morrell, J. Delaune, and D. Scaramuzza, "Exploring Event Camera-Based Odometry for Planetary Robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8651–8658, Oct. 2022, iEEE Robotics and Automation Letters.

[48] D. Liu, A. Parra, Y. Latif, B. Chen, T.-J. Chin, and I. Reid, "Asynchronous Optimisation for Event-based Visual Odometry," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9432–9438.

[49] C. Brandli, J. Strubel, S. Keller, D. Scaramuzza, and T. Delbruck, "ELiSeD-An event-based line segment detector," *International Conference on Event-Based Control, Communication, and Signal Processing, EBCCSP 2016*, 2016.

[50] L. Dardelet, S.-H. Ieng, and R. Benosman, "Event-Based Features Selection and Tracking from Intertwined Estimation of Velocity and Generative Contours," *arXiv:1811.07839 [cs]*, Nov. 2018.

[51] X. Peng, W. Xu, J. Yang123, and L. Kneip, "Continuous event-line constraint for closed-form velocity initialization," 2021.

[52] F. Tschopp, C. von Einem, A. Cramariuc, D. Hug, A. W. Palmer, R. Siegwart, M. Chli, and J. Nieto, “Hough2Map – Iterative Event-based Hough Transform for High-Speed Railway Mapping,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2745–2752, Apr. 2021.

[53] C. Le Gentil, F. Tschopp, I. Alzugaray, T. Vidal-Calleja, R. Siegwart, and J. Nieto, “IDOL: A Framework for IMU-DVS Odometry using Lines,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 5863–5870.

[54] W. Chamorro, J. Andrade-Cetto, and J. Solà, “High Speed Event Camera Tracking,” in *31st British Machine Vision Virtual Conference (BMVC 2020)*, Oct. 2020.

[55] W. Chamorro, J. Solà, and J. Andrade-Cetto, “Event-Based Line SLAM in Real-Time,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8146–8153, July 2022, iEEE Robotics and Automation Letters.

[56] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[57] T. D. Barfoot, C. H. Tong, and S. Särkkä, “Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression,” *Robotics: Science and Systems*, 2014.

[58] S. Anderson and T. D. Barfoot, “Full STEAM ahead: Exactly sparse Gaussian process regression for batch continuous-time trajectory estimation on SE(3),” *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, no. 3, pp. 157–164, 2015.

[59] C. Le Gentil and T. Vidal-Calleja, “Continuous Integration over SO(3) for IMU Preintegration,” *Robotics: Science and Systems XVII*, 2021.

[60] L. Wu, K. M. B. Lee, L. Liu, and T. Vidal-Calleja, “Faithful euclidean distance field from log-gaussian process implicit surfaces,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2461–2468, 2021.

[61] O. Williams and A. W. Fitzgibbon, “Gaussian process implicit surfaces,” 2006.

[62] B. Lee, C. Zhang, Z. Huang, and D. D. Lee, “Online continuous mapping using gaussian process implicit surfaces,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6884–6890.

[63] C. Le Gentil, O.-L. Ouabi, L. Wu, C. Pradalier, and T. Vidal-Calleja, “Accurate gaussian process distance fields with applications to echolocation and mapping,” 2023. [Online]. Available: https://arxiv.org/abs/2302.13005

[64] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, 2001.

[65] G. Gallego, H. Rebecq, and D. Scaramuzza, “A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3867–3876.

[66] H. Rebecq, D. Gehrig, and D. Scaramuzza, “ESIM: An Open Event Camera Simulator,” in *2nd Conference on Robot Learning (CoRL 2018)*, Zurich, Switzerland, 2018, p. 14.

[67] A. G. Wilson and H. Nickisch, “Kernel interpolation for scalable structured gaussian processes (kiss-gp),” in *Proceedings of the 32 nd International Conference on Machine Learning*, vol. 37, 2015.

[68] R. Giubilato, C. Le Gentil, M. Vayugundla, T. Vidal-Calleja, and R. Triebel, “Gpgm-slam: Towards a robust slam system for unstructured planetary environments with gaussian process gradient maps,” in *IROS Workshop on Planetary Exploration Robots: Challenges and Opportunities (PLANROBO20)*, 2020.