

# LoLa-SLAM: Low-latency LiDAR SLAM using Continuous Scan Slicing

Mojtaba Karimi, Martin Oelsch, Oliver Stengel, Edwin Babaïans, and Eckehard Steinbach

**Abstract**—Real-time 6D pose estimation is a key component for autonomous indoor navigation of Unmanned Aerial Vehicles (UAVs). This paper presents a low-latency LiDAR SLAM framework based on LiDAR scan slicing and concurrent matching, called LoLa-SLAM. Our framework uses sliced point cloud data from a rotating LiDAR in a concurrent multi-threaded matching pipeline for 6D pose estimation with high update rate and low latency. The LiDAR is actuated using a 2D Lissajous spinning pattern to overcome the sensor's limited FoV. We propose a two-dimensional roughness model to extract the feature points for fine matching and registration of the point cloud. In addition, the pose estimator engages a temporal motion predictor that assists in finding the feature correspondences in the map for the fast convergence of the non-linear optimizer. Subsequently, an Extended Kalman Filter (EKF) is adopted for final pose fusion. The framework is evaluated in multiple experiments by comparing the accuracy, latency, and the update rate of the pose estimation for the trajectories flown in an indoor environment. We quantify the superior quality of the generated volumetric map in comparison to the state-of-the-art frameworks. We further examine the localization precision using ground truth pose information recorded by a total station unit.

## I. INTRODUCTION

The demand for unmanned aerial vehicles (UAVs) for autonomous exploration and inspection is growing. The compact design-factor, relatively low cost, and maneuverability of UAVs make them well suited for various tasks [1]. Autonomous UAVs require a reliable navigation system to operate in challenging environments, such as GPS-denied or cluttered indoor areas [2], [3]. Tackling the general navigation mission, one needs to address a set of problems ranging from 6D pose estimation to trajectory planning [4], [5]. In practice, the navigation task's performance depends mainly on the accuracy, update rate, and latency of the perception unit (see Fig. 1). For instance, the trajectory tracking controller of the UAV requires a high-rate pose estimation (in the range of 0.1 to 1.0 kHz) with comparatively low latency (maximum 50 ms) to achieve a fully autonomous flight [6]–[8]. In this context, latency is the time difference from the sensor observation until the localization algorithm provides the estimated pose.

This paper was recommended for publication by Editor O. Allison and Editor P. Pounds upon evaluation of the Associate Editor and Reviewers' comments. This work is supported by the German Aerospace Center (DLR) with funds from the Federal Ministry of Economic Affairs and Energy (BMWi) on the basis of a resolution of the German Bundestag under the reference '20X1707C'.

All authors are with the Department of Electrical and Computer Engineering, Technical University of Munich (TUM), Germany. {mojtaba.karimi, martin.oelsch, oliver.stengel, edwin.babaïans, eckehard.steinbach}@tum.de

This paper has supplementary downloadable multimedia material (20 MB in size) available at <http://ieeexplore.ieee.org>, provided by the authors.

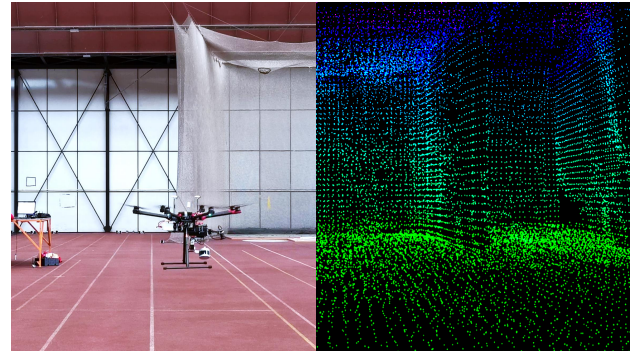


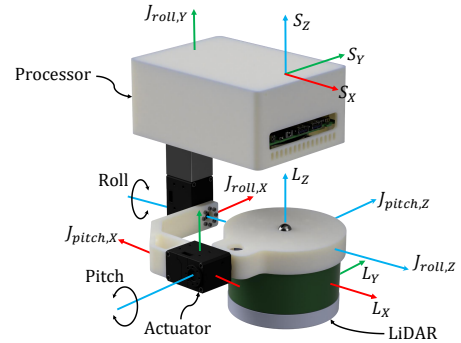
Fig. 1: Real-time localization of a UAV in an indoor environment using the proposed SLAM framework. It provides low-latency 6D pose estimation with a high temporal update rate for autonomous navigation while generating a full-scale 3D map of the environment.

Visual-inertial localization and mapping is well studied for drone localization as it can meet the aforementioned requirements [9], [10]. However, in large indoor areas, the accuracy of visual-inertial localization approaches tends to degrade dramatically [11], [12]. This is mainly due to the considerable distance between the camera and the scene, which causes feature tracking failures in the camera frames. As an alternative, laser scanning technology is employed to capture precise range measurements enabling LiDAR-based localization and mapping systems. To address the 6D pose estimation task, three-dimensional simultaneous localization and mapping (3D SLAM) is the preferred way. 3D SLAM is mainly addressed in the literature by using multiple laser scanners, which are installed in both horizontal and vertical frames [13], [14]. However, due to the weight and processing constraints in UAVs, in practice, the use of an actuated LiDAR is preferred rather than installing multiple or heavier sensors with an inherently larger FoV [15], [16]. In this regard, different spinning mechanisms have been developed to be mounted on drones [17], [18]. Although generally successful, existing solutions come with some shortcomings, such as long revisit time, blind spots, inconsistencies in scanning the environment, and skewing issues. These shortcomings cause problems in determining the correspondences within the point cloud data for 6D positioning [19], [20].

While LiDAR-based SLAM has been studied widely in the past decade for mobile robots [19], [21]–[24], there is still a major gap in utilizing these systems for real-time navigation of UAVs. This is mainly because it is not possible to rely on an aerial vehicle remaining sufficiently motionless. The drone position must be represented at a high temporal update rate with low-latency to achieve stable

closed-loop control. Addressing the challenge mentioned above, continuous-time trajectory estimators were developed in the literature [25], [26]. However, due to the limited FoV and the substantial time needed for a complete scan period of the LiDAR, estimating continuous odometry and global registration introduces a significant latency and low-fidelity 6D pose estimation in such a system. To address this issue, LiDAR-inertial SLAM systems use an additional Inertial Measurement Unit (IMU) to produce high-rate pose updates [24]. They provide acceptable results for ground-based mobile robots. However, as UAVs experience high dynamic motions and inevitable high-frequency vibrations due to the propellers rotation, IMU-based approaches leads to having a fluctuating pose estimation in such applications [27].

## II. PROBLEM STATEMENT AND SENSOR PLATFORM



### III. LIDAR ACTUATION USING LISSAJOUS PATTERN

dedicated actuators in order to increase the FoV. However, such observation requires a non-negligible amount of time to capture. This becomes problematic for drones while they are floating in the air. Periodic actuation of the LiDAR sensor in roll and/or pitch direction repeats the observation of the surface patches and allows for a larger FoV. To enhance this procedure, researchers have developed non-raster scan actuation motions, called Lissajous pattern [28], [29], [33]. Considering the fact that a state-of-the-art rotating LiDAR provides a constant sampling rate, the main benefit of using a Lissajous scan pattern-based actuation of the LiDAR is the reduced revisit time (by 54%) within the defined FoV [29]. In addition, compared to the other actuation models, avoiding the orderly nature of the raster scan by using a simultaneously periodic sinusoidal trajectory can be defined as another advantage of the Lissajous pattern.

Motivated by these works, we propose a 2D Lissajous pattern for a rotating 360° LiDAR. This pattern allows us to rotate the LiDAR to observe the vertical and horizontal surfaces equitably. We designed and developed an actuated LiDAR payload with 2-DoF, as shown in Fig. 2, to be able to precisely execute this pattern. Based on the systematic study of the scan skewing problem [34], we developed a mechanism that actuates the laser scanner in a combined roll and pitch motion around a centralized axes to minimize the measurement distortion. The platform is manufactured using printed ABS material. The proposed 2D Lissajous scan pattern at time  $t$  for the roll ( $\alpha$ ) and pitch ( $\beta$ ) axes are defined by

$$\begin{aligned}\alpha(t) &= A_{roll}(t) \sin(2\pi f_c t) , \\ \beta(t) &= A_{pitch}(t) \cos(2\pi f_c t) ,\end{aligned}\quad (1)$$

where the frequency  $f_c$  is the period of one full sweep, and the amplitudes  $A_{roll}$  and  $A_{pitch}$  are defined as the maximum angles of the motion. We set the period of one full sweep to 8.0 seconds and the maximum amplitude to 45°. These values are obtained heuristically based on the dynamics of the drone and the FoV of the sensor. We use a slow start mechanism, which increases the amplitude of the Lissajous actuation pattern from zero to its maximum value within the first three sweeps. We use the smooth start mainly to prevent damage to the rotating laser scanner as the sudden and fast motion will cause hardware damage in long term use. Although it is not necessary, this smooth

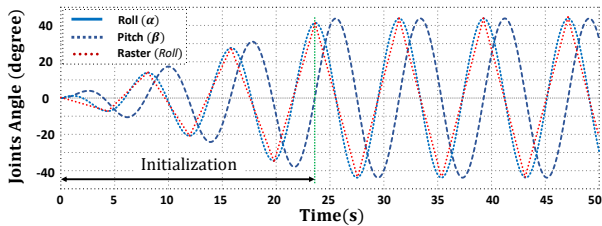


Fig. 3: The proposed 2D Lissajous pattern for the roll and pitch joints of the sensor platform. The frequency of the pattern is defined to perform one sweep in 8.0 seconds. In the start, a smooth increase in the amplitude is used to minimize potential hardware damage and also to generate the initial map.

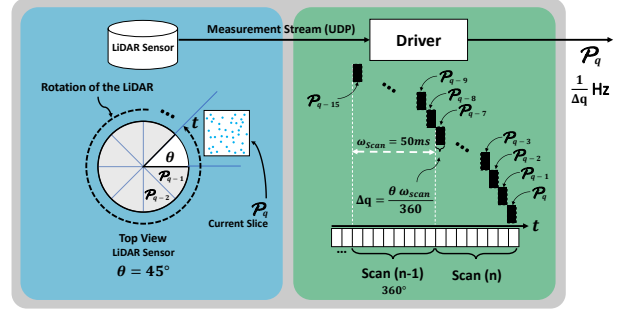


Fig. 4: Continuous Slicing Point Cloud (CSPC) model. This model gathers smaller collections of measurements that can be used in a concurrent pipeline instead of the normally used full 360° scan. Each slice  $P_q$  is defined for the last  $\theta^\circ$  of the rotation of the LiDAR.

increase of the amplitude at the start, while the robot is stationary (e.g., before takeoff), helps in the initialization of the SLAM algorithm. This is because, at the start, there is a large number of the newly observed points, which must be added to the map. The generated Lissajous pattern in comparison with a single axis raster scan is illustrated in Fig. 3.

#### IV. CONTINUOUS SCAN SLICING

The laser scanner used in this work rotates with a maximum of 1200 rounds per minute (rpm) and provides 300,000 points per second. A 360° scan is available every  $\omega_{scan} = 50$  milliseconds and contains 15 thousand measurements. The laser scanner is continuously measuring and creates a constant output stream associated with time and angle. These measurements are accumulated over time at the driver level. When the laser scanner completes a full 360° scan, it releases all the recorded points as a single scan. State-of-the-art SLAM frameworks are developed based on these 360° scans for ego-motion estimation [22], [23], [27]. In practice, the point cloud arrival rate defines the frequency of the pose estimation. However, the ego-motion estimation can be performed after any other measurement, which means 360° is an arbitrary margin, and individual measurements can be bundled freely.

Although it is reasonable to compare complete 360° scans in the odometry unit; we show later in the proposed localization framework that our SLAM model does not depend on the odometry unit. The proposed SLAM framework uses continuous slices of a full 360° scan for low-latency localization using direct scan matching on the global map. The main intention of the proposed continuous slicing point cloud (CSPC) is to gather smaller collections of measurements that can be used in a successive manner instead of waiting for a full 360° scan. In this context, describing the term continuous in other words, we collect the measurements from the upcoming scan into a small point cloud slice for the last  $\theta$  angle (i.e., only the data of the measurements within the last  $\theta = 45^\circ$  of the rotation), and publish them one after another. As shown in Fig. 4, slicing of the points is developed in the driver level of the laser scanner, and the CSPC is published at a rate of  $\frac{1}{\Delta q}$  Hz. Considering each





The geometric properties of the point cloud  $\mathcal{P}_{(q,i)}^S$  in the sensor coordinate frame (i.e., roughness and normal vector) remain unchanged after performing the rigid transformations. Before the feature extraction, we use linear interpolation between the predicted pose and the previously estimated pose to remove the distortion of the point cloud similar to the approach from [22]. We extract the roughness value as a feature representation in this work. In this context, roughness value defines a term to evaluate the smoothness of the local surface. The feature extraction process is similar to the method used in [21]. However, instead of extracting the one-dimensional roughness of the points along the LiDAR's beam-lines, we extract a two-dimensional roughness value from the organized point cloud. We extract the two-dimensional roughness value by taking all neighboring points within a predefined spherical area and calculate the roughness value in both vertical and horizontal directions. First, we find the set of adjacent points using the nearest neighbor search along the KD-tree. To improve the system's performance, as the size of each slice is smaller compared to the full scan mode, we filter correspondences to maintain small patches using ties broken by distance. This allows us to reduce the estimation time of the feature points for each slice. We define the two-dimensional roughness value  $\mathcal{Z}$  for the point  $i$  by calculating the differences of the neighboring point set  $\mathcal{N}_{(q,j),n}^S$  and normalize the term according to the distance as

$$\mathcal{Z}_i = 1 - \left\| \frac{1}{n\mathcal{P}_{(q,i)}^S} \sum_{j=1}^n \mathcal{N}_{(q,j),n}^S \right\|, \quad (5)$$

where  $n$  is the number of the considered nearest neighbors of point  $i$ . The feature point set  $\hat{\mathcal{D}}_{(q,i)}^M$  is stored with the corresponding calculated roughness value. Recalling that each slice is a small portion of one full rotation of the LiDAR, the extracted feature points, which are sorted based on their roughness value, are well distributed within the observed point cloud for the entire FoV. In addition, we use the local entropy to assess the saliency of the feature points [35], and we remove those that have a higher local entropy value. This procedure helps us to avoid selecting points that have a high roughness score while not being reliable feature points. This can be inferred from the fact that their nearby points are not well aligned. For a simple example, consider the points distributed along with the objects on a crowded table. These points might have a high roughness score, but they are not as reliable as the corner points on the wall's edge for the localization. Fig. 6 illustrates the result of the proposed method for feature extraction.

### C. Low-Latency Localization and Mapping

*Pose Estimation:* The localization framework proposed in this paper uses the predicted pose as an initial condition. It matches the feature points of the sliced scan into the map and extracts the corresponding transformation. This procedure is concurrently performed on each of the combined sliced point sets  $\mathcal{D}_q^S$ . The output of this process are synchronous individual pose estimates. The combined feature point set

$\mathcal{D}_q^S$  can be determined by

$$\mathcal{D}_q^S = {}^M\mathbf{T}_{S,q-1}^{-1} \mathcal{D}_{q-1}^M + \tilde{\mathcal{D}}_q. \quad (6)$$

The estimated poses are later used for updating the state estimator in the EKF. Considering the point sets  $\mathcal{D}_q^S$  and  $\mathcal{M}_q^M$  in correspondence  $\mathcal{D}_q^S \leftrightarrow \mathcal{M}_q^M$ , and that they are related via a rigid body transform, in each thread we seek to estimate  $\mathbf{R}$  and  $\mathbf{t}$  such that

$$\mathcal{M}_q^M = \mathbf{R}\mathcal{D}_q^S + \mathbf{t} = {}^M\mathbf{T}_{S,q}(t)\mathcal{D}_q^S, \quad (7)$$

where the problem is well-studied with various closed-form solutions in the literature. To solve for the pose within the map, we need to establish a geometric relationship between  $\mathcal{D}_q^M$  and  $\mathcal{M}_q^M$ . Using the predicted transformation  ${}^M\hat{\mathbf{T}}_{S,q}(t + \Delta q)$ , which we extracted from the previous observations of the motion, we first transform the feature points in  $\mathcal{D}_q^S$  into the map coordinate frame using

$$\hat{\mathcal{D}}_q^M = {}^M\hat{\mathbf{T}}_{S,q}(t + \Delta q) \mathcal{D}_q^S. \quad (8)$$

Both of the point sets now are in the same coordinate frame, but they are not aligned perfectly as we used only the predicted transformation. However, we can find the nearest neighbor points for each feature point  $i$  in  $\hat{\mathcal{D}}_q^M$  within the map point set. It is worth to mention that some of the points do not have any correspondence on the map. This means the closest point in the map is far from the selected feature point in  $\hat{\mathcal{D}}_q^M$ ; therefore, we filter them using a simple threshold value based on Euclidean distance. Now we can derive a geometric relationship between all of the corresponding selected points in  $\hat{\mathcal{D}}_q^M$ , and the nearest points in the map as a non-linear function

$$f_{q,i}(\hat{\mathcal{D}}_{q,i}^M, {}^M\mathbf{T}_{S,q}) = \mathbf{d}_i, \quad i \in \hat{\mathcal{D}}_q^M \in \mathcal{M}_q^M, \quad (9)$$

where stacking Eq. 9 for each feature point in  $\mathcal{D}_q^S$ , we can obtain a non-linear function  $f_q$  and rewrite it as

$$f_q(\mathcal{D}_q^S, {}^M\hat{\mathbf{T}}_{S,q}^{-1} {}^M\mathbf{T}_{S,q}) = \mathbf{d}, \quad q \in \mathbb{Z}^+, \quad (10)$$

where  $\mathbf{d}$  contains the corresponding Euclidean distances for each point, and each row of  $f$  is associated with one feature point. We use the Levenberg-Marquardt trust-region algorithm as a non-linear least squares solver for this problem [36]. In our optimization problem, a groups of scalars must be converged together. This means the three components of a translation vector and the four components of the Quaternion that define the sensor's pose are used as a parameter set. We use the Huber loss function to reduce the influence of outliers on the non-linear least squares solver's solution. To solve Eq. 10, we compute the Jacobian matrix  $\mathbf{J}$  for the non-linear function  $f$  with respect to  ${}^M\mathbf{T}_{S,q}$  where

$$\mathbf{J} = \frac{\partial f}{\partial {}^M\mathbf{T}_{S,q}}, \quad (11)$$

and we further solve Eq. 10 with non-linear iteration toward minimizing the  $\mathbf{d}$  to zero.

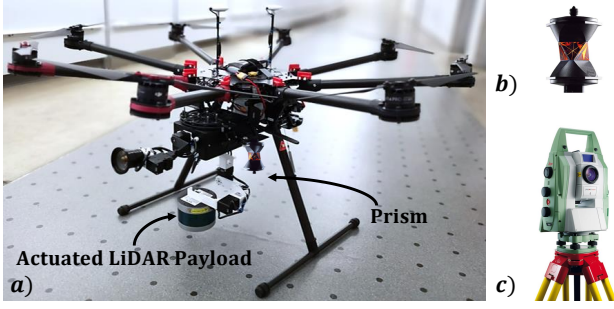


Fig. 7: a) A UAV platform with an on-board computing unit and an active LiDAR payload. b) The 360° reflector prism for tracking of the UAV c) The Leica TS60 Tachymeter total station.

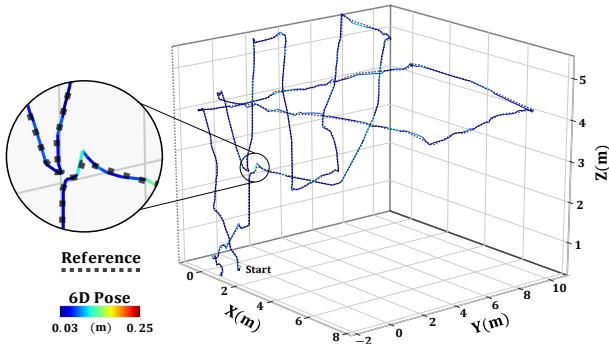


Fig. 8: Absolute trajectory error of the UAV in a sports hall in comparison to the GT pose trajectory from the total station.

**Linear Pose Predictor:** We use a linear prediction based on the observed history of the estimated pose as an initial transformation for finding the correspondences within the map. We define the predicted transformation  ${}^M\hat{\mathbf{T}}_{S,q}(t + \Delta q) = (\hat{\mathbf{R}}[\Theta^{-1}], \hat{\mathbf{t}})$  using the linear prediction model

$${}^M\hat{\mathbf{t}}_{S,q}(t + \Delta q) = \sum_{i=0}^b \mathbf{A}_i {}^M\mathbf{t}_{S,q}(t - i\Delta q), \quad (12)$$

$${}^M\hat{\mathbf{R}}_{S,q}(t + \Delta q) = {}^M\mathbf{R}_{S,q}[\sum_{i=0}^b \mathbf{A}_i \Theta^{-1}(t - i\Delta q)], \quad (13)$$

where  $b$  indicates the number of previous observations,  $\mathbf{A}_i = \lambda_{i,b} \mathbf{I}_3$  is the predictor coefficients, and  $\Theta$  is the element-wise Euler angle vector representation that is used to calculate the rotation matrix. Because the previously observed poses are equally-spaced values in time, a polynomial interpolation can be defined as a linear combination of the given observation. The predictor coefficient  $\lambda_{i,b}$  is a scalar multiplication for the identity matrix  $\mathbf{I}_3$ . In our linear model these elements defined by the  $i$ th component of the  $b$ th row of Pascal's triangular matrix of the binomial coefficients [37].

## VI. EXPERIMENTAL EVALUATION

To assess the performance of the proposed framework, we tested our approach on a recorded data set from a large sports hall. To evaluate the pose estimation accuracy, an octocopter equipped with the sensor payload is tracked with a Leica TS60 Tachymeter total station. The total station in tracking mode has a stated measurement accuracy of 0.2 cm at a

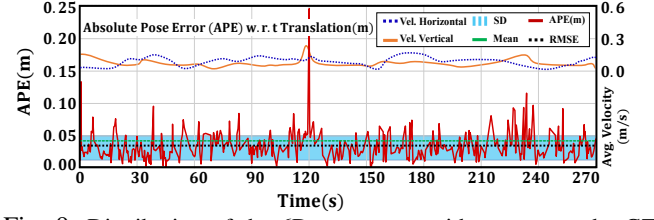


Fig. 9: Distribution of the 6D pose error with respect to the GT data of the total station. The root mean square error (RMSE) for this trajectory is equal to 3.6 cm.

rate of 10 Hz. To enable tracking with the total station, a 360° reflector prism is mounted on the octocopter as shown in Fig. 7. The static transformation is applied to have the total station's measurements in the same coordinate system as the UAV. The octocopter is flown multiple times with different trajectories in a sports hall. Flight duration in total is about 36 minutes, and the velocity varies from 0.05 to 6.0 km/h. The data were recorded applying Lissajous, raster, and static modes. In the static mode, the LiDAR was held in a horizontal state. In the other two modes, the LiDAR was actuated using the defined patterns. Five data acquisition options are used for the LiDAR point cloud; the default mode (Full) is a 360° scan at 20 Hz, and the other modes are using scan slicing with a segment size of 30, 45, 90, and 180 degrees, respectively.

An example of 6D trajectory tracking using LoLa-SLAM and the total station is illustrated in Fig. 8. The results show that there is less deviation in the horizontal direction compared to the vertical direction. We can see that the larger errors mostly appear when the drone moves fast in the vertical direction. This issue could have two origins; First, for the linear predictor, as the map matching uses this prediction as an initial input in the localization step, a poor prediction could cause such a wrong matching. Second, imperfect observations caused by the LiDAR actuation: due to the insufficient points in the vertical face compared to the horizontal. To investigate more on this problem, we further compared the proposed method with the static LiDAR; this means even fewer points are observed in the vertical faces, and as expected, the error in the vertical direction increased considerably. In addition to this, as shown in Fig. 9, the RMSE of a selected trajectory is only 3.6 cm; however, due to the fast motion, the maximum pose error increased up to 26.8 cm at some points.

Furthermore, the latency and accuracy of the pose estima-

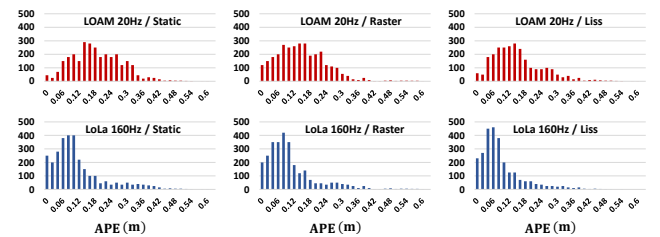


Fig. 10: Histogram of the absolute pose error (APE) distribution when performing active-controlled hovering (PD controller) with 20 Hz and 160 Hz pose feedback.



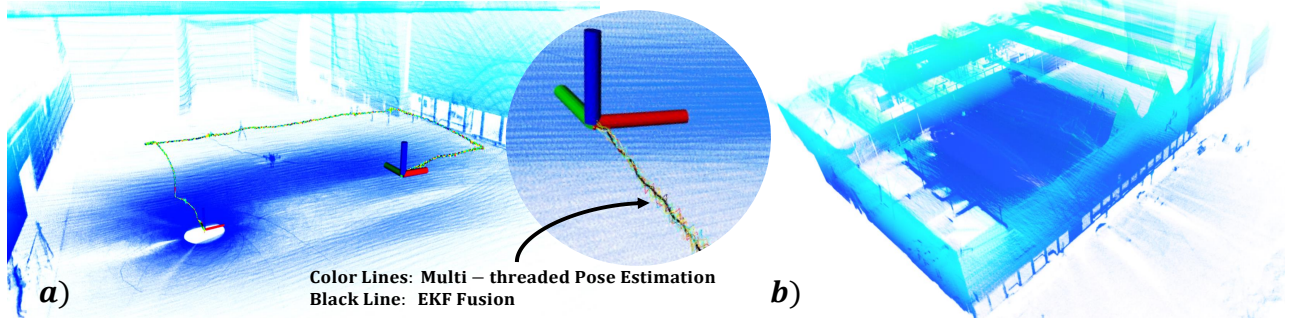


Fig. 11: Reconstructed 3D map of a sports hall using the LoLa-SLAM framework. *a)* Concurrent pose estimations from the multi-threaded localization and the fused 6D pose of the UAV. *b)* The dense point cloud map from a different perspective.

TABLE I: Results of multiple experiments using the recorded data set in a sports hall. All the experiments are conducted using the onboard Core.i7 Intel processor - 4 Cores and with 16 GB of RAM.

Experiment ( $\theta^\circ$ + Pattern)	Total Dist. (m)	CPU Usage Avg.	RMSE - R ( $^\circ$ )	RMSE - T (m)	Max Error (m)	SD (m)	Update Rate (Hz)	Latency (ms)
<b>30°+Lissajous</b>	184	-	-	<b>Failed</b>	-	-	-	-
<b>30°+Raster</b>	175	-	-	<b>Failed</b>	-	-	-	-
<b>30°+Static</b>	193	92%	7.4	<b>0.079</b>	0.48	0.041	<b>240</b>	<b>16</b>
<b>45°+Lissajous</b>	184	85%	4.3	<b>0.039</b>	0.27	0.028	<b>160</b>	<b>19</b>
<b>45°+Raster</b>	175	83%	4.4	<b>0.057</b>	0.41	0.033	<b>160</b>	<b>21</b>
<b>45°+Static</b>	193	77%	5.8	<b>0.070</b>	0.42	0.034	<b>160</b>	<b>18</b>
<b>90°+Lissajous</b>	184	84%	4.4	<b>0.036</b>	0.27	0.033	<b>80</b>	<b>39</b>
<b>90°+Raster</b>	175	83%	4.4	<b>0.055</b>	0.43	0.032	<b>80</b>	<b>43</b>
<b>90°+Static</b>	193	75%	5.8	<b>0.067</b>	0.46	0.034	<b>80</b>	<b>38</b>
<b>180°+Lissajous</b>	184	55%	4.1	<b>0.035</b>	0.25	0.027	<b>40</b>	<b>78</b>
<b>180°+Raster</b>	175	51%	4.9	<b>0.052</b>	0.39	0.029	<b>40</b>	<b>85</b>
<b>180°+Static</b>	193	52%	3.7	<b>0.066</b>	0.38	0.031	<b>40</b>	<b>77</b>
<b>LOAM Full+Lissajous</b>	184	53%	4.6	<b>0.034</b>	0.26	0.024	<b>20</b>	<b>93</b>
<b>LOAM Full+Raster</b>	175	56%	4.8	<b>0.050</b>	0.44	0.031	<b>20</b>	<b>97</b>
<b>LOAM Full+Static</b>	193	49%	3.9	<b>0.065</b>	0.47	0.033	<b>20</b>	<b>89</b>

tion are investigated. In Table. I, we provide a comparison to the well known state-of-the-art SLAM framework LOAM [21]. By analyzing the results, we can see a trade-off between the chosen slice size, pose estimation accuracy, and the update rate. Choosing a smaller slice size results in a higher update rate; however, this means fewer features in each slice are used for localization. Consequently, the accuracy of the pose estimation starts deviating. In an extreme scenario, with using the slice size  $\theta = 30^\circ$ , the system cannot track the pose, and therefore the localization starts drifting. If the drift is larger than 0.5 meter, we consider the localization to fail.

Considering a slice size  $\theta = 45^\circ$  is a satisfactory candidate, we achieve a latency of less than 20 ms on average, which is acceptable for real-time drone navigation. In com-

parison, using the full scan mode, we observe a latency of 93 ms. Besides, we investigate the use of LoLa-SLAM in a closed-loop system while the drone was hovering using a PD controller. The histogram of the APE distribution is shown in Fig. 10. The figure shows that, while the accuracy of the pose estimation is lower using LoLa-SLAM compared to LOAM, the drone deviation is smaller due to the shorter latency and higher temporal update rate. While LoLa-SLAM provides a higher update rate and lower latency, it is computationally more expensive than the other approaches. For instance, on average, the proposed framework needs around 30% more processing power compared to LOAM. However, thanks to the concurrent multi-threaded map matching, the system still runs in real-time.

The accuracy and overall validity of the proposed system are additionally assessed using the created dense 3D map. To evaluate the map quality, Fig. 11 shows the resulting point cloud from different perspectives, which allows for better interpretation of the scene. The tripod of the total station and pillars in the sports hall can be identified in the globally aligned and voxelized 3D map. It is worth stating that this level of 3D mapping is not obtainable without a precise localization. To evaluate the superior quality of the created map compared to LOAM, we further illustrate the density comparison between the point clouds similar to the method described in [25]. The point densities show the uniformity of the generated map. The distribution of the point spacing (nearest-neighbor distance) for randomly samples points is visualized in Fig. 12. The densities are represented as distributions of point spacing, as determined by each point's nearest neighbor. We use randomly selected 500K points within each map and determined the density in a sphere with 0.2 m area around each point. This comparison shows that LoLa-SLAM has the highest proportion of points in the distance of less than 3.0 cm.

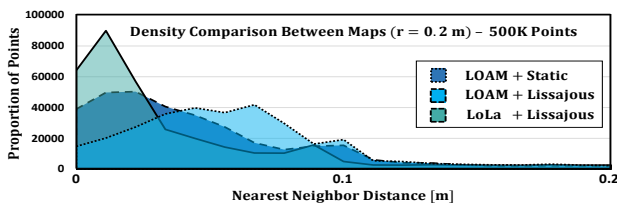


Fig. 12: Density comparison between the 3D maps. The distribution of the point spacing (nearest-neighbor distance) for randomly selected 500K points within each map.

## VII. CONCLUSIONS

In this work, we presented a customized system to localize a UAV in an indoor GPS-denied environment. We approached this challenge by employing an actuated LiDAR platform and utilizing the 2D Lissajous spinning pattern. We further developed a novel SLAM framework based on the continuous sliced point cloud to reduce the ego-motion estimation latency, named LoLa-SLAM. We showed that the

proposed SLAM system's accuracy and reliability remain adequate, and this is while the temporal update rate increased by at least 2X for the pose estimation. Besides, we demonstrated the proposed framework could simultaneously generate a high-quality 3D map. We demonstrate both the low-latency pose estimation and the overall accuracy of LoLa-SLAM by multiple experiments in an indoor environment. We showed that the update rate increases by choosing a smaller slice size; however, the proposed system can fail in the feature matching process if the slice size is too small. In addition, we examined the computational complexity which increases by 30% compared to the common methods.

In future work, to further improve the system, the odometry predictor and the EKF fusion can be integrated with redundant attitude and heading reference systems (AHRS) [24], [38]. Furthermore, adding an adaptive slice size controller based on the UAV's velocity can also be investigated to improve the system performance. Besides, environment aware attentive control of the LiDAR can be employed to improve the point cloud observation, map generation, and reliable pose estimation in complex indoor environments.

## REFERENCES

- [1] J. Miranda, S. Larnier, A. Herbulot, and M. Devy, "Uav-based inspection of airplane exterior screws with computer vision," in *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019.
- [2] F. Valenti, D. Giaquinto, L. Musto, A. Zinelli, M. Bertozzi, and A. Broggi, "Enabling computer vision-based autonomous navigation for unmanned aerial vehicles in cluttered gps-denied environments," in *International Conference on Intelligent Transportation Systems*, 2018.
- [3] J. Siva and C. Poellabauer, "Robot and drone localization in gps-denied areas," in *Mission-Oriented Sensor Networks and Systems: Art and Science*. Springer, 2019, pp. 597–631.
- [4] Y. D. Yasuda, L. E. G. Martins, and F. A. Cappabianco, "Autonomous visual navigation for mobile robots: A systematic literature review," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–34, 2020.
- [5] X. Zhou, Z. Yi, Y. Liu, K. Huang, and H. Huang, "Survey on path and view planning for uavs," *Virtual Reality & Intelligent Hardware*, vol. 2, no. 1, pp. 56–69, 2020.
- [6] S. Jung, S. Hwang, H. Shin, and D. H. Shim, "Perception, guidance, and navigation for indoor autonomous drone racing using deep learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, 2018.
- [7] R. S. Dimitrova, M. Gehrig, D. Brescianini, and D. Scaramuzza, "Towards low-latency high-bandwidth control of quadrotors using event cameras," *arXiv preprint arXiv:1911.04553*, 2019.
- [8] R. Pérez-Alcocer and J. Moreno-Valenzuela, "A novel lyapunov-based trajectory tracking controller for a quadrotor: Experimental analysis by using two motion tasks," *Mechatronics*, vol. 61, pp. 58–68, 2019.
- [9] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, 2017.
- [10] D. Van Opdenbosch, M. Oelsch, A. Garcea, T. Aykut, and E. Steinbach, "Selection and compression of local binary features for remote visual slam," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7270–7277.
- [11] S. Lynen, B. Zeisl, D. Aiger, M. Bosse, J. Hesch, M. Pollefeys, R. Siegwart, and T. Sattler, "Large-scale, real-time visual-inertial localization revisited," *arXiv preprint arXiv:1907.00338*, 2019.
- [12] M. W. Achtelik, S. Lynen, S. Weiss, L. Kneip, M. Chli, and R. Siegwart, "Visual-inertial slam for a small helicopter in large outdoor environments," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2651–2652.
- [13] D. Zhang, Z. Gong, Y. Chen, J. Zelek, and J. Li, "Slam-based multi-sensor backpack lidar systems in gnss-denied environments," in *Geoscience and Remote Sensing Symposium*. IEEE, 2019.
- [14] M. Velas, M. Spanel, T. Slezziak, J. Habrovec, and A. Herout, "Indoor and outdoor backpack mapping with calibrated pair of velodyne lidars," *Sensors*, vol. 19, no. 18, p. 3944, 2019.
- [15] J. Lin and F. Zhang, "Loam.livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov," *arXiv preprint arXiv:1909.06700*, 2019.
- [16] —, "A fast, complete, point cloud based loop closure for lidar odometry and mapping," *arXiv preprint arXiv:1909.11811*, 2019.
- [17] H. Qin, Y. Bi, K. Z. Ang, K. Wang, J. Li, M. Lan, M. Shan, and F. Lin, "A stereo and rotating laser framework for uav navigation in gps denied environment," in *Industrial Electronics*. IEEE, 2016.
- [18] W. Zhen and S. Scherer, "A unified 3d mapping framework using a 3d or 2d lidar," *arXiv preprint arXiv:1810.12515*, 2018.
- [19] H. Qin, Z. Meng, W. Meng, X. Chen, H. Sun, F. Lin, and M. H. Ang, "Autonomous exploration and mapping system using heterogeneous uavs and ugvs in gps-denied environments," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1339–1350, 2019.
- [20] R. Voges, C. S. Wiegardt, and B. Wagner, "Timestamp offset determination between an actuated laser scanner and its corresponding motor," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, p. 99, 2017.
- [21] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.
- [22] —, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.
- [23] X. Ji, L. Zuo, C. Zhang, and Y. Liu, "Lloam: Lidar odometry and mapping with loop-closure detection based correction," in *International Conference on Mechatronics and Automation*. IEEE, 2019.
- [24] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," *arXiv preprint arXiv:2007.00258*, 2020.
- [25] L. Kaul, R. Zlot, and M. Bosse, "Continuous-time three-dimensional mapping for micro aerial vehicles with a passively actuated rotating laser scanner," *Journal of Field Robotics*, vol. 33, no. 1, 2016.
- [26] S. Anderson and T. D. Barfoot, "Towards relative continuous-time slam," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1033–1040.
- [27] M. Bosse, R. Zlot, and P. Flick, "Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1104–1119, 2012.
- [28] J. W. Anderson and G. M. Clayton, "Lissajous-like scan pattern for a gimbalized lidar," in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechanisms*. IEEE, 2014, pp. 1171–1176.
- [29] M. Benson, J. Nikolaidis, and G. M. Clayton, "Lissajous-like scan pattern for a nodding multi-beam lidar," in *ASME 2018 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers Digital Collection, 2018.
- [30] D. Droschel, M. Nieuwenhuisen, M. Beul, D. Holz, J. Stückler, and S. Behnke, "Multilayered mapping and navigation for autonomous micro aerial vehicles," *Journal of Field Robotics*, vol. 33, no. 4, 2016.
- [31] W. Zhen, S. Zeng, and S. Soberer, "Robust localization and localizability estimation with a rotating laser scanner," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2017.
- [32] H. Nakagomi, Y. Fuse, H. Hosaka, H. Miyamoto, T. Nakamura, A. Yoneyama, M. Yokotsuka, A. Kamimura, H. Watanabe, T. Tanzawa et al., "3d scan matching for mobile robot localization over rough terrain," *Electrical Engineering in Japan*, vol. 209, no. 3-4, 2019.
- [33] M. T. Benson, H. Sathishchandra, G. M. Clayton, and S. B. Andersson, "Compressive sensing-based reconstruction of lissajous-like nodding lidar data," in *Dynamic Systems and Control Conference*, vol. 59162. American Society of Mechanical Engineers, 2019, p. V003T21A010.
- [34] A. Al-Nuaimi, W. Lopes, P. Zeller, A. Garcea, C. Lopes, and E. Steinbach, "Analyzing lidar scan skewing and its impact on scan matching," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2016, pp. 1–8.
- [35] G. Li, Y. Geng, and W. Zhang, "Autonomous planetary rover navigation via active slam," *Aircraft Engineering and Aerospace Tec*, 2018.
- [36] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [37] C. K. Williams, "Prediction with gaussian processes: From linear regression to linear prediction and beyond," in *Learning in graphical models*. Springer, 1998, pp. 599–621.
- [38] M. Karimi, E. Babaian, M. Oelsch, T. Aykut, and E. Steinbach, "Skewed-redundant hall-effect magnetic sensor fusion for perturbation-free indoor heading estimation," in *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, 2020.