Continuous-Time Stereo-Inertial Odometry

David Hug, *Graduate Student Member, IEEE*, Philipp Bänninger, *Member, IEEE*, Ignacio Alzugaray, *Member, IEEE*, and Margarita Chli, *Member, IEEE*

***Abstract*—The emerging paradigm of Continuous-Time Simultaneous Localization And Mapping (CTSLAM) has become a competitive alternative to conventional discrete-time approaches in recent times and holds the additional promise of fusing multi-modal sensor setups in a truly generic manner, rendering its importance to robotic navigation and manipulation seminal. In this spirit, this work expands upon continuous-time concepts, evaluates their suitability in common stereo and stereo-inertial online configurations and provides an extensible, generic, robust and modular open-source implementation to the community. The presented experimental analysis records the performance of our approach in these setups against the state-of-the-art in discrete-time Simultaneous Localization And Mapping (SLAM) on established datasets, achieving competitive results, and provides a direct comparison between online discrete- and continuous-time approaches for the first time. Targeting the absence of open-sourced, continuous-time pipelines and their associated, oftentimes prohibitive, initial developmental overhead, our implementation is made public.**

***Index Terms*—Visual-Inertial SLAM, Sensor Fusion.**

I. Introduction

THE joint task of estimating a system's ego-motion and simultaneously mapping its surroundings, known as SLAM, is achieved in practice by combining multiple, complementary sensing modalities. Over the last decade, a number of cost-effective and portable sensor setups have been consolidated in machine perception, including visual-monocular [1]–[4], stereo-visual [5], Inertial Measurement Units (IMUs) [6]–[10], Global Navigation Satellite System (GNSS) / Real-Time Kinematic Positioning (RTK) [11], [12] and Light Detection And Ranging (LIDAR) [13] sensors, among others. The SLAM optimization problem has traditionally been posed over a discrete set consisting of the most-informative states for motion estimation and mapping (*e.g.* keyframes in visual SLAM), effectively keeping the growth of the problem tractable under realistic hardware constraints. Common difficulties amongst these techniques lie with the absence of inter-state motion information, necessitating discretized kinematic constraints, the introduction of quantization errors and the cumbersome integration/synchronization of (asynchronous) multi-rate sensing modalities. For instance, IMU measurements are commonly pre-integrated into relative constraints, whereas scan lines of rolling-shutter cameras and LIDARs sensors prove to be problematic due time offsets between individual measurements. Alternative approaches based on assigning each measurement its individual state, however, swiftly become computationally prohibitive, especially for high-rate sensing modalities. CTSLAM formulations target these shortcomings and allow asynchronous measurements to be registered at their exact acquisition time, making them appealing in the context of multi-modal sensor calibration [14]–[16], rolling shutter camera [16]–[18] and event-based vision [19], [20] applications. Continuous-time parametrizations not only enable more straightforward sensor fusion schemes, but also allow for a reduction in overall state size in scenarios with low dynamicity, occurring aboard trains or cars [21], [22] for instance. While CTSLAM systems are typically more computationally demanding than their discretized counterparts, recent works [23], [24] show promising results towards a more efficient use of continuous-time parametrizations.

In light of these advances and advantages, this work presents a novel stereo-inertial CTSLAM pipeline that extends upon *Hyper*SLAM [23], which presented a monocular proof-of-concept in purely simulated batch optimization scenarios. Based on several continuous-time adaptions of classical methods, the proposed system is capable of reliably estimating the ego-motion in real world experiments while being generic, modular and readily extensible to other sensing modalities. The presented approach is benchmarked against state-of-the-art discrete-time SLAM systems [5], [6], [9], [25] on well-established, public datasets [26], [27] achieving competitive results. To the authors' best knowledge, this is the first work providing a direct cross-comparison between online continuous- and discrete-time frameworks on multiple benchmarking datasets and, thus, breaks ground for more transparent comparisons across continuous-time approaches. Concisely, this work

a) investigates the suitability of continuous-time methods in common, real world setups,
b) extends these approaches to the realm of online (*i.e.* non-batch) operations and stereo-inertial cases and
c) provides an extensive and modular open-source implementation of the presented algorithms.

II. Related Work

The theoretical groundwork for the application of continuous-time SLAM was done by Furgale *et al.* [14], who formulated it as

Manuscript received February 24, 2022; accepted April 20, 2022. Date of publication May 10, 2022; date of current version May 17, 2022. This letter was recommended for publication by Associate Editor F. Caballero and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported in part by the SNSF under Grant PP00P2 157585, and in part by the NCCR Robotics and the ETH Mobility Initiative. *(Corresponding author: David Hug.)*

The authors are with the Vision for Robotics Lab, Department of Mechanical and Process Engineering, ETH Zürich, CH-8092 Zurich, Switzerland (e-mail: dhug@ethz.ch; baephili@ethz.ch; aignacio@ethz.ch; chlim@ethz.ch).

Code—https://github.com/VIS4ROB-lab/HyperSLAM
Video—https://youtu.be/Q6PH9KKsMyQ
Digital Object Identifier 10.1109/LRA.2022.3173705

a Maximum *a posteriori* Likelihood Estimation (MLE) problem with the aim of significantly compressing the state size of the underlying optimization problem associated with multi-sensor configurations. Given the benefit of such a formulation for the monocular, visual-inertial case, it was later extended to include temporal offset calibration between a camera and an IMU in [15]. Around the same time, Lovegrove *et al.* [17] explored the CTSLAM problem with rolling-shutter cameras in a simulated monocular visual-inertial setup, where the characteristics of the rolling-shutter effect were explicitly modelled by exploiting the continuous-time parametrization. Analogously, in [19], [20], Mueggler *et al.* applied a continuous-time formulation to handle the asynchronous nature of sensing cues produced by event cameras in a monocular visual(-inertial) setup. The problem of addressing loop closure in large scale environments in continuous-time was first investigated by Anderson *et al.* [28]. Work conducted by Droeschel and Behnke [13] provided insights into the benefits of utilizing continuous-time state estimation in LIDAR applications aboard vehicles, and was able to improve upon state-of-the-art mapping methods at the time. Park *et al.* [29], [30] also explored means to enable life-long applications as well as target- and structureless calibration methods in the context of continuous-time LIDAR approaches. Anderson *et al.* also analyzed the impact of moving to a hierarchical wavelet decomposition to represent the continuous-time motion in their adjacent work [31], which allowed for adaptive refinement of the underlying representation. A further example of the advantages of continuous-time representations when treating inertial data is presented by Rehder *et al.* [32], who applied it to the intrinsic calibration of multiple IMUs. Zhang and Scaramuzza [33] utilized Gaussian processes to model continuous-time representations in a probabilistic context to tackle temporal associations in a principled manner. The prospects of applying generalized, non-uniform B-Splines of arbitrary order to simulated visual monocular problems was further researched in our previous work [23]. Moreover, several authors, including Haarbach *et al.* [34], Ovrén and Forssén [35] and Sommer *et al.* [24], all advocated in favour of more efficient parametrization and interpolation methods. A recent, comparative study by Cioffi *et al.* [36] further substantiated the advantages of continuous-time approaches over conventional approaches in batch operation on common datasets. Consolidating insights from [23], [24] and building on the bulk, state-of-the-art continuous-time research, we further explore stereo and stereo-inertial configurations, demonstrating the competitiveness of continuous-time approaches against established discrete-time frameworks.

III. Methodology

A. Parametrization-Agnostic Non-Linear Optimization

Let Θ be the collection of motion-parametrizing states, Φ the collection of sensor-specific parameters (*e.g.* intrinsics, distortion parameters, *etc.*) and Ψ the exteroceptive parameters (*e.g.* visual landmarks, gravity, *etc.*). Furthermore, let $m(t_m) \in \mathcal{M}_s$ represent a measurement acquired by the sensor $s \in \mathcal{S}$ at time t_m, where $\mathcal{S}$ denotes a collection of sensors. Individual measurements m are compared against their associated predictions $\hat{m}(t_m, \Theta, \Phi_s, \Psi)$, through the definition of a weighted residual $\|\boldsymbol{r}_s\|_{\boldsymbol{\Sigma}_s} = \boldsymbol{r}_s^\top \boldsymbol{\Sigma}_s \boldsymbol{r}_s$. Dropping non-essential dependencies and potential regularization terms, the generic, parametrization-agnostic, multi-modal minimization problem, *i.e.* valid for discrete- and continuous-time approaches and an arbitrary combination of sensors, is defined as

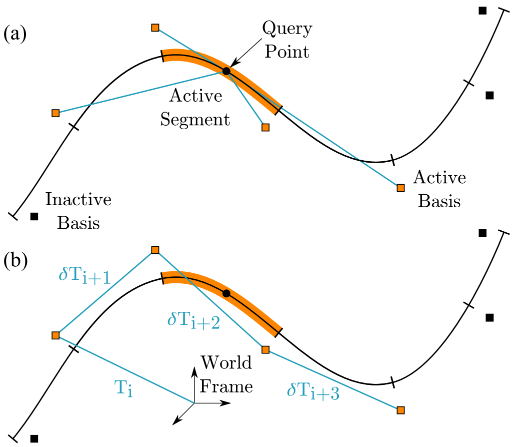

Fig. 1. a) Associated segment and bases (in orange) corresponding to an interpolated estimate for a given query point. b) Contributions of increments between adjacent bases (in blue) to the interpolated query point.

$$\Theta^*, \Phi^*, \Psi^* = \underset{\Theta,\Phi,\Psi}{\operatorname{argmin}} \sum_{s \in \mathcal{S}} \sum_{m \in \mathcal{M}_s} \|\boldsymbol{r}_s\left(\hat{m}, m\right)\|_{\boldsymbol{\Sigma}_s}. \tag{1}$$

B. Continuous-Time Parametrization

Continuous-time representations [37], [38] have long been established and utilized across a variety of applications where continuous approximation, interpolation or regression is essential. In recent times, some of these representations, such as B-Splines [23], [35], [38], have also gained traction within the field of Computer Vision due to several advantageous characteristics such as compact representation, local support and inherent smoothness. Consequently, they are well suited to model time-dependent transformations $\boldsymbol{T}_{wb}(t)$ between the world frame w and the body frame b as illustrated in Fig. 1. Split representations, which parametrize transformations $\boldsymbol{T}_{wb}(t)$ as pairs of unit quaternions $\boldsymbol{q}_{wb}(t) \in \mathbb{SU}(2)$ and translations ${}^{w}\boldsymbol{x}_{wb}(t) \in \mathbb{R}^3$, in particular, were shown to enhance the overall accuracy and convergence rate in Non-Linear Least Squares (NLLS) optimizations in [34], [35], whilst accommodating different sampling rates in rotations and translations, rendering them especially suitable. Hence, we adopt the split parametrization from [23] which is based on the formalism of non-uniform, $\mathcal{C}^2$-continuous, cumulative, cubic B-Splines and interpolates transformations $\boldsymbol{T}_{wb}(t)$ as follows,

$$\boldsymbol{T}_{wb}(t) = \begin{bmatrix} \boldsymbol{R}\left(\boldsymbol{q}_{wb}(t)\right) & {}^{w}\boldsymbol{x}_{wb}(t) \\ 0 & 1 \end{bmatrix} \in \mathbb{SE}(3) \quad \text{with} \tag{2}$$

$$\boldsymbol{q}_{wb}(t) = \boldsymbol{q}_{wi} * \prod_{j=1}^{k} \left(\boldsymbol{q}_{w(i+j-1)}^{-1} * \boldsymbol{q}_{w(i+j)}\right)^{\lambda_j(t)} \tag{3}$$

$${}^{w}\boldsymbol{x}_{wb}(t) = {}^{w}\boldsymbol{x}_{wi} + \sum_{j=1}^{k} \left({}^{w}\boldsymbol{x}_{w(i+j)} - {}^{w}\boldsymbol{x}_{w(i+j-1)}\right) \lambda_j(t), \tag{4}$$

where $\boldsymbol{R}(\boldsymbol{q})$ converts a quaternion to a rotation matrix and $\boldsymbol{q}^{\lambda}$ denotes the quaternion power. Interpolated transformations $\boldsymbol{T}_{wb}(t)$ are computed by blending a collection of $k+1$ bases $\{\mathcal{B}_i, \ldots, \mathcal{B}_{i+k}\}$ (see Fig. 1), where the i-th basis with associated timestamp t_i is defined as $\mathcal{B}_i := \{t_i, \boldsymbol{q}_{wi}, {}^{w}\boldsymbol{x}_{wi}\}$. Here, k represents the Degree of Freedom (DoF) of the spline of choice, which is set to $k = 3$ in this manuscript to arrive at a cubic representation. In (3) and (4), the bases are blended according to $\lambda_j(t)$, which is the j-th entry of the k-dimensional interpolation vector $\boldsymbol{\Lambda}_i(t)$ from (5). $\boldsymbol{\Lambda}_i(t)$ itself is defined as the matrix multiplication between the generalized mixing matrix $\boldsymbol{\Pi}_i$ from [23], [37] and the stacked vector of normalized times $\boldsymbol{U}_i(t)$. In particular, $\boldsymbol{U}_i(t)$ is defined as $[1,\, u_i(t),\, u_i^2(t),\, \ldots,\, u_i^k(t)]^\top$ where individual entries are powers of the normalized time $u_i(t) = (t - t_{i-1})/(t_i - t_{i-1})$ such that $t_{i-1} \leq t < t_i$ holds for valid query times t. The expression for $\boldsymbol{\Lambda}_i(t)$ simplifies to

$$\boldsymbol{\Lambda}_i(t) = \boldsymbol{\Pi}_i \boldsymbol{U}_i(t) = \frac{1}{3!} \begin{bmatrix} 6 & 0 & 0 & 0 \\ 5 & 3 & -3 & 1 \\ 1 & 3 & 3 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{U}_i(t) \tag{5}$$

for uniform, cumulative, cubic B-Splines. As presented in [23], [24], the above formulae also give rise to closed-form solutions for instantaneous angular and linear velocities/accelerations, which are omitted here.

C. Visual Sensor Model

Extending upon notions from Section III-A, the utilized visual sensor model comprises the relative transformation $\boldsymbol{T}_{bs}$ (with body frame b and sensor frame s), the intrinsic matrix as well as parameters related to the chosen lens distortion model in its set of optimizable quantities Φ_s. Abstracting from the specific distortion model, one defines a generic function $\Gamma : \mathbb{R}^3 \rightarrow \mathbb{S}^2$, mapping distorted, homogeneous, normalized pixel coordinates of landmark detections in the image plane to undistorted on-unit-sphere vectors ${}^{s}\boldsymbol{b} \in \mathbb{S}^2$, so-called bearings, as introduced in [23]. Conversely, predicted homogeneous landmarks ${}^{s}\hat{\boldsymbol{L}}$ at time t_m are obtained by transformation of the corresponding ${}^{w}\hat{\boldsymbol{L}} \in \Psi$ to the sensor frame using $\hat{\boldsymbol{T}}_{wb}(t_m)$ according to

$${}^{s}\hat{\boldsymbol{L}}(t_m) = \boldsymbol{T}_{sb} \hat{\boldsymbol{T}}_{bw}(t_m) {}^{w}\hat{\boldsymbol{L}} \in \mathbb{R}^4. \tag{6}$$

Visual predictions are compared against their corresponding measurements ${}^{s}\boldsymbol{b}$ *via*

$$\boldsymbol{r}_s = \angle\left(\hat{m}, m\right) = \angle\left({}^{s}\hat{\boldsymbol{L}}, {}^{s}\boldsymbol{b}\right) \in \mathbb{R}, \tag{7}$$

where $\angle(\cdot,\cdot)$ measures the angle between vectors. In this work, we forgo a detailed treatment of the distortion covariance propagation trough Γ and, instead, select a constant weight $\boldsymbol{\Sigma}_s$, as defined in (1), for all visual residuals $\boldsymbol{r}_s$. Note that (7) remains valid for unsynchronized and rolling shutter setups, and merely requires appropriate modification of the individual measurement times t_m to model these effects.

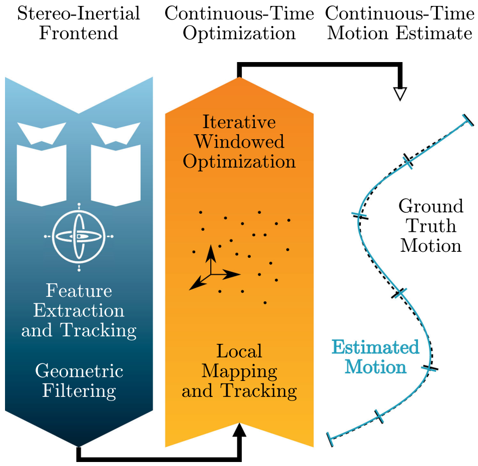
Stereo-Inertial Frontend
Continuous-Time Optimization
Continuous-Time Motion Estimate
Feature Extraction and Tracking
Geometric Filtering
Iterative Windowed Optimization
Local Mapping and Tracking
Ground Truth Motion
Estimated Motion

Fig. 2. Schematic system overview outlining (from left to right) the most relevant steps and components of the implemented pipeline.

D. Inertial Sensor Model

We make use of the inertial sensor model introduced by Rehder *et al.* [32], which comprises the sensor-specific transformation $\boldsymbol{T}_{bs}$, the estimated gyroscope and accelerometer biases $\hat{\boldsymbol{B}}_\omega(t)$ and $\hat{\boldsymbol{B}}_\alpha(t) \in \mathbb{R}^3$, the non-orthonormality axis-alignment matrices $\boldsymbol{S}_\omega$ and $\boldsymbol{S}_\alpha \in \mathbb{R}^{3\times 3}$ alongside a gravity estimate ${}^{w}\hat{\boldsymbol{g}} \in \Psi$. Its corresponding residual is defined as

$$\boldsymbol{r}_s = \hat{m} - m = \begin{bmatrix} {}^{s}\hat{\boldsymbol{\omega}}_{ws} \\ {}^{s}\hat{\boldsymbol{\alpha}}_{ws} \end{bmatrix} - \begin{bmatrix} {}^{s}\boldsymbol{\omega}_{ws} \\ {}^{s}\boldsymbol{\alpha}_{ws} \end{bmatrix}, \tag{8}$$

where predictions $\hat{m}$, comprising angular velocities ${}^{s}\hat{\boldsymbol{\omega}}_{ws}$ and linear accelerations ${}^{s}\hat{\boldsymbol{\alpha}}_{ws}$, are derived from the associated, predicted temporal derivatives of $\hat{\boldsymbol{T}}_{wb}(t_m)$. In particular,

$$\boldsymbol{r}_s = \begin{bmatrix} \boldsymbol{S}_\omega \boldsymbol{R}_{sb} {}^{b}\hat{\boldsymbol{\omega}}_{wb} + \hat{\boldsymbol{B}}_\omega \\ \boldsymbol{S}_\alpha \boldsymbol{R}_{sb} \hat{\boldsymbol{R}}_{bw} \left({}^{w}\hat{\boldsymbol{\alpha}}_{wb} - {}^{w}\hat{\boldsymbol{g}}\right) + \hat{\boldsymbol{B}}_\alpha \end{bmatrix} - \begin{bmatrix} {}^{s}\boldsymbol{\omega}_{ws} \\ {}^{s}\boldsymbol{\alpha}_{ws} \end{bmatrix}, \tag{9}$$

where additional terms for off-centered inertial sensors as well as explicit time-dependencies are dropped for conciseness. In this work, $\boldsymbol{S}_\omega$ and $\boldsymbol{S}_\alpha$ are set to identity and kept constant and the biases $\hat{\boldsymbol{B}}_\omega$ and $\hat{\boldsymbol{B}}_\alpha$ are modeled as splines.

E. System Description

1) System Overview: The proposed pipeline, depicted in Fig. 2, follows established paradigms and conceptually distinguishes between the initial acquisition and pre-processing of raw inputs in the frontend and the subsequent integration of the extracted information into an NLLS optimization problem in the backend. As illustrated in Fig. 3, the backend also manages contractions and expansions of the sliding optimization window to allow online operations, adds, or removes, residuals to/from the active set of optimizable parameters (see (1)) and integrates additional constraints to assert the stability of the minimization.

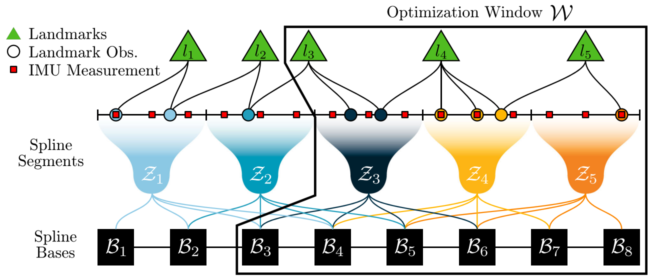

Fig. 3. Factor-graph-like representation of the underlying, continuous-time optimization problem of the proposed pipeline. The continuous-time representation is constructed from individual segments $\mathcal{Z}_i$, which depend on a collection of (shared) optimizable bases $\mathcal{B}_i$. (Asynchronous) measurements (*e.g.* observations of landmarks $l_i \in \mathcal{L}$ or IMU measurements) depend on the bases associated with the segment at which the sensor data is received. Individual bases $\mathcal{B}_i$ reside in the optimization problem as long as their corresponding segment is partially or entirely contained in the optimization window $\mathcal{W}$.

Algorithm 1: Sliding Optimization Window Update.

```
Input
  Measurement $m(t_m)$ from sensor $s$
if $\mathcal{W} = \emptyset$ then
  Initialize motion parametrization $\Theta$ s.t. $t_m \in \mathcal{T}$
else If $t_m \notin \mathcal{W}$ then
  Optimize for optimal parameters $\Theta^*$, $\Phi^*$ and $\Psi^*$
  while $t_m \notin \mathcal{W}$ do
    Extrapolate new basis $\mathcal{B}_{\text{new}}$ and add it to $\Theta$
  end while
  while $\|\mathcal{W}\| > t_{\max}$ do
    Remove residuals $\boldsymbol{r}_s$ associated with oldest $\mathcal{B}_{\text{oldest}}$
    Remove oldest $\mathcal{B}_{\text{oldest}}$ from $\Theta$
  end while
end if
Create residual $\boldsymbol{r}_s$ and add it to the optimization
```

The optimization itself relies on the Ceres Solver [39] to solve for the optimal states Θ^*, Φ^* and Ψ^* upon expansions of the sliding window.

2) Visual Frontend: The presented system extracts and tracks visual cues in the stereo-visual image stream based on the established Kanade–Lucas–Tomasi (KLT) [40], [41] approach. In particular, salient visual cues in the image stream are identified using the Shi-Tomasi feature detector [42], followed by initial tracking of visual features between corresponding stereo image pairs (*i.e.* geometric tracking), and subsequent tracking across consecutive frames (*i.e.* temporal tracking) in the image stream to generate a consistent and information-rich set of feature tracks similar to the method from Qin *et al.* [5], [8]. In order to increase the reliability of detected tracks, an additional, tighter constraint, requiring candidate cues to be successfully tracked across two independent paths, is introduced. In particular, it is enforced that (left) temporal tracking followed by (left-right) geometric tracking yields the same result as (left-right) geometric tracking followed by (right) temporal tracking across complete stereo frames. Aiming to further reduce the number of low-quality feature tracks and boost the overall reliability of the frontend, cross-checked features are further required to fulfil the epipolar constraint from (10), where $\boldsymbol{r}_E$ is the violation of the epipolar constraint, $\boldsymbol{E}(\cdot)$ denotes the essential matrix, ${}^f\boldsymbol{b}$ expresses a bearing vector ($\in \mathbb{S}^2$) in frame f, and the angle $\theta_{\max}$ corresponds to the angular equivalent of a given pixel error on the image plane. Specifically, $\theta_{\max} = \arctan(x_{\max}/f)$ with pixel error $x_{\max}$ and focal length f. The remaining, purified visual tracks are subsequently transmitted to the backend in an independent and unsynchronized manner and await their conversion into visual residuals $\boldsymbol{r}_s(\hat{m}, m)$ (see Section III-C) and ultimate integration into the optimization problem.

$$\boldsymbol{r}_E := {}^a\boldsymbol{b}^\top \frac{\boldsymbol{E}\left(\boldsymbol{T}_{ab}\right){}^b\boldsymbol{b}}{\left\|\boldsymbol{E}\left(\boldsymbol{T}_{ab}\right){}^b\boldsymbol{b}\right\|}, \quad \left\|\boldsymbol{r}_E\right\| \overset{!}{\leq} \sin\left(\theta_{\max}\right) \tag{10}$$

3) Inertial Frontend: Based on the ability of continuous-time methods to directly fuse inertial measurements into the optimization problem, we circumvent conventional pre-integration of inertial measurements and are able to limit the pre-processing to the conversion into compatible, internal formats before asynchronous submission to the backend.

4) Sliding Window Optimization: As introduced in (1), discrete- and continuous-time SLAM approaches both, in principle, pose the associated optimization problem over a finite collection of motion-parametrizing, optimizable states Θ, and manage to effectively bound the problem size by only considering a limited number of most recent and relevant optimizable states in Θ. This concept is depicted as the optimization window $\mathcal{W}$ in the factor graph Fig. 3, where one also observes that the set of active landmarks (*i.e.* landmarks with at least one associated measurement time contained in the window $\mathcal{W}$) depends on the evolution of $\mathcal{W}$ as well. In contrast to conventional formulations, however, continuous-time representations interpret Θ as a collection of interpolating bases $\mathcal{B}_i$ according to Eq. (2), rather than directly treating them as a finite set of optimizable transformations in $\mathbb{SE}(3)$, abstracting subsets of bases to segments as illustrated in Fig. 3. Said abstraction of the states Θ into segments represents one of the essential ideas to continuous-time methods and empowers them to process incoming measurements in an entirely time-based and asynchronous manner, running contrary to notions of conventional discretization requirements, and allows native support of arbitrary measurement times t_m. In fact, not only does it allow to query transformations $\hat{\boldsymbol{T}}_{wb}(t_m)$ at arbitrary times contained inside the valid time range $\mathcal{T} = [t_{\text{start}}, t_{\text{end}})$, induced by the span of individual segments $\mathcal{Z}_i$ between $\mathcal{Z}_{\text{start}}$ and $\mathcal{Z}_{\text{end}}$ (*i.e.* first and last segment), but also drives the prediction of instantaneous velocities and accelerations.

Based on these notations, the update of the optimization window, which runs upon processing a new measurement in the backend, is presented in Algorithm 1. In Algorithm 1, the extrapolation step for the motion-parametrizing states Θ aims to predict a plausible path of motion for the system, and, thus, assumes a constant velocity model to initialize new bases $\mathcal{B}$ into the future. Furthermore, in the case of visual observations, we make use of the algorithm introduced by Lee and Civera [43]

TABLE I
THE RMSES OBTAINED ON THE BENCHMARK DATASETS FOR *HYPER*SLAM AND STATE-OF-THE-ART SYSTEMS. ALL VALUES ARE PROVIDED IN METERS

KITTI (Stereo)	0	1	2	3	4	5	6	7	8	9	10	Avg.
ORB-SLAM3 [9]	4.244	12.242	6.756	1.275	0.242	2.069	1.982	1.287	3.895	3.014	1.261	3.479
VINS-Fusion [5]	13.093	7.947	21.019	1.642	1.295	6.386	3.536	2.107	10.027	7.801	3.769	7.148
*Hyper*SLAM (Ours)	7.696	×[a]	14.421	0.935	0.661	3.355	3.111	2.855	10.123	3.927	4.318	5.140
EuRoC (Stereo)	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	Avg.
ORB-SLAM3 [9]	0.035	0.018	0.026	0.083	0.128	0.036	0.409	0.279	0.043	0.085	×	0.1142
SVO[b] [25]	0.080	0.070	0.270	2.420	0.540	0.040	0.080	0.360	0.070	0.140	×	0.4070
VINS-Fusion [5]	0.571	0.525	0.494	0.844	0.664	0.549	0.232	×	0.230	0.209	×	0.4798
*Hyper*SLAM (Ours)	0.071	0.072	0.207	0.269	0.223	0.210	0.099	×	0.118	0.116	×	0.1539
*Hyper*SLAM (Ours, async.)	0.078	0.095	0.299	0.412	0.228	0.253	0.134	×	0.129	0.139	×	0.1963
EuRoC (Stereo-Inertial)	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	Avg.
ORB-SLAM3 [9]	0.034	0.040	0.043	0.056	0.089	0.059	0.030	0.043	0.034	0.031	0.067	0.0478
OKVIS [6]	0.212	0.101	0.141	0.185	0.255	0.046	0.058	0.111	0.066	0.076	×	0.1251
VINS-Fusion [5]	0.252	0.217	0.326	0.413	0.308	0.113	0.108	0.108	0.130	0.120	0.318	0.2193
*Hyper*SLAM (Ours)	0.115	0.071	0.183	0.192	0.149	0.079	0.159	0.178	0.051	0.111	×	0.1288
*Hyper*SLAM (Ours, async.)	0.079	0.057	0.166	0.200	0.235	0.093	0.133	0.314	0.074	0.120	×	0.1471

[a] Entries with a × are not considered in the average.

[b] Values are provided as reported in [25] without motion priors (i.e. edgelet-only configuration) for fairness of comparisons.

to obtain initial landmark triangulations from full stereo frames, and additionally reduce the influence of outliers by applying a Huber loss to the visual residuals $\boldsymbol{r}_s$ (see Section III-C). In contrast to other approaches [9], the presented pipeline does not support loop closure detection at present, which remains to be adapted to continuous-time approaches, serving as aspiration for future extensions.

IV. EXPERIMENTS

We evaluate the proposed pipeline on two real-world benchmark datasets, namely the KITTI [26] and the EuRoC [27] datasets. Across all sequences in the datasets, the time window $\mathcal{W}$ is adapted to contain the same, fixed number of frames in order to arrive at similar amounts of information within the optimization window for each instance in time. Specifically, we select $\mathcal{W}$ to encompass the last 60 frames, which, in turn, corresponds to a temporal window size of 3 s for EuRoC and 6 s for KITTI. Empirically, we choose the temporal spacing between bases $\mathcal{B}_i$ to be twice the camera frequency, which suffices to parametrize the dynamicity of all considered sequences adequately (see Section IV-C), that is, $t_i = 0.1$ s for EuRoC and $t_i = 0.2$ s for KITTI.

Owing to the absence of other directly comparable, publicly available online CTSLAM systems, we compare the proposed continuous-time approach, against other state-of-the-art, discrete-time approaches; namely, VINS-Fusion [5], SVO [25], OKVIS [6], and ORB-SLAM3 [9]. While these methods follow different paradigms with respect to the motion parametrization, this comparison aims to contextualize the performance achieved by the proposed framework against other popular discrete-time alternatives to SLAM from the literature. Since the proposed pipeline does not currently support loop-closing (see Section III-E4), loop-closure-capabilities offered by VINS-Fusion and ORB-SLAM3 are deactivated for the sake of fairness. The evaluated Root Mean Square Errors (RMSEs) and relative translation and rotation errors for the compared methods are presented against the provided ground-truth information using open-source evaluation tools [44], [45]. All experiments were run using a AMD Ryzen 7 4700U@4 GHz CPU with one dedicated thread each for the front- and the backend.

A. Evaluation on KITTI

The KITTI dataset focuses on large-scale outdoor environments captured atop a moving vehicle and employs a sensor-suite comprising multiple stereo rigs, a LIDAR sensor as well as an Inertial Navigation System (INS). It consists of eleven sequences, which vary in complexity with respect to their underlying motion as well as with respect to the occurrence of dynamic objects in the scenes. Here, we utilize the pre-rectified stereo image streams, captured at 10 Hz and at a resolution of 1241×376 pixels, to evaluate our approach in the stereo case against the provided ground-truth motion estimate, which was obtained from a combined global positioning and INS.

From Table I one observes that, considering the overall scale of the environment, *Hyper*SLAM achieves similar performance with respect to ORB-SLAM3 on most sequences and almost consistently better estimates than VINS-Fusion, which is also reflected in the relative error plots Fig. 5. Failure cases are limited to a single sequence, namely KITTI01, where dynamic obstacles are present, which *Hyper*SLAM cannot currently handle. Further analysis also reveals that the small gap in performance with respect to ORB-SLAM3 is caused by substantially slower movements along the vertical axis (relative to the horizontal, in-plane motions), which our method cannot properly estimate due to the relatively short optimization window compared to the typical duration of the sequences. Despite this, as Fig. 4 reveals, the observed drift for the in-plane motion remains unaffected.

B. Evaluation on EuRoC

The EuRoC dataset comprises a collection of 6-DoF flight paths, for which stereo-inertial data was captured aboard a Micro Aerial Vehicle (MAV) in indoor environments. Every sequence provides gray-scale stereo image streams, captured with synchronized, global-shutter cameras operating at 20 Hz with a resolution of 752×420 pixels, alongside time-synchronized inertial measurements, ground-truth camera poses, and corresponding

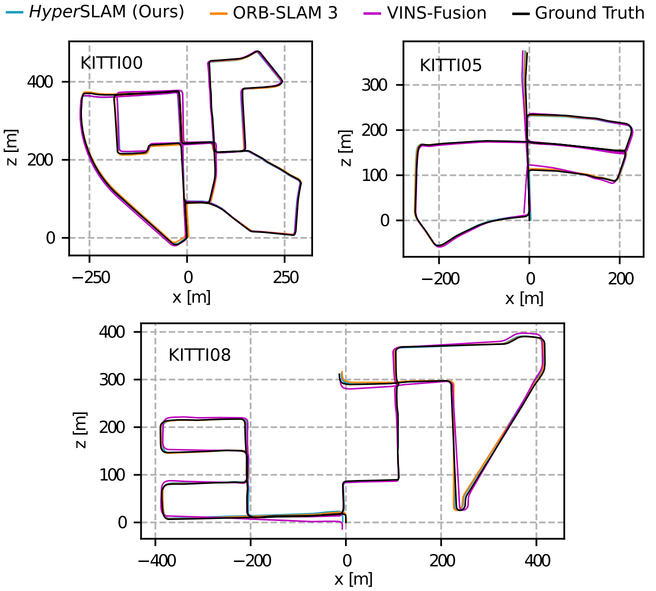

Fig. 4. Metric motion estimates after $\mathbb{SE}(3)$ alignment for selected KITTI sequences for the stereo configuration.

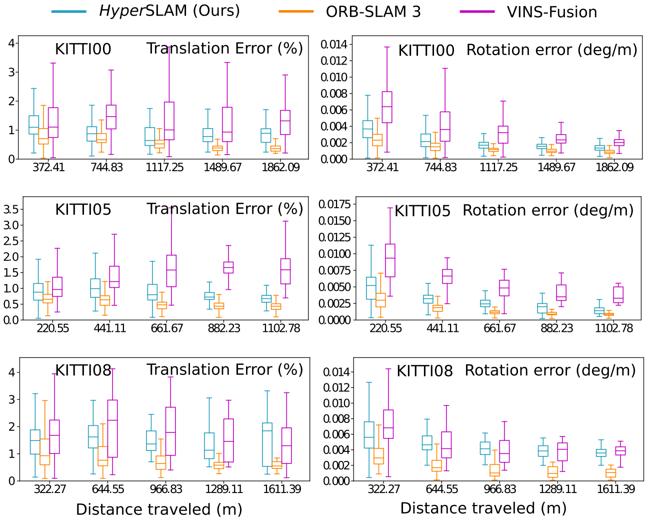

Fig. 5. Relative translation and rotation errors as a function of travelled distances on selected KITTI sequences for the stereo configuration.

calibration files. The recorded flight paths consist of sequences from both the large-scale Machine Hall (MH) as well as the small scale Vicon Room (VR) indoor environments, which vary in the complexity of the undergone motion. We make use of the provided visual and inertial information in these sequences and fix the ex-/intrinsic camera parameters to the ones provided in the calibration.

Table I reveals that our continuous-time method performs considerably better from a *global* perspective than VINS-Fusion [5], on par with SVO [25] and OKVIS [6], and worse than ORB-SLAM3 [9] in terms of RMSE. A common failure case across all approaches is the sequence V203 in the stereo setup, where the apparent motion is considerable, leading to substantial motion blur and, ultimately, loss of reliable visual information. A similar reason underlies the other failure case of *Hyper*SLAM and VINS-Fusion on V103, where the undergone motion is fast and similar KLT feature tracking strategies lead to impaired quality in the tracked, visual cues for both approaches. In the stereo-inertial case, one observes a similar ordering of the considered methods in relative RMSE performance with lower absolute values among the compared frameworks with one failure case each for OKVIS and *Hyper*SLAM. Both failures are restricted to one of the most challenging sequences, V203, where both motion estimates yield detached *local* solutions, but inconsistent *global* ones. Considering that open-loop approaches can never recover from single point failures like these, the presented RMSEs only provide insights on the general, *global* performance of the different methods. In order to also investigate the *local* estimation accuracy, we further illustrate the obtained $\mathbb{SE}(3)$-aligned trajectories in Fig. 6 and provide relative error plots in Fig. 7. These results reveal that *Hyper*SLAM compares well to all other approaches in terms of relative errors in translation and rotation and even outperforms most of them in terms of *local* drift for short distances, including ORB-SLAM3. Based on these results, we can infer that continuous-time approaches have the potential to achieve better *local* consistency than discrete-time methods.

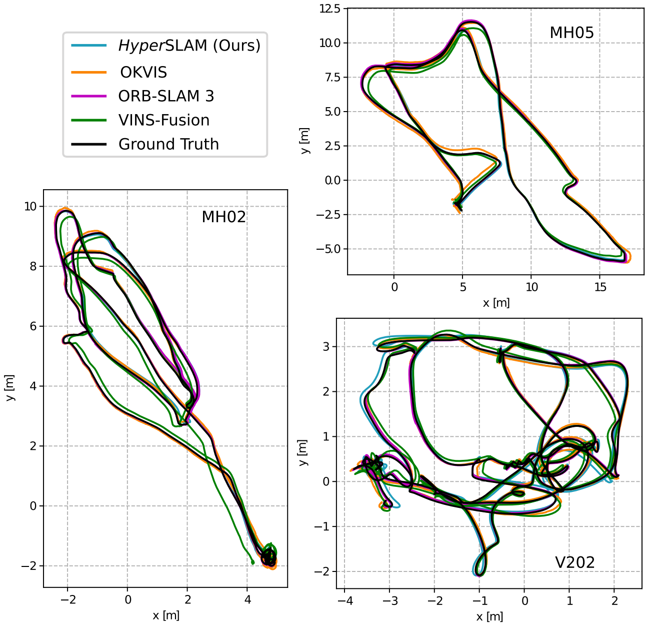

Fig. 6. Metric motion estimates after $\mathbb{SE}(3)$ alignment for selected EuRoC sequences for the stereo-inertial configuration.

C. Analysis on Continuous- Vs. Discrete-Time Formulations

As the temporal spacing between bases is one of the vital parameters influencing the accuracy of the proposed method, we present a quantitative analysis on the relation between the achieved RMSEs and the selected spacing, ranging from 60 to 600 ms, for the stereo case in Fig. 8. Three regions of interest

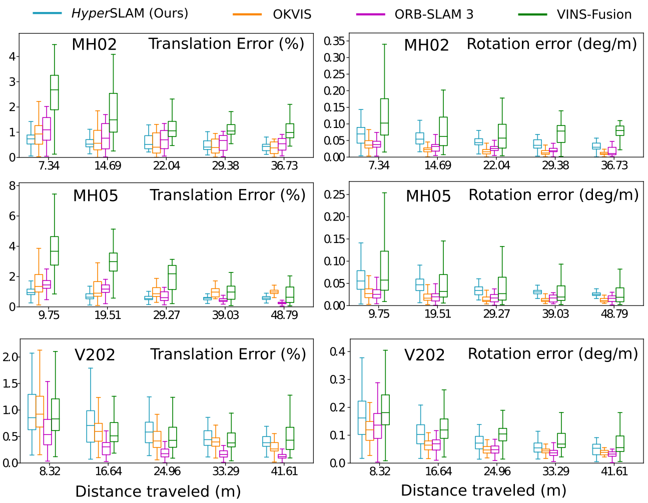

Fig. 7. Relative translation and rotation errors as a function of travelled distances on selected EuRoC sequences for the stereo-inertial configuration.

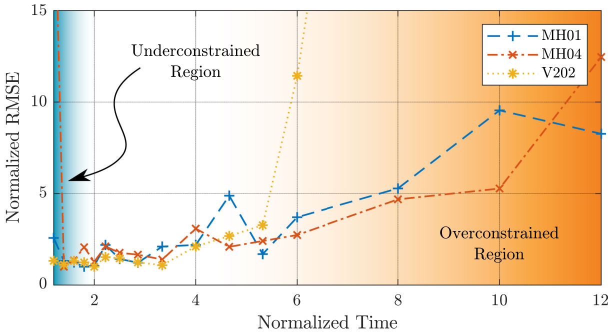

Fig. 8. RMSEs achieved by *Hyper*SLAM for MH01, MH02 and V202, in the stereo configuration, normalized by the respective minimally achieved RMSE as a function of the temporal spacing between bases, normalized by the image acquisition frequency.

emerge: (a) an underconstrained region due lack of measurements, (b) a wide basin of well-behaved convergence, and (c) an overconstrained region, where the undergone motion can no longer be parametrized appropriately. The effect observed in case (c) becomes more predominant under motion with higher volatility, causing V202 to expose a smaller convergence basin than MH01 and MH04 and showcases the inter-dependence between the temporal spacing and the dynamicity of the motion. Fluctuations in RMSEs for case (b) can mostly be attributed to isolated, *local* under-parametrization of sections exhibiting high dynamics and/or abrupt change in orientation.

We found that choosing the spacing at twice the image acquisition frequency mitigates this under-parametrization and provides competitive *local* accuracy at a reasonable level of computational effort, entailing that our implementation achieves near real-time operation on KITTI, while still running sub-real-time on EuRoC. Taking into consideration that most discrete-time systems employ keyframing strategies, which effectively enlarge the scope of the optimization window artificially, while keeping the computational effort constant, the gap in *global* accuracy with respect to ORB-SLAM3 is put into perspective.

While the stereo-inertial configuration already serves an example of asynchronous processing of measurements, we further highlight the applicability of continuous-time methods to non-synchronized stereo and stereo-inertial inputs under realistic assumptions. To this end, we run our pipeline on a modified image stream, where every third frame is a complete pair of stereo images and intermediate frames are monocular only, constituting a setup which discrete-time approaches, like ORB-SLAM3, are unable to handle. The obtained RMSEs in Table I, referred to as "*Hyper*SLAM (Ours, async.)," confirm the ability of our method to largely preserve the *local* and *global* estimation quality in both considered configurations under the asynchronous stereo input, resulting in an average drop in accuracy of 20 and 13 percent, respectively.

Based on the above observations, we conclude one the one hand that the absence of a continuous-time analog to conventional keyframing is one of the main performance-limiting factors to the online operation of continuous-time approaches, while on the other hand, these approaches are inherently robust against non-synchronized inputs. Furthermore, an adaptive, non-uniform spacing of bases $\mathcal{B}$ could be deployed to further reduce the computational overhead and adaptively parametrize regions of low, respectively high, volatility in a more resourceful manner, while promising to increase the *local* accuracy. Another vital milestone in the development of increasingly competitive continuous-time approaches lies with the formulation of continuous-time loop-closure strategies, which were to benefit the *global* consistency as well as the long-term reliability of CTSLAM systems. To facilitate the aforementioned modifications, we propose to use interpolating motion parametrizations, which, in contrast to non-interpolating B-Splines, would greatly simplify the dynamic insertion and removal of bases.

V. Conclusion

Following the great promise of CTSLAM over traditional, keyframe-based approaches, this work consolidates insights from stereo-visual as well as stereo-inertial SLAM and continuous-time formulations into a novel, complete framework. We demonstrate that CTSLAM achieves competitive accuracy and performance compared to the state of the art in discrete-time SLAM on established datasets. For the first time, this work also offers a direct comparison between online discrete- and continuous-time methods on these benchmark datasets, setting a precedent for fairer comparison of future continuous-time approaches. Given the lack of publicly available CTSLAM systems and benchmarks, the proposed system narrows the gap between the application of continuous-time and discrete-time formulations to the SLAM problem. Future research directions will focus on incorporating additional sensors into the proposed framework, on obtaining a more globally consistent state estimation by means of transferring and profiting from established concepts used in discrete-time methods as well as reducing the computational overhead. Overall, the proposed method and the insights arising from the experimental analysis presented here open up promising research directions towards a more integrated, multi-sensor fusion framework, capable of fusing unsynchronized inputs in a principled and generic manner.

REFERENCES

[1] R. Wang, M. Schwörer, and D. Cremers, “Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras,” in *Proc. Int. Conf. Comput. Vis.*, Venice, Italy, 2017, pp. 3923–3931.

[2] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[3] R. Mur-Artal, J. Montiel, and J. D. Tardós, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[4] P. Schmuck and M. Chli, “CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams,” *J. Field Robot.*, vol. 36, no. 4, pp. 763–781, 2019.

[5] T. Qin, J. Pan, S. Cao, and S. Shen, “A general optimization-based framework for local odometry estimation with multiple sensors,” 2019, *arXiv:1901.03638*.

[6] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual–inertial odometry using nonlinear optimization,” *Int. J. Robot. Res.*, vol. 34, pp. 314–334, 2015.

[7] T. Qin and S. Shen, “Online temporal calibration for monocular visual-inertial systems,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3662–3669.

[8] T. Qin, P. Li, and S. Shen, “VINS-Mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[9] C. Campos, R. Elvira, J. J. Gómez, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM,” *IEEE Trans. Robot.*, vol. 37, pp. 1874–1890, 2021.

[10] M. Karrer and M. Chli, “Distributed variable-baseline stereo SLAM from two UAVs,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 82–88.

[11] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, “A robust and modular multi-sensor fusion approach applied to MAV navigation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 3923–3929.

[12] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli, “GOMSF: Graph-optimization based multi-sensor fusion for robust UAV pose estimation,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 1421–1428.

[13] D. Droeschel and S. Behnke, “Efficient continuous-time SLAM for 3D lidar-based online mapping,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 5000–5007.

[14] P. Furgale, T. D. Barfoot, and G. Sibley, “Continuous-time batch estimation using temporal basis functions,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 2088–2095.

[15] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1280–1286.

[16] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, “Rolling shutter camera calibration,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1360–1367.

[17] S. Lovegrove, A. Patron-Perez, and G. Sibley, “Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras,” in *Proc. BMVC - Electron. Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 93.1–93.11.

[18] Z. Wang and L. Kneip, “Fully automatic structure from motion with a spline-based environment representation,” 2018, *arXiv:1810.12532*.

[19] E. Mueggler, G. Gallego, and D. Scaramuzza, “Continuous-time trajectory estimation for event-based vision sensors,” in *Proc. Robot.: Sci. Syst.*, Rome, Italy, Jul. 2015, doi: 10.15607/RSS.2015.XI.036.

[20] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, “Continuous-time visual-inertial odometry for event cameras,” *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1425–1440, Dec. 2018.

[21] X. Wang, F. Xue, Z. Yan, W. Dong, Q. Wang, and H. Zha, “Continuous-time stereo visual odometry based on dynamics model,” in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 388–403.

[22] K. Huang *et al.*, “B-splines for purely vision-based localization and mapping on non-holonomic ground vehicles,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5374–5380.

[23] D. Hug and M. Chli, “HyperSLAM: A generic and modular approach to sensor fusion and simultaneous localization and mapping in continuous-time,” in *Proc. Int. Conf. 3D Vis.*, 2020, pp. 978–986.

[24] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, “Efficient derivative computation for cumulative b-splines on lie groups,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11145–11153.

[25] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “SVO: Semidirect visual odometry for monocular and multicamera systems,” *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, Apr. 2017.

[26] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[27] M. Burri *et al.*, “The euroc micro aerial vehicle datasets,” *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[28] S. Anderson and T. D. Barfoot, “Towards relative continuous-time SLAM,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 1033–1040.

[29] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Sridharan, “Elastic LiDAR fusion: Dense map-centric continuous-time SLAM,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 1206–1213.

[30] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes, “Spatiotemporal Camera-LiDAR calibration: A targetless and structureless approach,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1556–1563, Apr. 2020.

[31] S. Anderson, F. Dellaert, and T. D. Barfoot, “A hierarchical wavelet decomposition for continuous-time SLAM,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 373–380.

[32] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, “Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4304–4311.

[33] Z. Zhang and D. Scaramuzza, “Rethinking trajectory evaluation for SLAM: A probabilistic, continuous-time approach,” 2019, *arXiv:1906.03996*.

[34] A. Haarbach, T. Birdal, and S. Ilic, “Survey of higher order rigid body motion interpolation methods for keyframe animation and continuous-time trajectory estimation,” in *Proc. Int. Conf. 3D Vis.*, 2018, pp. 381–389.

[35] H. Ovrén and P. Forssén, “Trajectory representation and landmark projection for continuous-time structure from motion,” *Int. J. Robot. Res.*, vol. 38, pp. 686–701, 2019.

[36] G. Cioffi, T. Cieslewski, and D. Scaramuzza, “Continuous-time vs. discrete-time vision-based SLAM: A comparative study,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2399–2406, Apr. 2022.

[37] K. Qin, “General matrix representations for b-splines,” in *Proc. Pacific Graph. 6th Pacific Conf. Comput. Graph. Appl. (Cat. No.98EX208)*, 1998, pp. 37–43.

[38] G. Wang and M. Fang, “Unified and extended form of three types of splines,” *J. Comput. Appl. Math.*, vol. 216, no. 2, pp. 498–508, 2008.

[39] S. Agarwal, K. Mierle, and T. C. S. Team, “Ceres solver,” Apache-2.0, Version 2.1, Mar. 2022. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[40] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp 121–130.

[41] J.-Y. Bouguet *et al.*, “Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm,” *Intel Corporation*, vol. 5, pp. 1–10, 2001.

[42] J. Shi and Tomasi, “Good features to track,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.

[43] S. H. Lee and J. Civera, “Triangulation: Why optimize?,” in *Proc. Brit. Mach. Vis. Conf.*, 2019.

[44] Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244–7251.

[45] M. Grupp, “evo: Python package for the evaluation of odometry and SLAM,” 2017. [Online]. Available: https://github.com/MichaelGrupp/evo