Robotica (2019) volume 37, pp. 947–967. © Cambridge University Press 2018
doi:10.1017/S0263574717000601

Automated robotic monitoring and inspection of steel structures and bridges

Hung Manh La[†]*, Tran Hiep Dinh[‡], Nhan Huu Pham[†], Quang Phuc Ha[‡] and Anh Quyen Pham[†]

[†]*Advanced Robotics and Automation (ARA) Lab, Department of Computer Science and Engineering, University of Nevada, Reno, USA. E-mails: hla@unr.edu, nhan.ph@nevada.unr.edu, phamquyenanh.qb@gmail.com*
[‡]*School of Electrical, Mechanical and Mechatronic Systems, University of Technology Sydney, Australia. E-mails: TranHiep.Dinh@student.uts.edu.au, quang.ha@uts.edu.au*

(Accepted December 13, 2017. First published online: January 11, 2018)

SUMMARY

This paper presents visual and 3D structure inspection for steel structures and bridges using a developed climbing robot. The robot can move freely on a steel surface, carry sensors, collect data and then send to the ground station in real-time for monitoring as well as further processing. Steel surface image stitching and 3D map building are conducted to provide a current condition of the structure. Also, a computer vision-based method is implemented to detect surface defects on stitched images. The effectiveness of the climbing robot's inspection is tested in multiple circumstances to ensure strong steel adhesion and successful data collection. The detection method was also successfully evaluated on various test images, where steel cracks could be automatically identified, without the requirement of some heuristic reasoning.

KEYWORDS: Field robotics; Climbing robots; Steel bridge inspection; Image stitching; 3D map construction; Steel crack; Histogram thresholding; Image segmentation.

1. Motivation and Background

Steel structures and steel bridges, constituting a major part in civil infrastructure, require adequate maintenance and health monitoring. In the United States, more than 50,000 steel bridges are either deficient or functionally obsolete,[1] which present likely a growing threat to people's safety. Collapse of numerous bridges recorded over past 15 years has shown significant impact on the safety of all travelers. For instance, the Minneapolis I-35W Bridge in Minnesota, USA collapsed in 2007[2] due to undersized gusset plates, increased concrete surfacing load, and weight of construction supplies/equipment. This accident along with others has demanded more frequent and effective bridge inspection and maintenance. Currently, steel infrastructure inspection activities require great amount of human effort along with expensive and specialized equipment. Most steel bridge maintenance tasks are manually performed by using visual inspection or hammer tapping and chain dragging for delamination and corrosion detection, which are very time consuming. Moreover, it is difficult and dangerous for inspectors to climb up or hang on cables of large bridges with high structures, as shown in Fig. 1. In addition, reports from visual inspection may vary between inspectors so the structural health condition may not be assessed precisely. Therefore, there should be a suitable approach to provide consistent and accurate reports on steel bridge conditions along with high efficiency and safety assurance in the task execution.

There have been a significant number of studies related to utilizing advanced technologies for bridge inspection and maintenance. In La *et al.*,[3–7,10] an autonomous robotic system, integrated with non-destructive evaluation (NDE) sensors associated with advanced data processing algorithms,

* Corresponding author. E-mail: hla@unr.edu

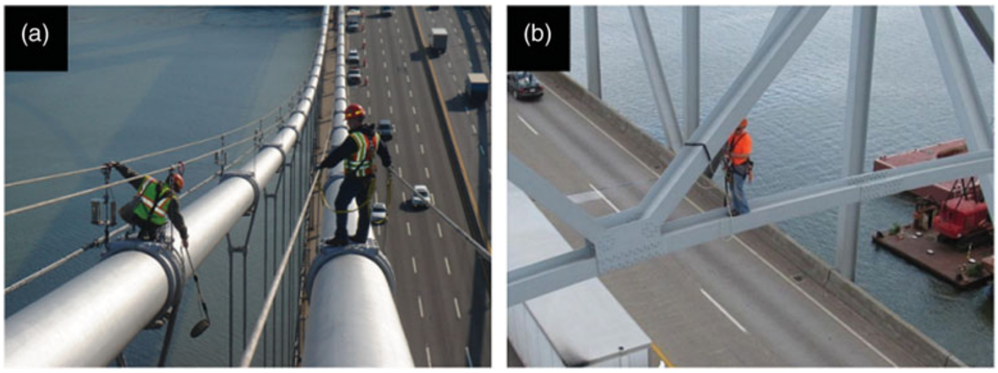

Fig. 1. Dangerous bridge inspection scenes, source: stantec.com.

was developed for high-efficiency bridge deck inspection and evaluation. Additionally, previous developments focusing on real-world bridge deck crack detection were reported in Prasanna *et al.*[20] and Lim *et al.*[8,9] Extensions of La's previous work to multi-functional robotic systems for civil infrastructure inspection were reported in refs. [15, 16]. Li *et al.*[11] also utilized the NDE technique to perform automatic inspection on bridge deck and record the health condition of a bridge. For concrete bridges, NDE sensor-integrated robotic systems were deployed by Gucunski *et al.*[12–14,19] to automate the bridge deck inspection, in which ground penetrating radar (GPR) is an effective tool for building attenuation condition map of bridge decks.[17,18] On the other hand, Xu *et al.*[21] introduced the design and experiments of a wheel-based cable inspection robotic system, consisting of charge-coupled device (CCD) cameras for the visual inspection purpose. A similar robot[22] enables effective visual inspection of the cable on suspension bridges. Moreover, there were several initial implementations of climbing robots for inspection of built infrastructure, including steel bridges. Mazumdar *et al.*[23] proposed a legged robot that can move across a steel structure for inspection. Strong permanent magnets embedded in each foot allow the robot to hang from a steel ceiling powerlessly, while the attractive force is modulated by tilting the foot against the steel surface. Wang, *et al.*[24] developed a robot with magnetic wheels, that is capable of carrying a Giant Magneto Resistive sensor array for crack and corrosion detection. Another bridge inspection method[25,26] featuring a wall-climbing robot with negative pressure adhesion mechanism is used to collect crack images with a high-resolution camera so that a crack can be extracted and analyzed precisely. Based on the attraction force created by permanent magnets, Leibbrandt *et al.*[27] and Leon-Rodriguez *et al.*[28] developed two different wall-climbing robots carrying NDE devices for detection of welding defects, cracks, corrosion testing that can be used to inspect oil tanks or steel bridges. San-Millan[29] presented the development of a tele-operated wall climbing robot equipped with various testing probes and cameras for different inspection tasks. Zhu *et al.*[30] used a magnetic wall-climbing robot capable of navigating on steel structures, measuring structural vibrations, processing measurement data and wirelessly exchanging information to investigate field performance of flexure-based mobile sensing nodes to be able to identify minor structural damage, illustrating a high sensitivity in damage detection. Along with the development of climbing robots, computer-vision-based methods for crack detection of concrete have been extensively researched and gradually employed in the field to assist with the reduction of costs and hazards of manual data collection.[31] Previously, Oh *et al.*[32] introduced a machine vision system, where potential cracks can be identified for bridge inspection by subtracting median filtered images of defects from the original ones. Edge detection algorithms were also employed by Adhikari *et al.*[33] to obtain the skeleton of cracks and compute the corresponding descriptors, such as branch points, length and width profile. Lins *et al.*[34] recently presented a crack detection algorithm by combining the RGB color model with a particle filter to approximate the probability distribution by a weight sample set. The common point of these techniques is the user input requirement to adjust the filter setting for the effectiveness of the proposed approach. However, this may limit the generic application of an automatic inspection system. Our contribution in this paper is a new crack detection technique based on a non-parametric peak detection algorithm where a threshold to isolate crack from the image background is automatically calculated. The proposed technique is only based on a basic knowledge that crack pixels are darker than their surroundings.

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

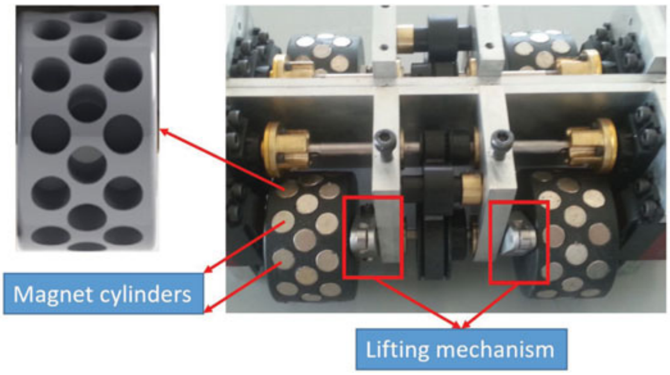

Fig. 2. Embedded magnet cylinders and lifting mechanism.

In this paper, the inspection technique using visual and 3D images coupled with histogram thresholding and image segmentation is applied to our developed climbing robot. Here, the robot is able to strongly adhere to and freely move on a steel surface to collect images from visual as well as 3D cameras. The captured images from the robotic visual camera are then stitched together to provide a whole image of the steel surface for ease of inspection. Then an automatic detection algorithm is developed to detect defects from steel surface's images. Moreover, a 3D map is built from 3D point cloud data in order to assist with robot navigation as well as inspection of bridge structures. With an advanced mechanical design, the robot is able to carry a payload up to approximately 7 kg, while still adhering to both inclined and upside-down surfaces. The robot can also transit from one surface to another with up to 90° change in orientation. Overall, steel surface image stitching and 3D structure mapping are performed from collected data to provide condition assessments for the steel structures.

The rest of paper is organized as follows. Section 2 describes the overall robotic system design. Section 3 presents data collection and image processing techniques. Section 4 shows experimental results to verify the system design and data processing. Finally, Section 5 gives the conclusion and future work.

2. Overall Design of the Inspection Robotic System

2.1. Mechanical design

A robot design with four motorized wheels is proposed, taking the advantage of permanent magnets for adhesion force creation. This allows the robot to adhere to steel surfaces without consuming any power. In addition, cylinder shaped magnets are used for convenience and flexibility in adjusting the attraction force. Moreover, eight high torque servo motors are utilized to drive four wheels for robot navigation and four shafts for robot lifting. A total of 36 Neodymium magnet cylinders[35] embedded in each wheel can create magnetic force up to 14 Kg/wheel. To enhance the friction with steel surface each wheel is covered by a thin layer of cloth. However, this also causes the magnetic force created is reduced to approximately 6 Kg/wheel. Hence, four modified servo motors with a higher torque (3.3 Nm) are used to drive the robot.[36] Additionally, a special mechanism has been designed in order to lift either the front or rear wheels off the ground if the robot is stuck on rough terrains. Figure 2 depicts the magnet cylinders installation and the lifting mechanism mentioned, while Figs. 3 and 4 illustrate the lifting mechanism design and the overall robot design with an initial robot prototype, respectively.

While moving on steel surfaces, it is required to maintain stability of the robot to overcome sliding and turn-over failures. It has been shown that in order to avoid these failures, the magnetic force created by all wheels should satisfy[54]

$$F_{mag} > max\left\{\frac{P\sin\alpha}{\mu} + P\cos\alpha; 2\frac{Pd}{L}\right\}, \tag{1}$$

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

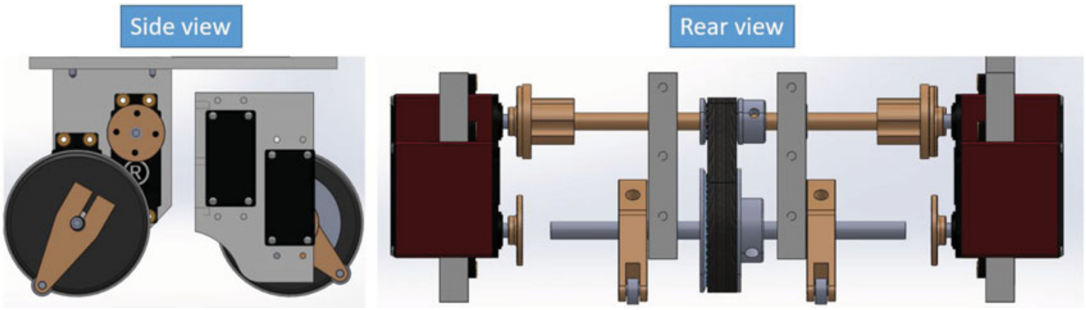

Fig. 3. Lifting mechanism design.

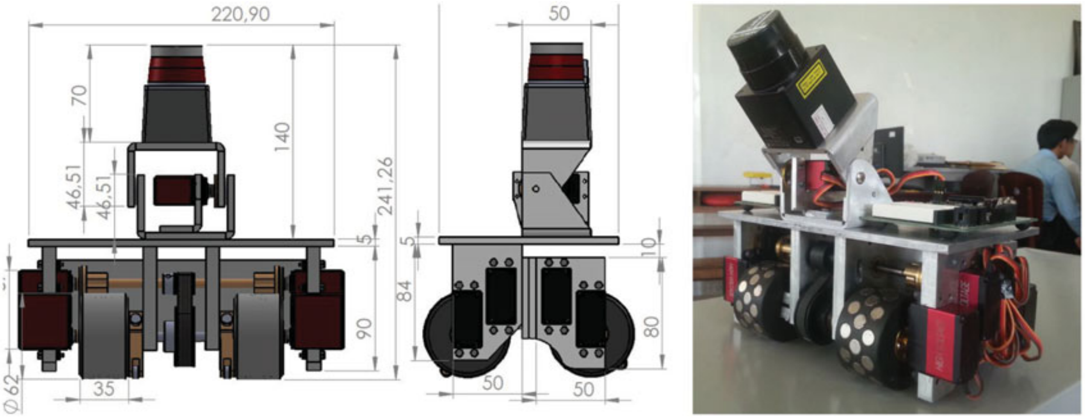

Fig. 4. Robot design and initial prototype.

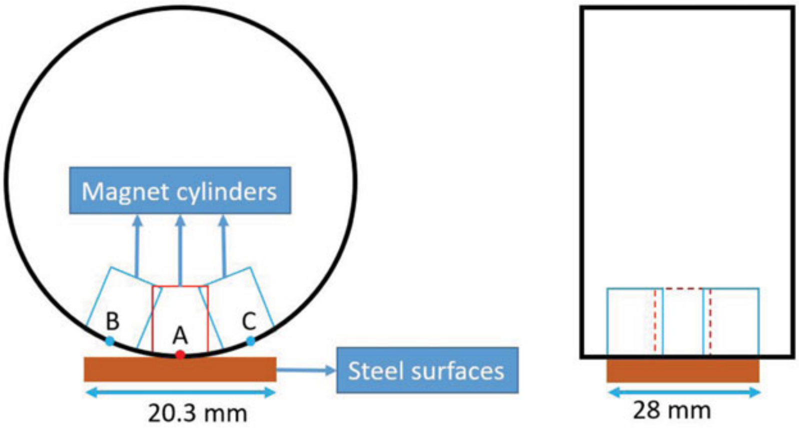

Fig. 5. Magnetic force analysis under limited steel surfaces.

where F_{mag} is the total magnetic force created by all wheels, α is the degree of inclination of the steel surface ($0 \leq \alpha \leq 90°$), μ is the frictional coefficient between the wheel cover and steel surface, P is the robot's weight, d is the distance between the center of mass to the surface, and L is the distance between the front and rear wheels. Condition (1) is essential for selecting appropriate robot design parameters. In detail, beside adjusting the total magnetic force, one can satisfy Eq. (1) by reducing the robot's weight or alternating the design to decrease ratio $\frac{d}{L}$. In this design, in order to maintain Eq. (1) the steel surface area underneath each robot's wheel must be at least 20.3 mm × 28 mm as shown in Fig. 5. Therefore, in the presence of bolts, if the contact areas are large enough, the robot will be able to pass through. Otherwise, the robot needs to transition to other surface to move forward. More details of the robot's mechanical design can be referred to refs. [54, 56].

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

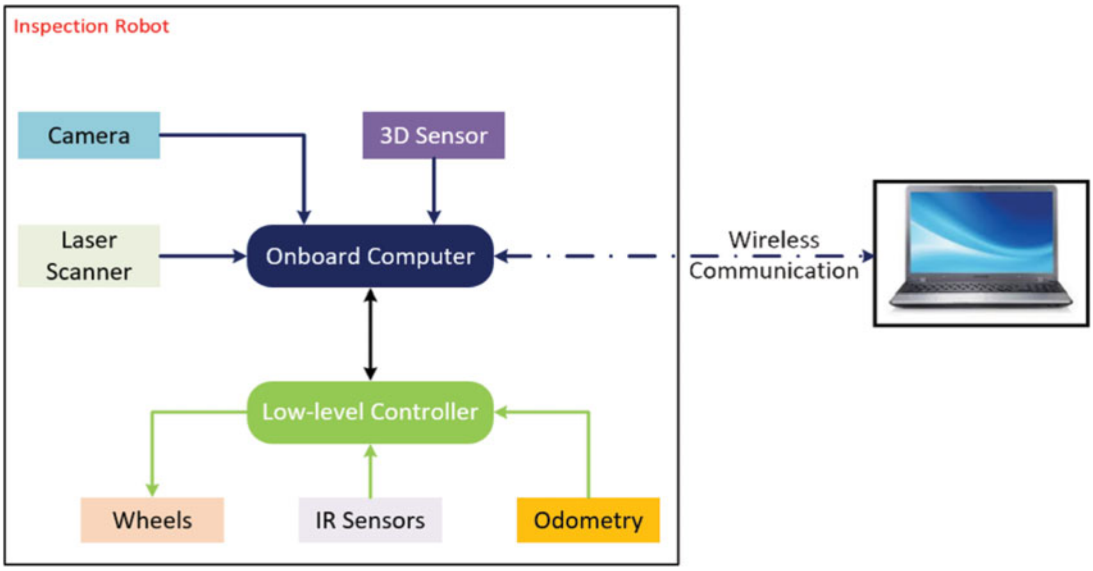

Fig. 6. Robotic system architecture.

2.2. *Robot control*

2.2.1. System integration. Regarding sensing devices, the robot is equipped with multiple imaging sensors: two video cameras for image capturing and video streaming, and a time-of-flight (ToF) camera for capturing 3D data. The USB camera at the back of the robot is installed facing downward in order to take images of steel surfaces and feed to the image stitching unit. Besides, the ToF camera is placed so that 3D images received from the camera can assist with robot navigation as well as building 3D structures.

Apart from cameras, eight Hall Effect sensors are used to detect the presence of a magnetic field. Two sensors are mounted next to each other and close to one wheel. By observing that magnet cylinders inside each wheel will move when the robot moves, we can extract the velocity and traveling distance of each wheel after combining the data collected from these two sensors.

Moreover, the robot has four IR range sensors mounted at four corners of the robot, which can detect whether there exists a surface underneath. Consequently, an edge avoidance algorithm can be implemented using this input to make sure that the robot can safely travel on steel surfaces. The robot is controlled by a micro-controller unit (MCU) handling low-level tasks and a more powerful on-board computer for complex processing and communication with the ground station. The low-level controller has the capability of receiving commands from the on-board computer via serial connections, including motion control and sensors data acquisition signals. The on-board computer is an Intel NUC Core i5 computer responsible for capturing video camera and 3D camera images, then sending them to the ground station over wireless LAN connection for data post processing and logging. It also executes the edge avoidance algorithm with sensors data received from the low-level controller to ensure safe traveling on steel surfaces. The whole robot is powered by two batteries with different voltage levels. One 12 V battery powers the on-board computer and cameras, while another 7.4 V battery supplies to all motors. Overall, the structure of the system is shown in Fig. 6, while the robot with fully installed sensors and other components is depicted in Fig. 7. More details of the robot control can be referred to ref. [54].

2.2.2. Robot navigation. While moving on steel surfaces, there exists a circumstance that the robot moves far away toward the edge of the surface, and may fall off. Therefore, an algorithm using input from IR range sensors is incorporated to prevent this incident. Let us denote $r\_cal_i (i = 1 : 4)$ the calibrated ranges before the robot starts moving, $r_i (i = 1 : 4)$ as IR sensor reading corresponding to $sensor_i$ and $d_i (i = 1 : 4)$ the travel distances calculated from Hall Effect sensors. Given a pre-defined threshold ϵ, when sensor reading is out of the range $[r\_cal_i - \epsilon; r\_cal_i + \epsilon]$, the robot considers that there is no surface below $sensor_i$. The algorithm then adjusts the robot's heading to avoid falling

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

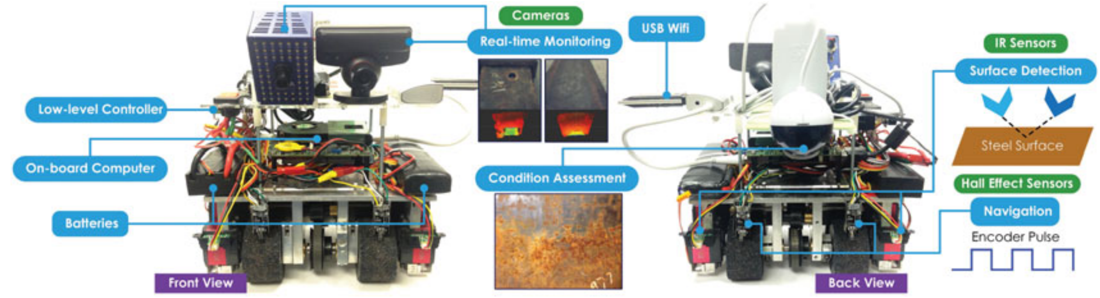

Fig. 7. Robot prototype with integrated sensors.

out of the surface. The summary of the edge avoidance algorithm for safe navigation is presented in Algorithm 1.

Algorithm 1: Edge Avoidance.

```
Input: $(r\_cal_1, r\_cal_2, r\_cal_3, r\_cal_4)$, $(r_1, r_2, r_3, r_4)$, $\epsilon$, $(d_1, d_2, d_3, d_4)$
for $i$=1:4 do
    if only one $(r_i) \notin [r\_cal_i - \epsilon; r\_cal_i + \epsilon]$ then
        if $i$== front right IR sensor then
            Stop;
            Go backward with a distance of 5cm ($\Delta d_i \approx 3$);
            Rotate left when travel distance of either right wheel reach 3cm ($\Delta d_i \approx 2$) ;
            Keep moving;
        Check other sensors and take similar actions;
    else
        stop and wait for commands;
```

It should be noted that the velocity and distance calculated from Hall Effect sensor reading also provide information to implement automatic operations of images capturing, so that we can periodically collect images then stitch them together before applying crack detection algorithms. To ensure that two images can be stitched together, the robot need to stop after traveling a particular distance so that the two consecutive images are overlapped with a minimum percentage.

3. Data Collection and Processing

3.1. Image stitching

In order to enhance steel surface inspection, we combine images captured from the camera at the back, considered as a case of image stitching. Captured images are saved to the memory so that they can be processed later when the robot has finished its operation. In this technique, it is required that two consecutively taken images are at least 30% overlapped as shown in Fig. 8. The camera can cover an area of 18cm × 14cm that means the robot should stop and capture images every 12 cm.

Camera motion is estimated incrementally using sparse feature-matching and image-to-image matching procedures.[37,38] The image is processed left-to-right to find the location of overlapping inside the other image. We can enhance the process by providing an initial estimation of the overlapping area between two consecutive images using robot motion estimation since the camera is fixed on the robot. Then, we can use an appearance-based template matching technique to achieve finer results of the camera motion. If two images have a significant difference in brightness, a technique is applied to determine a new exposure value of the combined image. If the arriving pixel is significantly brighter than the existing corresponding pixel, we replace the old pixel with the new one. A threshold value τ is used to consider whether it is brighter or not. The exposure compensation and blending

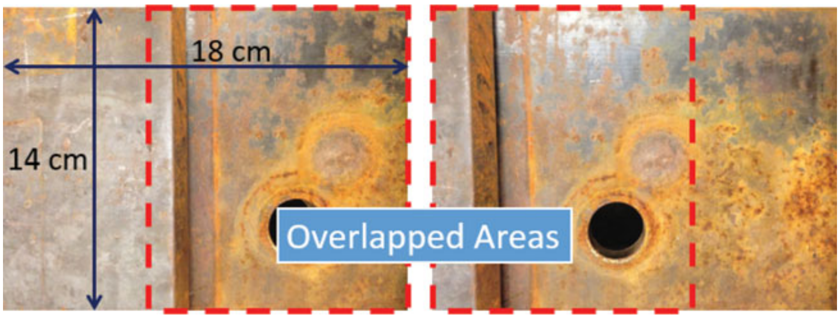

Fig. 8. Overlapped images used for stitching.

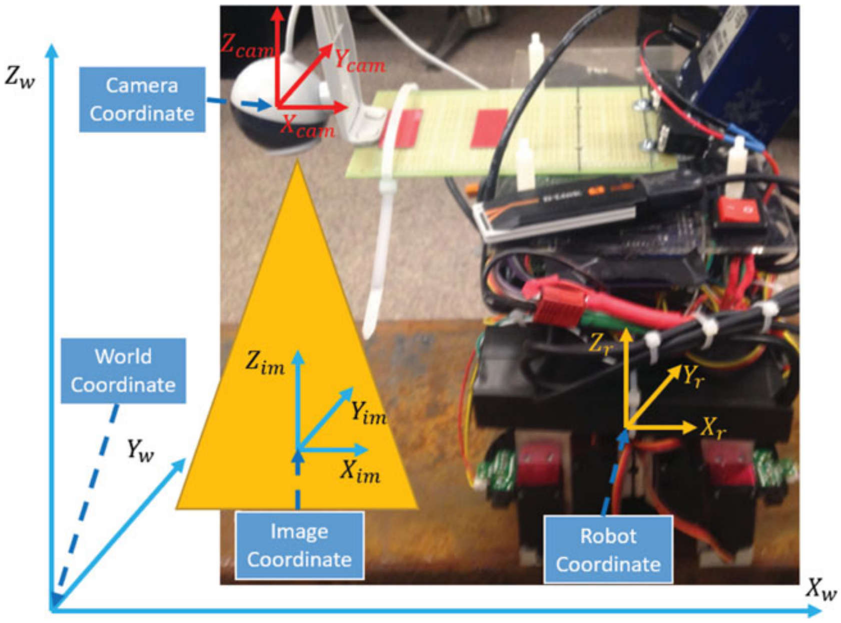

Fig. 9. Relationship between multiple frames.

technique applied to two consecutive images is then described by the following intensity:

$$I(x, y) = \begin{cases} I_2(x, y) & \text{if } \tau I_2(x, y) > I_1(x, y) \\ I_1(x, y) & \text{otherwise.} \end{cases} \tag{2}$$

Gaps in the region formed by the pixels to be used from the new image are filled by using a 2D median filter. As a result, the completeness of the shadow removal region is maintained.

While stitching all of the images, we can also transform the stitched image's coordinate to the world's coordinate. Denote (X_{im}, Y_{im}, Z_{im}) the coordinates in the image frame, $(X_{cam}, Y_{cam}, Z_{cam})$ coordinates in the camera frame, (X_r, Y_r, Z_r) coordinates of the robot frame and (X_w, Y_w, Z_w) coordinates of the world fixed frame, as shown in Fig. 9.

The following series of transformation should be done to convert coordinates in the image frame to the world frame:

$$(X_{im}, Y_{im}, Z_{im}) \xrightarrow{T_{ic}} (X_{cam}, Y_{cam}, Z_{cam}) \xrightarrow{T_{cr}} (X_r, Y_r, Z_r) \xrightarrow{T_{rw}} (X_w, Y_w, Z_w)$$

where T_{ij} are the transform matrix from frame i to frame j. Since the camera location is fixed on the robot, T_{ic} and T_{cr} can be easily obtained by measuring the distance between the camera and the steel surface and the robot's center of mass, while T_{rw} can be extracted from odometry.

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

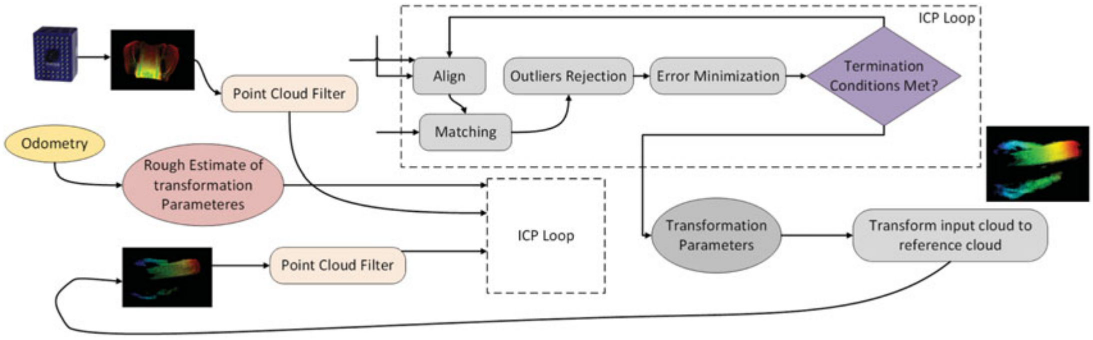
ICP Loop
Align
Outliers Rejection
Error Minimization
Termination Conditions Met?
Matching
Point Cloud Filter
Odometry
Rough Estimate of transformation Parameteres
ICP Loop
Point Cloud Filter
Transformation Parameters
Transform input cloud to reference cloud

Fig. 10. 3D map construction process employing ICP algorithm.

The resulting stitched image can be really useful in real-life application. Instead of viewing hundreds or thousands of steel surface images, inspectors can have an overall image of the whole steel surface that the robot has traveled. Hence, it improves the efficiency of visual inspection of a bridge.

3.2. 3D construction

The concept of 3D construction[55] or 3D registration is based on the Iterative Closest Point (ICP) algorithm introduced in Besl *et al.*[39] and Chen.[41] The algorithm has been used to construct 3D models of objects in robotic applications including mapping and localization. Here, the ICP goal is to find the transformation parameters (rotation and translation) that align an input point cloud to a reference point cloud. Those transformation parameters are presented in the coordinate frame of the reference point cloud. Figure 10 shows the process of 3D registration, including the ICP algorithm, which consists of the following steps:[42]

1. Selection: Input point clouds captured from a ToF camera are pre-sampled for a higher efficiency.
2. Matching: Odometry data can be used to estimate correspondences between the points in the sub-sampled point clouds, considered as the initial alignment. After that, we apply feature matching using Fast Point Feature Histograms (FPFH)[40] on two roughly aligned images to obtain finer transformation.
3. Rejection: Filtering the correspondences to reduce the number of outliers, multiple points with the same corresponding point are rejected.
4. Alignment: Computing assessment criteria, normally point-to-point or point-to-plane error metrics, then minimizing them to find an optimal transformation.

The algorithm stops at one of these cases as follows:

1. The error metrics decrease to within or remain constant below a threshold.
2. The algorithm does not converge after a given number of iterations.
3. Transformation parameters do not vary or are out of bound.

3.3. Steel crack detection

In robotic applications, many vision-based approaches have been proposed to deal with robotic color tracking and image segmentation[31,43,44,57] and employed to solve the defect detection problem.[45–47] To be applied in a fully automatic process, the proposed approach must be able to handle the input data with limited supervision and manual adjustment.In this work, a hybrid method is proposed to solve the mentioned problem by combining our automatic peak detection algorithm with image stitching and 3D registration. At this stage, the stitched images from data collected by the cameras are further processed to automatically detect a corrosion or crack on the steel coating surface. The potential structural or surface defect is first isolated from the image background by using a global threshold, which is obtained from our automatic peak detection algorithm. A Hessian matrix-based filter is then employed to eliminate the blob- or sheet-like noise and to emphasize the defect nature.[48,49] To avoid the disconnectivity of identified cracks or the misdetection of thin coating surfaces, the

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

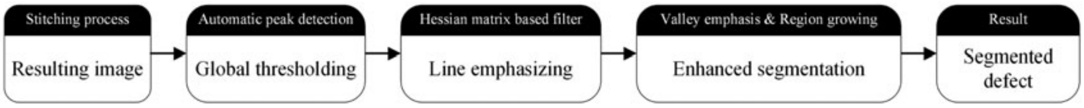

Fig. 11. Proposed approach for steel crack detection.

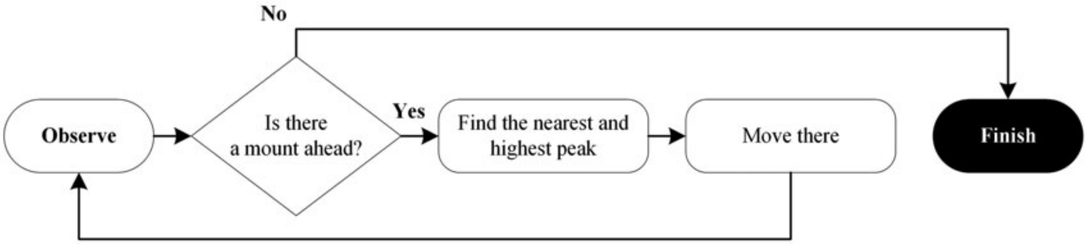

Fig. 12. Strategy for a lost mountains explorer.

region growing algorithm, see, e.g.,[50] is applied to enhance the segmentation performance. Figure 11 illustrates the processing steps for our proposed approach.

3.3.1. Automatic peak detection algorithm. The proposed algorithm used in this work is based on perceptions of a mountain explorer, whereby observation and planning remain the key factors to be considered when getting lost in mountain exploration. In this circumstance, a feasible solution is to repeat the process of finding a high location to observe and identifying a specific landmark to head to as well as planning for the next travel segment until the explorer finally gets back on track. To illustrate this strategic planning, a flowchart is described in Fig. 12. Therein, two main steps are involved, namely, searching for observing location and for the highest peak in the closest distance.

The gray-scale histogram of the resulting image is first smoothed using a moving average filter with the width of kernel equal to 3, taking into account the previous, current and next intensity level. This filter is chosen for our approach because of its compactness and robustness in eliminating random noise while retaining significant peaks of the histogram. Let $h(i)$ be the pixels number at intensity level i for $i = 1, 2, \dots, L+1$ where L is the maximum intensity level. After applying the moving average filter, the pixels number at intensity level i is determined as follows:

$$h(i) = \frac{1}{3}[h(i-1) + h(i) + h(i+1)]. \tag{3}$$

An initial peak of the smoothed histogram is identified if its intensity value is greater than that of its two nearest neighbors. The significant peak detection strategy is then built based on these initial ones. Therefore, initial peaks can be considered as a draft plan and stored in a cluster vector δ, where each element must meet the following criteria:

$$\begin{cases} h(\delta(k)) > h(\delta(k) - 1) \\ h(\delta(k)) > h(\delta(k) + 1). \end{cases} \tag{4}$$

Observing location: Let consider the smoothed histogram as the mount to be explored. A point on the intensity axis could then be considered as the observing location $\alpha(m)$ if the following condition is fulfilled:

$$\alpha(m) < g(\delta(k)) - L(\delta(k)), \tag{5}$$

where $L(\delta(k))$ is a dynamic offset distance from the current peak to the observing location and dependent on the draft plan, $g(\delta(k))$ is the intensity level at the kth initial peak and p_D is the number

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

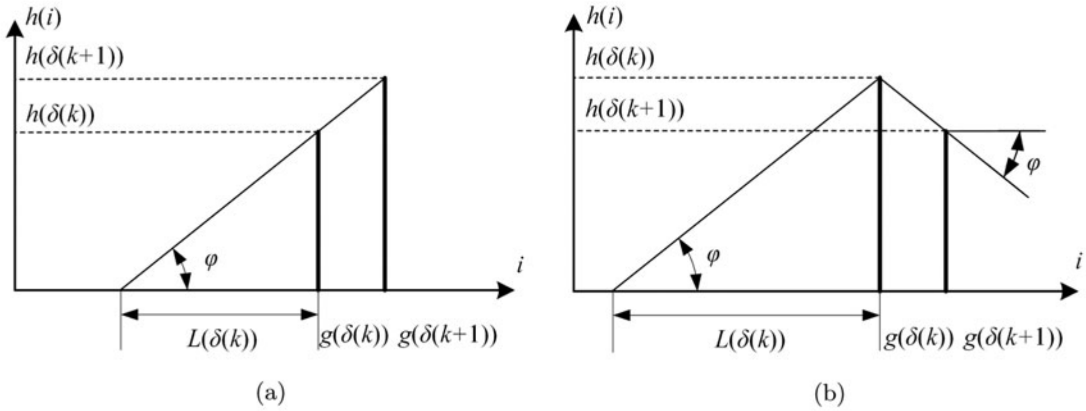

Fig. 13. Offset distance determination: (a) $g(\delta(k+1)) > g(\delta(k))$, and (b) $g(\delta(k)) > g(\delta(k+1))$.

of possible dominant peaks, $1 \leq m \leq p_D$. The offset distance is defined as follows:

$$L(\delta(k)) = \begin{cases} \dfrac{h(\delta(k))[g(\delta(k+1)) - g(\delta(k))]}{|h(\delta(k+1)) - h(\delta(k))|} & \text{if} \quad h(\delta(k+1)) \neq h(\delta(k)) \\ \dfrac{h(\delta(k))[g(\delta(k+1)) - g(\delta(k))]}{|\frac{k+1}{k} h(\delta(k+1)) - h(\delta(k))|} & \text{if} \quad h(\delta(k+1)) = h(\delta(k)). \end{cases} \tag{6}$$

It can be seen that this offset distance $L(\delta(k))$ is determined based on their own pixel number, and the two adjacent intensity levels correspondingly, as illustrated in Fig. 13. Based on the height difference and distance between two adjacent peaks, a higher peak could always be identified from the calculated observing location. In the implementation phase, the observing location is set at $g(\delta(k)) - L(\delta(k))$, hence

$$\alpha(m) = g(\delta(k)) - L(\delta(k)), \tag{7}$$

and the observing location is then considered as the highest in the neighborhood to maintain an unobstructed view. For the kth detected peak in the cluster vector, the crossover index is proposed as follows:

$$\theta(\delta(k)) = \frac{d(\delta(k))}{L(\delta(k))}, \tag{8}$$

where

$$d(\delta(k)) = h(\delta(k)) - \frac{min\{h(\delta(k)), h(\delta(k+1))\}}{2}.$$

Highest peak in the closest distance: A peak is considered as a nearest prominent peak, if it is high enough to be“observed” from the current observing location and to “block” the view to the next peak. This statement could be mathematically described as follows:

$$\begin{cases} \theta(\delta(k)) > \theta(\delta(k+1)) \\ \theta(\delta(k)) > \theta(\delta(k-1)). \end{cases} \tag{9}$$

The searching mechanism of our algorithm is illustrated in Fig. 14. First, $g(\delta(k))$ is identified as a dominant peak. The new location for observation will be updated to $g(\delta(k)) - L(\delta(k))$. The crossover index of $g(\delta(k))$ is reset, while this characteristic is re-calculated for $g(\delta(k+1)), g(\delta(k+2)), \ldots$ based on the updated observing location. A following dominant peak is then determined at $g(\delta(k+3))$. Similarly, when a new observing location is calculated, the crossover index of the next peak

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

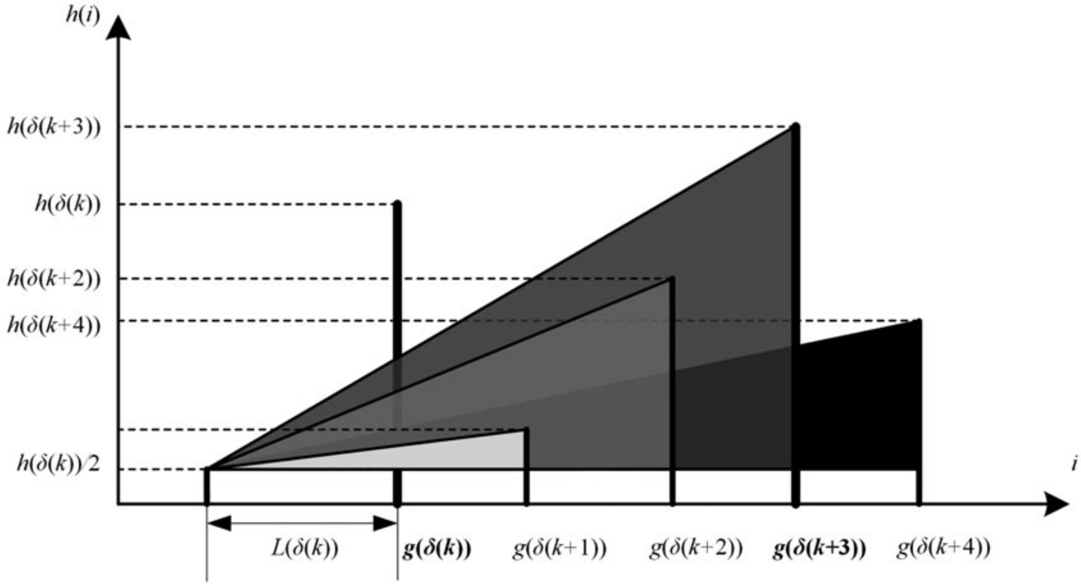

Fig. 14. Illustration of searching mechanism.

is consequently updated according to Eqs. (6)–(8) until the cluster vector is completely checked. Algorithm 2 illustrates the implementation of the proposed algorithm pseudo-code. The identified peaks are then used to partially segment the cracks and rust from the background.

Compared to some recent histogram-based peak detection algorithms,[51,52] the major advantage of the proposed approach is that no prior knowledge or information is required to detect all main peaks of the image histogram. While user-specified parameters are compulsory in methods based on the peak sensitivity and peaks number,[52] the offset distance and crossover index in our technique can be calculated and updated automatically based on the location of detected initial peaks. Using Intel(R) Core(TM) i5-5300U CPU @2.30 GHz with 64 bit Windows 7, the average computation time for detecting peaks of a 8 bit histogram was approximately 12 ms for our algorithm and 15 ms for the peak detection signal method[52] with the main reason being the smaller number of loops used in the searching process.

3.3.2. Line emphasizing. We implement a line emphasis filter, proposed by Sato *et al.*[48] and Fujita *et al.*[49] based on the Hessian matrix to remove blob-like and sheet-like noise and feature the line structure corresponding to a crack. The Hessian matrix of an image $I(\mathbf{x})$, where $\mathbf{x} = (x, y)$ is given by the following:

$$\nabla^2 I(x) = \begin{bmatrix} I_{xx}(\mathbf{x}) & I_{xy}(\mathbf{x}) \\ I_{yx}(\mathbf{x}) & I_{yy}(\mathbf{x}) \end{bmatrix}, \tag{10}$$

where

$$\begin{cases} I_{xx}(\mathbf{x}) = \dfrac{\partial^2}{\partial x^2} I(\mathbf{x}) \\ I_{xy}(\mathbf{x}) = \dfrac{\partial^2}{\partial x \partial y} I(\mathbf{x}) \\ I_{yy}(\mathbf{x}) = \dfrac{\partial^2}{\partial y^2} I(\mathbf{x}) \\ I_{yx}(\mathbf{x}) = \dfrac{\partial^2}{\partial y \partial x} I(\mathbf{x}). \end{cases} \tag{11}$$

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

Algorithm 2: AUTOMATIC PEAK DETECTION.

```
m=1;n=1;found = 0;
for $k$=$n$:length($\delta$) do
    // calculate peak's crossover index;
    for $j$=($n$+1):($k$-1) do
        if there is a dominant peak at level $j$ then
            $n = j + 1$;
            $found = 1$;
            // save the position of dominant peak in $\beta$;
            $\beta(m) = \delta(j)$;
            // update position of observing location;
            $m = m + 1$;
            $\alpha(m) = g(\delta(j)) - L(\delta(j))$;
            // break out of for loop;
            break;

    if found then
        fprintf("Dominant peak at position %d",$j$);
        fprintf("New observe position at %d",$\alpha(m)$);
    else
        fprintf("No dominant peak at this step");
```

Table I. Combination of eigenvalues and corresponding shape structures.

Relationships between eigenvectors	Structure
$\lvert\lambda_1\rvert \geq \lvert\lambda_2\rvert \approx 0$	Line
$\lvert\lambda_1\rvert \approx \lvert\lambda_2\rvert \geq 0$	Blob
$\lvert\lambda_1\rvert \approx \lvert\lambda_2\rvert \approx 0$	Sheet

The eigenvalues $\lambda_1(\mathbf{x})$, $\lambda_2(\mathbf{x})$ of $\nabla^2 I(\mathbf{x})$, where $\lambda_1 \geq \lambda_2$, are adopted to acquire a generalized measure of similarity to a line structure as follows:

$$\lambda_{12} = \begin{cases} |\lambda_2|\left(1 + \dfrac{\lambda_1}{|\lambda_2|}\right) = |\lambda_2| + \lambda_1 & \textbf{if } \lambda_2 \leq \lambda_1 \leq 0 \\ |\lambda_2|\left(1 - \mu\dfrac{\lambda_1}{|\lambda_2|}\right) = |\lambda_2| - \mu\lambda_1 & \textbf{if } \lambda_2 < 0 < \lambda_1 < \dfrac{|\lambda_2|}{\mu} \\ 0 \ \text{ otherwise,} \end{cases} \tag{12}$$

where $0 < \mu \leq 1$.

The line-, blob- or sheet structure in the image could be expressed by combining two eigenvalues λ_1 and λ_2 as described in Table I.

In order to recover line structures of various widths, the partial second derivatives of the image $I(\mathbf{x})$ in Eq. (11) can be combined with the Gaussian convolution, for example,

$$I_{xx}(\mathbf{x}; \sigma_f) = \left\{\frac{\partial^2}{\partial x^2} G(\mathbf{x}; \sigma_f)\right\} * I(\mathbf{x}), \tag{13}$$

where $G(\mathbf{x}; \sigma_f)$ is an isotropic Gaussian function with standard deviation σ_f. The maximum among the normalized multiple scales will be selected from the multi-scale integration of the filter responses of a pixel $\mathbf{x}$ within a region defined by $R(\mathbf{x})$ as follows:

$$R(\mathbf{x}) = \max_{\sigma_i} \sigma_1^2 \lambda_{12}(\mathbf{x}; \sigma_i), \tag{14}$$

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

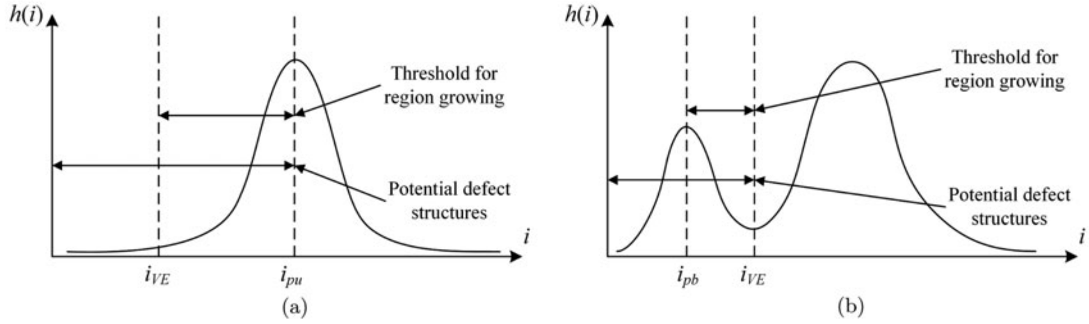

Fig. 15. Threshold for region growing: (a) Unimodal, and (b) Bimodal.

where $\sigma_i = s^{i-1}\sigma_1 (i = 1, 2, \dots, n)$ is a discrete sample of σ_f, with σ_1 and s being the minimum scale and scale factor of the sampling interval of σ_f, respectively.

3.3.3. Enhanced segmentation. The thresholding and emphasizing step can partially extract a defect of steel surfaces from the background of images but may face a difficulty in image segmentation of thin cracks or rusty areas. To overcome this disadvantage, we employ a region growing algorithm based on a dynamic threshold obtained from the Otsu thresholding method[45] and our peak automatic detection approach. The boundary pixels of each extracted abnormal region are identified as the initial seed points. Let g_S denote the mean intensity value of the concerning region and g_N denote the intensity value of any seed point in the neighborhood, a similarity criterion can be simply judged by the following difference:

$$e = |g_S - g_N|. \tag{15}$$

A neighbor pixel can be added in the detected defect region if its intensity difference with the region mean is smaller than a pre-defined threshold. In this paper, the Otsu method in combination with the location of detected peaks is applied to identify the threshold and to determine the valley in the unimodal and bimodal histogram, as illustrated in Fig. 15. The formulation for the valley-emphasis method is

$$t^* = \arg \max_{0 \le 1 < L} \left\{ (1 - p_t)(\omega_1(t)\mu_1^2(t) + \omega_2(t)\mu_2^2(t)) \right\}, \tag{16}$$

where t^* is the optimal value of the threshold t between the foreground class $C_1 = \{0, 1, \dots, t\}$ and the background class $C_2 = \{t + 1, t + 2, \dots, L\}$ of the processed image, in which L is the number of distinct gray levels. The probabilities of the two classes are, respectively,

$$\omega_1(t) = \sum_{i=0}^{t} p_i \quad \text{and} \quad \omega_2(t) = \sum_{i=t+1}^{L} p_i, \tag{17}$$

where $p_i = \frac{h(i)}{n}$ is the probability of occurrence of gray-level i, $\sum_{i=1}^{n} p_i = 1$, and n is the total number of pixels in the concerning image.

Here, by region growing, the detected defect area is expanded to the neighbor pixel by using the similarity criterion via the intensity difference between the histogram peak and the emphasized valley.

4. Experimental Results and Discussion

In order to verify the design and assess the performance of the developed robotic system, both indoor and outdoor experiments were conducted. The indoor test was within the lab environment on small steel bars, while the outdoor experiment was on a real-world steel bridge. The ability of climbing, handling navigation difficulties, and surface condition assessment were evaluated in a number of scenarios. During the tests, one 2S1P (2 cells) 7.4 V 5000 milliampere-hour (mAh) and one 3S1P 11.1 V 5000 mAh batteries are used to power the robot. One laptop that can connect to a wireless

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

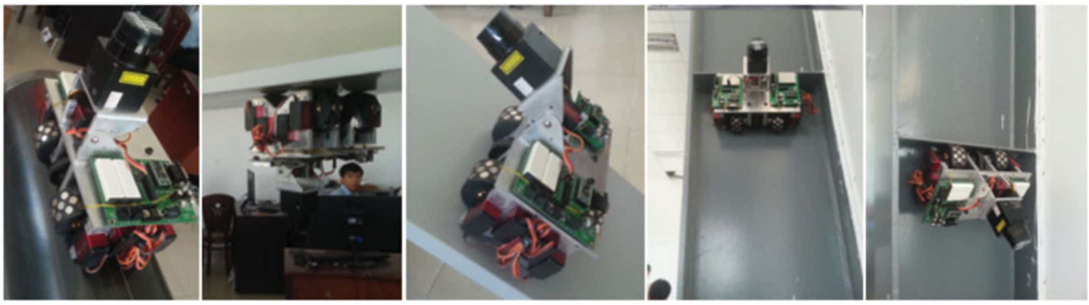

Fig. 16. Adhesion tests on different surfaces under different degrees of inclination.

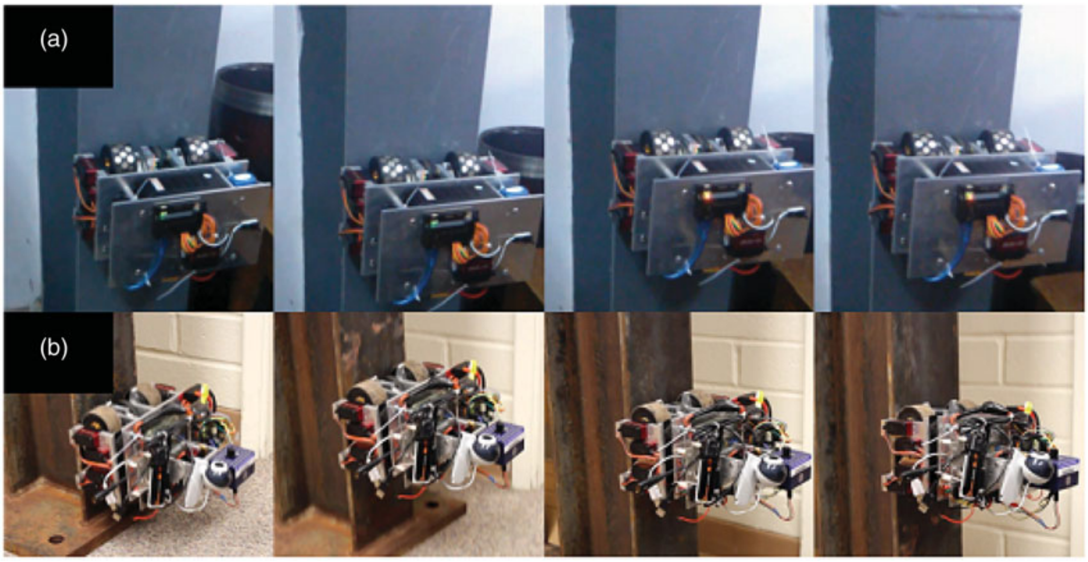

Fig. 17. Robot moves on steel surfaces: (a) Without load, and (b) With full load.

LAN is used as a ground station. The total weight of the robot is about 6 Kg, while the total magnetic force created is approximately 16 Kg, satisfying condition (1).

Figure 16 presents various cases when the robot was placed on steel surfaces under different degrees of inclination to ensure that the magnetic force is strong enough to adhere to steel surfaces when the robot does not move. During the experiments, the climbing capability tests are done on a bridge and on several steel structures nearby with coated or unclean surfaces. Although the surface is curvy, the robots can still adhere tightly to the steel structures. It also shows strong climbing capability on a rusty surface. Moreover, robot is capable of vertically moving, without carrying or with a load, as shown in Fig. 17.

Besides, multiple steel structures are combined to form a model bridge to test the data collection process. After navigating a particular distance, the robot stops to capture images of the surface underneath and send to the ground station. Motion of the robot is controlled remotely from ground station, while the acquired data are presented in Fig. 18. Acquired images are then stitched together to produce an overall image of the steel surface in inspection, as shown in Fig. 19. To this end, results of the steel crack detection algorithm are presented in Fig. 20. It is significant to see that a small crack shown in Fig. 20(a) was detected as can be seen in Fig. 20(b). The combination of the global thresholding and line emphasizing method can partially extract the crack from the image background but the result is not quite obvious as thin structures were not distinguished and hence the segmented image displayed a disconnection, as shown in Fig. 20(c). An improvement of the result is depicted in Fig. 20(d), where the crack is fully extracted from the background, demonstrating the merit of our proposed enhanced segmentation method. Small connected components and isolated pixels can be removed by morphological operations. Out of 231 collected crack images, the success rate of our proposed algorithm was 93.1%, where cracks of more than 3 mm width could be accurately detected.

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

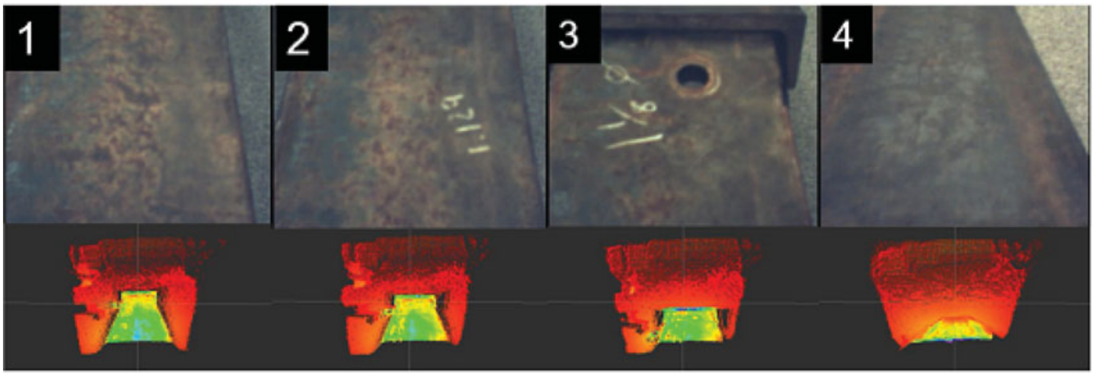

Fig. 18. Visual and 3D images acquired from cameras assist robot navigation and map construction. (Top) Visual image; (Bottom) 3D image of the structure.

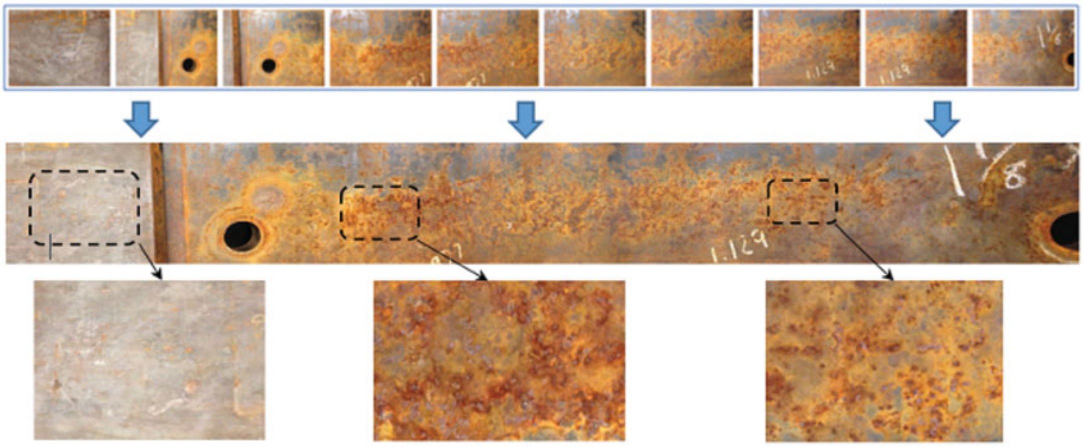

Fig. 19. Images stitching result: (Top) 10 individual images taken by the robot; (Middle) Stitching image result from those 10 individual images; (Bottom) Closer look (zoom-in) at some areas, from left-to-right, showing "good condition", "serious deteriorated condition", and "light deteriorated condition" of the steel surface, respectively.

Only 15 images taken under poor lighting conditions were wrongly classified as the contrast between the crack and the image background is minimized. Indeed, we have compared our approach with two other popular binarization algorithms, the Otsu method and Sauvola–Pietikinen (S–P) method,[53] evaluated in accordance with the following criteria when comparing the segmentation results and the groundtruth:

- The precision index (PI):

$$PI = \frac{TP}{TP + FP}, \tag{18}$$

- The sensitivity index (SI):

$$SI = \frac{TP}{TP + FN}, \tag{19}$$

- The Dice similarity coefficient (DSC):

$$DSC = \frac{2TP}{2TP + FP + FN}, \tag{20}$$

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

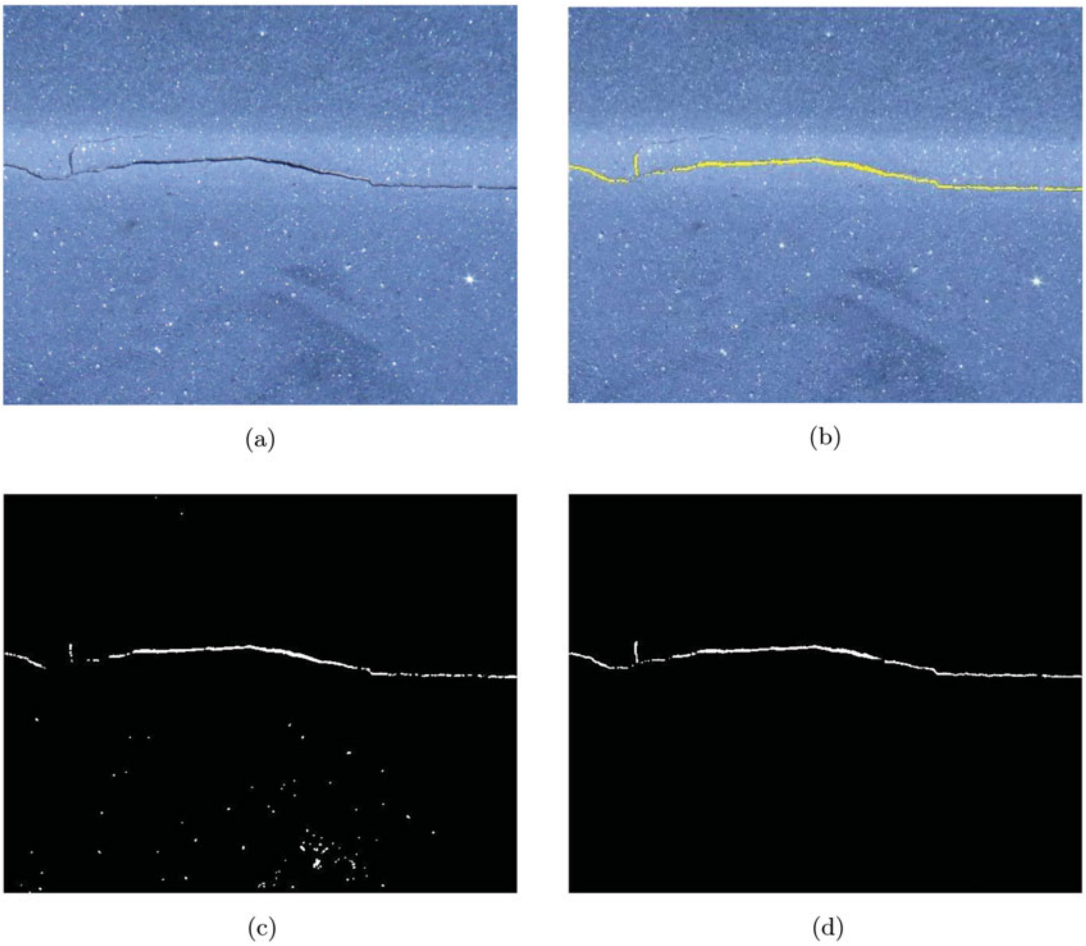
(a)
(b)
(c)
(d)

Fig. 20. Crack detection result: (a) Surface image, (b) Detected crack on surface image, (c) Global thresholding and line emphasizing and (d) Enhanced segmentation.

Table II. Comparison of various methods for crack detection by segmentation.

	PI		SI		DSC	
Method	Normal lighting	Low lighting	Normal lighting	Low lighting	Normal lighting	Low lighting
Proposed	0.9360	0.0079	0.6507	0.7365	0.7677	0.0156
S–P	0.9210	0.7679	0.8555	0.8831	0.8871	0.8215
Otsu	0.0076	0.0089	0.9734	0.8836	0.0151	0.0177

where TP and TN are the correctly-detected pixels corresponding, respectively, to the crack and background objects, FP is the wrongly detected crack pixels and FN is the crack pixels missing in the detection process. Under normal lighting and uniform background conditions as shown in Fig. 21(a), a crack candidate is fully extracted by using our proposed method and the S–P method with only a few small FPs, as shown in Fig. 21(b) and (c), while the crack segmentation using the Otsu method appears to be affected by noise as depicted in Fig. 21(d). Figure 22(a) shows a particular case of low lighting conditions, where the intensity of a part of the background is smaller than that of the crack. In this case, our approach as well as the Otsu method may incorrectly detect the dark part of the background as presented in Fig. 22(b) and (d). For the sake of comparison, Table II summarizes the PI, SI, and DSC of the segmentation results using each method in normal and poor lighting conditions. It can be seen that the proposed approach provides the highest precision level for the crack in the normal lighting condition as shown in Fig. 21(a) with a small number of FP pixels.

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

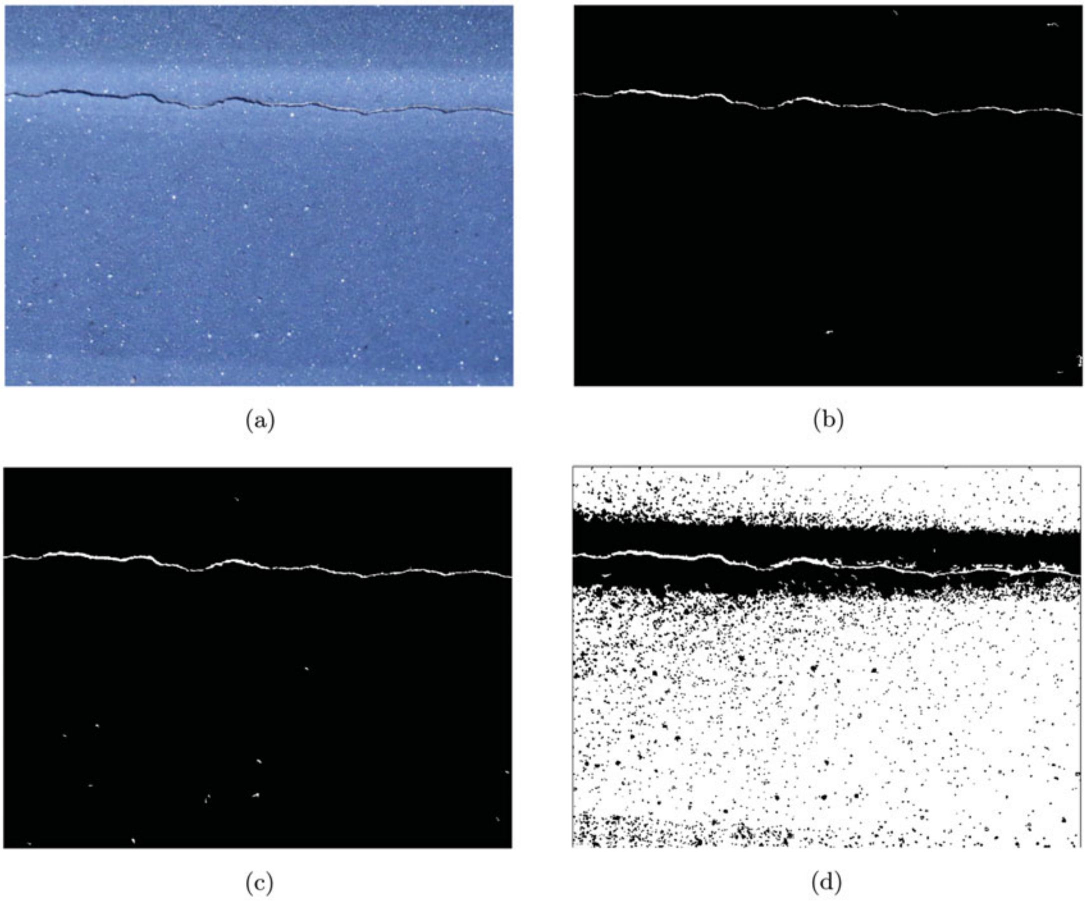

Fig. 21. Crack detection with segmentation in normal lighting conditions: (a) Surface image, (b) Proposed method, (c) S-K method, and (d) Otsu method.

Regarding the 3D construction capability for navigation planning and further processing purposes, the robot can capture 3D images from ToF camera while moving on the steel surface. The data are saved to the on-board computer's hard disk. Then, we apply the registration process that uses the ICP algorithm to align multiple point cloud data. The 3D construction results are presented in Fig. 23.

5. Conclusion

This paper has presented the development of a steel climbing robot and its data processing system for steel infrastructure monitoring. The robot is capable of carrying multiple sensors for its navigation and for steel surface inspection. The initial prototype is implemented and validated to ensure the robot is able to strongly adhere on steel surfaces in different situations. Captured images are stitched together to create a single image of the steel surface then steel crack detection algorithms are implemented to locate a defect on the stitched image. A 3D map is also generated from images captured by the robot. Thus, in addition to various sensors being integrated for navigation and surface inspection, and the collected visual and 3D images can be transferred to the ground station for further monitoring purposes. Further work will include localization using combined odometry and camera data, improvement of the map construction process as well as detection algorithms from stitched images. Additionally, a cooperative process can be used to employ multiple robots for inspecting and monitoring large-scale steel infrastructure. Multi-robots can form an efficient mobile sensor network to perform faster inspection, in which cooperative learning, control, and sensing algorithms[58–62] can be applied.

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

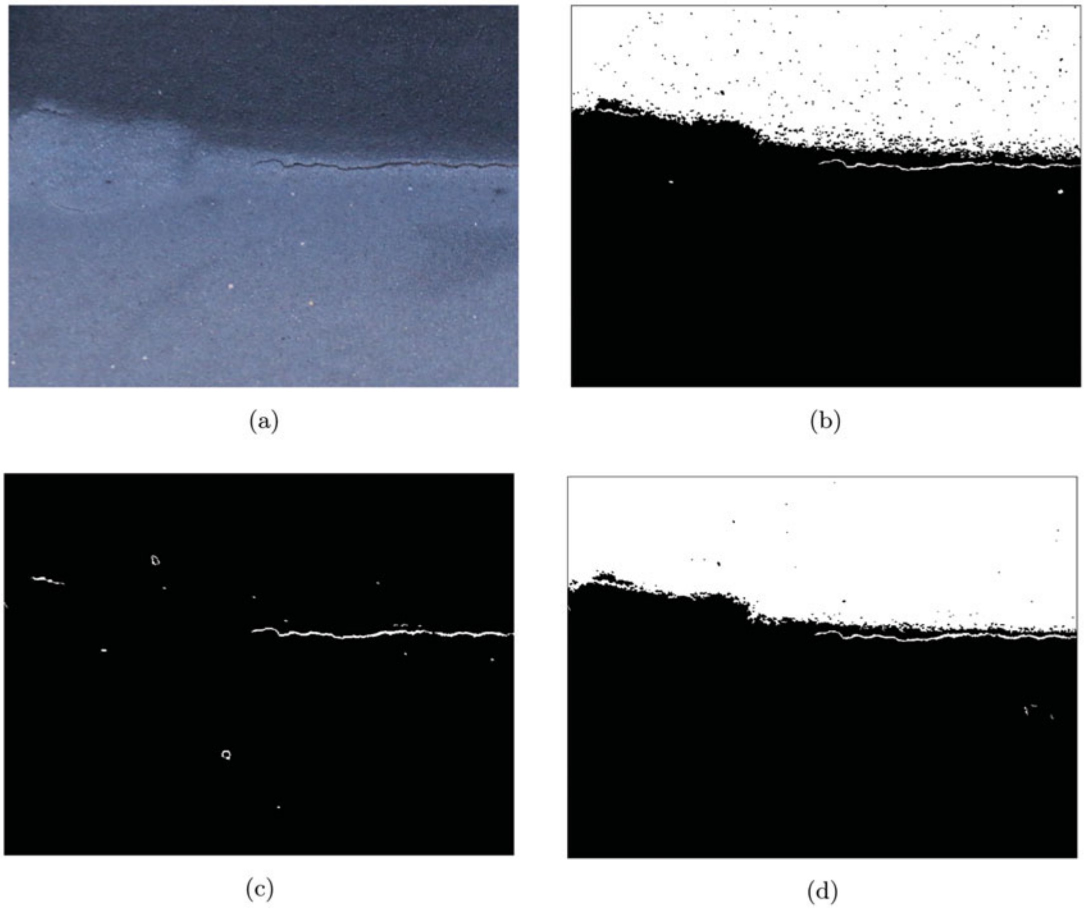

Fig. 22. Crack detection with segmentation in low lighting conditions: (a) Surface image, (b) Proposed method, (c) S–P method, and (d) Otsu method.

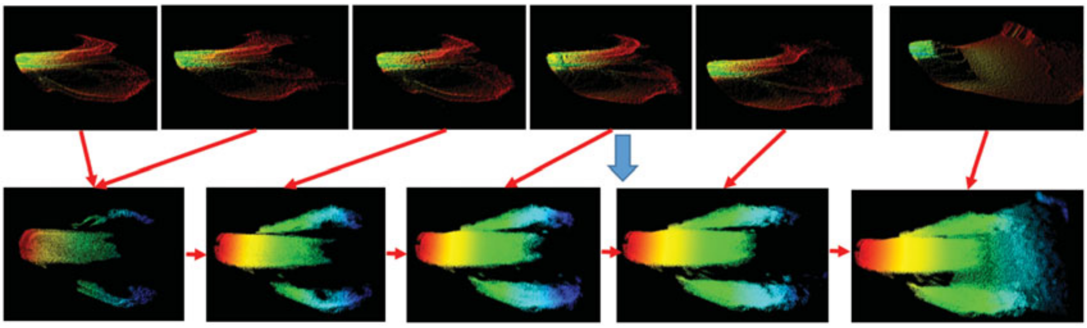

Fig. 23. 3D registration and stitching from point cloud data.

Acknowledgments

This work is supported by the National Science Foundation (NSF) under the grant NSF-IIP #1559942, the National Aeronautics and Space Administration (NASA) under Grant no. NNX15AI02H issued through the Nevada NASA Research Infrastructure Development Seed Grant, and the Nevada NASA Space Grant Consortium; and the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (USDOT/OST-R) under Grant no. 69A3551747126 through INSPIRE University Transportation Center (http://inspire-utc.mst.edu) at Missouri University of Science and Technology. The views, opinions, findings, and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the NASA, NSF, USDOT/OST-R, or any State or other entity. This work is also partially supported by the 2016

Data Arena grant from the University of Technology Sydney. The second author would like to acknowledge support from an Australian Awards scholarship.

References

1. U.S. Department of Transportation Federal Highway Administration, "National bridge inventory data," Available at: http://www.fhwa.dot.gov/bridge/nbi.cfm, accessed January 30, 2016.
2. Minnesota Department of Transportation, "I-35W St. Anthony Falls Bridge collapse," Available at: http://www.dot.state.mn.us/i35wbridge/collapse.html, accessed January 30, 2016.
3. H. M. La, R. S. Lim, B. B. Basily, N. Gucunski, J. Yi, A. Maher, F. A. Romero and H. Parvardeh, "Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation," *IEEE/ASME Trans. Mech.* **18**(6), 1655–1664 (2013).
4. H. M. La, N. Gucunski, S. H. Kee and L. V. Nguyen, "Data analysis and visualization for the bridge deck inspection and evaluation robotic system," *Vis. Eng.* **3**(1), 1–16 (2015).
5. H. M. La, N. Gucunski, S. H. Kee and L. V. Nguyen, "Visual and Acoustic Data Analysis for the Bridge Deck Inspection Robotic System," *Proceedings of the 31st International Symposium on Automation and Robotics in Construction and Mining (ISARC)* (2014) pp. 50–57.
6. H. M. La, N. Gucunski, S. H. Kee J. Yi, T. Senlet and L. V. Nguyen, "Autonomous Robotic System for Bridge Deck Data Collection and Analysis" *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, USA (Sep. 14–18, 2014) pp. 1950–1955.
7. H. M. La, N. Gucunski, K. Dana, and S. H. Kee. "Development of an Autonomous Bridge Deck Inspection Robotic System," *J. Field Robot.* **34**(8), 1489–1504 (2017).
8. R. S. Lim, H. M. La, Z. Shan and W. Sheng, "Developing a Crack Inspection Robot for Bridge Maintenance," *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA)*, (2011) pp. 6288–6293.
9. R. S. Lim, H. M. La and W. Sheng, "A robotic crack inspection and mapping system for bridge deck maintenance," *IEEE Trans. Autom. Sci. Eng.* **11**(2), 367–378 (2014).
10. H. M. La, R. S. Lim, B. B. Basily, N. Gucunski, J. Yi, A. Maher, F. A. Romero and H. Parvardeh, "Autonomous Robotic System for High-Efficiency Non-Destructive Bridge Deck Inspection and Evaluation," *Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE),* Madison, WI, USA (August 17–21, 2013) pp. 1053–1058.
11. B. Li, J. Cao, J. Xiao, X. Zhang and H. Wang, "Robotic Impact-Echo Non-Destructive Evaluation Based On FFT and SVM," *Proceedings of the 11th World Congress on Intelligent Control and Automation (WCICA)* (2014) pp. 2854–2859.
12. N. Gucunski, B. Basily, S-H. Kee, H. M. La, H. Parvardeh, A. Maher and H. Ghasemi, "Multi NDE Technology Condition Assessment of Concrete Bridge Decks by RABITTM Platform," *Proceedings of the NDE/NDT for Structural Materials Technology for Highway & Bridges* (2014) pp. 161–168.
13. N. Gucunski, S.-H. Kee, H. M. La, B. Basily, A. Maher and H. Ghasemi, "Implementation of a Fully Autonomous Platform for Assessment of Concrete Bridge Decks RABIT," *Proceedings of Structures Congress* (2015) pp. 367–378.
14. N. Gucunski, S.-H. Kee, H. M. La, B. Basily and A. Maher, "Delamination and concrete quality assessment of concrete bridge decks using a fully autonomous RABIT platform," *Int. J. Struct. Monit. Maint.* **2**(1), 19–34 (2015).
15. S. Gibb, T. D. Le, H. M. La, R. Schmid and T. Berendsen, "A Multi-Functional Inspection Robot for Civil Infrastructure Evaluation and Maintenance," *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada (Sep. 24–28, 2017).
16. T. D. Le, S. Gibb, N. H. Pham, H. M. La, L. Falk and T. Berendsen, "Autonomous Robotic System using Non-Destructive Evaluation methods for Bridge Deck Inspection," *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 29–June 3, 2017, Singapore.
17. S. Gibb and H. M. La, "Automated Rebar Detection for Ground-Penetrating Radar," *Proceedings of the 12th International Symposium on Visual Computing,* Las Vegas, Nevada, USA (Dec. 12–14, 2016).
18. K. Dinh, N. Gucunski, J. Y. Kim, T. Duong and H. M. La, "Attenuation-based Methodology for Condition Assessment of Concrete Bridge Decks using GPR," *Proceedings of the 32nd International Symposium on Automation and Robotics in Construction and Mining (ISARC)*, Oulu, Finland, Jun. 15–18, 2015.
19. N. Gucunski, A. Maher, B. B. Basily, H. M. La, R. S. Lim, H. Parvardeh and S. H. Kee, "Robotic Platform RABIT for Condition Assessment of Concrete Bridge Decks Using Multiple NDE Technologies," *J. Croatian Soc. Non Destr. Testing* **12**, 5–12 (2013).
20. P. Prasanna, K. J. Dana, N. Gucunski, B. B. Basily, H. M. La, R. S. Lim and H. Parvardeh, "Automated crack detection on concrete bridges," *IEEE Trans. Autom. Sci. Eng.* **13**(2), 591–599 (Apr. 2016).
21. F. Xu and X. Wang, "Design and Experiments on A New Wheel-Based Cable Climbing Robot," *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (2008) pp. 418–423.
22. 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," *IEEE/ASME Trans. Mech.* **18**(6), 1665–1674 (2013).
23. A. Mazumdar and H. H. Asada, "Mag-Foot: A Steel Bridge Inspection Robot," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2009) pp. 1691–1696.

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

24. R. Wang and Y. Kawamura, "A Magnetic Climbing Robot for Steel Bridge Inspection," *Proceedings of the 11th World Congress on Intelligent Control and Automation (WCICA)* (2014) pp. 3303–3308.
25. Q. Liu and Y. Liu, "An Approach for Auto Bridge Inspection Based On Climbing Robot," *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)* (2013) pp. 2581–2586.
26. Y. Liu, Q. Dai and Q. Liu, "Adhesion-Adaptive Control of a Novel Bridge-Climbing Robot," *Proceedings of the IEEE 3rd Annual International Conference on Cyber Technology in Automation, Control and Intelligent Systems (CYBER)* (2013) pp. 102–107.
27. A. Leibbrandt, G. Caprari, U. Angst, R. Y. Siegwart, R. J. Flatt and B. Elsener, "Climbing Robot for Corrosion Monitoring of Reinforced Concrete Structures," *Proceedings of the 2nd International Conference on Applied Robotics for the Power Industry (CARPI)* (2012) pp. 10–15.
28. H. Leon-Rodriguez, S. Hussain and T. Sattar, "A Compact Wall-Climbing and Surface Adaptation Robot for Non-Destructive Testing," *Proceedings of the 12th International Conference on Control, Automation and Systems (ICCAS)* (2012) pp. 404–409.
29. A. San-Millan, "Design of a Teleoperated Wall Climbing Robot for Oil Tank Inspection," *Proceedings of the 23th Mediterranean Conference on Control and Automation (MED)* (2015) pp. 255–261.
30. 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," *IEEE/ASME Trans. Mech.* **17**(3), 499–507 (2012).
31. C. Koch, K. Georgieva, V. Kasireddy, B. Akinci and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Adv. Eng. Inform.* **29**(2), 196–210 (2015).
32. J. K. Oh *et al.*, "Bridge inspection robot system with machine vision," *Autom. Constr.* **18**, 929–941 (2009).
33. R. S. Adhikari, O. Moselhi and A. Bagchi, "Image-based retrieval of concrete crack properties for bridge inspection," *Autom. Constr.* **39**, 180–194 (2014).
34. R. G. Lins and S. N. Givigi, "Automatic Crack Detection and Measurement Based on Image Analysis," *IEEE Trans. Instrum. Meas.* **65**(3), 583–590, (2016).
35. K&J Magnetics Inc. Orignial magnet calculator, Available a: https://www.kjmagnetics.com/calculator.asp, access November 10, 2016.
36. DS8911HV Servo, Available at: http://www.jramericas.com/116293/JRPS8911HV/, access November 10, 2016.
37. D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, 2nd Edition, Nov 5, 2011 (Prentice Hall, Upper Saddle River, NJ).
38. M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vis.* **74**(1), 59–73 (2007).
39. P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (1992).
40. R. B. Rusu, N. Blodow and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA '09,* (2009) pp. 3212–3217.
41. Y. Chen and G. Medioni, "Object Modelling by Registration of Multiple Range Images," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3 (2015) pp. 2724–2729.
42. D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu and S. Behnke, "Registration with the point cloud library: A modular framework for aligning in 3-D," *IEEE Robot. Autom. Mag.* **2**(4), 110–124 (2015).
43. Y.-H. Yu, N. M. Kwok and Q. P. Ha, "Color tracking for multiple robot control using a system-on-programmable-chip," *Autom. Constr.* **20**(6), 669–676 (2011).
44. T. H. Dinh, M. T. Pham, M. D. Phung, D. M. Nguyen and Q. V. Tran, "Image Segmentation Based on Histogram of Depth and an Application in Driver Distraction Detection," *Proceedings of the 13th International Conference on Control Automation Robotics & Vision (ICARCV)* (2014) pp. 969–974.
45. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979).
46. A. Dirami, K. Hammouche, M. Diaf and P. Siarry, "Fast multilevel thresholding for image segmentation through a multiphase level set method," *Signal Proces.*, **93**(1), 139–153 (2013).
47. X. C. Yuan, L. S. Wu and Q. Peng, "An improved Otsu method using the weighted object variance for defect detection," *Appl. Surface Sci.* **349**, 472–484 (2015).
48. Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig and R. Kikinis, "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images," *Medical Image Anal.* **2**(2), 143–168 (1998).
49. Y. Fujita and Y. Hamamoto, "A robust automatic crack detection method from noisy concrete surfaces," *Mach. Vis. Appl.* **22**(2), 245–254 (2011).
50. Y. Q. Zhao, X. H. Wang, X. F. Wang and F. Y. Shih, "Retinal vessels segmentation based on level set and region growing," *Pattern Recognit.* **47**(7), 2437–2446 (2014).
51. M. I. Sezan, "A peak detection algorithm and its application to histogram-based image data reduction." *Comput. Vis. Graph. Image Process.* **49**(1), 36–51 (1990).
52. B. Yuan and M. Liu, "Power histogram for circle detection on images," *Pattern Recognit.* **48** (10) 3268–3280 (2015).

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press

53. J. Sauvola and M. Pietikäinen, “Adaptive document image binarization,” *Pattern Recognit.* **33**(2), 225–236 (2000).
54. N. H. Pham and H. M. La, “Design and Implementation of an Autonomous Robot for Steel Bridge Inspection,” *Proceedings of the 54th Annual Allerton Conference on Communication, Control, and Computing*, Urbana-Champaign, Illinois, USA (Sep. 27–30, 2016) pp. 556–562.
55. L. V. Nguyen, H. M. La, J. Sanchez and T. Vu, “A Smart Shoe for Building a Real-Time 3D Map,” *Elsevier J. Autom. Constr.* **71**, 2–12 (Sep. 2016).
56. N. H. Pham, H. M. La, Q. P. Ha, S. N. Dang, A. H. Vo and Q. H. Dinh, “Visual and 3D mapping for steel bridge inspection using a climbing robot,” *Proceedings of the 33rd International Symposium on Automation and Robotics in Construction and Mining (ISARC),* Auburn, Alabama, USA (July 18–21, 2016).
57. T. H. Dinh, Q. P. Ha and H. M. La, “Computer vision-based method for concrete crack detection,” *Proceedings of the 14th International Conference on Control, Automation, Robotics and Vision (ICARCV),* Phuket, Thailand (Nov. 13–15, 2016).
58. H. M. La, W. Sheng and J. Chen, “Cooperative and active sensing in mobile sensor networks for scalar field mapping,” *IEEE Trans. Syst. Man Cybern.: Syst.* **45**(1), 1–12 (Jan. 2015).
59. H. M. La, R. Lim and W. Sheng, “Multi-robot cooperative learning for predator avoidance,” *IEEE Trans. Control Syst. Technol.* **23**(1), 52–63 (Jan. 2015).
60. H. M. La and W. Sheng, “Multi-agent motion control in cluttered and noisy environments,” *J. Commun.* **8**(1), 32–46 (Jan. 2013).
61. H. M. La and W. Sheng, “Dynamic targets tracking and observing in a mobile sensor network,” *Elsevier J. Robot. Autom. Syst.* **60**(7), 996–1009 (Jul. 2012).
62. H. M. La and W. Sheng, “Distributed sensor fusion for scalar field mapping using mobile sensor networks,” *IEEE Trans. Cybern.* **43**(2), 766–778 (April, 2013).

https://doi.org/10.1017/S0263574717000601 Published online by Cambridge University Press