The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-1/W1-2023
12th International Symposium on Mobile Mapping Technology (MMT 2023), 24–26 May 2023, Padua, Italy

SIMULATION OF LOW-COST MEMS-LIDAR AND ANALYSIS OF ITS EFFECT ON THE PERFORMANCES OF STATE-OF-THE-ART SLAMS

F. Vultaggio*, F. d'Apolito, C. Sulzbachner, P. Fanta-Jende

AIT - Austrian Institute of Technology - Center for Vision, Automation and Control – Assistive Autonomous Systems (francesco.vultaggio, francesco.dapolito, christoph.sulzbachner, phillipp.fanta-jende)@ait.ac.at

KEY WORDS: LiDAR SLAM, LiDAR simulation, SLAM benchmark, Gazebo, MEMS LiDAR, Indoor UAV.

ABSTRACT:

Indoor Unmanned Aerial Vehicles have often been tasked with performing SLAM, and the sensors most used in literature and industry have been cameras. Spanning from stereo to event cameras, visual algorithms have often been the de facto choice for localization. While visual SLAM has reached a high level of accuracy in localization, accurate map reconstruction still proves to be challenging. Meanwhile, LiDAR sensors have been used for years to obtain accurate maps. First in surveying applications and in the past ten years in the automotive sector. The weight, power, and size constraints of most traditional LiDARs have prevented their installation on UAVs for indoor use. MEMS-based LiDARs have already been used in UAVs but had to rely on algorithms designed to deal with their small FOV. Recently, a MEMS-based LiDAR with a wide field of view (360°*59°) and weighing 265 g has sparked interest in its potential for indoor UAV SLAM. We performed an extensive battery of tests in simulation environments to provide a first look into its effect on state-of-the-art SLAM algorithms, highlight which ones can provide the best results, and what improvements may be most beneficial. This paper aims to provide assistance in further research in the field by releasing the tool used for this work.

1. INTRODUCTION

In environments where Global Navigation Satellite System (GNSS) signals are not available or reliable, Simultaneous Localization And Mapping (SLAM) is a crucial technology for Unmanned Aerial Vehicles (UAVs) to navigate and map the environment. SLAM algorithms use sensors to estimate the pose and create a map of the environment in real time. RGB cameras are often used as the primary sensor due to their low Size, Weight, And Power (SWAP) requirements, making them ideal for indoor UAVs applications. LiDAR systems offer an attractive alternative due to their greater accuracy, resilience to challenging lighting conditions, and textureless environments. Thus far, however, LiDARs have been primarily limited to automotive and Unmanned Ground Vehicles (UGVs) applications due to their high SWAP.

Recently commercialized low-weight MEMS-based LiDAR systems, such as the Livox Mid-360[1], which are lighter and more power efficient, have opened new opportunities for integration in indoor UAVs. The Livox Mid-360, by leveraging a helicoidal non-repeating scanning pattern, achieves wide coverage in the horizontal and vertical field of view (360*59 degrees). The point cloud is sufficiently dense for real-time SLAM operation. Moreover, if left stationary, the scanning pattern fully saturates the environment making the sensor viable for dense map generation.

An essential tool for the continued development of LiDAR centric SLAM systems is the availability of simulation tools. To this end, we leveraged the recently released pattern of the Livox Mid-360[2] to implement a custom plugin for the simulation of the Livox Mid-360 and performed an extensive campaign of tests. We are releasing this plugin to the benefit of the SLAM community[3]. n contrast to other works simulating the Livox Mid-360 (Kong et al., 2023), our plugin integrates with simulation environments that do not rely on highly detailed, and manually refined, point cloud maps. Using meshes to build the maps allows for the testing of SLAM systems that use both cameras

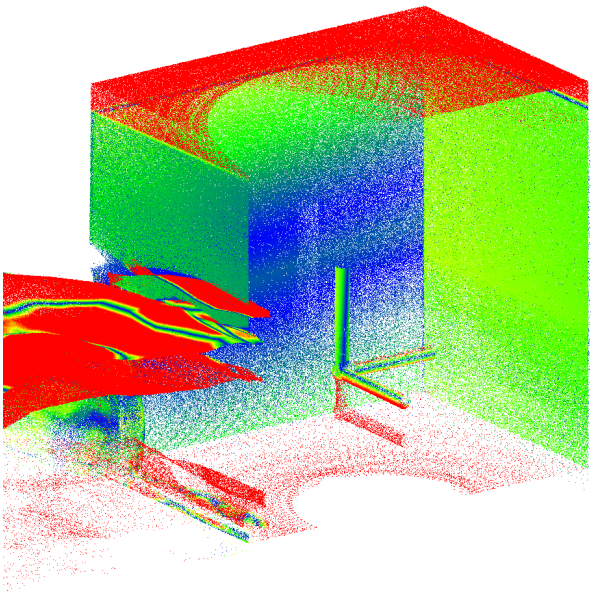

Figure 1. Example of point cloud where each point in red represents a point distant more than 20 cm from the ground truth. Final map of the *Cave* generated by Fast-LIO 2 using data from the Livox Mid-360

* Corresponding author

[1] https://www.livoxtech.com/mid-360

[2] https://github.com/Livox-SDK/livox_laser_simulation

[3] https://github.com/fratopa/Mid360_simulation_plugin

and LiDAR systems. The goal of this work is to provide insights into the potential benefits and challenges of using the Livox Mid-360 sensor in SLAM systems, particularly for indoor UAV applications.

Traditional SLAM benchmarking efforts, while extensive, have focused on odometry performances, e.g. (Delmerico and Scaramuzza, 2018), (Xu et al., 2022b). To provide an understanding of the SLAM system in terms of mapping quality, this contribution will offer an analysis of both the odometry and mapping performances. The former will be used to quantify the local accuracy and the latter to measure the global consistency. This comparison effort will be simplified by the fact that the meshes used for the virtual environment generation will act as ground truth for the map evaluation step.

To evaluate the differences between this novel MEMS sensor and regular optomechanical ones, simulated data coming from both kinds of sensors will be collected and used to run multiple LiDAR-centric SLAM pipelines. To generate the data coming from a regular optomechanical LiDAR we simulated the Ouster Os0-128. Table 1 shows that while its SWAP makes it a poor candidate for integration in an indoor UAV, its performance makes it an ideal candidate to act as a baseline in our benchmark.

2. METHOD

2.1 Data collection

LiDAR SLAM algorithms are often tested on datasets collected from optomechanical LiDARs mounted on cars: (Geiger et al., 2012) and (Jeong et al., 2019), or other ground vehicles (Carlevaris-Bianco et al., 2015). More recent datasets, like the ones proposed by (Giubilato et al., 2022) or (Qingqing et al., 2022), have incorporated hand-held sequences and MEMS LiDARs in their suite of sensors. While an improvement over the more traditional datasets, these lack data coming from the Livox Mid-360 and were not collected from flying platforms. In general, the motion profiles of these datasets are not representative of the challenges that UAVs would face while exploring indoor environments. To address this limitation, we attached the simulated sensor to a virtual quadcopter and piloted it along two custom maps we created.

One map is a cave-like maze, which presents a particular challenge for most algorithms due to its larger size and feature-poor environment. The second map is a house with multiple rooms of varying levels of detail, which presents a mapping challenge if improper scan-to-map registration occurs. The 3D models of the *Cave* and *House* maps are in Figure 2.

To isolate the impact of the Livox Mid-360's unique scanning pattern on each algorithm, the quadcopter mounted a traditional LiDAR sensor as well. Specifically, we aimed to mimic an Ouster Os0-128 sensor, a comparison between the two can be seen in Table 1.

With their wide vertical field of view, high density, and long range, the point clouds coming from the Os0-128-like sensor represent an ideal data source for indoor LiDAR-SLAM applications and can thus act as an effective baseline to compare the performances of the Mid-360.

The simulation of these sensors has been integrated within the open-source simulation environment Gazebo (Koenig and Howard, 2004). Compared to alternative simulation frameworks such as AirSim (Shah et al., 2017), Carla (Dosovitskiy et al., 2017), or Nvidia Omniverse[4], Gazebo offers increased performances under limited computational resources while still offering the possibility to integrate high-fidelity meshes in the scene. Alternative simulation environments support more advanced visual rendering pipelines, making them better suited for advanced visual SLAM applications. However, their limited modularity and poorer integration with the Robotic Operating System (ROS) may hinder the integration with ROS-based SLAM frameworks.

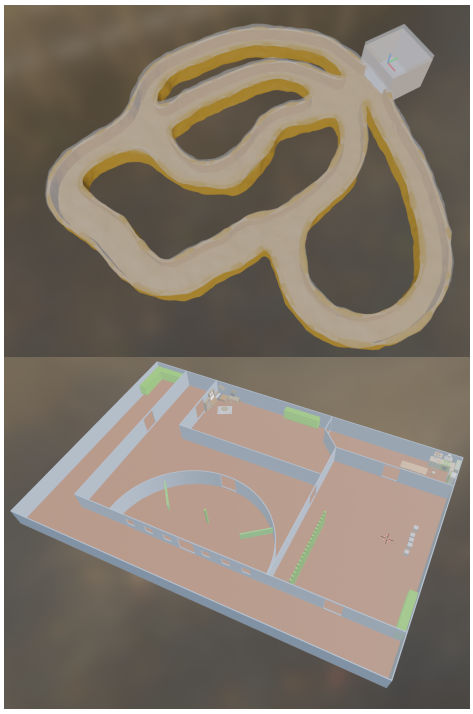

Figure 2. On top is the cave-like maze and on the bottom is the multi-room map.

	Os0-128	**Mid-360**
H-FOV (°)	360	360
V-FOV (°)	90	60
Points (hz)	2.4M	0.2M
Range (m)	100	40
Weight (g)	447	265
Power (W)	14-20	6.5 -14
Price (€)	$>$ 10k	$<$ 1k

Table 1. Comparison between the Livox Mid-360 and the Ouster Os0-128

While we used the official pattern released by the manufacturer, we did not use their plugin since, as can be seen in Figure 3, we found it to introduce distortions in the point clouds which unacceptably degraded the SLAM performances and are not present in the real sensor.

2.2 Algorithms

All tests have been performed on a workstation mounting an Intel Xeon E5-1650, 32GB of RAM, and an NVIDIA Quadro M4000.

[4] `https://www.nvidia.com/en-us/omniverse`

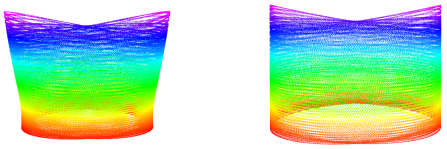

Figure 3. Point cloud of the virtual sensor in a cylindrical enclosure. On the left, the point cloud coming from the Livox official plugin while on the right the point cloud coming from our plugin

Each algorithm chosen for evaluation was required to operate in real-time and process data coming from the Os0-128 and the new Mid-360. Well-known frameworks such as F-LOAM (Wang et al., 2021) and LeGo-LOAM (Shan and Englot, 2018) were excluded from the testing campaign due to their incompatibility with the point clouds generated by the Livox Mid360. Conversely, other frameworks, like Livox-LOAM (Pan et al., 2021), could not use the point clouds generated by the Os0-128. Others, like MULLS (Pan et al., 2021), , which can process point clouds from both sensors, have not been able to do so in real-time and have been excluded too.

The algorithms tested still constitute a representative sample of established and modern techniques used in the LiDAR community. Now follows a brief introduction to each of them:

2.2.1 CT-ICP The core of the CT-ICP (Dellenbach et al., 2022) algorithm is its novel approach to the scan-to-map registration step. One of the main limitations associated with employing the naive ICP algorithm for matching successive scans stems from the distortion experienced by each new scan as the sensor undergoes movement during the acquisition process, see Figure 4. This work is not the first to implement a de-skewing

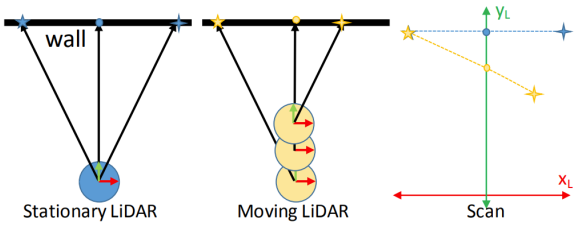

Figure 4. Scan skewing process. The ego motion of the sensors causes a non-rigid deformation of the endpoint cloud.

strategy prior to the scan matching step. However, the approach taken here is novel in that it does not rely on a constant velocity model for the sensor motion or IMU (Inertial Measurement Unit) data. This system proposes an elastic formulation of the trajectory with continuity of start and end pose in the scan acquisition step while preserving discontinuity between adjacent scans. In particular, during the de-skewing step no assumption is made about the fact that the pose of the sensor at the beginning of scan $T_b(n)$ is equal to that of at the end of the previous one $T_e(n-1)$. This approach preserves high-frequency motion while still correcting for the scan skew. In the same paper, the authors also introduce a back end pose graph optimizer based on g2o (Kümmerle et al., 2011) used to implement Loop Closure (lc), however, at the time of writing, it has not been integrated with the ROS framework and can only be used to process datasets off-line.

2.2.2 Fast-LIO2 Fast-LIO2 (Xu et al., 2022a) is a SLAM system relying on LiDAR and IMU data. Like in its previous formulation (Xu and Zhang, 2021), it still uses an iterated Extended Kalman Filter (iEKF) to deskew the incoming point clouds and to initialize the scan matching process. However, it no longer requires the feature extraction step to perform scan matching. The core strength of this new formulation is its capacity to sustain dense maps, which in turn enables efficient scan-to-map matching. While most LiDAR SLAMs rely on either sub-sampling or feature extraction to render the problem of scan matching tractable in real time, FAST-LIO2 instead introduces a tailor-made implementation of a k-d tree to store and manage the map data. K-d trees are data structures that allow for fast point matching using kNN-search. However, inserting new points in this map representation requires re-balancing the whole tree resulting in non-real-time operation, especially if incremental map generation is required. To mitigate this problem the authors implemented a self-balancing k-d tree based on the *Scapegoat Tree* idea proposed by (Galperin and Rivest, 1993). The final algorithm is able to manage extremely dense maps while still maintaining real-time operation.

2.2.3 KISS-ICP The KISS-ICP framework (Vizzo et al., 2023) aims to be a general approach applicable to any LiDAR sensor with minimal to no tuning of its parameters. It does so by having no assumption on the kind of sensor being used and avoiding hand-crafted descriptors to aid in the scan matching step. To this end KISS-ICP forgoes most of the sophisticated optimization techniques and reduces the odometry estimation loop to four steps, namely; de-skewing the incoming scan using the constant velocity model, sub-sampling of the deskewed point cloud to bound complexity, scan-to-map matching to recover the incremental odometry, and lastly updating of the local map stored using a voxel grid using the previously subsampled point cloud. During the scan-to-map matching step, the regular ICP algorithm is utilized. However, instead of establishing a maximum limit for iterations, an adaptive threshold is employed, using the point-to-point distance between the local map and the incoming scans as metric.

2.2.4 LIO-SAM LIO-SAM (Shan et al., 2021) is a tightly integrated inertial-LiDAR odometry and mapping pipeline. The IMU is used both in the de-skewing step and to provide an initial guess to the scan registration step. The integrated IMU data and the LiDAR odometry data, together with optionally loop closure and GNSS signals, are merged into a factor graph which is optimized using GTSAM (Dellaert and Kaess, 2017). The scan registration step generates the LiDAR odometry factor, i.e., each scan is registered to a local map by first extracting edge and planar features based on the local roughness (Zhang and Singh, 2014). The factor graph formulation allows for the integration of loop closure in the LIO-SAM framework. Loop detection is initialized based on the Euclidean distance between the current pose and the previous trajectory. Upon triggering, the current features are matched against the local marginalized ones, and both the path and the map are subsequently adjusted.

3. EXPERIMENTS

We used both map and odometry data to measure the impact of the Mid-360 sensor on the SLAM performances. The most popular metrics used to determine the accuracy of SLAM systems have been the Absolute Trajectory Error (ATE) and the Relative Pose Error (RPE) (Prokhorov et al., 2019). These metrics

rely on odometry data to quantify the global consistency and the local accuracy, respectively.

To compute the RPE, the reference trajectory is divided into uniformly spaced intervals of 1 m. The relative transformation $\Delta_{i,j}$ between the pose at the beginning P_i and at the end P_j of each segment is then computed for both trajectories. Finally, using the inverse compositional operator (Lu and Milios, 1997), denoted as $\ominus$, we can compute the RPE between each transformation in the reference and relative trajectory. Using the RPE is possible to quantify the drift per meter in each run and in doing so have a metric to represent the local accuracy of each SLAM pipeline.

$$RPE_{i,j} = \Delta_{est_{i,j}} \ominus \Delta_{ref_{i,j}} = \\ = (P_{ref,i}^{-1} P_{ref,j})^{-1} (P_{est,i}^{-1} P_{est,j}) \quad (1)$$

Having access to the ground truth mesh underlying each environment, we decided to use the map generated by each SLAM system to measure the global consistency instead of the trajectory.

In analogy to the ATE we

- Aligned the point cloud of each global map to the ground truth mesh
- Computed the distance between each point and the closest vertex in the mesh
- Computed the Root Mean Square Error (RMSE) over all the point-to-mesh distances

Moreover, we identified as *outliers* all the points whose distance from the mesh was greater than 20 cm. These points severely degrade the quality of the final map manifesting themselves as *ghosting* artifacts (Chamseddine et al., 2021), an example of which can be seen in Figure 1. We then computed the percentage of outliers in each final map.

3.1 Local accuracy

Table 2 shows the local accuracy of each algorithm across the two maps, it is evident that the *House* map proved to be more challenging for most algorithms compared with the *Cave*. This may be explained by the fact that the *Cave*, being more constrained, limits the space of possible motions that the UAV can undergo during its flight. In contrast, the open spaces afforded by the *House* allowed for sudden and drastic changes in both heading and speed. LiDAR-based odometry is particularly vulnerable to high-speed motion due to the low scan acquisition rate which characterizes this family of sensors. High-speed motion paired with a low scan acquisition rate yields poorly overlapping scans, even if perfectly de-skewed. In the end, the poorly overlapping scans degrade the quality of the scan-matching step inherent in any SLAM algorithm and in turn, this manifests as a degradation of the mapping performance. The interaction between a low data acquisition rate and odometry estimation has been studied in depth by the visual SLAM community (Gallego et al., 2020).

CT-ICP generates impressive results, especially considering that it is not relying on IMU data. We attribute the high local accuracy to its scan registration method which is able to preserve the high-frequency motion components of the sensor. This is especially notable looking at the performances in the *House* which display a comparable level of accuracy to LIO-SAM.

Cave					
	CT ICP	KISS ICP	LIO SAM lc	LIO SAM	Fast LIO2
Mid-360	1.78	**3.34**	2.50	2.61	<u>1.08</u>
Os0-128	**1.53**	8.17	**1.26**	**1.24**	**<u>0.49</u>**
House					
	CT ICP	KISS ICP	LIO SAM lc	LIO SAM	Fast LIO2
Mid-360	**2.72**	**6.88**	3.54	2.7	<u>1.61</u>
Os0-128	2.79	7.1	**1.65**	**1.51**	**<u>1.11</u>**

Table 2. RMSE of the drift in cm per 1 m interval.
In **bold** the best-performing sensor for each algorithm,
<u>underlined</u> the best-performing algorithm for each sensor

Examining KISS-ICP we can see the only instance where the local accuracy decreased going from the *Cave* to the *House* map. Nonetheless, it performed worse than other algorithms. It can be observed that also the other purely LiDAR-based algorithm outperforms KISS-ICP, as the constant velocity model, employed to provide the initial guess for scan-to-map registration, frequently experiences significant violations during our tests where the sensor was mounted on a UAV subjected to high speed motion.

LIO-SAM, both with and without loop closure, has been able to preserve high local accuracy in both maps. We attribute this to its use of the IMU pre-integration used to provide a first guess to the scan alignment step. The local accuracy doesn't show a significant improvement by activating the loop closure since it is only able to improve the global consistency and not the local accuracy, we will discuss its impact on global consistency in the next section.

	LIO-SAM	Fast-LIO2
Mid-360	+3.3%	+32.5 %
Os0-128	+17.9%	+55.8%

Table 3. Percentage of drift increase from *Cave* to *House* map

The performances of Fast-LIO2 are a testament to the impact that dense matching can have on LiDAR-SLAM. Fast-LIO2 shows the lowest drift of all the algorithms in either map and using either sensor. While both Fast-LIO2 and LIO-SAM rely on IMU data to remain accurate at high-speed, the use of dense matching improves the accuracy of each registration yielding more accurate odometry. However, computing the drift increase going from the *Cave* to the *House* map, see Table 3, suggests the pose graph strategy implemented by LIO-SAM may be a comparatively more robust method than the iEKF implemented by Fast-LIO 2, albeit less precise in absolute terms.

Looking at the impact of the sensor itself, it is evident that the CT-ICP algorithm's ability to perform odometry estimation in an indoor environment is largely unaffected by which sensor is being used. By storing the local map in a sparsified voxel structure the algorithm, CT-ICP, is able to reduce its computational footprint while at the same time having the side effect of uniformly processing point clouds with different densities. Conversely, LIO-SAM shows a clear preference for higher-density

point clouds, this is due to the fact that in order to extract the features it needs to perform scan matching, and higher-density point clouds are greatly beneficial in detecting the features in the first place.

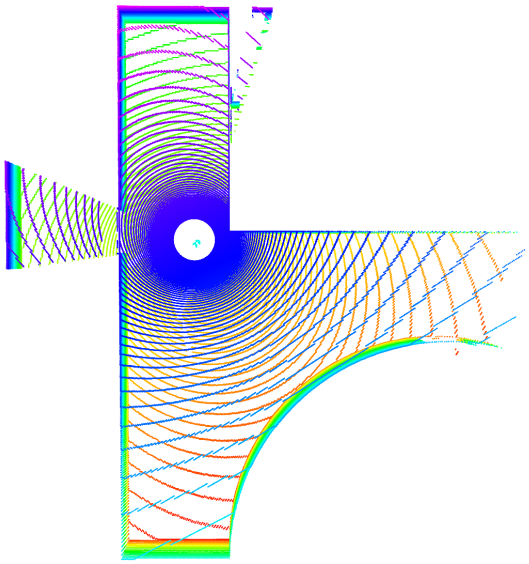

Figure 5. Scan from the Os0-128 captured in the *House* map

3.2 Global consistency

Looking at the histograms in Figures 6 and 7 we can see that the *Cave* map has been the most difficult to maintain global consistency in. All algorithms succeeded in generating an accurate final map of the *House* with both sensors. This is likely due to the fact that the many openings present in the map, combined with its smaller size, effectively turned any local map into an equivalent *global* one, this can be seen in Figure 5. Registering points to a *global* map rather than to a local one improves the overall consistency of the final map.

While CT-ICP did not show a strong preference for either of the two sensors when looking at the local accuracy, the global consistency decreases significantly using the data provided by the Mid-360, Table 4 shows a tenfold increase in the number of outliers. As the authors themselves have highlighted, local maps maintained on the distance d to the last registered scan, and not on a sliding window of the n most recent frames, are particularly susceptible to degradation of the global consistency following bad scan insertion. CT-ICP mitigates these events by using a more conservative approach to scan-to-map registration when the estimated pose changes suddenly. Sudden but small changes in motion have a greater impact on point clouds collected by sensors with longer-range capabilities, which subsequently facilitates the detection of such changes in the first place. The smaller range of the Mid-360 point cloud makes it harder to activate this procedure and thus produces ghosting artifacts.

The data collected from KISS-ICP highlight the effectiveness of the optimizations implemented in CT-ICP we just described. KISS-ICP uses a voxel structure too in order to maintain the local map, however, it does not seem to implement any strategy to mitigate possible bad scan registration. In such scenarios, the point clouds provided by the Os0-128 generate even more ghosting artifacts in the final map.

Cave					
	CT ICP	KISS ICP	LIO SAM lc	LIO SAM	Fast LIO2
Mid-360	14.95	**31.91**	3.68	**10.68**	15.59
Os0-128	**1.40**	58.25	**0.17**	21.78	**0.18**
House					
	CT ICP	KISS ICP	LIO SAM lc	LIO SAM	Fast LIO2
Mid-360	0.13	3.17	4.24	2.28	**5.92e-3**
Os0-128	**0.02**	**0.45**	**9.54e-4**	**9.34e-4**	0.02

Table 4. Percentages of outliers in each final point cloud.
In **bold** the best-performing sensor for each algorithm,
underlined the best-performing algorithm for each sensor

Cave					
	CT ICP	KISS ICP	LIO SAM lc	LIO SAM	Fast LIO2
Mid-360	20.06	**22.58**	9.99	**14.27**	19.98
Os0-128	**7.68**	51.47	**6.26**	17.19	**5.52**
House					
	CT ICP	KISS ICP	LIO SAM lc	LIO SAM	FAST LIO2
Mid-360	6.45	6.86	6.95	5.90	**5.27**
Os0-128	**4.33**	**4.88**	**3.65**	**4.19**	5.52

Table 5. RMSE in cm compued from the point-to-mesh distance.
In **bold** the best-performing sensor for each algorithm,
underlined the best-performing algorithm for each sensor

The maps generated by LIO-SAM demonstrate the reliance that this pipeline has on loop closure to maintain global consistency. From Figure 6 it is possible to see a pronounced bulge in the point-to-mesh distance histogram of the final map generated by LIO-SAM without the loop closure module active. These are due to entirely misaligned portions of the final map. Activating the loop closure realigns both the trajectory and the map. These misalignments occur infrequently but are difficult to predict, loop closure is the best tool to counteract this phenomenon. However, finding the right set of parameters for loop detection in LIO-SAM is a trade-off between real-time operation and the global consistency of the map. As the authors themselves wrote, the loop detection module, being based on the Euclidean distance between the current pose and the trajectory, is *naive but effective* and more advanced methods are present in the literature (Kim and Kim, 2018).

Fast-LIO 2 produces impressive results in both the *House* and the *Cave* environments alike. The inconsistent proportion of outliers across each of the four tests is indicative of how the dense scan-to-map registration approach is susceptible to drift accumulation which is never able to correct. Fig 1. is indicative of the results that can be expected in a worst-case scenario.

However, preserving global consistency using a dense global map comes with the caveat of considerable memory consumption. The i-k-d tree implemented in Fast-LIO 2 allows for real-time interaction with a dense global map, as far as CPU utilization is concerned. However, it does not reduce its size. Figure 8 shows the memory utilization of the algorithm over time and it is evident that the memory utilization grows linearly with time by a factor proportional to the size of the point cloud being used.

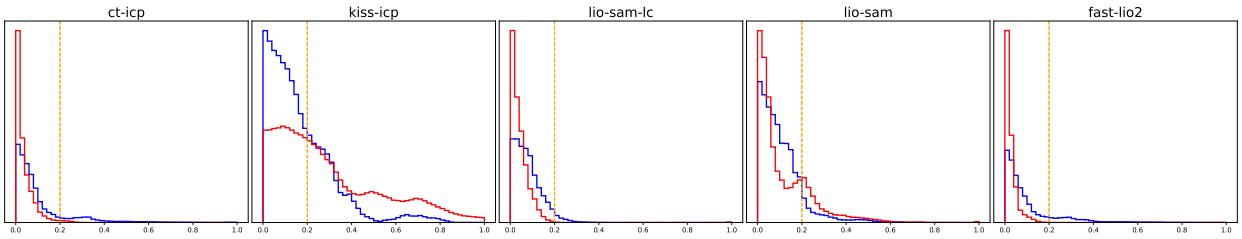

Figure 6. Histogram of the point-to-mesh distances in the *Cave* map.
In blue the distribution of the Livox Mid-360, in red the distribution of the Ouster Os0-128, in yellow the threshold set at 0.2m

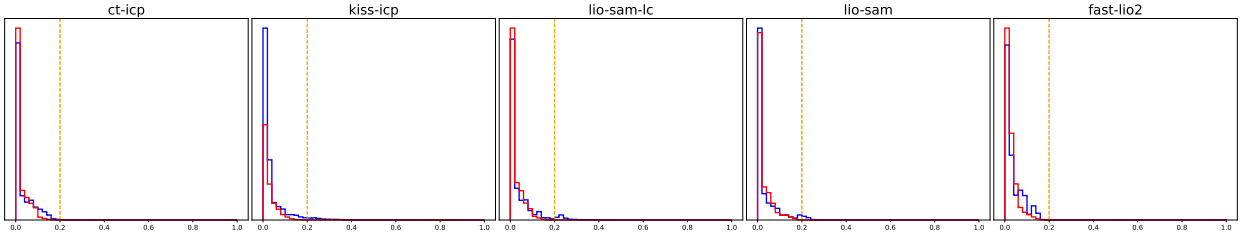

Figure 7. Histogram of the point-to-mesh distances in the *House* map.
In blue the distribution of the Livox Mid-360, in red the distribution of the Ouster Os0-128, in yellow the threshold set at 0.2m

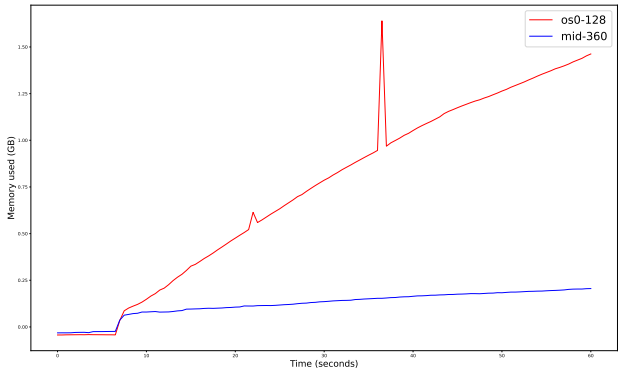

Figure 8. Memory usage of Fast-LIO2.

In our tests, after a few hundred seconds the map has grown to occupy a few gigabytes. The algorithm starts to truncate the global map only when the sensor is detected as being outside a predetermined distance from the first scan which by default is set to 1 km. This approach is inherently memory unsafe, not because of the size of the default distance, but because it is possible for a sensor to remain static and saturate the system RAM.

4. CONCLUSION

In this paper, we explore how MEMS LiDAR systems, such as the Livox Mid-360, affect the performance of popular open-source LiDAR SLAM frameworks. The data have been generated in a simulation environment using our custom plugin which is able to generate distortion-free point clouds having an arbitrarily complex pattern. Each sensor has been mounted on a UAV to test the ability of each system to handle high-speed motion. We used the odometry data to measure the local accuracy of the SLAM systems and the final maps to measure their global consistency.

Our experiments demonstrated that MEMS LiDARs show promise as sensors for indoor UAV operations. However, due to their lower density compared to traditional LiDARs, they do not work optimally with SLAM systems that rely on feature extraction. Additionally, maintaining high levels of local accuracy requires more advanced motion models than those typically used for ground vehicles. Our results suggest that integrating IMU data or employing more complex motion models are both effective strategies for improving performance. Moreover, achieving global consistency remains a challenge for most SLAM systems. Although in our tests loop correction has been shown to be highly effective, it requires more robust methods to detect the accumulated drift in the first place.

To help the research community develop SLAM systems tailored for these new setups our custom plugin will be made publicly available.

ACKNOWLEDGMENTS

Funded by
the European Union

REFERENCES

Carlevaris-Bianco, N., Ushani, A. K., Eustice, R. M., 2015. University of Michigan North Campus long-term vision and lidar dataset. *International Journal of Robotics Research*, 35(9), 1023–1035.

Chamseddine, M., Rambach, J., Stricker, D., Wasenmuller, O., 2021. Ghost Target Detection in 3D Radar Data using Point Cloud based Deep Neural Network. *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 10398–10403.

Dellaert, F., Kaess, M., 2017. *Factor Graphs for Robot Perception*. Foundations and Trends in Robotics, Vol. 6.

Dellenbach, P., Deschaud, J.-E., Jacquet, B., Goulette, F., 2022. CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure. *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 5580–5586.

Delmerico, J., Scaramuzza, D., 2018. A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2502–2509.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. CARLA: An Open Urban Driving Simulator. *Conference on Robot Learning*, PMLR, 1–16.

Gallego, G., Delbrück, T., Orchard, G. et al., 2020. Event-Based Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(1), 154–180.

Galperin, I., Rivest, R. L., 1993. Scapegoat trees. *SODA '93: Proceedings of the fourth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, USA, 165–174.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Giubilato, R., Stürzl, W., Wedler, A., Triebel, R., 2022. Challenges of SLAM in extremely unstructured environments: the DLR Planetary Stereo, Solid-State LiDAR, Inertial Dataset. *IEEE Robotics and Automation Letters*, 1-8.

Jeong, J., Cho, Y., Shin, Y.-S., Roh, H., Kim, A., 2019. Complex Urban Dataset with Multi-level Sensors from Highly Diverse Urban Environments. *International Journal of Robotics Research*, 38(6), 642–657.

Kim, G., Kim, A., 2018. Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 4802–4809.

Koenig, N., Howard, A., 2004. Design and use paradigms for Gazebo, an open-source multi-robot simulator. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 3, IEEE, 2149–2154vol.3.

Kong, F., Liu, X., Tang, B. et al., 2023. MARSIM: A Light-Weight Point-Realistic Simulator for LiDAR-Based UAVs. *IEEE Rob. Autom. Lett.*, 8(5), 2954–2961.

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W., 2011. G2o: A general framework for graph optimization. *2011 IEEE International Conference on Robotics and Automation*, IEEE, 3607–3613.

Lu, F., Milios, E., 1997. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4), 333–349.

Pan, Y., Xiao, P., He, Y. et al., 2021. MULLS: Versatile LiDAR SLAM via Multi-metric Linear Least Square. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 11633–11640.

Prokhorov, D., Zhukov, D., Barinova, O., Anton, K., Vorontsova, A., 2019. Measuring robustness of Visual SLAM. *2019 16th International Conference on Machine Vision Applications (MVA)*, IEEE, 1–6.

Qingqing, L., Xianjia, Y., Queralta, J. P., Westerlund, T., 2022. Multi-modal lidar dataset for benchmarking general-purpose localization and mapping algorithms. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 3837–3844.

Shah, S., Dey, D., Lovett, C., Kapoor, A., 2017. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. *Field and Service Robotics*, Springer, Cham, Switzerland, 621–635.

Shan, T., Englot, B., 2018. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 4758–4765.

Shan, T., Englot, B., Meyers, D. et al., 2021. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 5135–5142.

Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., Stachniss, C., 2023. KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robotics and Automation Letters (RA-L)*, 8(2), 1029–1036.

Wang, H., Wang, C., Chen, C.-L. et al., 2021. F-LOAM : Fast LiDAR Odometry and Mapping. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 4390–4396.

Xu, W., Cai, Y., He, D. et al., 2022a. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Trans. Rob.*, 38(4), 2053–2073.

Xu, W., Zhang, F., 2021. FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter. *IEEE Rob. Autom. Lett.*, 6(2), 3317–3324.

Xu, X., Zhang, L., Yang, J., Cao, C., Wang, W., Ran, Y., Tan, Z., Luo, M., 2022b. A Review of Multi-Sensor Fusion SLAM Systems Based on 3D LIDAR. *Remote Sens.*, 14(12), 2835.

Zhang, J., Singh, S., 2014. LOAM: lidar odometry and mapping in real-time. D. Fox, L. E. Kavraki, H. Kurniawati (eds), *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014*.