A Sliding Window Approach to Exploration for 3D Map Building Using a Biologically Inspired Bridge Inspection Robot

Gavin Paul, Phillip Quin, Andrew Wing Keung To, Dikai Liu
Center for Autonomous Systems
University of Technology Sydney, Sydney 2007, Australia
Gavin.Paul@uts.edu.au, Phillip.D.Quin@student.uts.edu.au, Andrew.To@uts.edu.au, Dikai.Liu@uts.edu.au

***Abstract*—This paper presents a Sliding Window approach to viewpoint selection when exploring an environment using a RGB-D sensor mounted to the end-effector of an inchworm climbing robot for inspecting areas inside steel bridge archways which cannot be easily accessed by workers. The proposed exploration approach uses a kinematic chain robot model and information theory-based next best view calculations to predict poses which are safe and are able to reduce the information remaining in an environment. At each exploration step, a viewpoint is selected by analysing the Pareto efficiency of the predicted information gain and the required movement for a set of candidate poses. In contrast to previous approaches, a sliding window is used to determine candidate poses so as to avoid the costly operation of assessing the set of candidates in its entirety. Experimental results in simulation and on a prototype climbing robot platform show the approach requires fewer gain calculations and less robot movement, and therefore is more efficient than other approaches when exploring a complex 3D steel bridge structure.**

I. INTRODUCTION

It is estimated that there are more than 42,000 steel bridges in the European Union, 270,000 in the United States [1] and an estimated 4,000 in Australia. Regular inspection and maintenance of bridges present considerable challenges to workers due to the threat of moving traffic, and working both at heights (Figure 1a) and in confined spaces (Figure 1b). Advances in robotics and sensor technology are beginning to make it possible to automate increasingly complicated tasks, such as robotic surface inspection [2]. However, traversing a steel bridge environment requires that a robot be able to transfer between surfaces with all possible orientations, negotiate rivets and go through small manholes. A 7DOF inchworm manipulator design [3] with magnetic footpads for climbing steel structures has been developed and field-tested in steel bridge environments.

There are many challenges associated with placing a robot in an unknown environment to autonomously perform an inspection task. Generally in industry, robotic manipulators are used in controlled and known areas where infrastructure is built around them or tasks are brought to them. However, in the case of a bridge inspection task, it is not possible for the environment to be restructured for the robot. Thus, the robot must have the capability to generate a map of the environment for two purposes: (a) to enable the robot to further explore while avoiding collisions and (b) to convey information about the geometry and condition of surfaces back to the inspector for further analysis and storage in an asset management system.

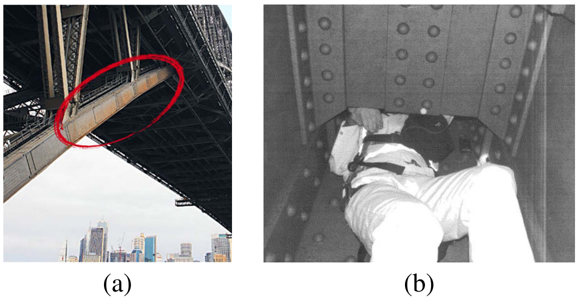


Fig. 1. (a) Condition inspection required inside the arch indicated; (b) Worker performing inspection inside the confined space of the archway.

In order for a map of the environment to be useful for robot motion planning and surface inspection, the map needs to accurately encapsulate the 3D surface geometry. The map also needs to contain information about the occupancy state (i.e. existence of surfaces) and allow for the integration of surface condition and material-type information. Therefore, an approach is required to safely, efficiently and autonomously explore and build a 3D map of an initially unknown area, such as the environment in Figure 1b. It is important to explore in such a way that the map is generated online and is a high-quality 3D representation that accurately represents the up-to-date state and layout of the environment.

II. RELATED WORK

There has been significant interest in robot mapping in the robotics community for many years. In particular, SLAM has been used for building small and large scale maps [4]. Exploration and geometry mapping in 2D, 2.5D or 3D using a mobile robot equipped with sensors such as laser range finders and stereo cameras is a extensively researched field [5]. Often the focus of such work is a robot moving on a 2D plane to build 2.5D or 3D maps of an environment [6], or the focus is localisation [7] rather than on generating highly-accurate 3D maps of the geometry of surfaces.

Exploring the surfaces in a 3D environment using a sensor mounted on a robot manipulator is challenging since the sensor needs to be positioned anywhere in the 3D environment to collect detailed information about the geometry and surface condition, while keeping both the sensor and the manipulator's kinematic chain in freespace that is known to be empty (i.e. unoccupied). In the specific scenario addressed in this paper, it is necessary to determine a sequence of discrete viewpoints where the manipulator can position a sensor to gather information for generating a 3D map with the required quality. The robot manipulator movements used to position the sensor in a partially known environment must be safe and efficient, so as to avoid potential collisions with known objects or objects that may be present in currently unknown regions.

Mapping techniques using robot manipulators [8], [9], [10] to produce quality surrounding maps or part models, typically have pre-programmed robot movements or have historically been time consuming. When a manipulator's configuration space is explored [11], the exploration process facilitates path planning by sensing the surrounding space that affects the ability to plan safe paths - resulting in the discovery of the occupancy of space in the immediate vicinity of the robot manipulator without offering a guarantee about map completeness [12]. Coverage and next best view (NBV) algorithms for manipulators in obstacle-free and known environments need to be extended to address the problem of gathering information while also considering collision-free exploration. One such extension was presented in [13] with a fixed base eye-in-hand manipulator setup being used to identify the structural elements in an environment. This approach has three drawbacks that make it unsuitable for the current mobile system: 1) it relies upon a large number of information gain calculations to be computed at every iteration; 2) viewpoints are selected without considering possible time-consuming robot joint movements; and 3) the sensor minimum range issue is disregarded because of the sensing hardware.

This paper presents the Autonomous eXploration to Build a Map - Sliding Window (AXBAM_SW) approach that extends the manipulator-based exploration algorithm from previous work on selecting viewpoints to map a surrounding environment [13]. Instead of using an exhaustive search to find a suitable viewpoint, a sliding window is used to first select a subset of candidate viewpoints. The approach incorporates a sensor's minimum range for improved information gain predictions, as well as selecting viewpoints in nearby configuration space to reduce a robot's joints movement. AXBAM_SW's information gain calculator can also enable a planner to compare potential robot base locations for exploration. The remainder of this paper is organised as follows. Section III details the proposed approach to predicting exploration viewpoints, planning the movements, and generating a map. Section IV presents experimental results using both simulated and data collected in our laboratory. Finally, Section V provides concluding remarks.

III. METHODOLOGY

Prior research [13] into an exploration approach for exploring with a fixed-base industrial robotic manipulator found that an information theoretical approach to exploration could be used to estimate the potential geometric information gain given a sensor's position and orientation. A brief overview of exploration information theory is presented henceforth.

An environment to be explored can be divided into n_v equally sized, independent voxels (volumetric pixels), analogous to "occupancy grids" that can be used to generate maps given incomplete sensor data [14]. Each voxel has a 3D position in space and a probability of occupancy. A voxel's probability of occupancy will take one of the three states: freespace, unknown and occupied (i.e. containing a surface). Where ${}^jX$ is a discrete state variable, the probability that the jth voxel is occupied is contained in a tristate buffer, $P({}^jX = {}^jx_o)$ for the voxels, $j \in \{1, \dots n_v\}$, and states, $o \in \{\text{free}, \text{unknown}, \text{occupied}\}$. Therefore the probability of occupancy is $P({}^jx_o) \in \{0, 0.5, 1\}$, if the jth voxel's occupancy state, ${}^jx_o$ is respectively {freespace, unknown, occupied}. The occupancy state of freespace and occupied are complementary (i.e. the probability of freespace is $1-P({}^jx_o)$).

A standard way to measure information is with the mathematical notion of entropy which describes the information which remains to be discovered. Initially all voxels, except those in the volume occupied by the robot, are unknown ($P({}^jx_o) = 0.5$). In accordance with Shannon's entropy the measure of geometric information, $H({}^jX)$, can be taken of a voxel where ${}^jX$ is a discrete state random variable that can take on the occupancy states. Since the 'unknown' state occurs where there is an equal probability of the state being free or occupied, the state space is effectively reduced to two states (i.e. a Bernoulli distribution model) and binomial entropy is used to calculate the information in the jth voxel:

$$\begin{aligned} H({}^jX) = -P({}^jx_o) \log\left(P({}^jx_o)\right) - \\ \left(1 - P({}^jx_o)\right) \log\left(1 - P({}^jx_o)\right) \end{aligned} \tag{1}$$

As the environment is discretised into n_v voxels with a probability of occupancy $P({}^jx_o)$, the geometric information remaining, $\mathbf{H}(\mathbf{X})$, in all voxels with occupancy state vector, $\mathbf{X}$, can be used as a measure of cumulative information remaining in all n_v independent voxels.

$$\mathbf{H}(\mathbf{X}) = \sum_{j=1}^{n_v} H({}^jX) \tag{2}$$

The information remaining is zero only when there is no uncertainty (i.e. $P({}^jx_o) = 0$ or $P({}^jx_o) = 1$). Therefore the information remaining measure, $\mathbf{H}(\mathbf{X})$, is directly related to the number of unknown voxels, n_u.

Exploration by a robot at base location, ${}^0T_b$, requires predictions to be made in order to select a viewpoint which minimises the remaining occupancy information. Given an nDOF manipulator pose, $\mathbf{q} = [q_1, q_2, \dots q_n]^T$ and using forward kinematics, it is possible to compute the position/ orientation of the end-effector where a sensing tool, such as an Asus Xtion depth camera is mounted. An end-effector position/orientation in base coordinate frame can be expressed as a homogeneous transformation matrix, ${}^bT_f(\mathbf{q})$, and computed by performing transformations using joint angles, $q_i \forall i \in \{1, \dots n\}$ as,

$${}^bT_f(\mathbf{q}) = \prod_{i=1}^{n} {}^{i-1}T_i(q_i) \tag{3}$$

The transformation matrix between the end-effector and the sensor is denoted ${}^fT_s$. Together these two matrices describe the viewpoint in global coordinate frame, ${}^0T_s(\mathbf{q}) = {}^0T_b {}^bT_f(\mathbf{q}) {}^fT_s$. Henceforth, $\mathbf{q}$ will refer to both a particular robot pose and the resulting viewpoint.

The unknown voxel's volume of space which can be sensed, is equivalent to $\mathbf{H}(\mathbf{X})$ and is denoted, $V_{new}(\mathbf{q})$ since

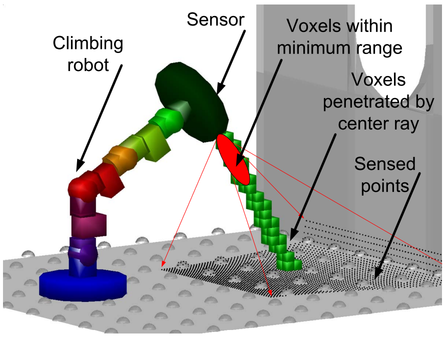


Fig. 2. Using robot and sensor models, the information gain calculation can predict voxels that will be penetrated by rays cast from the sensor - and thus be declared free after the scan. Only the penetration of voxels by the center ray is shown, along with points on the surface that will be ray-cast to.

this volume is a function of the sensor location. The prediction of $V_{new}(\mathbf{q})$ can be determined by ray casting from the sensor into the partial map. Figure 2 shows how for each of the sensor's depth pixel values, a ray is cast from the proposed sensor viewpoint, ${}^0T_s(\mathbf{q})$, through the partially known map to the extent of the sensing range. Voxels that are penetrated within the sensor's minimum range are discarded. If the ray doesn't intersect with a known surface, all unknown voxels (beyond the minimum range) which the ray passes through (up to its maximum range) are included in the predicted volume, $V_{new}(\mathbf{q})$. If the ray intersects with one or more surfaces then only the unknown voxels between the sensor and the closest surface intersection point are included in $V_{new}(\mathbf{q})$. Thus, the geometric information reduction can be predicted for each $\mathbf{q}$.

When selecting the next viewpoint it is necessary to balance the predicted information gain with the effort needed to reposition the sensor. Movements between exploration poses must be safe; i.e. avoid any intersection (collision) of the robot with obstacles or with unexplored space. In the eye-in-hand 3D exploration scenario where a sensor is mounted to a manipulator's end-effector, both the manipulator's kinematic chain and the sensor need to be within known freespace. The entire robot must remain within known freespace throughout the entire path whereby the manipulator moves from the current pose to the next viewpoint. Planning a manipulator's trajectory between two poses is a well established problem even with many obstacles in the environment. An established bidirectional RRT planner [15] computes multiple paths, shortens plans with Voronoi path shortening, then selects the shortest plan consisting of a joint movement trajectory. Collision-free poses and plans can be achieved by performing safety checks in a map using ellipsoidal virtual bounding fields [16]. The effort required to move between the current pose and candidate NBV poses is thus calculable and available for consideration by the exploration algorithm.

Figure 3 shows how, at each iteration of the exploration process, the currently available map is analysed so as to determine the next safe viewpoint. The next pose is chosen based upon the predicted potential to reduce the information remaining as well as the effort required to move to this pose. Using the information gain and the joint movement from the planner, the poses are ranked and then analysed. Attention is restricted to poses that are Pareto efficient with regard to gain and effort. A pose, $\mathbf{q_{nbv}}$, is chosen from the Pareto frontier by selecting the pose with the highest ratio of gain over effort. Once the new data has been acquired by scanning at the viewpoint, the map is updated.

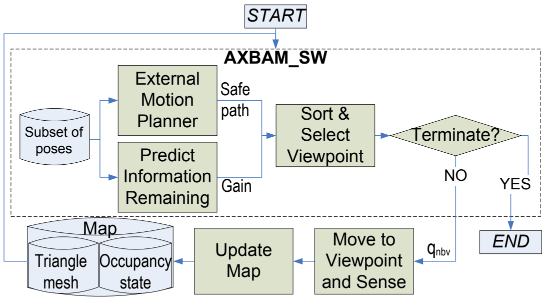


Fig. 3. The exploration approach includes information gain calculations and motion planning to select next viewpoints.

Exploration gain calculations, which ray cast into a map, have been shown to be time consuming [13]. To avoid intractable numbers of these gain calculations inherent to an exhaustive viewpoint search, it is necessary to select a subset of n poses, $Q = \mathbf{q_i} \forall i \in n$, such that the small pose set is closely representative of all possible poses without being intractable. In [13] a greedy approach was adopted which maximised the information gain for the entire set of poses, however this was found to be inefficient, especially when (as is the case in this application) the robot base is mobile and the robot's DOF count is increased. Alternatively, the nearest neighbour could be selected by analysing the pose set and moving to the nearest pose in configuration space (C-space) [17]. Randomly selecting a safe pose from the set would require no gain calculations but can be expected to be inefficient since it is undirected.

Therefore, this paper makes several important improvements so as to: (a) utilise the sensor model to assist with information gain estimations such that the poses in Q are ranked according to ideal information gain; (b) utilise the robot model so as to compute and avoid poses which are in collision with obstacles; and (c) a sliding window approach to select a subset of n_{max} Pareto optimal viewpoint poses, Q_{sw} to perform gain calculations on, and planning to (rather than using the entire pose set, Q). During the exploration approach, poses are removed from the set Q if they are: (1) selected for use; or (2) found to result in a collision with obstacles in the environment; or (3) predicted to have less than an information gain threshold, V_{min}. Algorithm 1 shows the details of AXBAM_SW's selection of the NBV pose, $\mathbf{q_{nbv}}$, given the current robot pose, $\mathbf{q_{curr}}$, the map, M, and the sorted candidate pose database, Q.

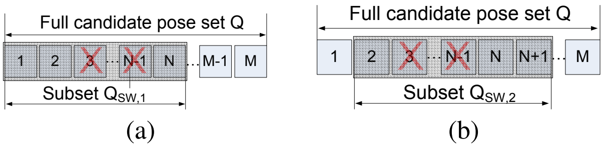


Fig. 4. Sliding window showing the subset of poses which are to be analysed at an iteration of AXBAM_SW. Note unsafe, already used or poses with little information gain are crossed. *a*) First subset; *b*) Second subset after pose 1 is used to scan.

A subset of poses in a sliding window of interest is determined, and each of these evaluated, rather than the entire set. The sliding window consists of a subset of poses such as $Q_{sw,1}$ (Figure 4a) from the ordered pose set, Q, where the poses in $Q_{sw,1} \subset Q$ are analysed, i.e. have their information gain estimated. Then the robot's joint rotation angle difference from the current pose, as well as the predicted information gain are jointly considered. The next time the viewpoint needs to be selected from the updated Q, the sliding window is moved, as shown by $Q_{sw,2}$ (Figure 4b). Sliding the window over just the best subset of the sorted Q overcomes the inefficiency drawback of the previous method when selecting $\mathbf{q_{nbv}}$ from a large candidate pose set.

Considering only a subset of the poses is an improvement over the exhaustive search (that maximises the information gain) in two regards: firstly, since the C-space distance is considered, poses are expected to be closer together and thus require less movement of the robot. Secondly, since it is a subset it requires significantly fewer gain calculations than exhaustively checking the entire set. It was found empirically that sliding windows size, n_{max} of 1% of the total pose set (equivalent to several hundred) is desirable in most cases. The AXBAM_SW algorithm terminates once all candidate poses have been considered, and have either been used, are are determined unsafe or can not provide an adequate amount of new information.

```
Input: $\mathbf{q_{curr}}, M, Q$
Output: $\mathbf{q_{nbv}}, path, Q$
$\mathbf{q_{nbv}} = \emptyset$, $V_{new} = \emptyset$;
$Q = RemoveUnsafeOrLowGainPoses(Q, M)$;
if $Q \neq \emptyset$ then
    while $size(Q_{sw}) < n_{max}$ do
        $\mathbf{q} \leftarrow POP(Q)$;
        $gain = G(\mathbf{q}, M)$;
        if $gain < V_{min}$ then
            continue;
        $V_{new}(\mathbf{q}) = gain$;
        $PUSH(Q_{sw}, \mathbf{q})$;
    $rank1 = sort(|Q_{sw} - \mathbf{q_{curr}}|)$;
    $rank2 = sort(V_{new})$;
    $Q_{sw} = ParetoOptimalSet(rank1, rank2)$;
    while $0 < size(Q_{sw})$ do
        $\mathbf{q} \leftarrow POP(Q_{sw})$;
        $[valid, path] = PlanPath(\mathbf{q_{curr}}, \mathbf{q}, M)$;
        if $valid$ then
            $\mathbf{q_{nbv}} = \mathbf{q}$;
            return
if $Q_{sw} == \emptyset$ then
    $TerminateExploration$
```

Algorithm 1: AXBAM_SW Algorithm

To support fusion of multiple scans into surface location estimates, an adaptive distance field map representation (volumetric technique) based on [18], is implemented as a real-time solution. The distance field is also used to store the occupancy state of voxels in the map. The output of this iterative data fusion process is a map which can be rapidly queried. Once exploration is completed locally at a base location, the normals on the generated surface map can be sampled. The variation in normals and the continuity of the surfaces is analysed using previous work on surface segmentation [19] to find holes, surface edges, and surface perturbations. Areas near the boundaries of surfaces are considered as potential future base locations. Using the presented gain calculation technique on the state of the map (i.e. the unknown, unoccupied and occupied voxels), it then becomes possible to estimate the potential knowledge that can be gained from a candidate base location.

IV. Experiments

In order to test the presented approach, two experiments have been conducted. Experiment 1 presents the exploration of two simulated bridge archway tunnel environments in comparison with alternative methods for selecting the next viewpoint. Experiment 2 presents results from exploring a real environment built with the dimensions of the target environment, and with similar surface materials and rivets. The simulations and control have been run on an Intel Core i7 quad-core (2.7 GHz) with 4 GB of RAM. Both experiments are conducted using the same prototype 7DOF degree of freedom climbing robot model. All joints are revolute, and the Denavit-Hartenberg parameters of the robot are $\alpha = \pi/2$ for odd joint numbers and $\alpha = -\pi/2$ for even joints, and $d = [0, 0.14, 0, 0.12, 0, 0.14, 0]$. The minimum and maximum ranges of the joint motion are $\pm 90^o$ for all joints except 1 ($[q_{1,min}, q_{1,max}] = [-110, 90]^o$) and joint 7 ($[q_{7,min}, q_{7,max}] = [-110, 90]^o$). The initial manipulator pose is the same for all experiments. Experiment 2 used a prototype robot and takes scans using an eye-hand calibrated Asus Xtion sensor attached to the end-effector.

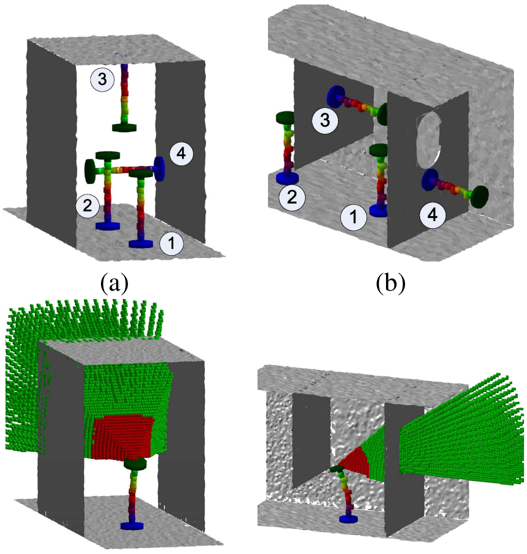
A. Experiment 1: Exploring Simulated Environments

In this experiment the Method 1: AXBAM_SW approach is compared to three other exploration methods. These are Method 2: Random Poses, Method 3: Maximum Information Gain [13], Method 4: C-Space Discretised. A large pose set is initially constructed offline by sampling each joint 5 or 6 times in their range. The angle between each joint is chosen to match the sensor's vertical FOV. Poses that would cause the robot to collide with itself or the base are discarded. A pose set, Q of approximately 50,000 candidate poses are therefore available for each method to select the next viewpoint pose from, and for AXBAM_SW the sliding window size, $n_{max} = 500$. All methods terminate once 95% of the voxels that can be sensed (based upon an exhaustive search), have been sensed.

In Method 2: Random Poses, the poses are selected at random from the candidate set. Although this method does not perform the time-consuming information gain calculations (i.e. the selected poses are suboptimal), Method 2 was still hypothesised to be the least efficient since a large number of poses and robot movements are expected to be required to explore the environment. Method 3: Maximise Information Gain is based upon the previously proposed exploration algorithm [13]. At every viewpoint every possible viewpoint is checked for information gain and the viewpoint that has the largest predicted information gain is selected. This was expected to require fewer viewpoints than the other three methods since each viewpoint is selected due to the prediction that it will have the largest information gain. However, this method has previously been shown to be time consuming due to the number of gain calculations necessary. Method 3 also does not consider the robot motions required. Thus, it is expected that the robot movements required per viewpoint will be significant. Method 4: C-Space Discretised uses a predetermined set of configuration space (C-Space) poses that are uniformly distributed within the robot's C-Space. Each of these poses is sequentially iterated through.

Two simulated environments (Figure 5a and b) are used for this experiment. Figure 5a consists of a simple tunnel environment with 2 walls, a floor and ceiling. Figure 5b is more complex, consisting of an open tunnel, with 1 wall, a floor and ceiling, a manhole and a partition plate within the tunnel. Figures 5c and d show the voxels penetrated from a single pose using the information gain calculation technique. Note the voxels from the one scan are in green and the disregarded voxels (within the sensor's minimum range) are in red.

Table I shows the results (e.g. average and standard deviation, $\mu(\sigma)$ of the number of poses, the voxels gained per viewpoint, the computation time, and the movement time) for the two environments with the average of the 4 base locations shown in Figure 5a and b. In this simulation the movement times are calculated based upon the robot joints moving at a maximum of 10RPM. In each environment the robot is placed at 4 different base locations. The stochastic method, Method 2: Random Poses is run 10 times at each of these base positions, so for each environment it is run a total of 40 times and $\mu(\sigma)$ is shown. The three deterministic methods are run once at each of the 4 base locations.



(a) (b)

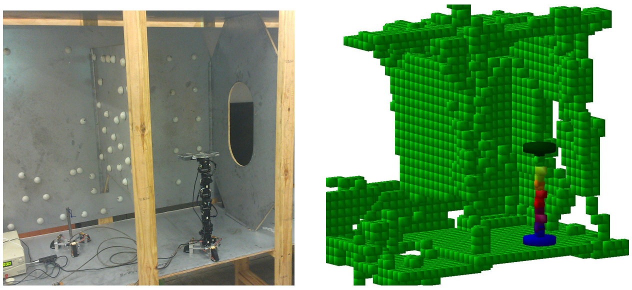
(c) (d)

Fig. 5. The 4 base locations of a single robot in *a*) Environment 1: simple tunnel with walls, roof and floor and; *b*) Environment 2: complex example containing a partial partition and manhole plate; *c,d*) A single exploration scan from a viewpoint at the first base location in each environment.

As predicted, Method 2 requires the least computation time; however, due to the large number of viewpoints required it is the least efficient. Method 3 is shown to require fewer poses on average but has a significant amount of computation time and chooses poses without regard for the robot movements required. In Environment 1, Method 3 has similar overall movements (426 seconds worth) to AXBAM_SW, despite it only using 36 poses on average where Method 1 uses 50 on average. Note also that the σ of the poses required for Method 3 is low since it always requires a similar number of viewpoints to obtain 95% coverage. In Environment 2, Method 1 AXBAM_SW outperforms Method 3 for the 4 base locations. AXBAM_SW required the fewest number of poses and the least amount of robot motion. Also it requires less planning time than all except Method 3: Random (as expected). In general AXBAM_SW is always able to terminate with less iterations and less movement than both Method 2 and Method 4. This is even though on average AXBAM_SW takes slightly longer to determine the next pose, 36 seconds per pose for Environment 1 and 50 seconds per pose for Environment 2.

B. Experiment 2: Prototype Robot Exploring Environment

In this experiment a prototype climbing robot with identical DH parameters to the previous experiment is shown in Figure 6a. In this experiment the exploration approach is tested in a real-world environment. Figure 6b shows a portion of the occupied region of the occupancy map produced through 53 iterations of exploration. Each gain calculation takes between 30 seconds and 1 minute. In total only 4 minutes was required for the robot movements. The voxel resolution is 50mm where in the simulated case it was 100mm. Also the environment size is slightly larger. In total, 93,870 voxels were sensed, of which 61,502 were found to be unoccupied. This closely matched the ground truth model of the environment.



(a) (b)

Fig. 6. *a*) Prototype climbing robot in mock bridge maintenance environment; *b*) Resulting map from exploration.

C. Discussion

As shown in these results AXBAM_SW is able to select exploration viewpoints for a mobile manipulator such that the time to explore is reduced compared to several stochastic and deterministic alternatives. The completeness and resolution of the maps that are generated is high (above 95%). The exploration algorithm succeeds in selecting new and safe viewpoints to move the entire manipulator so as to explore the environment in a minimal number of iterations. The overall outcome of the system is a valid solution for building a map of the internal archway tunnels for bridge condition inspection. These maps can also provide a 3D geometric map which can be integrated with other information including: surface material-type, surface condition, and high resolution color images. It would be be interesting to analyse the effect that the sliding window size has on improving the exploration efficiency. It

Environment 1	Method 1: AXBAM_SW	Method 2: Random Poses	Method 3: Maximum Information Gain	Method 4: C-Space Discretised
Average Poses	50(13)	168(18)	36(6)	130(16)
Voxels Per Pose	75.36	22.42	104.1	28.71
Comp. Time (sec)	1834(586)	783(322)	2961(1403)	1712(840)
Move Time (sec)	492(104)	1931(198)	426(80)	1513(195)
Environment 2				
Average Poses	13(4)	117(64)	22(11)	56(26)
Voxels Per Pose	104.37	11.53	61.42	24.15
Comp. Time (sec)	665(244)	460(183)	872(302)	963(762)
Move Time (sec)	139(32)	1230(236)	284(163)	669(307)

TABLE I. Exploration result in two environments from 4 base positions. Results are rounded and in the form: $\mu(\sigma)$.

is noted that the prototype climbing robot faces issues that mean the end-effector location cannot always be accurately determined. Map alignment corrections or more ridge joints, would theoretically improve the map quality. The method used to determine the next base position can in future be further optimised to ensure that the exploration is complete. There are other potential ways to improve the AXBAM_SW approach such as combining it with a search in the nearest neighbourhood of manipulator's configuration space.

V. Conclusions

This paper has presented AXBAM_SW, an approach to autonomously explore an environment using a depth sensor mounted to the end-effector of a mobile robot. Testing has been conducted in a steel tunnel environment using a biologically-inspired climbing robot. The presented exploration and mapping approach used the end-effector transform calculated from a kinematic chain robot model to project rays into a partial map. Simulation and real-world experimental results were collected which verified the presented algorithm. AXBAM_SW requires fewer sensor scans to observe the environment and planning takes less time than previous approaches. Safe base locations can also be calculated at the extremities of the environment to direct the robot next-base-location exploration plans. Future work will investigate the combination of nearest neighbour searches with the presented algorithm to reduce both the path planning required and the number of gain calculations that must be performed when determining the next viewpoint.

Acknowledgment

This work is supported by the Australian Research Council (ARC) Linkage Grant (LP100200750), the NSW Roads and Maritime Services, and the Centre for Autonomous Systems (CAS) at the University of Technology, Sydney.

References

[1] C. Balaguer, A. Gimenez, and A. Jardon, "Climbing robots' mobility for inspection and maintenance of 3d complex environments," *Autonomous Robots*, vol. 18, no. 2, pp. 157–169, 2005.

[2] M. Eich and T. Vogele, "Design and control of a lightweight magnetic climbing robot for vessel inspection," in *Control Automation, 2011 19th Mediterranean Conference on. MED.*, Corfu, 2011, pp. 1200–1205.

[3] P. Ward, G. Paul, P. Quin, D. Pagano, C. Yang, D. Liu, K. Waldron, G. Dissanayake, P. Brooks, P. Mann, W. Kaluarachchi, M. P., and L. Matkovic, "Climbing robot for steel bridge inspection: Design challenges," in *9th Austroads Bridge Conference*, Sydney, 2014.

[4] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[5] S. Thrun, C. Martin, L. Yufeng, D. Hahnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard, "A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 433–443, 2004.

[6] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust tracking for real-time dense rgb-d mapping with kintinuous," 2012.

[7] D. F. Wolf, G. S. Sukhatme, D. Fox, and W. Burgard, "Autonomous terrain mapping and classification using hidden markov models," in *Proc. IEEE International Conference on Robotics and Automation, ICRA*, Barcelona, 2005, pp. 2026–2031.

[8] V. A. Sujan and S. Dubowsky, "Efficient information-based visual robotic mapping in unstructured environments," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 275–293, 2005.

[9] M. Callieri, A. Fasano, G. Impoco, P. Cignoni, R. Scopigno, G. Parrini, and G. Biagini, "Roboscan: an automatic system for accurate and unattended 3d scanning," in *International Symposium on 3D Data Processing, Visualization and Transmission*, 2004, pp. 805 – 812.

[10] L. Torabi and K. Gupta, "An autonomous six-dof eye-in-hand system for in situ 3d object modeling," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 82–100, 2011.

[11] P. Wang and K. Gupta, "Computing c-space entropy for view planning based on beam sensor model," in *IEEE/RSJ International Conference on Intelligent Robots and System*, vol. 3, Lausanne, 2002, pp. 2389–2394.

[12] M. Fernandez, K. Gupta, and J. Fraile, "Simultaneous path planning and exploration for manipulators with eye and skin sensors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, 2003, pp. 914 – 919.

[13] G. Paul, S. Webb, D. K. Liu, and G. Dissanayake, "Autonomous robot manipulator-based exploration and mapping system for bridge maintenance," *Robotics and Autonomous Systems*, vol. 59, no. 7-8, pp. 543–554, 2011.

[14] P. Stepan, M. Kulich, and L. Preucil, "Robust data fusion with occupancy grid," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 1, pp. 106–115, 2005.

[15] M. Clifton, G. Paul, N. Kwok, D. Liu, and D. L. Wang, "Evaluating performance of multiple rrts," in *Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on*, Beijing, 2008, pp. 564–569.

[16] G. Paul, N. Kirchner, D. K. Liu, and G. Dissanayake, "An effective approach to simultaneous mapping and surface-type identifcation of complex 3d environments," *Journal of Field Robotics*, vol. 26, no. 11-12, pp. 915–933, 2009.

[17] P. Quin, G. Paul, A. Alempijevic, D. Liu, and G. Dissanayake, "Efficient neighbourhood-based information gain approach for exploration of complex 3d environments," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 1343–1348.

[18] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Computer graphics proceedings, annual conference series*, vol. 2006. New Orleans: Association for Computing Machinery SIGGRAPH, 1996, pp. 303–312.

[19] G. Paul, N. Kwok, and D. Liu, "A novel surface segmentation approach for robotic manipulator-based maintenance operation planning," *Automation in Construction*, vol. 29, pp. 136–147, 2013.