Online Localisation and Colored Mesh Reconstruction Architecture for 3D Visual Feedback in Robotic Exploration Missions

Quentin Serdel, Christophe Grand, Julien Marzat and Julien Moras

Abstract— **This paper introduces an Online Localisation and Colored Mesh Reconstruction (OLCMR) ROS perception architecture for ground exploration robots aiming to perform robust Simultaneous Localisation And Mapping (SLAM) in challenging unknown environments and provide an associated colored 3D mesh representation in real time. It is intended to be used by a remote human operator to easily visualise the mapped environment during or after the mission or as a development base for further researches in the field of exploration robotics. The architecture is mainly composed of carefully-selected open-source ROS implementations of a LiDAR-based SLAM algorithm alongside a colored surface reconstruction procedure using a point cloud and RGB camera images projected into the 3D space. The overall performances are evaluated on the *Newer College* handheld LiDAR-Vision reference dataset and on two experimental trajectories gathered on board of representative wheeled robots in respectively urban and countryside outdoor environments.**

Index Terms— **Field Robots, Mapping, SLAM, Colored Surface Reconstruction**

I. Introduction

Embedded architectures for autonomous robots performing exploration missions are continuously improving, such that localisation and map construction in a previously uncharted environment becomes possible with limited human intervention [1]. Many SLAM-based localisation methods have been proposed and extensively evaluated on reference datasets. They rely either on monocular cameras or stereovision [2]–[4], or on 2D or 3D LiDAR scanners [5]–[8], with some recent attempts to combine both types of sensors [9]. Several computationally-efficient online mapping and 3D reconstruction approaches have been proposed in parallel [10], while the specific issue of the colourisation of mesh or point cloud representations has been investigated independently [11], [12]. However, there remains the need to further evaluate in realistic conditions the behaviour and performances of full systems which are able to combine online localisation, mapping and mesh colourisation on real ground robots equipped with 3D LiDAR and vision sensors [13].

Early work on the subject of real-time 3D mesh reconstruction in [14] introduced real-time 3D surface mesh reconstruction in an urban environment using stereo camera images and pose estimation obtained by fusing GNSS measurements and visual odometry. The method proposed in [15] performs real-time 3D mapping of house-sized indoor environments through the application of Truncated Signed Distance Function [16] and dynamic voxel hashing [17], with Visual-Inertial Odometry (VIO) as localisation source. While 3D mesh reconstruction was mainly intended to be used as a visualisation tool for human operators, later work has focused on the aspect of surfacic mesh mapping for navigation purposes, arguing that surface-based maps contain more dense information compared to sparse point clouds. This makes them more exploitable for autonomous driving or robot navigation (either autonomous or remotely-operated). Limitation of memory usage and computational cost along with scalability of the 3D mesh reconstruction solution are major concerns when addressing the deployment of such systems on the field. In [18], a method has been presented to carry out online 3D large-scale mesh reconstruction using manifold mapping and monocular-camera-based localisation applied to urban mapping and evaluated on the KITTI autonomous driving dataset [19]. An online localisation and dense scalable 3D map reconstruction architecture has been presented in [20] with grayscale coloring. It implements surfel based methods with the use of RGB-D, stereo and monocular cameras. Texture projection on reconstructed meshes is addressed in [21] and [22], however the presented methods consist of a post-processing of the whole data and are thus performed offline. In [23], a multi-robot system using stereovision-based localisation and 3D TSDF manifold mapping with grayscale coloring has been shown to run in real-time in an indoor environment. A real-time approach for creating and maintaining a colourised or textured surface reconstruction from RGB-D sensors has been introduced in [24], with a special focus on memory management for scalability.

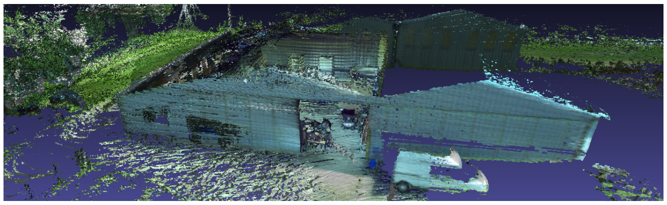
Fig. 1. Example of online colored mesh rendered by the proposed system

The Online Localisation and Colored Mesh Reconstruction (OLCMR) architecture proposed in the present paper is a complete system that performs both localisation and colored mesh reconstruction in real-time on board of a ground robot equipped with a 3D LiDAR scanner and multiple cameras. Figure 1 depicts an example of reconstruction produced by this system. The designed system builds upon recent open-source implementations of LiDAR-based SLAM and 3D mesh reconstruction methods that are summarized in

Q. Serdel, J. Marzat and J. Moras are with DTIS, ONERA, Université Paris-Saclay, 91123 Palaiseau, France.

C. Grand is with DTIS, ONERA, University of Toulouse, France.

Emails: {firstname.lastname}@onera.fr

Section II in perspective with related work, along with a description of the adaptations that were necessary to tackle the common objective pursued here. The overall performances of the proposed OLCMR architecture have then been evaluated on the handheld Newer College reference benchmark [25], [26] and are reported in Section III. The results and computational needs achieved on two experimental datasets acquired with tele-operated ground robots in urban and countryside outdoor environments are given in Section IV to demonstrate the versatility and wide applicability of the system. The results on these three different test cases include the evaluation of localisation (with and without loop-closure) and mapping accuracy with respect to independent reference models. Illustrations of mesh coloration are also presented for each dataset alongside images taken from the robot camera to highlight the quality of the whole reconstruction.

II. OLCMR Architecture description

The system architecture (Figure 2) has been designed to process data on-board of a ground robot for online missions in diverse uncharted environments. The main requirement was to be able to combine data from a 3D Laser scanner, one or several monocular cameras and an IMU, given intrinsic and extrinsic calibration parameters of these sensors (using e.g. [27]). The main objective is to compute online a dense colored 3D reconstruction with sufficient localisation accuracy, in order to be provided during the autonomous robot mission. The architecture is intended to function in various types of scenarios presenting challenging characteristics such as GNSS-denied, unstructured surroundings, dim or varying lights, uneven terrain.

It has been chosen to rely on a LiDAR-based SLAM (*Lidarslam_ROS2*) [28] for localisation to be light-invariant and to navigate in environments with possibly long ranges to points of interest, thus ruling out most of vision-based methods [4], [6]. The choice of the LiDAR as the main localization and mapping sensor also readily provides a 3D point cloud without any additional computational cost for the embedded CPU. IMU measurements and kinematic odometry (based on wheel encoders) are additionally used to robustify the robot localisation through EKF sensor fusion. The camera images are intended to enrich the produced 3D mesh by incorporating color levels or any kind of visually extracted data (e.g., semantic classification in a future perspective) through their projection in the 3D dense reconstruction using their relationship with the LiDAR point cloud. The TSDF-based mapping method *Voxblox* [29] is used to obtain this 3D dense reconstruction of the environment

The processing required by OLCMR is entirely performed on CPU (see related evaluation results in Section IV). The architecture is implemented under ROS2 Galactic, with the 3D mesh reconstruction running under ROS Noetic and communicating through a ROS1/ROS2 bridge [30].

A. Localisation

Reviews of open-source ROS SLAM implementations based on the use of stereo cameras, depth cameras or LiDAR are proposed in [3]–[5], [7]. For the previously stated reasons, our choice for a SLAM implementation has been restricted to the LiDAR based approaches. Since OLCMR is supposed to operate in complex outdoor environments (possibly unstructured and with uneven terrain) and relies on a 3D point cloud for mapping purposes, 2D LiDAR based approaches have been excluded. Recent SLAM implementations are often composed of two distinct parts. The SLAM front-end allows real-time localisation of the robot relative to its close environment, relying on high-rate sensors, either proprioceptive or external. The back-end estimates and corrects the localisation drift induced by the front-end over time. It run at a lower rate and relies either on absolute measurements such as GNSS as in LIO-SAM [31], or landmarks of known absolute position or on the recognition of previously visited areas (loop-closure). The vast majority of LiDAR-based SLAM front-end implementations are centered on the use of a scan-to-scan or scan-to-submap matching algorithm. The most used approach for scan-to-scan matching is Iterative Closest Point (ICP) [32], which however presents significant drawbacks in the studied context where the LiDAR clouds are composed of a vast number of points, depending on the sensor angular resolution and beam number (typical numbers being 16, 32, 64, 128). Thus, the sole application of an iterative, exhaustive algorithm such as ICP for scan matching can be suboptimal requiring a consequent amount of calculation to converge and grant satisfying results. A widely used solution to that issue is the extraction of interest points for each LiDAR scan using geometric properties of the cloud's distribution, as in LEGO-LOAM [33]. The ICP matching is then performed only using these more relevant points thus requiring way less computational capacity. However, the interest point extraction is still a costly process and requires the point cloud to present a somehow organised distribution for them to be extracted efficiently. As an alternative, a stochastic method based on normal approximation of the point cloud distribution named Normal Distribution Transform (NDT) was introduced in [34]. This makes the SLAM algorithm appropriate for evolution in unstructured environment as well as in geometrically rich ones, and requires limited computational power with no feature point extraction processing.

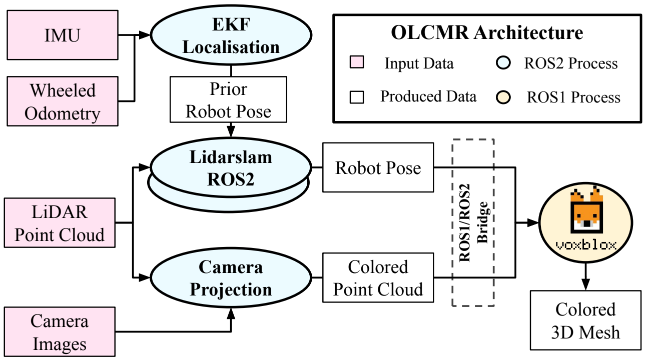

Fig. 2. OLCMR architecture overview

For these reasons, the *Lidarslam_ROS2* open-source

SLAM implementation of a LiDAR scan-to-scan matching has been integrated in the proposed architecture. In order to reduce the impact of locations that could be difficult to map (e.g. corridors with repeating geometrical patterns or bare flat fields) and for the SLAM to converge faster, the scan matcher takes a prior transform and differentiates it from the latest prior to use as initial guess for the transform between two scans. The prior choice is left to the discretion of the user. In the current case, the prior must present a good trade-off between accuracy and computational cost. It has thus been chosen to estimate it by fusing the forward velocity from wheeled odometry and the orientation and angular velocities from the IMU using an Extended Kalman Filter. The chosen method requires a significant overlap between successive scans, which could make it unsuitable to fast moving vehicles such as autonomous cars. In order to avoid some local failures in the scan-to-scan pose estimation (typically happening during fast rotations or very jittery motions), we proposed the following procedure: when the estimated transform from the last scan pose is greater than a threshold (set to 0.5 m) from the prior pose, this prior is used as final estimated transform and the associated scan is not registered into the map. The LidarSlam back-end performs loop-closure detection by comparing the current scan to stored key-frames using NDT and pose graph optimisation with the g^{2}o framework [35].

B. 3D Dense Reconstruction

The idea of producing a dense representation of the explored environment from the LiDAR point-clouds instead of using the sparse map produced by the SLAM implementation is driven by two different needs. In the first place, a point-cloud map is not the most adequate for visualisation by a human operator as the structure of mapped objects remains ambiguous when looking at it, especially in cluttered environments. Secondly, a dense representation is much more adapted to navigation purposes as it allows the robot to infer the terrain traversability at any coordinates without requiring further interpolation and therefore navigate in the full exploration space. Various methods for 3D dense representation of the environment have been developed by the robotic and computer vision community. Since OLCMR is intended to function in real-time, offline methods such as Structure from Motion and Poisson surface reconstruction have been dismissed. Voxel-based visualisation produced by methods such as Octomap [36] is well-suited for inclusion in autonomous navigation loops but less in terms of visualisation. Online surfacic methods such as the previously mentioned TSDF [16] use the information of sensor position relative to points in the cloud, thus removing the ambiguity of surface orientation and allowing to identify free space between the sensor and the mapped points. For these reasons, the 3D mesh reconstruction relies on the open-source implementation of the *Voxblox* ROS package [29] to build incrementally a surfacic mesh from each LiDAR scan using the generated point-cloud and the current robot pose (given by the SLAM/EKF process described in Section II-A) before fusing it in the globally reconstructed mesh.

C. 2D-3D color re-projection

A dedicated process handles the colourisation of the LiDAR points with the corresponding pixel values from the RGB camera images, for further inclusion in a 3D colored mesh. The projection of the 3D points into the camera images is computed geometrically [37] as follows. For each camera, the coordinates of the 3D LiDAR points are expressed into the camera frame, with R and t respectively being the rotation matrix and translation vector between the LiDAR (L) and camera (C) frames.

$$\begin{bmatrix} x_i & y_i & z_i \end{bmatrix}_C^T = R.\begin{bmatrix} x_i & y_i & z_i \end{bmatrix}_L^T + t \tag{1}$$

The coordinates are normalised by the point depth values.

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} 1/z_i & 0 \\ 0 & 1/z_i \end{bmatrix} . \begin{bmatrix} x_i \\ y_i \end{bmatrix}_C \tag{2}$$

The coordinates are projected into the image plane using the intrinsic camera matrix, where the parameters p_x, p_y, c_u and c_v come from the camera calibration process.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_x & 0 & c_u \\ 0 & p_y & c_v \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \tag{3}$$

Let $V(u, v)$ be the value (e.g. RGB color) of the pixel of coordinates u, v. Because cameras are affected by distortion, undistortion is applied to compute the image U using the distortion model and parameters given during calibration. If the resulting coordinates are inside the image bounds, the RGB value of the pixel of coordinates $[u, v]$ is allocated to the corresponding 3D LiDAR point. The color field is denoted as C.

$$C(x_i, y_i, z_i) = U(u, v) \tag{4}$$

Each surface reconstructed by the TSDF is then colored by the *Voxblox* pipeline with a recursive average filtering of its vertex colors.

For the evaluations of Section III, the full calibration parameters were available in the reference dataset. For the robot acquisitions from Section IV, the calibration process was performed using the Kalibr ROS package [27]. The intrinsic parameters and distortion model are estimated for each camera using a collection of images of an AprilTag grid of known dimensions. Kalibr exploits IMU measurements and the camera overlaps to estimate their extrinsic parameters. The transform between the LiDAR and the cameras was determined manually using the CAD model of the robots. Note that it is well known that combined LiDAR-vision systems can be very sensitive to the calibration between these two sensors for re-projection or SLAM purposes [9].

III. Evaluation on the Newer College dataset

Robot-held public datasets such as [38] or [39] do not contain all the required sensors at once (e.g. LiDAR and cameras). Vehicle-based datasets such as KITTI [19] are not adapted to our architecture, with large moving objects and reduced overlap between successive scans. The performances of the OLCMR architecture have thus been evaluated in terms of localisation precision and 3D reconstruction quality with the Newer College Dataset [25] and its 2021 extension [26], and a qualitative assessment of the mesh colourisation is also presented. The Newer College dataset extension provides LiDAR scans, IMU measurements and monocular images gathered from 4 cameras aboard a handheld device along various trajectories inside New College, Oxford. Ground-truth for the evaluation of SLAM and 3D reconstruction are provided using a tripod-mounted survey LiDAR and ICP registration, and many modern SLAM solutions have been evaluated on this dataset. For these reasons, this dataset has been deemed to be relevant for the evaluation of the OLCMR architecture performances. Although the architecture has been developed to function optimally aboard a ground robot, a few changes allowed it to be efficient while treating the data gathered from these handheld trajectories. The robot kinematic odometry used as prior has been replaced with a constant forward speed of 1.0 m/s and the 128 beam LiDAR point cloud has been down-sampled 10 times for SLAM input to limit its CPU usage. This section presents these evaluation results with relevant comparison to state-of-the-art. All evaluations were performed on a Intel Xeon(R) W-2123 8 core 3.60GHz CPU with 16 GB of RAM.

A. Localisation Evaluation

The localisation building blocks (LiDAR SLAM and EKF management of prior) of the proposed architecture have been evaluated on this dataset. The goal of this evaluation is to ensure that this localisation is sufficiently precise to be used for the overall colored mesh reconstruction. Following the localisation evaluation protocol proposed in [40], the Relative Pose Error over 10 m (RPE) and the Absolute Trajectory Error (ATE) are respectively computed for the *2021 quad-easy* and *2020 short-experiment* datasets. They are compared to state-of-the art SLAM evaluations on the same trajectories (as respectively reported in [41] and [42]). Fig. 3 shows the trajectories estimated by the SLAM scan matcher and loop closure optimiser superimposed on the ground truth trajectory, as well as yaw errors. Table I summarizes the SLAM performances. The localisation performance is consistent with the best currently available SLAM methods and the loop closure yields a small performance improvement, since the front-end SLAM presented only a small drift in this richly-textured environment.

B. 3D Reconstruction Evaluation

The Newer College dataset offers ground-truth 3D meshes of the visited area used to evaluate the quality of our 3D mesh reconstruction. After running the mapping pipeline of the architecture (composed of the LiDAR SLAM and Voxblox) on the Newer College dataset, the resulting uncolored mesh is treated and compared to the ground truth model using the CloudCompare open-source software[1]. Both meshes are sampled into dense point clouds which are then manually stripped of aberrant points produced by reflective surfaces (e.g. windows) by applying a planar cut-off beyond said surfaces. The closest point error between the two resulting point clouds is performed using the M3C2 method described in [48]. Points that have no relevant match are removed. 90% of our reconstruction model points show a distance error to the ground truth model lesser than 0.54 m. As a comparison, [8] states that 90% of the points from their reconstructed model show a distance error lesser than 0.50 m on a similar dataset. The results are illustrated in Figure 4. This validates the soundness of the overall LiDAR-based localisation and dense mapping algorithms incorporated in the architecture. The projection of the camera grayscale levels has also been carried out as detailed in Section II-C, and the associated qualitative result is given in Figure 5. The CPU and RAM usage have also been monitored during the processing of the trajectories (see Figure 6). It turns out that the CPU needs have been successfully adjusted to avoid the saturation of available resources with limited growth over time, and

TABLE I
SLAM EVALUATION ON THE NEWER COLLEDGE DATASET

nc 2021 quad-easy		***nc 2020 short-experiment***	
Duration	198.7 s	**Duration**	1530.0 s
Length	245.7 m	**Length**	1570.3 m
SLAM	**RPE**	**SLAM**	**APE**
OpenVINS [43]	1.01 m	A-LOAM [44]	3.308 m
ORB-SLAM3 [45]	0.23 m	F-LOAM [46]	101.9 m
VILENS-MC [41]	0.31 m	MARS [42]	1.978 m
VILENS-Stereo	0.30 m	SuMa [47]	2.048 m
LidarSlam SM	0.28 m	LidarSlam SM	5.141 m
LidarSlam LC	0.27 m	LidarSlam LC	4.010 m

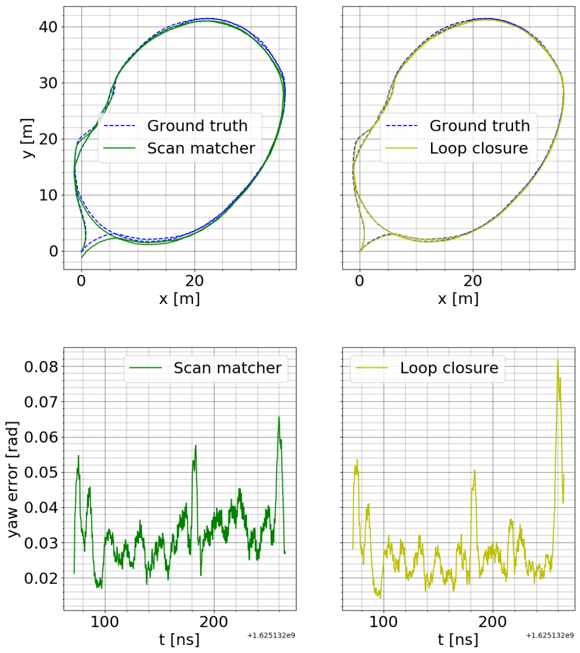

Fig. 3. Newer College localisation evaluation, top shows trajectories estimated after scan matching (left) and loop closure optimisation (right) superposed to the ground truth trajectory. Bottom shows Absolute Yaw Error smoothed over 50 samples for same trajectories.

[1] CloudCompare website : `https://www.danielgm.net/cc/`

that the RAM requirements increase quite linearly which is a usual behaviour of SLAM systems as the map size increases. By performing a linear fit to the RAM usage, the maximum duration of a mission with similar settings is roughly estimated to 3142 s.

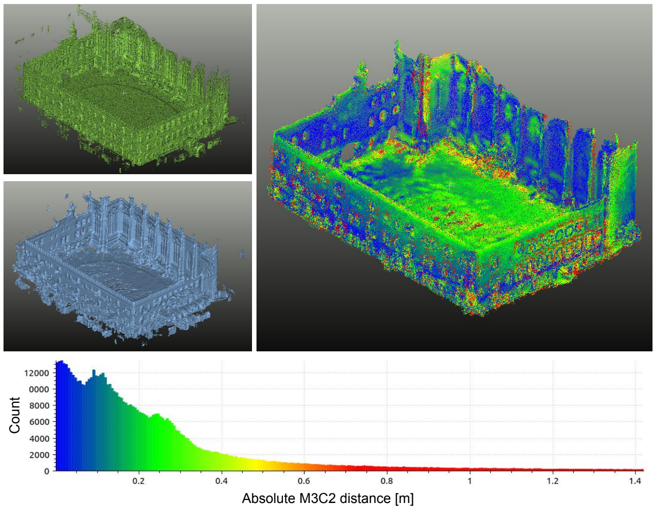

Fig. 4. Newer College reconstruction comparison: Top left shows the reference 3D model from the dataset, bottom left shows the output of our architecture and right shows the absolute reconstruction error point cloud with histogram, output of MC3C2 absolute distance computed on the Newer College dataset.
Mean = 0.236 m, Std = 0.248 m

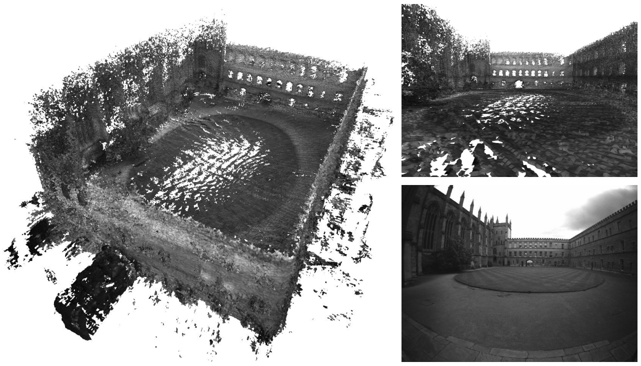

Fig. 5. 3D mesh with mono camera projection, output of the application of the complete OLCMR architecture on the Newer College Dataset. Left shows the complete reconstructed mesh, right shows a comparison between a camera frame and a corresponding view of the reconstructed mesh.

IV. Evaluation on Field Robots Trajectories

The proposed method obtained good performances on the Newer College dataset, which contains handheld trajectories acquired in a highly-textured environment. To further evaluate the OLCMR architecture in situations closer to its goal applications, we acquired two dedicated experimental datasets produced by tele-operating wheeled ground robots along predefined trajectories and gathering relevant perception data. These two datasets were respectively produced on-board of a Robotnik Summit-XL robot in a urban environment and an Agilex Scout robot in a countryside environment. These two platforms are four-wheel differentially driven, equipped with on-board CPUs running Ubuntu and ROS, and their respective sensor suites are summarized in Table II. OLCMR components main parameters for each dataset are referenced in Table IV. The data processing has been performed on the same computer as for the reference dataset evaluation from Section III, which presents similar computational power as the robots' embedded computers.

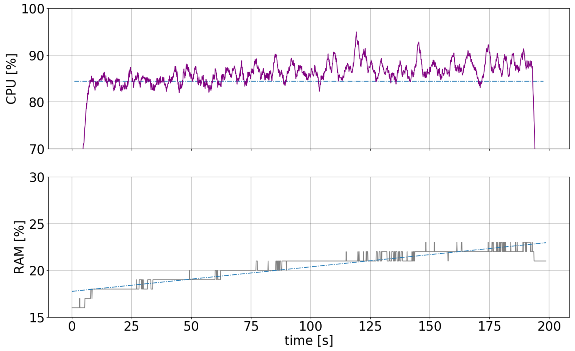

Fig. 6. CPU (mean: 84.4 %, std: 11.7 %) and RAM (max: 23 %) usage of OLCMR while running on the Newer College dataset. Mean CPU usage and linear regression of RAM usage are displayed on relevant figures.

TABLE II
Sensors embedded on ONERA-DTIS Summit and Scout robots

Sensor	*Technical Reference*	*Rate*
	Summit	
Wheel Encoders	Robotnik embedded	50 Hz
IMU	VectorNav VN-100-t	400 Hz
LiDAR	Velodyne VLP-16	10 Hz
3x RGB Cameras	uEye IDS 3241LE-C	10 Hz
	Scout	
Wheel Encoders	AgileX embedded	50 Hz
IMU	XSens MTi 300	400 Hz
LiDAR	Ouster OS1-32	10 Hz
3x RGB Cameras	Basler dart daA1600-60uc	10 Hz

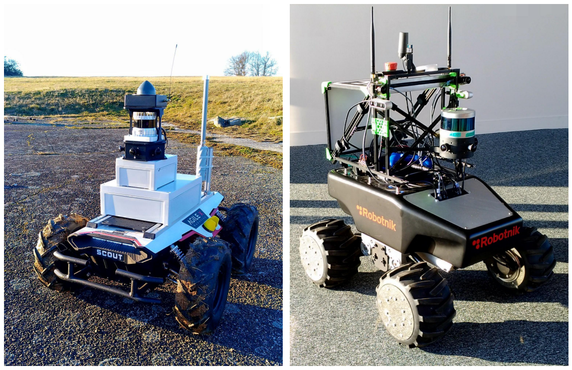

Fig. 7. ONERA-DTIS Agilex Scout and Robotnik Summit XL robot setups

A. Localisation Evaluation

The localisation global performance has been evaluated with respect to the total drift of the trajectories, so this drift is estimated by the difference between the last pose and the first pose computed given the fact that robots were operated in order for the actual ending pose to be approximately equal to the actual starting pose. The characteristics of the trajectories

TABLE III

SLAM EVALUATION ON FIELD DATASETS

Summit (urban)	
Trajectory duration	516.605 s
Trajectory length	400.295 m
Scan matcher final APE	4.794 m
Loop closure final APE	1.943 m
Final loop closure correction	3.120 m
Scout (countryside)	
Trajectory duration	399.399 s
Trajectory length	407.045 m
Scan matcher final APE	8.582 m
Loop closure final APE	1.486 m
Final loop closure correction	8.034 m

and the values of the approximated final APE and loop-closure total corrections are summarized in Table III, and Figure 8 presents the robot estimated trajectories. The overall drift without loop-closure remains between 1 and 2 percents, with a higher value in the less-textured environment. These degradations compared to the previous dataset could be interpreted by the transition from high-quality handheld sensors to robot-mounted lower-grade IMU and LiDAR sensors. These performances remain acceptable to carry out the online 3D dense model rendering process.

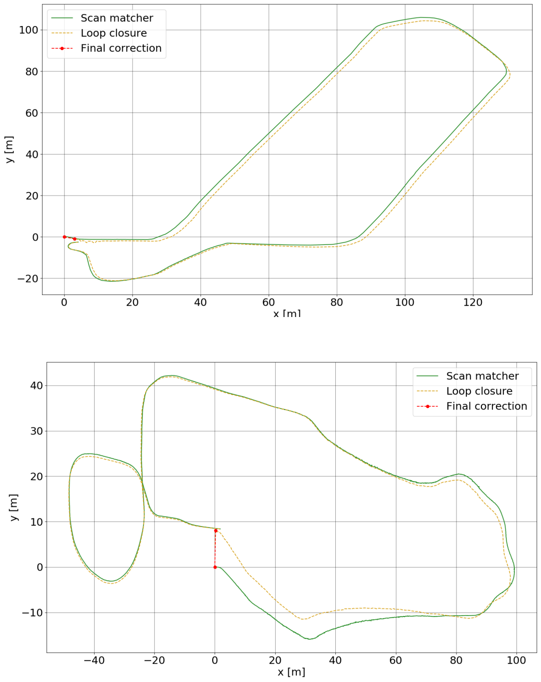

Fig. 8. Robot trajectories on field datasets. Top with the Summit robot, bottom with the Scout robot.

B. 3D Reconstruction Evaluation

Images have been acquired independently from the robot setups in the two new test environments using handheld camera devices, respectively a stereo bench composed of 2 uEye IDS 1241LE-M monocular cameras for the Summit (urban) dataset and a HERO7 GoPro for the Scout (countryside) dataset. An offline photogrammetric 3D reconstruction has then been obtained using the Structure-From-Motion *Colmap* software [49]. The reconstruction errors between the mapping obtained with the OLCMR architecture using the robots embedded sensors and these Colmap-generated reference models have been analyzed using CloudCompare (see Figure 9). The colored rendering mesh is presented in Figure 10. The CPU and RAM usage evaluations (Figure 11) show a similar behaviour as on the Newer College dataset, which seem to be compatible with on-board actual deployment for trajectories lengths of one-kilometer order of magnitude.

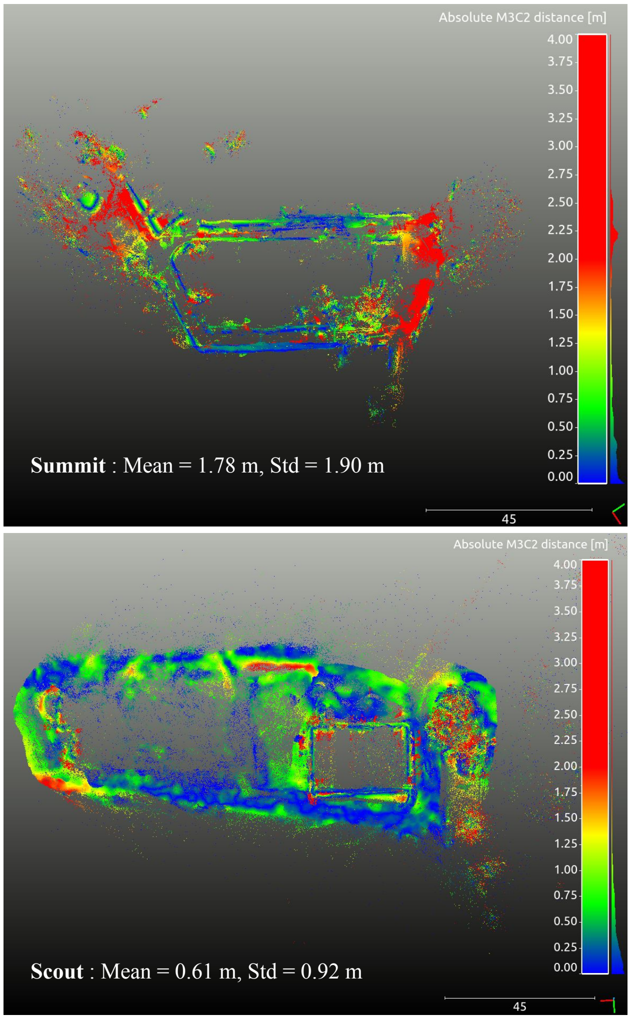

Fig. 9. Reconstruction error point cloud map and histogram, output of M3C2 absolute distance computed from Summit (top) and Scout (bottom).

V. Conclusions and Perspectives

An architecture running in real-time combining LiDAR-based localisation with 3D dense mapping and mesh colourisation using multiple cameras has been proposed in this paper. It is based on recent open-source ROS package implementations of a LiDAR SLAM and a TSDF-based mapping algorithms, with specific developments to assemble the overall pipeline. The full system has been thoroughly evaluated on datasets exhibiting different characteristics, namely the *Newer College* handheld benchmark and two dedicated trajectories acquired with wheeled ground robots

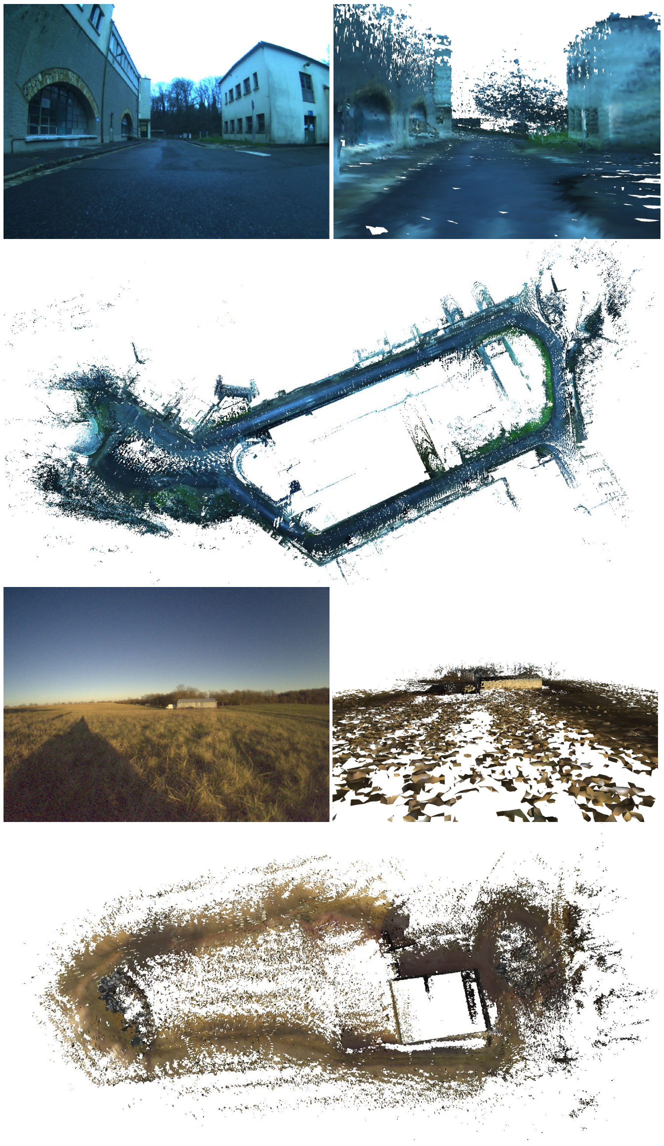

Fig. 10. Mesh rendering for Summit robot (urban) dataset (top) and Scout robot (countryside) dataset (bottom). For each dataset, example of picture with corresponding view of the OLCMR colored mesh and bird-eye view of the global colored mesh are shown.

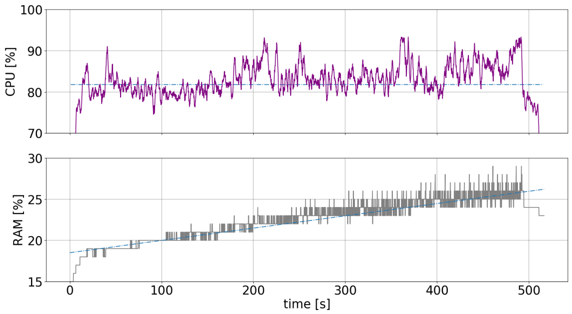

Fig. 11. CPU (mean: 81.782 %, std: 11.7 %) and RAM (max: 29 %) usage of OLCMR while on the Summit robot data. Mean CPU usage and linear regression of RAM usage are displayed on relevant figures.

TABLE IV

OLCMR open-source components main parameters for each experiment

parameter	*Newer College*	*Summit*	*Scout*
LidarSlam scan matcher			
ndt resolution	1.5	1.0	0.8
vg size input	0.1	0.1	0.1
vg size map	0.1	0.1	0.1
min range	1.0	1.0	3.0
max range	50.0	50.0	100.0
num targeted cloud	20	20	20
LidarSlam loop closure			
ndt resolution	1.0	1.0	1.0
voxel leaf size	0.1	0.1	0.1
detection period	7500	4000	4000
threshold loop closure	15.0	15.0	15.0
distance loop closure	50.0	50.0	50.0
loop closure search range	50.0	50.0	50.0
num submap searched	10	20	20
num adj. pose constraints	10	20	20
Voxblox			
voxel size	0.2	0.125	0.15
voxels per side	4	8	8
carving	true	false	false
use free space	false	false	false
method	fast	merged	merged
constant weight	false	false	false
allow clear	false	false	false
min ray length	2.0	0.5	0.0
max ray length	200.0	200.0	200.0

in urban and countryside environments. The architecture performed well in all of these conditions, which make it suitable for future field deployment of tele-operated or autonomous robotic exploration.

The loop-closure is currently used solely by the localisation stack. To improve the development of the OLCMR architecture, it could also be integrated within the mapping stack in a manifold framework such as [23]. Additional layers could be added to this perception architecture, e.g. for semantic navigation in unknown environments. The next step towards that goal would be to implement semantic segmentation on camera images before projecting them into the 3D space in order to create semantic maps that could be used by the robot for safer and better autonomous navigation.

References

[1] Iker Lluvia, Elena Lazkano, and Ander Ansuategi. Active mapping and robot exploration: A survey. *Sensors*, 21(7):2445, 2021.

[2] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81, 2015.

[3] Yuri DV Yasuda, Luiz Eduardo G Martins, and Fabio AM Cappabianco. Autonomous visual navigation for mobile robots: A systematic literature review. *ACM Computing Surveys (CSUR)*, 53(1):1–34, 2020.

[4] Alexey Merzlyakov and Steve Macenski. A comparison of modern general-purpose visual SLAM approaches. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9190–9197, 2021.

[5] Mahdi Elhousni and Xinming Huang. A survey on 3D LiDAR localization for autonomous vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1879–1884, 2020.

[6] Qin Zou, Qin Sun, Long Chen, Bu Nie, and Qingquan Li. A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[7] Mathieu Labbé and François Michaud. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446, 2019.

[8] Yiduo Wang, Nils Funk, Milad Ramezani, Sotiris Papatheodorou, Marija Popović, Marco Camurri, Stefan Leutenegger, and Maurice Fallon. Elastic and efficient LiDAR reconstruction for large-scale exploration tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5035–5041, 2021.

[9] César Debeunne and Damien Vivet. A review of visual-lidar fusion based simultaneous localization and mapping. *Sensors*, 20(7):2068, 2020.

[10] Pileun Kim, Jingdao Chen, and Yong K Cho. SLAM-driven robotic mapping and registration of 3D point clouds. *Automation in Construction*, 89:38–48, 2018.

[11] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! large-scale texturing of 3D reconstructions. In *European Conference on Computer Vision (ECCV)*, pages 836–850, 2014.

[12] Ting On Chan, Hang Xiao, Lixin Liu, Yeran Sun, Tingting Chen, Wei Lang, and Ming Ho Li. A post-scan point cloud colorization method for cultural heritage documentation. *ISPRS International Journal of Geo-Information*, 10(11):737, 2021.

[13] André Silva Aguiar, Filipe Neves dos Santos, José Boaventura Cunha, Héber Sobreira, and Armando Jorge Sousa. Localization and mapping for robots in agriculture and forestry: A survey. *Robotics*, 9(4):97, 2020.

[14] Marc Pollefeys and al. Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision*, 78:143–167, 07 2008.

[15] M. Klingensmith, I. Dryanovski, S. Srinivasa, and J. Xiao. CHISEL: Real Time Large Scale 3D Reconstruction Onboard a Mobile Device using Spatially Hashed Signed Distance Fields. In *Robotics: Science and Systems XI*, volume 4, 2015.

[16] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996.

[17] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 2013.

[18] Enrico Piazza, Andrea Romanoni, and Matteo Matteucci. Real-time CPU-based large-scale 3D mesh reconstruction. *arXiv preprint arXiv:1801.05230*, 2018.

[19] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[20] Kaixuan Wang, Fei Gao, and Shaojie Shen. Real-time scalable dense surfel mapping.

[21] Andrea Romanoni, Daniele Fiorenti, and Matteo Matteucci. Mesh-based 3D textured urban mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3460–3466, 2017.

[22] Mohamed Boussaha, Bruno Vallet, and Patrick Rives. Large scale textured mesh reconstruction from mobile mapping images and LiDAR scans. In *International Society for Photogrammetry and Remote Sensing*, pages 49–56, 2018.

[23] Thibaud Duhautbout, Julien Moras, and Julien Marzat. Distributed 3D TSDF manifold mapping for multi-robot systems. In *European Conference on Mobile Robots (ECMR)*, 2019.

[24] Simon Schreiberhuber, Johann Prankl, Timothy Patten, and Markus Vincze. Scalablefusion: High-resolution mesh-based real-time 3D reconstruction. In *International Conference on Robotics and Automation (ICRA)*, pages 140–146, 2019.

[25] Milad Ramezani, Yiduo Wang, Marco Camurri, David Wisth, Matias Mattamala, and Maurice Fallon. The newer college dataset: Handheld LiDAR, inertial and vision with ground truth. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[26] Lintong Zhang, Marco Camurri, and Maurice Fallon. Multi-camera LiDAR inertial extension to the Newer College dataset. *arXiv preprint arXiv:2112.08854*, 2021.

[27] Jérôme Maye, Paul Furgale, and Roland Siegwart. Self-supervised calibration for robotic systems. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 473–480, 2013.

[28] Lidarslam ros2. "https://github.com/rsasaki0109/lidarslam_ros2", 2020.

[29] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3D euclidean signed distance fields for on-board MAV planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[30] George Stavrinos. ROS2 for ROS1 users. In *Robot Operating System (ROS)*, pages 31–42. Springer, 2021.

[31] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Rus Daniela. LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE, 2020.

[32] Juyong Zhang, Yuxin Yao, and Bailin Deng. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[33] Tixiao Shan and Brendan Englot. LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, 2018.

[34] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2743–2748, 2003.

[35] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G^{2}o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.

[36] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013.

[37] Robert M Haralick. Using perspective transformations in scene analysis. *Computer Graphics and Image Processing*, 13(3):191–221, 1980.

[38] Thierry Peynot, Steve Scheding, and Sami Terho. The Marulan data sets: Multi-sensor perception in a natural environment with challenging conditions. *The International Journal of Robotics Research*, 29(13):1602–1607, 2010.

[39] Abdulbaki Aybakan, Garen Haddeler, M. Caner Akay, Osman Ervan, and Hakan Temeltas. A 3D LiDAR dataset of ITU heterogeneous robot team. In *Proceedings of the 2019 5th International Conference on Robotics and Artificial Intelligence*, ICRAI '19, page 12–17, 2019.

[40] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580, 2012.

[41] Lintong Zhang, David Wisth, Marco Camurri, and Maurice Fallon. Balancing the budget: Feature selection and tracking for multi-camera visual-inertial odometry. *IEEE Robotics and Automation Letters*, 7(2):1182–1189, 2022.

[42] Jan Quenzel and Sven Behnke. Real-time multi-adaptive-resolution-surfel 6d lidar odometry using continuous-time trajectory optimization, 2021.

[43] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. OpenVINS: A research platform for visual-inertial estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020.

[44] Ji Zhang and Sanjiv Singh. Loam : Lidar odometry and mapping in real-time. *Robotics: Science and Systems Conference (RSS)*, pages 109–111, 01 2014.

[45] Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M.M Montiel, and Juan D. Tardos. ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[46] Han Wang, Chen Wang, Chun-Lin Chen, and Lihua Xie. F-loam: Fast lidar odometry and mapping. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2021.

[47] Jens Behley and Cyrill Stachniss. Efficient surfel-based slam using 3d laser range data in urban environments. 06 2018.

[48] Mike R. James, Stuart Robson, and Mark W. Smith. 3-D uncertainty-based topographic change detection with structure-from-motion photogrammetry: precision maps for ground control and directly georeferenced surveys. *Earth Surface Processes and Landforms*, 42(12):1769–1788, 2017.

[49] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.