Robotics and Autonomous Systems 155 (2022) 104185

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

# Strategies for large scale elastic and semantic LiDAR reconstruction

Yiduo Wang [a,*], Milad Ramezani [a,b,1], Matias Mattamala [a], Sundara Tejaswi Digumarti [a], Maurice Fallon [a]

[a] Oxford Robotics Institute, The George Building, 23 Banbury Road, Oxford OX2 6NN, UK
[b] CSIRO Data61, Queensland Centre for Advanced Technologies, 1 Technology Court, Pullenvale QLD 4069, Australia

ARTICLE INFO

*Article history:*
Available online 26 June 2022

*Keywords:*
LiDAR reconstruction
Pose-graph SLAM
Submaps
Semantic segmentation

ABSTRACT

This paper presents novel strategies for spawning and fusing submaps within an elastic dense 3D reconstruction system. The proposed system uses spatial understanding of the scanned environment to control memory usage growth by fusing overlapping submaps in different ways. This allows the number of submaps and memory consumption to scale with the size of the environment rather than the duration of exploration. By analysing spatial overlap and semantic information, our system segments distinct spaces on-the-fly during exploration, such as rooms, stairwells, and indoor–outdoor transitions. The proposed system associates semantically labelled submaps with poses of SLAM pose graph to enable global elasticity. A probabilistic model to merge the voxel labels of the different submaps is incorporated to ensure correct semantic submap fusion when SLAM loop closures occur. Additionally, we present a new mathematical formulation of relative uncertainty between poses to improve the global consistency of the reconstruction. Performance is demonstrated using experiments exploring multi-floor multi-room indoor environments, indoor–outdoor transitions and large-scale outdoor experiments. Relative to our baseline, the presented approach demonstrates improved scalability and accuracy.



## 1. Introduction

3D reconstruction is a common part of applications such as active mapping [1,2], collision avoidance [3] and the inspection of industrial assets [4,5]. Building Information Models (BIM) are commonly available for modern buildings, but there are scenarios where these models no longer represent the real situation, e.g. after renovations or disasters. Although systems have been developed to reconstruct these models offline using point clouds from laser scanners, autonomous exploration and reconstruction in multi-storey environments is still an open challenge in mobile robotics. This has been the motivation of international competitions such as the DARPA SubT challenge [6,7].

In this paper, we propose a dense 3D reconstruction system that fuses LiDAR scans to map complex multi-storey scenarios, such as the example shown in Fig. 1. This paper is an extension to a previous paper from European Conference on Mobile Robots (ECMR) [8], which introduces multi-camera semantic information into the dense 3D LiDAR reconstruction.

The proposed system uses a state-of-the-art reconstruction method, *supereight* [9], to create efficient high-resolution representations of large-scale environments with long range sensing. The system's frontend relies on accurate multi-sensor odometry to build a local occupancy map around the robot; its backend creates submaps and associates them with the corresponding pose from a SLAM pose graph — instead of building a single global map. This approach can account for future loop closure corrections and achieve a globally consistent reconstruction.

We present strategies for spawning and fusing submaps based on geometric and semantic understanding of the observed spaces, enabling on-the-fly segmentation of areas that are physically isolated or significantly different from one another, e.g. individual rooms indoors or the transition between indoor and outdoor spaces.

*Supereight* provides the ability to store either Truncated Signed Distance Function (TSDF) or occupancy probability. In our work we use the occupancy representation, which explicitly models free space to analyse spatial overlap among submaps. It also fuses submaps together to reduce redundant reconstruction. In addition, by computing the relative uncertainty among poses in a SLAM pose graph, our system can avoid fusing submaps incorrectly to improve overall reconstruction accuracy.

* Corresponding author.
*E-mail addresses:* ywang@robots.ox.ac.uk (Y. Wang), milad.ramezani@data61.csiro.au (M. Ramezani), matias@robots.ox.ac.uk (M. Mattamala), tejaswid@robots.ox.ac.uk (S.T. Digumarti), mfallon@robots.ox.ac.uk (M. Fallon).

[1] This presented work was done when the author was with the Oxford Robotics Institute.

https://doi.org/10.1016/j.robot.2022.104185

**Fig. 1.** The proposed system has been tested in large-scale outdoor and multi-storey indoor environments with both a handheld LiDAR system (top, left) and the quadruped robot Spot (top, right); **Bottom**: Spatially clustered pose graph nodes and the semantically labelled LiDAR reconstruction from our indoor–outdoor transition experiment. The semantic labels are applied outdoors while the multi-storey indoor section is indicated in grey. More demonstrations are available in the supplementary video: https://youtu.be/GwS2KqP3lLo.

Our approach also integrates per-voxel semantic information into the occupancy submaps and ensures spatial and semantic consistency during submap fusion. This is achieved by probabilistically merging the semantic labels of the voxels across multiple submaps.

We demonstrate the performance of the proposed system using both simulation and real-world experiments, especially its improvement in reconstruction accuracy, system scalability, and the capability of room segmentation and indoor–outdoor detection.

The features and contributions of our research are the following:

- New strategies for spawning and fusing submaps using probabilistic, semantic and spatial understanding.
- A new formulation for relative uncertainty derived from the work of Mangelson et al. [10] and GTSAM [11], and a formal treatment of uncertainty in submap fusion.
- On-the-fly adjustment of reconstruction parameters when transiting between indoors and outdoors, based on semantic analysis.
- Details of modifications to the *supereight* LiDAR reconstruction pipeline [9] to incorporate motion undistortion.
- Introduction of a state-of-the-art dense semantic mapping module [12] into an elastic submapping framework.
- Improved reconstruction accuracy and scalability in both outdoor exploration experiments and a multi-storey multi-room exploration with a legged robot.

The overarching goal of the work is to achieve scalability with the size of the environment instead of the exploration length by controlling the growth of the submap number and memory consumption[2].

The remainder of this paper is organised as follows. In Section 2 we discuss related work, and the proposed system is explained in detail in Section 3. The new features of the extended system compared to our previous work [8] are presented in Sections 3.4 and 3.5. Section 4 demonstrates the experimental results, with new experiments on integrating semantics included in Section 4.3. Conclusions and future work are discussed in Section 5.

## 2. Related work

While the literature on dense 3D reconstruction is substantial, in this paper we focus on systems incorporating submaps, methods designed for room segmentation and semantic segmentation approaches because these aspects are the most relevant to our work.

### 2.1. 3D Reconstruction via submaps

Submapping is a common technique in SLAM systems such as the *Atlas* framework by Bosse et al. [13] and DenseSLAM by Nieto et al. [14]. These SLAM systems reuse existing submaps for localisation when revisiting a known area rather than remapping that location. Maintaining a collection of submaps keeps memory and computation bounded as opposed to creating a single global map.

Submaps can also enable elasticity when a reconstruction requires correction at the event of a loop closure. Leveraging OctoMap-based [15] submaps, Ho et al. [16] and Sodhi et al. [17] designed elastic occupancy maps. Voxgraph [18] uses the same strategy to create a globally consistent reconstruction, instead represented by a series of Signed Distance Function (SDF) submaps. In our previous work [19], we developed submap-based elastic reconstruction pipelines which supported both occupancy and SDF representations.

These approaches spawn new submaps after either a temporal interval [16–18] or distance travelled [19] to bound the local odometry drift within each submap. Extending upon [19], this paper proposes submap spawning and fusion based on a deeper spatial and semantic analysis to uniquely represent confined or distinct areas such as rooms. This allows local planning to use only a submap in challenging exploration tasks such as [20].

### 2.2. 3D Room segmentation

Separately, several works have developed methods to segment LiDAR reconstructions or floor plans into individual enclosed spaces. Turner and Zakhor [21] designed a room-segmentation pipeline that partitioned 2D point cloud maps into 2.5D building models via triangulation. To achieve 3D reconstruction, the approach assumed that interior walls were vertical and planar.

More sophisticated methods were developed to parse 3D point clouds into rooms, such as detecting void spaces between walls using point density histograms [22], and extracting planar features before partitioning separate rooms via a multi-label energy minimisation formulation [23,24]. These methods were limited to single-storey reconstructions.

Ochmann et al. [25] and Nikoohemat et al. [26] proposed methods that handle unstructured 3D point clouds for multi-storey room segmentation. Ochmann et al. introduced a versatile

[2] Sparsification of the underlying SLAM pose graph is a related research topic which we do not explore in this work.

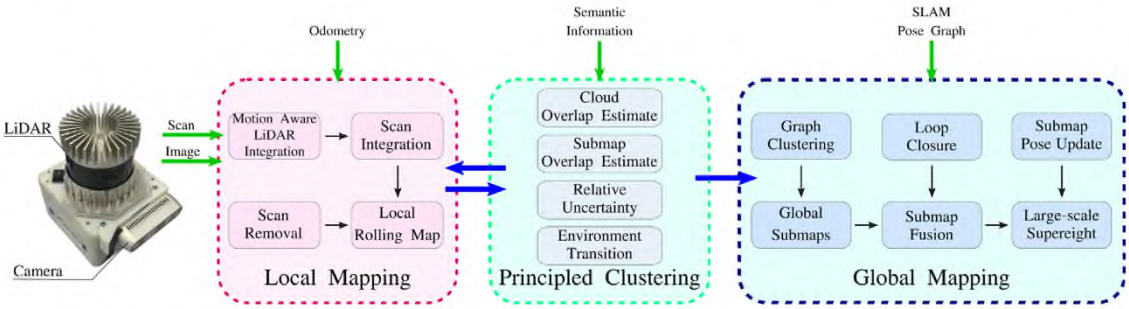



**Fig. 2.** An overview of the proposed system that consists of a frontend (*Local Mapping*) and a backend (*Global Mapping*). *Principled Clustering* contains the proposed new strategies which estimate cloud and submap overlap, measure relative pose uncertainty in the SLAM graph, and leverage semantic information to detect transitions between indoor and outdoor environments. This module interacts with *Local Mapping* and *Global Mapping* to spawn new submaps (Section 3.1), propose or reject submap fusion (Sections 3.2 and 3.3), and on-the-fly adjust reconstruction parameters based on semantic information (Section 3.4).

integer linear programming method to incorporate hard constraints, e.g. wall connectivity, to ensure a plausible reconstruction. Nikoohemat et al. employed a mobile LiDAR SLAM system and separated building levels by assuming that sloped trajectory segments represent staircase traversals. They further segmented rooms using an adjacency graph of planer segments.

While the effect of partitioning confined spaces is similar to our proposed system, we focus on online segmentation running onboard a mobile robot instead of segmenting a complete reconstruction offline. Our proposed system aims to create segmented 3D reconstructions of confined spaces such as individual rooms and staircases on the fly in the context of multi-floor exploration, and distinguish between indoor and outdoor environments to create elastic connections in the map.

### 2.3. Semantic segmentation

Richer representations of a robot's environment, including for example semantic information of the objects and structures around the robot, can enable better decision making and informed navigational planning.

Estimating dense per-pixel semantic labels for images has been an active area of research [27]. Early approaches to this task used fully convolutional networks with skip connections [28] and dilated convolutions to aggregate multi-scale features [29]. Badrinarayanan et al. [30] introduced SegNet with an encoder–decoder architecture with non-linear upsampling in the decoder and max pooled indices in the encoder to improve computational and memory efficiency. A related network structure that is widely used is the UNet [31], which uses skip connections between the encoder and decoder layers to overcome the vanishing gradients problem. Later works improved the performance and learning capabilities of these networks by incorporating temporal consistency [32], learning saliency to avoid over-fitting [33] and motion estimation [34]. Recent work has also looked at novel network architectures [35,36] to improve efficiency and robustness.

The use of 2D convolutions for semantic segmentation has been extended to 3D point clouds, by projecting 3D output from LiDAR scanners to images, using spherical projection [37,38]. Alternatively, 3D convolutions have also been applied on voxelized representations of point clouds [39,40], but are inefficient due to the sparsity of points, especially in outdoor environments. Zhu et al. [41] propose a cylindrical partitioning of 3D space with asymmetrical 3D convolutions to address this issue. Recently, pointwise convolutions, which can be applied directly on 3D points, have been proposed by Thomas et al. [42] to improve robustness to varying densities of the point cloud.

However, direct 3D approaches have higher computational requirements relative to 2D approaches and they also do not scale well to large scenes. As a result, their deployment on a mobile robotics platform is still an open challenge. When multiple sensor modalities (such as cameras and LiDARs) are present it is preferable to extract semantics from camera images and use these predictions to label the 3D points. We take this approach where images from multiple cameras mounted on the robot are used to extract per-pixel semantics. These are then used to label the 3D map reconstructed by aggregating the LiDAR scans.

Given per-pixel semantic labels, a subsequent challenge is fusing them into a consistent map over time. This has been explored in [43,44]. More recently, Gan et al. [12] presented a probabilistic framework to fuse multiple semantic labels to generate a continuous 3D semantic occupancy map. While the results presented are convincing, they are limited to a single camera mounted on a mobile robot and did not consider scalability or issues of revisiting already explored space.

In our work, we extend their framework and address the challenges of fusing semantic labels from multiple cameras, with different viewpoints, mounted on the robot. This is key for improving navigational capabilities for mobile robots where awareness of the environment all around the robot is necessary. Furthermore, we integrate this extended framework into a submapping system and fuse predictions across submaps to maintain a consistent semantically annotated map under elastic deformations due to loop closure corrections.

## 3. Methods

The proposed system expands upon our previous framework [19] by adding principled strategies that use spatial understanding, semantic information and pose graph relative uncertainty to improve the spawning and fusion of submaps. The proposed system employs the *supereight* [9] reconstruction method, which allows scans to be integrated efficiently via adaptive resolution. The options for map representation given by *supereight* include both TSDF and occupancy probability, and we focus on the latter in the proposed system. Our previous work extended the projection and uncertainty model to better suit the characteristics of long-range LiDAR sensors [19]. In this work, we further modified the LiDAR *supereight* pipeline to efficiently integrate motion undistorted LiDAR measurements. We use a multithreaded frontend/backend structure to maintain both a local rolling map and a collection of global submaps during exploration. Fig. 2 provides an overview of the proposed system.

While *supereight* also provides an odometry functionality, we do not use it and instead rely on external odometry and SLAM modules [45]. The odometry module, such as one built upon [46], is an input to the frontend of the proposed system, to provide the pose of the LiDAR sensor frame $\{\mathcal{L}\}$ in a global odometry frame $\{\mathcal{O}\}$ at time $t$, denoted as ${}^{\mathcal{O}}\mathbf{T}_{\mathcal{L}_t} \in \mathbf{SE}(3)$. This pose is associated with a raw point cloud $C_t$ expressed in the LiDAR frame $\{\mathcal{L}_t\}$.

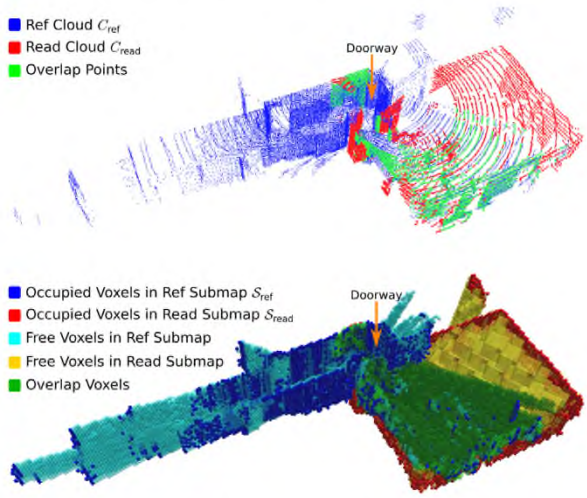



**Fig. 3.** An example of Cloud Overlap Estimate and Submap Overlap Estimate. **Top**: Limited overlap between the reference cloud $C_{\text{ref}}$ and the read cloud $C_{\text{read}}$ when entering a room through a narrow doorway. **Bottom**: The voxel overlap between the volumetric occupancy reconstructions of the reference submap $\mathcal{S}_{\text{ref}}$ and the read submap $\mathcal{S}_{\text{read}}$.

Our reconstruction pipeline also requires access to the solution of a graph SLAM system, i.e. $Q+1$ nodes $X_k$, $k \in \{0,\ldots,Q\}$, and the Hessian matrix associated with the solution [47]. Each node describes the estimated pose of the LiDAR frame $\{\mathcal{L}_k\}$ with respect to a fixed map frame $\{\mathcal{M}\}$, denoted as ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{L}_k} \in \mathbf{SE}(3)$.

The SLAM system used is a pose-graph method based on Ramezani et al. [45], where the only inputs are relative odometry measurements and loop closure candidates, both represented as $\mathbf{SE}(3)$ matrices. The covariances associated with these measurements are fixed and tuned using line-search. The solution is determined in an incremental fashion using the iSAM2 algorithm [48] available in the GTSAM library [11].

The output of the reconstruction system consists of a local rolling map and a collection of $N+1$ submaps. *Local Mapping* creates and maintains a local map by integrating the latest point cloud $C_t$ into a map in $\{\mathcal{O}\}$. To bound the memory consumption of this map, we crop it around pose ${}^{\mathcal{O}}\mathbf{T}_{\mathcal{L}_t}$ based on the sensing range of the LiDAR.

In *Global Mapping*, each submap $\mathcal{S}_i, i \in \{0,\ldots,N\}$ contains an accumulated submap cloud $C_{\mathcal{S}_i}$ in $\{\mathcal{M}\}$, a volumetric *occupancy* reconstruction in the submap frame $\{\mathcal{S}_i\}$ and its root pose ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_i}$. Occupancy representation is chosen as the focus in the proposed work because of its explicit representation of free space. This aids the submap overlap estimation and will be detailed in Section 3.2.

We compute and maintain Axis-Aligned Bounding Boxes (AABB) for each submap in $\{\mathcal{M}\}$ using the reconstruction. The Submap's AABB is significantly affected by the orientation of $\{\mathcal{M}\}$, so we use AABBs not to accurately estimate scanned space, but as a lightweight method to determine non-overlapping submaps (Section 3.2).

In the proposed system, we introduce the following set of strategies which we use to decide when to spawn or fuse submaps:

- *Cloud Overlap Estimate*: to adjust the submap spawning decisions made by Local Mapping.
- *Submap Overlap Estimate*: to propose submap fusion for Global Mapping.
- *Relative Uncertainty*: to reject unreliable submap fusions.
- *Environment Transition Criterion*: to differentiate between indoor and outdoor environments based on semantic information.

**Algorithm 1:** Cloud Overlap Estimate.

**input**: New LiDAR cloud $C_{\text{read}}$ and submap cloud $C_{\mathcal{S}_{\text{ref}}}$,
**output**: Cloud overlap ratio $R_{\text{point,read}}$
**begin**
  Voxel filter $C_{\text{read}}$ and $C_{\mathcal{S}_{\text{ref}}}$ to resolution $r_{\text{filter}}$
  **for** Point $P_i \subset C_{\text{read}}$ **do**
    Search for $P_{\text{neighbour}} \subset C_{\mathcal{S}_{\text{ref}}}$ that is the closest to $P_i$
    **if** $||P_i, P_{\text{neighbour}}|| < \sqrt{3} \times r_{\text{filter}}$ **then**
      $N_{\text{point,overlap}} = N_{\text{point,overlap}} + 1$
    **end**
  **end**
  $R_{\text{point,read}} = N_{\text{point,overlap}} / N_{\text{point,read}}$
  return $R_{\text{point,read}}$
**end**

### 3.1. Cloud overlap estimate

Many systems such as [16,18] spawn submaps at fixed frequencies to bound the size of each submap as well as robot odometry drift within the submap. Our previous system [19] used a travel distance threshold to achieve the same purpose, under the assumption that the odometry drift is proportional to the distance travelled.

Cloud Overlap Estimate adds another trigger to spawn submaps based on point cloud overlap. Because our odometry system is based on Iterative Closest Point (ICP) [46], performance is affected by major changes in overlap such as when entering a new room [49]. Hence when traversing between two disconnected spaces via a narrow passage, i.e. the scenario presented in Fig. 3 (top), it is beneficial to spawn a new submap and create an elastic connection.

When the robot crosses through a doorway, there will be a significant degree of the scene which changes — with LiDAR beams blocked by the doorway. Traversing through any narrow constriction with a LiDAR will cause there to be low overlap between consecutive point cloud scans. In such a scenario, our proposed system will spawn a new submap when going through a doorway and create a new room, as demonstrated in Fig. 1.

Alg. 1 presents how we compute the overlap ratio $R_{\text{point,read}}$ between the point cloud of a new scan $C_{\text{read}}$ and the accumulated submap cloud $C_{\mathcal{S}_{\text{ref}}}$ of a reference submap $\mathcal{S}_{\text{ref}}$. Both $C_{\text{read}}$ and $C_{\mathcal{S}_{\text{ref}}}$ are filtered to the same resolution $r_{\text{filter}}$ for uniformity in overlap estimation. Points from $C_{\text{read}}$ that are within a thresholded distance from their closest points in $C_{\mathcal{S}_{\text{ref}}}$ are considered as overlapping points. The threshold is set to $\sqrt{3} \times r_{\text{filter}}$ — the diagonal of a cube with length $r_{\text{filter}}$.

If $C_{\text{read}}$ shares sufficient overlap ($R_{\text{point,read}} > 0.6$) with the accumulated submap cloud $C_{\mathcal{S}_{\text{ref}}}$, the new scan is integrated into a submap $\mathcal{S}_{\text{ref}}$, and the accumulated submap cloud $C_{\mathcal{S}_{\text{ref}}}$ grows by adding $C_{\text{read}}$. For instance, in Fig. 4(a) and (b), node $\mathcal{L}_{17}$ is integrated into submap $\mathcal{S}_5$, and LiDAR scan $C_{17}$ is accumulated into the submap cloud $C_{\mathcal{S}_5}$. However, we constrain point cloud accumulation of each submap by not combining accumulated submap clouds together during submap fusion. In Fig. 4, the volumetric occupancy submaps $\mathcal{S}_0$ and $\mathcal{S}_5$ in (c) are fused together into one submap $\mathcal{S}_0'$ in (d), but submap cloud $C_{\mathcal{S}_5}$ is not combined with $C_{\mathcal{S}_0}$. This prevents these submap clouds from growing indefinitely as exploration continues.

The criterion ($R_{\text{point,read}} > 0.6$) for spawning a new submap is based on the intuition that 50% of the point cloud is inside and outside of a room when the LiDAR is perfectly in the doorway. In our experiments, setting the threshold at 60% means a more conservative submap spawning behaviour in which the reconstruction becomes more elastic when the reliability of ICP in

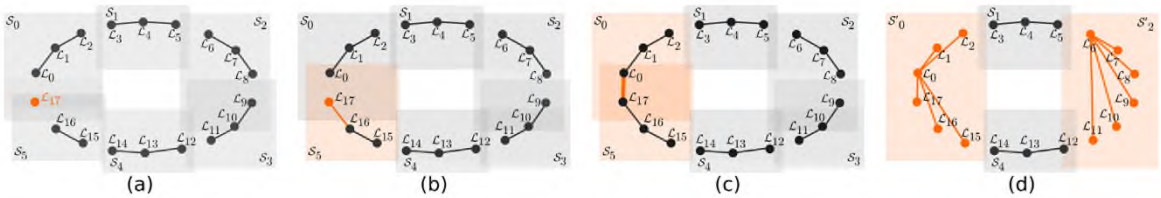



**Fig. 4.** An example of scan integration and submap fusion. **(a)** There are 17 LiDAR scans $\mathcal{L}_{0:16}$ (denoted by ●) in the existing reconstruction. The scans are clustered into 6 submaps $\mathcal{S}_{0:5}$, shown as rectangles (■) which represent the AABB of each submap. $\mathcal{L}_{17}$ (●) is the latest new LiDAR scan, and there have been no loop closures. **(b)** *Graph Clustering* allocates $\mathcal{L}_{17}$ to submap $\mathcal{S}_5$, and the new scan passes the *Cloud Overlap Estimate* (Section 3.1). Then $\mathcal{L}_{17}$ is integrated into $\mathcal{S}_5$, expanding the AABB (as shown by ■). **(c)** There is a new loop closure edge (orange line —) given by the SLAM pose graph between $\mathcal{L}_0$ and $\mathcal{L}_{17}$. The head and tail of the loop closure connection define the submap overlap search range from $\mathcal{S}_0$ to $\mathcal{S}_5$ (shown in ■). **(d)** *Graph Clustering* proposes the fusion of submaps $\mathcal{S}_5$ and $\mathcal{S}_0$ into $\mathcal{S}'_0$. *Submap Overlap Estimate* (Section 3.2) proposes the fusion between submaps $\mathcal{S}_3$ and $\mathcal{S}_2$ in $\mathcal{S}'_2$. If both proposals pass the *Relative Uncertainty* criterion (Section 3.3), fusion is executed and all submap indices are updated accordingly. The AABBs of $\mathcal{S}_0$ and $\mathcal{S}_2$ are therefore expanded (shown by ■), but their accumulated submap clouds are not, as explained in Section 3.1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Algorithm 2:** Submap Overlap Estimate.

**input**: Pair of occupancy submaps $\mathcal{S}_{\text{read}}$ and $\mathcal{S}_{\text{ref}}$,
**output**: Submap overlap ratios $R_{\text{voxel,read}}$ and $R_{\text{voxel,ref}}$
**begin**
  **for** Voxel $V_{\text{read}} \subset \mathcal{S}_{\text{read}}$ **do**
    Find $V_{\text{ref}} \subset \mathcal{S}_{\text{ref}}$ at the same coordinates as $V_{\text{read}}$
    **if** $V_{\text{ref}}$ *is not unknown* **then**
      **if** *Both* $V_{\text{ref}}$ *and* $V_{\text{read}}$ *are free or occupied* **then**
        $N_{\text{voxel,overlap}} = N_{\text{voxel,overlap}} + 1$
      **end**
    **end**
  **end**
  $R_{\text{voxel,read}} = N_{\text{voxel,overlap}}/N_{\text{voxel,read}}$
  $R_{\text{voxel,ref}} = N_{\text{voxel,overlap}}/N_{\text{voxel,ref}}$
  return $R_{\text{voxel,read}}$ and $R_{\text{voxel,ref}}$
**end**

odometry and SLAM is low due to limited Cloud Overlap Estimate (e.g. corridors, stairs). Increasing the threshold further leads to a higher memory cost. A threshold lower than 50 % could affect the functionality of room segmentation in the proposed system.

### 3.2. Submap overlap estimate

Submap fusion merges existing submaps, and reduces the memory usage of the overall system by fusing repeated reconstructions of the same physical space together. Our previous pipeline [19] triggered submap fusion using the loop closures detected in the SLAM system. However, it only merged submaps that were created when the robot travelled very close to a previous pose, i.e. the submaps at the head and tail ends of the loop closure.

In a large-scale (outdoor) environment, a long range (≈60 m) LiDAR sensor can repeatedly scan the same space from poses that are far away from one another, resulting in significant redundancy between submaps that loop closure fusion alone cannot address. Therefore we introduce an additional strategy for submap fusion based on the overlap of scanned spaces, utilising the explicit representation of free space in our volumetric occupancy map. This improves the reconstruction scalability when revisiting explored areas, in spite of the SLAM graph growing linearly.

Submap overlap is computed by comparing the volumetric reconstruction as well as the occupancy information stored in each individual submap. Fig. 3 (bottom) demonstrates such a case. Alg. 2 describes the estimation of voxel overlap between a pair of submaps $\mathcal{S}_{\text{read}}$ and $\mathcal{S}_{\text{ref}}$. This pair of submaps are fused together if either the ratio $R_{\text{voxel,ref}}$ or $R_{\text{voxel,read}}$ exceeds a configurable threshold $\lambda_{\text{fusion}}$. This threshold is empirically chosen as 0.7. It represents significant redundancy among submaps and does not require further tuning between experiments.

To ensure that the root poses of submaps are corrected by loop closure before fusion, we define a submap overlap search range using the head and tail of each loop closure. For example, in Fig. 4(c), SLAM loop closure is between $\mathcal{L}_0$ and $\mathcal{L}_{17}$, and they belong to $\mathcal{S}_0$ and $\mathcal{S}_5$, respectively. Hence the search range for Submap Overlap Estimate is $\mathcal{S}_{0:5}$. Candidate submap pairs for fusion are those that have an overlap greater than the threshold, e.g. $\mathcal{S}_2$ and $\mathcal{S}_3$.

Iterating through all voxels is a computationally intense process. Therefore, we add a conservative but efficient preliminary heuristic based on the AABB of each submap before computing submap voxel overlap. Using the AABBs of $\mathcal{S}_{\text{ref}}$ and $\mathcal{S}_{\text{read}}$, we compute the volumetric overlap percentages $\{R_{\text{aabb,ref}}, R_{\text{aabb,read}}\}$ and compare them with $\lambda_{\text{fusion}}$. If both AABB overlaps are smaller than the threshold, such as $\mathcal{S}_1$ and $\mathcal{S}_4$ in Fig. 4(c), the proposed system skips computing voxel overlap.

### 3.3. Relative uncertainty

To stay memory efficient, our system discards the individual submaps after fusing them into a parent. However, since each submap is internally rigid, local consistency is essential within the submaps. To improve submap fusion reliability and, consequently, retain global consistency, we propose a strategy to keep the errors within each submap bounded.

Our approach uses the relative uncertainty between the root poses of the candidate submaps to measure fusion confidence. To compute the relative uncertainty, we use the Hessian matrix of the SLAM problem to extract a joint covariance of the root poses of submaps. Then we use the approach of Mangelson et al. [10] to determine a single covariance matrix encoding the relative uncertainty between the poses. Please refer to Appendix for our derivation, which is consistent with the convention used in the GTSAM library. Lastly, we compute the eigenvalues of the relative uncertainty and apply a threshold $\lambda_{\text{uncertainty}}$ on them to finally decide if the fusion is accepted.

As an illustration, Fig. 4 presents the case of fusing two pairs of submaps, namely $\mathcal{S}_0$ and $\mathcal{S}_5$, and $\mathcal{S}_2$ and $\mathcal{S}_3$. In the example of fusing $\mathcal{S}_0$ and $\mathcal{S}_5$, we first compute the relative uncertainty between the root poses of $\mathcal{S}_0$ and $\mathcal{S}_5$, which are $\mathcal{L}_0$ and $\mathcal{L}_{15}$.

For $\mathbf{SE}(3)$ transformations ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_0}, {}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_5} \in \mathbb{R}^6$, the relative uncertainty $\mathbf{\Sigma}_{\mathcal{S}_0\mathcal{S}_5}$ is a $6 \times 6$ matrix, and there are 6 eigenvalues — 3 for translation and 3 for rotation. We compare the 3 translation eigenvalues against the configurable threshold $\lambda_{\text{uncertainty}}$. If one of the eigenvalues exceeds the threshold, the fusion between $\mathcal{S}_0$ and $\mathcal{S}_5$ is rejected. We set $\lambda_{\text{uncertainty}}$ to be 0.2 m in our experiments based on our SLAM noise model, which is not subject to change between experiments.

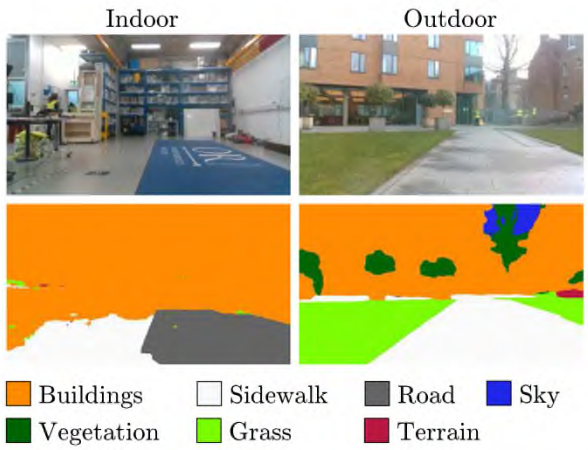



**Fig. 5.** Representative images of the environment in which the robot was operated (top row) and the corresponding results of the semantic segmentation network (bottom row).

### 3.4. Semantic segmentation

The robot platform in our experiments (Boston Dynamics Spot) carries 6 cameras in total, covering the full 360° horizontally. Dense per-pixel semantic labels are predicted for each camera image using a deep neural network. Our neural network follows the architecture presented in [12], but adapts the UNet with a MobileNet backbone [50], pre-trained on the ImageNet [51] dataset and fine tuned on the Extended NCLT dataset [52]. Representative results of the semantic segmentation network are presented in Fig. 5. The network was trained only on images of outdoor environments but estimates meaningful labels for indoor environments — walls are classified as buildings and floor as either sidewalk or road.

We use the semantically-labelled images to detect transitions of the mobile robot between indoor and outdoor environments. We first categorise all semantic labels into whether they are typically and distinctly *outdoor* or not. Class labels such as vegetation, grass, water, terrain etc. belong to the former category. We analyse the distribution of these predicted outdoor labels in the images. For each image, we compute the percentage of pixels with outdoor labels amongst the whole image. The proposed system then combines this with LiDAR range measurements to detect whether the robot platform is indoor or outdoor. If the percentage of outdoor labels is consistently larger than an empirically determined threshold for a set of successive frames, and there are long range LiDAR returns, the robot is considered to be outdoors.

When the robot is exploring outdoor environments, our proposed system integrates semantic information into each submap. Submaps that are created when the robot is indoors are all given the indoor class label.

In our framework, we predict semantic labels for images from all cameras mounted on the robot to achieve a panoramic view of the surroundings. To integrate semantic labels into the dense 3D reconstruction created from LiDAR, the proposed system iterates through voxels in each submap, and by using the tracked camera pose of $j$th view ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{V}_j}$, transforms the coordinate of a voxel ${}^{\mathcal{M}}x \in \mathbb{R}^3$ into the camera's frame to compute a pixel coordinate $u_{j,x} = \pi\left({}^{\mathcal{M}}\mathbf{T}_{\mathcal{V}_j}^{-1}{}^{\mathcal{M}}x\right)$, where $\pi$ represents the camera's projection operation using its intrinsic matrix.

The pixel coordinate $u_{j,x}$ will then result in a categorical distribution representing the probability of this pixel belonging to a semantic class. This distribution is then stored in the voxel if latter has not yet been labelled by any semantic classes, or used to update the existing distribution in the voxel. The formulation to update voxel probability distributions is as follows. Let $\mathbf{I}_j$ denote the semantic information of this image, each pixel contains a probability distribution across all $M$ class labels $P(Z_u = l_m|\mathbf{I}_j)$. We use the recursive Bayesian model presented in [44] to update the probabilistic distribution within voxel $x$:

$$P(l_m|\mathbf{I}_{0,\dots,j}) = \frac{1}{\alpha}P(l_m|\mathbf{I}_{0,\dots,j-1})P(Z_{u_{j,x}} = l_m|\mathbf{I}_j), \tag{1}$$

where $\alpha$ is a normalising factor.

For each new semantically segmented image input into the proposed system, each voxel will receive at most one update to the probability distribution stored in it, multiple voxels can be labelled by the same image pixel. However, the robot platform used in our experiments carries 6 cameras, which have overlapping frustums. As the robot continues its exploration, images streaming from even a single camera overlap. Eq. (1) provides a mathematically sound model to continuously update the probability distribution in each voxel, and to address any conflicting labelling from different observations. In addition, upon loop closure and submap fusion, different distributions in overlapping voxels from multiple submaps are resolved using the same model. The computational complexity of this operation is $O(M \cdot J \cdot V)$ for a submap associated with $J$ camera views that contains $V$ occupied voxels.

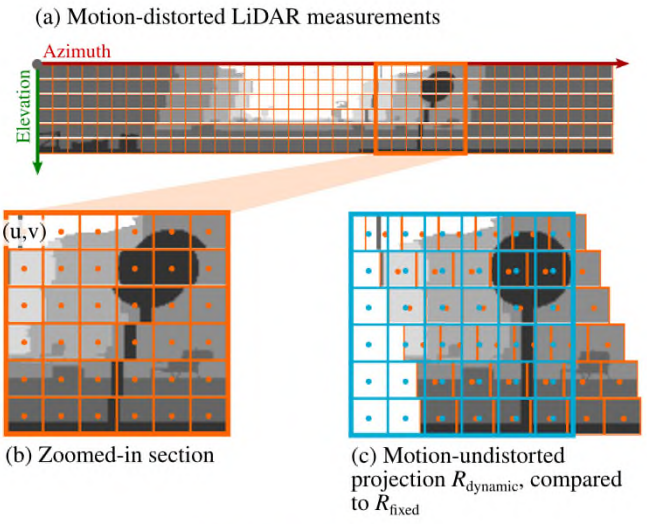
### 3.5. Motion aware LiDAR integration

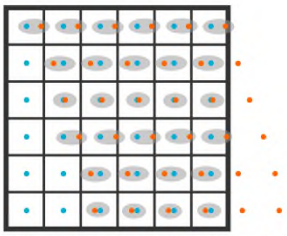
Modern 3D LiDAR sensors collect range returns continuously over time. A complete 360° scan is accumulated and then made available to downstream applications. The high dynamics of a legged robot, especially in rotation, leads to motion distortion of these scans and imprecise maps, if uncorrected (Fig. 6 (a) and (b)).

As the input to our proposed reconstruction system, the external odometry module incorporated a LiDAR motion undistortion component.[3] This component first uses an odometry input with higher frequency, such as from Inertia Measurement Unit (IMU) or legged odometry, to interpolate the sensor pose between consecutive LiDAR measurements (at 10 Hz). It then computes the direction and origin of each LiDAR beam on-the-fly and reprojects it as if measured instantaneously, correcting the motion-induced distortion. While the undistorted point cloud is necessary for an accurate reconstruction, the undistortion module is not within the scope of this work and therefore not assessed in the presented experiments. Instead, the proposed system focuses on preserving the motion undistortion effect in the reconstruction. The motion undistortion procedure in the odometry module results in non-uniform elevation and azimuth angles for each point, violating the uniform grid assumption of the LiDAR projection model that our reconstruction pipeline [9,19] relies on (Fig. 6(c)).

This assumption of LiDAR beams being uniformly distributed in angular space is beneficial for the efficiency of scan integration. In the reconstruction core of the proposed system, a voxel's coordinates in the map frame ${}^{\mathcal{M}}x \in \mathbb{R}^3$ are projected into the LiDAR frame ${}^{\mathcal{L}}x = {}^{\mathcal{M}}\mathbf{T}_{\mathcal{L}}^{-1}{}^{\mathcal{M}}x$ and normalised into a unit vector that represents its direction, before being converted into a pair of column and row indices $(u, v)$ for depth measurement look-up and occupancy update. With the uniform grid assumption, this conversion from ${}^{\mathcal{M}}x$ to $(u, v)$ is a simple arithmetic process, the complexity of which is $O\left(L^3 \cdot r^{-3}\right)$ — $L$ represents the LiDAR sensing range and $r$ represents the voxel resolution. However, with motion undistortion dynamically adjusting the elevation and

[3] The motion undistortion method was inspired by this package - https://github.com/ethz-asl/lidar_undistortion.



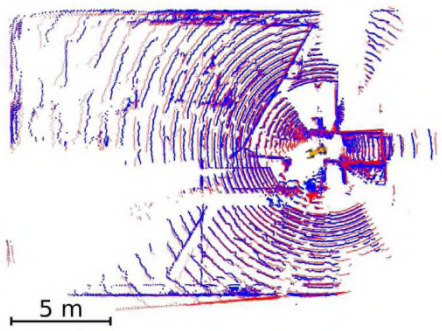




| u | u+1 | u+2 | u+3 | u+4 | u+5 |
|---|---|---|---|---|---|
| NaN | u | u+1 | u+2 | u+3 | u+4 |
| NaN | u | u+1 | u+2 | u+3 | u+4 |
| NaN | u | u+1 | u+2 | u+3 | u+4 |
| NaN | NaN | u | u+1 | u+2 | u+3 |
| NaN | NaN | u | u+1 | u+2 | u+3 |

| v | v | v | v | v | v |
|---|---|---|---|---|---|
| NaN | v+1 | v+1 | v+1 | v+1 | v+1 |
| NaN | v+2 | v+2 | v+2 | v+2 | v+2 |
| NaN | v+3 | v+3 | v+3 | v+3 | v+3 |
| NaN | NaN | v+4 | v+4 | v+4 | v+4 |
| NaN | NaN | v+5 | v+5 | v+5 | v+5 |

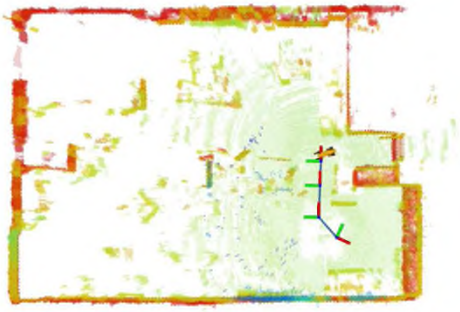
(e) Resulting $U$-$V$ sets to map coordinates in $R_{\text{dynamic}}$ to $R_{\text{fixed}}$

**Fig. 6.** Motion aware LiDAR integration module to remap LiDAR points during dynamic motions. **(a)** – Motion-distorted LiDAR scan showing azimuth and elevation. **(b)** – Zoom in of a section. Each circle ● indicates a ray, with its corresponding cell in the grid. **(c)** – After motion undistortion, the undistorted projection $R_{\text{dynamic}}$ (in orange ●) does not match the uniform grid (in light blue ●). **(d)** We iterate over the undistorted rays to find the closest ray assuming an uniform grid. **(e)** The output of the association are adjusted coordinates for azimuth ($U$) and elevation ($V$) angles of the undistorted projection. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)




(a) Motion distorted raw LiDAR point cloud (red) and the undistorted version (blue).

(b) Reconstruction without considering correcting the dynamic projection model in undistortion, causes misalignment.



(c) Corrected reconstruction via motion undistortion.

**Fig. 7.** Proposed motion aware LiDAR integration method (Section 3.5) remaps a dynamic projection model, improving the reconstruction – particularly when turning sharply. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

azimuth angles of each LiDAR beam in every scan, to look up a corresponding depth measurement in the direction of a voxel will require a search through neighbouring LiDAR beams to find the closest ray. The complexity therefore becomes $O\left(L^3 \cdot r^{-3} \cdot n\right)$ where $n$ represents the number of neighbours that need to be searched. Given the long LiDAR sensing range and the fine voxel resolution, this approach quickly becomes infeasible.

Instead, we designed a method to remap the motion undistorted LiDAR scans, which allows us to preserve not only measurement accuracy, but also the efficiency of the projection model used by our reconstruction pipeline. For each LiDAR scan of constant height $h$ and width $w$ ($64 \times 1024$ in our experiments), let $R_{\text{fixed}}$ be the default set of ray-casting directions according to the sensor's specification, and $R_{\text{dynamic}}$ be the unit vectors for all motion-undistorted LiDAR beams. For each ray $r_{uv} \in R_{\text{fixed}}$ at row $v$ and column $u$, we then search in its neighbours to find the $\hat{r}_{uv} \in R_{\text{dynamic}}$ that has the smallest angular difference $\theta_{\text{diff}}$ from $r_{uv}$ to establish a match (Fig. 6(d)). The size of the search region is determined by the resolution of the LiDAR and the characteristics of the robot, such as its rotation rate. We can also terminate the search if $\theta_{\text{diff}} < \theta_{\text{thres}}$ to further speed up the procedure; we used $\theta_{\text{thres}} = 0.001$ rad, corresponding to 5 cm voxel resolution at a maximum sensor range of 50 m.

As a result of the remapping, we find corrected coordinates for each ray in the undistorted scan, given by the sets of indices $U = u_{(0,0)} \ldots u_{(h-1,w-1)}$ and $V = v_{(0,0)} \ldots v_{(h-1,w-1)}$, where NaN values are used for the unmatched rays, as shown in Fig. 6(e). The $U - V$ sets are then input into the *supereight* reconstruction pipeline. We further update the projection model of *supereight* to find the corresponding motion-undistorted LiDAR range measurement

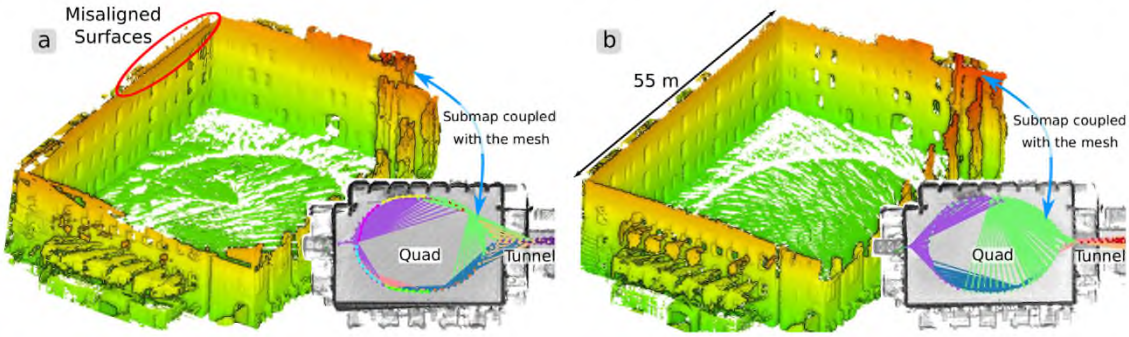



**Fig. 8. Exp** 4.1 – The proposed spatial overlap analysis improves the global consistency in the reconstruction. Figure **(a)** and **(b)** present the mesh and volumetric representations of NCD Long experiment, created by the baseline and the proposed system, respectively. The volumetric representations are shown in grey overlaid with submap clusters. The mesh representations are created from the green submaps. Using spatial understanding of the environment leads to more reliable submap fusion and therefore better alignment. The mesh created by the previous pipeline has misaligned double surfaces while the proposed system improves the consistency in the mesh. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

based on indices in $U$ and $V$ for each ray-casting vector in $R_{\text{fixed}}$ for LiDAR integration. The complexity of this remapping process is $O(h \cdot w \cdot n)$, and is not dependent on the LiDAR range or the voxel resolution and therefore more efficient.

To demonstrate the effect of this particular module, we conducted a brief experiment (Fig. 7) where we teleoperated a legged robot around our indoor lab space. At times of high rotation rate obvious motion distortion in the scan can be seen in Fig. 7(a). The distorted scan (red) is misaligned along the lower wall, but the odometry system accounted for the distortion and produced the undistorted scan (blue).

When moving to occupancy mapping, we can see that without properly addressing the dynamic projection model the default *supereight* misaligns surfaces in its reconstruction (Fig. 7(b)). By applying the method described above, the reconstruction is corrected as shown in Fig. 7(c), keeping its computational efficiency.

## 4. Experiments and evaluation

In this section we present the series of experiments that the proposed system has been assessed with, as shown in Fig. 1. These experiments include:

- Exp 4.1 (NCD Long):
  A large-scale outdoor experiment with a handheld device in the Newer College Dataset (NCD) [53].
- Exp 4.2 (ORI):
  A multi-storey multi-room exploration experiment in the Oxford Robotics Institute (ORI) with a Boston Dynamics Spot robot.
- Exp 4.3 (Indoor–outdoor):
  An exploration through multiple storeys and rooms and between indoor and outdoor environments.
- Exp 4.4 (Simulation):
  Looping explorations in a small and a large room network using the Gazebo simulator.

In these experiments, we evaluate the improvement in system scalability and global consistency brought about by the proposed submap spawning and fusion strategies, compared with our previous reconstruction pipeline [19] as the baseline. We also demonstrate the capability of distinguishing between different surroundings and labelling volumetric LiDAR reconstruction using semantic information, while maintaining global consistency upon SLAM loop closure.

Table 1 gives details of the different LiDAR sensors used in these experiments. The LiDAR sensors produce organised point cloud scans of 64 × 1024 points at 10 Hz. The SLAM system creates a node in its pose graph every 2 m travelled when exploring.

**Table 1**
LiDAR sensors used in the experiments and their properties. VFoV: Vertical Field of View; HFoV: Horizontal Field of View.

| Experiment | Section | LiDAR properties | | | |
|---|---|---|---|---|---|
| | | Model | VFoV | HFoV | Max range (m) |
| NCD Long (handheld) | 4.1 | Ouster OS1–64 | 33.2° | 360° | 120 |
| ORI (Spot) | 4.2 | Ouster OS0–64 | 90° | 360° | 50 |
| Indoor–outdoor (Spot) | 4.3 | Ouster OS0–64 | 90° | 360° | 50 |
| Simulation (UGV) | 4.4 | Ouster OS0–64 | 90° | 360° | 50 |

The proposed system and the baseline integrated the LiDAR scans at each SLAM node using the *MultiresOFusion* mode in [19] for volumetric occupancy reconstruction. The voxel resolution used in these experiments was 6.5 cm and we integrated LiDAR ranges between 0.5 m and 60 m. These settings give high resolution and long range while retaining 3 Hz integration.

Surface mesh representations of the reconstruction, for example Fig. 8, were created by applying the Marching Cubes algorithm [54] on the zero-crossings of occupancy.

### 4.1. Large-scale outdoor handheld experiment

In this section we present the performance of the proposed system and the baseline when tested using NCD Long. This dataset consists of a 2.2 km exploration over 44 min in a 135 × 122 $m^2$ environment. The principled clustering strategies in the proposed system improve the consistency of the global reconstruction, as shown in Fig. 8. The proposed system also demonstrates improved scalability in memory usage compared with the baseline.

#### 4.1.1. Reconstruction accuracy

Fig. 8(a) shows the mesh and volumetric reconstructions created by the baseline. In particular the green submap in the Quad area was created as a rigid fusion of several submaps after multiple loop closures were established. As shown in the birds-eye view next to the mesh representation, this submap contains scans taken both in the Quad and the Tunnel. These scans have limited overlap with one another and thus registration between them is unreliable. This led to a duplicate reconstruction of the indicated wall of the Quad.

By introducing our proposed submap fusion algorithms, the reconstruction's consistency is improved as shown in Fig. 8(b). The Cloud Overlap Estimate strategy makes better decisions when the handheld device travels between the Quad and the Tunnel — maintaining elastic connections between these two spaces in the global reconstruction. The measurement of relative uncertainty

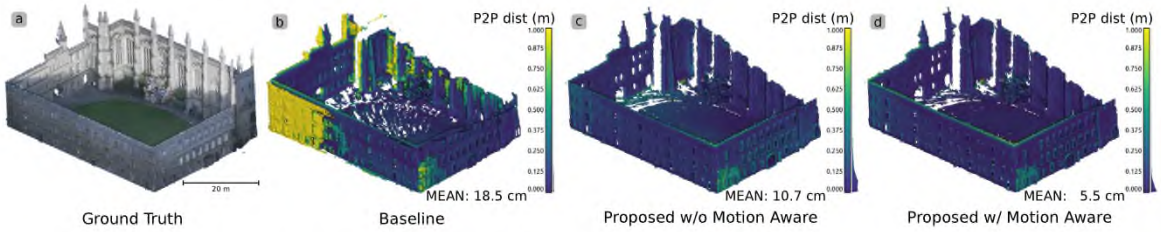



**Fig. 9. Exp** 4.1 – The comparison between **(a)** the ground truth map of NCD Long experiment and each reconstruction created by **(b)** the baseline, **(c)** the proposed system without motion aware LiDAR integration and **(d)** the proposed system with motion aware LiDAR integration. Colours indicate point-to-point distances (P2P dist) between each reconstruction and the ground truth, and the distributions are also presented beside the colour bar. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

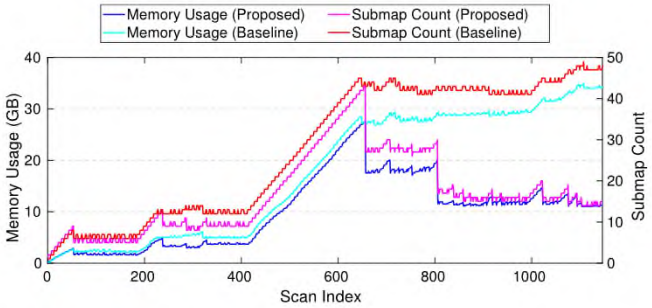



**Fig. 10. Exp** 4.1 – The memory usage and submap counters of the proposed system and the baseline system in the NCD Long experiment.

also rejects unreliable submap fusions. The global volumetric map and the submap mesh both demonstrate improved accuracy in surface alignment.

The two submap reconstructions presented in Fig. 8 were also compared with the ground truth point cloud provided in the dataset [53]. We used CloudCompare[4] to sample dense point clouds from both meshes, align the reconstructed point clouds with the ground truth, and compute the point-to-point distance error between them. As an ablation study, we include both the result with and without the motion aware LiDAR integration module. The results of a quantitative evaluation are presented in Fig. 9, together with the average point-to-point error of each reconstruction.

In Fig. 9(b), misaligned surfaces created by the baseline method resulted in highly erroneous (>1 m) points behind existing ones, which are coloured yellow according to the colour map. With Cloud Overlap Analysis, the proposed system improved surface alignment and in turn the average point-to-point distance error (Fig. 9(c)).

By adding the motion aware LiDAR integration module, the accuracy of the reconstruction is further improved, as presented in Fig. 9(d). The floor is better aligned with the ground truth, and more points have <5 cm errors.

#### 4.1.2. Memory consumption

In the proposed system, submaps that have significant overlapping scan volumes are fused according to their Submap Overlap Estimate. This is in addition to submap fusion based on loop closures, and properly allows the map to scale with the size of the environment rather than the length of the exploration. Fig. 10 presents the memory usage and submap counter of the proposed system compared to the baseline in NCD Long. We compute the memory usage by summing the size of allocated memory for each submap's octree in RAM.

[4] https://www.danielgm.net/cc/.

In our previous work [19], the baseline method was compared against two state-of-the-art reconstruction pipelines, i.e. OctoMap [15] and Voxgraph [18], and demonstrated its advantage of efficient reconstruction memory usage.

In the baseline method the submaps can only be merged when loop closures occur which causes memory usage to grow over time. In contrast, our proposed approach can merge spatially overlapping volumes such that the number of submaps can plateau. For the NCD Long experiment, submap count stabilised at about 30 submaps when the entire environment has been explored at scan 650. Memory usage actually decreased after scan 650 to ∼18 GB while maintaining the 6.5 cm resolution reconstruction. By the end of the experiment, there was a 65 % reduction in memory usage compared to the baseline.

### 4.2. Multi-storey multi-room indoor exploration

In the ORI experiment a quadruped robot Spot explored three floors of a typical university research lab (Fig. 11); we show the reconstruction of every floor in the building in Fig. 12.

New submaps were spawned when the robot entered or exited rooms because the proposed system detected a decrease in cloud overlap. Spatial overlap analysis then merged overlapping submaps in each room, creating a unique reconstruction for each enclosed space. It further ensured that these submaps remained independent allowing future SLAM loop closures to re-position the room submaps as needed. In the supplementary video we demonstrate the full experiment and the incremental mapping of the building.

Segmenting rooms on the fly allows real-time applications such as path planning and obstacle avoidance to consider only the minimal submaps rather than the entire global reconstruction. The proposed system can thus improve the scalability of other applications.

### 4.3. Indoor–outdoor transition experiment

To demonstrate semantically annotated mapping, we conducted an experiment with the robot moving from indoors to an outdoor environment. The goal was to use the predicted semantic labels to determine the indoor/outdoor state.

Semantic classes such as vegetation, grass, water and terrain are characteristic of outdoor environments. Using the distribution of predicted labels in each image, we computed the percentage of labels belonging to such *outdoor* classes to determine if the robot was indoors or outdoors at a given time. If for a set of successive frames, the percentage of outdoor classes is larger than an empirically determined threshold (here 20 %) and there are long range LiDAR returns (>30 m) we determine that the robot is in an outdoor environment. Relying on LiDAR data makes the system robust to occasional incorrect predictions by the segmentation network. While this could have been done in a more involved

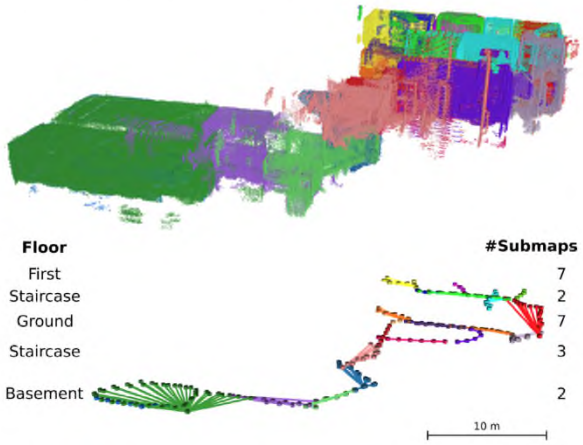



**Fig. 11. Exp** 4.2 – The proposed system was tested in a multi-floor multi-room indoor environments. **Top**: The reconstruction result where each room is segmented by a submap indicated by a unique colour; **Bottom**: The clustered pose graph nodes using spatial overlap analysis on each floor. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

way, we found that this simple approach worked well in our experiments.

We used this knowledge of the environment to alter the length (∼size) of the submaps; 5 m for indoors and 8 m for outdoors. This helps reduce the memory required in larger environments. Results of this experiment are shown in Fig. 13, where parts of the map that are determined to be indoors are coloured in grey and parts of the map that are outdoors are coloured using the semantic labels.

In addition, Fig. 14 demonstrates the elasticity granted by submaps. Fig. 14(a) shows the semantically annotated LiDAR reconstruction before SLAM loop closure. The enlarged view highlights a misalignment on the facade of a building, labelled as orange voxels. In Fig. 14(b), SLAM loop closure corrected the pose graph, which in turn realigned the walls in the zoomed-in view.

Fig. 15 presents the quantitative evaluation of reconstruction accuracy using the same method as the one described in Section 4.1.1. The average point-to-point distance error is 5.4 cm, with the most accurate points around the ground and the lower floors of buildings. For the robot to scan the upper floors of the building it must do so from further away which causes there to be slightly higher error — around 10 cm. The highest errors (>50 cm, coloured yellow) are the dynamic objects in the scene, e.g. the robot operator.

### 4.4. Room networks in simulation

To test the performance for long term operation missions, we carried out two experiments in the Gazebo simulator with a wheel robot carrying a simulated 3D LiDAR in the environments in Fig. 16.

In the small room network, the mission looped around the environment three times in both directions. As shown by the clustered poses in Fig. 17(a) there is a clear division between submaps at each doorway. Each room was constructed with either one or two submaps — even after multiple revisits. Fig. 17(b) demonstrated an approximately 700 m exploration in a section of the large room network. Similarly, the room in the middle only contains two major submaps even after about 10 loops with different routes. Scans along corridors were also fused together based on overlap and loop closures, creating only 10 submaps in the end.

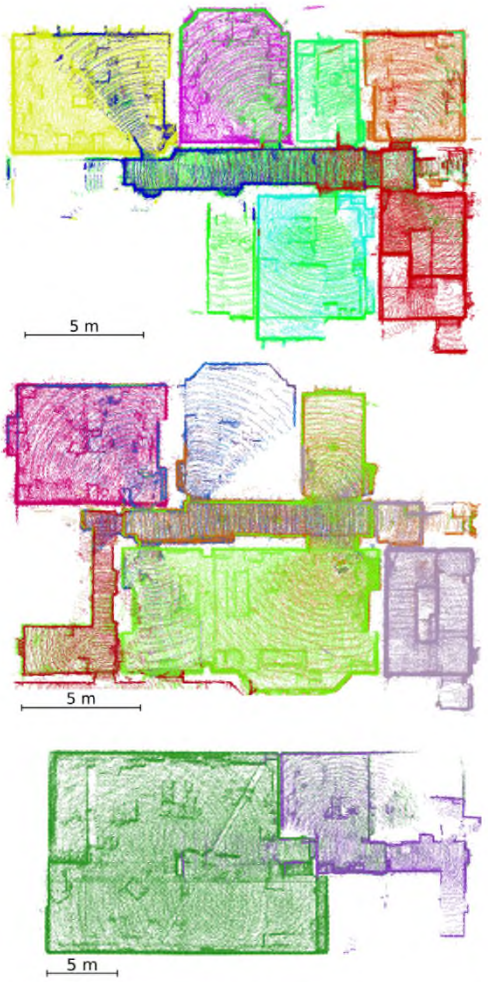



**Fig. 12. Exp** 4.2 – The volumetric submap reconstructions of each floor (**Top:** first; **Middle:** ground; **Bottom:** basement) in ORI, with each room segmented into unique submaps by the proposed system on the fly during exploration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 18 shows that the submap count and memory usage in both simulation experiments plateaued as the environments were repeatedly scanned. Overall, the proposed method decreased the memory usage of both experiments by 50% compared to the baseline method even after extensive exploration and revisiting, such as the experiment in the large room network (Fig. 17(b)).

## 5. Conclusion and future work

This paper introduced a set of principled strategies that leverage spatial overlap analysis to improve the spawning and fusion of 3D reconstruction submaps. The proposed system can merge together submaps that scan the same space, allowing the memory consumption of our reconstruction to scale with the volume of the explored area rather than the duration of operation. Additionally, we improved submap fusion reliability by directly computing relative uncertainty between pairs of poses in the SLAM pose graph. These strategies delay submap fusion in 3D reconstruction until there is sufficient confidence to improve global consistency of the overall map.

Furthermore, our system can segment individual rooms on the fly as a robot explores a room network by using spatial overlap. This feature can be beneficial to planners exploring room networks such as in the DARPA SubT challenge.

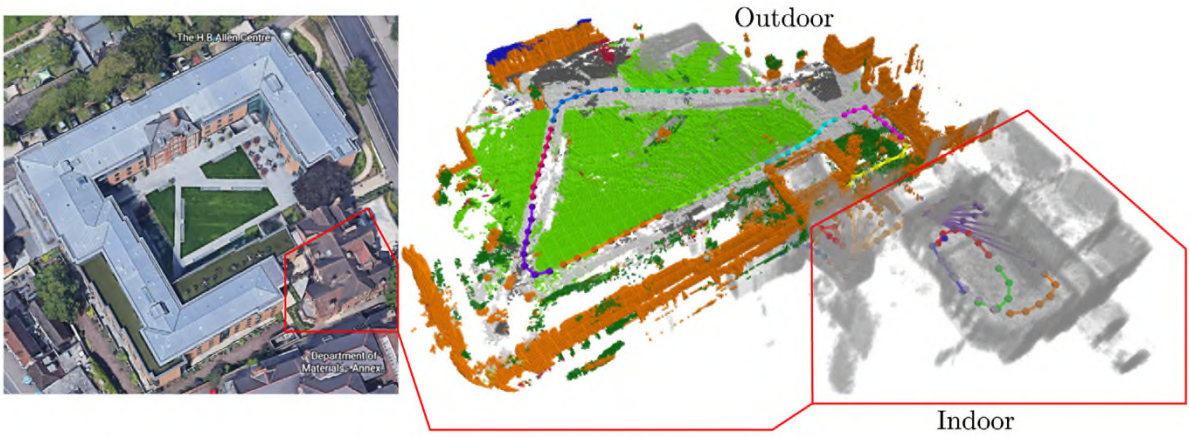



**Fig. 13. Exp** 4.3 – Environment Classification: Parts of the constructed map that are classified as indoors (bottom right) are coloured in grey and parts of the map that are outdoors are coloured by their semantic labels. Lengths of the submaps (number of nodes) are also different between the two regions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

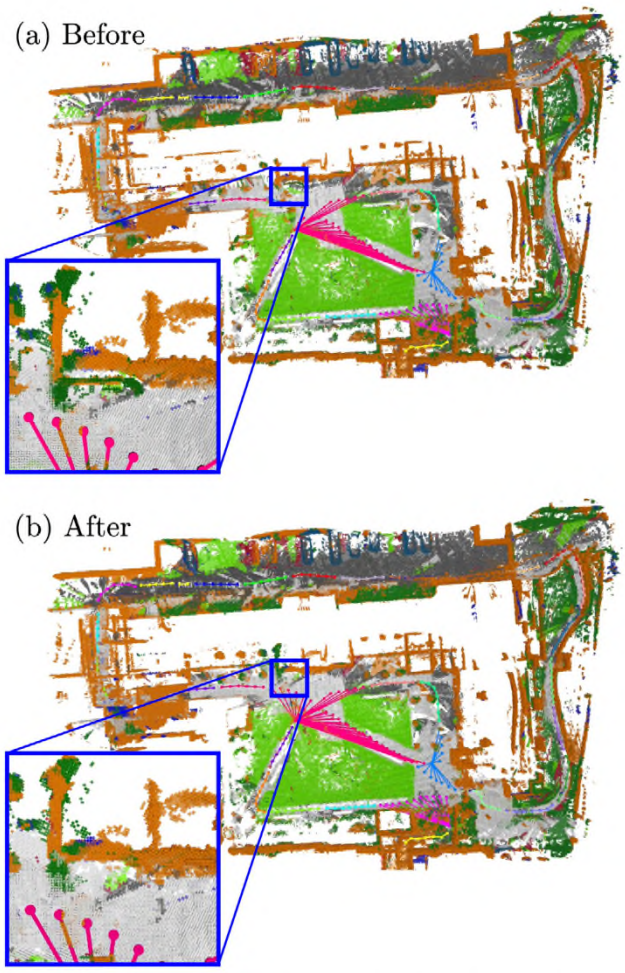



**Fig. 14. Exp** 4.3 – Drift correction: **(a)** – the semantically annotated LiDAR reconstruction with building misalignment (orange) before loop closure; (b) – the elastic reconstruction after loop closure correcting the SLAM pose graph. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

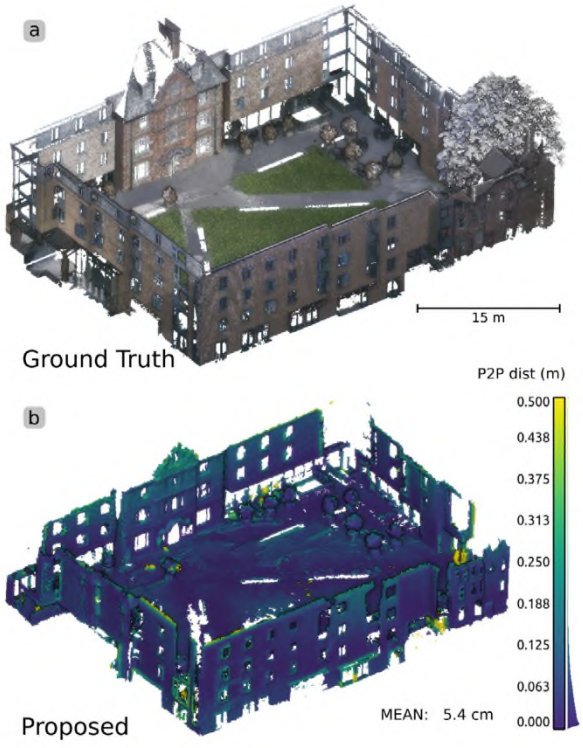



**Fig. 15. Exp** 4.3 – Reconstruction of the proposed system **(b)** compared with the ground truth **(a)**. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

By introducing semantic segmentation into the pipeline, our system can annotate a volumetric LiDAR reconstruction using semantic labels while maintaining elasticity using submaps. Semantic information allows our system to detect the change in surroundings when transitioning between indoor and outdoor environments, enabling adjustments to mapping parameters on the fly. The classification of each voxel is probabilistically updated during exploration and submap fusion, ensuring the consistency of semantic information across submaps.

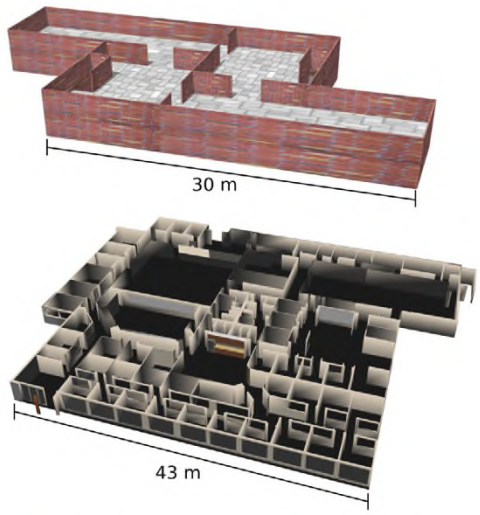
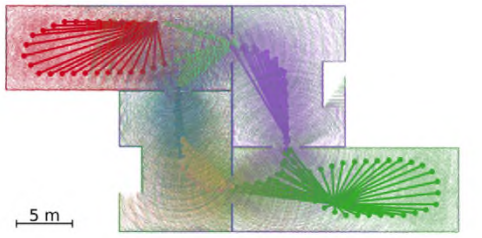
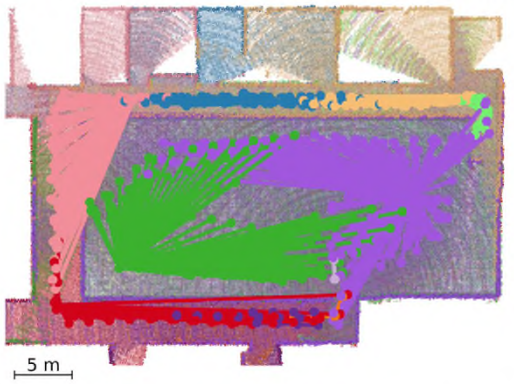



**Fig. 16. Exp** 4.4 – The Gazebo environments of a small (top) and a large (bottom) room network for experiments in simulation.



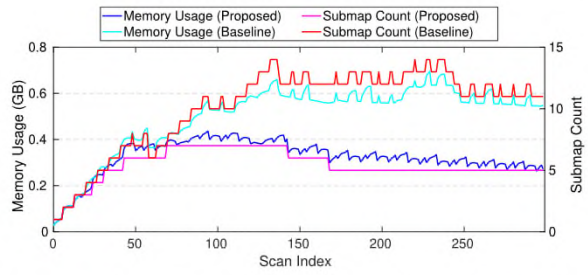

(a) Submap reconstruction in the small room network



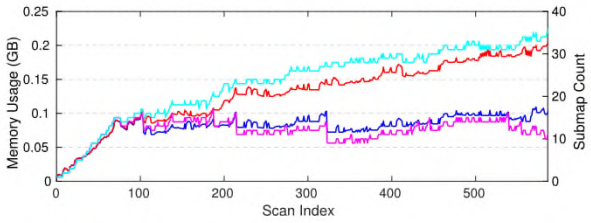

(b) Submap reconstruction in the large room network

**Fig. 17. Exp** 4.4 – The proposed system segments individual rooms and fuses redundant submaps during simulation experiments.

In future work, we intend to improve the accuracy of annotating LiDAR map using semantic information, especially around the boundary of objects. Calibration and synchronisation between LiDAR and multiple cameras can benefit the direct projection of measurements between them. Frame-to-frame alignment across camera measurements can also reduce the impact of robot's motion to semantic segmentation. Furthermore, we would like to expand the capability of our semantic segmentation module. In particular, indoor semantic labelling will assist robot's exploration, such as autonomously inferring floor plans from a multi-storey reconstruction and enabling high-level path planning between floors.




(a) The performance in the small room network (simulated)




(b) The performance in the large room network (simulated)

**Fig. 18.** Exp 4.4 – The memory usage and submap counters of the proposed system and the baseline system in simulated experiments.

## CRediT authorship contribution statement

**Yiduo Wang:** Conceptualization of this study, Methodology, Software, Writing. **Milad Ramezani:** Conceptualization of this study, Methodology, Software, Writing. **Matias Mattamala:** Conceptualization of this study, Methodology, Software, Writing. **Sundara Tejaswi Digumarti:** Conceptualization of this study, Methodology, Software, Writing. **Maurice Fallon:** Conceptualization of this study, Methodology, Software, Writing.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Maurice Fallon reports financial support was provided by Royal Society University Research Fellowship. Maurice Fallon reports financial support was provided by EPSRC ORCA Robotics Hub. Matias Mattamala reports financial support was provided by Chilean National Agency for Research and Development (ANID). Sundara Tejaswi Digumarti reports a relationship with The University of Sydney that includes: employment. Milad Ramezani reports a relationship with Commonwealth Scientific and Industrial Research Organisation that includes: employment.

## Acknowledgements

This research is supported by the EPSRC ORCA Robotics Hub, UK (EP/R026173/1). Maurice Fallon is supported by a Royal Society University Research Fellowship. Matias Mattamala is supported by the Chilean National Agency for Research and Development (ANID) DOCTORADO BECAS CHILE/2019-72200291. The authors would like to acknowledge the help of Prof. Dr. Stefan Leutenegger (TU Münich) and Dr. Marco Camurri in the design of the supereight motion aware LiDAR integration component of the proposed work. The authors would also like to thank Mikael Yeghiazaryan for his contribution to the semantic segmentation network and Benoit Casseau for his help with running the experiments on the Spot robot.

## Appendix. Computation of relative uncertainty in pose graph SLAM

To compute the relative uncertainty between any pair of nodes in our pose graph, we derived the following formula based on the conventions of GTSAM with reference to Mangelson et al. [10].

First we define the probability distribution of the relative transformation from submap $\mathcal{S}_i$ to node $\mathcal{S}_j$ as:

$$ {}^{\mathcal{S}_i}\mathbf{T}_{\mathcal{S}_j} = {}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_i}^{-1}\,{}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_j} \tag{2} $$

where the poses ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_i}$ and ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_j}$ indicate probability distributions on $\mathbf{SE}(3)$ following a right-hand composition:

$$ \mathbf{T} = \bar{\mathbf{T}}\,\mathrm{Exp}(\xi) \tag{3} $$

$\bar{\mathbf{T}}$ is the mean transformation of the distribution, and $\xi$ is a perturbation that follows a Gaussian distribution. In Eq. (2) we consider that the poses have covariances $\Sigma_{\mathcal{M}\mathcal{S}_i}$ and $\Sigma_{\mathcal{M}\mathcal{S}_j}$, respectively.

In order to derive the expressions for the relative uncertainty, we need the adjoint action of $\bar{\mathbf{T}}$ on $\xi$, denoted as $\mathrm{Ad}_{\bar{\mathbf{T}}}(\xi)$, defined as follows:

$$ \begin{aligned} \mathrm{Ad}_{\bar{\mathbf{T}}}(\xi) := \mathrm{Ad}_{\bar{\mathbf{T}}}\xi &= \mathrm{Log}(\bar{\mathbf{T}}\,\mathrm{Exp}(\xi)\bar{\mathbf{T}}^{-1}) \\ \mathrm{Exp}(\mathrm{Ad}_{\bar{\mathbf{T}}}\xi) &= \bar{\mathbf{T}}\,\mathrm{Exp}(\xi)\bar{\mathbf{T}}^{-1} \\ \bar{\mathbf{T}}^{-1}\mathrm{Exp}(\mathrm{Ad}_{\bar{\mathbf{T}}}\xi) &= \mathrm{Exp}(\xi)\bar{\mathbf{T}}^{-1} \\ \mathrm{Exp}(\xi)\bar{\mathbf{T}} &= \bar{\mathbf{T}}\,\mathrm{Exp}(\mathrm{Ad}_{\bar{\mathbf{T}}^{-1}}\xi) \end{aligned} \tag{4} $$

Expanding Eq. (2) using Eqs. (3) and (4):

$$ \begin{aligned} &{}^{\mathcal{S}_i}\bar{\mathbf{T}}_{\mathcal{S}_j}\mathrm{Exp}({}^{\mathcal{S}_i}\xi_{\mathcal{S}_j}) \\ &= \mathrm{Exp}(-{}^{\mathcal{M}}\xi_{\mathcal{S}_i})\,{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}^{-1}\,{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}\mathrm{Exp}({}^{\mathcal{M}}\xi_{\mathcal{S}_j}) \\ &= {}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}^{-1}\mathrm{Exp}(-\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}}\,{}^{\mathcal{M}}\xi_{\mathcal{S}_i})\,{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}\mathrm{Exp}({}^{\mathcal{M}}\xi_{\mathcal{S}_j}) \\ &= {}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}^{-1}\,{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}\mathrm{Exp}(-\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}}\,{}^{\mathcal{M}}\xi_{\mathcal{S}_i})\mathrm{Exp}({}^{\mathcal{M}}\xi_{\mathcal{S}_j}) \end{aligned} \tag{5} $$

Let ${}^{\mathcal{S}_i}\bar{\mathbf{T}}_{\mathcal{S}_j} \triangleq {}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}^{-1}\,{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}$, then we can establish the following equivalence:

$$ \mathrm{Exp}({}^{\mathcal{S}_i}\xi_{\mathcal{S}_j}) = \mathrm{Exp}(-\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}}\,{}^{\mathcal{M}}\xi_{\mathcal{S}_i})\mathrm{Exp}({}^{\mathcal{M}}\xi_{\mathcal{S}_j}) \tag{6} $$

The covariance of the perturbation on the left should be equal to the one on the right. However, we cannot compute the covariance directly because of the properties of the exponential map. Instead, we define ${}^{\mathcal{S}_i}\xi'_{\mathcal{S}_j} = -\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}}\,{}^{\mathcal{M}}\xi_{\mathcal{S}_i}$, and use the Baker–Campbell–Hausdorff (BCH) formula [55] up to first order:

$$ \begin{aligned} E[{}^{\mathcal{S}_i}\xi_{\mathcal{S}_j}\,{}^{\mathcal{S}_i}\xi_{\mathcal{S}_j}^T] \approx\; & E[{}^{\mathcal{M}}\xi'_{\mathcal{S}_i}\,{}^{\mathcal{M}}\xi'^T_{\mathcal{S}_i}] + E[{}^{\mathcal{M}}\xi_{\mathcal{S}_j}\,{}^{\mathcal{M}}\xi_{\mathcal{S}_j}^T] \\ & + E[{}^{\mathcal{M}}\xi'_{\mathcal{S}_i}\,{}^{\mathcal{M}}\xi_{\mathcal{S}_j}^T] + E[{}^{\mathcal{M}}\xi_{\mathcal{S}_j}\,{}^{\mathcal{M}}\xi'^T_{\mathcal{S}_i}] \end{aligned} \tag{7} $$

which, after computing the covariance terms, provides an approximation for the covariance of the relative transformation:

$$ \begin{aligned} \boldsymbol{\Sigma}_{\mathcal{S}_i\mathcal{S}_j} \approx\; & (\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}})\boldsymbol{\Sigma}_{\mathcal{M}\mathcal{S}_i}(\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}})^T \\ & + \boldsymbol{\Sigma}_{\mathcal{M}\mathcal{S}_j} \\ & - (\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}})\boldsymbol{\Sigma}_{\mathcal{M}\mathcal{S}_i,\mathcal{M}\mathcal{S}_j} \\ & - \boldsymbol{\Sigma}^T_{\mathcal{M}\mathcal{S}_i,\mathcal{M}\mathcal{S}_j}(\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}})^T \end{aligned} \tag{8} $$

## References


[1] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, R. Siegwart, Receding horizon path planning for 3D exploration and surface inspection, Auton. Robots 42 (2) (2018) 291–306.

[2] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, S. Leutenegger, Fast frontier-based information-driven autonomous exploration with an MAV, in: Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA), 2020, pp. 9570–9576.

[3] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, J. Nieto, Voxblox: Incremental 3D Euclidean signed distance fields for on-board MAV planning, in: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2017, pp. 1366–1373.

[4] G.A. Hollinger, B. Englot, F.S. Hover, U. Mitra, G.S. Sukhatme, Active planning for underwater inspection and the benefit of adaptivity, Int. J. Robot. Res. 32 (1) (2013) 3–18.

[5] F.S. Hover, R.M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, J.J. Leonard, Advanced perception, navigation and planning for autonomous in-water ship hull inspection, Int. J. Robot. Res. 31 (2016) 1445–1464.

[6] A. Bouman, M.F. Ginting, N. Alatur, M. Palieri, D.D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick, A.-a. Agha-Mohammadi, Autonomous spot: long-range autonomous exploration of extreme environments with legged locomotion, in: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2020, pp. 2518–2525.

[7] K. Ebadi, Y. Chang, M. Palieri, A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood, et al., LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments, in: Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA), IEEE, 2020, pp. 80–86.

[8] Y. Wang, M. Ramezani, M. Mattamala, M. Fallon, Scalable and elastic LiDAR reconstruction in complex environments through spatial analysis, in: Proc. of the European Conference on Mobile Robotics (ECMR), 2021.

[9] N. Funk, J. Tarrio, S. Papatheodorou, M. Popović, P.F. Alcantarilla, S. Leutenegger, Multi-resolution 3D mapping with explicit free space representation for fast and accurate mobile robot motion planning, IEEE Robot. Autom. Lett. 6 (2) (2021) 3553–3560.

[10] J.G. Mangelson, M. Ghaffari, R. Vasudevan, R.M. Eustice, Characterizing the uncertainty of jointly distributed poses in the Lie algebra, IEEE Trans. Robot. 36 (5) (2020) 1371–1388.

[11] F. Dellaert, C. Beall, GTSAM 4.0, 2017, URL: https://bitbucket.org/gtborg/gtsam.

[12] L. Gan, R. Zhang, J.W. Grizzle, R.M. Eustice, M. Ghaffari, BayesIan spatial kernel smoothing for scalable dense semantic mapping, IEEE Robot. Autom. Lett. 5 (2) (2020) 790–797.

[13] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, S. Teller, An Atlas framework for scalable mapping, in: Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA), Vol. 2, 2003, pp. 1899–1906.

[14] J. Nieto, J. Guivant, E. Nebot, DenseSLAM: Simultaneous localization and dense mapping, Int. J. Robot. Res. 25 (8) (2006) 711–744.

[15] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, OctoMap: An efficient probabilistic 3D mapping framework based on octrees, Auton. Robots 34 (3) (2013) 189–206.

[16] B.-J. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, M. Kaess, Virtual occupancy grid map for submap-based pose graph SLAM and planning in 3D environments, in: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 2175–2182.

[17] P. Sodhi, B.-J. Ho, M. Kaess, Online and consistent occupancy grid mapping for planning in unknown environments, in: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2019, pp. 7879–7886.

[18] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, J. Nieto, Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps, IEEE Robot. Autom. Lett. 5 (1) (2020) 227–234.

[19] Y. Wang, N. Funk, M. Ramezani, S. Papatheodorou, M. Popovic, M. Camurri, S. Leutenegger, M. Fallon, Elastic and efficient LiDAR reconstruction for large-scale exploration tasks, in: Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA), 2021, pp. 5035–5041.

[20] D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Guenther, M. Tranzatto, P. Fankhauser, M. Hutter, Advances in real-world applications for legged robots, J. Field Robot. 35 (2018) 1311–1326.

[21] E. Turner, A. Zakhor, Floor plan generation and room labeling of indoor environments from laser range data, in: Proc. of the IEEE Intl. Conf. on Computer Graphics Theory and Applications (GRAPP), 2014, pp. 1–12.

[22] I. Armeni, O. Sener, A.R. Zamir, H. Jiang, I. Brilakis, M. Fischer, S. Savarese, 3D semantic parsing of large-scale indoor spaces, in: Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition, 2016, pp. 1534–1543.

[23] C. Mura, O. Mattausch, A.J. Villanueva, E. Gobbetti, R. Pajarola, Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts, Comput. Graph. 44 (2014) 20–32.

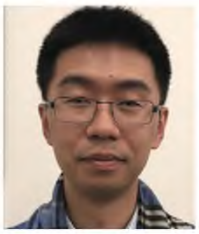
[24] C. Mura, O. Mattausch, R. Pajarola, Piecewise-planar reconstruction of multi-room interiors with arbitrary wall arrangements, in: Computer Graphics Forum, Vol. 35, Wiley Online Library, 2016, pp. 179–188.

[25] S. Ochmann, R. Vock, R. Klein, Automatic reconstruction of fully volumetric 3D building models from oriented point clouds, ISPRS J. Photogramm. Remote Sens. 151 (2019) 251–262.

[26] S. Nikoohemat, A.A. Diakité, S. Zlatanova, G. Vosselman, Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management, Autom. Constr. 113 (2020) 103109.

[27] S. Garg, N. Sünderhauf, F. Dayoub, D. Morrison, A. Cosgun, G. Carneiro, Q. Wu, T.-J. Chin, I. Reid, S. Gould, et al., Semantics for robotic mapping, perception and interaction: A survey, Found. Trends Robot. 8 (2020) 1–224.

[28] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[29] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: Intl. Conf. on Learning Representations (ICLR), 2016.
[30] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: a deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017) 2481–2495.
[31] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Proc. of the Intl. Conf. on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015 pp. 234–241.
[32] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, A. Sorkine-Hornung, A benchmark dataset and evaluation methodology for video object segmentation, in: Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition, 2016, pp. 724–732.
[33] W. Wang, J. Shen, R. Yang, F. Porikli, Saliency-aware video object segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 40 (1) (2017) 20–33.
[34] D. Pathak, R. Girshick, P. Dollár, T. Darrell, B. Hariharan, Learning features by watching objects move, in: Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition, 2017, pp. 2701–2710.
[35] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proc. of the Eur. Conf. on Computer Vision (ECCV), 2018, pp. 801–818.
[36] T. Takikawa, D. Acuna, V. Jampani, S. Fidler, Gated-scnn: gated shape cnns for semantic segmentation, in: Proc. of the Intl. Conf. on Computer Vision (ICCV), 2019, pp. 5229–5238.
[37] B. Wu, A. Wan, X. Yue, K. Keutzer, Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud, in: Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA), IEEE, 2018, pp. 1887–1893.
[38] A. Milioto, I. Vizzo, J. Behley, C. Stachniss, Rangenet++: Fast and accurate lidar semantic segmentation, in: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), IEEE, 2019, pp. 4213–4220.
[39] B. Graham, M. Engelcke, L. Van Der Maaten, 3D semantic segmentation with submanifold sparse convolutional networks, in: Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition, 2018, pp. 9224–9232.
[40] L. Han, T. Zheng, L. Xu, L. Fang, Occuseg: occupancy-aware 3d instance segmentation, in: Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition, 2020, pp. 2940–2949.
[41] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, D. Lin, Cylindrical and asymmetrical 3d convolution networks for lidar segmentation, in: Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition, 2021, pp. 9939–9948.
[42] H. Thomas, C.R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, L.J. Guibas, Kpconv: flexible and deformable convolution for point clouds, in: Proc. of the Intl. Conf. on Computer Vision (ICCV), 2019, pp. 6411–6420.
[43] S. Yang, Y. Huang, S. Scherer, Semantic 3D occupancy mapping through efficient high order CRFs, in: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), IEEE, 2017, pp. 590–597.
[44] J. McCormac, A. Handa, A. Davison, S. Leutenegger, Semanticfusion: Dense 3d semantic mapping with convolutional neural networks, in: Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA), IEEE, 2017 pp. 4628–4635.
[45] M. Ramezani, G. Tinchev, E. Iuganov, M. Fallon, Online LiDAR-SLAM for legged robots with robust registration and deep-learned loop closure, in: Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA), 2020, pp. 4158–4164.
[46] F. Pomerleau, F. Colas, R. Siegwart, S. Magnenat, Comparing ICP variants on real-world data sets, Auton. Robots 34 (3) (2013) 133–148.
[47] F. Dellaert, M. Kaess, Factor graphs for robot perception, Found. Trends® Robot. 6 (1–2) (2017) 1–139.
[48] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J.J. Leonard, F. Dellaert, iSAM2: incremental smoothing and mapping using the Bayes tree, Int. J. Robot. Res. 31 (2) (2012) 216–235.
[49] S. Nobili, G. Tinchev, M. Fallon, Predicting alignment risk to prevent localization failure, in: Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA), 2018, pp. 1003–1010.
[50] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, Rtseg: real-time semantic segmentation comparative study, in: Proc. of the IEEE Intl. Conf. on Image Processing (ICIP), 2018, pp. 1603–1607.
[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[52] N. Carlevaris-Bianco, A.K. Ushani, R.M. Eustice, University of michigan north campus long-term vision and lidar dataset, Int. J. Robot. Res. 35 (9) (2016) 1023–1035.
[53] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, M. Fallon, The newer college dataset: handheld LiDAR, inertial and vision with ground truth, in: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2020, pp. 4353–4360.
[54] W.E. Lorensen, H.E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, SIGGRAPH Comput. Graph. 21 (4) (1987) 163–169.
[55] S. Klarsfeld, J. Oteo, The Baker-Campbell-Hausdorff formula and the convergence of the Magnus expansion, J. Phys. A: Math. Gen. 22 (21) (1989) 4565.



**Yiduo Wang** has been a DPhil student at the ORI since October 2018. He is a member of the Dynamic Robots Systems Group (DRS) under Assoc. Prof. Maurice Fallon. The focus of his research is active mapping using various robot platforms. He holds an MRes Robotics degree awarded by UCL, where his research focused on combining SLAM with semantic segmentation for dynamic environments. He also holds a Bachelor of Engineering (Mechatronic) degree from UNSW, Australia, where his thesis focused on EKF-based SLAM systems. His research interests span localisation, mapping, SLAM and autonomous exploration.

**Milad Ramezani** is a Postdoctoral Fellow in the Embodied AI research team of the Robotics and Autonomous Systems Group at CSIRO Data61. He studied computer vision and photogrammetry at K.N. Toosi University of Technology, Iran. He carried out his Ph.D. research in localisation via GPS/IMU integration and visual-inertial odometry at the University of Melbourne. He was in the DRS Group at ORI (2018–2021). His research focuses on robust and accurate SLAM systems deployable on various robotic platforms. He has contributed to 3D LiDAR SLAM, reconstruction, and visual navigation.

**Matias Mattamala** has been a DPhil student at the ORI since October 2019. He is working in the DRS Group under Assoc. Prof. Maurice Fallon. His research focus is on novel visual navigation methods for legged robots in challenging environments. He completed his B.Sc, Ingeniería Civil Eléctrica, and M.Sc in Electrical Engineering at the Universidad de Chile. His master project was on the development of a visual-proprioceptive SLAM system for humanoid robots. His research interests span from state estimation and computer vision, to mathematical methods and geometry in robotics.

**Sundara Tejaswi Digumarti** is a post-doc at the DRS Group at the ORI. He received his B.Tech in Electrical Engineering from IIT Jodhpur, India and MSc. and Ph.D. in Robotics from ETH Zurich, Switzerland. From 2019–2021 he was a post-doc at ACFR, Australia, where he worked on semantic segmentation of large scale point clouds of forests and learning-based approaches to interpret novel camera systems. His research focuses on using semantics and geometry to improve navigation, scene understanding and 3D reconstruction.

**Maurice Fallon** received his B.Eng (electrical engineering) from University College Dublin, Ireland and his Ph.D. (acoustic source tracking) from University of Cambridge, UK. From 2008–2012 he was a post-doc and research scientist in MIT Marine Robotics Group working on SLAM. Later, he was the perception lead of MIT's team in the DARPA Robotics Challenge. From 2017 he has been a Royal Society University Research Fellow and Associate Professor at University of Oxford, UK. He leads the DRS Group in ORI. His research focuses on probabilistic methods for localisation, mapping, dynamic motion planning, and multi-sensor fusion.