A Novel Approach to Steel Rivet Detection in Poorly Illuminated Steel Structural Environments

Gavin Paul
Centre for Autonomous Systems
University of Technology, Sydney
E-mail: Gavin.Paul@uts.edu.au

LiYang Liu
Centre for Autonomous Systems
University of Technology, Sydney
E-mail: Liyang.Liu@student.uts.edu.au

Dikai Liu
Centre for Autonomous Systems
University of Technology, Sydney
E-mail: Dikai.Liu@uts.edu.au

***Abstract*—It is becoming increasingly achievable for steel bridge structures, which are normally both inaccessible and hazardous for humans, to be inspected and maintained by autonomous robots. Steel bridges have been traditionally constructed by securing plate members together with rivets. However, rivets present a challenge for robots both in terms of cleaning and surface traversal. This paper presents a novel approach to RGB-D image and point cloud analysis that enables rivets to be rapidly and robustly located using low cost, non-contact sensing devices that can be easily affixed to a robot. The approach performs classification based on: (a) high-intensity blobs in color images, (b) the non-linear perturbations in depth images, and (c) surface normal clusters in 3D point clouds. The predicted rivet locations from the three classifiers are combined using a probabilistic occupancy mapping technique. Experiments are conducted in several different lab and real-world steel bridge environments, where there is no external lighting infrastructure, and the sensors are attached to a mobile platform, i.e. a climbing inspection robot. The location of rivets within 2m of the robot can be robustly located within 10mm of their correct location. The state of voxels can be predicted with above 95% accuracy, in approximately 1 second per frame.**

I. Introduction

The inspection and maintenance of large scale infrastructure such as steel bridges is vital to ensure the integrity is preserved. However, often these tasks require humans to work in environments that are dangerous or difficult to access, such as due to the presence of traffic, working at heights in confined spaces, heavy manual handling, or nearby harmful contaminants, e.g. asbestos and lead-based paint. Thus, there is motivation to develop automated robotic tools [1], [2]. An autonomous robot that is tasked with inspecting and maintaining an unknown and complex 3D real-world environment must be capable of generating an accurate and reliable map of the surrounding surface geometry.

Recent advancements in sensing and robotic technology are enabling infrastructure health inspection tasks to be automated, even in complicated environments [1], [3]. It is advantageous for a robot, tasked with performing interactive operations on steel surfaces that contain rivets, to be able to accurately and robustly identify the rivets. An accurate 3D geometric surface map, coupled with rivet locations, can enable maintenance operations, such as grit blasting or painting, to be properly planned [4]. Building maps that incorporate rivet locations is also important for mobile inspection robots that perform surface traversals and transitions, especially for a bipedal climbing robot, where safe foot placement and stepping motions must be precisely planned [5].

Two example applications, shown in Fig. 1, that require an accurate, far-field rivet-detection solution include, a steel bridge maintenance robot capable of automatic grit-blasting, and a bio-inspired climbing robot that traverses and inspects surfaces covered in rivets [1], [4]. Each robot, is equipped with an RGB-D sensor, to collect color and depth measurements of the surrounding steel surfaces. In practice the density of surface data is insufficient to accurately match templates of specific rivet sizes. Instead rivet-shaped protrusions, divots, and heavy rust patches need to be identified since they require particular maintenance attention and should be avoided by a mobile robot that is traversing the surface.

Mapping and exploration approaches exist to generate 3D geometric maps of an environment that surround a robot. Most engineering approaches greedily select sensing viewpoints that are predicted to significantly reduce the uncertainty within the map [6]–[8]. Alternatively, approaches may adopt optimal exploration schemes based upon locating structural features about an environment [5], [8], so as to fit templates that encapsulate prior knowledge to the data [9].

In terms of determining the location of rivets, approaches can be considered to broadly fall into two categories: near-field non-destructive evaluation (NDE) and far-field techniques. In applications such as inspecting ageing aircraft various NDE techniques such as Magneto-optic Imaging (MOI) are used to detect the presence of cracks initiated at rivet holes [10]. Existing eddy current [11] and ultrasonics techniques, such as Guided Lamb Waves [12], [13] and acoustic structured wave propagation [12], require contact or close contact with surfaces. These NDE techniques can provide high resolution detail of the condition of surfaces in, and around rivets. However, they exhibit necessarily low efficiency and require close or actual contact with surfaces. In our target application internal material analysis is unnecessary and the location of rivets needs to be found more rapidly and at a distance, in order to avoid stepping on them, or to perform targeted surface maintenance tasks.

The other category of rivet and surface inspection is far-field (i.e. long range). Relevant methods in the literature are generally dependant on reliable, controllable illumination conditions, such as in a factory. One such method, using ”Edge of Light” technology and image processing techniques, has presented rivet detection results [14]. Significant literature describes the identification of features using RGB-D images, such as rust on infrastructure [15], people [16], or objects [17]. However, the existing approaches do not accurately and robustly detect rivet locations from data collected by a mobile robot in poorly lit, real-world environments.

The main contribution of this paper is three classifiers that utilise data from different modes of sensing, and a probabilistic fusion technique which improves the accuracy and robustness of rivet detection. The environment is not lit by off-board lighting infrastructure, and is naturally dim or dark. The RGB-D data collected can be noisy but the location of rivets that are within several meters of the robot's sensor are found so that nearby rivets, and areas determined to be free of rivets, can augment the geometric map that is used for planning purposes. This paper is organized as follows, Section II describes the three classifiers: color, depth and surface normal, then the methods for probabilistically fusing the outputs into a rivet-occupancy map containing both rivet locations and space which does not contain rivets. Section III presents experimental results using data collected both in a laboratory and on-site in a real steel bridge environment. Section IV discusses the limitations of the approach. Section V provides conclusions and future work.

II. Methodology

A. Robot and Sensors' Models

Consider an n Degree of Freedom (DoF) kinematic chain robot, positioned at a base location described by a homogeneous transform, ${}^0T_b$. Given a model and an n-dimensional vector of joint angles, $\mathbf{q} = [q_1, \dots q_n]^T$, the robots end-effector location, ${}^bT_f(\mathbf{q})$, can be computed using forward kinematics. Where depth and color cameras are rigidly mounted close together and on the end-effector, and the position relative to the end-effector is given by ${}^fT_c$, then the position and orientation of the sensor is,

$${}^0T_c(\mathbf{q}) = {}^0T_b{}^bT_f(\mathbf{q}){}^fT_c, \tag{1}$$

which describes both the camera's center position, $\mathbf{p_c}(\mathbf{q})$ and a projection line from the camera's center normal to the image plane, $\mathbf{n_c}(\mathbf{q})$. A depth camera, such as a Structure Sensor, returns a grayscale image with resolution $M_d \times N_d$ (e.g. 640 x 480) of depth values, $D = d_{m,n} \forall \{m, n\} \in \{\mathbf{M}_d, \mathbf{N}_d\}$, where $\mathbf{N}_d = \{1 \dots N_d\}$ and $\mathbf{M}_d = \{1 \dots M_d\}$. A color camera, with a field-of-view wider than the depth camera's, can be calibrated and synchronised such that the two cameras form a stereo camera system. By using the cameras' intrinsic and extrinsic parameters from calibration and perspective projection [18], each pixel of the depth image can be registered with an RGB pixel hence converted into a colored 3D-point, $p_{m,n}$, and grouped into a colored point cloud, $\mathcal{P} = p_{m,n} \forall \{m, n\} \in \{\mathbf{M}_d, \mathbf{N}_d\}$.

The overall approach is shown inside the red box of Fig. 2. The goal is to turn the raw color and depth images into probability rivet-occupancy maps that describe the predicted locations of rivets, the volumes predicted to be free of rivets, and all other volumes where the state is unknown. Three classifiers: two image-based (color and depth), and one surface-normal based classifier, each outputs a point cloud which holds the determined location of the rivets in the same coordinate frame as the robot. These point clouds are then put into individual probability maps, consisting of voxels that store the probability that a given volume contains a rivet. The maps output from the approach could be fused into one probability map for further analysis or planning purposes.

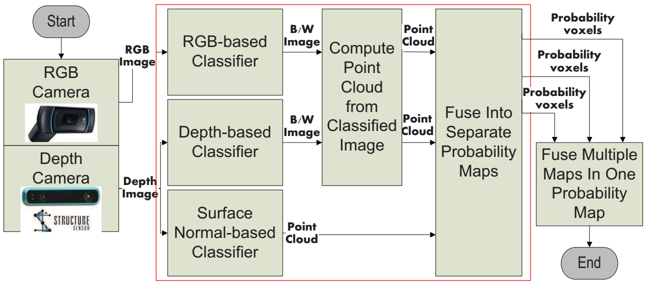


Fig. 2. Overview of data flow with the approach shown inside the red box. The fusion of maps from the three classifiers and from multiple sensor readings is shown for completeness but is outside the scope of this paper.

B. Color Image Classifier

In order to illuminate the normally pitch-black application space for inspection purposes, a small light source is mounted close to the camera. This camera sensor-light source combination basically turns a passive camera into an 'active sensor', which most significantly illuminates protruding objects, such as steel rivets due to their dome-shaped head. These protruding objects generally give strong reflection from wherever they are observed. The flowchart of the color image classifier is shown in Fig. 3.

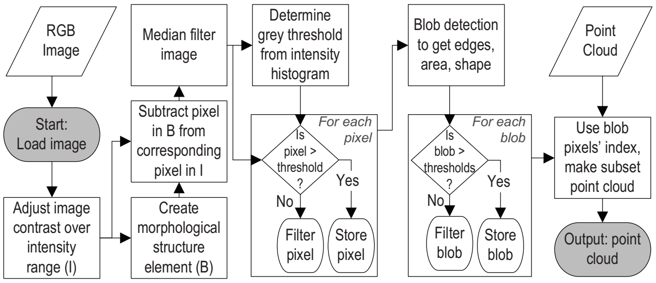

RGB Image
Start: Load image
Adjust image contrast over intensity range (I)
Median filter image
Subtract pixel in B from corresponding pixel in I
Create morphological structure element (B)
Determine grey threshold from intensity histogram
For each pixel
Is pixel > threshold ?
No
Yes
Filter pixel
Store pixel
Blob detection to get edges, area, shape
For each blob
Is blob > thresholds ?
No
Yes
Filter blob
Store blob
Point Cloud
Use blob pixels' index, make subset point cloud
Output: point cloud

Fig. 3. Overview of image processing in the image classifier with a registered point cloud output for predicted rivet locations.

The RGB image, is normalized to fully occupy its dynamic range for optimal threshold-based detection. Contrast enhancement is crucial for successful morphological operations, which consist of three steps: (1) a intensity histogram equalization step for contrast enhancement; (2) a dilation and erosion step to achieve maximum separation between foreground objects from background image; (3) the remaining foreground image undergoes yet another filtering and histogram equalization to come into a fully normalized state. This new image has clean

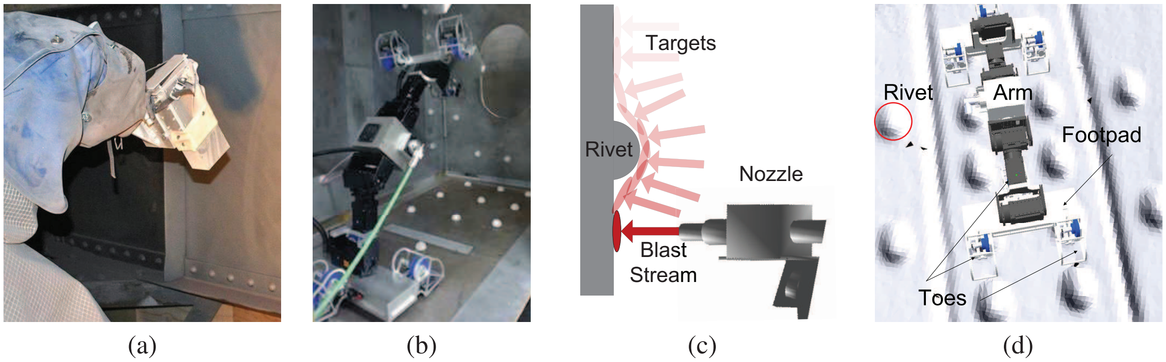


Fig. 1. *a*) Grit-blasting robot [4], *b*) Climbing inspection robot [1], *c*) Simulated blasting robot cleaning rivets by pointing a blast stream of high-velocity air and garnet particles at target points on the surface with an appropriate orientation, *d*) Simulated climbing robot stepping around rivets on a surface.

foreground objects embedded in the background. The image is segmented and connected objects (i.e. blobs) are labelled. At this point, foreground and backgrounds are completely separated. The next stage is identification of elliptical shaped rivets from blobs. This is achieved by firstly applying Canny Edge Detection to obtain contours of the blobs, then ellipse fitting to the connected contours. Thresholds, based upon *a priori* rivet shape knowledge (e.g. ellipse area, long-to-short axis ratio and "goodness" of fit), are applied to each fitted ellipse to filter out noise. Registered colored 3D-points of the rivets are then recovered from the 2D pixel position.

C. Depth Image Classifier

This classifier is similar to the Color Image Classifier, with the normalization and background removal stage replaced with a plane removal process. Planes are detected using an iterative RANSAC plane fitting algorithm as shown in Algorithm 1. In each iteration, i, RANSAC [19] processes points in $\mathcal{P}$ and outputs the largest plane, Π_i. If the plane is valid, i.e. the point count, $size(\mathcal{P}_i)$ is above a threshold, $\tau_r = 1\%$, then these fitted points are removed from $\mathcal{P}$ which becomes the input to the next iteration. The algorithm terminates when the latest plane is too small.

Algorithm 1: Iterative RANSAC Large-plane Removal

```
while $0 < size(\mathcal{P})$ do
    Pass $\mathcal{P}$ into RANSAC;
    Output is largest plane, $\Pi_i$ with point set $\mathcal{P}_i$ ;
    if $\tau_r < size(\mathcal{P}_i)$ then
        $\mathcal{P} = \mathcal{P} - \mathcal{P}_i$;
    else
        exit
```

RANSAC can mistakenly fit a group of rivets into one plane when sections of the rivet data points are within plane fitting tolerance. This problem can be resolved by performing a dispersion test. A plane dispersion measure is defined as,

$$\sigma^2 = \frac{A_{p.i}}{size(\mathcal{P}_i)} \tag{2}$$

where $A_{p.i}$ is the ith plane's contour area (i.e. area of the contour of points on the plane in the 2D depth image), and $size(\mathcal{P}_i)$ is the plane's point count. A small σ^2 indicates a high density of closely located points. However, a large σ^2 shows that points are sparsely distributed, as is typically found at rivets, therefore the plane is discarded.

All pixels, I_{Plane} and points associated with the N large planes, $\Pi_i, \forall i \in \{1, \ldots N\}$, shown in Fig. 4*a* are then removed. Such that an image, $I_D^* = I_D - I_{Plane}$ has plane-associated points as black, and RANSAC outliers (i.e. rivets and perturbations) as yellow (Fig. 4*b*). I_D^* is then passed through the image classifier (Fig. 3).

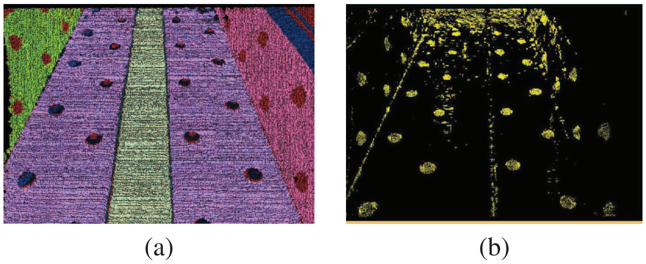


Fig. 4. *a*) Original image with RANSAC applied to point cloud. *b*) Depth image once all points on the main planes have been set to black.

D. Surface-based Normal and Patch Classifier

The third classifier, shown in Fig. 5, analyses the surface normals of a triangulated point cloud to determine the points with dissimilar normals, and thus a high probability of describing a rivet.

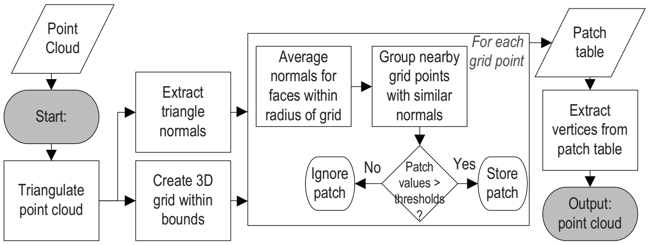
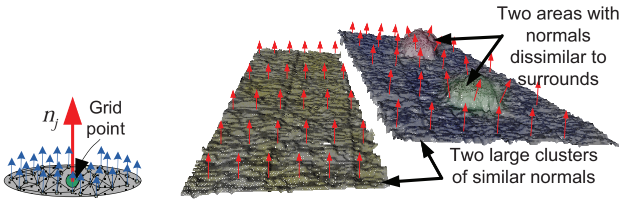

Point Cloud
Start:
Triangulate point cloud
Extract triangle normals
Create 3D grid within bounds
Average normals for faces within radius of grid
Group nearby grid points with similar normals
For each grid point
Ignore patch
No
Patch values > thresholds ?
Yes
Store patch
Patch table
Extract vertices from patch table
Output: point cloud

Fig. 5. Overview of surface classifier.

The classifier takes in the point cloud and triangulates it by grouping nearby sets of 3 vertices into a face definition provided that: (1) the length of all edges $||a||$, $||b||$ and $||c||$ are less than a reasonable bounds (i.e. 5mm), and (2) the set of ratios of all sides, $s = \{\frac{a}{b}, \frac{a}{c}, \frac{b}{c}\}$ are within a pre-specified thresholds range. Thus, preventing spurious data points creating non-existent triangles. The classifier then analyses and groups similar normals together to iteratively generates patches [20]. The output is a new point cloud that is a subset of the grid-based samples generated by the methodology explained henceforth.

A grid is created over the space, then at each grid point the normals of the triangles are averaged as shown in Fig. 6*a*, and an average normal value for that grid point is output as shown in Fig. 6*b*. Note how the grid points do not correspond to locations of the vertices, but instead are axis aligned and regularly sampled. The grid points are clustered based upon the normals. Patches with large numbers of points are discarded and only the remaining vertices are returned. These are generally on, or nearby, rivets or surface edges.


n_j
Grid point
Two areas with normals dissimilar to surrounds
Two large clusters of similar normals

(a) (b)

Fig. 6. *a*) Average triangle surface normals, $\mathbf{n}_j$ at a grid point, j. *b*) Normal clustering example so large clusters can be identified and removed, leaving normals dissimilar to surrounding normals as likely rivets candidates.

Although this classifier may appear similar to the Depth image classifier, there are several important differences that allow rivet identification in certain situations, and mean that the outputs are in fact quite different. Clustering for plane filtering is done based upon the triangulated surface normals, rather than RANSAC, and segmentation is based upon axis-aligned sampled grid points. Thus, this classifier does not attempt to locate rivets, but instead is looking for voxels that contain rivets, which is a slight but important difference. This classifier produces noisy results, but will identify rivets that one of the image classifier may miss or filter out. Note that it is difficult to recover the original RGB-D image from the grid points since they are in different coordinate frames.

E. Probabilistic Map Storage

Since this is a classic perception problem with multiple information sources, it was decided that information fusion be done using a probability mapping technique [21]. The output point clouds, $\mathcal{P}_{cx}$ from the three classifiers, $cx \in \{c1, c2, c3\}$ are fused using a probabilistic rivet-occupancy estimation and the OctoMap library. Octrees are used as a hierarchical data structure for 3D spatial subdivision. An octree node represents space contained in a voxel (i.e. volumetric pixel), and enables rapid searching by recursively subdividing a volume into eight sub-volumes, and reduced memory requirements since when a node's children are in the same state they are pruned. Each voxel can be assigned, and updated with a probability that a rivet occupies this volume.

Firstly a 3D occupancy map is constructed with an axis-aligned 3D grid at a resolution that is at half the size of a rivet. Then each voxel is individually treated as having a mutually exclusive reading about the existence of a rivet, which are independent of the distance it was measured from the sensor. For a certain classifier, cx, the probability, $P_{cx}(n|z_{1:t})$ of a leaf node, n containing a rivet according to the t sensor measurements, $z_{1:t}$ is estimated according to

$$P_{cx}(n|z_{1:t}) = \left[1 + \frac{1-P_{cx}(n|z_t)}{P_{cx}(n|z_t)} \frac{1-P_{cx}(n|z_{1:t-1})}{P_{cx}(n|z_{1:t-1})} \frac{P_{cx}(n)}{1-P_{cx}(n)}\right]^{-1} \qquad (3)$$

This update formula depends on the current measurement z_t, a prior probability, $P_{cx}(n)$, and the previous estimate $P_{cx}(n|z_{1:t-1})$. The term $P_{cx}(n|z_t)$ denotes the probability of voxel n containing a rivet given the classification output from a specific classifier, z_t which can theoretically be different, depending upon the sensor model and the trustworthiness of the classifier output. Then the map, $P_{m,cx}$ of classifier, cx, is updated using a Bayesian update given a prior and posterior so that afterwards each voxel has a weighting. Each point in the point clouds output by each classifier is iterated through so that in the end there is a probability value for each classifier, cx, for each voxel, $P_{cx}(n|z_{1:t})$.

Although, it is possible for a single map to be generated rather than producing separate maps, in this paper separate maps allows for the results from the classifiers to be compared. For practicality and visualisation, the output of probability maps can be reduced to a point cloud where a point represents the likely location of a rivets. The point clouds can then be given to the climbing robot step path planner as obstacles that the robot must avoid, or to other surface interaction planners, such as a grit-blasting robot's cleaning/maintenance module, so as to enable well-rounded, complete coverage of rivets.

III. Results

Three experiments have been conducted in the steel bridge tunnel environments shown in Fig. 7 using a 7DoF climbing inspection robot with two cameras mounted to the end-effector: a Structure Sensor depth camera, and a Logitech C930e RGB camera. The robot can attach one or two footpads to steel surfaces in the environment using 3 controllable, permanent-magnet toes that are in each footpad. Each RGB-D frame is processed: the resultant point cloud data is triangulated to generate a mesh, a large plane set, and the tunnel and manhole plate are detected [8]. The presented approach then generates a probability map for each classifier. Experiment 1 tests the accuracy of the two image-based classifiers by comparing the distance between detected rivets with ground truth in a simple lab environment. Experiments 2 and 3 aim to demonstrate the robustness of the approach by processing several different RGB-D datasets taken from different poses in both the lab and in a real-world environment. The mesh generated is manually classified to generate a ground truth probability rivet-occupancy map. For each experiment, statistics are provided about the accuracy, sensitivity and specificity, as well as relevant computation times of the classifiers.

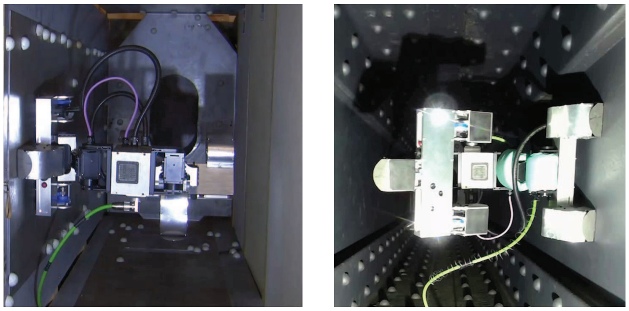

(a) (b)

Fig. 7. Climbing robot performing inspection in two steel bridge tunnel environments containing no lighting infrastructure: *a*) In the lab tunnel; *b*) In the real-world tunnel, walking towards the photographer.

Experiment 1 was conducted in a laboratory mock-steel bridge environment (Fig. 7*a*) containing 12 approximately parallel plates (including steel, timber and foam), there are also patches of surface roughness, blemishes and rust, and the rivets (in white) that are trying to be detected. Fig. 8*a* shows a collected RGB image overlayed with the Depth image. Note how the field of view of the RGB camera is wider than the depth camera. The blue arrows in Fig. 8*a* represent the known ground truth distances between selected rivet pairs as measured with Vernier calipers. The single frame shown is classified and the location of the rivets from the first two image classifiers is recorded and the errors are shown in Fig. 8*b*. As previously mentioned the surface classifier produces probabilities for each voxel rather than discrete rivet locations, so is not shown here. The majority of the errors are less than ±10mm even up to 1m away from the sensor. Fig. 8*c* shows that errors within this range are not a function of the sensor-to-rivet distance.

Experiment 2 has a similar setup to Experiment 1 with the robot inspecting the controlled lab environment in Fig. 7*a*. The robot was placed at several base locations and poses so as to collect a variety of RGB-D images of groups of rivets on the floor and walls. 10 different images were collected and processed. Each triangle mesh collected has the vertices manually classified as shown in Fig. 9*a* and a rivet-occupancy probability map is generated. The ground truth is then compared against the resulting maps from individual classifiers as shown in Fig. 9*b*. Fig. 9*c* shows the output classification from the RBG classifier, note one false positive misclassification labelled as rivet 8.

In Experiment 3 the robot inspected several unlit, real-world steel bridge environments as it walked along the walls (Fig. 7*b*). The ground truth of the rivet locations were manually classified. Data past 2m is discarded since it is neither trustworthy (due to noise), nor useful, due to images being taken every 1.5m for inspection purposes anyway. Fig. 9*d* shows a color-image classifier result from Experiment 3.

As mentioned, each vertex from the triangle mesh of the surfaces is manually classified to produce a ground truth. Vertices are fused into a probability rivet-occupancy grid, such that voxels are effectively in three states: containing a rivet with high certainty, not containing a rivet, or unknown. The test results are compared with ground truth and the binary classification statistical measures of Accuracy, Sensitivity and Specificity are calculated. Firstly each voxel is classified, and compared to the ground truth, it is considered a true positive, TP if a voxel containing a rivet is correctly identified as containing a rivet; a true negative, TN if a voxel free of rivets is correctly identified as not containing a rivet; a false positive, FP if a rivet-free voxel is incorrectly identified as containing a rivet; and finally a false negative, FN where voxel's containing a rivet is incorrectly classified as free of rivets. Thus, the goal for each classifier is to output only "true" results and no "false" results. The Accuracy measure is the total number correctly classified as given by

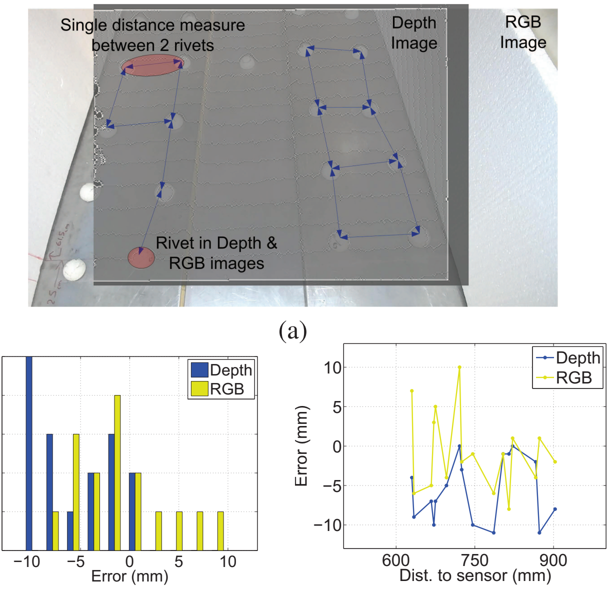
$$Accuracy = \left(\sum TP + \sum TN\right)/\Delta \tag{4}$$

where $\Delta = \sum TP + \sum TN + \sum FP + \sum FP$. The Sensitivity measure is the test's ability to correctly detect voxels which do in fact contain a rivet as,

$$Sensitivity = \sum TP / \left(\sum TP + \sum FN\right) \tag{5}$$

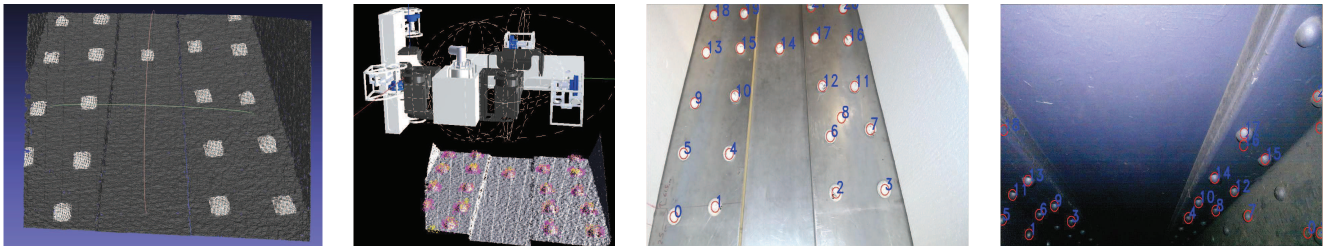
The Specificity measure relates to the test's ability to correctly detect rivets without a voxels as,

$$Specificity = \sum TN / \left(\sum TN + \sum FP\right). \tag{6}$$



(a)
(b) (c)

Fig. 8. *a*) Distance between rivets with RGB and Depth image overlayed. Measurements (shown as blue arrow) taken manually to be compared with the classification results. *b*) Rivet Separation Error Histogram; *c*) Rivet Separation Error relative to Distance from Sensor:



(a)

(b)

(c)

(d)

Fig. 9. *a*) Manually classified rivets from a single scan mesh; *b*) Simulated robot showing actual robot's pose, and sensor data in a triangle mesh, with manually classified rivet voxel locations in Magenta and depth-based classifier results in yellow, *c*) Results of just the RGB color classifier Exp. 1, *d*) and Exp. 3.

Table I presents the statistical measures results of individual classifiers: 1=color image, 2=depth image, 3=surface normal.

Exp.	Classifier	*1*	*2*	*3*
1	Accuracy	0.978	0.981	0.967
	Sensitivity	0.641	0.669	0.622
	Specificity	0.988	0.99	0.977
2	Accuracy	0.983	0.987	0.976
	Sensitivity	0.654	0.689	0.648
	Specificity	0.99	0.994	0.983
3	Accuracy	0.994	0.996	0.979
	Sensitivity	0.307	0.418	0.575
	Specificity	0.998	0.998	0.981

TABLE I
AVERAGE STATISTICAL MEASURES FOR EACH EXPERIMENT'S DATASET.

The Accuracy of classifier 2 (i.e. depth image-based) was the highest in the 3 experiments. However it is less sensitive than classifier 3 (surface normal-based) in the field environment (Experiment 3). The image-based classifiers (i.e. 1 and 2) produce a cleaner result, whereas classifier 3 produces more noise, due to the higher number of false positives. Each classifier produces a relatively high count of true negatives such that all classifiers reach upwards of 95% accuracy and specificity. The sensitivity of the results is relatively low in the field environment. It was found by looking at the overlapping spheres of voxels classified as rivets and those manually classified that the manual classification generally includes the skirt around the rivets even where there may be sparse data patches. Rivets are generally not missed, instead the manual classification overestimates the rivet size, leading to increased False negatives and thus lower sensitivity values.

The data from the three experiments was processed 10 times each, for a total of 280 runs of each classifier. For each experiment the processing time for each classifier was recorded and is shown as box plots in Fig. 10. Overall, the colour image and Surface normal based classifiers (i.e. 1 and 3) consistently take approximately 1.8secs, no matter the environment. In the lab environment the Depth-based classifier (i.e. 2) is significantly faster, averaging 750ms. However in the field environment, which contains more surface anomalies, the time taken varies significantly, although on average it is still less than classifier 1 and 3. Classifier 2 depends on the RANSAC algorithm to generate the tunnel wall planes, and this takes the majority of the time and varies most widely. Each classifier is written in C++ and runs in its own thread, and since there is no dependence between the data, all 3 classifiers can run in 3 separate threads simultaneously - such that the total time is similar to the makespan of the three parallel classification tasks (i.e. max. time of the three). The time to fuse the classification data together into probability rivet-occupancy maps, is negligible, taking less than 100 milliseconds.

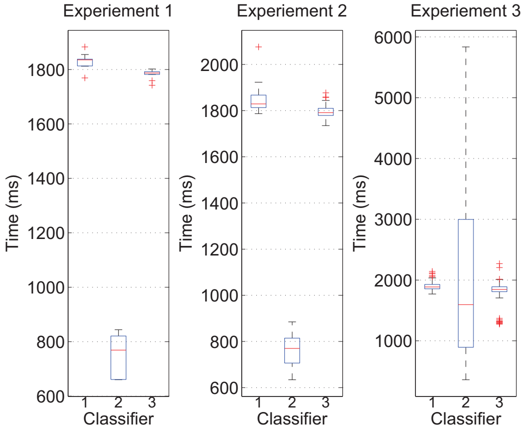


Fig. 10. Time taken to complete a single classification using each classifier for each dataset (i.e. Depth and RGB frame) in each of the 3 experiments.

IV. Discussion

The proposed approach has been shown to detect the rivets in several variations of the target environments. The approach can generally process data within 2 seconds and is robust to noise due to the probability fusion. There are several issues that require further analysis, such as if the grid for the occupancy map should be forced to align with prior knowledge about rivet patterns, if it were available, rather than being axis aligned. It has been observed that the patterns formed in infrastructure, such as bridges, are not stochastic and generally follow some sort of pattern, which have not been fully exploited by this approach. Currently the classifier outputs have been shown to be independent of the distance away from the sensor within the test range. However, it has been observed that outside of the stated range, the predictions become poorer possibly as a function of this distance. Therefore, outside this range, a different sensor model may be required to update the rivet-occupancy probability map, due to the likely sensor-rivet distance dependence.

V. Conclusions

This paper has presented an approach that utilises low-cost, non-contact RGB-D sensors and classifies the data so as to rapidly and robustly determine the location of rivets in steel bridge environments. The predicted rivet locations output from the classifiers are fused using probabilistic mapping. The approach has been shown to work in several different laboratory and real-world environments, even when there is no external light infrastructure, and when the sensors are attached to a mobile platform. Although the raw RGB-D data may be noisy, the location of nearby rivets can be found and the output probability map can augment existing geometric maps, which are required for planning purposes. Future work will evaluate improvements and additions to the classifiers functions, investigate and optimise the parameters of the sensor model so as to improve the output map, and incorporate the approach into a guided exploration approach so as to build a complete rivet map of the surrounds.

Acknowledgments

Thank you to Peter Ward and Chia-han (John) Yang for R&D on the climbing robot platform and help collect the results. This work is supported by the NSW Roads and Maritime Services, and the Centre for Autonomous Systems (CAS) at the University of Technology, Sydney

References

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

[2] G. Peters and G. Paul, "Maintaining an old icon with a new technology," *Journal of Protective Coatings & Linings*, vol. 32, no. 8, pp. 22–31, 2015.

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

[4] G. Paul, S. Webb, D. K. Liu, and G. Dissanayake, "A robotic system for steel bridge maintenance: Field testing," in *Proc. Australasian Conference on Robotics and Automation*, Brisbane, 2010, pp. 1–8.

[5] G. Paul, S. Mao, L. Liu, and R. Xiong, "Mapping repetitive structural tunnel environments for a biologically inspired climbing robot," in *18th Int. Conf. on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 2015, pp. 325–333.

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

[8] G. Paul, P. Quin, A. To, and D. Liu, "A sliding window approach to exploration for 3d map building using a biologically inspired bridge inspection robot," in *IEEE Int. Conf. on CYBER Technology in Automation, Control, and Intelligent Systems*, 2015, pp. 1097–1102.

[9] S. Sehestedt, G. Paul, D. Rushton-Smith, and D. Liu, "Prior-knowledge assisted fast 3d map building of structured environments for steel bridge maintenance," in *IEEE Int Conf. on Automation Science and Engineering (CASE)*, 2013, pp. 1040–1046.

[10] U. Park, Y. Fan, L. Udpa, P. Ramuhalli, W. Shih, and G. C. Stockman, "Automated rivet inspection for aging aircraft with magneto-optic imager," in *10th International Workshop on Electromagnetic Nondestructive Evaluation*, 2004.

[11] Y. He, F. Luo, M. Pan, F. Weng, X. Hu, J. Gao, and B. Liu, "Pulsed eddy current technique for defect detection in aircraft riveted structures," *NDT & E International*, vol. 43, no. 2, pp. 176–181, 2010.

[12] P. Fromme and M. B. Sayir, "Detection of cracks at rivet holes using guided waves," *Ultrasonics*, vol. 40, no. 1, pp. 199–203, 2002.

[13] R. A. Osegueda, V. Kreinovich, S. Nazarian, and E. Roldan, "Detection of cracks at rivet holes in thin plates using lamb-wave scanning," in *NDE for Health Monitoring and Diagnostics*. International Society for Optics and Photonics, 2003, pp. 55–66.

[14] Z. Liu, D. S. Forsyth, A. Marincak, and P. Vesley, "Automated rivet detection in the eol image for aircraft lap joints inspection," *NDT & E International*, vol. 39, no. 6, pp. 441–448, 2006.

[15] A. W. K. 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, 2014.

[16] L. Spinello and K. O. Arras, "People detection in rgb-d data," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3838–3843.

[17] S. Wan and J. Aggarwal, "Robust object recognition in rgb-d egocentric videos based on sparse affine hull kernel," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 97–104.

[18] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[19] M. A. Fischler and R. C. 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.

[20] G. Paul, D. K. Liu, and N. Kirchner, "An algorithm for surface growing from laser scan generated point clouds," in *Robotic Welding, Intelligence and Automation*, T. Tarn, S. Chen, and C. Zhou, Eds. Berlin: Springer-Verlag, 2007, pp. 481–491.

[21] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013. [Online]. Available: http://octomap.github.com