Continuous-Time vs. Discrete-Time Vision-based SLAM: A Comparative Study

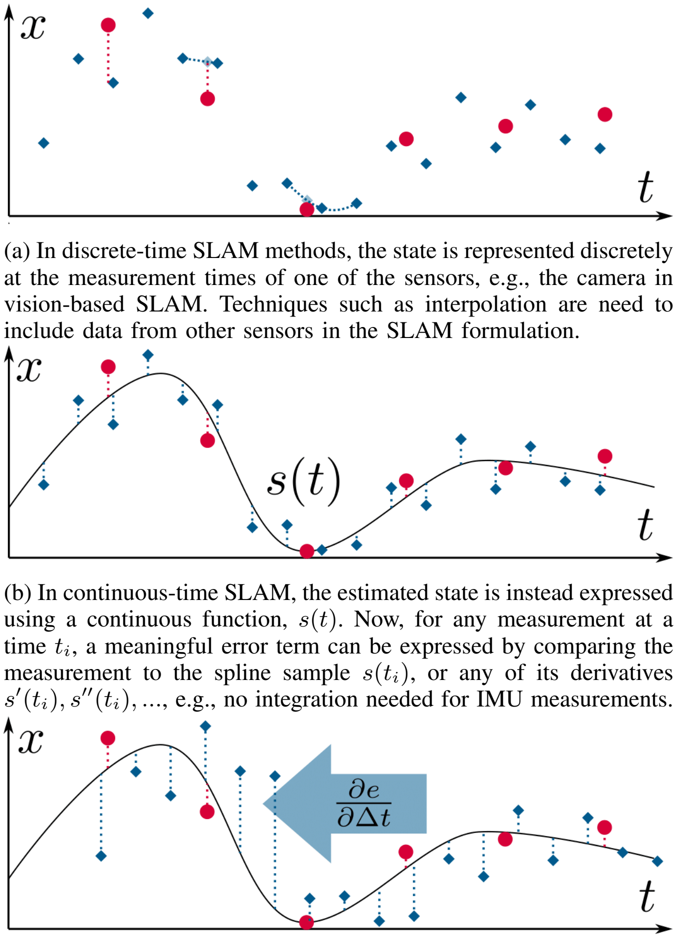
Giovanni Cioffi, Titus Cieslewski, and Davide Scaramuzza

***Abstract*—Robotic practitioners generally approach the vision-based SLAM problem through discrete-time formulations. This has the advantage of a consolidated theory and very good understanding of success and failure cases. However, discrete-time SLAM needs tailored algorithms and simplifying assumptions when high-rate and/or asynchronous measurements, coming from different sensors, are present in the estimation process. Conversely, continuous-time SLAM does not suffer from these limitations. Indeed, it allows integrating new sensor data asynchronously without adding a new optimization variable for each new measurement. In this way, the integration of asynchronous or continuous high-rate streams of sensor data does not require tailored and highly-engineered algorithms, enabling the fusion of multiple sensor modalities in an intuitive fashion. On the down side, continuous time introduces a prior that could worsen the trajectory estimates in some unfavorable situations. In this work, we aim at systematically comparing the advantages and limitations of the two formulations in vision-based SLAM. To do so, we perform an extensive experimental analysis, varying robot type, speed of motion, and sensor modalities. Our experimental analysis suggests that, independently of the trajectory type, continuous-time SLAM is superior to its discrete counterpart whenever the sensors are not time-synchronized. We release the code open-source: https://github.com/uzh-rpg/rpg_vision-based_slam**

I. Introduction and Related Work

Simultaneous localization and mapping (SLAM) is the problem of building a map of the environment and concurrently estimating the state of the robot. Among the plethora of sensors providing relevant information for localization and mapping, cameras are a very convenient solution in virtue of their information-rich measurements, low cost, and low weight. The most common vision-based SLAM formulation is based on the discrete-time (DT) trajectory representation [1]. The discrete-time formulation has the benefit of a very consolidated theory and many successful applications have been seen in the past years [1].

Although cameras can be used as the only source of information in SLAM systems, fusing multiple sensor modalities is beneficial for accuracy and robustness. In discrete-time SLAM, customized algorithms are necessary to include asynchronous measurements coming from different sources in the estimation process [2]. Similarly, ad-hoc solutions are needed to avoid adding a new state to the estimation problem every time a new measurement is available [3].



(a) In discrete-time SLAM methods, the state is represented discretely at the measurement times of one of the sensors, e.g., the camera in vision-based SLAM. Techniques such as interpolation are need to include data from other sensors in the SLAM formulation.

(b) In continuous-time SLAM, the estimated state is instead expressed using a continuous function, $s(t)$. Now, for any measurement at a time t_i, a meaningful error term can be expressed by comparing the measurement to the spline sample $s(t_i)$, or any of its derivatives $s'(t_i), s''(t_i), ...$, e.g., no integration needed for IMU measurements.

(c) Furthermore, a continuous-time representation allows the simultaneous estimation of time offset between sensors. To this end, the error terms of one of the sensors is simply expressed with s sampled at its measurement times plus a constant time offset shared among all measurements, $s(t_i + \Delta t)$. Then, Δt can be co-optimized with the parameters of s.

Fig. 1: Benefits of using a continuous-time state representation illustrated on a simple example where a variable $x(t)$ is estimated from two noisy sensors that measure x at different frequencies (red dots and blue diamonds).

In the past years, researchers have been investigating the use of continuous-time (CT) representations to encode the camera trajectory [4], [5]. The continuous-time formulation brings several advantages to the estimation problem. Firstly, continuous-time trajectories can be sampled at any time. This makes it easy to fuse asynchronous sensors and estimate time offsets. Secondly, the continuous-time formulation removes the need to include an optimization variable for every

This work was supported by the National Centre of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation (SNSF) and the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 871479 (AERIAL-CORE) and the European Research Council (ERC) under grant agreement No. 864042 (AGILEFLIGHT).

The authors are with the Robotics and Perception Group, Department of Informatics, University of Zurich, and Department of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland (http://rpg.ifi.uzh.ch).

sensor measurement. The computational complexity of the optimization problem is kept bounded, allowing to easily include high-rate sensors, such as inertial measurement units (IMU), in the estimation process. However, the continuous-time representation introduces a prior on the smoothness of the trajectory. Modeling this prior such that it can generalize to different levels of the trajectory smoothness is not an easy task.

To the best of our knowledge, there is no systematic comparison between the continuous- and discrete-time formulations for vision-based SLAM. Such systematic analysis is fundamental to guide the robotic practitioners in the design of future SLAM solutions. Therefore, we perform an extensive quantitative analysis to understand the respective advantages and limitations of the two trajectory representations. We focus on batch SLAM with visual, inertial, and global positional (i.e., Global Positioning System (GPS)) measurements. We run experiments in both hardware-in-the-loop simulation and on real-world trajectories of flying robots.

Our experiments indicate that discrete-time and continuous-time representations produce equivalent results when the sensors are time-synchronized. However, when there is an offset in the time synchronization, continuous-time is superior. The main reason behind this result is that the simplifying assumptions made for estimating the time offsets in discrete time do not always hold.

II. Methodology

We solve the estimation problem using a multi-step approach that involves a few initialization steps before the full-batch optimization.

We denote with W the fixed world frame, whose z axis is aligned with the gravity. B is the moving body frame. We set it equal to the IMU frame. C is the camera frame. P is the GPS antenna position. The position, orientation, and velocity of B with respect to W at time t_k are written as $\mathbf{p}^w_{b_k} \in \mathbb{R}^3$, $\mathbf{R}^w_{b_k} \in \mathbb{R}^{3\times3}$ part of the 3-D rotation group $SO(3)$, and $\mathbf{v}^w_{b_k} \in \mathbb{R}^3$, respectively. We use 4×4 matrices, $\mathbf{T} \in SE(3)$ (the special Euclidean group) to express 6-DOF Euclidean transformations. The time t^c_i is the time offset between camera and IMU such that $t_{imu} = t_{cam} + t^c_i$. Using the same convention, t^g_i is the GPS-IMU time offset.

The initial camera poses and 3-D landmarks are obtained by using COLMAP [6] and are expressed in the scaleless reference frame G.

A. Continuous-time representation

We use two B-splines to represent the position, $\mathbf{p}(u) \in \mathbb{R}^3$, and orientation, $\mathbf{R}(u) \in SO(3)$, of the trajectory, where u is the uniform time representation proposed in [7].

1) Initialization: The first step of our continuous-time trajectory estimation pipeline is to fit a B-spline to the K camera poses estimated by COLMAP obtaining the continuous-time trajectory ${}^c\mathcal{T}^g_c = \{\mathbf{p}^g_{c_i}, \mathbf{R}^g_{c_i}\}$.. We then transform the camera poses in body poses: ${}^c\mathcal{T}^g_b = \{\mathbf{p}^g_{b_i}, \mathbf{R}^g_{b_i}\}$.

The second step of our continuous-time trajectory estimation pipeline is to estimate the actual scale of the trajectory ${}^c\mathcal{T}^g_b$ as well as to find a transformation that aligns it to the gravity aligned frame. When GPS measurements are available, we obtain an initial estimation of the 6-DOF transformation $\mathbf{T}^w_g$ and scale s using the method proposed in [8]. Then, $\mathbf{T}^w_g$, s, $\mathbf{p}^b_p$, and t^g_i, are estimated by minimizing the difference between the GPS measurements, $\bar{\mathbf{p}}^w_{p_j}$, and the predicted GPS antenna positions sampled from the spline.

In the case when we do not use GPS measurements, we integrate the IMU measurements for a short period of time, usually few seconds, to obtain a small trajectory segment. This trajectory is expressed in a gravity aligned frame, I, which is estimated from the accelerometer measurements collected when the IMU is static. Similarly as before, we use [8] to obtain the transformation $s, \mathbf{T}^i_g$. This transformation is applied to transform ${}^c\mathcal{T}^g_b$ to the frame I.

2) Full-batch optimization: We use $\mathbf{T}^w_g$ and s estimated in the initialization step to transform the trajectory ${}^c\mathcal{T}^g_b$ to the global frame W (or I in the case when GPS is not used): ${}^c\mathcal{T}^w_b = \{\mathbf{p}^w_{b_i}, \mathbf{R}^w_{b_i}\}$. Similarly, the 3-D landmarks $\mathbf{p}^g_{l_r}$ are also transformed to W: $\mathbf{p}^w_{l_r} = s\mathbf{R}^w_g\mathbf{p}^g_{l_r} + \mathbf{p}^w_g$. In the full-batch optimization, the state vector ${}^c\mathcal{X} = \{{}^c\mathcal{T}^w_b, \mathcal{L}, t^c_i, \mathbf{T}^c_i, t^{\mathrm{g}}_i, \mathbf{p}^b_p, \mathbf{g}^w, {}^c\mathcal{B}\}$, is estimated by minimizing the cost function

$$\min_{{}^c\mathcal{X}} \sum_{k=1}^{K} \sum_{r\in\mathcal{R}_k} \left\|\mathbf{e}^{\mathrm{v}}_{k,r}\right\|^2_{\mathbf{W}_{\mathrm{v}}} + \sum_{m=1}^{M} (\left\|\mathbf{e}^{\mathrm{a}}_m\right\|^2_{\mathbf{W}_{\mathrm{a}}} + \left\|\mathbf{e}^{\omega}_m\right\|^2_{\mathbf{W}_{\mathrm{w}}}) + \sum_{d=1}^{D} \left\|\mathbf{e}^{\mathrm{gps}}_d\right\|^2_{\mathbf{W}_{\mathrm{gps}}} + \sum_{f=1}^{F} (\left\|\mathbf{e}^{\mathrm{b_a}}_f\right\|^2_{\mathbf{W}_{\mathrm{b_a}}} + \left\|\mathbf{e}^{\mathrm{b}_\omega}_f\right\|^2_{\mathbf{W}_{\mathrm{b}_\omega}}). \tag{1}$$

The error $\mathbf{e}^{\mathrm{v}}_{k,r}$ is the visual residuals, which describes the re-projection error of the landmark $\mathbf{p}^w_{l_r}$. The set $\mathcal{R}_k$ contains all the landmarks that project to the frame k. The image feature measurements $\bar{\mathbf{z}}^{k,r}$ are obtained from COLMAP. The quantity $\mathbf{e}^{\mathrm{a}}_m$ is the m-th accelerometer residual. The quantity $\mathbf{e}^{\omega}_m$ is the m-th gyroscope residual. We use cubic B-splines to represent accelerometer and gyroscope biases, $\mathbf{b}_a(u)$ and $\mathbf{b}_\omega(u)$ as in [4]. The errors $\mathbf{e}^{\mathrm{b_a}}_f$ and $\mathbf{e}^{\mathrm{b}_\omega}_f$ are residuals on the rate of the bias changes. The quantity $\mathbf{e}^{\mathrm{gps}}_d$ is the d-th GPS residual. The matrices $\mathbf{W}$ are the weights of the residuals.

B. Discrete-time representation

In the discrete-time formulation, the trajectory is represented by the body poses at the rate of the camera: ${}^d\mathcal{T}^w_b = \{\mathbf{p}^w_{b_k}, \mathbf{R}^w_{b_k}\}$. The camera-IMU time offset is estimated using the method proposed in [2]. This method proposes to shift the 2-D image features to account for the time offset between camera and IMU measurements. It makes the assumption that the camera motion has constant velocity in a short period of time (e.g., between consecutive frames), and, based on this assumption, it calculates the velocity of the 2-D features on the image plane. This velocity is then used to shift the feature position in the small period of time that corresponds to the camera-IMU time delay (see Eq. (4) in [2]). To define the GPS errors, the trajectory is interpolated at the time of the GPS measurements. The IMU-GPS time offset is taken into account in the interpolation factor similarly as in [9].

1) Initialization: Similarly to Sec. II-A1, we compute the body poses from the camera poses estimated by COLMAP and then transform them to the world frame W using the 6-DOF and scale transformation obtained by applying [8].

2) Full-batch optimization: Using a similar probabilistic SLAM formulation as in Sec. II-A2, we derive the cost function to minimize

$$\min_{^d\mathcal{X}} \sum_{k=1}^{K} \sum_{r \in \mathcal{R}_k} \left\| \mathbf{e}_{k,r}^{\mathrm{v}} \right\|_{\mathbf{W}_{\mathrm{v}}}^2 + \sum_{k=1}^{K} \left\| \mathbf{e}_{k}^{\mathrm{i}} \right\|_{\mathbf{W}_{\mathrm{i}}}^2 + \sum_{k=1}^{K} \left(\left\| \mathbf{e}_{k}^{\mathrm{b_a}} \right\|_{\mathbf{W}_{\mathrm{b_a}}}^2 + \right.$$

$$\left. \left\| \mathbf{e}_{k}^{\mathrm{b}_\omega} \right\|_{\mathbf{W}_{\mathrm{b}_\omega}}^2 \right) + \sum_{d=1}^{D} \left\| \mathbf{e}_{d}^{\mathrm{gps}} \right\|_{\mathbf{W}_{\mathrm{gps}}}^2 , \tag{2}$$

The state vector is $^d\mathcal{X} = \{^d\mathcal{T}_b^w, \mathcal{V}_b^w, \mathcal{L}, t_i^c, \mathbf{T}_i^c, t_i^g, \mathbf{p}_p^b, {}^d\mathcal{B}\}$. The set $\mathcal{V}_b^w$ contains the velocity vectors: $\mathbf{v}_{b_k}^w$. The set $^d\mathcal{B}$ contains the accelerometer and gyroscope bias vectors: $\mathbf{b}_{\mathrm{a}_k}$ and $\mathbf{b}_{\omega_k}$. The initial 3-D landmarks positions in W are obtained similarly as described in II-A2. The quantities $\mathbf{e}_{k,r}^{\mathrm{v}}$ and $\mathbf{e}_{d}^{\mathrm{gps}}$ are the the reprojection and GPS errors, respectively. The quantities $\mathbf{e}_{k}^{\mathrm{i}}$ are the inertial residuals computed as proposed in [3].

III. Experiments

We compare the continuous- and discrete-time representations in terms of accuracy of the estimated trajectory and time offsets. We use the metrics [10]: positional absolute trajectory error (ATE$_\mathrm{P}$) [m], and rotational absolute trajectory error (ATE$_\mathrm{R}$) [deg].

A. Hardware-in-the-Loop Simulation: EuRoC Dataset

The EuRoC dataset [11] contains sequences recorded onboard a hex-rotor flying robot equipped with a stereo camera and an IMU. This dataset is well-known for accurate ground-truth and hardware synchronizated sensors. We only use the sequences labeled with V_, which contain 6-DOF ground-truth from a motion capture system. We simulate GPS measurements by downsampling and corrupting the ground-truth positions with zero-mean Gaussian noise. The rate of the simulated GPS measurements is 10 Hz and the standard deviation of the Gaussian noise is 0.1 m.

1) Ablation study on the B-spline: We conducted a study to evaluate the effects of the order and frequency of the control points of the B-spline on the trajectory and camera-IMU time offset estimates. The initial value of the time offset was set to 0 ms. The results of the ablation study on the order of the B-spline are in Table I. A B-spline of order 6, which results in a cubic polynomial encoding accelerations, is needed to correctly estimate the camera-IMU time offset. This conclusion agrees with the findings in [12]. An order higher than 6 does not have any effect on the estimation results.

TABLE I: Ablation study on the order of the B-spline.

Order	**Ev. metric**	**EuRoC sequence**					
		V101	**V102**	**V103**	**V201**	**V202**	**V203**
4	ATE$_\mathrm{P}$ [m]	**0.024**	**0.014**	**0.011**	**0.011**	0.011	0.024
	ATE$_\mathrm{R}$ [deg]	**5.5**	**2.1**	**2.3**	**0.6**	0.8	1.0
	t_i^c [ms]	0.9	3.2	**-1.4**	10.8	1.0	2.2
5	ATE$_\mathrm{P}$ [m]	**0.024**	**0.014**	**0.011**	**0.011**	**0.010**	0.019
	ATE$_\mathrm{R}$ [deg]	**5.5**	2.2	**2.3**	0.9	**0.7**	0.8
	t_i^c [ms]	0.3	-5.8	2.0	-1.8	**0.0**	0.5
6	ATE$_\mathrm{P}$ [m]	**0.024**	**0.014**	**0.011**	0.012	**0.010**	**0.010**
	ATE$_\mathrm{R}$ [deg]	**5.5**	**2.1**	**2.3**	0.8	**0.7**	**0.6**
	t_i^c [ms]	**0.2**	**1.3**	**-1.4**	**1.2**	**0.0**	**0.2**

TABLE II: Study on the freq. of the B-spline control nodes.

Node freq.	**Ev. metric**	**EuRoC sequence**					
		V101	**V102**	**V103**	**V201**	**V202**	**V203**
5 Hz	ATE$_\mathrm{P}$ [m]	**0.023**	**0.014**	**0.011**	**0.010**	**0.010**	0.019
	ATE$_\mathrm{R}$ [deg]	5.6	**2.1**	2.3	0.9	0.8	0.8
	t_i^c [ms]	-1.9	-2.8	1.7	4.0	2.6	-1.5
10 Hz	ATE$_\mathrm{P}$ [m]	0.024	**0.014**	**0.011**	0.012	**0.010**	**0.010**
	ATE$_\mathrm{R}$ [deg]	**5.5**	**2.1**	2.3	**0.8**	**0.7**	**0.6**
	t_i^c [ms]	0.2	1.3	-1.4	1.2	**0.0**	**0.2**
20 Hz	ATE$_\mathrm{P}$ [m]	0.025	**0.014**	**0.011**	**0.010**	**0.010**	**0.010**
	ATE$_\mathrm{R}$ [deg]	**5.5**	**2.1**	**2.2**	1.2	**0.7**	**0.6**
	t_i^c [ms]	-2.4	**0.7**	**-0.9**	**-0.7**	-1.1	2.0
100 Hz	ATE$_\mathrm{P}$ [m]	0.024	0.226	0.117	0.060	0.168	0.136
	ATE$_\mathrm{R}$ [deg]	8.8	8.4	12.1	6.6	11.1	5.6
	t_i^c [ms]	**0.0**	-4.1	-3.0	0.0	-1.3	-0.5

We set the order of the spline to 6 and performed an ablation study on the control node frequency. The results are in Table II. The values of ATE and t_i^c suggest that a frequency of 10 Hz is the optimal choice. We conclude that order 6 and control nodes frequency 10/20 Hz are appropriate parameters for B-spline representing trajectories in this dataset.

2) Evaluation of the trajectory representation: In this set of experiments, we evaluated the continuous- and discrete-time trajectory representations in terms of ATE$_\mathrm{P}$, ATE$_\mathrm{R}$ and accuracy in estimating the camera-IMU time offset. Following the findings of Sec. III-A1, we used B-spline of order 6 and control node frequency of 10 Hz. To evaluate the accuracy in estimating the camera-IMU time offset, we simulated delays in the camera data stream, similarly to [2]. We ran experiments with delays of 0, 10, and 20 ms. The results of this comparison are in Table III. These results suggest that when the camera and IMU are time-synchronized both trajectory representations produce similar accuracy.

When the camera and IMU are not time-synchronized, continuous time is the best trajectory representation. This representation can properly estimate the time offset and produces ATE similar to the case of time-synchronized sensors. In particular, the discrete-time representation suffers in estimating the camera-IMU time offset in fast trajectories. The reason for this result is the assumption of camera motion with constant velocity in the period of time between consecutive camera frames, which is needed to compute the camera-IMU time offset. For agile motion, this assumption is no longer accurate.

B. Actual GPS with an outdoor flying robot

This dataset, courtesy of [13], contains outdoor flights of a flying robot equipped with a time-synchronized VI sensor, and a GPS receiver. The ground-truth position is provided by a Leica total station. Fig. 2 shows the first trajectory of the dataset. The ATE$_\mathrm{P}$ and the time offset estimates are in Table IV. For the continuous-time case, we used B-splines of order 6 and control node frequency of 10 Hz as suggested by the results in Sec. III-A. These results confirm the findings of Sec. III-A: when the sensor are time-synchronized the two trajectory representations produce similar results, as shown by the similar values of ATE$_\mathrm{P}$ and camera-IMU time offset.

C. Outdoor Trajectory: Ground robot

This experiment contains an evaluation of a ground robot trajectory. The robot is equipped with a time-synchronized

TABLE III: Comparison of the CT and DT approaches on the EuRoC dataset. $\mathrm{ATE_P}$ in [m], $\mathrm{ATE_R}$ in [deg], $\hat{t}_i^c$ (ground-truth) and t_i^c (estimated) time offset are in [mm].

Seq.	**Continuous-time**						**Discrete-time**					
	$\hat{t}_i^c$ = 0 [ms]		$\hat{t}_i^c$ = 10 [ms]		$\hat{t}_i^c$ = 20 [ms]		$\hat{t}_i^c$ = 0 [ms]		$\hat{t}_i^c$ = 10 [ms]		$\hat{t}_i^c$ = 20 [ms]	
	$\mathrm{ATE_P}$ / $\mathrm{ATE_R}$	t_i^c	$\mathrm{ATE_P}$ / $\mathrm{ATE_R}$	t_i^c	$\mathrm{ATE_P}$ / $\mathrm{ATE_R}$	t_i^c	$\mathrm{ATE_P}$ / $\mathrm{ATE_R}$	t_i^c	$\mathrm{ATE_P}$ / $\mathrm{ATE_R}$	t_i^c	$\mathrm{ATE_P}$ / $\mathrm{ATE_R}$	t_i^c
V101	0.024 / **5.5**	0.2	0.024 / **5.5**	11.0	0.024 / **5.5**	22.2	**0.016** / 5.6	0.3	**0.016** / 5.6	9.1	**0.016** / 5.6	18.6
V102	**0.014** / **2.1**	1.3	**0.014** / **2.1**	9.7	**0.014** / **2.1**	20.6	0.024 / 2.4	0.0	0.026 / 2.3	4.6	0.031 / 2.2	9.3
V103	**0.011** / **2.3**	-1.4	**0.011** / **2.3**	11.8	**0.011** / **2.3**	22.3	0.018 / 2.7	0.0	0.020 / 2.6	3.5	0.024 / 2.6	7.2
V201	0.012 / **0.8**	1.2	0.010 / 0.9	9.7	0.010 / 0.9	19.0	**0.009** / 1.0	0.3	0.010 / 1.0	8.1	0.012 / 1.0	16.4
V202	**0.010** / **0.7**	0.0	**0.010** / **0.7**	10.0	**0.010** / **0.7**	20.0	0.019 / 0.8	0.0	0.021 / 0.9	8.5	0.024 / 1.1	16.7
V203	**0.010** / **0.6**	0.2	**0.010** / **0.6**	10.6	**0.010** / **0.6**	21.5	0.033 / 1.1	0.0	0.036 / 1.2	4.0	0.040 / 1.3	7.5

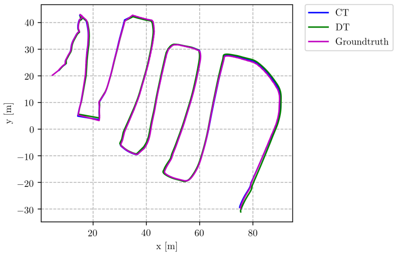


Fig. 2: XY-view of Seq. 1 in the flying robot dataset.

TABLE IV: Comparison of CT and DT approaches in the outdoor flying robot dataset.

Err. metric	**Traj. repr.**	**Seq. 1**	**Seq. 2**
$\mathrm{ATE_P}$ [m]	**CT**	**0.39**	**0.50**
	DT	0.60	0.86
$\mathbf{t_i^c}$ [ms]	**CT**	0.4	0.6
	DT	0.4	0.4
$\mathbf{t_i^g}$ [ms]	**CT**	-87.0	-118.0
	DT	-81.0	-119.0

monocular camera, IMU and GPS [1]. The 3-D position ground-truth is provided by a RTK-GPS system. Fig. 3 shows the traveled trajectory of the robot. Both approaches produce similar $\mathrm{ATE_P}$ as reported in Table V. The estimated time offsets are similar for all the configurations listed in Table V. In the continuous-time case, t_i^c, and t_i^g are -1.5 ms, and -26.0 ms, respectively. In the discrete-time case, they are -0.8 ms, and -36.3 ms. These results show that the findings of the experiments with a flying robot also apply to the case of a ground robot.

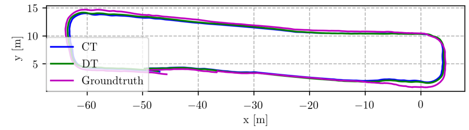


Fig. 3: XY-view of the trajectory traveled by the ground robot.

TABLE V: Comparison of CT and DT approaches in the outdoor ground robot dataset.

	Freq. [Hz]	**Order**		
		5	**6**	**7**
CT	10	0.93	0.92	0.93
	20	1.01	0.95	0.99
	100	0.80	0.89	**0.78**
DT	0.87			

IV. Conclusions

The objective of this work is to compare continuous vs. discrete vision-based SLAM formulations to guide practitioners in the development of SLAM algorithms. We find that when the camera and IMU are time-synchronized the two representations produce similar results. When a delay is present between the two measurement streams, the continuous-time representation is able to recover an accurate estimate of the time offset and consequently, produces lower ATE. In contrast, the discrete-time formulation fails in estimating the time offset, particularly when the robot moves fast, which consequently leads to high values of the ATE. The main reason of this result is that the assumption, which is necessary to estimate the camera-IMU time offset, of constant velocity of the camera motion in the period of time between consecutive camera frames does not always hold.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] T. Qin and S. Shen, "Online temporal calibration for monocular visual-inertial systems," in *IROS*, 2018.

[3] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, 2017.

[4] P. Furgale, T. D. Barfoot, and G. Sibley, "Continuous-time batch estimation using temporal basis functions," in *ICRA*, 2012.

[5] 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.

[6] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[7] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative b-splines on lie groups," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020.

[8] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pat. Anal. Mach. Intell.*, 1991.

[9] W. Lee, K. Eckenhoff, P. Geneva, and G. Huang, "Intermittent gps-aided vio: Online initialization and calibration," in *ICRA*, 2020.

[10] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *IROS*, 2018.

[11] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Research*, vol. 35, no. 10, pp. 1157–1163, 2015.

[12] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IROS*, 2013.

[13] J. Surber, L. Teixeira, and M. Chli, "Robust Visual-Inertial Localization with Weak GPS Priors for Repetitive UAV Flights," in *ICRA*, 2017.

[1] https://www.fixposition.com/