Prior-knowledge Assisted Fast 3D Map Building of Structured Environments for Steel Bridge Maintenance

Stephan Sehestedt[1], Gavin Paul[2], David Rushton-Smith[3] and Dikai Liu[4]

Abstract— **Practical application of a robot in a structured, yet unknown environment, such as in bridge maintenance, requires the robot to quickly generate an accurate map of the surfaces in the environment. A consistent and complete map is fundamental to achieving reliable and robust operation. In a real-world and field application, sensor noise and insufficient exploration oftentimes result in an incomplete map. This paper presents a robust environment mapping approach using prior knowledge in combination with a single depth camera mounted on the end-effector of a robotic manipulator. The approach has been successfully implemented in an industrial setting for the purpose of steel bridge maintenance. A prototype robot, which includes the presented map building approach in its software package, has recently been delivered to industry.**

I. INTRODUCTION

Advances in robotics and sensor technology increasingly enable complicated tasks to be automated. In order to perform many robotic tasks in complex three dimensional (3D) environments, a geometrically accurate and reliable map is required - particularly when a robot must plan online the precise motions for a given workspace.

In areas of automation such as car manufacturing, tasks are performed in a fully known, carefully structured and controlled environment. However, in a less controlled environment the environment's basic geometry may still be predicted, even where the exact location of individual elements is unknown. Noisy and incomplete sensor data may mean that even after exploring an environment the geometry cannot be determined with sufficiently high accuracy. In the case where partial or noisy sensor data is gathered by a robot, precise templates of individual elements could still be located in the environment based upon this sensor data - thus leading to a geometrically accurate map of the environment.

One task requiring an accurate map in a field environment is when using an assistive robot to grit-blast steel bridges [1]. Historically, the vital task of grit-blasting to remove rust and old paint for steel bridge maintenance has posed a high risk of injury to workers. Grit-blasting requires a worker to handle a heavy hose and significant reaction forces due to grit exiting the nozzle at high velocity. In this physically strenuous job, accidents can cause severe injury. Moreover, on many bridges the paint contains lead and thus the dust produced by grit-blasting poses a serious health and environmental risk. Nevertheless, paint and rust removal is a vital process when maintaining the steel bridges that are an essential part of modern infrastructure. Assistive robotics is being introduced as a way to reduce the risk to workers.

The Grit-blasting Assistive Device (GAD) is designed to autonomously perform grit-blasting on an industry partner's bridges with minimal operator remote interaction. The project's goal is to minimize the need for humans to perform grit-blasting and instead utilize a 6 Degree Of Freedom (DOF) robotic manipulator. Employees will only be required to remotely supervise the robot's operation and to blast the surfaces that are outside the manipulator's work envelope.

Although a steel bridge environment is often structurally repetitious due to being built using defined elements (e.g. I-beams, trusses, girders), in practice the exact location of a robot cannot be known in advance. This is due to the temporary nature of the maintenance site installation, where scaffolding, containments and the robotic system are setup manually. Moreover, variations can exist in structural members that are cut to size. With these challenges in mind, there are four goals for a mapping system in this practical application. **(a) Efficiency:** complete mapping efficiently so that the system's productivity is not negatively impacted (e.g. less than 1 minute). **(b) Safe exploration:** although the 6 DOF manipulator has no suitable sensors for collision avoidance, exploration must be completed safely and quickly (e.g. within 2 minutes). Hence only a limited number of safe viewpoints can be considered and the environment cannot be explored completely. **(c) Reliability:** the workers operating the robot cannot debug a complex robotic system, so the map generated must be correct, even when there is spurious sensor data. **(d) Geometrical accuracy:** representing the geometry of a complex 3D environment accurately is fundamental to achieving high-quality grit-blast results in corners, along edges, and on curved surfaces and rivet heads.

There is significant research interest in automating evermore complex tasks in industrial applications [2], [3]. One challenge that has received attention in the literature is a robot's inability to perceive the environment for tasks where the tasks' parameters cannot be defined statically, such as in automated drilling for manufacture [4] and robotic welding in the field [5]. In order for a robot to perform tasks which demand adaptability, the robot must possess the fundamental ability to effectively sense the workspace and identify the elements of interest. Grit-blasting on steel bridges, requires that a workspace is observed to identify the elements to blast [1], and the areas to avoid during autonomous blast planning [6], [7]. Thus, a geometric map of surfaces in the environment needs to be generated.

[1]S. Sehestedt [2]G. Paul, [3]D. Rushton-Smith and [4]D.K. Liu are with Faculty of Engineering and IT, University of Technology, Sydney [1]stephan.sehestedt at gmail.com,[3]david at rushton-smith.com,{[2]gavin.paul, [4]dikai.liu} at uts.edu.au

Eye-in-hand manipulator setups have been used to observe the object or environment of interest [6]. The time required for an eye-in-hand system to generate an accurate map has historically been a challenge [11], coupled with the need to ensure the map is complete [12]. Currently, selected objects can be mapped from only one viewing direction by relying upon a high-resolution point cloud [13], [14] and making the assumption that shape primitives like cylinders are always sufficient to model an object. Methods exist that fit geometric models to point cloud data such as [15] and [16]. In [17] and later [18] the Random Sample Consensus algorithm was proposed to fit point cloud data to a model. The main shortcoming of such methods is their restriction to basic geometric shapes, which is not generally sufficient in a complex automation scenario where multifaceted objects must be dealt with. These algorithms, however, remain powerful tools for the segmentation of point clouds.

Geometric maps that are generated by a robot will generally contain uncertainty due to noisy and incomplete observations, unpredictable positioning of the robot and a potentially cluttered workspace. Mapping under uncertainty has been a focus of the robotics community for many years. Many approaches build maps during mobile robot navigation, such as to reconstruct the 3D geometry from a sequence of segmented 2.5D range images [19]. In particular, Simultaneous Localisation And Mapping (SLAM) has received attention for both small and large scale map building [8]. In SLAM, the consistency of a resulting map is a mandatory requirement [9], [10], but the geometric accuracy and completeness on the small local scale are often only desirable requirements. Object mapping and reconstruction algorithms can achieve good geometrical accuracy at the expense of necessitating many observations of an object. This is problematic as the number of safe viewpoints may be limited and time constraints may be enforced in a practical automation environment.

Prior knowledge about structural elements in a workspace can conceivably be exploited, so that even when there are variations in the environment (e.g. scaffolding, workspace configurations, and incomplete sensor data), a robot can still identify the structural elements relative to its location. We propose a mapping approach which uses an eye-in-hand sensor configuration and prior knowledge encoded in a set of templates to achieve geometrically accurate mapping of a robot's workspace in a complex steel bridge maintenance environment to achieve an accurate geometric map. Furthermore, the time required for the exploration and mapping processes is kept short by restricting the number of viewpoints and utilising templates to fill in missing information.

The remainder of this paper is organized as follows. Section II describes the proposed approach to 3D template mapping, then formulates the approach as a finite state machine. Section III presents experimental results using data collected both in our laboratory and from a steel bridge maintenance site. Finally, Section IV provides conclusions.

II. MAPPING

In this section, a mapping approach is presented to satisfy the requirements for the automation of tasks where geometrical accuracy and usability are of paramount importance while ensuring a system's productivity is not significantly impacted. It is assumed that exploring the entire workspace is not always feasible because of limitations in both the safe workspace and the time available. The use of prior knowledge about the work environment is often possible as digital models may be available or are easy to produce. Such information could be used to achieve a highly accurate geometric representation of particular elements of the environment in the presence of noisy sensor readings. Furthermore, if the prior knowledge can be successfully associated with the sensor data, unobserved regions could be filled in. The usability requirement refers to having no operator intervention during the mapping process, since operators are expected to have limited robotics experience. Usability is achieved by the implementation of the mapping procedure as a finite state machine with fault detection and recovery. The mapping research challenges exist in real automation environments, such as the GAD, for steel bridge maintenance using a robot and digital models of particular structural elements.

A. Exploration

The GAD, shown in Fig. 1, consists of a 6 DOF manipulator and an Asus Xtion sensor at its end-effector. The robot is moved into its workspace on rails and begins its operation by exploring its surroundings with the sensor. So as to avoid collisions, the manipulator is restricted to a defined safe area during environment sensing.

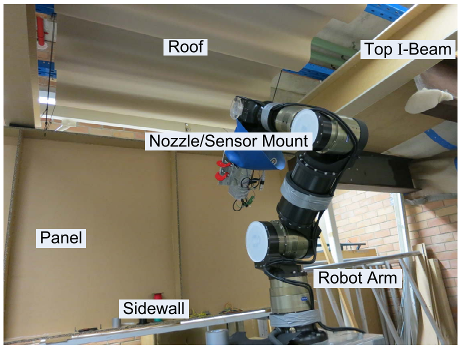


Fig. 1. The GAD robot in a test environment.

For the considered environment, a set of 8 viewpoints has been chosen between which safe trajectories are available. Given the manipulator's joint configuration, the instantaneous location and orientation of the sensor can be obtained with reference to a 3D coordinate system, located at the base of the robot manipulator. Given a manipulator pose, $\vec{Q} = [q_1, q_2, \dots q_6]^T$ that describes the angular position of each manipulator joint, and using forward kinematics, it is possible to compute the position and orientation of the manipulator's end-effector where the sensor is mounted.

A position and orientation combination of an end-effector can be expressed in the 4×4 rotation and translation homogeneous transformation matrix, ${}^{0}T_f(\vec{Q})$, by performing transformations based upon the manipulator model and the manipulator joint angles q_i for $i \in \{1, \dots 6\}$ as,

$$ {}^{0}T_f(\vec{Q}) = \prod_{i=1}^{6} {}^{i-1}T_i(q_i). \qquad (1)$$

The transformation matrix between the end-effector and the sensor is denoted as ${}^{f}T_s$. Together these two matrices describe the viewpoint in the manipulator base coordinate frame, ${}^{0}T_s(\vec{Q}) = {}^{0}T_f(\vec{Q}){}^{f}T_s$. Since the position and orientation of the sensor is known for each viewpoint, the resulting point clouds can be fused together using forward kinematics to acquire a map.

Consider the environment in Fig. 1 which consists of the structural elements: panel, top I-beam, beam and roof. Exploring the workspace produces a 3D point cloud,

$$V_{workspace} = V_p \cup V_r \cup V_I \cup V_o \qquad (2)$$

containing 3D points belonging to the known structural elements panel (V_p), roof (V_r), top I-beam (V_I) and outliers (V_o) which belong to structural and non-structural elements that are presently disregarded. Fig. 3 shows the resulting point cloud, where it can be seen that there are a large number of outliers as well as a significant number of unobserved regions.

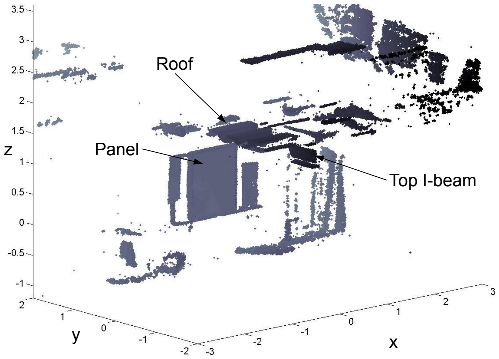


Fig. 2. A scan of a mock-up structure. Color encodes the 3D point's Y-value.

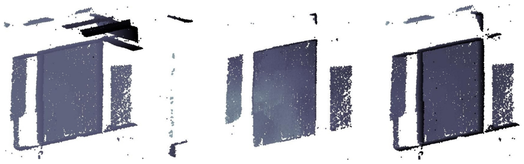
B. Template Mapping

Given $V_{workspace}$ is sufficient to identify the elements of interest, and a rough knowledge of the robot's position in the workspace, it is possible to segment the data in a process of successively narrowed bounding boxes, which define relevant Regions Of Interest (ROI). If this can be done successfully, templates can be matched to the data using a variant of the Iterative Closest Point (ICP) algorithm [20]. An intuitive approach is to identify a part of the workspace that is distinct and can be identified with relative ease. In the given example this could be the sidewall's panel as it is large and distinct from other structural elements. Henceforth, the basic procedure of template mapping will be presented using the sidewall's panel as an example.

Extract points inside a coarse ROI: Since ICP variants are vulnerable to outliers because of convergence to local minimums [21], the point cloud is initially segmented in order to remove outliers. One segmentation approach is to identify a basic element which is easily separated in the data as defined in Equation 2. For the considered environment the sidewall satisfies this criterion as sidewalls are large and spatially separated to a sufficient degree from other structural elements. Hence, a coarse ROI can be defined to acquire a point cloud which mostly consists of the points,

$$V_{sw} = V_p \cup V_o^{*}, \text{with } V_o^{*} \subseteq V_o \qquad (3)$$

that belong to a sidewall. This ROI will include the panel we want to locate as shown in Fig. 3a.



(a) (b) (c)

Fig. 3. (a) The points inside a coarse ROI. (b) Points extracted using RANSAC to find the sidewall. This set of points does not include the flanges and webs of the panel. (c) Using the results from the previous step, the ROI could be narrowed significantly.

Determine a narrower ROI: since V_{sw} may still contain a significant quantity of points not belonging to the sidewall. The popular Random Sampling Consensus (RANSAC) algorithm [17] is used to remove those outliers since RANSAC robustly fits a model to given data even when there are a significant numbers of outliers.

In the considered scenario, the result is a set of points belonging to the largest flat surface of the sidewall, V_{sw} as shown in Fig. 3c. This result can be used to define a very narrow ROI to include the flanges of the panel as illustrated in Fig. 3b, resulting in

$$V_{sw} = V_p \cup V_o^{**}, \text{with } V_o^{**} \subseteq V_o^{*}. \qquad (4)$$

Template matching using ICP: The final step of the mapping approach is to find the correct homogeneous transformation matrix to locate the panel's template in the workspace. At this stage the problem is reduced to registering a point cloud with several outliers to a given template.

The well known ICP approach consists of the two steps of finding nearest neighbor point pairs and minimizing a given error metric:

1) Compute nearest neighbor pairs of points in the scan data V_{sw} and template data M_{sw}. That is, find the nearest point $m_{sw}^{(j)} \subseteq M_{sw}$ to $v_{sw}^{(i)} \subseteq V_{sw}$ for all j:

$$d(v_{sw}^{(i)}, M_{sw}) = \min_{1 \leq j \leq K} d(v_{sw}^{(i)}, m_{sw}^{(j)}) \qquad (5)$$

with K being the number of points in M_{sw}. The number of points in V_{sw} and M_{sw} is commonly not equal and some regions do not overlap. Hence, the resulting number of points N in the set of point pairs may be of different size to V_{sw} and M_{sw}.

2) Compute transform T which minimizes a chosen distance metric between all point pairs.

Here, two variants are considered for step 2. In the first one, nonlinear least squares optimization is used to calculate T so that the error,

$$\varepsilon_{point} = \frac{1}{N} \sum_{i=1}^{N} \| T \cdot v_{sw}^{(i)} - m_{sw}^{(i)} \| \tag{6}$$

is minimized. Henceforth, this implementation will be referred to as ICP_{point}.

The second variant implements point to plane matching [22], where the distance between a scan point and the template surface is minimized along the surface normal. The error function is

$$\varepsilon_{plane} = \frac{1}{N} \sum_{i=1}^{N} \| \eta^{(i)} \cdot (T \cdot v_{sw}^{(i)} - m_{sw}^{(i)}) \| \tag{7}$$

where $\eta^{(i)}$ denotes the surface normal. This approach has gained popularity for its robustness and accuracy [23]. Henceforth, this variant will be referred to as ICP_{plane}. It is possible to supply a transform T_0 as an initial guess for ICP. The ICP algorithm is summarized in Algorithm 1.

```
Data: Scan points $V$, Template $M$, initial transform $T_0$
Result: Transform $T$, Root Mean Square Error (RMS)
if $T_0$ was supplied then
    Apply $T_0$ to $M$;
end
while $\varepsilon$ shrinks do
    Search nearest neighbor point pairs as Equation 5;
    Compute $T$ to minimize $\varepsilon$ as in Equation 6 or 7;
    Apply $T$ to $M$;
end
```

Algorithm 1: Iterative Closest Point Algorithm

C. The Mapping Approach State Machine

The proposed mapping approach is implemented in a finite state machine, as illustrated in Fig. 4. Once the data from the exploration is available, mapping starts by parsing a given set of configurations. For each structural element that is to be located, one configuration is supplied. The configuration parameters are: Template ID; Initial transform (optional); ROI in scan data (one or more); Dependence of the ROI on a previous template ID; RANSAC model (one or more); RANSAC tolerance; Compute narrower ROI (true/false).

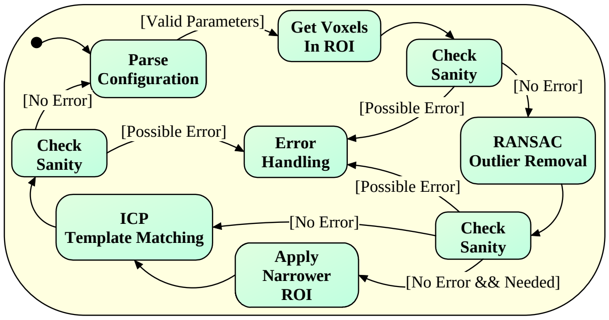


Fig. 4. The mapping approach is modeled as a state machine.

The template ID determines which template is to be used for the ICP step. An initial transform can also optionally be supplied. One or more ROIs can be supplied, where each given ROI will be used in order to target the template's location in the sensor data. Furthermore, it is possible to let RANSAC look for one or more models in the scan data (e.g. the panel consists of flanges and webs which could be detected as individual planes using RANSAC several times). Additionally, the output of RANSAC could be used to narrow the ROI before proceeding with ICP.

Currently, the top I-beam may be in one or both of two locations in the workspace. Therefore, several ROIs can be given for each parameter set and the state machine is facilitated to execute mapping accordingly. The sanity checks in the state machine use historical data (i.e. expected number of points) to allow for the Error Handling state. This can either rerun a previous state to try to rectify the error, or cause the exploration and mapping process to restart.

III. EXPERIMENTS

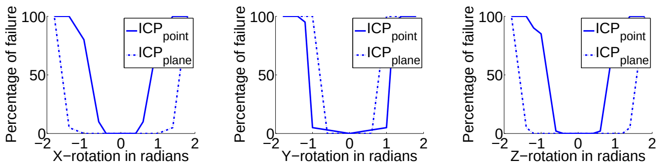
All presented experiments were conducted using the GAD robot shown in Fig. 1. The robot is designed to perform the complex task of grit-blasting on steel bridges. In this application the robot must first explore its environment to build a map. This map is used to plan the blasting motion which is then executed with the sensor turned off. It is necessary to turn off and protect the sensor as environment sensing is currently not feasible during grit-blasting due to the dense dust in the air.

Experiments were conducted in a laboratory mock-up structure and on a real-world steel bridge maintenance site. The mock-up structure closely mimics the bridge maintenance site in order to simulate realistic conditions. The mapping performance is considered separately for the sidewall, roof and top I-beam as there are different challenges involved. First, the sidewall needs to be mapped using a ROI that is as wide as possible because this element is used to determine the locations of the other elements. Then the roof and top I-beam are extracted from the scan data. However, due to the limited field of view and limited time available for exploration, those elements are scanned incompletely and hence mapping must be equipped to process incomplete data sets.

A. Laboratory Experiments

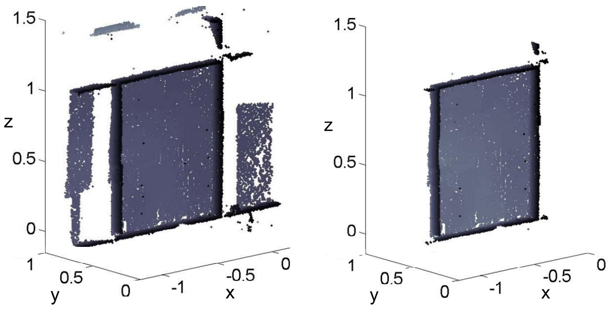
1) Convergence Characteristics for the Panel: Finding the correct location of the sidewall's panel is most critical as the other structural elements' ROIs will be determined on that basis. Hence, the first experiment aims to determine if the procedure can reliably detect the panel. A threshold for the RMS of 0.02 was experimentally determined to identify successful matches. If the RMS is above this threshold, this can results in a rotational error which could impact the mapping performance for the roof and top I-beam as well the blasting results.

To determine the region of convergence, scan data was rotated around the X, Y and Z axes separately. This was repeated in steps until the mapping approach was found to fail in 100% of cases. 20 recorded data sets were used to repeat the procedure 100 times for each rotation. The initial set of points from one data set is shown in Fig. 6a. The results in Fig. 5 show that both ICP_{point} (straight line) and ICP_{plane} (dotted line) have a wide enough region of convergence for the application.



(a) (b) (c)

Fig. 5. The region of convergence for ICP_{point} and ICP_{plane}. The scan data is rotated around the (a) X-axis, (b) Y-axis and (c) Z-axis.



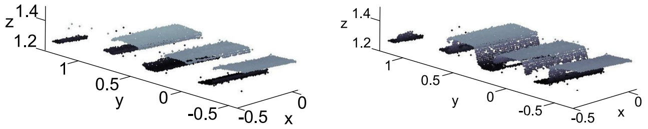
(a) (b)

Fig. 6. Scan data considered in the ICP step using (a) ROI_1; (b) ROI_2.

Another point of consideration is the size of the ROI. A larger ROI results in fewer restrictions on the physical placement of the robot, and thus a simpler setup procedure. Two ROIs were experimentally determined: the wider ROI_1 shown in Fig. 6a and the narrower ROI_2 shown in Fig. 6b.

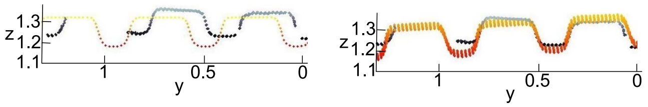
Table I shows in the first two columns the results using both ROIs and both ICP variants. The mapping was run on 30 data sets to collect the following data: the average RMS and standard deviation, σ_{RMS}, as well as the average time taken for the mapping process and the standard deviation, σ_{time}. Although both ICP variants produce good results when using the narrow ROI_2, ICP_{point} is not able to produce results with an RMS < 0.02 using ROI_1. Thus, mapping the panel using ICP_{plane} is the only choice.

2) Mapping the Roof and Top I-beam: The main issue for the roof and top I-beam is the incomplete scan data due to the limitations on the exploration process. Based on the panel's known location, a narrower ROI can be determined such that discarding outliers is more effective. Using RANSAC, two planes can be found which relate to the upper and lower part of the roof's corrugation as shown in Fig. 7a. This permits an accurate ROI for the roof where almost all outliers can be discarded as shown in Fig. 7b. Nevertheless, in the same figure it can be seen that there are significant amounts of data missing which makes template matching more difficult.



(a) (b)

Fig. 7. (a) The upper and lower plane of the corrugated roof can be found using RANSAC. (b) The points designated as roof after the ROI was narrowed down.



(a) (b)

Fig. 8. (a) The initial alignment of roof scan and template data. (b) Typically, ICP_{plane} converges with a small rotational error.

Again, ICP_{point} and ICP_{plane} are compared in terms of their matching performance for the roof. Fig. 8a shows the initial alignment of the roof's scan data and the template data in one test run. The results in columns 3 and 4 of Table I show that ICP_{plane} experiences difficulty with the available sensor data. A typical result can be seen in Fig. 8b, where a visible rotational error exists in the final transform. In contrast, ICP_{point} delivers accurate and consistent (small σ_{RMS}) performance with sufficiently short computation times of $7.8s$ on average. Hence, for mapping the roof, ICP_{point} is preferable.

After the panel and roof have been extracted, the focus is turned to the points belonging to the top I-beam, which is a support member on top of the sidewall that supports the roof. Using the same procedure as before, a set of points is extracted which contains very few outliers as shown in Fig. 9a. In the figure it can also be seen that not all of the top I-beam was observed and hence the challenge is again to match a template to incomplete data. The results in Table I show that ICP_{point} does well in this situation whereas ICP_{plane} performs poorly such as for the Roof and Top I-beam. Conversely, ICP_{point} produces poor results when there are too many outlier points such as in the case of the Panel.

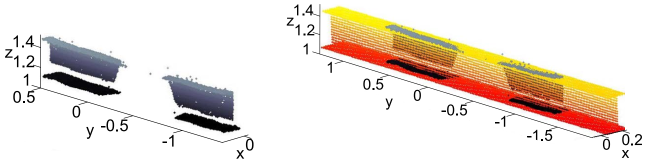
B. On Maintenance Site Experiments

The GAD system has been evaluated on a real bridge maintenance site shown in Fig. 10, where the robot is attached to a rail-mounted platform. This trial included an

	Panel ROI_1		Panel ROI_2		Roof		Top I-beam	
Method	ICP_{point}	ICP_{plane}	ICP_{point}	ICP_{plane}	ICP_{point}	ICP_{plane}	ICP_{point}	ICP_{plane}
Average RMS	0.1413	0.0119	0.0136	0.0125	0.0134	0.5966	0.0084	0.3134
σ_{RMS}	0.0041	0.0048	0.0094	0.0021	0.0006	0.5042	0.0002	0.2424
Average Time (s)	19.0788	7.8010	10.772	5.7521	7.8038	48.4875	0.4739	4.6350
σ_{Time}	0.6302	0.2433	0.3743	0.6321	0.6488	40.5043	4.5182	0.1575

TABLE I

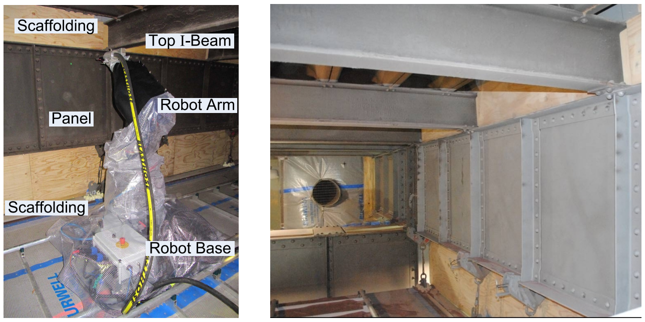
THE MAPPING PERFORMANCE FOR THE PANEL USING TWO DIFFERENT ROIS.



(a) (b)

Fig. 9. (a) The points extracted for the top I-Beam include few outliers. (b) Even though the scan data captures less than 50% of the top I-beam, it can still be successfully matched using ICP_{point}.

evaluation of the mapping sub-system during the intended robot operation as part of the normal bridge maintenance. There are many additional challenges when working in the field with this robotic maintenance system. In particular, time constraints must be adhered to so as not to impede the progress of the workers. The time available to setup the robot system is limited to several hours, and after grit-blasting commences additional setup work cannot be done easily. Low light levels and the uneven floor make precise positioning of the rails and platform difficult. Nevertheless, the setup work must be done only once and with no significant corrections after the installation is completed in order to keep to schedule.



(a) (b)

Fig. 10. (a) The GAD robot during onsite bridge maintenance system evaluations. (b) The workspaces after robotic grit-blasting (compare to the condition prior to blasting). The good coverage of the workspaces' elements is achieved due to the high geometrical accuracy of the maps.

	Panel	Roof	Top I-beam
Method	ICP_{plane}	ICP_{point}	ICP_{point}
Average RMS	0.0139	0.0116	0.0149
σ_{RMS}	0.0013	0.0021	0.0018

TABLE II

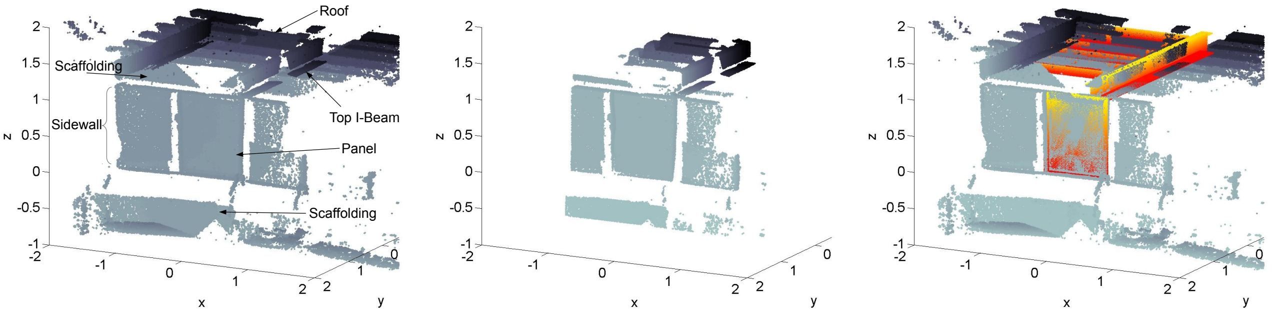
THE MAPPING PERFORMANCE FOR THE PANEL, ROOF AND TOP I-BEAM USING DATA FROM A REAL BRIDGE MAINTENANCE SITE.

During the trials a total of 16 data sets were recorded, all of which are presented here. Fig. 10 shows that additional scaffolding is installed as part of the containment above and below the sidewall. As shown in Fig. 11a this additional scaffolding is also apparent in the scan data. This adds a significant number of outliers, which are not discarded in the processing steps of the mapping approach. This can be seen in Fig. 11b, where the processed point clouds for the panel, roof and top I-beam used in the ICP step are shown together. Furthermore, the initial ROI was determined experimentally and was chosen to be wider compared to the laboratory experiments in order to maximize robustness.

Table II shows that the approach is effective for all data sets with similar RMS values compared to the laboratory trials, and a high consistency of the results is shown by the standard deviation σ_{RMS}. The processing time is typically below 30 seconds for an entire workspace which is orders of magnitude below the time it takes for grit-blasting so that the productivity will not be significantly impacted. Fig. 11c shows the mapping result for one workspace with the scan data being shown in grey and the template data shown in yellow/red.

IV. CONCLUSIONS

This work deals with the environment sensing and mapping by means of a robotic manipulator for steel bridge maintenance where it is necessary to automate tasks in a partially-known structured environment. Challenges exist due to the complexity of many 3D environments, the limited time available for exploration resulting in incomplete scan data, and the requirement of a geometrically accurate and reliable map. The presented approach focuses on mapping a fundamental element in the workspace of the robot and subsequently using that knowledge to identify other regions of interest. This is done by narrowing down ROIs until ICP-based template matching can be executed successfully. The use of templates leads to geometrically accurate maps, even in the presence of incomplete scan data. To demonstrate the viability of the approach for an industrial application, experiments were conducted using the GAD robot. Experiments using a mock-up structure as well as extensive experiments on an actual steel bridge maintenance installation have been conducted. The results show that the



(a) (b) (c)

Fig. 11. (a) The complete scan data for one workspace on a real steel bridge maintenance site. The panel of the current workspace is annotated. Scaffolding for the containment exists above and below the sidewall. (b) The data extracted for mapping. (c) The mapping result for one workspace.

proposed mapping approach consistently delivers high levels of accuracy with acceptable computation time. Furthermore, the approach is robust with respect to imprecise positioning making it suitable for real-world usage. This robustness stems from the segmentation process which initially makes few assumptions and then gradually narrows down ROIs. Future work will improve the efficiency of the approach and more closely integrate the approach with the workspace exploration process.

Acknowledgments

This work is supported by the Roads and Maritime Services (RMS) of NSW, and the University of Technology, Sydney (UTS). This work would not have been possible without the support of Gregory Peters, Andrew To and Gamini Dissanayake as well as the on-site staff of RMS.

References

[1] G. Paul, S. Webb, D. K. Liu, and G. Dissanayake, “A robotic system for steel bridge maintenance: Field testing,” in *Australasian Conference on Robotics and Automation*, 2010.

[2] Z. Gan, H. Zhang, and J. Wang, “Behavior-based intelligent robotic technologies in industrial applications,” in *Robotic Welding, Intelligence and Automation*, ser. Lecture Notes in Control and Information Sciences, T.-J. Tarn, S.-B. Chen, and C. Zhou, Eds. Springer Berlin / Heidelberg, 2007, vol. 362, pp. 1–12.

[3] K. Pathak, N. Vaskevicius, and A. Birk, “Uncertainty analysis for optimum plane extraction from noisy 3d range-sensor point-clouds,” *Intelligent Service Robotics*, vol. 3, pp. 37–48, 2010.

[4] Q. Zhan and X. Wang, “Hand-eye calibration and positioning for a robot drilling system,” *The International Journal of Advanced Manufacturing Technology*, pp. 1–11, 2011.

[5] K. Micallef, G. Fang, and M. Dinham, “Automatic seam detection and path planning in robotic welding,” in *Robotic Welding, Intelligence and Automation*, ser. Lecture Notes in Electrical Engineering, T.-J. Tarn, S.-B. Chen, and G. Fang, Eds. Springer Berlin Heidelberg, 2011, vol. 88, pp. 23–32.

[6] 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.

[7] G. Paul, N. Kirchner, D. K. Liu, and G. Dissanayake, “An effective exploration approach to simultaneous mapping and surface material-type identification of complex 3d environments,” *Journal of Field Robotics (S.I. 3D Mapping)*, vol. 26, no. 11-12 SI, pp. 915–933, 2009.

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

[9] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, “Consistency of the ekf-slam algorithm,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 3562–3568.

[10] T. Bailey, J. Nieto, and E. Nebot, “Consistency of the fastslam algorithm,” in *IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 424–429.

[11] M. Callieri, A. Fasano, G. Impoco, P. Cignoni, R. Scopigno, G. Parrini, and G. Biagini, “Roboscan: an automatic system for accurate and unattended 3d scanning,” in *Proc. 2nd Int. Symp. on 3D Data Processing, Visualization and Transmission*, 2004, pp. 805–812.

[12] M. Fernandez, K. Gupta, and J. Fraile, “Simultaneous path planning and exploration for manipulators with eye and skin sensors,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, 2003, pp. 914–919.

[13] R. Rusu, Z. Marton, N. Blodow, M. Dolha, and M. Beetz, “Functional object mapping of kitchen environments,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 3525–3532.

[14] Z.-C. Marton, L. Goron, R. Rusu, and M. Beetz, “Reconstruction and verification of 3d object models for grasping,” in *Robotics Research*, ser. Springer Tracts in Advanced Robotics. Springer Berlin, 2011, vol. 70, pp. 315–328.

[15] W. He, W. Ma, and H. Zha, “Automatic registration of range images based on correspondence of complete plane patches,” in *in Proc. 5th Int. Conf. 3-D Dig. Imag. Modeling*, 2005, pp. 470–475.

[16] T. Rabbani and F. van den Heuvel, “Efficient hough transform for automatic detection of cylinders in point clouds,” in *Conf. of Advanced School for Computing and Imaging*, 2005.

[17] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[18] R. Schnabel, R. Wahl, and R. Klein, “Efficient ransac for point-cloud shape detection,” *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[19] D. Fischer and P. Kohlhepp, “3d geometry reconstruction from multiple segmented surface descriptions using neuro-fuzzy similarity measures,” *Journal of Intelligent and Robotic Systems*, vol. 29, pp. 389–431, 2000.

[20] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[21] P. Foroughi, R. H. Taylor, and G. Fichtinger, “Automatic initialization for 3d bone registration,” in *Proceedings of the SPIE*, vol. 6918, 2008.

[22] Y. Chen and G. Medioni, “Object modeling by registration of multiple range images,” in *IEEE Int. Conf. on Robotics and Automation*, vol. 3, 1991, pp. 2724–2729.

[23] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy, “Geometrically stable sampling for the icp algorithm,” in *IEEE Int. Conf. on 3-D Digital Imaging and Modeling*, 2003, pp. 260–267.