Indoor Localization and Mapping: Towards Tracking Resilience Through a Multi-SLAM Approach — Source link ⧉

Pierre Alliez, Fabien Bonardi, Samia Bouchafa, Jean-Yves Didier ...+6 more authors

Institutions: French Institute for Research in Computer Science and Automation, Université Paris-Saclay

Related papers:

- A Comparative Analysis of LiDAR SLAM-Based Indoor Navigation for Autonomous Vehicles

- Hectorslam 2d Mapping for Simultaneous Localization and Mapping (slam)

- Three main paradigms of simultaneous localization and mapping (SLAM) problem.

- A Methodology for Principled Approximation in Visual SLAM

- Simultaneous Localization and Mapping Based on $\mu+1$-Evolution Strategy for Mobile Robots

Indoor Localization and Mapping: Towards Tracking Resilience Through a Multi-SLAM Approach

Pierre Alliez, Fabien Bonardi, Samia Bouchafa, Jean-Yves Didier, Hicham Hadj-Abdelkader, Fernando Israel Ireta Muñoz, Viachaslau Kachurka, Bastien Rault, Maxime Robin, David Roussel

HAL Id: hal-02611679
https://hal.inria.fr/hal-02611679

Submitted on 18 May 2020

Indoor Localization and Mapping: Towards Tracking Resilience Through a Multi-SLAM Approach

Pierre Alliez[1], Fabien Bonardi[2], Samia Bouchafa[2], Jean-Yves Didier[2], Hicham Hadj-Abdelkader[2],
Fernando Ireta Muñoz[1], Viachaslau Kachurka[2], Bastien Rault[3], Maxime Robin[3], David Roussel[2,*]

Abstract— **This paper presents a use case for SLAM techniques applied to real time localization and detailed mapping for emergency response personnel in non cooperative environments. Such environments tend to defeat conventional localization approaches, therefore we must ensure continuous operation of our localization and mapping regardless of the difficulties encountered (lack of GPS signals, lighting conditions, smoke, etc.). The proposed system fuses two SLAM algorithms, a LiDAR-based and a camera-based. Since LiDAR-based SLAM uses dense 3D measurements, it is well suited to the construction of a detailed map, while the visual SLAM allows to quickly recognize already visited places in order to apply loop closure corrections, by using a key frames graph. The currently proposed system allows collaboration between these two SLAMs through pose sharing and relocalization.**

I. INTRODUCTION

The problem of accurately localizing emergency response agents (civil security, firefighters, etc.), law enforcement or armed forces agents in a closed, unknown, non-cooperative environment remains an open problem nowadays, as well as mapping this environment. Many military and civil applications would benefit from such localization systems [1].

Localization and mapping tasks focus on the idea that headquarters (HQ) should have the most accurate location of their agent under every condition in real-time, while also receiving information about the environment (e.g. reckon missions in armed forces, or operative information on a fire) – thus formulating an extension to the indoor positioning problem [2] and a use case for Simultaneous Localization and Mapping (SLAM) techniques [3].

The main difficulty for SLAM techniques is the lack of suitable technologies that can take into account the technical limits (the equipment should be quite small and efficient), technological requirements (diversity of sensors to make the system more robust to ensure the mission) and environmental constraints (lack of satellite connection, hazardous or non-cooperative environment) [1]. The multi-sensor solutions have been studied in the field of mobile robotics for several decades and usually consider robotic platforms, such as wheeled vehicles, moving in homogeneous conditions [4] or,

more recently, flying drones [5]. In most cases, multi-sensor data fusion is required with special interest in combining Inertial Measurement Unit (IMU) with other sensors such as cameras [6] or LiDAR [7].

Among the aforementioned constraints and requirements, various challenging bottlenecks can be identified, such as:

- *Real-time tracking and mapping* – the current localization should be available (and possibly transmitted to control center) in real-time, whereas complete localization trajectory and reconstructed map can be retrieved and processed later.
- *Transitions between indoor and outdoor environments* with potentially very different sizes, therefore imposing difficulties on mapping and data management.
- *Tracking failures* due to erroneous or missing data, human-specific motion, environment factors, etc.
- *Environment-specific challenges*, which lead to sensor failures, such as rapid change of light conditions, smoke, etc.

This paper outlines the approaches we used to find a general compromise between real-time efficiency and precision performance, as well as addressing the maximum of the imposed challenges. It is organized as follows: the next section describes the general architecture of the proposed multi-sensor multi-SLAM system, as well as some technical details on the modifications we have done to the state-of-art algorithms used in it for both, localization and cartography. Section 3 shows the results and validation for these aforementioned modifications. Then we present our conclusion and some perspectives for future works.

II. TOWARDS TRACKING AND MAPPING RESILIENCE

The usual approach in the field of such localization task is to use high-sensitive GPS receivers coupled with IMUs [1]. However, GPS signals are not always available and complementary sensors are added to provide the tracking algorithm.

While during decades the imaging sensors appeared to be among the least appealing in the field of robust real-time indoor positioning due to high computational cost and prone to failures in non-cooperative environment [1] (e.g., a basic imaging sensor in total darkness cannot provide any usable data), recent convergence of visual SLAM field (as observed in [8]) enabled robuster approaches and reopened this niche.

More recently, LiDAR-based approaches for real time localization and mapping have also emerged [9] which can

[1] Université Côte d'Azur and INRIA, 06902 INRIA Sophia Antipolis Méditerrannée, France {`fernando.ireta-munoz, pierre.alliez`}`@inria.fr`
[2] IBISC, Univ Evry, Université Paris-Saclay, 91025, Evry, France {`viachaslau.kachurka, david.roussel, hicham.hadjabdelkader, fabien.bonardi, jeanyves.didier, samia.bouchafa`}`@univ-evry.fr`
[3] INNODURA TB, 69603, Villeurbanne, France {`bastien.rault, maxime.robin`}`@innodura.fr`
* All authors are listed in alphabetical order.

provide real time localization along with dense maps suitable for later use in cartography.
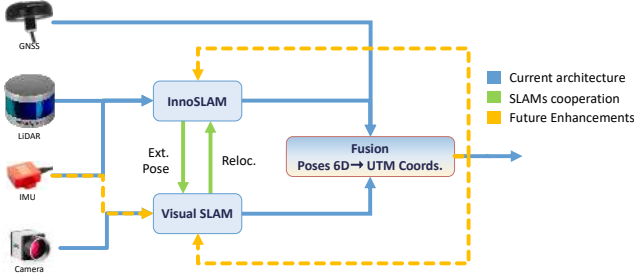

Fig. 1. System architecture and fusion scheme of the proposed localization system. LiDAR-based SLAM and visual SLAM can communicate pose optimization events, such as loop closures and relocalizations.

Fig. 1 presents the architecture of the proposed approach combining a LiDAR-based SLAM with a visual SLAM along with the fusion scheme explained in section II-C. To minimize the sensor failure factor, we chose to use a set of complementary sensors – LiDAR and an infrared camera in their respective SLAMs, while IMU and GPS receiver mainly contribute in a post SLAMs fusion process. Infrared sensors (especially SWIR: 900 – 1700 nm) work better in foggy or smoky environments. LiDAR sensors are not as affected by lighting conditions as cameras, but the loop closing techniques are not as efficient in LiDAR-based SLAMs compared to visual SLAMs; these approaches might be considered as complementary. Thus, LiDAR and camera data are processed by independent SLAMs and fused afterwards.

A. InnoSLAM: LiDAR-based SLAM

The aim is to estimate a motion between two LiDAR measurements while the sensor moves. A widely used algorithm is the Iterative Closest Point (ICP) and its variants [10]. They lead to a precise odometry with a small error but at a high computational cost. However, the small errors can be accumulated in time, leading to significant drifts. Furthermore, computing an ICP on a complete scan of a LiDAR such as a Velodyne VLP-16 (16×900 datapoints) cannot be performed at high framerate. Thus, our InnoSLAM is based on the LOAM algorithm [9], a state-of-the-art real-time and low drift SLAM. Finally, IMU data is used to ensure InnoSLAM vertical orientation and propagated to the following fusion process (see section II-C).

1) The LOAM algorithm: it consists of three sequential steps: features detection, ego-motion and mapping. The first step extracts features from the kth LiDAR scan $\mathcal{S}_k^L$ in its local coordinate system (L) according to the local curvature $\mathcal{C}_{k,i}$, where i is the point's index in the scan. This parameter is obtained by computing a standard PCA on each point $\mathcal{S}_{k,i}^L$ and its neighborhood through eigenvalues and eigenvectors. Depending on the value of $\mathcal{C}_{k,i}$, the point will be categorized as an edge feature (high enough value), as a planar feature (low enough value) or discarded. This extraction step
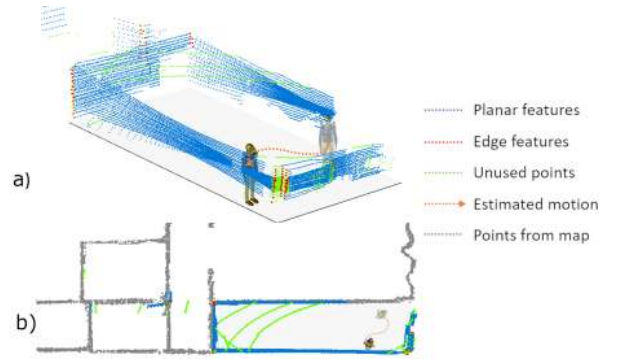

Fig. 2. Estimated motion: a) based on features from two consecutive frames; b). refined using features from maps

provides two pointclouds: $\mathcal{E}_k^L$ for edge features and $\mathcal{P}_k^L$ for planar features, as shown in Fig. 2.

These point clouds are then matched to geometric features derived from $\mathcal{E}_{k-1}^L$ and $\mathcal{P}_{k-1}^L$ from the previous scan $\mathcal{S}_{k-1}^L$ using multiple ICP algorithms (point-to-point for the edge features and point-to-line for the planar features). We then use a Levenberg-Marquardt error minimization to estimate the translation and rotation. This ego-motion step estimates pose transformation between two consecutive scans $\mathcal{S}_{k-1}^L$ and $\mathcal{S}_k^L$, as illustrated in Fig. 2.a, and by odometry, in world frame W coinciding with L at the initial position.

Usually, the accumulation of errors (drift) is rectified by loop closures as in visual SLAM approaches. In the case of LiDAR-based SLAM, it is challenging because of the sparsity of relevant geometric descriptors. In LOAM, drifts are minimized by the mapping step, which consists in finding correspondences between geometric features extracted from the scan ($\mathcal{E}_k^W$ and $\mathcal{P}_k^W$) and their respective feature maps ($(\mathcal{M}_e)_k$ and $(\mathcal{M}_p)_k$). Correspondences and world-frame position are found using point-to-model ICP algorithm followed by a Levenberg-Marquardt minimization again. The refined position is then used for updating both feature maps, using also a voxel grid filter [11] to reduce their density for real-time requirements. This principle illustrated in Fig. 2.b, minimizes drifts by the use of maps, comparing geometric features to all previous ones since the beginning of the sequence. In our experiments, results obtained on KITTI Benchmark [12] show that this method is effective for outdoor environments.

2) Modifications to LOAM introduced by InnoSLAM: We aim at a precise and real-time localization under very different conditions: outdoor environments, buildings with stairs, small rooms, etc. Furthermore, our system needs to keep running for long periods of time (at least 30 minutes) which could result in large maps. The LOAM algorithm has several limitations regarding the requirements of our system.

First, the map density determined by the voxel grid filter is constant regardless of the type of environment: indoor or outdoor, as these environments have potentially very different sizes. This can lead to either a map size exponential growth or a precision loss depending on the kind of environment targeted by the voxel grid filter settings.

Second, as maps become larger over time this can lead to higher computational times when comparing the current scans ($\mathcal{E}_k^W$ and $\mathcal{P}_k^W$) with the maps (($\mathcal{M}_e)_k$ and ($\mathcal{M}_p)_k$).

In order to tune the voxel grid filter parameters to the environment, we build our features maps as three-dimensional array of one cubic meter pointclouds (using hash map to reduce time computation and memory usage). Then, during the features extraction step, we compute an unoriented bounding box on both $\mathcal{E}_k^L$ and $\mathcal{P}_k^L$. According to the bounding box dimensions $\mathcal{B}_e(x, y, z)$ and $\mathcal{B}_p(x, y, z)$, we adapt the distribution of extracted features and the density used in map updates, denoted $\mathcal{D}_e$ and $\mathcal{D}_p$.

During the mapping step, not all features $\{\mathcal{M}_e, \mathcal{M}_p\}$ are actually needed, only the local ones in which $\mathcal{E}_k^W$ and $\mathcal{P}_k^W$ are included. Therefore, in tight environments, all extracted features are close to each other so we increase $\mathcal{D}_e$ and $\mathcal{D}_p$ to maintain an accurate pose estimation. Whereas, in wide environments (where GPS is often available), extracted features cover a large area so we lower the density to maintain a low computation time while preserving enough precision. The relation between local density and bounding box dimensions has been set empirically as followed: $\mathcal{D}_{[e|p]}(x, y, z) = \lfloor 14 - 1.7 * \sqrt{\mathcal{B}_{[e|p]}(x, y, z) - 2} \rfloor$.

At this point, we have proposed a solution to outdoor vs indoor accuracy, but feature maps grow indefinitely while exploring environments, which may cause memory usage issues. This is why we keep constant map sizes and make them roll. If the sensor position reaches a boundary of the built maps, we flush the other boundary and re-center maps on the sensor position. Flushed feature points can be saved in files for cartography purposes. This method keeps the memory usage safe, but may miss loop-closures since it flushes feature points in areas where we may go back later.

This way, we get a real-time SLAM which improves its precision in tight environments by increasing features density, while keeping enough density in wide environments to perform an accurate localization. Nevertheless ego-motion step and mapping step can fail, especially in stairs where altitude changes may be difficult to detect. Only a loop closure can trigger the rectification of the localization. Since InnoSLAM cannot detect a loop closure yet, we rely on the visual SLAM for this detection.

B. Visual SLAM

The choice of a visual SLAM in the frame of this project, as well as our modifications to it, have been discussed in our previous work [13], therefore here we will only focus on several key points: a visual SLAM algorithm, used in the context of human agent tracking in non-cooperative environment, should be a sparse feature-based indirect visual SLAM. Sparsity is needed to enhance real-time processing in the context of moderate resources; feature usage can provide a reliable recognition of previously visited locations using popular state-of-art bag-of-words (BoW) approaches. In order to build our system with a minimal set of sensors as well as respecting limited resources on the computer we opted for a monocular SLAM which in turn comes with

a scale undetermination problem (Each SLAM run yields a new map and trajectory scale). It can be solved either by using stereo or 3D camera or by combining with IMU measurements. In our case, it is solved by the cooperation between visual and LiDAR SLAMs.

ORB-SLAM [14] fulfills all aforementioned requirements. Moreover, ORB keypoints and descriptors are still robust in infrared bands as mentioned by Johansson et al. [15].

Our previously presented modification to the ORB-SLAM algorithm consists of "re-initialization" – after visual tracking is lost (when crossing a door for instance), a new tracking should be initialized automatically based on previously measured motion while preserving the already acquired trajectories and maps and hence improving the system's tolerance to failures. Improving on this idea, we can now establish a collaboration between the two SLAMs where the LiDAR SLAM's pose can be used during the visual SLAM initialization and re-initializations in order to ensure common location and scale at these specific times (see Fig. 1).

At the same time, since ORB-SLAM can quickly recognize previously visited locations, it can then determine an accurate differential pose $\Delta p_{i/j}^W$ between matched frames $\mathcal{F}_i^W$ and $\mathcal{F}_j^W$ during loop closure. This differential pose, as well as the indications of matched frames (i and j frame indices), are transferred to InnoSLAM in order to optimize its internal map and prune redundant edge and planar features.

C. Fusion

1) Fusion approaches: Data fusion approaches aim at combining redundant but flawed information by taking into account each sensor drawbacks and its model with according noises and biases. Most common robotic systems make use of odometry and absolute anchors position. Odometry can be measured on wheels or computer vision for example, considered as proprioceptive. On the other hand, GPS, radio beacons or computer vision with known visual features of the environment can be used as exteroceptive approaches. Our system is concerned by the fusion between SLAMs (visual and LiDAR-based), IMU data and GPS. Works from the state of the art put forward two classes of visio-inertial fusion: tightly-coupled and loosely-coupled fusion. In contrast to tightly-coupled approaches using visual features as inputs in parallel with IMU data, loosely-coupled architectures have a dedicated processing component which reconstruct attitude and path from the visual data only. These estimations of the position and orientation of the system are then fused with the IMU data without considering the visual features, hence the name "loosely-coupled" [16].

After several evaluations of visuo-inertial SLAM approaches, the results lead us towards a loosely-coupled method in order to fuse IMU and SLAM data. We chose a loosely-coupled strategy too for the 6D poses to UTM coordinates (See Fig. 1) because it allows us to isolate each SLAM (visual or LiDAR-based) from errors introduced by a faulty fused state: both visual and LiDAR SLAMs could drift in their own coordinates systems.

2) SLAM-IMU fusion: The algorithm used for the LiDAR-based SLAM and IMU data fusion is an Error-State Kalman Filter (*ESKF*). This Kalman filter flavour tackles non-linear systems by considering that the *True* signal state is a linear combination of a non-linear component, the *nominal state*, and an *Error state* component. The Error State evolution is supposed linear iteration step-wise and can be estimated by a usual Kalman process. Simultaneously, the Nominal State change is predicted according to its non-linear model without uncertainty. The correction stage requires then an additional step compared to a vanilla Kalman filter: the Error State is corrected, added to the Nominal State and reset to zero.

We envision to apply the same strategy to the visual SLAM and IMU data with according parameters fine tuning in future works.

3) Relative attitude to GPS fusion: The selected approach for SLAMs-GPS fusion is simpler: it involves a Kalman filter which estimation state includes attitude of the system in UTM coordinates and a relative transform of both SLAMs coordinates to UTM coordinates. This way, the prediction of the position can be processed during prolonged GPS-denied journeys (see Fig. 3).
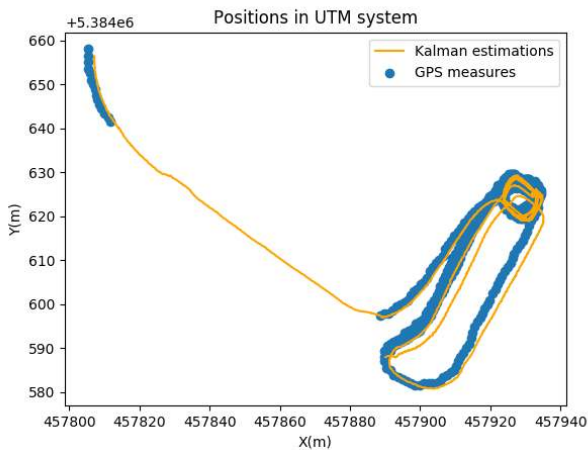
Fig. 3.   State estimation in GPS-denied area

D. Cartography

The 3D reconstruction framework proposed in this paper is inspired by a previous work [17], which provides different levels of details (LOD) for urban reconstruction, where a classification algorithm [18] is employed for detecting vegetation and permanent structures such as walls, ceilings and floors. For purposes of this paper and due to the nature of the employed datasets in our experiments (with indoor and outdoor information), a common LOD indoor reconstruction is performed as follows:

1) LOD0 – Floorplan: In LOD0, a semantic classification algorithm [18] groups all 3D points that share similar geometric information. All the points in the "wall" class are projected onto the main plane that better fits each floor (in case of buildings with several floors). A line detection algorithm
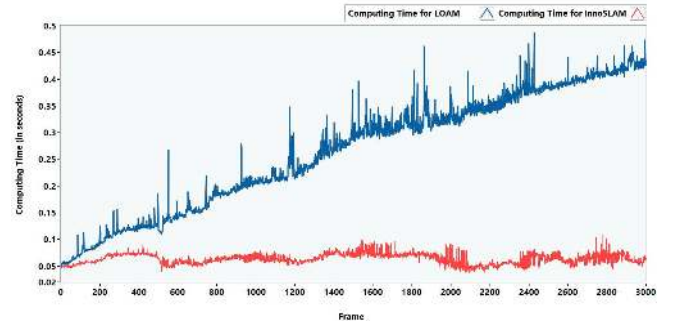
Fig. 4.   LOAM (blue) and InnoSLAM (red) computing times comparison.

based on region growing [19] is employed for estimating the segments that define the walls. These segments are then converted into a planar partition via a recent 2D kinetic geometric approach [20]. The rooms are then labelled for each generated floorplan. In case of buildings with multiple floors, each floor is detected by using the distribution peaks along the Z axis (height) of the pointcloud.

2) LOD1 – Permanent structures: For LOD1, a 3D extension of the kinetic approach [20] has been used. Associated points are projected onto the main detected planes. Planes are detected by using the Efficient RANSAC (RANdom SAmple Consensus) for shape detection presented in [21]. The closed form surface is generated by finding and optimizing candidate planes over the intersections between them. Each candidate plane is extended within a virtual box that encloses the pointcloud. Selected candidate planes are kept for representing permanent structures as walls, ceilings and floors.

3) LOD2 – Detailed reconstruction: In LOD2, we refine the 3D surface by reconstructing the map with non-rigid objects (furniture, doors, e.g.) within the pointcloud, we have employed a Poisson reconstruction for this purpose.

In case of indoor/outdoor mapping, each building has been manually isolated in order to perform its surface reconstruction. This strategy reduces the computational cost when large scenes are registered.

III. RESULTS

A. SLAMs performances

While the LOAM algorithm shows already good performance, our main focus for its modification was to reduce the cost to have the steady real-time processing. As LOAM builds constantly growing maps with a constant density, the computing time increases in linear fashion. Whereas InnoSLAM adapts the density and size of its processed maps, so the computing time stays stable as illustrated in Fig. 4.

Fig 5 (left) shows map points and trajectory obtained with InnoSLAM on a typical run featuring indoor and outdoor environments, doorway crossings and loops, highlighting the high density map along with a short-term low drift trajectory obtained in quasi-constant computing time. Fig. 5 (right) shows the map and trajectories of the visual SLAM on the same data where each door crossing led to a loss of tracking
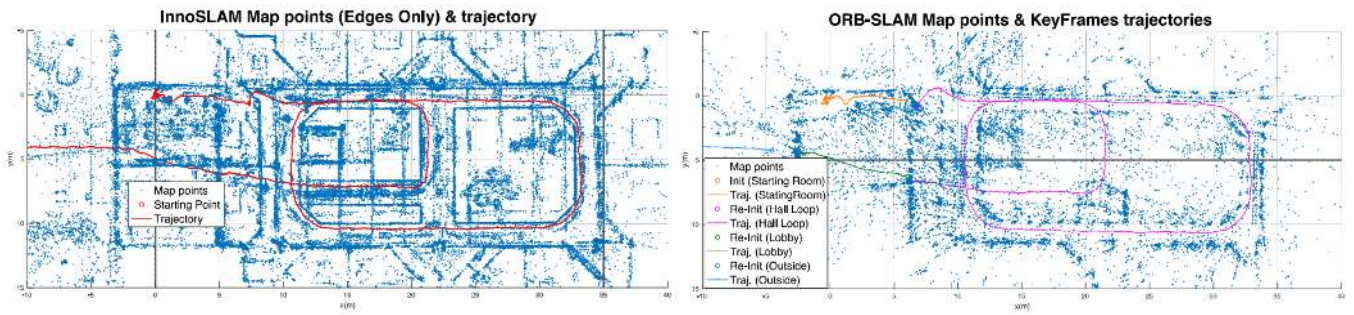
Fig. 5. Comparison of InnoSLAM (left) and visual SLAM (right) trajectories and map points
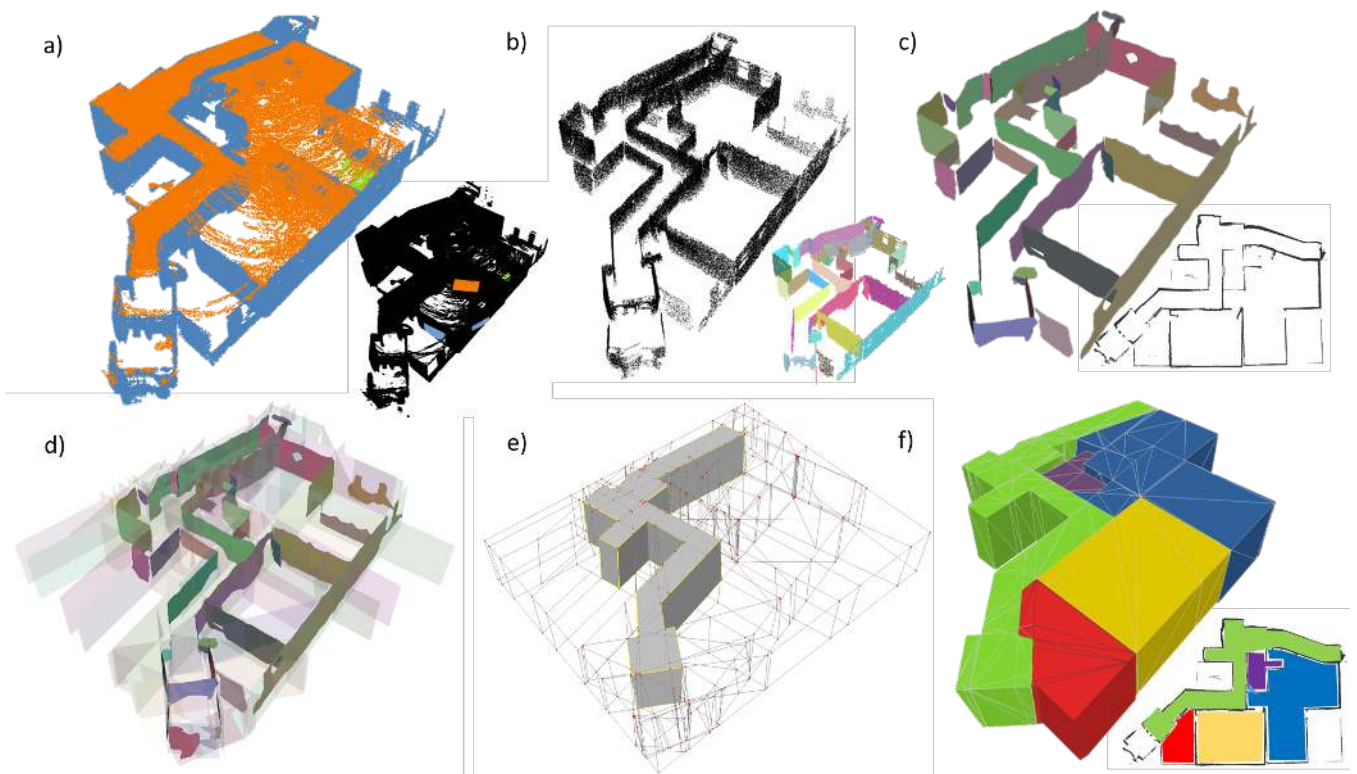

Fig. 6. Indoor 3D reconstruction pipeline: from point acquisitions to room labelization.

(a) Trajectory and LOD0
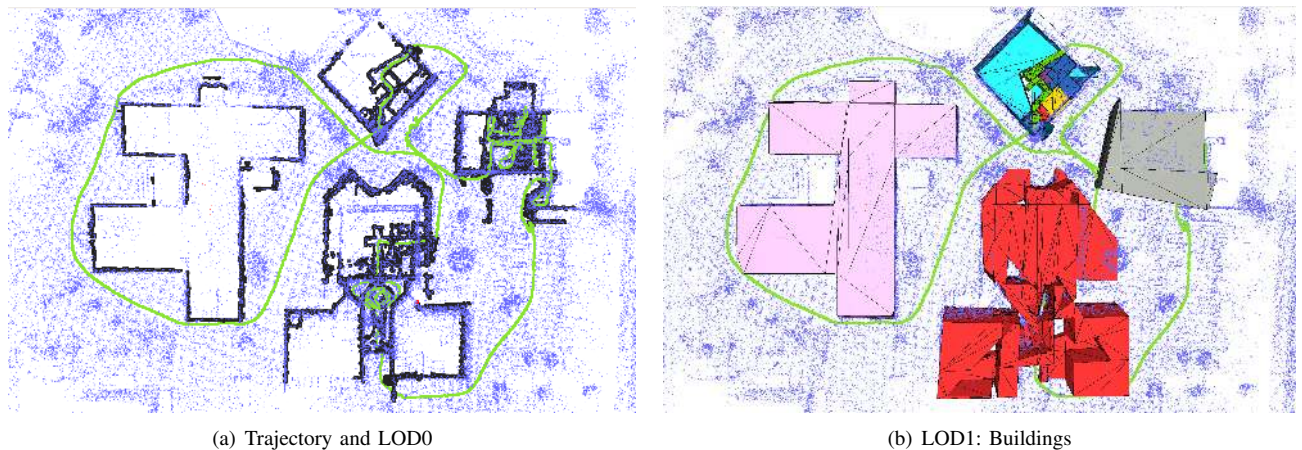
(b) LOD1: Buildings

Fig. 7. Indoor 3D reconstruction and indoor/outdoor localization example. Two LOD are provided along the estimated trajectory (in green) and the aligned pointcloud (in blue): a) Floorplan for the first stage of each building is shown in black. b) Extrusion of detected buildings by using the algorithm [20] For the building in blue, the indoor information is the LOD1 presented in Fig. 6.

then to a re-initialization using LiDAR SLAM's pose thus ensuring a consistent and common scale factor. Moreover, "Hall" trajectory features a loop closure allowing to correct the map and trajectory along the loop while being transmitted to the LiDAR SLAM.

B. Cartography

The pointcloud obtained by the localization system is processed offline for its 3D reconstruction. Aforementioned, a classification for labelling permanent structures, vegetation and ground (outdoor) is employed for isolating each building. For each building, a stage detector is employed for detecting the points that belongs to floor and ceiling at each level. Referring to the LOD stages shown in Fig. 6: a) For each stage, a subset of points that belongs to walls (blue), ceiling (orange) and floor (green) are selected for training the classification method [18]. b) For wall points, the shape detection algorithm [19] with connectivity factor of 0.1 and number of neighbors = 150 detect all planes. c) For each detected plane, all points are projected onto their main plane. These planes are projected onto the floor for obtaining the first LOD. d) Detected planes are extended within a virtual bounding box that encloses them. e) Intersections are found and the candidate facets are optimized over their oriented normals (algorithm [20]). f) LOD1 is completed by performing the previous stage for each room. A labellisation of the room is obtained by detecting the inside-out labels (using the orientation of the normals) of each optimization.

IV. CONCLUSION AND FUTURE WORKS

In this work, we have presented an overview of our localization and mapping system, based on dual LiDAR and camera SLAMs, allowing to provide real-time localization while saving detailed map points sets for later use in cartography. The different natures of the two SLAMs that we used, one dense and active and the other sparse and passive as well as the collaboration of these SLAM through the sharing of poses and loop closing events allows us to increase the resilience of the overall tracking and mapping system in difficult environments.

In the cartography component, LOD0 and LOD2 can provide a first good sight of the environment. But in LOD2 Poisson reconstruction is less accurate due to noise. As a future work, we will explore the estimation of normals for each 3D point. This will be helpful for inside/outside detection for buildings, rooms and stages.

An obvious addition to the visual SLAM is the introduction of IMU measurements in the odometry process in order to a) ensure a constant scale factor and b) improve the visual tracking robustness against high acceleration movements which can occur in human motion. And finally, a complete fusion between the SLAM poses can enhance the precision of the estimated state and compensate their respective drifts over time.

ACKNOWLEDGMENT

REFERENCES

[1] J. Rantakokko, J. Rydell, P. Strömbäck, P. Händel, J. Callmer, D. Törnqvist, F. Gustafsson, M. Jobs, and M. Grudén, "Accurate and reliable soldier and first responder indoor positioning: multisensor systems and cooperative localization," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 10–18, Apr. 2011.

[2] R. Mautz, "Overview of current indoor positioning systems," *Geodezija ir kartografija*, vol. 35, no. 1, pp. 18–22, 2009.

[3] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 55–81, Sept. 2015.

[4] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1. IEEE, July 2004, pp. I–I.

[5] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, June 2014, pp. 15–22.

[6] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and vision for absolute scale estimation in monocular SLAM," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 287–299, Jan. 2011.

[7] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2010, pp. 4372–4378.

[8] M. R. U. Saputra, A. Markham, and N. Trigoni, "Visual SLAM and structure from motion in dynamic environments: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, p. 37, June 2018.

[9] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Proceedings of the Robotics: Science and Systems*, 07 2014.

[10] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing icp variants on real-world data sets," *Autonomous Robots*, no. 1, Apr. 2013.

[11] R. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *IEEE International Conference on Robotics and Automation 2011 (ICRA 2011)*, 05 2011.

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[13] V. Kachurka, D. Roussel, H. Hadj-Abdelkader, F. Bonardi, J.-Y. Didier, and S. Bouchafa, "Swir camera-based localization and mapping in challenging environments," in *International Conference on Image Analysis and Processing*. Springer, 2019, pp. 446–456.

[14] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[15] J. Johansson, M. Solli, and A. Maki, "An evaluation of local feature detectors and descriptors for infrared images," in *European Conference on Computer Vision*. Springer, Oct. 2016, pp. 711–723.

[16] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial slam using nonlinear optimization," *Proceedings of Robotis Science and Systems (RSS) 2013*, 2013.

[17] Y. Verdie, F. Lafarge, and P. Alliez, "LOD generation for urban scenes," *ACM Trans. Graph.*, vol. 34, no. 3, May 2015. [Online]. Available: https://doi.org/10.1145/2732527

[18] F. Lafarge and C. Mallet, "Creating large-scale city models from 3d-point clouds: A robust approach with hybrid representation," *International Journal of Computer Vision*, vol. 99, 08 2012.

[19] S. Oesau, Y. Verdie, C. Jamin, P. Alliez, F. Lafarge, S. Giraudot, T. Hoang, and D. Anisimov, "Shape detection," in *CGAL User and Reference Manual*, 5th ed. CGAL Editorial Board, 2019. [Online]. Available: https://doc.cgal.org/5.0/Manual/packages.html#PkgShapeDetection

[20] J.-P. Bauchet, "Kinetic data structures for the geometric modeling of urban environments," Ph.D. Thesis, Université Côte d'Azur, Inria, France, Dec. 2019. [Online]. Available: https://hal.inria.fr/tel-02432386

[21] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Comput. Graph. Forum*, vol. 26, pp. 214–226, 06 2007.