

Autonomous Robot Manipulator-based Exploration and Mapping System for Bridge Maintenance

Gavin Paul, Stephen Webb, Dikai Liu, Gamini Dissanayake

*ARC Centre of Excellence for Autonomous Systems, University of Technology, Sydney,
NSW, Australia.*

Abstract

This paper presents a system for Autonomous eXploration to Build A Map (AXBAM) of an unknown, 3D complex steel bridge structure using a 6 degree-of-freedom anthropomorphic robot manipulator instrumented with a laser range scanner. The proposed algorithm considers the trade-off between the predicted environment information gain available from a sensing viewpoint and the manipulator joint angle changes required to position a sensor at that viewpoint, and then obtains collision-free paths through safe, previously explored regions. Information gathered from multiple viewpoints is fused to achieve a detailed 3D map. Experimental results show that the AXBAM system explores and builds quality maps of complex unknown regions in a consistent and timely manner.

Keywords: Robot Manipulator, Exploration, Bridge Maintenance, Laser Range Scanning, 3D Mapping

1. Introduction

The removal of paint and rust via grit-blasting, followed by the reapplication of a protective coating is an essential part of bridge maintenance: the process significantly reduces the likelihood of structurally damaging rust - hence extending bridges' lifespans. However, due to increased concern for workers' safety and strict Occupational Health and Safety regulations, it is no longer desirable for humans to manually perform the dangerous - due to close proximity with lead-based paints, asbestos and chromium - and strenuous [9] grit-blasting task. Thus, an attractive alternative is to perform a significant portion of this bridge maintenance work with a robotic system.

There are many challenges associated with placing a robotic manipulator in an unknown environment to autonomously perform the grit-blasting task. Generally in industry, robotic manipulators are used in controlled, known areas and infrastructure is built around them or tasks are brought to them. However,

Email addresses: gpaul@eng.uts.edu.au (Gavin Paul), swebb@eng.uts.edu.au (Stephen Webb), dkliu@eng.uts.edu.au (Dikai Liu), gdissa@eng.uts.edu.au (Gamini Dissanayake)

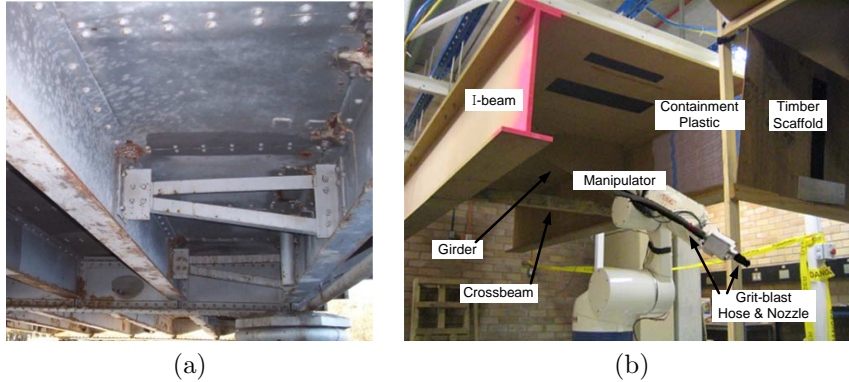


Figure 1: *a)* A structural bridge environment; *b)* I-beam channel scale model with robotic maintenance equipment.

in the case of bridge maintenance tasks, it is not feasible to set up the same level of controlled infrastructure. Therefore, the robotic grit-blasting maintenance equipment has been designed to be relatively simple, robust, and mobile enough to be placed on the available scaffolding and maneuvered inside a sealed work envelope. A manipulator-based system that must perform a grit-blasting task requires an up-to-date digitized environment map of the surrounding surfaces. These maps must contain details of the surrounding bridge infrastructure including the scaffold, and must allow for the integration of information about the current state of the surfaces [13]. Therefore, a system is required which is able to safely, efficiently and autonomously explore and build 3D maps of an initially unknown area, such as the environment in Figure 1, using an instrumented manipulator (eye-in-hand). The map generated must be a high-quality 3D representation suitable for grit-blasting planning. The system must be efficient so that the maps accurately represent the up-to-date state and layout of the environment immediately prior to blasting.

Significant progress has been made to solve various aspects of autonomous 3D mapping. However, it is not feasible to simply combine outcomes reported in the literature to generate the required quality 3D maps with an eye-in-hand manipulator. Geometry mapping and exploration in 2D, 2.5D or 3D using a mobile robot equipped with sensors such as laser range finders and stereo cameras is a heavily researched field [19, 23, 21, 6]. Often the focus of this work is a robot moving roughly on a 2D plane to build 2.5D or 3D maps of an environment [8, 20, 12] or the focus is on localization [24, 15] rather than on generating high-accuracy 3D maps of the geometry of surfaces. Exploring the 3D surfaces in an environment using a sensor mounted on a robot manipulator presents slightly different challenges, since both the sensor and the entire kinematic chain of the manipulator must be kept in space that is known to be empty. 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 infor-

mation. Incremental exploration must occur within the bridge structure so as to generate a 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.

Autonomous mapping techniques [1] using robot manipulators, with the ability to produce suitable quality maps for grit-blasting, typically have pre-programmed movements and are time consuming. In sensor-based motion planning, where the manipulator configuration space is explored [25, 26, 22], the purpose of exploration is to facilitate path planning by sensing the surrounding space that affects the ability to plan a safe path. This results in the exploration of commonly traversed configurations in the immediate vicinity of the robot manipulator, which means there can be no guarantee about map completeness [5]. Coverage and next best viewpoint-based algorithms for manipulators in obstacle-free known environments [1] need to be extended to address the problem of gathering information while also considering collision-free exploration.

This paper proposes an integrated system for manipulator-based exploration: Autonomous eXploration to Build a Map (AXBAM). The AXBAM system utilizes a 6DOF Denso robot manipulator instrumented with a Hokuyo Laser Range Finder (LRF) scanning sensor. The main contribution of this paper is an integrated system that is able to explore an unknown environment efficiently to generate a high-quality map. Strategies are presented to efficiently compute the information gain from a given viewpoint, determine the reachability of a given viewpoint in the robot workspace, and to avoid collisions with both the known obstacles as well as the unknown region of the workspace. This paper is organized as follows: Section 2 of this paper presents the fundamental concepts of the AXBAM system including 3D exploration predictions, and fast robot manipulator collision checking and path/motion planning. Section 3 presents implementation details of AXBAM. Section 4 presents verification results of exploring and mapping a typical bridge maintenance environment through a number of experiments.

2. Concepts

There are two significant challenges when determining the next exploration viewpoint: (a) predicting the quality of a candidate viewpoint, and (b) determining the ability of the manipulator to reach this viewpoint. The combination of these criteria allows permissible viewpoints to be compared and the most desirable to be selected.

2.1. Exploration Information Theory

During the exploration and mapping process, the information remaining about an environment decreases each time the environment is sensed. The information about an unknown environment can be quantified into ‘information measurements’ based on information theory, and then used to make exploration

decisions about quality viewpoints. A grid-based representation, conceptually similar to the Occupancy Grid (OG) defined by Elfes [4], was chosen since it facilitates a robust method to discretize an environment space and update the state-space belief. A standard way to measure information [7] is with the mathematical notion of entropy. Entropy describes the information which remains to be discovered, i.e. the amount of “uncertainty”. In accordance with Shannon’s entropy [18], the general measure of information, $H(^jY)$, can be taken of a single grid cell, j . In this case, jY is a discrete state random variable that takes the state values in a set $\{^jy_1, ^jy_2, \dots, ^jy_n\}$. Therefore, a state $^jY = ^jy_i$, over all the possible states $i \in \{1, \dots, n\}$, have probabilities of occurrence $P(^jY = ^jy_i)$, $i \in \{1, \dots, n\}$, such that

$$\sum_{i=1}^n P(^jY = ^jy_i) = \sum_{i=1}^n P(^jy_i) = 1 \quad (1)$$

Using information theory [18] the information remaining in the j th grid cell, $H(^jY)$, is defined as

$$H(^jY) = - \sum_{i=1}^n P(^jY = ^jy_i) \log(P(^jY = ^jy_i)) = - \sum_{i=1}^n P(^jy_i) \log(P(^jy_i)) \quad (2)$$

This enables the measurement of the entropy, $H(^jY)$, as the information remaining to be discovered about jY . Shannon specified in [18], that the information remaining is zero, $H(^jY) = 0$, if all the probabilities $P(^jy_i)$, of jY being in state jy_i , for $i = \{1, \dots, n\}$ are zero except one with index i^* , which has unity value as in

$$P(^jy_i) = \begin{cases} 0 & \text{for } i \in \{1, \dots, n\}, i \neq i^* \\ 1 & \text{for } i = i^* \end{cases} \quad (3)$$

Thus, only when the states are known does the information remaining, $H(^jY)$, vanish (i.e. $H(^jY) = 0$) since it is assumed $P(^jy_i) \log(P(^jy_i)) = 0$ for $P(^jy_i) = \{1, 0\}$. To have some information remaining, at least one state must be uncertain, since $(-P(^jy_i) \log(P(^jy_i))) > 0$ for $0 < P(^jy_i) < 1$. Given a number of possible states, $\{^jy_1, ^jy_2, \dots, ^jy_n\}$, then the information remaining in the j th grid cell $H(^jY)$ is a maximum when there is the most uncertainty about the state. Hence, the information remaining is a maximum when the probability of all states is equal, such as

$$P(^jY = ^jy_1) = P(^jY = ^jy_2) = \dots P(^jY = ^jy_n) = \frac{1}{n} \quad (4)$$

Then the upper bound of the information is measured as

$$H(^jY)^{up} = - \sum_{i=1}^n P(^jy_i) \log(P(^jy_i)) = n \left(-\frac{1}{n} \log\left(\frac{1}{n}\right) \right) = \log(n) \quad (5)$$

The remaining information, $H(^jY)$, for the j th grid cell (i.e. a source of information) with state variable jY , is within the range $[0, \log(n)]$ when there are n possible states.

In the case where there are a number of grid cells, m , then the information can be calculated for each grid cell and summed. The state variable vector \mathbf{Y} represents jY , for $j \in \{1, \dots, m\}$. If the number of states and the number of grid cells are assumed to be known *a priori* then it is possible to sum the information remaining in the j th cell over the m grid cells to calculate the total information which is denoted as $\mathbf{H}(\mathbf{Y})$ as follows

$$\mathbf{H}(\mathbf{Y}) = \sum_{j=1}^m H(^jY) = - \sum_{j=1}^m \sum_{i=1}^n P(^jy_i) \log(P(^jy_i)) \quad (6)$$

Similar to Equation 5, $\mathbf{H}(\mathbf{Y})$ is maximized when all grid cell states are unknown and hence the information is bounded by $[0, m \log n]$.

Let jX represent a discrete occupancy space variable. The space is discretized to contain n_v equally sized volumetric pixels (voxels). In the special case where a sensor that can detect all occupied regions within its field of view is available, the voxel's probability of occupancy will take one of the three values corresponding to the states: free, unknown and occupied (i.e. containing a surface). Where jX is a discrete state variable, the probability that the j th voxel is occupied is given by $P(^jX = ^jx_o)$ for the voxels, $j \in \{1, \dots, n_v\}$, and states $o \in \{\text{free, unknown, occupied}\}$. Therefore, the probability of occupancy is $P(^jx_o) = \{0, 0.5, 1\}$, if the j th voxel's occupancy state, jx_o , is $\{\text{freespace, unknown, occupied}\}$, respectively. Initially, all voxels except those in the volume occupied by the robot manipulator, are unknown, ($P(^jx_o) = 0.5$).

As the environment is discretized into n_v voxels with a probability of occupancy $P(^jx_o)$, the geometric entropy, $\mathbf{H}_g(\mathbf{X})$, of the environment's occupancy state, \mathbf{X} , can be used as a measure of cumulative information remaining in all voxels. Since voxels where $P(^jx_o) = 1$ (i.e. occupied state) and $P(^jx_o) = 0$ (i.e. freespace state) do not contribute to entropy ($\log(1) = 0$) the information remaining in the environment comes only from unknown state voxels. A sensing ray passing through a voxel switches the state to freespace. Detecting a surface point in a voxel switches the state to occupied. In the absence of those events the state is unknown and the probability is unchanged from the initial value: unknown. Therefore, the information remaining in the environment, $\mathbf{H}_g(\mathbf{X})$ is proportional to the number of voxels whose state is unknown, n_u .

During exploration, information gathered from a sequence of viewpoints can be used to update a voxel's probability of occupancy. Figure 2 illustrates the concept of exploration in 2D using a sensor that can perfectly detect all occupied regions within its field-of-view. White regions in this figure indicate space in an unknown state. The first scan, taken from a given and known-safe starting position, detects the freespace (light shading) and obstacles (dark shading). It is clear that subsequent steps result in an increase in the information gathered about the state of the surrounding area. The successive viewpoints, indicated by the numbered circles, must be within the space known to be empty and need

to be selected in order to maximize the information in the newest scan.

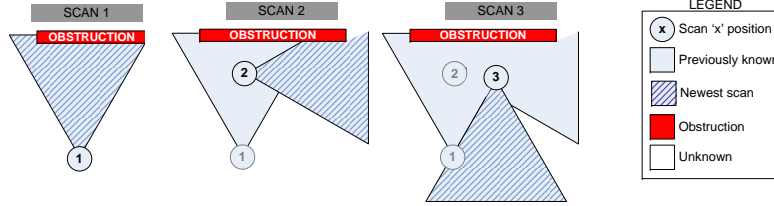


Figure 2: Three steps of a 2D version of exploration.

In the eye-in-hand 3D exploration scenario, where a sensor is mounted at the end-effector of a manipulator, both the complete kinematic chain of the manipulator as well as the sensor needs to be within known freespace. The entire manipulator must also remain within space that is guaranteed to be empty during the whole period where the manipulator moves from one viewpoint to another. Thus, the eye-in-hand 3D exploration scenario is significantly more challenging than the 2D case in Figure 2. Clearly, a suitable strategy is required to determine the locations of the viewpoints to be visited such that the environment is explored and a map of sufficient quality is generated. In the selection of the viewpoints, it is necessary to balance the information gain achieved with the effort needed to reposition the sensor.

In the typical robot bridge maintenance scenario, where a manipulator is initially placed in an I-beam channel, the most important region is where the manipulator can reach from a stationary base position. The regions which can be affected by the blast stream tool must also be mapped so they can be considered during blast planning. For grit-blasting, as opposed to operations that require surface contact (i.e. polishing), the distance between the robot end-effector and the surface to be treated is not required to be fixed. Therefore, an approximation of the workspace that can be easily computed is preferable. As shown in Figure 3, for the manipulator used in this work these regions can be approximated by ellipsoids, centered at the position of the first joint of the robot, which cover the maximum reach plus the grit-blast range. The region, A_1 can be traversed by the manipulator, therefore information about the state of the voxels in A_1 is the most valuable. The second region, A_2 , is the region which is to be blasted, and hence also requires a detailed map, but is not safety-critical in terms of collision avoidance. A_3 is outside the maximum blasting reach of the manipulator but still may be damaged by a grit-blasting stream. Therefore it is desirable, but not essential, to build a map of this region. The voxels in the three regions are given weights (C_{w1}, C_{w2}, C_{w3}) which indicate their relative importance during the exploration process.

Initially, the occupancy state of all voxels is unknown, except those voxels in the volume that safely contains the robot. Then $\mathbf{H}_w(\mathbf{X}_0)$ is the total weighted information remaining prior to exploration (i.e. at the 0th viewpoint where the environment's occupancy state is \mathbf{X}_0), which is proportional to the summation

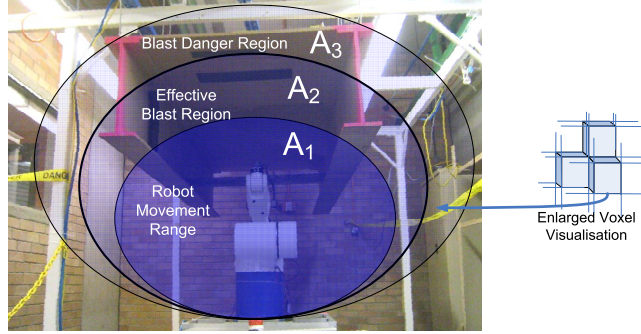


Figure 3: Regions of importance $\{A_1, A_2, A_3\}$ are weighted by $\{C_{w1}, C_{w2}, C_{w3}\}$

of the weighted count of unknown voxels in each region,

$$\mathbf{H}_w(\mathbf{X}_0) \propto n_{u,1}C_{w1} + n_{u,2}C_{w2} + n_{u,3}C_{w3} \quad (7)$$

where $n_{u,1}$ is the number of unknown voxels in A_1 , which is all voxels except those in the volume occupied by the robot. $n_{u,2}$ is the number of unknown voxels in A_2 and not in A_1 , and $n_{u,3}$ is the number in A_3 and not in A_1 or A_2 .

2.2. Prediction of Information Gain

When the LRF sensor is positioned by the 6DOF manipulator at a new viewpoint, ${}^0T_r(\vec{Q})$ for the manipulator pose, $\vec{Q} = [q_1, q_2, \dots, q_6]^T$ the unknown voxel's volume of space which can be sensed is denoted as $V_{new}(\vec{Q})$. ${}^0T_r(\vec{Q})$ is the 4×4 rotation and translation homogeneous matrix that describes the position and orientation of the scanner pose in the manipulator base coordinate frame. In order to *minimize* the remaining information, the number of unknown voxels whose states are determined at each viewpoint must be *maximized*. Therefore, it is necessary to be able to predict the number of unknown voxels which can be sensed from a viewpoint. These predictions must account for occlusions faced by the sensor. The LRF sensor arrangement used in the work is capable of detecting surfaces that are visible and within the approximately spherical field-of-view and within the range of four meters. This leads to an efficient strategy for calculating the information gain that can be achieved by moving to a new viewpoint as shown in Figure 4. Using the current triangle mesh map it is possible to create a reduced data set consisting of n_o partially overlapping Scale-Like Discs (SLDs) [14]. The SLD generation algorithm [14] transforms the fused triangle mesh maps via principal component analysis, into a format, consisting of a center point and a normal, which can be used to make efficient predictions. The prediction of $V_{new}(\vec{Q})$ can be determined by using the SLDs and the established technique of ray casting [10]. Figure 4 shows that for each of the laser scanner's rays, a ray is cast from the proposed sensor viewpoint, ${}^0T_r(\vec{Q})$, through the partially known map to the extent of the sensing range. If

the ray doesn't intersect with a SLD (i.e. a surface), all unknown voxels which the ray passes through (up to the maximum sensing range) are included in the predicted volume, $V_{new}(\vec{Q})$. If the ray intersects with one or more SLDs then, only the unknown voxels between the sensor and the closest SLD are included in $V_{new}(\vec{Q})$. Thus, the reduction in the number of unknown voxels can be predicted for each viewpoint, \vec{Q} .

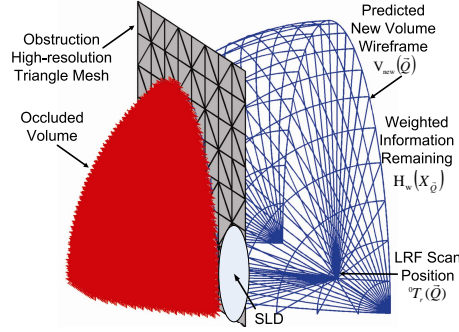


Figure 4: Occluded volume is removed from predictions of $V_{new}(\vec{Q})$ by ray casting into the partial map from the sensor. If a ray passes through a voxel it is determined to be free. Voxels containing a SLD are occupied and voxels which have not had a ray cast through them at any stage have an unknown occupancy state.

As it is likely that there will be some regions that are not visible from any vantage point, the limiting value of unknown volume remaining, will be some positive lower bound. Obviously it is intractable to move the sensor to all possible configurations, so the key challenge is to minimize the unknown volume while also minimizing the number of viewpoints.

2.3. Towards Map Completion

At each viewpoint, the output of each ray of the LRF scanner is a 3D point, $[x_r, y_r, z_r]^T$, corresponding to a range measurement, d_r . The position of the homogeneous point, P_r is calculated based on the pose of the LRF scanner,

$$P_r = [x_r, y_r, z_r, 1]^T = {}^0T_r(\vec{Q}) [0, -d_r \sin \theta_r, d_r \cos \theta_r, 1]^T \quad (8)$$

where ${}^0T_r(\vec{Q})$ is the scanner pose in the manipulator base coordinate frame for the pose, \vec{Q} . θ_r is the angle of the ray with respect to the axis of the scanner and d_r is the distance along ray, r to the nearest object. The sensor's angular resolution is θ_{res} and the manipulator tilt angular resolution is α_{res} .

For grit-blasting it is important to have a complete map of the surfaces of objects in the environment, particularly near corners that are more likely to be obstructed during sensing. As shown in Figure 5a when the entropy is computed it is possible to improve the probability of map-completeness by increasing the "importance" of voxels that are occluded. The regions that are occluded can be

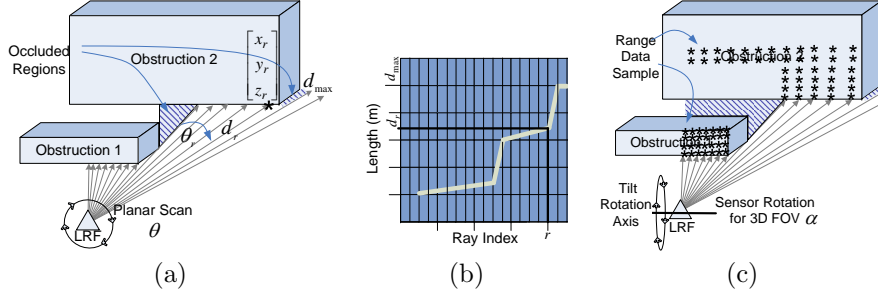


Figure 5: a) Determine occluded regions that have an unknown occupancy state; b) Graph of the partial scan data; c) Tilting the LRF to produce 3D range data.

detected by examining the rate of change of the ray lengths that are plotted in Figure 5b. The range measurements obtained by tilting the LRF through a 3D field-of-view as in Figure 5c can be presented as a matrix of ray lengths,

$$D_{\theta, \alpha} = \begin{pmatrix} d_{i,j} & d_{i+1,j} & \dots \\ d_{i,j+1} & d_{i+1,j+1} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad (9)$$

where the index i refers to the pan index (on 2D scan plane up to i_{max}), and j the tilt increment, this translates back to the r th ray by $r = i_{max}(j-1) + j$. If the range alters significantly over a change in either angular variable as,

$$\Delta d_{i,i+1} = d_{i,j} - d_{i+1,j} > C_L \text{ or } \Delta d_{j,j+1} = d_{i,j} - d_{i,j+1} > C_L \quad (10)$$

where C_L is a constant comparator, then there is a high probability that there is a region close to surfaces that is obscured. If this region could be viewed from an ideal end-effector pose it would reveal the state of the currently unknown space near a surface. In the case where this volume is unknown, the optimal position to view the region to reduce the uncertainty is perpendicular to the rays where the large change in range was found. The voxels in this region are thus given an additional importance weighting in $\mathbf{H}_w(\mathbf{X}_{\vec{Q}})$. Thus, viewpoints are selected where there is a higher probability of generating a complete map near surfaces in the important regions.

2.4. Collision-free Manipulator Planning

Each candidate exploration viewpoint out of all the possible viewpoints $\vec{Q} \in \mathbf{Q}$ must be examined to ensure there is a safe and efficient trajectory from the current manipulator pose, $\vec{Q}_{current}$. Since dynamic collision checking for manipulators is significantly challenging to do reliably and efficiently, path/motion planning for industrial manipulators has traditionally been performed offline, with known maps and goal states [16]. Many manipulator path/motion planning methods focus upon traversing configuration space by sampling paths at

some pre-specified resolution. Finer movement resolution checks increases accuracy but at a higher computational cost [17]. Generally path planning has a single goal state. However, in the case of the AXBAM system, it is necessary to have numerous possible exploration goal poses and this in turn leads to greater planning complexity. Therefore, it is necessary to streamline the collision-free path/motion planner so it still avoids singularities and joint limitations but has a greatly reduced search envelope, and exploits the fact that there are multiple possible goal states, \mathbf{Q} .

The AXBAM system gathers large quantities of sparse obstacle data during exploration. Since all manipulator configurations along a continuous path must be collision-free, we have modified an existing ellipsoid force-field planner [2] and collision checker to include a factor of safety. In exploration there are many candidate poses, and hence the constraint on the path planner: to reach a single goal pose, can be relaxed so as to improve the speed of the path planner. This results in a slight (empirically determined to be about 5%) reduction in the path planner's ability to find complicated paths, and thereby incorrectly rejecting some exploration viewpoints as unreachable. However, this is inconsequential since there are many possible goal viewpoints.

It is essential for the collision-free path planner to have appropriate force-field sizes that include a factor of safety so as to simplify the discretization of the manipulator movements during planning [16]. Hence, a force-field growing method incorporating a factor of safety has been developed. The safety factor enables the system to more coarsely discretize the movements though configuration space - subsequently minimizing the computational time of the collision checker. Ellipsoid force-fields are calculated with a factor of safety around each of the manipulator links, $i = \{1 \dots 7\}$ using the mesh-vertices $P_{v,i}$ from the supplied CAD model of the manipulator. Note that the number of manipulator links is one more than the manipulator joints. The factor of safety is a heuristically-determined distance buffer, d_s between each links' polygon model and the respective ellipsoidal force-field. Using this measure, it is possible to calculate the maximum allowable rotation angle for the first 3 joints δq_1 , δq_2 and δq_3 , so that no $P_{v,i}$ for any of the joints moves more than d_s . By designating d_l as the length of the remaining portion of the manipulator from the center of the joint being rotated to the tool on the manipulator polygon model, the maximum δq_j can be calculated so all points in $P_{v,i}$ move less than d_s ,

$$\delta q_j < \cos^{-1} \left(1 - \frac{d_s^2}{2d_l^2} \right), j \in \{1, 2, 3\} \quad (11)$$

Therefore, the planner has a maximum threshold joint configuration planning steps of less than δq_1 , δq_2 and δq_3 , so as to guarantee that no part of the manipulator moves more than d_s . This discretization simplifies planning searches and hence reduces the computational time.

Given the center of the i th ellipsoid is at $P_{c,i}$, ellipsoid force-fields are 'grown' by alternate methods of increasing the ellipsoid equatorial radii parameters a_i , b_i and the polar radius c_i so as to increase the ellipsoid size and best enclose

the manipulator. The polygon model, $P_{v,i}$ of the i th link is expanded by the safety distance, d_s to generate the set of points, $P_{s,i}$. This is done with the following algorithm that inherently increments *iteration*, and is based upon the ellipsoid distance formula. The algorithm is then performed once for each of the $i = \{1 \dots 7\}$ links:

```

1: while  $\{(P_{c,i} - P_{s,i})^T \cdot \text{diag}(a_i^{-2}, b_i^{-2}, c_i^{-2}) \cdot (P_{c,i} - P_{s,i}) > 1\}$  do
2:   if  $(\text{mod}(\text{iteration}, 2) = 0)$  then
3:      $[a_i, b_i, c_i] = [a_i, b_i, c_i] \times 1.01$ 
4:   else
5:      $[a_i, b_i, c_i] = [a_i, b_i, c_i] + \frac{3(a_i + b_i + c_i)}{100}$ 
6:   end if
7: end while

```

where *diag* is a diagonal matrix. We use the force-field collision checker with an integrated safety distance to specify the angular resolution at which checks must be performed. That is, given a path, \vec{Q}_{all_steps} then each of these discretized angular increments \vec{Q} along the path is checked for joint limits and collisions. This collision checker determines if there are voxels in an unknown state, \mathbf{P}_u or that contain obstacles, \mathbf{P}_o , inside any of the virtual ellipsoid fields covering the manipulator links. The centers and parameters of the ellipsoidal fields around the manipulator links are centered at $P_{c,i}$ with parameters $[a_i, b_i, c_i]$. The position of link i is described by the homogenous transform matrix ${}^0T_i(\vec{Q})$ in the manipulator base coordinate frame. For efficiency, the union of obstacle voxels \mathbf{P}_o and unknown-state voxels \mathbf{P}_u are transformed from the robot base coordinate frame into the i th ellipsoid's coordinate frame using the inverse homogeneous transform matrix, represented here as ${}^iT_0(\vec{Q})$. This results in a transformed set of homogeneous points for this manipulator link to avoid $\mathbf{P}_{T,i}$.

$$\mathbf{P}_{T,i} = {}^iT_0(\vec{Q}) \begin{bmatrix} \mathbf{P}_o, \mathbf{P}_u \\ 1, 1 \end{bmatrix} \quad (12)$$

The collision checking function is then performed for each link $i = \{1 \dots 7\}$,

$$result = f(\mathbf{P}_{T,i}, P_{c,i}, [a_i, b_i, c_i])_{check} \quad (13)$$

It is then possible to determine if there is an overlap between the transformed points $\mathbf{P}_{T,i}$ and the i th link's ellipsoid. If the following inequality is true for any points in $\mathbf{P}_{T,i}$, then the boolean *result* is returned false and the possible collision is reported and the path is modified or discarded.

$$(P_{T,i} - P_{c,i})^T \cdot \text{diag}(a_i^{-2}, b_i^{-2}, c_i^{-2}) \cdot (P_{T,i} - P_{c,i}) < 1 \quad (14)$$

Using the ellipsoidal fields created allows for discretized pose checking so that unsafe viewpoints are discarded. The motion plans and the path lengths can be calculated using [2]. The candidate viewpoints $\vec{Q} \in \mathbf{Q}$ are thus safe and are assigned two measures of quality: the information remaining reduction in regions of weighted importance, and the difficulty to move the manipulator to this viewpoint. Henceforth, the implementation of the AXBAM system is detailed,

including the exploration algorithm that selects the most desirable viewpoint using the criteria: information remaining, map building and manipulator movement.

3. AXBAM System Implementation

Based upon the concepts presented in Section 2, this section presents the implementation of the AXBAM system as shown in Figure 6.

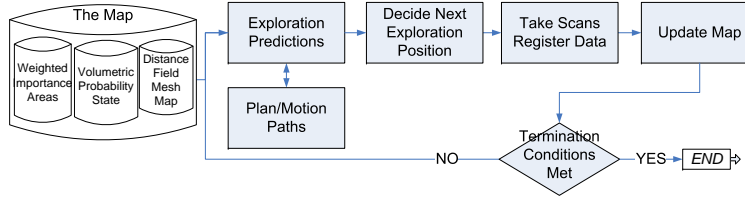


Figure 6: Flowchart of AXBAM system

Initially, the instrumented manipulator is placed into an unknown environment in a crouched position: a predetermined compact manipulator pose where only one possible sensing scan can be taken. At each subsequent iteration of the exploration process, the currently available map is analyzed so as to determine the next viewpoint. Exploration predictions are performed based upon the state of the map and the weighted importance of the remaining information. Collision checks are then performed as part of the processes to generate manipulator path and motion plans. At each viewpoint, once the new data has been acquired the map is updated and then reutilized. This process continues until exploration termination conditions are met. The system components have been necessarily streamlined for time/ processing efficiency. The following section details the system implementation of the exploration algorithm, the map building and finally the exploration termination conditions.

3.1. Exploration Algorithm

The exploration algorithm is used to determine the next viewpoint and is based upon the information remaining as presented in Section 2, and a planned manipulator path. As previously shown, configurations are considered more desirable when they are predicted to reduce the information remaining measure, $\mathbf{H}_w(\mathbf{X})$ due to an unobstructed view of an important region where the state of the voxels is unknown. Each configuration of the manipulator places the LRF sensor at a viewpoint which has an associated predicted information measure. It is intractable to perform a next viewpoint search through all possible configurations between the joint limits $q_{j,min} \rightarrow q_{j,max}$ for joints $j = \{1, 2 \dots 6\}$; however, it was determined that a joint resolution any finer than the angle δq_j results in rapid increases in the prediction calculation time without improving

the candidate viewpoints. The discretization heuristic that specifies the number of steps, $q_{j,steps}$ for the j th joint was determined experimentally for the first three joints. Additionally, since the LRF field-of-view is relatively large, it is possible to significantly limit the sets of discrete steps for joints 4 and 5 to $\delta q_{4,set}$ to $\delta q_{5,set}$, respectively. δq_6 , is used to ensure the LRF scan plane normals are all coplanar so as to maximize the information remaining reduction at each viewpoint. This reduced set is still representative of the weighted predicted information remaining measure, which at \vec{Q} is given by $\mathbf{H}_w(\mathbf{X}_{\vec{Q}})$. The algorithm is:

```

1:  $\delta q_j = \frac{q_{j,max} - q_{j,min}}{q_{j,steps}}, j \in \{1, 2, 3\}$ 
2: for  $j \in \{1, 2, 3\}$  step through  $q_{j,min} \rightarrow q_{j,max}$  by  $\delta q_j$  do
3:   if  $\vec{Q}$  is new &  $\vec{Q} \notin \{failed\_list\}$  then
4:     for  $q_4 \in \{q_{4,set}\}$  do
5:       for  $q_5 \in \{q_{5,set}\}$  do
6:         Using  ${}^0T_r(\vec{Q})$ 
7:         if  ${}^0T_r(\vec{Q})$  is new &  $\mathbf{H}_w(\mathbf{X}_{\vec{Q}}) - \mathbf{H}_w(\mathbf{X}_{\vec{Q}_{current}}) < C_u$  then
8:           Add  $\vec{Q}$  to prediction list
9:           Determine  $V_{new}(\vec{Q})$  and thus  $\mathbf{H}_w(\mathbf{X}_{\vec{Q}})$  from  ${}^0T_r(\vec{Q})$ 
10:          Get  $\vec{Q}_{current} \rightarrow \vec{Q}$  manipulator path length
11:         end if
12:       end for
13:     end for
14:   end if
15: end for
16: Order  $\vec{Q} \in \mathbf{Q}$ : minimize  $\mathbf{H}_w(\mathbf{X}_{\vec{Q}})$  & minimize  $\vec{Q}_{current} \rightarrow \vec{Q}$  path length

```

To improve the implementation speed, an ideal-case information reduction is precalculated offline and stored for the set of candidate poses $\vec{Q} \in \mathbf{Q}$. The predicted information reduction is then compared to a constant C_u to determine if the viewpoint from the sensor transform ${}^0T_r(\vec{Q})$ can possibly significantly reduce the information remaining in the environment. *failed_list* is a list of \vec{Q} known to be unsafe due to obstacles. Once this algorithm is complete the AXBAM system orders the candidate viewpoints using weighted predicted reduction in the information remaining $\mathbf{H}_w(\mathbf{X}_{\vec{Q}})$, and the joint movements required to move the manipulator to \vec{Q} . The next exploration viewpoint is then selected as the best out of the available list.

3.2. Map Building

The AXBAM system aims to develop a detailed and comprehensive 3D geometric surface map. To support fusion of multiple scans into surface location estimates, an adaptive distance field map representation (volumetric technique) based on work by Curless [3], is implemented as a real-time, online solution. The technique has been extended to handle thin plates and sharp features; however details of these extensions are beyond the scope of this paper. Fig. 7 shows a

block diagram of the processing pipeline that transforms range data and manipulator configuration data into a 2D grid of 3D points in the manipulator base coordinate frame. The accurately calibrated manipulator provides time-stamped configuration data at 10Hz. Range data is continuously acquired and time-stamped while the manipulator moves.

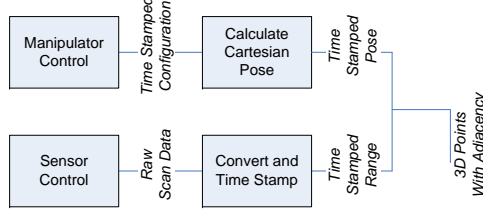


Figure 7: A grid of 3D points in the robot base coordinate frame is produced from the manipulator pose and the LRF range data.

Figure 8 shows a block diagram of the processing pipeline that transforms a 2D grid of 3D points into distance field updates. In the first process we extract triangular mesh then four adjacent grid points form two triangular surface facets if the triangle edges are not longer than a threshold value (100mm). This effectively removes any isolated data points caused by some sensor aberration.

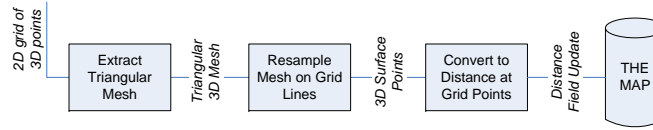


Figure 8: The distance field is updated using 3D points at the intersection of axis-aligned grid lines with the input surface.

To update the map, the meshes are re-sampled on grid lines using trilinear interpolation to calculate the point where a regularly spaced grid intersects with the surface. The interpolated surface point is used to update a weighted signed scalar distance field which notionally records the distance to a surface from each point in 3D space. The weighted signed distance at each grid point, is then updated using the weighted average approach [3]. The output of this iterative process is the map which can be rapidly queried by other components of the AXBAM system, and so as to facilitate the grit-blasting planning processes that are not included in the AXBAM system. The external blast planning system requires a set of targets with normals that completely represent the surfaces. Targets are created using a grit-blasting target selection process [14] which uses the mesh map generated by the AXBAM system. Surface material-type identification data generated by [13] can also be fused into the map. The map is then used by a force-field planning algorithm [2] to determine the optimal movements of the blast nozzle over the targets given the blasting constraints.

3.3. Algorithm Termination

The AXBAM system terminate when the information remaining is not likely to be reduced any further. In the case of termination, it is determined that the voxels that still remain in an unknown state cannot be seen from any achievable viewpoint. Several tests from [11] are used to determine when to terminate exploration based upon the relative information remaining gradient, and the ratio between the predicted information reduction and the actual information reduction. Analysis of the regions that are unknown forms the objectives: to reach a satisfactory level of understanding about the environment and about the map surface continuity. The information remaining gradients, $\mathbf{H}_w(\mathbf{X}_i) - \mathbf{H}_w(\mathbf{X}_{i-1})$ at exploration iteration i are summed for the previous n_s iterations relative to the information known at that step $\mathbf{H}_w(\mathbf{X}_i)$ and compared to an acceptable minimum relative gradient threshold C_t :

$$\sum_{i=n_s}^0 \frac{\mathbf{H}_w(\mathbf{X}_i) - \mathbf{H}_w(\mathbf{X}_{i-1})}{\mathbf{H}_w(\mathbf{X}_i)} < C_t \quad (15)$$

It is then possible to test the hypothesis that it is acceptable to terminate exploration. The AXBAM system will terminate exploration if: the environment information remaining reaches zero; or over the last n_s exploration iterations the total relative change in information remaining is less than the threshold; or the total number of iterations exceeds the maximum allowable.

4. Results

In order to test the AXBAM system, several experiments have been performed in environments modeled on the actual bridge maintenance environment. This section presents one detailed experiment of the AXBAM system tested in a 1:1 scale testing environment. Subsequently, two additional experiments are conducted in environments with increased complexity, and the results are presented. In order to measure the results of exploration and mapping in the three experiments, the modules of the AXBAM system are verified by several Key Performance Indicators (KPIs). The KPIs indicate the success of the system as it operates within the environment setups that are indicative of the onsite bridge maintenance environment. The enumerated KPIs have the following purposes:

1. Examine the efficacy of viewpoint selection including both information remaining predictions and path/motion planning. Since exploration must be efficient, each viewpoint selected must be effective. Scan speed and manipulator movement speed is almost constant and generally moving to viewpoint and scanning takes between 1-3 minutes. It is desirable that decisions, planning and map updates be made in a fraction (less than 20%) of this time (i.e. 12-36 seconds).
2. Evaluate the ability of the AXBAM system to look around objects in order to complete surfaces by analyzing the viewpoints selected and by visually inspecting the map generated for discontinuities.

3. Examine the termination point of the algorithm and the conditions preceding this by comparing the percentage of information gained to the maximum information available. The presumed 100% value is based upon 30 exploration iterations and this KPI is considered successful when 95% is exceeded.
4. Demonstrate collision-free paths between exploration viewpoints.
5. Determine the quality of the map generated by comparing selected patches to a ground truth map. The normal distance between the map mesh vertices and the ground truth plane is collected and the error mean, μ and standard deviation, σ of each patch is calculated. These measures can identify high resolution errors in the map. Grit-blast targets are constructed by segmenting and smoothing the map [14]. Therefore, provided the majority of the vertices lie near of the ground truth plane (i.e. $|\mu \pm \sigma| < 25mm$), and the distribution of these points is approximately Gaussian then this KPI is satisfied.
6. Compare the determined availability of free configuration space with a ground-truth value. If the map provided to the grit-blast planning processes is overly conservative or erroneous, there will be a limited range of movement available to the manipulator or there may be collisions. At the mesh and the voxel level, a ground truth map and AXBAM-built map are compared for similarity. Then high-resolution discrete joint configuration poses are simulated in both maps so as to compare the joint configurations that are actually possible to those that are considered possible in the AXBAM-built map.

4.1. Experiment Description

The first experiment is performed using the 1:1 scale model environment shown in Figure 9a. The environment contains various obstacles including: two painted steel scaffold members; 300mm square unpainted steel plates; standard rubber blasting hose; containment plastic sheets; decayed, painted and solid timber; and aluminium sheeting. Inside the 3.6 meter I-beam channel a girder is placed between the two parallel I-beams, and a 1.22 meter timber crossbeam is placed on the bottom flange of both I-beams so as to span the I-beam channel. Similarly to the bridge maintenance application, the environment contains scaffolding which supports the I-beam channel model. Scaffolding members are additional obstacles which the AXBAM system must avoid in order to build the map of the environment.

4.2. Experiment Procedure

A standard Denso VM-6083 industrial robot manipulator arm is mounted on a platform which is placed into the I-beam channel amidst the obstacles. The platform is static during exploration. A Hokuyo URG-04LX Laser Range Finder (LRF) scanner tool is attached to a mounting aluminium plate at the end-effector of the manipulator. A 3D sensing field-of-view with the LRF is then possible by rotating the 5th joint of the manipulator. When the AXBAM system



Figure 9: *a)* Case study environment - scale model of the target bridge environment; *b)* Resulting point cloud map.

commences exploration, the manipulator is in a predefined crouching pose called the ‘zero’ pose. At the beginning, nothing is known about the environment except that this ‘zero’ pose is safe, and that it is possible to safely move the last two joints. The system then executes a 3D scan of the environment, thereby building a partial map upon which the future exploration decisions can be based. Surface meshes gained through subsequent viewpoint scans are fused into the map, and freespace voxels traversed by the scan rays are registered as known freespace. The AXBAM system continues the exploration process until the termination conditions are met.

4.3. Experiment Outcomes

In the first experiment, after the initial ‘zero’ scan, AXBAM went through twelve exploration scan iterations taking 13.6 minutes to complete all scanning, manipulator movements and computations. Only 10% (81 sec) was required for exploration predictions, path/motion planning, and map building/updating (on an embedded-PC: equivalent P4-2.0GHz, with code in C++ and Matlab). The average time (6.75sec) per exploration decision including the multi-goal path/motion planning time is acceptable for KPI 1, as is the speed of convergence upon information available. Figure 10*a* shows the selected exploration viewpoints for the LRF sensor and the corresponding manipulator poses. The information gain at each viewpoint is shown in Figure 10*b* and converges upon the information available in the environment.

KPI 2 is realized by visual inspection of the selected exploration viewpoints (Figure 10*a*) and the vertexes from the triangular mesh map that is built as shown in Figure 9*b*. Poses 1 and 2 are similar viewpoints and look upwards and around the robot manipulator. In pose 3 the manipulator is directed to a viewpoint on the left-hand side of the I-beam. In poses 4 and 6 AXBAM is opening up configuration space so as to move to poses 5 and 7, respectively. Poses 8 and pose 9 look in front of the manipulator. Pose 10 looks around the base, which due to the lack of remaining information in other regions, consequently contains the most important information remaining. Pose 11 and pose 12 (not shown) target a portion of the remaining information above the I-beam channel.

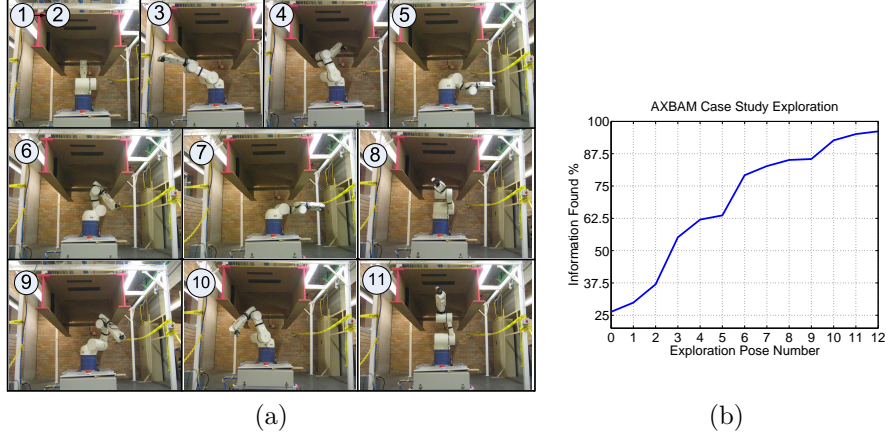


Figure 10: *a)* First 11 exploration manipulator poses; *b)* The information obtained by the AXBAM system in 12 viewpoints.

However, these poses do not result in significant information gain due to the occluding surfaces on the roof.

A visual inspection of Figure 9b shows the map to be sufficiently complete. However, there are regions of the map with discontinuities such as the areas behind the manipulator on both the girder and the left-hand flange. These are due to the crossbeam and q_1 's limited range of motion behind the manipulator - thus limiting the ability to reach a satisfactory viewpoint. Fortunately, these map discontinuities are behind the manipulator in regions where it is impossible to perform grit-blasting and so they are acceptable.

As shown in Figure 10b, the information gained by the AXBAM system rapidly approaches the maximum available. After 12 exploration viewpoints the map reaches a state whereby 96% of the available information is known - so KPI 3 is met since the algorithm terminates appropriately. Additionally, there are no collisions in either the poses or the movements between the poses, which therefore fulfils the criteria for KPI 4.

Since the ultimate aim of this system is to build a map that can be used in blasting, it is important to compare the exploration results of the mapping and the data fusion process to the actual environment that is being mapped (i.e. KPI 5). As shown in Figure 11a ten different significant planar surface patches from CAD drawings have been selected from the original ground truth environment and are compared to the map gained by the AXBAM system. The patches are 1=left web front; 2=right web front; 3=left web back; 4=right web back; 5=right roof front; 6=right roof back; 7=left roof front; 8=left roof back; 9=right bottom flange; 10=left bottom flange. The boxplots of the error for each patch, including the mean (μ) and standard deviation (σ) (in mm), is shown in Figure 11b. From this figure it is clear that the mean error for the patches is $\mu \approx 0$. Also, σ shows that there is not a large spread in the

patches since the majority of the distribution function lies with density around μ . The only notable exceptions are the patches on the roof (i.e. 5 and 7) towards the front where some shiny metal was present. The highly reflective surface made it difficult to obtain reliable data from the LRF scans. To meet

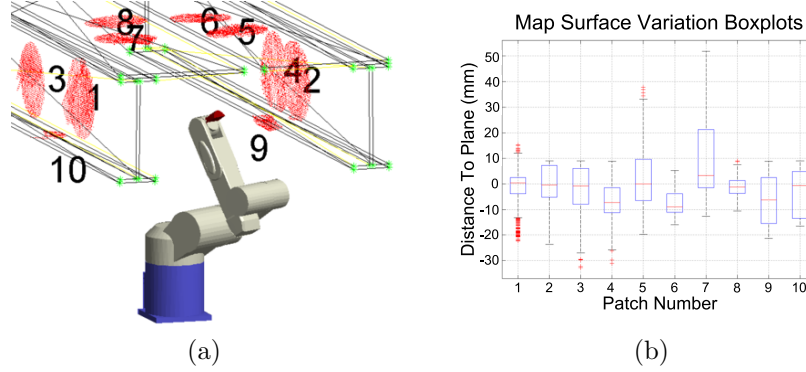


Figure 11: *a)* Significant surface patches; *b)* Map data variance boxplots for selected patches.

KPI 6, the AXBAM map should be similar to ground truth map and allow for the correct range of movement during future blasting. Overall, the map similarity to the ground truth map is high. The mean distance of actual surfaces from representative meshes (error) is approximately 3.8mm with a Gaussian distribution and standard deviation of 16mm. Given the set of fine-resolution joint configurations, both maps were tested and it was found that 96% of the configuration space available in the ground truth map were available in the map created by the AXBAM system. This meets the requirements of KPI 6. Hence, for this experiment all the KPIs were successfully met.

4.4. Additional Tests

Two additional tests were conducted on variations of the bridge maintenance environments to further test the capability of the AXBAM system. Figure 12*a* shows the environment for Additional Test 1 (AT1) and Figure 12*b* shows the environment for Additional Test 2 (AT2). The KPIs that were established in detail in the first experiment are used again here as a measure of performance for these two additional test cases.

4.4.1. Additional Test 1

In AT1 a crossbeam is placed in front of the manipulator and an additional crossbeam and timber lattice is installed behind. These additional obstacles restrict the possible movement of the manipulator in front and above. For safety, the manipulator movement speed was reduced in this experiment. The AT1 KPI results are shown in Table 1.

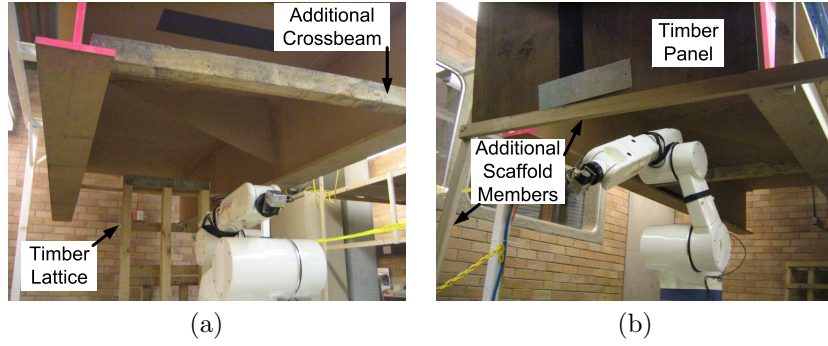


Figure 12: Two Additional Environments: *a)* AT1; *b)* AT2.

Table 1: Additional Test 1 - Results

KPI1	AXBAM used 18 poses (i.e. viewpoints) which is more than in the first experiment. The decision making time was longer (34 seconds per decision) due to the path planning required to handle the complexity of the environment. However, the time was only 2% of the total movement time and since the paths were more complicated and the manipulator speed was reduced for safety.									
KPI2	Initial poses selected were similar to the initial experiment. However, due to the additional obstacle in front, more exploration was done around the obstacle. The map looked more complete than the map in the initial experiment since more data was available behind the robot particularly near the shiny metal plates on the roof.									
KPI3	Termination occurred at 99% of maximum possible information.									
KPI4	No collisions in any movements.									
KPI5	1	2	3	4	5	6	7	8	9	10
μ	-3	0	-2	-4	-29	-14	-9	5	-4	-6
σ	14	6	8	7	6	11	12	4	9	11
KPI6	Overall map error, μ was 2.7mm with a Gaussian distribution about μ and σ of 25mm. There was a 98% similarity between the true and the discovered possible manipulator movements.									

All KPIs were met except for KPI5 with the 5th patch: the shiny-steel roof area of the map. It was found that multiple viewpoints of the shiny surface actually led to the surface in the map converging upon an incorrect surface position, which has been found to be due to limitations with the sensing technology [13]. The exploration decision speed was improved significantly as a percentage of total exploration time due to the robot manipulator speed being lowered in this experiment. The KPI target of less than 36 seconds was met even though decision times are significantly longer than in the initial experiment. The decision times were due to the increased difficulty of motion planning in the AT1 envi-

ronment that contains additional obstacles impeding manipulator movements. Overall AT1 was considered successful.

4.4.2. Additional Test 2

In AT2 the environment is once again more complex than the first experiment. A crossbeam is placed both in front of and behind the manipulator. The girder is still in place. Additional scaffolding is positioned in front of the robot so that various members are in obstructive positions. Based on the previous results, the viewpoint selection times can be expected to increase due to the increased complexity. In AT2 the manipulator speed is set to be the same as experiment 1 and only the q_5 tilt speed for scans is slightly decreased with the aim to see smoother map meshes. The AT2 KPI results are shown in Table 2.

Table 2: Additional Test 2 - Results

KPI1	In AT2 there were 13 exploration viewpoints. Convergence on the information available was rapid and the average exploration decision time (20.3 sec) was better than in AT1 although the overall time was larger. This was due to slower scan speeds and having to path planning away from local minima several times.									
KPI2	There were several futile movements to positions that, had the environment been known, would not have been selected. There were many obstacles in the environment so that at every new viewpoint more obstacles within range of the manipulator were discovered. Visually, the map's continuity is similar to that of AT1.									
KPI3	Termination occurred at 93% of the possible information due to the excessive time in movements. The information available was converged upon in the first 6 viewpoints and the remaining viewpoints were not as productive. In the presumed 100% case used for comparison a gap between two surfaces was used, which was not found by the AXBAM algorithm, and this allowed for significant information gain.									
KPI4	No collisions in any movements.									
KPI5	1	2	3	4	5	6	7	8	9	10
μ	-1	0	-1	-6	-1	4	6	-1	-7	-7
σ	6	7	8	7	9	8	11	3	8	14
KPI6	The map error was particularly low $\mu = 0.02mm$ and $\sigma = 15mm$, 96% of actually reachable poses were found to be possible.									

Overall all KPIs were met except KPI 3 which only just failed to achieve the required 95% benchmark. This is because the algorithm did not utilize a particular difficult-to-reach viewpoint; however conversely, there were no collisions. As expected, the overall time for the AXBAM system to complete exploration increased due to the environment complexity, it is still acceptable for online mapping. The map that was built is of a high quality, and is less error prone than the maps in previous tests, due to the slower tilting scan speed.

4.5. Discussion

As is shown in these results the AXBAM system encompasses solutions which are highly integrated and have been developed so as to minimize decision making time for both exploration viewpoint selection and manipulator path/motion planning. The quality of the maps that are built is high. 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 AXBAM system is a valid solution for building a map of the complex bridge maintenance environments that can be used in grit-blast planning.

5. Conclusion

This paper has described the AXBAM system for Autonomous eXploration to Build A Map of a bridge maintenance environment using a 6 degree-of-freedom robot manipulator, instrumented with a laser ranger finder scanning sensor. This paper has demonstrated a methodology for predicting information gain, comparing differential joint configurations and providing collision-free path/motion planning. This in turn allows the system to maximize the effective exploration space while minimizing the exploration time and the required number of viewpoints. The 3D map meshes gained are fused together so they can be used by grit-blast planning processes.

The outcome of this research is an integrated system that contains an online, near-optimal exploration technique, a collision-free path planner, and a mapping algorithm, which have been combined and verified experimentally. The algorithms presented have been shown to be a valid solution to the problem of how to safely and efficiently collect the required surface geometry data that is required to build a map of a complex 3D environment. Hence, AXBAM forms an integral part of the robotic grit-blasting system for steel bridge maintenance. Future work will focus on increasing path planning efficiency, improving the map building technique, and identifying the surface material-type during exploration.

Acknowledgments

This work is supported by the ARC Linkage Grant (ARC-LP0776312), by the ARC Centre of Excellence for Autonomous Systems (CAS), the Roads and Traffic Authority (RTA) NSW, and the University of Technology (UTS), Sydney. This work would not have been possible without the assistance of Nathan Kirchner and Ngai Ming Kwok who are valued members of the Centre for Autonomous Systems, UTS.

References

- [1] Callieri, M., Fasano, A., Impoco, G., Cignoni, P., Scopigno, R., Parrini, G., Biagini, G., 2004. Roboscan: an automatic system for accurate and

- unattended 3d scanning. In: Proc. 2nd Intl. Symp. on 3D Data Processing, Visualization and Transmission. Thessaloniki, pp. 805–812.
- [2] Chotiprayanakul, P., Liu, D. K., Wang, D., Dissanayake, G., 2007. A 3-dimensional force field method for robot collision avoidance in bridge maintenance. In: Proc. 24th Intl. Symp. on Automation and Robotics in Construction. Kochin, India, pp. 139–145.
 - [3] Curless, B., Levoy, M., 1996. A volumetric method for building complex models from range images. In: Computer graphics proceedings, annual conference series. Vol. 2006. Association for Computing Machinery SIGGRAPH, New Orleans, pp. 303–312.
 - [4] Elfes, A., 1989. Using occupancy grids for mobile robot perception and navigation. *Computer* 22 (6), 46–57.
 - [5] Fernandez, M., Gupta, K., Fraile, J. C., 2003. Simultaneous path planning and exploration for manipulators with eye and skin sensors. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS. Vol. 1. Las Vegas, pp. 914–919.
 - [6] Guo, Y., Balakrishnan, M., 2006. Complete coverage control for nonholonomic mobile robots in dynamic environments. In: Proc. IEEE International Conference on Robotics and Automation, ICRA. Orlando, pp. 1704–1709.
 - [7] Ihara, S., 1993. *Information Theory for Continuous Systems*. World Scientific, Singapore.
 - [8] Isler, I. V., 2004. Algorithms for distributed and mobile sensing. Ph.D. thesis, University of Pennsylvania.
 - [9] Kirchner, N., Paul, G., Liu, D. K., 2006. Bridge maintenance robotic arm: Mechanical technique to reduce the nozzle force of a sandblasting rig. In: Proc. 1st International Symposium on Digital Manufacturing. Wuhan, China, pp. 12–18.
 - [10] Kummerle, R., Pfaff, P., Triebel, R., Burgard, W., 2008. Active monte carlo localization in outdoor terrains using multi-level surface maps. In: *Autonome Mobile Systeme 2007*. Informatik aktuell. Springer, Heidelberg, pp. 29–35.
 - [11] Kwok, N. M., Ha, Q. P., Liu, D. K., Fang, G., Tan, K. C., 2007. Efficient particle swarm optimization: a termination condition based on the decision-making approach. In: Ha, Q. P. (Ed.), *Evolutionary Computation, 2007. CEC 2007*. IEEE Congress on. pp. 3353–3360.
 - [12] Montemerlo, M., Haehnel, D., Ferguson, D., Triebel, R., Burgard, W., Thayer, S., Whittaker, W., Thrun, S., 2002. A system for three-dimensional robotic mapping of underground mines. Tech. rep., Carnegie-Mellon Univ Pittsburgh, School Of Computer Science.

- [13] Paul, G., Kirchner, N., Liu, D. K., Dissanayake, G., 2009. An effective exploration approach to simultaneous mapping and surface material-type identification of complex 3d environments. *Journal of Field Robotics*, Special Issue on Three-Dimensional Mapping 26 (11-12 SI), 915–933.
- [14] Paul, G., Liu, D. K., Kirchner, N., 2007. An algorithm for surface growing from laser scan generated point clouds. In: Tarn, T., Chen, S., Zhou, C. (Eds.), *Robotic Welding, Intelligence and Automation*. Springer-Verlag, Berlin, pp. 481–491.
- [15] Saez, J. M., Escolano, F., 2004. A global 3d map-building approach using stereo vision. In: *Proc. IEEE International Conference on Robotics and Automation, ICRA*. Vol. 2. New Orleans, pp. 1197–1202.
- [16] Schwarzer, F., Saha, M., Latombe, J., 2002. Exact collision checking of robot paths. In: *Workshop on Algorithmic Foundations of Robotics*. Nice.
- [17] Schwarzer, F., Saha, M., Latombe, J. C., 2005. Adaptive dynamic collision checking for single and multiple articulated robots in complex environments. *Robotics, IEEE Transactions on* 21 (3), 338–353, 1552–3098.
- [18] Shannon, C. E., 1948. A mathematical theory of communication. Reprinted with corrections from *The Bell System Technical Journal* 27, 379–423, 623–656.
- [19] Suján, V. A., Dubowsky, S., 2005. Efficient information-based visual robotic mapping in unstructured environments. *The International Journal of Robotics Research* 24 (4), 275–293.
- [20] Surmann, H., Nuchter, A., Hertzberg, J., 2003. An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. *Robotics and Autonomous Systems* 45 (3-4), 181–198.
- [21] Thrun, S., Martin, C., Yufeng, L., Hahnel, D., Emery-Montemerlo, R., Chakrabarti, D., Burgard, W., 2004. A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots. *IEEE Transactions on Robotics and Automation* 20 (3), 433–443.
- [22] Wang, P., Gupta, K., 2002. Computing c-space entropy for view planning based on beam sensor model. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and System*. Vol. 3. Lausanne, pp. 2389–2394.
- [23] Weingarten, J. W., Gruener, G., Siegwart, R., 2004. Probabilistic plane fitting in 3d and an application to robotic mapping. In: *Proc. IEEE International Conference on Robotics and Automation, ICRA*. Vol. 1. New Orleans, pp. 927–932.

- [24] Wolf, D. F., Sukhatme, G. S., Fox, D., Burgard, W., 2005. Autonomous terrain mapping and classification using hidden markov models. In: Proc. IEEE International Conference on Robotics and Automation, ICRA. Barcelona, pp. 2026–2031.
- [25] Yu, Y., 2000. An information theoretical incremental approach to sensor-based motion planning for eye-in-hand systems. Ph.D. thesis, Simon Fraser University.
- [26] Yu, Y., Gupta, K., 2000. Sensor-based probabilistic roadmaps: experiments with an eye-in-hand system. *Advanced Robotics* 14 (6), 515–536.