

Autonomous Surveying of Underfloor Voids

Miguel Juliá, Q-Bot Ltd., London, UK,
 Mathew Holloway, Imperial College and Q-Bot Ltd., London, UK,
 Oscar Reinoso, Miguel Hernández University, Elche, Spain,
 Peter R. N. Childs, Imperial College, London, UK.,

Abstract

In this paper, a novel robotic system that solves the problem of autonomous mapping an underfloor void is presented. The approach is based on a 3D laser scanner. A real time navigation system and a new high level planner that selects the next best scanning position controls the motion of the robot. Multiple scans are aligned using ICP and graph optimization techniques. Finally, a point cloud fusion algorithm creates a global model of the environment from the aligned scans. The survey robot has been successfully deployed in a commercial application for scanning underfloor voids before and after the application of thermal insulation. Using this system, the robot was successfully able to autonomously map the controlled test scenario. For some applications the quantity of rubble within the void caused the real time navigation to fail and teleoperation and manual initialization of the ICP algorithm was necessary.

1 Introduction

Autonomous exploration and surveying of an underfloor void or crawl space is a challenge for mobile robots, but one for where there are many applications such as mapping and inspecting services e.g. looking for leaking pipes, damaged insulation or faulty wiring; checking for hazards e.g. surveying for the presence of asbestos.

Crawl spaces and under floor voids provide many problems not encountered in other applications where autonomous exploration and surveying is more common. This includes operating in confined spaces with irregular 3D paths, restricted openings, unknown obstacles, rough terrain, rubble, sand and mud. There are also difficulties for vision systems including dust, poor and inconsistent illumination, shadows and occlusions.

Due to the nature of these environments umbilical cords that can get caught on obstacles are not desirable, while wireless communications have limited range. Therefore it is highly desirable to create robots that can operate autonomously without relying on operator input.

The first contribution of this paper consists of the development and integration of a robotic system that solves this problem by combining a 3D scanner system, ICP-based alignment and 3D model reconstruction with a real time navigation system that includes an autonomous survey planning system that intelligently selects the next best scanning position. In this regard, the second contribution is the extension of the traditional 2D exploration next-best-view algorithm [1] to 3D models. Our approach considers travelling costs and information gain and it also includes a localizability term in order to facilitate the alignments.

2 Related Work

The exploration problem [2] can be defined as the process of autonomously mapping an unknown area. This is usually done by repeatedly selecting the best destination from a subset of candidates according to some criteria such as the shortest travelling distance [3], or a combination of the travelling cost with information gain [1]. The set of candidate destinations comprise the frontiers between the explored and the unexplored areas. This is known as frontier based exploration.

In addition, many authors have focussed in the coordination of multiple robots to explore the environment faster. In this sense, different ways of coordinating the robots, by means of selecting destinations for each robot based on utility functions that measure the trade of between cost and information gain, have been proposed [4] [5]. Furthermore, some authors have studied how the structure of the environment can be used in order to improve the coordination [6]. Other authors have centred their attention on how the planned trajectories are related to the mapping process. In this sense, different trajectories can positively or negatively affect the localization and therefore the accuracy of the created maps [7]. The exploration methods that take into account the relation of the path planning with the simultaneous localization and mapping are normally called *integrated exploration* approaches [8].

Most of these exploration techniques work with a 2D occupancy grid map built from laser or sonar readings. However, with the appearance of new sensors techniques for 3D dense depth perception such as dense stereo camera systems, RBGD cameras, or 3D laser scanners, in the recent years, there has been an increased interest in 3D mapping

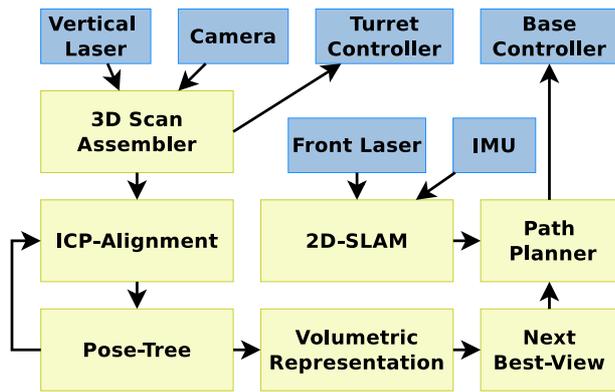


Figure 1 Architecture of the proposed real time system.

with mobile robots. In this regard, many authors have focussed in developing techniques for quickly registering 3D depth data and built global 3D models. The iterative closest point algorithm (ICP) [9] and variants of it [10] have become a quite common choice for registering the data and it has been integrated in many mapping systems with depth sensors using different map representations as, for instance, volumetric distance functions [11] or surfel-based representations [12]. ICP is used for finding the alignment between two datasets. In this sense, it can be used in order to find the alignment between new data and a global model for a subsequent update of that global model in an incremental way, or it can be used in order to align multiple datasets, thus creating a pose graph that can be later optimized [13].

3 Approach Overview

The underfloor void case scenario presents many challenges from the exploration point of view. While the scenario can vary significantly from one real-case site to another, in general, the confined spaces does not suit well a multi-robot system for a fast coordinated exploration. In addition, the relevant features of the environment, e.g. insulation, pipes or other services, jointly with structural features like joists or walls require a 3D mapping system. This makes necessary a different exploration approach than the traditional 2D map based exploration methods.

Furthermore, the robot has to move in an irregular surface that makes real-time localization difficult and operate in a poorly illuminated confined space that makes vision systems more complex. In this regard, depth cameras use to have a limited depth range and a narrow field of view. Therefore, they are not the best choice for operating in a confined space with poor illumination. Consequently, given the difficulties for registering the data in these conditions, it was decided to base the approach on a 3D textured laser scanner in order to gather a large amount of data from a small set of scanning positions.

In this way, the 3D exploration problem is reduced to capturing and aligning multiple scans and fusing them into a global model. The proposed architecture is illustrated in

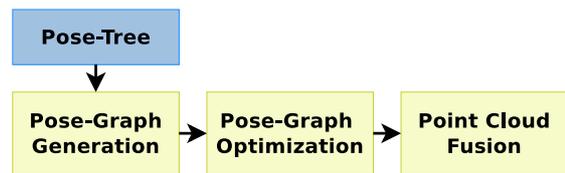


Figure 2 Post-processing procedure.

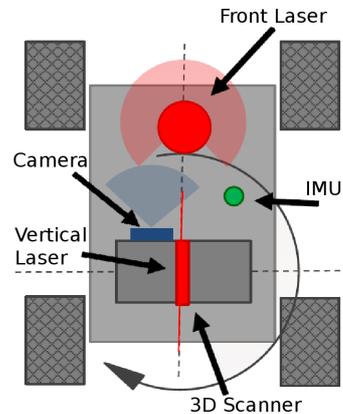


Figure 3 Diagram of the robot and sensors placement.

Figure 1. The 3D scanner consists of a camera and a vertically mounted laser on top of a rotating turret. The depth and colour data from these two sensors are collected as the turret rotates and assembled into a 3D textured point cloud. This system and its calibration is described in Section 4.1. The multiple scans are sequentially aligned in a pose-tree using ICP. A low resolution volumetric representation is updated after each new aligned scan. This SLAM system will be explained in Section 4.2. However, it only evaluates the localization of the robot for places where a scan is performed. In this sense, a 2D SLAM system runs in parallel and is used for navigating between the scanning poses as presented in Section 4.3. In addition, a new 3D exploration method is introduced in Section 4.4 for selecting the next best scanning position using the volumetric representation. Finally, as illustrated by Figure 2, a post-processing stage takes place that optimizes the alignment of multiple scans in a graph and fuses the data in a global is summarized in Section 4.5.

4 System Description

4.1 Robot Architecture

The surveying robot designed consists of a small 4 wheeled robotic platform with a front horizontal laser, an IMU and an actuated camera-laser system (3D scanner). Figure 3 shows a diagram of the robot and the sensors placement. The ROS framework [14] is used over a distributed system between an on-board computer and an external operator station.

4.1.1 3D Scanning System

The 3D scanner consists of an actuated camera-laser system. The laser scans a vertical plane that turns as the turret rotates. The camera is used in order to add colour data to the depth measurements of the laser.

In this sense, the laser provides a set of distance readings ρ_i at different angles θ_i . These readings can be expressed as 3D points in the laser frame of reference $\mathbf{p}_i^{[l]} \in \mathbb{R}^3 = [\rho_i \cos \theta_i \quad \rho_i \sin \theta_i \quad 0]^T$. This is translated to the fixed frame by:

$$\mathbf{p}_{i,j}^{[f]} = \mathbf{R}_x(\phi_j) \mathbf{T}_{laser} \mathbf{p}_{i,j}^{[l]} \quad (1)$$

where $\mathbf{T}_{laser} \in \mathbb{SE}_3$ is the calibrated position of the laser in the moving 3D scanner frame of reference and $\mathbf{R}(\phi_j) \in \mathbb{SO}_3$ is the corresponding rotation of the actuator of an angle ϕ_j . Note that, for simplicity of notation, the conversion between 3-vectors and the corresponding homogeneous 4-vectors (needed for multiplication with \mathbf{T}_{laser}) has been omitted.

Equation (1) is used in order to project every point acquired during a 360 degrees turn into the fixed 3D scanner frame of reference, thus generating a point cloud. Finally, the colour of the corresponding pixels in the images are associated with each point by:

$$\mathbf{c}_{\{i,j\},k} = I_k \left(\pi \left(\mathbf{K} \mathbf{T}_{cam} \mathbf{R}_x(-\phi_k) \mathbf{p}_{i,j}^{[f]} \right) \right) \quad (2)$$

where $\mathbf{R}(-\phi_k) \in \mathbb{SO}_3$ is the rotation matrix corresponding to the actuator at the angle ϕ_k at which the image was acquired, $\mathbf{T}_{cam} \in \mathbb{SE}_3$ is the transformation corresponding to the calibrated camera pose in the 3D scanner moving frame, K is the calibrated camera matrix, $\mathbf{u} = \pi(\mathbf{x})$ is a function that performs the dehomogenisation of $\mathbf{x} \in \mathbb{R}^3 = (x, y, z)$ in order to obtain $\mathbf{u} \in \Omega = (x/z, y/z)$, and $I_k : \Omega \mapsto \mathbb{N}^3$ is the subpixel mapping between the image space domain $\Omega \subset \mathbb{R}^2$ and the colour values corresponding to the rectified image k .

4.1.2 3D Scanner Calibration

In order to calibrate the pose of the laser in the moving frame $\mathbf{T}_{laser} \in \mathbb{SE}_3$, it is necessary to split the vertical laser points into two parts (front and rear) as shown in [15]. Then the laser pose can be calibrated by minimizing the error between corresponding points of the front and the rear parts of the scan after a 360 degrees movement of the turret. Thus the scene is completely covered with each part of the scan. In this sense, gradient descent is applied in order to minimize the error:

$$E_{laser}(\mathbf{T}_{laser}) = \sum_{m=1}^M \|\mathbf{p}_m^{[f]} - \mathbf{p}_{m'}^{[f]}\|^2 \quad (3)$$

where $\mathbf{p}_{m'}^{[f]}$ is the nearest point in data from the rear part of the scan to the point $\mathbf{p}_m^{[f]}$ from the front part of the scan expressed in the fixed frame of reference. In this case, the gradient is estimated numerically.

Similarly, the camera pose $\mathbf{T}_{cam} \in \mathbb{SE}_3 = [\mathbf{R}_c \quad \mathbf{t}_c]$ needs to be also calibrated. To achieve this, a planar chess-

board pattern of known dimensions is 3D scanned at multiple locations. In addition, an image is captured from each location with the turret positioned at the zero angle ($\phi = 0$). The equation of the plane containing the pattern can be easily obtained for each pair of 3D laser scan (referred to the fixed frame) and image (in the camera frame). While typical methods from intrinsic camera parameters calibration can be used for detecting the chessboard in the image, RANSAC can be applied for plane fitting to the laser scan. The normals of these planes are arranged as two matrices $\mathbf{N}_L, \mathbf{N}_c$, one for the laser in the fixed frame and one for the camera in the camera frame. Then, the covariance $\mathbf{H} = \text{cov}(\mathbf{N}_L, \mathbf{N}_c)$ between these two matrices can be decomposed $\mathbf{H} = \mathbf{U} \mathbf{S} \mathbf{V}'$ using SVD in order to obtain the rotation part of the camera transform $\mathbf{R}_c = \mathbf{V} \mathbf{U}'$ in a similar way to [16]. In order to find the translation, it is necessary one pair of corresponding points between the laser and camera data. A possible choice, that can be extracted directly from the plane equations, is the pair or solutions corresponding to the least squares problems $\mathbf{N}_c \mathbf{X}_c + \mathbf{D}_c = 0$ and $\mathbf{N}_L \mathbf{X}_L + \mathbf{D}_L = 0$, where \mathbf{D}_c and \mathbf{D}_L are column vectors of the independent terms in the plane equations. Then, the camera translation is $\mathbf{t}_c = \mathbf{R}_c \mathbf{X}_L - \mathbf{X}_c$.

4.2 ICP-SLAM

Different point clouds are aligned using the Iterative Closest Point Algorithm (ICP) [10]. In this case, the goal of ICP is to find the transformation $\mathbf{T} \in \mathbb{SE}_3$ that minimizes the point to plane error function:

$$E(\mathbf{T}) = \sum_{m=1}^M ((\mathbf{T} \mathbf{p}_m - \mathbf{p}'_m) \cdot \mathbf{n}_{p_m})^2 \quad (4)$$

where \mathbf{p}_m and \mathbf{p}'_m are corresponding pairs of points between two point clouds obtained by nearest neighbour search with the help of a kd-tree, and \mathbf{n}_{p_m} is the normal vector to point p_m . This non-linear minimization is solved by linearisation [17].

A tree of aligned 3D scanning positions is incrementally built by means of aligning the new point clouds with the closest cloud in the tree according to the initial estimate.

Furthermore, a low resolution volumetric representation of the occupation is updated with each new aligned 3D scan. In addition to the occupation state (free, occupied or unobserved), surface normals are calculated and saved for each voxel in this representation.

4.3 Autonomous Navigation

The ICP-SLAM system described in the previous section is only updated at the points where a 3D scan is performed. Therefore, it is necessary to perform a real time localization and reactive obstacle modelling in order to control the motion of the robot when navigating to the next 3D scanning position. The uneven terrain present in underfloor voids exacerbates this task.

The main sensors involved in the real time localization and obstacle modelling are the front laser and the IMU. Typical laser navigation approaches are not suitable in rough

terrains since they are designed for planar surfaces. In this regard, using the IMU, the laser readings of points corresponding to the ground can be filtered and the remaining points are projected to the horizontal plane before using laser scan matching for localization and typical occupancy grids for obstacle modelling [18]. The global position of the scan matcher is reinitialized after each new 3D scan has been aligned.

Dijkstra's Algorithm is used in order to plan the path to the next scanning position and a dynamic window approach is applied for low level planning [19].

4.4 Next Best View

The position where the next 3D scan will be performed is decided from the evaluation of multiple candidate destinations. In [1] the next utility function that considers information gain and travelling cost was proposed for performing this evaluation:

$$V(q) = G(q)e^{-\lambda C(q)} \quad (5)$$

where the utility $V(q)$ is proportional to the information gain term $G(q)$, and has an exponential decay with the distance to the target $C(q)$, and λ is a design parameter. The information gain term is calculated counting the previously unobserved cells in a 2D occupancy grid map that could be visible from the evaluated candidate position q ,

While it was used for the 2D case in [1], this utility function can be easily extended to the 3D case by means of evaluating the $G(q)$ term with a 3D ray trace counter in the previously explained low resolution volumetric representation. However, that leads to selecting target destinations that maximize information gain without considering how easy would it be to align it with the previous data. In this regard, it is proposed here to use the next utility function that introduces a localizability term:

$$V(q) = G(q)L(q)e^{-\lambda C(q)} \quad (6)$$

This localizability term $L(q)$ is evaluated as follows:

$$L(q) = \sum_{\mathbf{p}_v} h \left(\frac{\mathbf{n}_{\mathbf{p}_v} \cdot (\mathbf{p}_v - \mathbf{p}_q)}{\|\mathbf{p}_v - \mathbf{p}_q\|} \right) \quad (7)$$

where \mathbf{p}_v is the position of voxel $v \in V \subset \mathbb{N}^3$ from the subset of visible occupied voxels in sensor range from the candidate position \mathbf{p}_q , and $\mathbf{n}_{\mathbf{p}_v}$ are the normal unit vectors associated to the voxel v . Note that this term counts the number of visible occupied voxels weighted by the cosine of the angle between the surface normal vector and the observation direction. The function $h(\mathbf{x})$ produces that the observations from behind do not contribute to the count:

$$h(\mathbf{x}) = \begin{cases} \mathbf{x}, & \mathbf{x} \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

4.5 3D Model Post-Processing

4.5.1 Pose-Graph

The previous systems work online as the robot captures the data and decides where to move next. However, a better model can be created from a pose-graph including loops

with multiple alignments between scans than with the simple pose-tree. Since the ICP alignment is a computationally expensive step, it is left for a post-processing stage to find all the extra alignments for building a pose-graph of the scanning positions. In this sense, additional edges are added to the initial tree for each new valid alignment found with ICP. An alignment is considered to be valid to be inserted in the graph, if the residual error (Equation 4) is low while the number of inlier points in the alignment is above a threshold.

Next, the pose-graph is optimized using stochastic gradient descent (SGD) using the tree-based network optimizer [13]. This solves the loop closure discrepancies that appear with the accumulation of error as the new data is incrementally aligned.

4.5.2 Point Cloud Fusion

The next step is to fuse the aligned point cloud data associated to each pose in the graph. A surface in the scene would have been likely observed from multiple poses, having therefore redundant data of the same area with different resolutions, noise and texture due to changes in exposure and illumination. In this sense, in order to facilitate the fusion process, a radius that depends on the distance is associated to each point (similar to the radius used in surfel representations of point clouds [12]).

The fusion process is summarized in Algorithm 1. The global point cloud is initialized with the point cloud corresponding to the first node in the pose graph. This global point cloud is incrementally updated with the other nodes. A set of correspondences is found by associating to each point in the global cloud the nearest point in the input cloud inside its radius. Next, the points in the input cloud that remain without correspondence are inserted in the global cloud. The points with correspondence are ignored if they have a larger radius than the corresponding point in the global cloud. In case they have a smaller radius, they are inserted in the global cloud and the corresponding point removed.

5 Experiments

The robot has been evaluated in a controlled test scenario and also in real world underfloor voids. The main goals of the experiments was to study the validity of the system as a 3D mapping tool for measuring depth of installed underfloor insulation and the viability of it being fully autonomous in such challenging scenarios. Next the main results of these tests are presented.

5.1 Experiments in a test scenario

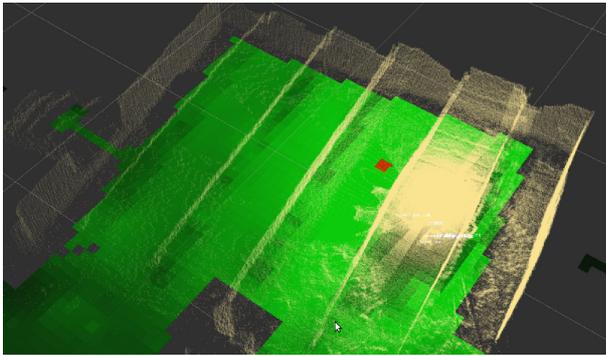
Figure 4 shows an example of the functioning of the presented next-best scanning position algorithm. The lightness of the green colour represents the normalized values of the profit function of Equation 6. The previously scanned area is shown as a yellow point cloud. The red square represents the selected next best scanning position. As it can be

Algorithm 1 Point Cloud Fusion

```

GlobalCloud ← Node[0].PointCloud
for n ∈ [1, N - 1] do
  for p ∈ GlobalCloud do
    q ← Node[n].findNearestPoint(p)
    if ||p - q|| < p.radius then
      q.match ← true
      q.corr ← p
    end if
  end for
  for q ∈ Node[n] do
    if q.match then
      if q.radius < q.corr.radius then
        GlobalCloud.remove(q.corr)
        GlobalCloud.insert(q)
      end if
    else
      GlobalCloud.insert(q)
    end if
  end for
end for

```

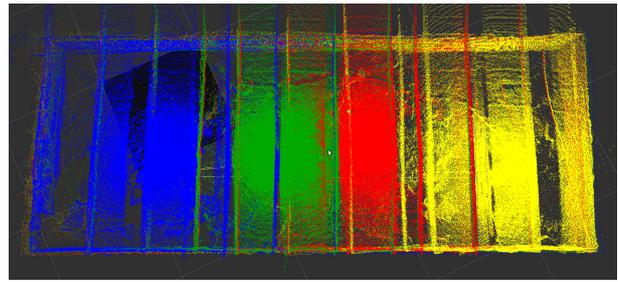
**Figure 4** Next-best scanning position example.

seen, this method selects a point not too far away in order to reduce navigation time and assure a good ICP alignment of the next scan but also selects a point between the joists of the void that were occluded in the previous scan.

Using this system, the robot was successfully able to map the controlled test scenario using a total of 7 scans with an average time of 6 minutes for each scan. These 6 minutes consisted of the scanning time (4 min), the ICP-SLAM update time (45 sec), planning time (45 sec) and trajectory following time (30 sec).

5.2 Experiments in a real underfloor void

The tests in real world underfloor voids presented significant problems regarding the navigation algorithms described in Section 4.3. The amount of rubble in the void could cause the scan matcher to fail. This has a significant impact in the full automation of the approach, since the selected scanning position were sometimes not reached correctly or appeared as unreachable because of the poor real-time localization. In addition, this also caused a good

**Figure 5** ICP-SLAM example.**Figure 6** Point cloud fusion example.

position guess to not be available to initialize the ICP process. This generates an incorrect global model that consequently influences the goal generation algorithm.

In this regard, input from the operator was necessary for teleoperating the robot to the selected scanning positions and manual initialization of the ICP algorithm. Figure 5 shows the results of the alignment of several point clouds generated from 3D scans in a real world underfloor void.

The fusion process described in Section 4.5 was applied and the resulting global textured point cloud is shown in Figure 6.

5.3 Results measuring insulation depth

One of the goals of the surveying of underfloor voids with the robot was to perform coverage and depth measurements of underfloor insulation. This enables the void to be mapped before and after the installation of the insulating foam. By aligning these two models the depth measurements can be obtained.

Figure 7 shows an example of insulation depth measurement. For a better visualization, the figure shows only the points corresponding to the top surface of the void where the foam is installed. The two different surfaces that appear in the figure correspond with the before model and after model. The colour code shows the difference in height, and it can be clearly observed an average of 150 mm of depth in the installed insulation.

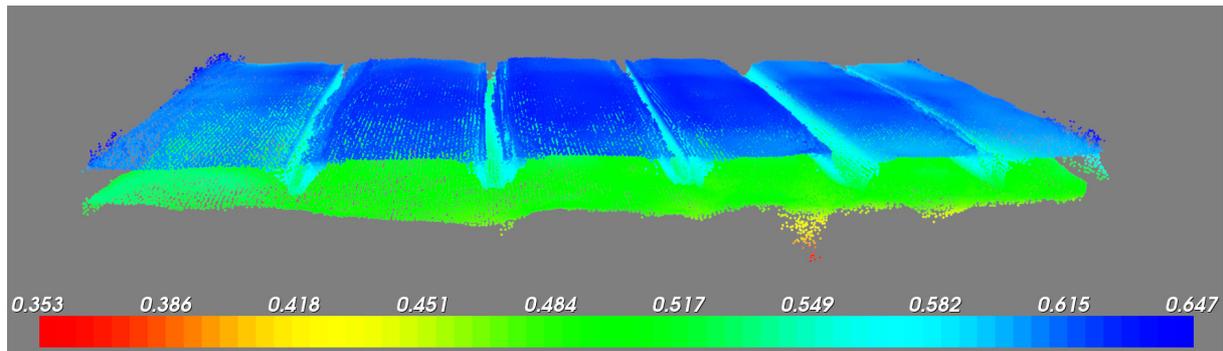


Figure 7 Insulation depth measurement example.

6 Conclusions and Future Work

A novel solution for the autonomous survey of underfloor voids has been proposed and demonstrated in this paper. The solution presented is based on a 3D scanner system that with an associated calibration procedure. A real time navigation system was integrated in the robot and a new high level planner, that selects the next best scanning position, has been designed. Multiple 3D scans are aligned using ICP to provide the necessary models for the target selection and later global model generation. In this sense, a fusion algorithm was designed in order to consistently combine all the data.

The tests performed show that the solution is viable for mapping the void and measuring insulation depth. However, some improvements are necessary for a robust automated survey. While tests in a controlled scenario were successful, the real-time localization was found unreliable for real world scenarios with large amounts of rubble in the floor. This leads to missing the selected scanning position and having a poor initial estimate for the ICP algorithm. When the ICP algorithm produces a bad alignment, this affects the next-best scanning position generation that affects the full approach for autonomous data acquisition.

Consequently, one of the points for further research is the design of a better real-time localization of the robot in these difficult conditions. Despite it being influenced by real time localization, the initialization of the ICP algorithm can be considered a different issue. In this sense, visual features could be used in the alignment. In addition, the total time between scans would be improved by using a faster 3D scanner and parallelization of some of the algorithms involved in the ICP-SLAM and next-best scanning position algorithm.

Furthermore, while the angular resolution of commercial laser scanners is about 0.25-0.36 degrees, a high resolution camera can provide more angular resolution even using wide angle lenses. In this sense, there is more texture colour data available in the 3D scanner than depth data. While the process explained in Section 4.1 was being limited to only points with depth data, the remaining image points can still be used with interpolated depth producing a higher definition point cloud. In this way, the proposed de-

sign that provides initially about 8×10^5 points with depth data, can generate about 9×10^6 points clouds using interpolated depth. However increasing the number of points by one order of magnitude slows the system too much for on-line processing and therefore only can be used during the post-processing stage. In this sense, different alignments techniques could be applied using the photometric data.

7 Acknowledgements

The authors and Q-Bot Ltd would like to thank the following organisations for their support in the development of the technology: InnovateUK, The Royal Commission for the Exhibition of 1851, The Department of Energy and Climate Change, and The EU Commission.

8 Literature

- [1] H. H. González-Baños and J.-C. Latombe, "Navigation Strategies for Exploring Indoor Environments.," *I. J. Robot Res.*, vol. 21, no. 10-11, pp. 829–848, 2002.
- [2] M. Juliá, A. Gil, and Ó. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments.," *Auton. Robots*, vol. 33, no. 4, pp. 427–444, 2012.
- [3] B. Yamauchi, "A frontier-based approach for autonomous exploration.," in *CIRA*, pp. 146–151, IEEE Computer Society, 1997.
- [4] R. G. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. L. S. Younes, "Coordination for Multi-Robot Exploration and Mapping.," in *AAAI/IAAI* (H. A. Kautz and B. W. Porter, eds.), pp. 852–858, AAAI Press / The MIT Press, 2000.
- [5] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer, "Multi-Robot Exploration Controlled by a Market Economy.," in *ICRA*, pp. 3016–3023, IEEE, 2002.
- [6] K. M. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment.," in *IROS*, pp. 1160–1165, IEEE, 2008.
- [7] C. Stachniss, G. Grisetti, and W. Burgard, "In-

- formation Gain-based Exploration Using Rao-Blackwellized Particle Filters.,” in *Robotics: Science and Systems* (S. Thrun, G. S. Sukhatme, and S. Schaal, eds.), pp. 65–72, The MIT Press, 2005.
- [8] A. Makarenko, S. Williams, F. Bourgault, and H. Durrant-Whyte, “An experiment in integrated exploration,” *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, vol. 1, pp. 534–539 vol.1, 2002.
- [9] P. J. Besl and N. D. McKay, “A Method for Registration of 3-D Shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.
- [10] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, “Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D.,” *IEEE Robot. Automat. Mag.*, vol. 22, no. 4, pp. 110–124, 2015.
- [11] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, “KinectFusion: Real-time dense surface mapping and tracking.,” in *ISMAR*, pp. 127–136, IEEE Computer Society, 2011.
- [12] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, “ElasticFusion: Dense SLAM Without A Pose Graph.,” in *Robotics: Science and Systems*, 2015.
- [13] G. Grisetti, C. Stachniss, and W. Burgard, “Non-linear Constraint Network Optimization for Efficient Map Learning.,” *IEEE Trans. Intelligent Transportation Systems*, vol. 10, no. 3, pp. 428–439, 2009.
- [14] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA Workshop on Open Source Software*, 2009.
- [15] H. Alismail and B. Browning, “Automatic Calibration of Spinning Actuated Lidar Internal Parameters.,” *J. Field Robotics*, vol. 32, no. 5, pp. 723–747, 2015.
- [16] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-Squares Fitting of Two 3-D Point Sets.,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 5, pp. 698–700, 1987.
- [17] K.-L. Low, “Linear least-squares optimization for point-to-plane icp surface registration,” *Chapel Hill, University of North Carolina*, vol. 4, 2004.
- [18] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation,” in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pp. 155–160, Nov 2011.
- [19] E. Marder-Eppstein, E. Berger, T. Foote, B. P. Gerkey, and K. Konolige, “The Office Marathon: Robust navigation in an indoor office environment.,” in *ICRA*, pp. 300–307, IEEE, 2010.