Author accepted manuscript version of the publication in
2023 11th International Conference on Control, Mechatronics and Automation (ICCMA)
Published version: https://doi.org/10.1109/ICCMA59762.2023.10374942

Multi-Modal SLAM for Accurate Localisation in Self-similar Environments

1st Peter Ørnulf Ivarsen
Smart Sensors and Microsystems
SINTEF Digital
Oslo, Norway
orcid.org/0009-0000-4576-3093

2nd Jan Sramota
Mathematics and Cybernetics
SINTEF Digital
Trondheim, Norway
orcid.org/0000-0002-3314-9026

3rd Martin Eek Gerhardsen
Smart Sensors and Microsystems
SINTEF Digital
Oslo, Norway
orcid.org/0009-0002-3864-9430

4th Henrik Lundkvist
Mathematics and Cybernetics
SINTEF Digital
Trondheim, Norway
orcid.org/0000-0002-7885-4677

5th Richard J. D. Moore
Smart Sensors and Microsystems
SINTEF Digital
Oslo, Norway
orcid.org/0000-0002-9967-0308

***Abstract*—Regular inspection and maintenance (I&M) of road tunnels is critical for ensuring safe operation and maximising the infrastructure's longevity. Today's I&M operations are time-consuming and disruptive to normal operations, but advances within robotics, automation, and digitalisation promise significant productivity gains. Accurate and reliable localisation is key to achieving this, but poses significant challenges in tunnels due to the absence of GNSS signals and the self-similar nature of the environment. This paper presents a novel approach for achieving real-time high-accuracy localisation in tunnels such that it can be used for autonomous navigation. The proposed system implements a simultaneous localisation and mapping (SLAM) solution that integrates data from scanning LiDAR, camera and inertial measurement unit (IMU). We have developed a novel approach that fuses the information from these sensors at the feature level and jointly optimises over all constraints. This enables our system to overcome the degeneracy of typical SLAM solutions in self-similar environments such as tunnels. To evaluate the performance of the proposed system, experiments and autonomous missions were conducted in real tunnels, and comparisons were made against existing localisation methods. The results demonstrate that the proposed system achieves high accuracy and exhibits good robustness in challenging tunnel conditions.**

***Index Terms*—SLAM, localisation, autonomy, inspection, tunnel, road, infrastructure**

I. Introduction

Systems for autonomous infrastructure inspection aim to save time, limit human exposure to dirty and dangerous environments, and to deliver structured and repeatable data. Key requirements for such systems are accurate associations between observations and locations as well as the ability to repeat inspections with high positional accuracy over time. These systems are therefore dependent on a robust and accurate localisation solution.

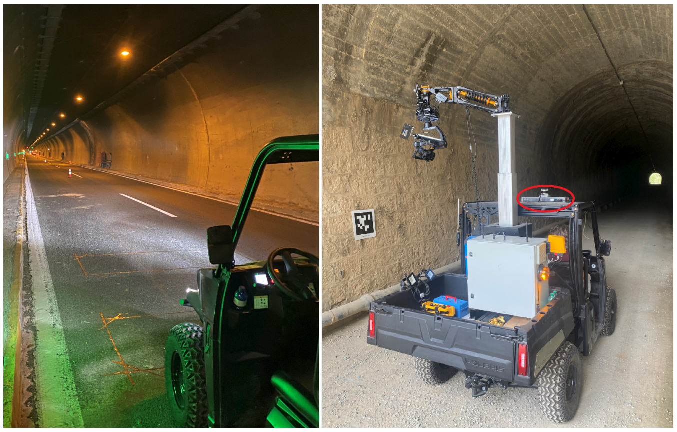

Fig. 1. Images of the autonomous cart with which we tested our localisation solution in a highway tunnel (left) and ex-train tunnel (right). Our system is mounted to the roof of the vehicle. Also visible is the crane-mounted inspection camera.

Many inspection tasks are performed in GNSS-denied environments such as indoor, underground, or in outdoor areas where large buildings or geological structures makes GNSS positioning unreliable. In these environments, simultaneous localisation and mapping (SLAM) is an often used alternative.

SLAM methods have been developed to support different sensor modalities. For platforms where weight and power consumption are constraints such as drones, camera or camera and inertial measurement unit (IMU)-based SLAM, also known as visual SLAM [3], or visual-inertial SLAM [11], are commonly used. On platforms that do not have such limitations, a standard configuration is to apply LiDAR [17], or LiDAR inertial SLAM methods [13].

In short, LiDAR SLAM is based on sampling the surrounding geometry and aligning the resulting scan with previous scans to estimate ego-motion. This is a robust and well tested approach much used in robotics. Visual SLAM, on the other

This project has received funding from the European Union's (EU) Horizon 2020 research and innovation programme within the project PILOTING under grant agreement number 871542. Described work in this article was carried solely by SINTEF Digital based in Trondheim and Oslo.

© 2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

hand, is based on matching 2D features between two or more images and from this, estimating local geometry and motion. This process is in general less constrained and more prone to accumulating pose and/or scale drift. However, an image can in many cases provide more unique information as it will pick up intensity information from a scene as well as the projection of geometries.

The main challenges of delivering well functioning localisation in the case of tunnels are connected to the particular conditions we find in such environments:

- Geometry is not locally unique, hence traditional LiDAR based SLAM solutions will not be able to asses ego-motion uniquely in any straight part of a tunnel.
- The environment contains repeated structures such as signs, lamps and road markings. This will make global place recognition vulnerable to proposing erroneous data association and loop closures.
- In trafficked tunnels, a large part of the environment can consist of dynamic objects such as vehicles or cyclists. If the SLAM system bases its estimated ego-motion on these objects instead of the static environment, errors will occur and accumulate.

In the particular case of tunnel inspection, Filip et al. [5] tested several state-of-the-art LiDAR-based SLAM solutions and found positional root mean square error (RMSE) for all methods to be several meters on a 40 m trajectory. This will for the inspection use-cases be too large an error.

Combining vision and LiDAR within a SLAM system is not a first; open source systems such as [14] and [18] use LiDAR-assisted visual odometry as an initialisation to a LiDAR SLAM method, others such as [4] fuse LiDAR- and camera-based pose estimates and geometries in a joint bundle adjustment. But for the tunnel case, both these approaches have a similar problem, as they leave the scan matcher to align an incoming scan with a previous scan, or a map, without sufficiently constraining the motion along the length of the tunnel. Even when scan matching is initialised with a pose that reflects the agent's actual motion, the matching algorithm will tend to "pull" the estimated velocity towards zero, as this creates the best overlap between scans. Hence both of these approaches expose the solution to the same vulnerabilities as a LiDAR-only solution, and the motion estimate will not be sufficiently constrained.

To handle these challenges, the proposed method utilises camera, LiDAR, IMU, and optionally wheel optometry. With the combination of these data streams we achieve the robustness of a LiDAR based system, whilst constraining the problem also along the degenerate axis of the environment by jointly optimising over structural features, image features projected into 3D, as well as an optional motion constraint from the wheel odometry.

In the Horizon 2020 EU project PILOTING [1], three pilots for autonomous infrastructure inspection are being developed. This paper describes our contribution to one of the pilots, namely an autonomous vehicle produced by Robotnik Automation S.L., aka the "cart", intended to perform inspection in road and drainage tunnels. Localisation is used for the navigation of the autonomous vehicle, for the tagging inspection data with position, and for the re-localisation of points of interest. An image of the vehicle at two of the test sites can be seen in Fig. 1.

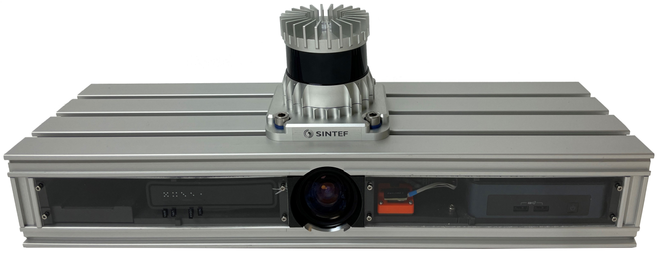

Fig. 2. Developed localisation system

II. Method

We divide the description of our approach into three sections: development, calibration, and synchronisation of the sensor suite (II-A); our novel SLAM algorithm (II-B); and our implementation onboard an autonomous robot for high-precision and robust inspection missions (II-C).

A. Hardware

The navigation payload shown on Fig. 2 consists of an Intel NUC computer with an i7-8665U CPU and three sensors – an Ouster OS0-128 LiDAR; a forward-facing FLIR BFS-U3-17S7M-C camera with Kowa LM6HC 1/1.8 lens and IR-cut filter; and a XSENS MTI-630 IMU. The navigation payload has no internal light source so the camera relies on ambient light or the light from the vehicle. Data are time synchronised with use of the precision time protocol (PTP) defined in IEEE 1588-2019 [9], which can achieve synchronisation at the sub-microsecond level. Since only the LiDAR and onboard computer supports hardware PTP, this precision is achieved only between these two sensors; and camera-IMU synchronisation is affected by transmission delays in the order of milliseconds.

Our algorithm requires that data from all three sensors must be precisely time-synchronised. Camera frame acquisition is governed by the LiDAR's trigger signal that is set by the beam at 0 degrees, hence the cameras and the LiDAR's time stamp are set identical. The IMU captures at a much higher frequency and timestamps are assigned as an offset to the LiDAR scan, based on time of arrival to the NUC.

B. SLAM solution

The SLAM solution is based on LOAM – LiDAR Odometry and Mapping in real-time [17]. This combines an odometry thread that computes the motion of the LiDAR between sweeps with a mapping thread that incrementally builds a map from the point cloud and computes the pose of the LiDAR in the map. Initially, we experimented using LIO-SAM [13], a LiDAR-inertial odometry solution to provide state estimates.

23,2024 at 11:30:13 UTC from IEEE Xplore. Restrictions apply.

This solution did at times provide a high degree of accuracy, but we also experienced that the solution lacked robustness, and at times state estimates would fail to reflect the cart's motion, or spiral into the sunset due to faulty IMU integration. As robustness is a priority, we transitioned to a LiDAR-only SLAM solution and expanded on this by adding additional constraints from a camera and optionally from the vehicle's wheel odometry to the state estimates. The core of the solution is the novel iterative scan matching jointly estimating pose over all available constraints. The LiDAR-based estimate of the cart's pose in a tunnel is well constrained in five out of six degrees of freedom, namely the rotational components and the translation in the vertical and lateral dimensions of the tunnel. However, the translation along the longitudinal axis of the tunnel will drift as the geometry of each scan along this axis will be more or less identical. To fix this degeneracy in state estimation a camera is used to add non-structural features that constrain the optimisation also along the degenerate axis. Two types of visual features are extracted:

- ORB features [12] are extracted per scan and added as additional constraints in both the scan matching performed in the odometry thread and in the mapping thread. These features are considered only locally unique and are thus only used to improve motion estimates locally. These might be extracted from lighting fixtures, road markings, signs, and markings on the tunnel wall.
- Augmented Reality University of Cordoba (ArUco) features [7] and [6] are extracted whenever they are available and are added to the scan matching in the mapping thread. As most visual features are repeated several times through a tunnel, ArUco markers are the only features considered globally unique and can thus be used to correct for drift in the motion estimate when the robot is visiting a place it has seen before i.e. when localising in an existing map. Thus the ArUco markers are mainly of utility when we are performing localisation in a previously mapped asset.

As we need the 3D coordinates of the visual features to use them as constraints in scan matching, we project the LiDAR cloud into the image, and find the depth of each visual feature from a depth based clustering of the nearest neighbours in the transformed LiDAR points. An illustration of the projected visual features can be seen in Fig. 3.

Wheel odometry has also been added as an optional input to the algorithm. Initially, this was used as a correction if a degeneracy was detected on the translation part of the pose estimate. The degeneracy detection proposed in [16] was performed by analysing the covariance matrix of the state estimate in the mapping thread, if a degeneracy was detected the estimates along the degenerate axis were replaced by those of the wheel odometry. In the current version, the wheel odometry from the cart can instead be added directly as an optional constraint in the scan matching performed in the mapping thread. The wheel odometry can be considered as a redundancy in the system and will contribute to constraining the translation along the travelling distance of the tunnel, a useful constraint when operating in heavily trafficked environments such as a road tunnel.

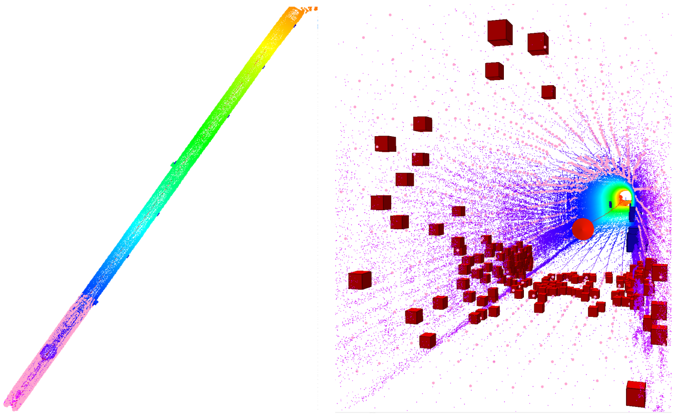

Fig. 3. Left: top down view of mapped tunnel. Right: Cart pose (red arrow), surface features (pink dots), ORB features (red cubes) and ArUco features (blue cubes) projected into the point cloud

In the following, a brief overview of the full expression to be minimised, as well as the per-constraint cost terms. For all optimisation the Ceres Solver [2] is used.

$$ {}^{0}T_t = arg\min_{T}(r_p^t + r_v^t + r_w^t); \tag{1a} $$

$$ r_p^t = \sum_{n=1}^{N} \| ({}^{0}\widetilde{T}_t p_n^s - p_n^m) \cdot n_n^m \|^2 \tag{1b} $$

$$ r_v^t = \sum_{m=1}^{M} w^v \| {}^{0}\widetilde{T}_t v_n^s - v_n^m \|^2 \tag{1c} $$

$$ r_w^t = w^w \| {}^{0}t_t - {}^{0}T_{t-1} {}^{t-1}\hat{t}_t \|^2 \tag{1d} $$

The full cost function is given by (1a), here ${}^{0}T_t$ is the current estimate of the global pose at time t and ${}^{0}\widetilde{T}_t$ is the estimate of this pose at the previous iteration of the scanmatcher. The structural feature constraint is given by the point to plane residual term (1b), where $p_i^s \in P_s$ is a structural feature from the current scan, $p_i^m \in P_m$ is the associated point in the map and n_i^m is the corresponding point normal. For structural features, correspondence search is done as in most ICP-based scan matchers, with a nearest neighbour search per iteration. For the projected visual feature points on the other hand, correspondences are found either directly by matching ArUco tag ID, or in the case of an ORB feature through an initial feature matching followed by a Random Sample Consensus (RANSAC) outlier rejection scheme, aligning the set of current 3D visual features with visual features from a fixed lag window in the map coordinate frame. The visual features residual term is given in (1c), here v_i^s is the i$^{\text{th}}$ visual feature projected into the incoming scan, v_i^m is the corresponding feature in the map frame, and w^v is a weighting term. Finally the wheel odometry residual, which only constrain the translational part of the motion estimate, ${}^{0}t_t$, is given in (1d), where the relative translation measured by wheel odometry since last scan ${}^{t-1}\hat{t}_t$ is transformed to the previous scan's frame by ${}^{0}T_{t-1}$.

09 2024 at 11:30:19 UTC from IEEE Xplore. Restrictions apply.

A graphical overview of the SLAM method can be seen in Fig. 4. The LiDAR cloud is dewarped by integrating up the IMU's gyro since the last scan and thus finding the robot's rotation during the scan period. This compensates for the changes in the robot's orientation during the acquisition of the LiDAR scan. After this, surface features are extracted from the LiDAR scan. In parallel, ArUco features and ORB features are extracted from the current image. These features are projected into 3D by sampling the LiDAR cloud for depths. Following this, scan-to-scan matching is performed in the odometry thread, matching surface features and ORB features from the previous and the current scan to estimate the relative motion between scans. Motion estimates from the odometry thread as well as surface, ORB and ArUco features are passed on to the mapping thread. Here, scan-to-map matching is performed to align the current LiDAR data to the map. After scan-to-map matching is successful, surface points and ArUco corner points are added to the map.

C. Map creation

The intended use case for the system is to make repeatable observations of faults in a tunnel over time. A way to make localisation as repeatable as possible is to create a map of an asset once, and then use this map for all successive missions. In this way all observations will be in a common reference frame and we will be able to limit the positional drift to a minimum. When creating this map, one wants it to carry sufficient information to correct for drift, contain as little noise as possible, and be as accurate as possible. These three aspects are discussed in the following.

Environments such as tunnels can be very self-similar for both LiDAR and camera sensors, thus attaching ArUco markers on the tunnel wall at approximately 20 meter intervals gives explicit landmarks prior to mapping. These will be registered by the SLAM method to the map, which will thus consist of a dense point-cloud representing the geometry of the tunnel, in addition to a set of points per observed tag with a unique identifier.

Another challenge when mapping a new site, is that dynamic objects, such as cars, will be introduced as noise in the map. When the system then reuses the map, these dynamic elements can align with current dynamic elements, which introduce errors in the estimated motion. To avoid this issue a post-processing step is applied to remove such dynamic features stored in the map. Thus for the cases where the system operates in very dynamic environments, such as a highway tunnel, the Removert algorithm [10] is used to clean the map, removing most dynamic features.

Even though we found our system to provide accurate and robust pose estimates, we found that localisation accuracy could be further improved by utilising a total station to constrain the cart's position during offline map building. This requires some setup, and can only be done offline both because the total station lack long range communications with the cart and due to that coordinate frames and clocks between the cart and the total station needs aligning. As the total station provides accurate positioning but no orientation we treat this additional constraint similar to the wheel odometry constraint 1d, but on the carts absolute position. This workflow requires a larger setup, but allow for a theoretical sub-centimetre accuracy when building a map of a new site. It is worth noting that even though the total station is a valuable constraint for SLAM, it is not suitable for deployment on long range inspections due to the line of sight requirement.

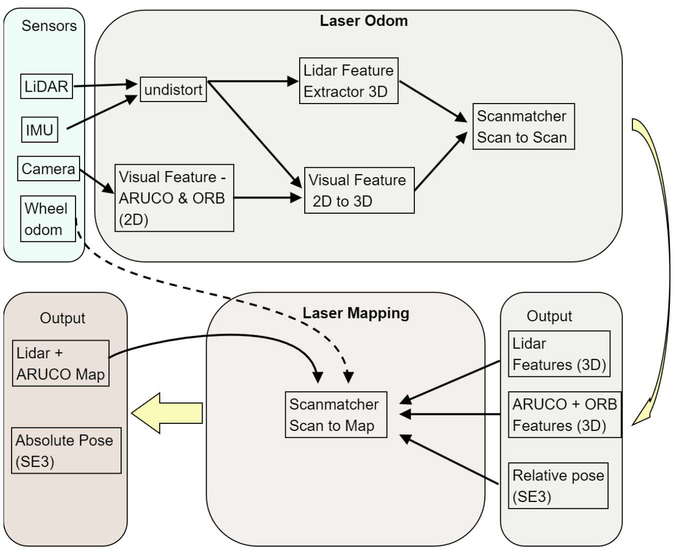

Fig. 4. An overview of the dataflow in the SLAM system

III. Data

We have recorded datasets with the system mounted on an inspection vehicle in a 175 m long former railway tunnel outside the city of Coripe, Spain. Accompanying these recordings we have position measurements, as reported by a Leica robotic total station tracking a prism mounted on the cart. The total station reports position with sub-centimetre precision at 10 Hz, and can thus be considered as a ground truth with which to evaluate the system.

We have in total three data-sets, the first set (A) is a round-trip in the tunnel at inspection speed, about 1 m/s, with only the tunnel lights used as illumination. The vehicle first drives forwards through the tunnel and then reverses back to the starting point. The second data-set (B) is recorded under the same conditions but with a speed of about 4m/s. And the third data-set (C) contains a single forward pass with a similar speed as (B) and with the vehicle's headlights switched on.

As we did not have the option to synchronise the clocks and align the frames between the total station and the cart during the field tests, this was done after the fact by cross-correlating the trajectories and then aligning frames by Umeyama's method [15].

IV. Results

We have evaluated two configurations of our proposed approach:

1) Only LiDAR, camera, and IMU data are available during map building and localisation – i.e. here we are performing SLAM in an unknown environment
2) Identical sensor setup to the first configuration, except here we are localising against a pre-built high accuracy map created by using the total station as an additional constraint as described is section II-C.

In both configurations, the position data from the total station is used to provide ground truth to evaluate the system. We have also benchmarked our approach against two existing SLAM systems, the LiDAR inertial SLAM system named LIO-SAM [13] and a LiDAR visual inertial SLAM system, LVI-SAM [14]. We have made a reasonable attempt to tune the two systems to achieve good performance, but further tuning could likely be performed to improve results.

As an evaluation framework we have used TUMs evo package for comparison of trajectory output of SLAM algorithms [8]. For the reported RMSE we have aligned the estimated trajectory with the ground truth. The reported RMSE for the three data-sets can be found in Table I.

TABLE I
FULL TRAJECTORY RMSE FOR THREE TUNNEL DATASETS

Method	**RMSE (m)**		
	A	***B***	***C***
LIO-SAM	52.8	51.9	52.5
LVI-SAM	44.5	28.8	45.2
Ours	0.605	0.314	0.400
Ours (pre-built map)	**0.086**	**0.156**	**0.112**

*Smallest error in **bold**

As a sanity check, we first evaluated our approach against the two comparison state-of-the-art methods, LIO-SAM and LVI-SAM, in a well-structured environment. Here we expected all methods to give qualitatively similar results, and this is confirmed by the trajectories shown in Figure 5.

We then tested all methods on the three tunnel datasets described in section III. Table I shows that both LIO-SAM and LVI-SAM produced trajectories with very large RMSEs for all three test datasets. Position error for these approaches evolves systematically in all three tests and a typical trace is shown in Figure 6. In fact for all our test cases, LIO-SAM and LVI-SAM underestimated the motion of the vehicle when in the tunnel, causing position error to grow in proportion to the vehicle's distance from the start point. This is due to the design of the scan matching step in these algorithms, which leads to an underestimation of observer motion in cases where the lidar-based motion estimate is poorly constrained, as discussed in section I.

Our proposed approach, on the other hand, achieves an average RMSE from all three tunnel tests of 0.44 m when running live without a pre-built map, and 0.12 m when utilising a map pre-built with the aid of a total station. Furthermore, Figure 6 shows that our approach is consistent with the ground truth at all times and clearly outperforms the comparison state-of-the-art methods. These results support our hypothesis that 'anchoring' the lidar-based scan matching step with visual feature correspondences will sufficiently constrain the problem of self-localisation even in environments that are structurally self-similar, such as tunnels. The significant improvement in accuracy of our proposed approach over other LiDAR-visual-inertial approaches such as LVI-SAM, demonstrates the importance of jointly optimising lidar and visual feature correspondences rather than simply initialising lidar-based scan matching with a pose estimate.

In addition to the above tests we have validated the accuracy, robustness, and real-time capability of our approach by integrating our localisation module with the target inspection vehicle (Figure 1) and performing a series of fully autonomous inspection missions at several tunnel sites in Spain and Greece. During these missions, our localisation module provided real-time (10 Hz) pose data, which was used on board the vehicle for path planning and control.

V. CONCLUSIONS

In this paper we have presented a sensor suite and SLAM solution for localising within self similar environments such as tunnels. The solution is derived from LOAM, but as observed for multiple LiDAR-based SLAM solutions the tunnel environment does not have enough structural features to accurately estimate the ego-motion. We have therefore built a sensor payload with additional sensors data from camera and IMU to address this limitation. Experimental evaluation in a tunnel shows that this results in good accuracy, which is important for the use cases such as tunnel inspection and maintenance. The algorithms will be further validated in additional tunnels with traffic to ensure that the dynamic objects are properly filtered.

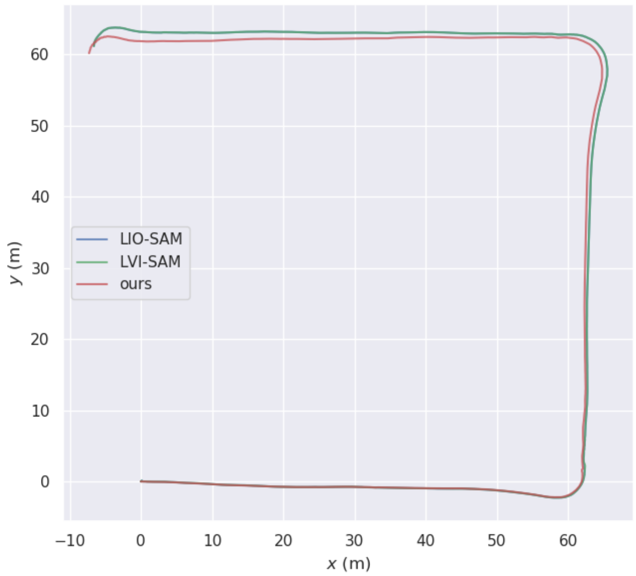

Fig. 5. Comparison of estimated trajectories from a dataset recorded in an urban environment containing lots of structural features. No ground truth available, but from visual inspection all methods produce reasonable results

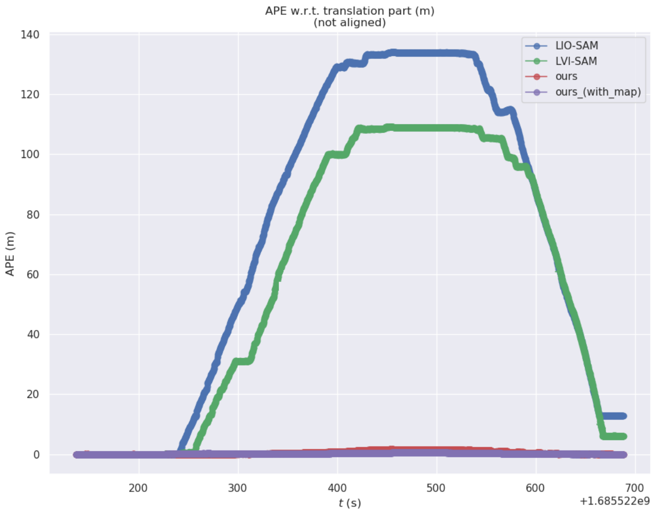

Fig. 6. Comparison of the absolute position error of the estimated trajectories based on dataset A – a pass back and forth through the 175m long tunnel at inspection speed.

Acknowledgement

Verification of the developed system in the tunnel environment was possible thanks to contribution of several partners in the PILOTING consortium. We are grateful for the cooperation and good work performed on this project by our partners.

References

[1] EU Horizon 2020. Pilots for robotic inspection and maintenance grounded on advanced intelligent platforms and prototype applications.

[2] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 3 2022.

[3] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, December 2021. arXiv:2007.11898.

[4] Chih-Chung Chou and Cheng-Fu Chou. Efficient and Accurate Tightly-Coupled Visual-Lidar SLAM. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14509–14523, September 2022. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

[5] Iulian Filip, Juhyun Pyo, Meungsuk Lee, and Hangil Joe. Lidar SLAM Comparison in a Featureless Tunnel Environment. In *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, pages 1648–1653, November 2022. ISSN: 2642-3901.

[6] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and R. Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recognition*, 51:481–491, 3 2016.

[7] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[8] Michael Grupp. evo: Python package for the evaluation of odometry and slam., 2017.

[9] IEEE. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008)*, pages 1–499, 2020.

[10] Giseop Kim and Ayoung Kim. Remove, then revert: Static point cloud map construction using multiresolution range images. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10758–10765, 2020.

[11] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, August 2018. Conference Name: IEEE Transactions on Robotics.

[12] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571, November 2011. ISSN: 2380-7504.

[13] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. *arXiv:2007.00258 [cs]*, July 2020. arXiv: 2007.00258.

[14] Tixiao Shan, Brendan Englot, Carlo Ratti, and Daniela Rus. LVI-SAM: Tightly-coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping. *arXiv:2104.10831 [cs]*, April 2021. arXiv: 2104.10831.

[15] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, April 1991.

[16] Ji Zhang, Michael Kaess, and Sanjiv Singh. On degeneracy of optimization-based state estimation problems. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 809–816, Stockholm, Sweden, May 2016. IEEE.

[17] Ji Zhang and Sanjiv Singh. LOAM: Lidar Odometry and Mapping in Real-time. In *Robotics: Science and Systems X*. Robotics: Science and Systems Foundation, July 2014.

[18] Ji Zhang and Sanjiv Singh. Visual-lidar odometry and mapping: low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181, Seattle, WA, USA, May 2015. IEEE.

2024 at 11:30:13 UTC from IEEE Xplore. Restrictions apply.