

“© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

# Experimental Evaluation of Nearest Neighbour Exploration Approach in Field Environments

Phillip Quin, Gavin Paul, and Dikai Liu

**Abstract**—Inspecting and evaluating surface conditions in 3D environments such as steel bridges is a complex, time-consuming, and often hazardous undertaking that is an essential part of tasks such as bridge maintenance. Developing an autonomous exploration strategy for a mobile climbing robot would allow for such tasks to be completed more quickly and more safely than is possible with human inspectors. The exploration strategy tested in this paper, called the Nearest Neighbours Exploration Approach (NNEA), aims to reduce the overall exploration time by reducing the number of sensor position evaluations that need to be performed. NNEA achieves this by first considering at each time step only a small set of poses near to the current robot as candidates for the Next Best View (NBV). This approach is compared to another exploration strategy for similar robots performing the same task. The improvements between the new and previous strategy are demonstrated through trials on a test rig, and also in field trials on a ferromagnetic bridge structure.

Note to Practitioners: **Abstract**—This paper was motivated by the problem of inspecting confined spaces for rust and flaking paint with a manipulator robot arm. Existing approaches involve creating a large set of candidate robot poses to take a scan from. Evaluating all these candidate poses is very time consuming if full coverage is guaranteed. This paper suggests a principled method for restricting the size of this set in a way that doesn't reduce inspection coverage but decreases overall time taken for inspection.

**Index Terms**—Autonomous exploration, bridge maintenance, manipulator robot.

## I. INTRODUCTION

**A**UTONOMOUS exploration of unknown environments is a complex but important component of many industrial applications for which robots could be used. Various robot systems have been used in industrial applications such as cleaning skyscraper windows [1], ship hull maintenance [2], and inspecting nuclear reactors [3]. A comprehensive summary of robots used in the inspection of man-made structures can be found in [4].

Another set of large structures requiring inspection is bridges. Inspection of bridges and other complex steel structures is a critically important [5], but time-consuming and potentially dangerous task for human engineers [6] as a result of heights, lead-based paints, and confined spaces (Fig. 1). The results of such inspections are subjective, as different human investigators will conceivably give different ratings to the same surface [7]. To eliminate the risk to people and to have objective ratings of surface condition, it is useful to be able to deploy robots into an environment and have

them collect the required information autonomously [8]. The advantage of autonomous robots could be seen during the Fukushima nuclear plant crisis in 2011 when a tethered robot's communications cable became caught and was pulled out of its socket. Not only was the robot unable to complete its mission, it had to be abandoned inside the rubble [9]. Wireless robots also presented problems because limitations on the range of the wireless transmitters meant the robot operators were exposed to radiation risk. Finally, a high-DOF robot can be counter-intuitive to operate and autonomous operation eliminates the need for specialised user training and control interfaces.

Several robotic systems exist for examining bridge decks using manually controlled cameras [10], [11], and others for examining specific elements of a bridge using highly specialised robots [12]. Some work has been done towards a robot able to move not just on flat surfaces like bridge decks, but in the structure of the bridge itself using magnets [13]. Aerial inspection of bridges using UAVs is also possible [14], though UAVs are not suited for confined spaces inside a bridge girder. Magnetic adhesion is also used in the Mag-Foot system, which consists of two magnets at either end of a kinematic chain with three degrees of freedom [15].

These examples of bridge inspection present platforms capable of performing the inspection task in some circumstances, but do not present algorithms enabling a complex 7DOF robot platform to perform inspection autonomously.

The robot used in field trials in this paper is a robot with seven degrees of freedom (7DOF) based on an inchworm-like design which extends work by [16]. At either end of the robot a magnetic footpad is affixed which can be securely attached to ferromagnetic surfaces [17]. This design allows the robot to perform acute surface transitions, making it possible to traverse complex and obstructed environments. In addition to the magnetic footpads, a sensor such as the PrimeSense depth camera can be attached to either end.

To accomplish full exploration of a previously unknown environment, robots must be able to determine where it is safe to move, i.e. avoid known obstacles and unknown space. They must be able to determine where the information they need to collect is located, if any remains, and how to go about collecting it.

There are many strategies for exploring in two dimensional (2D) environments, such as frontier-based exploration [18], and potential-field exploration [19]. Frontiers and potential fields have been combined and demonstrated in real two dimensional (2D) environments by Shade and Newman [20], with applications in three dimensions (3D) being demonstrated in simulation. Many such strategies rely on abstracting the

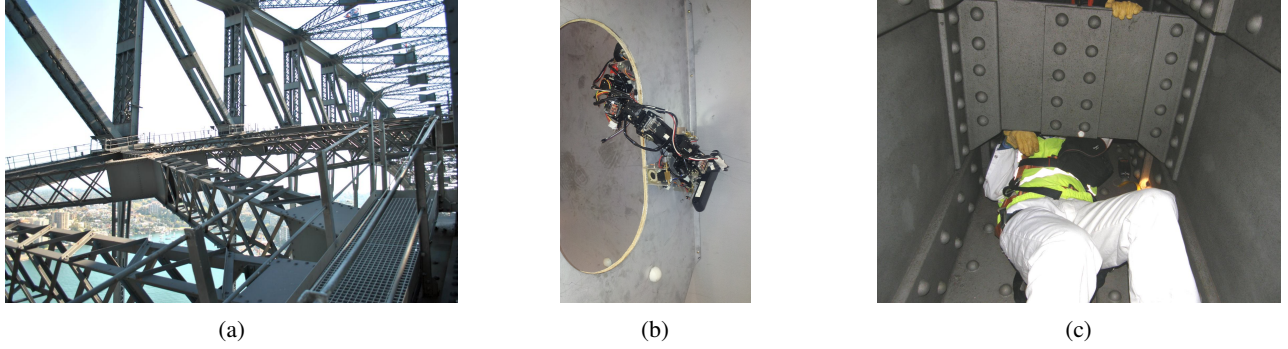


Fig. 1: (a) An example of a complex 3D environment: structural beams of a ferromagnetic bridge. (b) A tunnel blocked by regular obstacles, and the robot navigating its way past. (c) An inspector moving through a confined space. Navigating any of these environments would be very difficult without a high-DOF robot, capable of performing complex maneuvers.

robot as a point robot, therefore simplifying the problem of determining where it is feasible for the robot to move. Such approximations are not suited to robots with high degrees of freedom (DOF), nor are they suited to environments such as Fig. 1 where robots confined to a single plane would find certain obstacles impassable.

There are several examples of exploring environments with a robot manipulator using the principles of frontier exploration [21], which can be combined with other techniques such as detecting the location of voids in the robot's map [22]. [23] samples the exploration space to determine a Next Best View (NBV), and uses probabilistic road maps (PRMs) to create a motion plan from the robot's current position to the NBV; each path to the NBV has an estimated information gain, and the most information-rich path is selected.

Exploration strategies exist that make use of configuration space (C-space), or a combination of C-space and physical Euclidean space [24], [25]. Such approaches eliminate the need for expensive inverse kinematic computations and naturally bound exploration to feasible robot positions.

This paper presents field trial results of an exploration approach that makes use, where possible, of nearest neighbouring poses in the robot's configuration space [26], [27]. An overview of the approach is given in Section II, along with a theoretical analysis of running times in Section III. Section IV-A describes the steel bridge environment in which experiments are conducted and the robot being used. Section IV-B discusses the design of the experiments conducted, and the engineering issues encountered during experiments are described in Section IV-C, along with proposed solutions. The results, which include a comparison to another exploration strategy tailored to the same exploration scenario [28], [29], are given in Section IV-D. The manner in which ground truth values were determined for the experiments is discussed Section V along with an analysis of various metrics. Conclusions and future work are presented in Section VI.

## II. NEAREST NEIGHBOURS EXPLORATION APPROACH

The Nearest Neighbours NBV algorithm (Algorithm 1) is used each time a NBV must be chosen. It involves evaluating a small set of configurations neighbouring the current robot configuration (Algorithm 2). If no valid neighbouring configuration is found, the entire robot's C-space is sampled and a

NBV is selected from these samples (Algorithm 3). Exploration terminates when no NBV can be found in this C-space sampling (see Fig. 2). The map fusion process that occurs after each observation, and potentially involves a SLAM step, is handled separately by another module, and hence is outside the scope of this paper. The following subsections provide more detail about each iteration of the algorithm.

### A. Definitions and Notation

A robot, denoted  $\mathcal{A}$ , begins in a pose,  $q_{curr}^0$  which is a vector of joint angles  $[\theta_1, \dots, \theta_j]$  where  $j$  is the number of degrees of freedom (DOF) possessed by the robot. A vector,  $q$  of robot joint angles is therefore a particular point in the robot's configuration space, denoted  $\mathcal{C}$ , such that  $q \in \mathcal{C}$ .

The robot is affixed with a sensor. Given a configuration  $q$  and using forward kinematics, a homogenous transform matrix  ${}^oT_s^q$  is calculated which represents the sensor's position and orientation in Euclidean space relative to the robot's base.

$\mathcal{P}$  represents physical Euclidean space ( $\mathbb{R}^2$  for the 2D case or  $\mathbb{R}^3$  for the 3D case), which is discretised into grid cells for the 2D case, or voxels for the 3D case. In the rest of this paper, it is assumed that the 3D case is the one being considered.

The number  $i \in \mathbb{N}^0$  represents the iteration of exploration the algorithm is currently in (i.e. the number of times the sensor has taken a scan of the environment). The space known

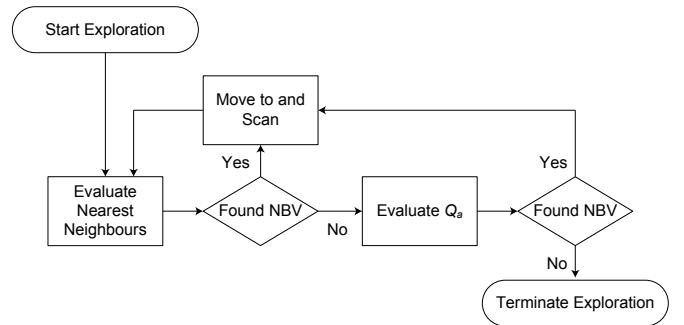


Fig. 2: An overview of exploration using the Nearest Neighbours Exploration Approach [26], [27]

to be free or known to be occupied by obstacles at iteration  $i$  can therefore be represented as  $\mathcal{P}_{free}^i$  and  $\mathcal{P}_{occ}^i$ , respectively.  $\mathcal{P}_{unk}^i$  denotes space whose occupancy value is unknown; it is not known to be free, nor is it known to contain an obstacle. The volume of space known to be free before exploration must be initialised such that,  $\mathcal{P}_{free}^0 \neq \emptyset$ , in order for the robot to have space in which to perform initial exploration.

Given a robot configuration  $q \in \mathcal{C}$ ,  $\mathcal{A}(q)$  is the volume of physical space taken up by the robot, such that  $\mathcal{A}(q) \subset \mathcal{P}$ . In this paper, the volume is determined using minimum bounding ellipsoids [30]. A particular pose  $q$  is known to be in collision if  $\mathcal{A}(q) \cap \mathcal{P}_{occ}^i \neq \emptyset$ . Conversely however, if  $\mathcal{A}(q) \cap \mathcal{P}_{occ}^i = \emptyset$ , it must still be determined whether  $\mathcal{A}(q)$  intersects with  $\mathcal{P}_{unk}^i$ . If  $\mathcal{A}(q) \cap \mathcal{P}_{unk}^i = \emptyset$ , then  $q$  is known to be a valid robot configuration. If  $\mathcal{A}(q) \cap \mathcal{P}_{unk}^i \neq \emptyset$ , then the validity of  $q$  is unknown.

### B. Finding Nearest Neighbour NBV

Assuming the robot's current configuration at iteration  $i$ ,  $q_{curr}^i$ , the set of nearest neighbours  $Q_n^i$ , is constructed. For each  $\theta_k$  in  $q_{curr}^i$  where  $k \in [1, \dots, j]$ , a pair of vectors equal to  $q_{curr}^i$  is added to an initially empty  $Q_n$ , one in which a chosen angle  $\Delta\theta$  is added to  $\theta_k$  and another where  $\Delta\theta$  is subtracted from  $\theta_k$ ,

$$Q_n^i = \{q_{curr}^i + I_j[k; 1, j] \times \Delta\theta \mid k \in 1, \dots, j\} \cup \{q_{curr}^i - I_j[k; 1, j] \times \Delta\theta \mid k \in 1, \dots, j\} \quad (1)$$

where  $I_j[k; 1, j]$  denotes the  $k^{th}$  row of the identity matrix  $I_j$ . The resulting set  $Q_n^i$  has a size of  $2 \times j$ . Given any  $q \in Q_n^i$  and defining  $\Delta_q = q - q_{curr}^i$ , an ordered set of configurations  $P_q$  can be defined representing a direct path from  $q_{curr}^i$  to  $q$ .

$$P_q = \{q_{curr}^i + \Delta_q \times v \mid v \in \mathbb{R}, 0 \leq v \leq 1\} \quad (2)$$

Defining  $\mathcal{A}(P_q)$  as the union of the physical space taken up by all poses in set  $P_q$ , the angle  $\Delta\theta$  is chosen so that for all  $q \in Q_n^i$ , and all resulting  $P_q$ ,

$$\mathcal{A}(P_q) = \mathcal{A}(q_{curr}^i) \cup \mathcal{A}(q) \quad (3)$$

such that if  $(\mathcal{A}(q_{curr}^i) \cup \mathcal{A}(q)) \cap \mathcal{P}_{occ}^i \neq \emptyset$ , then  $(\bigcup \mathcal{A}(p), \forall p \in P_q) \cap \mathcal{P}_{occ}^i \neq \emptyset$ . This means that  $q_{curr}^i$  and  $q$  are sufficiently close to one another and that if both poses are known to be valid robot configurations, then there are no obstacles between them and the robot can safely move directly from one pose to the other without performing further path planning or obstacle avoidance (see Fig. 3). Since the robot is in configuration  $q_{curr}^i$ , it can be assumed to be a valid pose. It is therefore only necessary to verify the validity of all  $q \in Q_n^i$ , and remove from  $Q_n^i$  any  $q$  where  $\mathcal{A}(q) \cap \mathcal{P}_{occ}^i \neq \emptyset$  or  $\mathcal{A}(q) \cap \mathcal{P}_{unk}^i \neq \emptyset$  (i.e. where  $q$  results in the robot being in a collision with obstacles or unknown space). Note that this is only possible when  $\mathcal{P}$  is discretised, and the resolution of this discretisation will determine the value of  $\Delta\theta$ .

Choosing a smaller  $\Delta\theta$  will still result in safe robot motion, but as neighbouring configurations are closer together, the observations made from those configurations are likely to

overlap more, making it more likely that less new information will be collected.

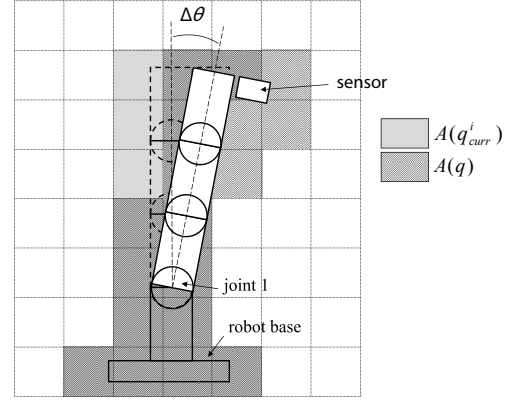


Fig. 3: Choosing  $\Delta\theta$  based on the discretisation of  $\mathcal{P}$ .

Once the set  $Q_n^i$  is constructed and invalid configurations removed, each configuration  $q \in Q_n^i$  has an information value calculated,  $\mathcal{H}(q)$ , which estimates the amount of new information that would be added to the environment representation by a sensor scan performed with the robot in that configuration<sup>1</sup>. The estimated information value,  $\mathcal{H}(q)$ , of each  $q$  is calculated by raytracing [31] from  ${}^oT_s^q$  into  $\mathcal{P}$  along rays that represent the sensor's field of view (FOV). Starting at the origin of each ray and moving along it, traversed voxels that are in  $\mathcal{P}_{unk}^i$  are added to the set of unknown voxels that are expected to be covered by the sensor at that position and orientation. The ray is traced until a voxel in  $\mathcal{P}_{occ}^i$  is encountered or the maximum range of the sensor is reached. The estimated information value for the configuration  $q$  is the size of the set of voxels created.

A threshold value  $\tau_n$  is chosen, which represents a minimum value of information gain required by a configuration,  $q$  in order for it to be considered as a potential NBV. Any  $q \in Q_n^i$  where  $\mathcal{H}(q) \leq \tau_n$  is discarded. The remaining configurations in  $Q_n$  are ranked by information value  $\mathcal{H}(q)$  and the one with the highest value is chosen as the NBV,  $q_{nbv}^i$ . The robot then moves directly to  $q_{nbv}^i$  and a scan is taken with the sensor.

If  $Q_n$  becomes the empty set as a result of discarding configurations, then the algorithm proceeds to the next phase, which is the sampling of all C-space.

### C. Sampling all C-space for NBV

At iteration 0, a set  $Q_a$  of configurations is created which contains a representative sampling of the robot's entire C-space. If  $x$  samples are selected for each joint, and given the robot has  $j$  joints,  $Q_a$  is then the set of all combinations of the samples for each joint, giving a total of  $x^j$  configurations (e.g. taking 3 samples from each of 7 joints would result in  $3^7$  or 2187 robot configurations).

An example method for selecting joint angles is shown in Fig. 4, where samples denoted  $\beta_1$  through to  $\beta_x$  are spread equally apart in the joint space.

<sup>1</sup> $\mathcal{H}$  is often used to denote entropy, which measures the lack of information in a system. For simplicity however, the notation is assumed here to directly represent a measure of information.



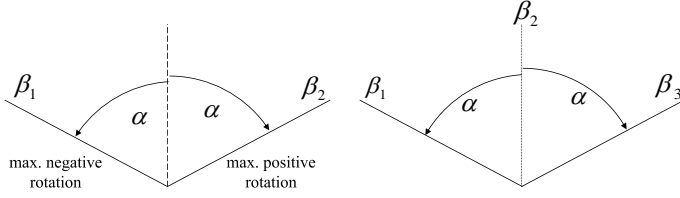


Fig. 4: Taking sample angles from a joint's range of motion: spreading samples  $\beta_1$  to  $\beta_x$  evenly (left:  $x = 2$ , right:  $x = 3$ ).

When no neighbouring configuration is selected as the NBV, set  $Q_a$  is evaluated to determine which member configurations are valid in the current iteration. Only poses that do not collide with unknown space or with obstacles need be considered as potential NBVs. Three subsets of  $Q_a$  can be defined,  $Q_{valid}$ ,  $Q_{invalid}$ , and  $Q_{unk}$ , representing configurations that do not result in collisions with  $\mathcal{P}_{occ}^i$ , configurations that *do* result in collisions with  $\mathcal{P}_{occ}^i$ , and configurations that collide with unknown space  $\mathcal{P}_{unk}^i$  (but not obstacles  $\mathcal{P}_{occ}^i$ ) respectively.

$$Q_{invalid} = \{q|q \in Q_a, \mathcal{A}(q) \cap \mathcal{P}_{occ}^i \neq \emptyset\} \quad (4)$$

$$Q_{unk} = \{q|q \in (Q_a \setminus Q_{invalid}), \mathcal{A}(q) \cap \mathcal{P}_{unk}^i \neq \emptyset\} \quad (5)$$

$$Q_{valid} = (Q_a \setminus Q_{invalid}) \setminus Q_{unk} \quad (6)$$

Note for two sets  $A$  and  $B$ , the notation  $A \setminus B$  is used to represent the relative complement of  $B$  in  $A$ , i.e. the set of elements in  $A$  but not in  $B$ .

A function  $Valid(Q, M)$  is defined which takes a set of poses,  $Q$ , and the map,  $M$ , and returns two sets of poses,  $Q_{valid}$  and  $Q_{unk}$ , which contain respectively, known safe poses, and poses that result in part of the robot intersecting with unknown space.

For each of the  $q \in Q_{valid}$ ,  $\mathcal{H}(q)$  is calculated as before in the case of neighbour configurations. A threshold value  $\tau_a$  is chosen, which may or may not be chosen to be equal to  $\tau_n$ , and that represents the minimum value of information gain required by a configuration,  $q$  in order for it to be considered as a potential NBV. Any  $q \in Q_{valid}$  where  $\mathcal{H}(q) \leq \tau_a$  is discarded. If  $Q_{valid}$  becomes the empty set then exploration is terminated. In addition to the information score, the effort required to move from  $q_{curr}^i$  to each  $q$  is determined,  $e([q_{curr}^i, q])$ , defined by the equation:

$$e([q_1, \dots, q_n]) = \sum_{i=2}^n \|q_i - q_{i-1}\| \quad (7)$$

Configurations in  $Q_{valid}$  are ranked by the ratio of  $\mathcal{H}(q)$  to  $e([q_{curr}^i, q])$ , and the highest ranking is chosen as the NBV,  $q_{nbv}^i$ . This ensures that the most expected information is collected per unit of joint effort. Choosing the best configuration based on a weighted sum of information gain and joint effort would also be suitable.

Once the NBV has been selected,  $Q_a$  becomes  $Q_{unk} \cup Q_{valid}$ . The configuration  $q_{nbv}^i$  can also be removed from  $Q_a$ , but would naturally be removed in a subsequent iteration if its information value fell below  $\tau_a$ . Upon reaching  $q_{nbv}^i$  and performing a scan, the exploration algorithms reverts to

choosing the NBV from neighbouring poses.

---

#### Algorithm 1: NNEA

---

**Input:**  $q_{curr}$   $\leftarrow$  Current robot pose,  $M$   $\leftarrow$  Octomap,  
 $Q$   $\leftarrow$  Set of poses under consideration

**Output:**  $q_{nbv}, Q$

```

1  $Q_n = \text{Generate\_Neighbour\_Poses}(q_{curr}, M);$ 
2  $q_{nbv} = \emptyset;$ 
3  $best\_neighbour\_gain = 0;$ 
4 if  $Q_n \neq \emptyset$  then
5   for  $q \in Q_n$  do
6      $gain = \mathcal{H}(q);$ 
7     if  $gain > best\_neighbour\_gain$  then
8        $best\_neighbour\_gain = gain;$ 
9        $q_{nbv} = q;$ 
10 if  $q_{nbv} == \emptyset$  then
11    $q_{nbv}, Q = \text{AXBAM\_NBV}(q_{curr}, M, Q);$ 
```

---



---

#### Algorithm 2: Generate\_Neighbour\_Poses

---

**Input:**  $q_{curr} = (\theta_1, \dots, \theta_j)$ ,  $M$   $\leftarrow$  Octomap,  
 $\Delta_q$   $\leftarrow$  Step size

**Output:**  $Q_{valid}$

```

1 for  $i \in [1, j]$  do
2    $q_{new} = q_{curr};$ 
3    $q_{new}.\theta_i = q.\theta_i + \Delta_q;$ 
4    $Q_n = Q_n \cup q_{new};$ 
5    $q_{new}.\theta_i = q.\theta_i - \Delta_q;$ 
6    $Q_n = Q_n \cup q_{new};$ 
7  $Q_{valid}, Q_{unk} = \text{Valid}(Q_n, M);$ 
```

---

### III. THEORETICAL ANALYSIS

The underlying motivation for NNEA is to reduce the amount of candidate viewpoints that will need to be evaluated while still offering near-optimal results, optimal being defined as exploring the largest amount of the environment in the least amount of time, and with the least amount of associated motion.

Consider an optimal planner, choosing between  $n$  candidate viewpoints. In the first iteration, all  $n$  will be evaluated before one is chosen as the optimal choice. Assuming that once selected, a candidate will never be reselected, then in the second iteration,  $n - 1$  candidates will be evaluated, and so on, until one final evaluation is made which determines there are no more observations worth making.

If  $k$  observations chosen in this way are required before the environment is fully explored, then the most number of evaluations performed by the optimal exploration algorithm will be:

$$\sum_{i=n-k}^n i = \frac{(k+1)(2n-k)}{2} \quad (8)$$

By contrast consider the evaluations required by NNEA. Since viewpoints are selected within a neighbouring region of

---

**Algorithm 3: AXBAM\_Next\_Best\_View**


---

**Input:**  $q_{curr} \leftarrow$  Current robot pose,  $M \leftarrow$  Octomap,  
 $Q \leftarrow$  Poses under consideration

**Output:**  $q_{nbv}, Q$

```

1  $Q_{valid}, Q_{unk} = \text{Valid}(Q, M)$ ;
2  $Candidates = \emptyset$ ;
3  $q_{nbv} = \emptyset$ ;
4 for  $q \in Q_{valid}$  do
5    $gain = \mathcal{H}(q)$ ;
6    $effort \leftarrow e([q_{curr}, q])$ ;
7   if  $gain == 0$  then
8      $Q_{valid} = Q_{valid} - q$ ;
9   else
10     $candidate = (q, gain, effort)$ ;
11     $Candidates = Candidates \cup candidate$ ;
12 if  $Candidates \neq \emptyset$  then
13    $Candidates = \text{ParetoFrontier}(Candidates)$ ;
14    $q_{nbv} = \text{GetHighestRanked}(Candidates)$ ;
15    $Q_{valid} = Q_{valid} - q_{nbv}$ ;
16  $Q = Q_{valid} \cup Q_{unk}$ ;

```

---

the current viewpoint, they will result in overlapping observations. Assume that for every observation made by an optimal planner,  $\alpha$  observations will need to be made by NNEA to collect the same information, and that at each iteration, NNEA evaluates  $c$  neighbouring candidates. Whenever a neighbouring pose is not chosen, the set  $Q_a$  will be evaluated.

Assuming that  $m$  observations of the total  $k$  are successfully performed by NNEA, rather than by evaluating  $Q_a$ , then the number of  $Q_a$  evaluations saved by NNEA compared to the optimal planner is at least  $\sum_{i=n-k}^{n-k+m} i$ .

The total evaluations required by NNEA is therefore at most:

$$\underbrace{\alpha \times c \times m}_{\text{Successful NNEA steps}} + \underbrace{(k-m) \times c}_{\text{Unsuccessful NNEA steps}} + \underbrace{\sum_{i=n-k+m}^n i}_{\text{Evaluating } Q_a} \quad (9)$$

For NNEA to be worthwhile, the number of evaluations that take place as part of the neighbours step must be smaller than the savings on evaluations of  $Q_a$ . Equation 9 is subtracted from Equation 8:

$$\begin{aligned} & \left( \sum_{i=n-k}^n i \right) - \left( \alpha \times c \times m + (k-m) \times c + \sum_{i=n-k+m}^n i \right) \\ &= \left( \sum_{i=n-k}^{n-k+m} i \right) - \alpha \times c \times m - (k-m) \times c \\ &= \frac{m^2}{2} + \left( n-k + c(1-\alpha) - \frac{1}{2} \right) m - kc \end{aligned} \quad (10)$$

To get a sense of when NNEA results in a benefit, Equation (10) can be plotted with example values of  $\alpha$  and  $c$ . The number of neighbours evaluated at each step,  $c$ , was chosen to

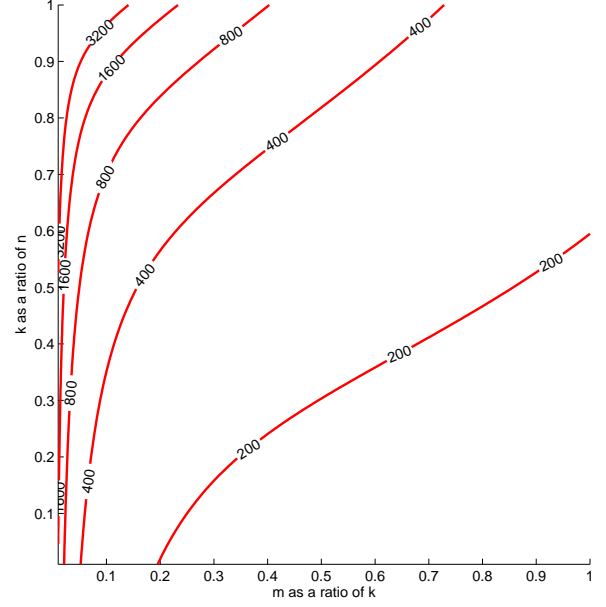


Fig. 5: The values of  $m$  and  $k$  for which NNEA (red) breaks even with the optimal planner if  $\alpha = 10$  and  $c = 14$ . The values of  $k$  and  $m$  below the lines are where NNEA results in fewer viewpoint evaluations.

be 14, and the amount of overlap between neighbouring poses was expected to be such that  $\alpha = 10$ .

As shown in Figure 5, by the time  $n = 800$ , at worst only 40% of the required observations need to be made by sequences of neighbouring observations for NNEA to be worthwhile, but if only 70% of the  $n$  possible viewpoints are required, then NNEA will be worthwhile if it results in 10% of the observed space. Figure 6 shows the result of equation (10) for a single value of  $n$ .

#### IV. EXPERIMENTS

This section details the experiments used to evaluate the performance of NNEA. First the robot, equipment, and the environments are described, followed by an explanation of how experiments are conducted, and descriptions of how particular implementation issues are resolved. Finally, results are presented and analysed.

##### A. Robot and Environment

The experiments in this paper involved exploring the volume of space surrounding the robot when a footpad was attached in a single position. In this configuration, the magnetic footpad at the free end of the robot was removed from the robot, and a Carmine 1.09 Primesense depth-camera sensor was attached to this end only<sup>2</sup>, forming an eye-in-hand arrangement.

<sup>2</sup>It is possible to have both a footpad and a sensor at either end simultaneously, however the torque generated from such an arrangement is such that movement with the prototype robot becomes affected. Newer models of motors have been released that resolve this issue.

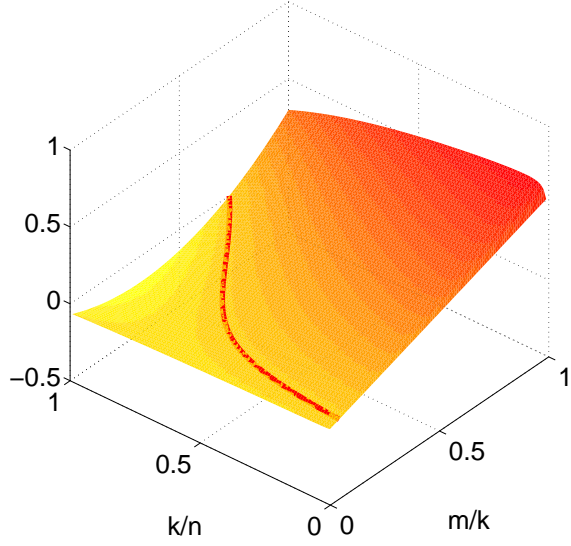


Fig. 6: The percentage of calculations saved by NNEA, shown in the vertical axis, if  $n = 400$ ,  $\alpha = 10$ , and  $c = 14$ . The horizontal axes describe  $m$  as a ratio of  $k$ , and  $k$  as a ratio of  $n$ . The red line shows when NNEA offers no advantage or disadvantage.

The Primesense depth-camera allows depth readings to be taken, so that a mesh of the environment can be constructed, and RGB images, which can be used for inspection either manually or with an automated algorithm. Since the depth image and RGB images are taken simultaneously, it is possible to associate the RGB images with a particular location on the final 3D mesh. As a result, a clearer model of the bridge's condition can be created which will assist in choosing any required maintenance actions.

The experiments were conducted in two separate environments. The first was a ferromagnetic rig built to be as similar as possible to a representative semi-enclosed tunnel in one part of a bridge structure. This mock environment includes examples of obstacles such as rivets, a manhole and a partition plate that make inspection by humans difficult. The robot was placed in five different positions of which two are shown in Fig. 7. The others positions were on the left, bottom and back surfaces.

The second environment was the inside of a base girder, part of a real ferromagnetic bridge (Fig. 8). It is noted that the tunnel in this environment had a width of 0.7 meters. Given the minimum sensor range of 0.3 meters, this means that the set of poses allowing the sidewalls of the tunnel to be detected is small and the robot is forced to explore whilst in close proximity to obstacles. This constraint could be reduced by using a sensor with a smaller minimum range. The set  $Q_a$  can also be made larger, making it more likely that suitable poses will be found that collect sufficient information and are further from any obstacle.

The robot began with a map, represented using an octomap structure [32], that contained no information about the en-

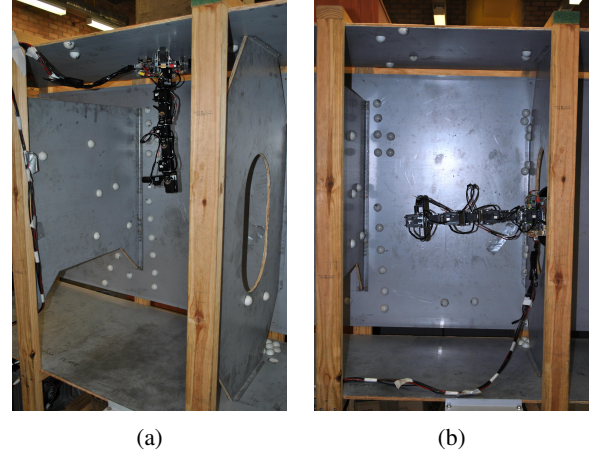


Fig. 7: The robot in (a) lab site 1, (b) lab site 5.

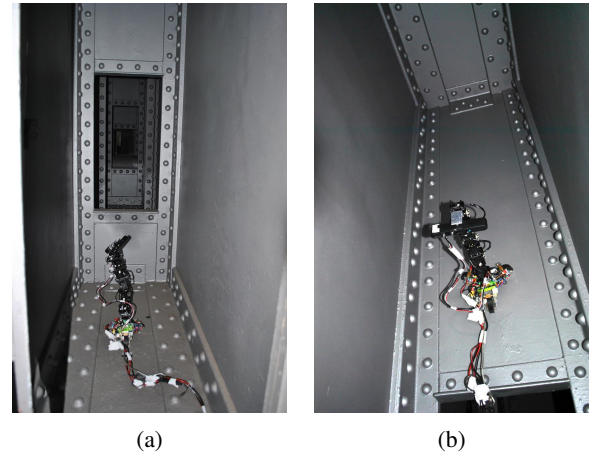


Fig. 8: The robot positioned at (a) field site 1 (on the floor of the girder), (b) field site 2 (above the manhole).

vironment other than a volume of space known to be free in which the robot could begin its exploration. This initial space is necessarily constrictive ( $0.7 \times 0.7 \times 0.7 m^3$ ), since the environments themselves are confined.

## B. Experiment Design

The first goal of the experiments was to demonstrate that NNEA was effective, and could collect sufficient amounts of geometric information about the environment. AXBAM provided a minimum benchmark for how much information could be collected. Since there is no widely used and agreed upon exploration approach for manipulator robots, AXBAM was selected as a comparison to NNEA. This is because of AXBAM's applicability to the same robot platform and exploration scenario and the fact that it has been tested in field experiments [28].

The second goal of the experiments was to determine whether or not NNEA performed better on average than AXBAM. NNEA would be considered as having performed better than AXBAM in the field trials if a similar or greater amount of information was collected whilst requiring less time, and similar or lesser robot total joint effort. The hypothesis

prior to conducting the field trial experiments was that NNEA would require fewer gain calculations, and that this would result in shorter exploration running times.

Time profiles were kept for all algorithms executed. In the case where one algorithm outperformed the other, the collected time profiles could be analysed to determine the cause of the performance difference, and how this related to theoretical expectations [26].

The metrics tracked and measured are:

- **Time:** amount of time taken for the algorithms to run to completion,
- **Number of Scans:** the number of scans taken,
- **Number of Viewpoint Evaluations:** the number of information gain calculations performed,
- **Information:** the number of voxels known to be free or occupied as a result of exploration. This is determined by counting the unknown voxels within a large bounding box centered on the robot before and after exploration terminates,
- **Total Joint Effort:** the amount of motion performed, in radians, by the robot's joints (see Equation 7),
- **Maximal Joint Effort:** since the robot joints move simultaneously when moving from one configuration to another, summing the largest difference between configurations in a sequence of poses is assumed to give a better measure of the time taken to move. Given a sequence of configurations  $[q_1, \dots, q_n]$ , this value is given by  $e_{par}([q_1, \dots, q_n])$  (11).

$$e_{par}([q_1, \dots, q_n]) = \sum_{i=2}^n \max(|q_i - q_{i-1}|) \quad (11)$$

Both algorithms were tested four times (only three times in the girder environment), each time with different sized sets  $Q_a$ , to determine the effect of using larger sets. Based on previous simulation work [26], it was hypothesized that larger sets would take more time, but result in more information being collected. It was also expected that any difference between NNEA and AXBAM's performances would be exacerbated by an increase in the size of the pose set  $Q_a$ .

In both environments, both AXBAM and NNEA algorithms were run from several different starting locations with the robot base attached to different surfaces in the environment. Five different base positions were used for the rig environment, and two base positions for the bridge environment.

A concern with running NNEA in such confined environments was that given the limited space (most spaces were only 0.7 meters wide) and given such a large minimum sensor range relative to the tunnel width (0.3 meters minimum range) a high proportion of neighbour calls (Algorithm 3) might fail, and that NNEA would in this instance revert into AXBAM plus computational overhead. If this were the case, then it would be expected that NNEA would perform worse than AXBAM alone.

### C. Engineering Challenges

Performing tests in field environments means taking into consideration issues that might not occur in simulations.

Several of these issues are discussed here, including sensor minimum range, robot self-scans, and deflection of robot joints due to gravity. The methods used to manage these issues are also presented.

1) *Sensor Minimum Range:* Sensors like the Primesense depth camera used in the field trials have a minimum range,  $R_{min}$ . If an obstacle exists within  $R_{min}$  of the sensor origin (e.g. 0.5 meters for the Primsense), then the returned reading will be the same as the returned reading would be for an out of range reading (i.e. free space as far as the sensor can detect).

This creates a situation where the robot system, when being given a range reading of 0, must be able to correctly discriminate between the two possible scenarios. In the case of out-of-range readings, the map should have the appropriate voxels marked as free space. In the case of a minimum range reading, then no change should be made to the map.

One method to determine which scenario has occurred is to raycast from the sensor's origin to the returned ray's endpoint into the robot's map of the environment. If the ray encounters an unknown voxel inside the minimum range, then the 0 reading should be treated as if it were the minimum range case, since it is possible some unseen obstacle lies in the unknown space (Fig. 10). If the ray passes beyond  $R_{min}$  without encountering any unknown voxels, then the reading must be from the out-of-range case, and the appropriate voxels can be marked as free.

This method means that any gain estimation function must also deal with minimum range in a similar way if it is to correctly estimate the amount of information that can be collected from a particular configuration  $q$  at iteration  $i$ . Rays are cast from the viewpoint origin in the map to the ray endpoints that define the maximum trusted range of the Primesense sensor. These need only be of sufficient resolution to guarantee all voxels in the sensor FOV will be traversed, and they do not need to be of the same resolution as the sensor.

Upon encountering an unknown voxel inside the sensor minimum range, the gain estimation for that ray terminates with a value of 0. As shown in Fig. 9b, all other rays that do not penetrate an unknown voxel within the minimum range are projected into the map as normal. Failure to account for minimum range in this manner would mean potentially overestimating the information that a scan would return from a particular viewpoint, and such viewpoints would be repeatedly evaluated rather than discarded or set aside.

2) *Managing Occlusion of Sensor FOV by the Robot:* When using sensors in the field, it is possible that a robot, particularly a robot with high joint redundancy, will be visible in some sensor scans. It is obviously undesirable for the robot to be added to the environment map, and it is undesirable to overestimate what can be seen from a particular point of view by failing to take into account the occlusion of the sensor FOV by the robot itself.

Many collision avoidance algorithms use minimum bounding ellipsoids to encapsulate the space taken up by a robot. This paper makes use of algorithms presented in [30] to develop such a set of ellipsoids. Scans of the robot can therefore be removed from the point cloud returned by a sensor

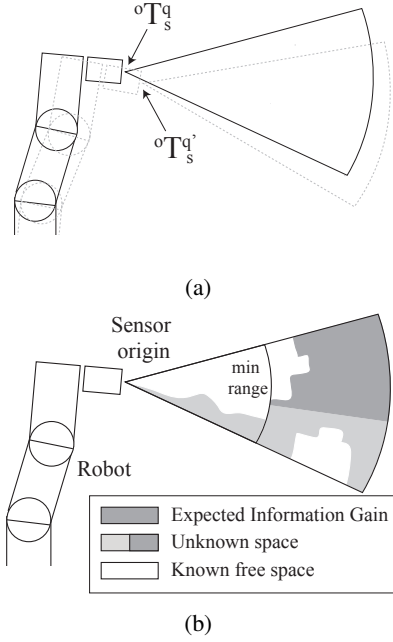


Fig. 9: (a) Joint deflection resulting in a desired configuration  $q$  becoming the actual configuration  $q'$  and therefore in sensor location  ${}^oT_s^q$  becoming  ${}^oT_s^{q'}$ . (b) Counting information gain with unknown inside minimum range.

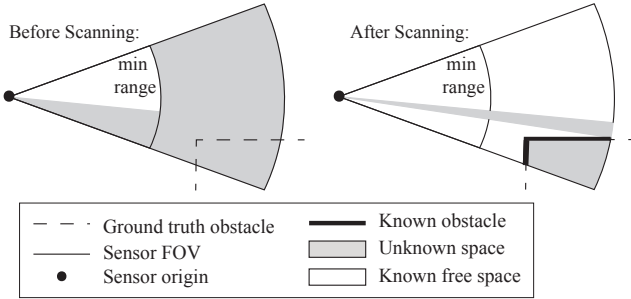


Fig. 10: Filling in free space when receiving 0 range reading.

scan by deleting vertices or points that lie in the space within any of the bounding ellipsoids.

These ellipsoids are also used to account for robot occlusion of the sensor when estimating the information value of a candidate sensor viewpoint. The normal ray endpoints used when raycasting should simply be replaced with the intersections of the sensor rays with the robot's ellipsoids, ignoring the ellipsoid that contains the sensor.

3) *Deflection of Robot Joints Due to Gravity*: An initial assumption of the implementation of NNEA for these field trials was the use of a rigid bodied robot such as an industrial robot, and that as a result, there would be little pose uncertainty. This assumption did not hold with the final robot used, and deflection in the slightly non-rigid joints due to gravity meant that the robot was not where it believed itself to be when performing a motion or holding a position (see Fig. 9a). This could conceivably result in collisions with obstacles, and meant that there was some error when adding scan data into

the map (see Fig. 12). Incorrect map fusion would also reduce the accuracy of the information estimation function,  $\mathcal{H}(q)$ .

Collisions were eliminated by making the ellipsoids used to avoid collisions with the environment larger than they would otherwise be. In a similar way, the joint space of the robot was reduced to avoid collisions with itself. Limiting joint angles and creating more conservative obstacle avoidance meant that both exploration strategies had fewer possible poses to choose from when scanning. This slightly reduced the amount of information that could possibly be collected by either algorithm.

#### D. Results

In both environments, in all experiments where  $|Q_a| \approx 2000$  or greater (a total of 19 trials), NNEA required less time than AXBAM to collect at least similar amounts of information ( $\pm 5\%$ ) or greater (in one case 22.56% more). The reduction in time spent exploring ranged between 10.40% up to 64.43%. One exception was where NNEA, run with  $|Q_a| \approx 2000$ , collected 7.18% less information than collected by AXBAM, though in 49.57% less time. Meshes of the environments resulting from exploration can be seen in Fig. 12, showing that the majority of surfaces were observed by the sensor.

The reflectivity of the surface was such that if an observation was made from an angle almost parallel to the plane, some surface areas would be patchy and noisy. Despite this, NNEA's performance, and its performance relative to the comparison algorithm, seemed unaffected.

For experiments where  $|Q_a| \approx 100$ , in 6 out of 7 experiments, NNEA took longer than AXBAM to terminate but collected more information than AXBAM (ranging from an additional 1.66% to 159.42%). In field site 2, the case where  $|Q_a| \approx 100$  and where NNEA terminated sooner than AXBAM, 81.59% more information was collected with NNEA than with AXBAM.

Fig. 13a, which plots the time taken by each trial to complete exploration on a logarithmic axis, shows the exponentially growing difference in termination times as  $Q_a$  trends towards larger sizes.

The effect of increasing the size of  $Q_a$  is shown in Fig. 13b for each environment, and for each algorithm. The maximum amount of information was, for each set of trials, defined as the largest amount of information that was collected by any algorithm, with any  $|Q_a|$ . The only exception to this was for results of the trials at field sites 1 and 2, which involved environments where an estimate of the total expected discoverable information could be determined (see Section V-B). Note that, due to how configurations are culled from  $Q_a$  and the effect of minimum sensor range on a configurations estimated information value (See Section IV-C1), increasing  $|Q_a|$  does not guarantee that more information is collected, though it does increase the likelihood of collecting more information. It can be seen that in all but one case, in order to allow AXBAM to collect amounts of information closer to NNEA, the set  $Q_a$  was required to be of size  $3^7$  and above.

A benefit of NNEA over AXBAM found through previous simulation work [26], and confirmed here in field trials (Fig.



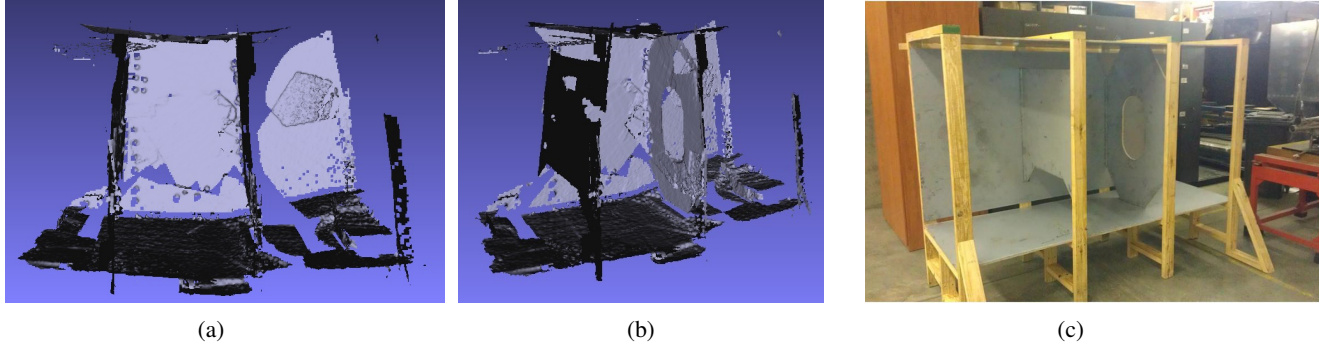


Fig. 11: Meshes resulting from exploration in the mock rig (a) the mock rig (front view), (b) the mock rig (angled view), and (c) a photo of the mock rig.

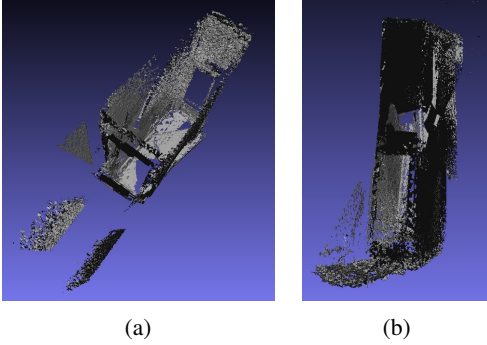


Fig. 12: Meshes resulting from exploration in (a) field site 1 and (b) field site 2.

13) is that the sequence of configurations required to explore the environment requires less joint effort, and less maximal joint effort. The increase of required maximal effort when exploring with larger sized sets  $Q_a$  is consistent with the fact that more scans are performed and more information is collected in the general case.

Fig. 13a shows the averaged time profile information collected for each trial. The tasks represented are:

- Scanning: Collecting a scan with the sensor and integrating this into the environment representation,
- Viewpoint Evaluations: evaluating configurations to determine their estimated information value,
- Validating Configurations: determining whether configurations are members of  $Q_{invalid}$ ,  $Q_{unk}$ , or  $Q_{valid}$ ,
- Path Planning: creating a trajectory from  $q_{curr}^i$  to  $q_{nbv}^i$ ,
- Moving: moving the robot to  $q_{nbv}^i$ ,
- Other: all other tasks,

Significantly more time was spent scanning in NNEA than in AXBAM. This is expected since there were more scans performed. This is a result of the fact that most NBV positions are neighbouring positions to the current position and that there is more overlap in sensor FOVs, meaning less new information is collected with each scan. It also follows from the fact that more information is collected generally by NNEA than by AXBAM.

In Section IV-B, it was posited that NNEA would result in fewer  $\mathcal{H}(q)$  estimation operations needing to be performed.

The decrease in the number of operations can be seen in Fig. 14b. The time taken to evaluate information values is reduced in NNEA compared to AXBAM in all trials where  $|Q_a| \approx 2000$  or greater. This decrease ranges from 36.02% to 69.96%. The time spent validating configurations follows a similar pattern for the same reasons, as seen in Fig. 14c.

In all trials, path planning time was lower for NNEA than for AXBAM. This is a result of the fact that no path planning is required between neighbouring poses (see Section II-B).

## V. DISCUSSION

### A. Quality of Environments Used

The field environment provided by the industry partner was representative of the types of environments expected for the industrial application that is a target of this research. Larger environments are most often repeated variations of the sites shown in this paper. From the standpoint of robot exploration however, the environment was relatively simple. There is no theoretical reason that NNEA would not perform as well in more complex environments, as the approach is based on robot configuration space and not on properties of the environment, but this remains untested due to the availability of the test sites. It remains as part of the future work.

The environments consisted of many feature-less grey surfaces in poorly illuminated conditions. In cases where there were features such as rivets, these occurred in regular repeating patterns. This regularity makes it difficult to correctly match features seen from two different viewpoints. As a result, the effectiveness of sensor fusion algorithms such as ICP [33] would have been detrimentally affected; since ICP requires the identification of unique features seen across multiple sensor observations. We have learned from these experiments that existing ICP-based approaches cannot be directly applied to the scenarios described in this paper. Instead, tailored approaches to ICP will need to be investigated that can operate in such challenging environments, particularly using the robot's odometry to develop a prior for the sensor location. However this avenue was not explored in the scope of this paper.

### B. Estimation of Possible Information Gain

Field sites 1 and 2 were relatively enclosed and this made it possible to determine a reasonable estimate of the information

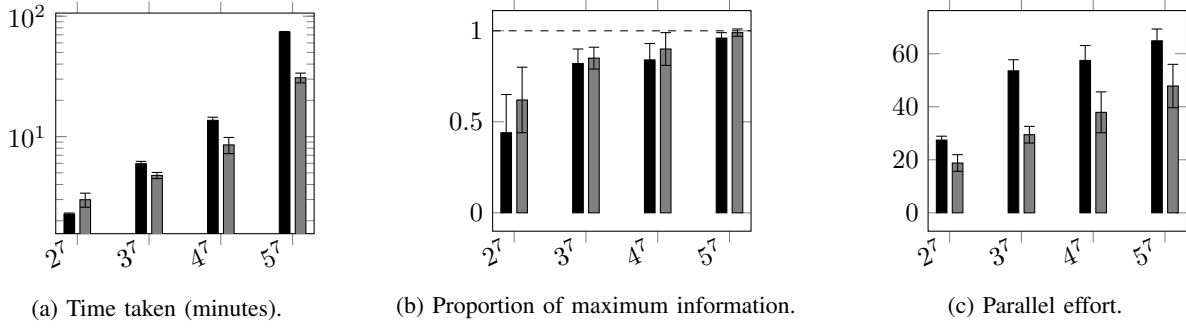


Fig. 13: Average metrics from each environment, at  $|Q_a| = 2^7$ ,  $|Q_a| = 3^7$ ,  $|Q_a| = 4^7$  and  $|Q_a| = 5^7$ . AXBAM is shown in black, and NNEA in grey.

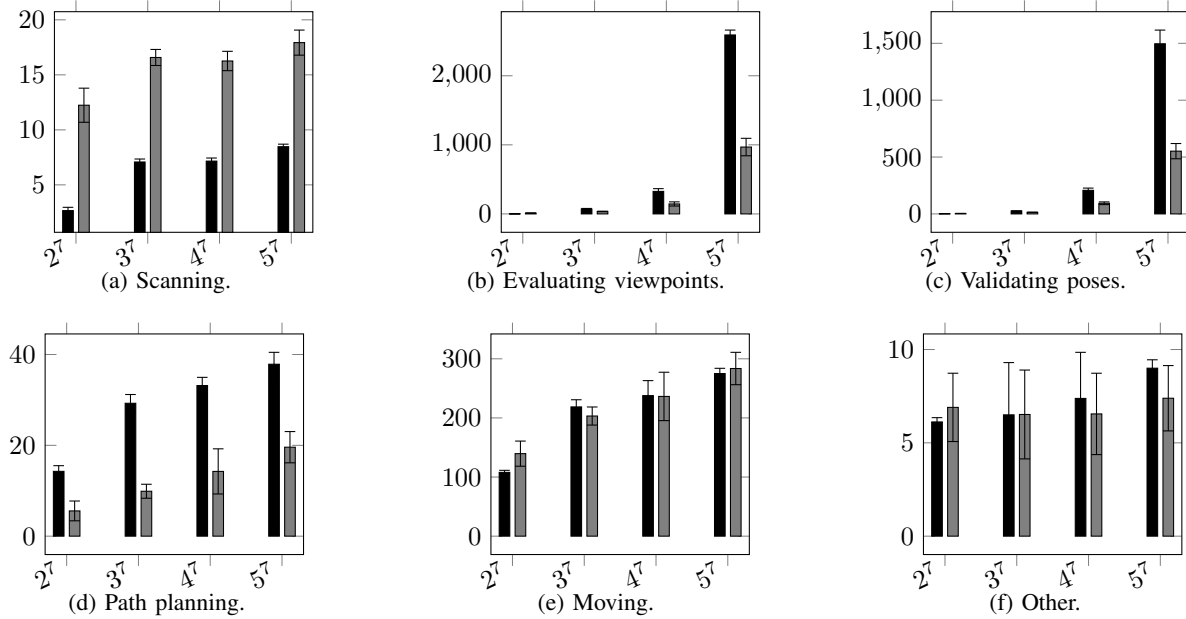


Fig. 14: Averaged time in seconds for each task in each environment, at  $|Q_a| = 2^7$ ,  $|Q_a| = 3^7$ ,  $|Q_a| = 4^7$  and  $|Q_a| = 5^7$ . Shown in black are times for AXBAM and in grey for NNEA.

that could be collected by the exploration algorithms using simple geometry. The robot was approximated as being able to cover all space within a radius of robot length plus sensor range. Considering the tunnels in profile along their length, and knowing several measurements of the environment, the covered area was calculated for the 2D case, removing occluded regions. Figures 15 shows the areas considered discoverable for both field sites.

The areas were multiplied by the tunnel width to determine the volume. The volume of the space known to the robot at the beginning of exploration was subtracted. The resulting volume was then divided by the volume of a single voxel, giving the final estimate of the total number of voxels that should be discoverable by the exploration algorithms. Table I shows the number of voxels that are estimated as being discoverable, the voxels collected in each trial, and the percentages these represent of the estimate.

These values are not a perfect ground truth. They overestimate the capability of the sensor, since it has a large minimum range, and the capabilities of the robot, since not all

joint configurations were valid. In other respects the estimate values do not properly take into account the robot's ability to peer around occlusions; and are therefore not a maximum. Despite these qualifiers, the estimates give an indication of each exploration algorithm's ability to cover the environment.

One option for providing an improved measure of ground truth in these experiments might have been to use a computer model of the environment to simulate what could have been observed from each sensor position used during the experiments. The real environment would have had to be kept clear of any objects not part of the model (e.g. tools, chairs, etc.), perhaps by using backdrops to prevent the sensor from observing certain regions. The real positions of the sensor could have been recorded using a high-accuracy indoor tracking system. These resources were not available at the time of the experiments in this paper.



TABLE I: The percentages of the ground truth collected by each algorithm in each field site.

	Estimate	$ Q_a  \approx 100$		$ Q_a  \approx 2000$		$ Q_a  \approx 2000$	
		AXBAM	NN	AXBAM	NN	AXBAM	NN
Field Site 1	49272	79.89%	81.30%	78.43%	81.23%	85.69%	87.48%
Field Site 2	47037	47.82%	71.32%	97.11%	93.73%	92.60%	99.38%

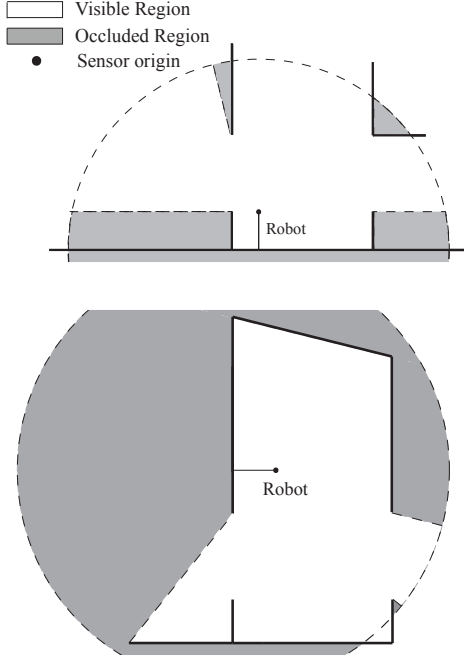


Fig. 15: Field site 1 (above) and 2 (below) seen from the side.

### C. Analysis of Maximal Effort as a Predictor of Time Spent in Motion

It was expected that the maximal effort metric would give a good estimation of the time it would take for the robot to move along the trajectories determined by NNEA and AXBAM. Though maximal effort was in all cases lower with NNEA, it was not the case that the robot always spent less time in motion for NNEA than AXBAM. To determine whether using maximal effort as an indicator of movement time was appropriate, the correlation between difference in maximal effort and difference in time spent moving was calculated (Fig. 16a). The Pearson correlation coefficient,  $r$  for the sample data is 0.70, denoting a strong correlation [34].

For similar maximal effort scores, NNEA results in higher movement times than AXBAM. This is possibly because NNEA motions are shorter and involve more frequent stops to take a scan, whereas AXBAM tends towards longer uninterrupted motions. Frequent stops mean the robot never reaches top movement speed and is instead constantly accelerating or decelerating. Fig. 16 lends support to this hypothesis. AXBAM displays a stronger correlation coefficient of 0.99 between maximal effort scores and time spent in motion, while NNEA's maximal effort scores vs. time spent in motion have a correlation coefficient of 0.95. The equations defining time spent in motion as a function of maximal effort can also be calculated for both AXBAM and NNEA:  $f_{axbam}(x) = 4.29 \times x - 8.72$

and  $f_{nn}(x) = 4.64 \times x + 60.29$ .

The use of maximal effort as a representation of expected time spent in motion is therefore supported, but the particular movement behaviour of specific exploration strategies must be taken into account.

## VI. CONCLUSIONS AND REMARKS

This paper has presented the results of trials in real environments of the Nearest Neighbours Exploration Approach (NNEA). The approach has been demonstrated on a 7DOF robot on a ferromagnetic bridge but can be generalised to a robot with fewer or more degrees of freedom, operating in different types of environments.

Where a large enough sampling  $Q_a$  was given, NNEA outperformed AXBAM in the majority of trials both in reduced exploration times and in amount of information collected. When smaller sampling sets  $Q_a$  were used, NNEA took more time to terminate, but collected more information.

NNEA achieved these improvements by reducing the number of viewpoints considered and evaluated at each iteration, and over the whole of exploration. Time was also saved through a reduction in required path planning.

In approximately half the trials, NNEA resulted in more time spent moving the robot, even though maximal effort had been significantly reduced. This is due to NNEA's more frequent stopping to perform scans of the environment. Due to this outcome, and the fact that more scans were performed in all NNEA than AXBAM, this approach would not be suitable for a robot system where gathering sensor data is a significantly time consuming process.

Both algorithms resulted in maps with inaccuracies due to errors in the estimation of the current sensor position and orientation. Solving the error in pose estimation might be done with an approach such as GraphSLAM [35]. Alternatively, fusing sensor data taken from different viewpoints, despite the relative error in coordinate frames, might be improved with ICP [33], though ICP's performance will be detrimentally affected by the lack of unique features in the environments. Integrating an uncertainty management and a sensor fusion approach is part of ongoing and future work.

## ACKNOWLEDGEMENT

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

## REFERENCES

- [1] H. Zhang, J. Zhang, G. Zong, W. Wang, and R. Liu, "Sky cleaner 3: a real pneumatic climbing robot for glass-wall cleaning," *Robotics Automation Magazine, IEEE*, vol. 13, no. 1, pp. 32–41, 2006.

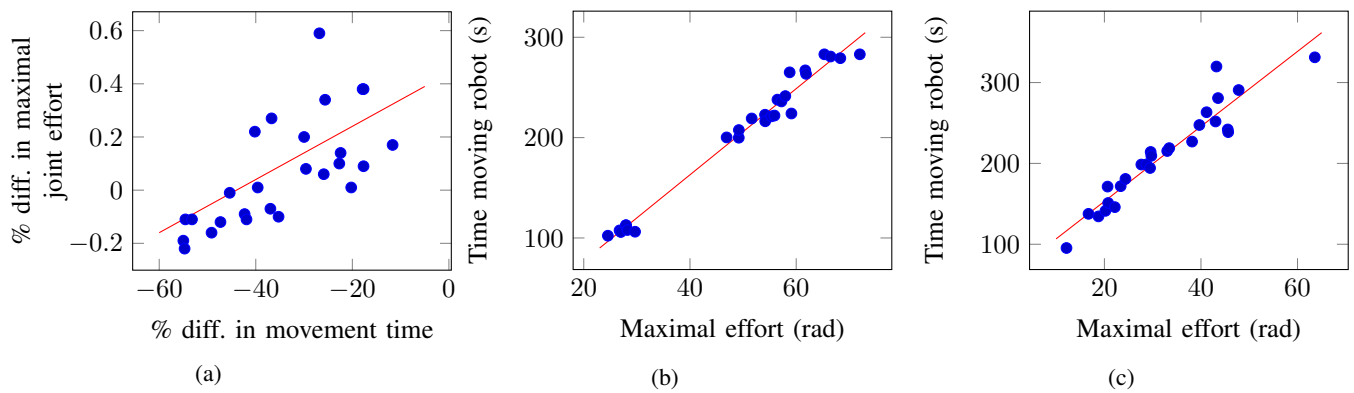


Fig. 16: (a) Difference in maximal effort vs. difference in movement time between AXBAM and NNEA. (b) and (c) Maximal effort vs. movement time for AXBAM and NNEA. Red lines shows the linear regression.

- [2] A. Fafña, D. Souto, A. Deibe, F. López-Peña, R. J. Duro, and X. Fernández, "Development of a climbing robot for grit blasting operations in shipyards," in *Robotics and Automation (ICRA), Proceedings of the 2009 IEEE International Conference on*, 2009, pp. 2560–2565.
- [3] K. Nagatani, S. Kiribayashi, Y. Okada, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, and Y. Hada, "Redesign of rescue mobile robot quince," in *Safety, Security, and Rescue Robotics (SSRR), IEEE International Symposium on*, 2011, pp. 13–18.
- [4] D. Schmidt and K. Berns, "Climbing robots for maintenance and inspections of vertical structures a survey of design aspects and technologies," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1288 – 1305, 2013.
- [5] C. Hare, *Protective coatings for bridge steel*, ser. National Cooperative Highway Research Program Reports. Transportation Research Board, National Research Council, 1987.
- [6] P. Romey, A. Nhan, K. Williams, and M. Dunn, "Sydney harbour bridge conservation management plan 2007," Roads and Traffic Authority, Tech. Rep., 2007.
- [7] B. M. Phares, D. D. Rolander, B. A. Graybeal, and G. A. Washer, "Reliability of bridge inspection," *Public Roads*, vol. 64(5), pp. 22–29, 2001.
- [8] A. To, G. Paul, and D. Liu, "Surface-type classification using rgb-d," *Automation Science and Engineering, IEEE Transactions on*, vol. 11, no. 2, pp. 359–366, April 2014.
- [9] E. Guizzo. (2011) Fukushima robot operator writes tell-all blog. In: IEEE Spectrum. [Online]. Available: <http://spectrum.ieee.org/automaton/robotics/industrial-robots/fukushima-robot-operator-diaries>
- [10] H. M. La, N. Gucunski, S.-H. Kee, J. Yi, T. Senlet, and L. Nguyen, "Autonomous robotic system for bridge deck data collection and analysis," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 1950–1955.
- [11] P.-C. Tung, Y.-R. Hwang, and M.-C. Wu, "The development of a mobile manipulator imaging system for bridge crack inspection," *Automation in construction*, vol. 11, no. 6, pp. 717–729, 2002.
- [12] K. H. Cho, H. M. Kim, Y. H. Jin, F. Liu, H. Moon, J. C. Koo, and H. R. Choi, "Inspection robot for hanger cable of suspension bridge: Mechanism design and analysis," *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 6, pp. 1665–1674, 2013.
- [13] D. Zhu, J. Guo, C. Cho, Y. Wang, and K.-M. Lee, "Wireless mobile sensor network for the system identification of a space frame bridge," *Mechatronics, IEEE/ASME Transactions on*, vol. 17, no. 3, pp. 499–507, 2012.
- [14] N. Metni and T. Hamel, "A {UAV} for bridge inspection: Visual servoing control law with orientation limits," *Automation in Construction*, vol. 17, no. 1, pp. 3 – 10, 2007.
- [15] A. Mazumdar and H. H. Asada, "Mag-foot: A steel bridge inspection robot," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 1691–1696.
- [16] D. Pagano, D. Liu, and K. Waldron, "A method for optimal design of an inchworm climbing robot," in *Robotics and Biomimetics (ROBIO), IEEE International Conference on*, 2012, pp. 1293–1298.
- [17] P. Ward, D. Liu, K. Waldron, and M. Hassan, "Optimal design of a magnetic adhesion system for climbing robots," in *Climbing and Walking Robots (CLAWAR), International Conference on*, 2013.
- [18] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Robotics and Automation (ICRA), Proc. Symp. IEEE Int Computational Intelligence in*, 1997, pp. 146 – 151.
- [19] E. P. e Silva, P. M. Engel, M. Trevisan, and M. A. P. Idiart, "Exploration method using harmonic functions," *Robotics and Autonomous Systems*, vol. 40, no. 1, pp. 25 – 42, 2002.
- [20] R. Shade and P. Newman, "Choosing where to go: Complete 3D exploration with stereo," in *Robotics and Automation (ICRA), Proc. IEEE International Conference on*, 2011, pp. 2806–2811.
- [21] L. Freda, G. Oriolo, and F. Vecchioli, "Sensor-based exploration for general robotic systems," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 2157–2164.
- [22] C. Dornhege and A. Kleiner, "A frontier-void-based approach for autonomous exploration in 3D," in *Safety, Security, and Rescue Robotics (SSRR), IEEE International Symposium on*, 2011, pp. 351–356.
- [23] C. Potthast and G. S. Sukhatme, "Next best view estimation with eye in hand camera," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS): The PR2 Workshop*, 2011.
- [24] M. Morales, R. Pearce, and N. Amato, "Analysis of the evolution of c-space models built through incremental exploration," in *Robotics and Automation (ICRA), IEEE International Conference on*, April 2007, pp. 1029–1034.
- [25] Y. Yu and K. K. Gupta, "C-space entropy: A measure for view planning and exploration for general robot-sensor systems in unknown environments," *I. J. Robotic Res.*, vol. 23, no. 12, pp. 1197–1223, 2004.
- [26] P. Quin, G. Paul, A. Alempijevic, D. K. Liu, and G. Dissanayake, "Efficient neighbourhood-based information gain approach for exploration of complex 3d environments," in *Robotics and Automation (ICRA), Proc. IEEE International Conference on*, 2013, pp. 1335–1340.
- [27] P. Quin, G. Paul, D. Liu, and A. Alempijevic, "Nearest neighbour exploration with backtracking for robotic exploration of complex 3d environments," in *Australasian Conference on Robotics and Automation, Proceedings of*, 2013.
- [28] 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.
- [29] G. Paul, D. Liu, N. Kirchner, and G. Dissanayake, "An effective exploration approach to simultaneous mapping and surface material type identification of complex three-dimensional environments," *J. Field Robot.*, vol. 26, pp. 915–933, 2009.
- [30] P. Chotiprayanakul, D. K. Liu, D. Wang, and G. Dissanayake, "A 3-dimensional force field method for robot collision avoidance in complex environments," in *Proceedings of the 24th International Symposium on Automation and Robotics in Construction (ISARC)*, 2007.
- [31] J. Amanatides and A. Woo, "A fast voxel traversal algorithm for ray tracing," in *Eurographics*, 1987, pp. 3–10.
- [32] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems," in *Proc. of the ICRA 2010 workshop*, 2010.
- [33] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Computer graphics forum*, vol. 32, no. 5. Wiley Online Library, 2013, pp. 113–123.
- [34] C. Dancy and J. Reidy, "Statistics without maths for psychology," Harlow: Pearson Education Limited, 2004.

- [35] S. Thrun and M. Montemerlo, “The graph slam algorithm with applications to large-scale mapping of urban structures,” *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

**Phillip Quin** graduated with a B.Sc. in Computer Science, Majoring in Robotics, at the University of New South Wales (UNSW). He is currently completing his PhD in Engineering (Robotics) at UTS.



**Dikai Liu** is a Professor at the University of Technology, Sydney (UTS). His main research interests are field robotics and physical human-robot interaction.



**Gavin Paul** graduated with a B.Eng. in Computer Systems Engineering at the University of Technology, Sydney (UTS). In 2010 he received his PhD in Engineering (Robotics) at UTS.