

Actively Mapping Industrial Structures with Information Gain-Based Planning on a Quadruped Robot

Yiduo Wang, Milad Ramezani and Maurice Fallon

Abstract—In this paper, we develop an online active mapping system to enable a quadruped robot to autonomously survey large physical structures. We describe the perception, planning and control modules needed to scan and reconstruct an object of interest, without requiring a prior model. The system builds a voxel representation of the object, and iteratively determines the Next-Best-View (NBV) to extend the representation, according to both the reconstruction itself and to avoid collisions with the environment. By computing the expected information gain of a set of candidate scan locations sampled on the as-sensed terrain map, as well as the cost of reaching these candidates, the robot decides the NBV for further exploration. The robot plans an optimal path towards the NBV, avoiding obstacles and un-traversable terrain. Experimental results on both simulated and real-world environments show the capability and efficiency of our system. Finally we present a full system demonstration on the real robot, the ANYbotics ANYmal, autonomously reconstructing a building facade and an industrial structure.

I. INTRODUCTION

In the context of robotics, active perceptual planning refers to exploration by a mobile robot equipped with sensors to conduct a survey of an object or environment of interest. It can be of assistance for the regular inspection and monitoring of remote or dangerous facilities such as offshore platforms.

Although active mapping has been investigated for many applications such as inspection [1] and virtual modelling [2], and on robotic platforms as varied as aerial [3], [4], wheeled [5], [6] and underwater robots [7], [8], the online deployment of such a system on a real robot is still a challenge, thus requiring further investigation.

Advances in quadruped mobility and hardware reliability have been significant and the first industrial prototypes are being tested on live industrial facilities [9]. Quadrupeds can cover the same terrain as wheeled or tracked robots but can also cross mobility hazards and climb stairs. While UAVs are being actively deployed for these kinds of missions, it is difficult to operate aerial platforms within confined spaces and their sensing payload is limited.

Our approach is to build and maintain an accurate 3D model of the object of interest as well as the local environment and to use it to plan actions of a quadruped robot to improve and extend the model.

Our active mapping framework adapts the Information Gain (IG) approach originally formulated in [5], which

This research is supported by the ES/PRC ORCA Robotics Hub (EP/R026173/1). M. Fallon is supported by a Royal Society University Research Fellowship.

The authors are with the Oxford Robotics Institute, University of Oxford, UK. {ywang, milad, mfallon}@robots.ox.ac.uk

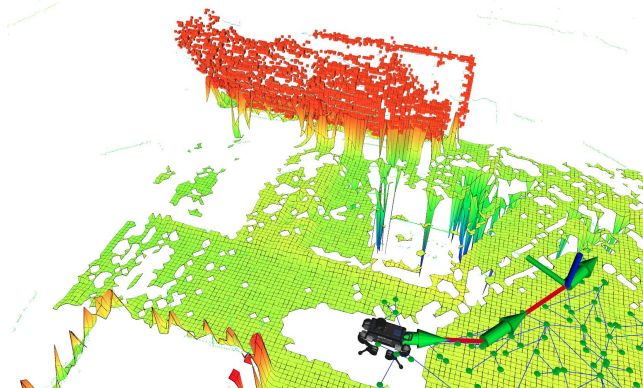


Fig. 1. Top: The ANYmal robot actively mapping a mockup helicopter at the Fire Service College in Gloucestershire, UK, as shown at <https://youtu.be/K348uuCB8gY>. The safety stanchions, stairwells and the skirt under the helicopter are mobility hazards which our approach can avoid. Bottom: the state of the mapping system showing the object reconstruction, LiDAR elevation map, an RRT plan and the next walking goal.

focused on the IG formulation with minimum attention to the problems that an actual robot faces in realistic inspections.

We aim for a more realistic validation in an industrial setting (Fig. 1). The object of interest will be of unknown shape, surrounded by uneven terrain and mobility hazards and our solution will be embodied on the ANYmal quadruped robot. We present a complete active mapping system in the experimental section of this paper.

The contributions of our research are as follows:

- Implementation of an active mapping system on a quadruped, enabling the robot to not only traverse an unstructured environment, but also to scan an object of interest in an optimal manner using LiDAR.
- Formulation of the approach as a Next-Best-View (NBV) problem, which determines the best pose for the robot to conduct the next scan on the basis of metrics

drawn from the partial reconstruction (information gain) and the environment map (cost of mobility).

- Evaluation of the system in simulated and real-world environments and the real-time deployment of our system on the ANYmal quadruped robot.

The remainder of this paper is organised as follows. In Section II we discuss related works with our method detailed in Section III. Experimental results are given in Section IV before discussing conclusions and future work in Section V.

II. RELATED WORKS

First we will review existing active mapping systems grouped according to two aspects: the prior assumptions made and the type of representation used.

A. Prior Assumptions

Active mapping systems are typically divided between model-based and model-free approaches [10]. Model-based methods are essential for routine survey and inspection. They are typically applied in industrial scenarios because CAD models are often available [11].

Blaer and Allen [12] designed a model-based system for 3D mapping of large environments by solving an Art Gallery Problem (AGP) and a Travelling Salesman Problem (TSP) for path optimisation. The system of Hollinger *et al.* [1] was designed to decrease the uncertainty in a ship hull surface mesh reconstruction and improve its quality. Their system assumed the availability of an initial mesh reconstruction and planned viewpoints that could inspect high uncertainty areas of the mesh.

Model-free active vision systems are more versatile and can be applied to a wider variety of objects and sites. For instance, the system of Bircher *et al.* [3], [13] aimed to explore unknown spaces of different scales, and the approach of Kriegel *et al.* [2], [14] was designed to reconstruct objects of arbitrary shape but confined size.

Model-based approaches can also be adapted to incorporate uncertainties in the prior model and to improve the quality of reconstruction. In a work following up on [1], Hover *et al.* [8] addressed the potential lack of prior information by carrying out a coarse-to-fine multi-stage survey.

B. Representations

Many active mapping systems employ either surface mesh or voxel space representations. However, there are works such as [14], [15] which benefited from both - surface mesh for reconstruction and voxel space for collision avoidance.

Hollinger *et al.* [1] and Hover *et al.* [8] utilised a surface mesh to precisely reconstruct ship hulls with the mesh providing information about surface coverage, boundaries and holes. Schmid *et al.* [16] used a coarse Digital Surface Model (DSM) as a prior map to plan viewpoints for a UAV. Their algorithm’s complexity is linearly proportional to the size of the site, limiting the applicability of this system to smaller scale environments.

In volumetric approaches, view selection metrics such as IG are normally used to determine an NBV.

Bircher *et al.* [3], [13] utilised a volumetric representation with their receding horizon planning strategy to progressively explore unknown environments. They grew a Rapidly-exploring Random Tree (RRT) [17] and selected the best branch based upon IG that evaluates the amount of observable unknown space.

Isler *et al.* [5] proposed a collection of Volumetric Information (VI) measures. *Occlusion Aware* computes the entropy of all observable voxels. *Unobserved Voxel* only counts unknown voxels, thereby inclining the system towards exploring void spaces. *Rear Side Voxel* and *Rear Side Entropy* focus the sensor on the object of interest. *Proximity Count* was shown to be advantageous in ensuring coverage of the object in their experiments but has a disadvantage of potentially pointing the sensor away from the object. The authors demonstrated these measures on a KUKA Youbot with a 5 Degree of Freedom (DoF) arm in an office room.

Delmerico *et al.* [4] then conducted more experiments, comparing their VI formulations with the approaches of [15] and [18], to determine the best choice for the NBV selection in a volumetric representation.

We aim for a model-free system able to fully scan an object of interest but also to explore in an unknown environment, therefore our system minimises the entropy/uncertainty in the environment while focusing on the object. We pair this with an octree as our robot’s volumetric representation, storing occupancy probability using OctoMap [19].

C. Localization and Mapping

Accurate mapping is crucial for precise model reconstruction and active planning. When a prior model is available, it can be used for pure localization, however a model-free approach requires a complete Simultaneous Localisation and Mapping (SLAM) system - itself an active research problem.

By its nature, a SLAM system will drift during exploration. Planning methods which use rigid representations such as a single OctoMap would struggle to respond to loop closures. The approach of [7] is interesting—using a deformable reconstruction with *Virtual Occupancy Maps* attached to a pose graph to be flexible to new loop closures. For our real world experiments, we used a rigid map representation but are interested in this approach for future work.

III. METHOD

In this section we detail the modules of our active mapping system. Fig. 2 illustrates a block diagram of the system architecture. The system is based on an iterative pipeline. At the start of each iteration, the robot executes a scanning action while standing (further described in Sec. IV-A) to collect a sensor sweep. These measurements are incorporated into a map, a route to a new scan location is planned and the robot is requested to walk to the NBV for further exploration. This sequence is repeated, until a termination criterion (such as map completion) is met.

The LiDAR measurements \mathcal{L}_B are sensed and then stored relative to the base frame $\{\mathcal{B}\}$. During a scanning action, \mathcal{L}_B is transformed into the map frame $\{\mathcal{M}\}$ based on the current

pose of the robot $\mathbf{x}_M \in \mathbb{R}^6$ and is accumulated into a larger point cloud. Our robot runs a localization system with little drift on the scale of our current experiments, allowing us to assume that the pose \mathbf{x}_M is accurate. This is discussed further in Sec. IV.

The accumulated point cloud is denoted *sweep* \mathcal{S}_M in our system. \mathcal{S}_M is then downsampled for uniformity and filtered to remove outliers. Next, the system uses the processed *sweep* \mathcal{S}_M as well as the pose of the robot \mathbf{x}_M to update the occupancy probabilities of voxels in its OctoMap.

We also use the LiDAR measurements to generate an elevation map of the environment. The elevation map has a useful range of about 10 m, allowing only local planning. The path planning module evaluates terrain traversability subject to the elevation map and builds an RRT to generate a collection of scan candidates \mathcal{C} . A scan candidate $c \in \mathcal{C}$ is a pose where the robot could go to for the next scanning action.

We use a utility function \mathcal{U}_c to determine the best scan candidate (NBV), c_{best} , from the set of candidates \mathcal{C} :

$$\mathcal{U}_c = \mathcal{G}_c \times (1 - \mathcal{P}_c) \times (1 - \mathcal{T}_c). \quad (1)$$

This function combines contributions from

- information gain \mathcal{G}_c : which measures the expected improvement of the model if given a sweep from that pose,
- position cost \mathcal{P}_c : which penalises poses that have already been visited or are too close to the object,
- traversal cost \mathcal{T}_c : which models the cost of travelling to a specific scan candidate pose.

These measures are discussed in the following sections. Finally, our system replans an optimised path using RRT* [20] before the robot takes the next mapping action.

A. Volumetric Information (VI) Gain

Given a partial model of the object of interest, our system needs to determine the expected improvement in the model should a scan be made from a particular scan candidate pose. The approach is to trace a series of rays from a hypothetical pose and to estimate the expected information gain of observable voxels.

Let \mathcal{R}_c denote the set of rays cast by a scan candidate c . For each ray $r \in \mathcal{R}_c$, \mathcal{V}_r is the set of voxels that the ray intersects with before reaching its endpoint. The information gain \mathcal{G}_c at scan candidate c is the sum of VIs, \mathcal{I} , in every voxel $v \in \mathcal{V}_r$ along each ray $r \in \mathcal{R}_c$:

$$\mathcal{G}_c = \sum_{\forall r \in \mathcal{R}_c} \sum_{\forall v \in \mathcal{V}_r} \mathcal{I}. \quad (2)$$

We implemented two formulations for \mathcal{I} from Isler *et al.* [5], namely *Occlusion Aware* \mathcal{I}_{OA} and *Rear Side Entropy* \mathcal{I}_{RSE} which we summarise here.

Other proposed formulations are less relevant due to our sensor's long range and 360° Field of View (FoV).

1) *Occlusion Aware*: This measure determines how effectively uncertainty will be reduced by scanning at a certain pose considering voxel visibility.

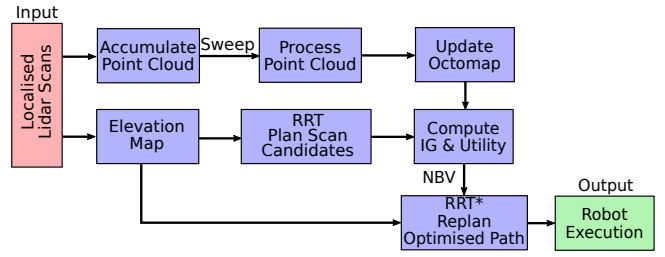


Fig. 2. Block diagram of the system architecture.

Given the occupancy probability $P_o(v)$ of voxel v , the entropy of the voxel is obtained from:

$$H(v) = -P_o(v) \ln P_o(v) - (1 - P_o(v)) \ln(1 - P_o(v)). \quad (3)$$

Then the *Occlusion Aware* VI of voxel v , $\mathcal{I}_{OA}(v)$, is:

$$\mathcal{I}_{OA}(v) = P_v(v)H(v), \quad (4)$$

where $P_v(v)$ is the visibility probability of voxel v , which is computed as :

$$P_v(v_n) = \prod_{i=0}^{n-1} (1 - P_o(v_i)), \quad (5)$$

where v_n is the n -th voxel along the ray r and v_i , $i = 0 \dots n-1$, is a voxel that r intersects before reaching v_n .

2) *Rear Side Entropy*: This measure is based on the *Occlusion Aware* VI but focuses on voxels at the rear of observed surfaces. Rear Side Entropy is formulated as:

$$\mathcal{I}_{RSE}(v) = \begin{cases} \mathcal{I}_{OA}(v) & v \text{ is a Rear Side Voxel,} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The idea is that a *Rear Side Voxel* is also likely to be occupied by the object. Focusing exploration on these voxels concentrates scans on the object rather than on surrounding free space. While these metrics proposed by Isler *et al.* [5] are useful, their experimental validation was limited to lab experiments with a stereo camera planning over a fixed set of poses. We are motivated to develop a more complete field system which operates in a large scale industrial site.

Our system instead plans scan candidates progressively using an RRT which plans on the LiDAR elevation map. Our robot explores using a Velodyne LiDAR, which has a 360° FoV horizontally and long sensing range, which is suitable for scanning large-scale objects or environments.

In our experimental section (Sec. IV) we compare *Rear Side Entropy* and *Occlusion Aware* in field experiments.

B. Position and Traversal Cost

In our path planning module (Sec. III-C), the RRT grows only within the traversable area of the elevation map, therefore the collection of scan candidates \mathcal{C} does not contain invalid or unreachable poses. As a result, the utility value \mathcal{U}_c of each scan candidate is penalised based on the nature of ANYmal and the configuration of the LiDAR system.

1) *Position Cost*: The position cost \mathcal{P}_c is defined as:

$$\mathcal{P}_c = \begin{cases} 1 - d_{thres}^{-1} \times d_c & d_{thres} \geq d_c \geq 0, \\ 0 & d_c > d_{thres}, \end{cases} \quad (7)$$

where d_c is the distance to an already visited scanning pose or the object itself, and d_{thres} is a user defined threshold.

\mathcal{P}_c is used to avoid rescanning a previously used region and maintains a reasonable distance between robot and object.

Using *Occlusion Aware VI*, the system plans NBVs in regions where the robot can observe more void space. The main contribution to the information gain \mathcal{G}_c is from void rays r_{void} — rays that do not hit any surfaces. Voxels $v_{void} \in \mathcal{V}_{r_{void}}$ are mainly unknown (occupancy probability $P_o(v_{void}) = 0.5$) and have high entropy.

In our current system, $P_o(v)$ of a voxel $v \in \mathcal{V}_r$ is only updated when ray r hits a surface, so $P_o(v_{void})$ does not change when observed by r_{void} . As a result for *Occlusion Aware*, $\mathcal{I}_{OA}(v_{void})$ will not decrease, causing the robot to stop exploring. By applying the position cost, our system can also avoid visiting fully scanned areas. We plan to utilise r_{void} to update voxel occupancy in the future version.

In contrast, r_{void} do not contribute to *Rear Side Entropy VI*. Every scan decreases the entropy of observed voxels.

In addition, the position cost \mathcal{P}_c that applies to c close to the object encourages candidates farther away, resulting our system observing a wider view. Conversely, if a scan candidate pose c is farther away, less rays in \mathcal{R}_c are able to observe the object, hence IG of this pose \mathcal{G}_c would be lower compared to closer candidates. This discourages our system from selecting distant NBVs, ensuring a high resolution scan.

Isler *et al.* [5] predefined a set of scan candidates in their system so that the distance of scan poses to the object surface was fixed. However, in our system, the distance between the robot and the scan surface is dynamic so that the robot can avoid obstacles in the environment. Furthermore, since the ANYmal operates on a 2.5D manifold, it is necessary for the quadruped to adjust the distance to the object surface so as to efficiently scan objects of different sizes. By combining IGs with a position cost, our system achieves a balance between coverage and resolution.

2) *Traversal Cost*: The traversal cost \mathcal{T}_c represents the difficulty for the robot to execute a certain path to candidate c because of the roughness of terrain and the distance.

Currently our approach classifies the elevation map discretely as either safe ($\mathcal{T}_c = 0$) or not traversable ($\mathcal{T}_c = 1$).

In addition, a constant traversal cost penalises scan candidates that are behind the robot, because large turns are more difficult for the robot to execute. This policy also encourages the robot to explore forward rather than alternating direction. This makes the system more time and energy efficient.

C. Path Planning

The path planning module in our system consists of two phases, as indicated in Fig. 2. Both phases rely on an elevation grid map generated from LiDAR measurements of the environment. We used the approach of [21] to compute the slope and normal of each cell and in turn a measure of the traversability of the terrain. The traversability is used to determine which states planned by the RRT and RRT* are valid and reachable.

In the first phase, the RRT grows into the traversable area without a goal until a user-defined number of nodes have been generated. These nodes are scan candidates \mathcal{C} . We then

compute the utility value \mathcal{U}_c for each scan candidate $c \in \mathcal{C}$ independently and choose the NBV c_{best} with the highest individual value. Following that, the second phase of our path planning module uses RRT* to replan the route to NBV, optimising travel distance.

D. Termination Condition

In a model-free active mapping system, it is difficult to evaluate the completeness of reconstruction. We terminate operation using a user-defined threshold on the utility value u_{thres} after a planning sequence.

When the utility value of the NBV $\mathcal{U}_{c_{best}}$ falls below the threshold (Eq. (8)), no new scan candidate has satisfactory quality, and the active mapping procedure terminates.

$$\mathcal{U}_{c_{best}} < u_{thres} \quad \forall c \in \mathcal{C}. \quad (8)$$

IV. EXPERIMENTS AND EVALUATION

To demonstrate our system’s functionality and to test the VI formulations, we carried out evaluations of increasing complexity—with the simple virtual models in Fig. 4 and Gazebo reconstructions of our envisaged test locations to test our system’s ability to avoid collisions. Finally, we deployed our system on the real ANYmal robot in these environments. The results are detailed in the following sections.

The real experiments involved scanning a building facade at Green Templeton College (GTC) ($10 \times 35 \times 4 \text{ m}^3$) in Oxford (Fig. 3) and a mock-up helicopter on the oil rig training site at the Fire Service College (FSC) ($3 \times 8 \times 3 \text{ m}^3$) in Gloucestershire (Fig. 1).

In these experiments, we used a LIDAR localization system running on the robot’s navigation PC. The system registered LiDAR clouds against a prior point cloud map using Iterative Closest Point (ICP) [22] seeded with legged odometry. At the scale of our experiments, a deformable map representation was not needed.

A. Hardware

1) *Platform*: The robot platform employed in this work is an ANYbotics ANYmal (version B) [23]. The robot has 12 actuated joints, as well as the 6 DoF floating base link. It is capable of trotting at the maximum speed of 1.0 m/s and traversing complex terrain, e.g. stairs, kerbs and ramps.

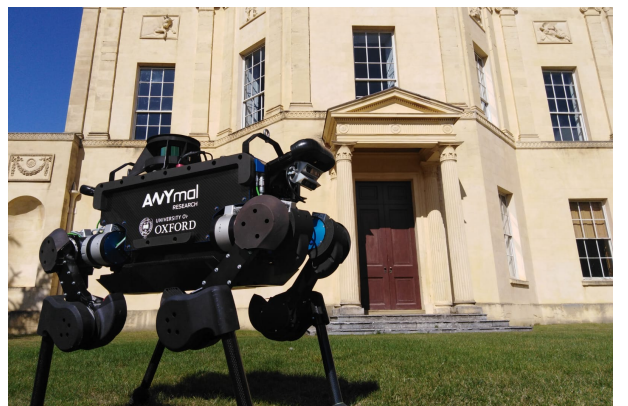


Fig. 3. One of our experiments in Sec. IV mapped this building facade at Green Templeton College (GTC), Oxford.

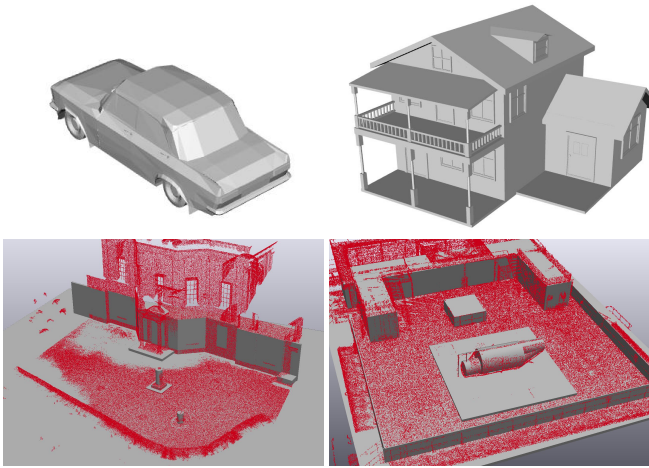


Fig. 4. 3D models of a car and house (top) and of our test sites (bottom) are used to evaluate our system. We created the 3D models using reconstructions made using a survey-grade LiDAR (in red).

2) *Sensor*: The primary sensor of our system is a Velodyne VLP-16 LiDAR which has 16 laser beams spread across a $\pm 15.0^\circ$ vertical FoV and measures ranges with the accuracy of ± 3 cm across full 360° horizontally.

Utilising the robot’s wide range of motion, we designed a scanning action to roll the base from 40° to -40° (while standing). The action improves the vertical FoV of our system to $\pm 55^\circ$ and allows mapping objects much taller than the robot. Using this action, our system collects individual LiDAR sweeps.

B. Simulated Experiments

We conducted experiments in simulation to map models of a car and a house (Fig. 4 (top)). We then used a Leica BLK360 laser scanner to create accurate reconstructions of our two test sites, the facade of a building and a helicopter deck ((Fig. 4 (bottom)). We modelled the major surfaces of these test sites to create Gazebo simulations of the test sites.

In these experiments, the approximate location and size of the object of interest are known, which aids segmentation from the accumulated sweep. This informs our system about where the OctoMap should be constructed and the VIs be computed. We chose a 5 cm resolution for our OctoMap octree, which suits the resolution of the Velodyne LiDAR.

For path planning and NBV selection, our system grows an RRT up to 150 nodes, every iteration, within a 12×12 m² elevation map centred around the robot. This allows the robot to plan and conduct mapping actions around the object.

To quantify the mapping results, we exploit different criteria including point cloud coverage (c_p), travel distance (d_t) and number of scan actions (n_s). In addition, we compute the overall task time (t_{all}) as well as the average time per-scan spent computing information gains and determining the NBVs (t_{nbv}) to evaluate the system’s online feasibility.

To compute point cloud coverage, we aligned the accumulated point cloud with the ground truth and determined the points in the accumulated cloud within 4.3 cm of the nearest point in the ground truth, approximately the distance between

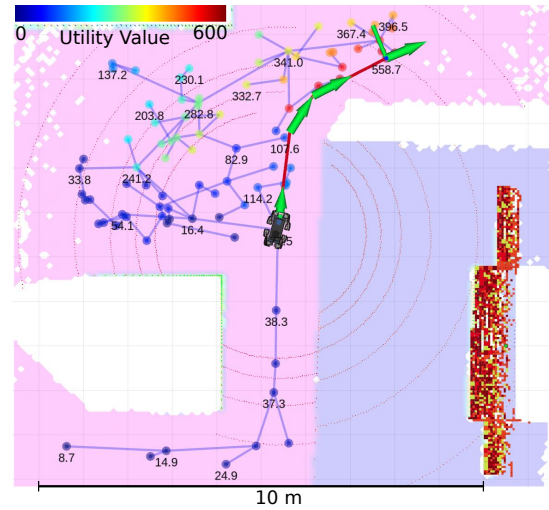


Fig. 5. Illustration of the proposed system mapping the simulated model of the helicopter/oil rig site. The system grows an RRT around the current pose of the robot in the traversable area. The utility metric is computed at the tree nodes. The node with the maximum utility is selected as the NBV. Finally, using the RRT* algorithm, the path from the current pose to the NBV is replanned.

the centre of our OctoMap voxel and its vertex ($\frac{\sqrt{3}}{2} \times 5$ cm). These points are classified as observed.

Point cloud coverage c_p is then defined as:

$$c_p = \frac{N_O}{N_{GT}} \quad (9)$$

where N_{GT} and N_O are the total number of points in the ground truth model and the number of points observed in the model so far, respectively.

As summarised in Table I, the point cloud coverage gained with *Occlusion Aware* IG is slightly higher than that with *Rear Side Entropy* IG. This can also be seen in Fig. 6, which demonstrates the point cloud coverage per scan. The maximum coverage never reaches 100% in our system as the top surfaces of objects are higher than the robot and cannot be observed from the ground, as shown in Fig. 8.

While there is on average an 8.5% reduction in travel distance when our system employs *Occlusion Aware* compared to *Rear Side Entropy*, our system is subject to the random scan candidate placement by RRT. Hence the performance difference between two VIs in simulation so far is not significant enough for us to make a conclusive decision on which is the better formulation. Both approaches allowed our system to accomplish the mapping task. In both cases, the travel distance, the overall run time and the NBV computation time are all feasible for real experiments.

C. Real-World Experiments

Based on the simulated results in the previous section, we used the *Occlusion Aware* VI gain metric in our real experiments.

Table II summarises the evaluation of the reconstruction results in both experiments. In these, because our elevation map is partly corrupted by odometry noise (see attached video) and the LiDAR sensor is just 70 cm from the ground, we can only plan in a 7×7 m² area around the robot. We

Object	IG	Experiments Evaluation				
		c_p (%)	d_t (m)	n_s	t_{all} (mm:ss)	t_{nbv} (sec)
Car	OA	69.69	35.39	8	08:02	2.70
	RSE	69.01	40.18	10	10:21	2.46
House	OA	59.41	42.89	12	10:06	3.78
	RSE	58.98	44.64	12	11:17	3.65
Facade	OA	95.11	33.98	9	07:44	17.82
	RSE	94.24	37.82	9	08:21	19.69
Helicopter	OA	83.76	41.56	12	12:26	5.45
	RSE	83.01	43.61	13	13:06	5.20

TABLE I

COMPARISON BETWEEN TWO VI MEASURES IN SIMULATION ENVIRONMENTS (OA - Occlusion Aware; RSE - Rear Side Entropy).

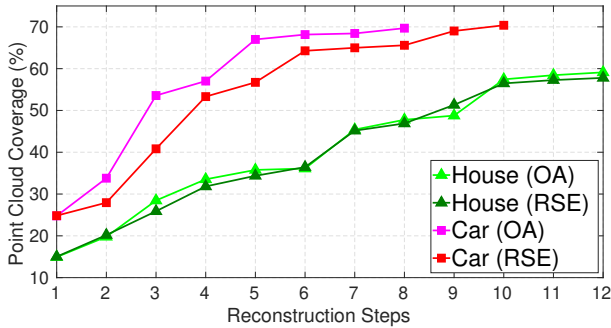


Fig. 6. Point cloud coverage per step for the car and house models.

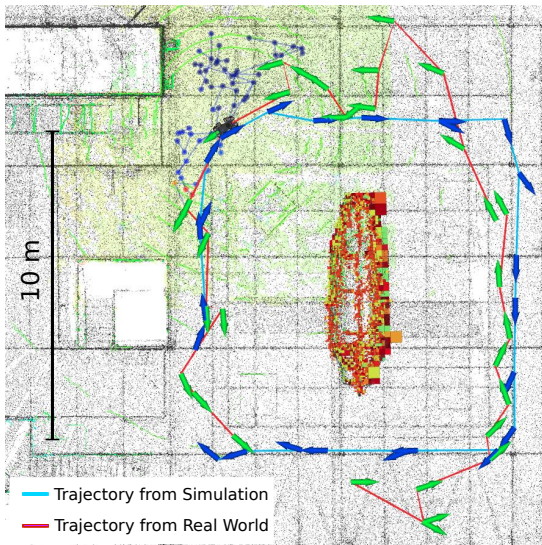


Fig. 7. Example of our system mapping the real helicopter. The route the robot took in the real experiment is shown in red with the reconstruction of the helicopter (as an octree) shown in the centre. Also illustrated, in blue, is the route taken by our method running in simulated model, as a comparison.

therefore decreased the number of RRT scan candidates from 150 to 75, consequently decreasing the computation time of determining the NBV t_{nbv} .

Comparing Table II with Table I, one can see that the computation times for the real experiments at facade and helicopter locations are on average half of the time taken in simulation.

Our approach allows the robot to avoid the mobility hazards for the helicopter experiment in Fig. 1: stairwells, open edges on the deck and a skirt around the helicopter.

Fig. 7 demonstrates a comparison between the robot trajectories in the real FSC helicopter experiment (counter-

Object	Experiments evaluation				
	c_p (%)	d_t (m)	n_s	t_{all} (mm:ss)	t_{nbv} (sec)
Facade	88.06	37.61	15	20:56	9.82
Helicopter	78.60	49.19	26	35:25	2.00

TABLE II

RESULTS FOR OUR SYSTEM IN THE REAL-WORLD EXPERIMENTS.

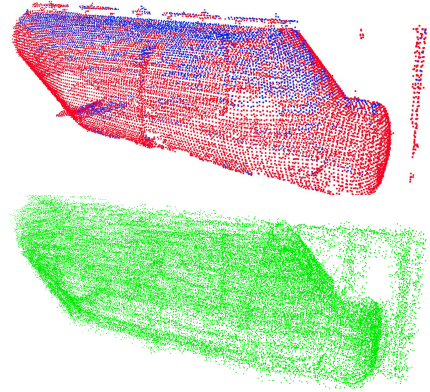


Fig. 8. Top: Point cloud coverage in the real helicopter experiment. Red cloud indicates the observed area and blue represents the unobserved part of the helicopter. Bottom: Our system succeeded in reconstructing an accurate model of the helicopter body.

clockwise) and in simulation (clockwise). The paths taken by our system in both scenarios are similar, demonstrating the practicality of our system in real scenarios. However, as noted before, because of more unknown areas existing within the elevation map, the system had to plan around void cells.

Fig. 8 demonstrates the success of our system in reconstructing the helicopter body (in green), compared to the ground truth (in red). Due to the limitation in the elevation map and in turn our path planning module, our system planned more scans than in simulation.

In the helicopter scenario, the number of scans required more than doubled and as a result the run time almost tripled. A major part of that difference is due to the time spent by the robot operator judging if planned paths were safe.

V. CONCLUSION AND FUTURE WORK

In this work we presented a model-free active mapping system using a volumetric representation. The system allows a quadruped robot to explore and reconstruct both small and large scale objects, in particular industrial assets, with few assumptions about the test environment and requiring only high level human supervision. We tested our system in fully realistic scenarios and our approach allowed the robot to accomplish mapping missions in a complicated environment, creating accurate reconstructions online.

In the future, we plan to improve the quality of elevation map and to incorporate full traversability estimation to allow our robot to navigate over rough terrain, such as kerbs and ramps, so as to fully utilise the dynamics of a legged robot.

In addition, we plan to base localisation on a pose-graph SLAM system [24] so that our active mapping approach can explore larger environments, with the benefit of loop closure. Hence, we would like to modify the reconstruction to be deformable, as in the manner proposed by Ho *et al.* [7].

REFERENCES

- [1] G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," *Intl. J. of Robotics Research*, vol. 32, no. 1, pp. 3–18, 2013.
- [2] S. Kriegel, C. Rink, T. Bodenmu, and M. Suppa, "Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects," *J. of Real-Time Image Processing*, vol. 10, pp. 611–631, 2015.
- [3] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3D exploration and surface inspection," *Autonomous Robots*, vol. 42, no. 2, pp. 291–306, 2018.
- [4] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3D object reconstruction," *Autonomous Robots*, vol. 42, no. 2, pp. 197–208, 2018.
- [5] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3D reconstruction," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2016, pp. 3477–3484.
- [6] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "View planning for 3D object reconstruction with a mobile manipulator robot," *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 4227–4233, 2014.
- [7] B.-j. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, and M. Kaess, "Virtual Occupancy Grid Map for Submap-based Pose Graph SLAM and Planning in 3D Environments," *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, vol. 1, no. 2, pp. 2175–2182, 2018.
- [8] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *Intl. J. of Robotics Research*, vol. 31, pp. 1445–1464, 2016.
- [9] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter, "ANYmal in the field: Solving industrial inspection of an offshore HVDC platform with a quadrupedal robot," in *Field and Service Robotics*, 2019.
- [10] M. Karaszewski, M. Adamczyk, and R. Sitnik, "Assessment of next-best-view algorithms performance with various 3D scanners and manipulator," *ISPRS J. of Photogrammetry and Remote Sensing*, vol. 119, pp. 320–333, 2016.
- [11] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *Intl. J. of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.
- [12] P. S. Blaer and P. K. Allen, "Data Acquisition and View Planning for 3-D Modeling Tasks," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2007, pp. 417–422.
- [13] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon next-best-view planner for 3D exploration," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2016, pp. 1462–1468.
- [14] S. Kriegel, C. Rink, T. Bodenmüller, A. Narr, M. Suppa, and G. Hirzinger, "Next-best-scan planning for autonomous 3d modeling," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct 2012, pp. 2850–2856.
- [15] S. Kriegel, M. Brucker, Z. C. Marton, T. Bodenmuller, and M. Suppa, "Combining object modeling and recognition for active scene exploration," *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2384–2391, 2013.
- [16] K. Schmid, H. Hirschmüller, A. Dömel, I. Grixa, M. Suppa, and G. Hirzinger, "View planning for multi-view stereo 3D reconstruction using an autonomous multicopter," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 309–323, 2012.
- [17] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," *Technical Report*, vol. 129, pp. 98–11, 1998.
- [18] J. I. Vasquez-Gomez, L. E. Sucar, R. Murrieta-Cid, and E. Lopez-Damian, "Volumetric next-best-view planning for 3D object reconstruction with positioning error," *Intl. J. of Advanced Robotic Systems*, vol. 11, 2014.
- [19] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.
- [20] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Intl. J. of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [21] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 3019–3026, 2018.
- [22] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, Apr. 2013.
- [23] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal - a highly mobile and dynamic quadrupedal robot," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016, pp. 38–44.
- [24] M. Ramezani, G. Tinchev, E. Iuganov, and M. Fallon, "Online lidar-slam for legged robots with robust registration and deep-learned loop closure," *arXiv preprint arXiv:2001.10249*, 2020.