

Developing a Crack Inspection Robot for Bridge Maintenance

Ronny Salim Lim, Hung Manh La, Zeyong Shan and Weihua Sheng

Abstract— One of the important tasks for bridge maintenance is bridge deck crack inspection. Traditionally, a human inspector detects cracks using his/her eyes and finds the location of cracks manually. Thus the accuracy of the inspection result is low due to the subjective nature of human judgement. We propose a system that uses a mobile robot to conduct the inspection, where the robot collects bridge deck images with a high resolution camera. In this method, the Laplacian of Gaussian algorithm is used to detect cracks and the global crack map is obtained through camera calibration and robot localization. To ensure that the robot collects all the images on the bridge deck, we develop a complete coverage path planning algorithm for the mobile robot. We compare it with other path planning strategies. Finally, we validate our proposed system through experiments and simulation.

I. INTRODUCTION

A. Motivation

For many engineered transportation structures, including civil, mechanical and aerospace structures, timely awareness of their structural health can prevent functional failures which may lead to catastrophic consequences. On August 1, 2007, the collapse of the I-35W Mississippi River Bridge that carried Interstate 35W across the Mississippi River in Minneapolis left 13 dead and more than 100 injured [1], not to mention its big impact on the traffic and business in the surrounding areas. This accident has clearly demonstrated the catastrophic results of structural failures and the importance of timely awareness of structural health. Bridge decks are typically the elements of first maintenance on a bridge. Since the surface of a bridge deck is exposed to the environment, direct loading from vehicles and exposure to deicing chemicals, constant maintenance is a must. Therefore bridge deck inspection is helpful and can provide owners warning to the future deterioration of the bridge deck. Currently bridge decks are inspected with very rudimentary methods in the form of visual inspection by a trained engineer. While these inspections are useful, they are only useful if the deterioration of the bridge deck is significant. The inspectors usually walk though the bridges and measure the crack locations and crack sizes. This kind of manual approach has several disadvantages. First, it is prone to human errors. Second, it has very limited accuracy due to the limited visual capability of human inspectors. Third, it cannot guarantee the full coverage of the whole bridge deck. Additionally, conducting visual crack inspection of a bridge deck is a dangerous job with passing traffic. Therefore, we need a robotic crack

Ronny Salim Lim, Hung Manh La and Weihua Sheng are with the school of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA e-mail: (ronny.lim, hung.la, weihua.sheng)@okstate.edu.

Zeyong Shan is with the Safety Vision, Houston, TX 77041, USA e-mail: zeyong.shan@gmail.com.

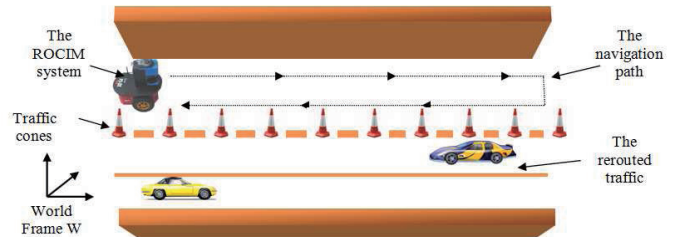


Fig. 1. The scenario of crack inspection and mapping using the ROCIM system.

inspection and mapping (ROCIM) system which can conduct accurate assessment of cracking on bridge decks.

B. Literature review

Recent years have witnessed growing research interests in structural health monitoring (SHM) [2],[3],[4],[5],[6] for bridges, buildings and other civil infrastructures. Research in structure inspection using robotic devices has resulted in several prototypes. Yu *et al.* [7] presented an auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. In their system an illuminator is used to distinguish the crack and non-crack areas. Sinha *et al.* [8] developed a statistical filter for crack detection in pipes. Tung *et al.* [9] proposed the development of a mobile manipulator imaging system for bridge crack inspection. The manipulator system is equipped with a binocular charge coupled devices (CCD) camera. Their system aims to replace human inspectors. Lee *et al.* [10] and Oh *et al.* [11] proposed a bridge inspection system which consists of a specially designed car, a robotic mechanism and a control system for automatic crack detection. Sohn *et al.* [12] monitored cracks changes in concrete structure. Their system focuses on quantifying the change of cracks from multi temporal images during the monitoring period. Ito *et al.* [13] demonstrated an automatic measurement system for the concrete block inspection by means of fine crack extraction. Most of these studies classify, measure, and detect cracks. None of these crack inspection studies finds the global location of cracks. In this paper, we develop an overall system for crack inspection. We detect cracks using a high resolution camera. Then we find the location of cracks in the global map. To ensure that the mobile robot takes all images, we also propose a new strategy for complete coverage path planning.

The remainder of this paper is organized as follows. In Section II, we discuss the overview of the ROCIM system. Section III presents the solution to camera calibration. The crack detection is described in Section IV. The complete coverage path planning is presented in Section V. Section VI

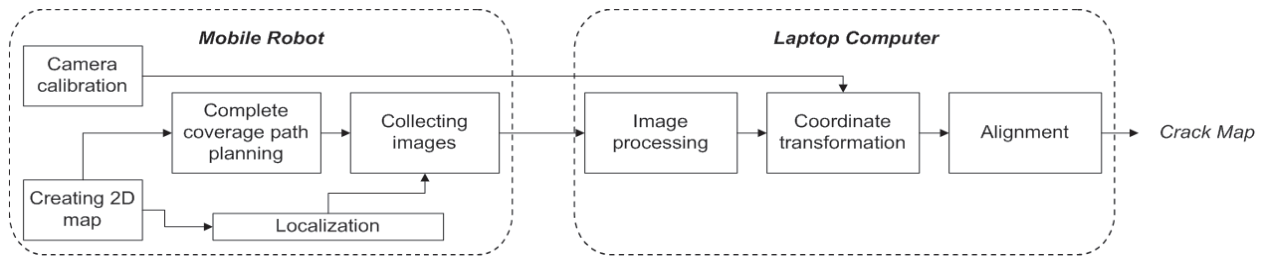


Fig. 2. The principle of the ROCIM system.

provides the experiment results. Finally, Section VII proposes some ideas for future work and gives the conclusions.

II. ROCIM SYSTEM OVERVIEW

Figure 3 shows the hardware setup of the ROCIM system, which includes:

- one Pioneer3-DX mobile robot
- one Hitachi KP-F83GV CCD camera and
- one LMS-200 laser sensor

We mount the high resolution Hitachi camera using a tripod on the robot body. The resolution of this camera is 1024x768 (XGA). The laser sensor is used to create the 2D map of the bridge deck and localize the robot.

To conduct bridge deck crack inspection and mapping, as illustrated in Figure 1, we will block half of the bridge. The ROCIM system will then be deployed to inspect the blocked half of the bridge. Once completed, the traffic will be switched to the completed half and the other half will be inspected. There are three steps in the crack inspection and mapping.

- 1) Navigation map building: A 2D bridge deck map will be created first, which will be used to localize the robot during the data collection step.
- 2) Data collection: The robot will navigate on the bridge deck to collect the image data at different locations. The raw image data can be stored in the on-board computer or transferred to a nearby laptop computer using wireless connection.
- 3) Crack detection and crack map generation: Cracks will be detected through image processing. The global crack map will be created by piecing together multiple local crack maps. This step can be done off-line.

The proposed crack inspection system works according to Figure 2. After the mobile robot creates the bridge deck map through the simultaneous localization and mapping (SLAM) algorithm [14], the robot can later on localize itself based on that map using Monte Carlo Localization (MCL) algorithm [15]. To ensure that the robot collects all the images on the bridge deck, we propose a unique complete coverage path planning strategy. When all images are collected, they will be processed using the Laplacian of Gaussian (LoG) algorithm [16] to detect cracks. In order to create the crack map, we convert the crack locations into the global coordinate system. In the last part of the ROCIM system, there is an alignment process to reduce the location errors. In this paper, we mainly

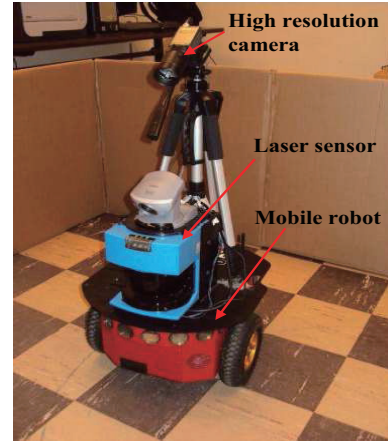


Fig. 3. The mobile robot of the ROCIM system.

focus on developing the overall solution for crack detection and the path planning strategy for the ROCIM system.

III. CAMERA CALIBRATION

The purpose of the camera calibration is to find the extrinsic and the intrinsic parameters of the camera with respect to the mobile robot. Subsequently, we use those parameters to map all crack locations to the 2D global map. The camera calibration involves three coordinate systems, as shown in Figure 4: camera coordinate system (F_C), robot coordinate system (F_R) and global coordinate system (F_G). We use the Camera Calibration Matlab Toolbox to conduct the calibration [17]. After we finish the calibration, we have the intrinsic and the extrinsic parameters. The intrinsic parameters are focal length (f), skew value (s) and origin of image coordinate system (μ_0, ν_0) as in Equation (1). The extrinsic parameters are rotation (\mathbf{R}) and translation (\mathbf{t}) matrices as in Equation (2) [18].

$$\mathbf{P} = \begin{bmatrix} sf & 0 & \mu_0 & 0 \\ 0 & f & \nu_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (1)$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2)$$

In the ROCIM system, when the robot takes an image, it saves the current location. After cracks in each image are detected, we can plot all the crack locations in the global coordinate system. To do this, first, we have the

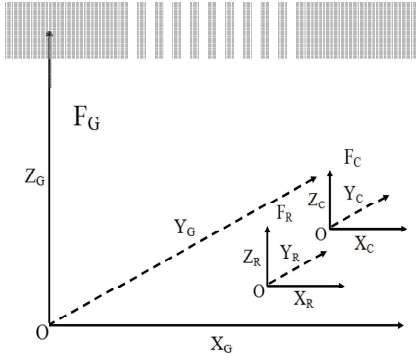


Fig. 4. The coordinate systems in the ROCIM system.

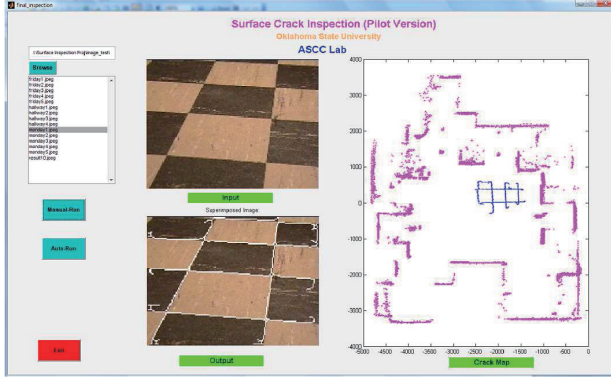


Fig. 5. Graphical user interface for crack detection and crack mapping.

crack locations in the image coordinate system (X_C, Y_C) . Then, we map the crack locations to the robot coordinate system (X_R, Y_R) using the camera-robot transformation ${}^R T_C$, which is derived from the camera calibration parameters. Second, we apply another 2D transformation from the robot coordinate system to the global coordinate system using the transformation ${}^G T_R$, which can be calculated from the robot location (x, y, θ) . Therefore from the camera coordinate system to the global coordinate system, we have the following transformation:

$${}^G T_C = {}^G T_R {}^R T_C \quad (3)$$

IV. CRACK DETECTION

In this section, we describe how to detect cracks in each image captured by the mobile robot. The main idea of detecting cracks is to find out the edge points in an image. These edge points can be detected by finding the zero crossings of the second derivative of the image intensity. However, calculating second derivative is very sensitive to noise. Hence, this noise should be filtered out. To achieve this, the Laplacian of Gaussian (LoG) algorithm is used. This method combines Gaussian filtering with the Laplacian for edge detection. In the LoG edge detection, there are mainly three phases: filtering, enhancement, and detection. Especially, Gaussian filter is used for smoothing and its second derivative is used for enhancement. The detection criteria is the presence of a zero crossing in the second derivative with the corresponding large peak in the first derivative. The Laplacian is a 2-D isotropic measure of

the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of fast intensity change and is often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian smoothing filter in order to reduce its sensitivity to noise.

To find the places where the value of the Laplacian passes through zero points and also changes sign, we use the zero crossing detector. We know that such points often occur at edges where the intensity of the image changes rapidly, but they also occur at places that are not easy to associate with edges. It is best to think of the zero crossing detector as some sort of feature detector rather than as a specific edge detector. Zero crossings always lie on closed contours, and so the output from the zero crossing detector is usually a binary image with single-pixel lines showing the positions of the zero crossing points.

The initial input for the zero crossing detector is an image which has been filtered using the Laplacian of Gaussian filter. The resulting zero crossings are strongly affected by the size of the Gaussian used for the smoothing stage of this operator. As the smoothing is increased, then fewer and fewer zero crossing contours will be found, and those that remain will correspond to features of larger and larger scale in the image. The summary of the crack detection algorithm is shown in Algorithm 1. We developed the operation interface for crack

Algorithm 1: Crack Detection

- Step 1.** Apply the LoG to the image.
 - Step 2.** Apply the zero-crossing detector in the image.
 - Step 3.** Filter the zero-crossings to keep only those strong ones (large difference between the positive maximum and the negative minimum).
 - Step 4.** Suppress the weak zero-crossings most likely caused by noise.
-

detection by using a graphical user interface (GUI) shown in Figure 5. In this figure, the camera images are shown on the left and the global crack map is shown on the right.

V. COMPLETE COVERAGE PATH PLANNING

The purpose of complete coverage path planning is to ensure that the mobile robot collects all the images on the bridge deck in an efficient way. Most of the research work on complete coverage path planning use the robot body or a disk centered around the robot to cover the entire area [19] [20] [21]. However, in our complete coverage path planning, we have to use a camera field of view instead of the robot body. There is an offset distance between the mobile robot and the camera field of view as can be seen in Figure 6. This offset distance is determined by the way the camera is mounted on the robot.

In order to detect very small cracks, the robot need to zoom in the camera, which makes the size of the camera view become smaller than the size of the mobile robot. We model this problem as in Figure 6, the red/darker square is

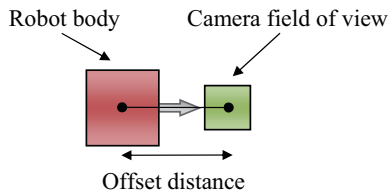


Fig. 6. The camera field of view of the ROCIM system.

the robot and the green/lighter square is the camera field of view. We assume the use of a PTZ (Pan-Tilt-Zoom) camera and the pan angle of the camera is changeable but limited to three values: $[-45^0, 0^0, 45^0]$. The tilt angle of the camera is fixed. In our path planning problem, we have to meet two requirements: First, the union of camera field of views should completely cover the bridge deck. Second, it is desirable to minimize the number of robot turns, traveling distance and inspection time. The coverage path planning problem is related to the covering salesman problem [22] which is an NP-hard problem.

Our path planning method is based on simple motion templates, which are easy to implement on real robots. We first plan the motion of the camera field of view to guarantee full coverage. Then based on the camera path we generate the robot path. This method is called *robot path planning based on camera path (RPP-CP)*. The main steps of the RPP-CP algorithm is shown in Algorithm 2. In Figure 7, the motion templates (a), (b), and (c) are used to capture the image and the motion template (d) is used to turn the robot.

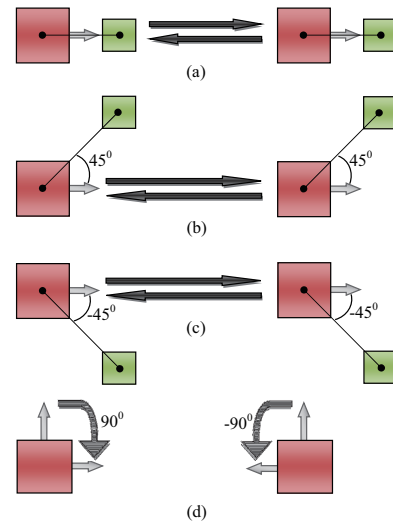


Fig. 7. (a), (b), and (c): the motion templates of moving forward or backward with 0^0 , 45^0 , and -45^0 camera pan angle, respectively. (d): the motion templates of turning the robot 90^0 and -90^0 .

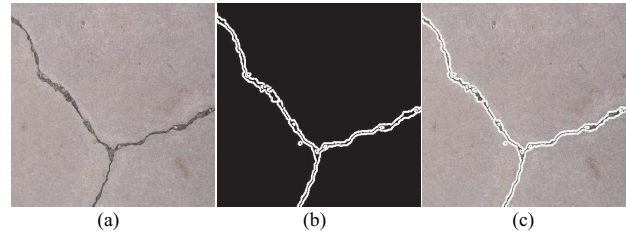


Fig. 8. The crack detection result on the real bridge deck.

Algorithm 2: RPP-CP

Step 1. Generate the camera path to cover the entire area. For example, using boustrophedon motion [23] to generate the camera path.

Step 2. For each camera path, select the motion template to get the appropriate camera field of view. The motion templates can be seen in Figure 7.

Step 3. Optimize the robot path planning. For each camera path, there are more than one motion template that can be used. However, in order to optimize the path planning, the system should keep the current motion template as long as it is still the solution.

VI. EXPERIMENTAL RESULTS

We conduct both experiments and simulations to validate the proposed ROCIM system and the associated algorithms. In the first experiment, we detect a real crack on a bridge deck. In the second experiment, we demonstrate the overall working of the ROCIM system in a laboratory environment. In the simulation, we evaluate the complete coverage path planning algorithm.

A. Crack Detection

In order to evaluate the performance of the ROCIM system, we first evaluate our crack detection algorithm

using real cracks collected from a bridge deck. The original image is shown in Figure 8 (a). Then the LoG algorithm is applied, and the results are shown in Figure 8 (b) and the superimposed crack and image are shown in Figure 8 (c). As can be seen from the result, the LoG algorithm successfully detects cracks in the image. Our initial test indicates that the smallest crack that can be detected is 1 mm.

B. The working of the ROCIM system

In the second experiment, we demonstrate the working of the ROCIM system in our laboratory. Due to the lack

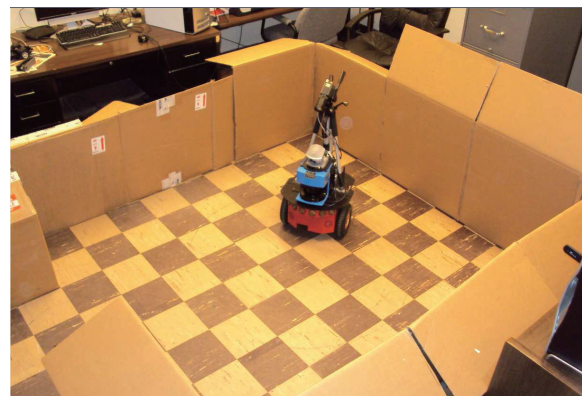


Fig. 9. Experiment setup for the overall test of the ROCIM system.

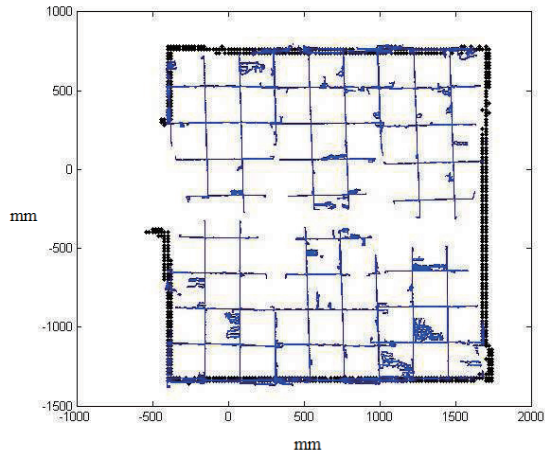


Fig. 10. The crack map obtained for the lab environment.

of cracks in our lab environment, we take the grids on the lab floor as cracks. The hardware setup for the experiment is shown in Figure 9. First, we use the Advanced Range Navigation Laser (ARNL) and the Mobile Eyes software from Mobile robot Inc. [24] to create the 2D map of the environment. Later, the 2D map is used for the mobile robot to localize itself during the inspection. In camera calibration, the intrinsic and extrinsic parameters of the camera with respect to the robot coordinate frame are obtained. There are six images collected from different places in the experimental area as shown in Figure 9. To avoid the blurring in the images, the robot has to stop when capturing the images. Then, the LoG algorithm is applied to detect cracks in each captured image. Next, we map all these local crack locations to the global map. An overall crack map based on these images is shown in Figure 10. The blue lines represent cracks and the black (thicker) lines represent the 2D global map of the environment. The current result still has limited accuracy due to the accuracy of the robot localization and the lack of the alignment process, which will be addressed in our future work.

C. Complete Coverage Path Planning

In this part, we evaluate the performance of the RPP-CP algorithm with two different types of camera path motion: boustrophedon and wall-following. Moreover, we also compare them with another algorithm without using RPP-CP. These evaluations are conducted through computer simulation. Overall, the three algorithms we considered are:

- Type I: Boustrophedon RPP-CP. The camera path is generated by the boustrophedon motion.
- Type II: Wall-following RPP-CP. The camera path is generated by a wall-following motion.
- Type III: Without RPP-CP. It simply applies the boustrophedon and zigzag motion to robot path planning without considering camera path.

For type I and II which use the RPP-CP algorithm, the pan angle of camera has three choices -45^0 , 0^0 and 45^0 . In type III, we assume the pan angle of camera is fixed at 0^0 .

TABLE I

PERFORMANCE COMPARISON OF THREE PATH PLANNING ALGORITHMS

| Type | Number of Robot Turns | Distance | Time |
|------|-----------------------|----------|----------|
| I | 54 | 2340 | 152.8 s |
| II | 89 | 3090 | 226.57 s |
| III | 131 | 3800 | 308.8 s |

We consider a square environment for simplicity and the environment is discretized into cells which have the same size with camera field of view. For type I planning algorithm, the mobile robot starts at location $(-50, 25)$ and the motion template is used as in Figure 11 (a). The first motion template is moving backward with -45^0 pan angle until the robot arrives at location $(50, -25)$. After that the robot turns 180^0 and moves backward with 45^0 pan angle to cover the left over area in the first row. At location $(25, 25)$ the robot turns 90^0 and moves forward to start collecting the camera field of view for the next row. The robot keep doing this until the robot arrives at location $(50, -25)$, at this time the coverage of camera field of view is achieved more than half of the entire environment. Then another boustrophedon motion for the camera path is applied from bottom to top. Finally, the mobile robot collects all the images and finishes at $(-50, 15)$. The trajectory of this motion planning is shown in Figure 11 (a). The idea of type II is similar to type I; however, the camera path is generated by a wall-following motion. The trajectory of this motion planning is shown in Figure 11-b. For type III, the mobile robot starts at location $(-50, 30)$ and finishes at location $(-50, -10)$. The trajectory of the robot is shown in Figure 11 (c). The boustrophedon motion planning is applied to the robot path planning from top to bottom and bottom to top. After that, the zig-zag motion is used to cover the leftover area. In Table I, we show the performance of the three different path planning algorithms, which clearly indicates that the type I has the best performance. To calculate the total time, we use the following equation:

$$Time = \alpha N + \beta d$$

where N is the number of robot turns and d is the traveling distance. The weight values (α, β) in this equation are determined by experimentally measuring the actual turning time and traveling speed of the Pioneer robot. Here we use the following values: $\alpha = 1.4$ and $\beta = 0.033$.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a robotic crack inspection and mapping (ROCIM) system. The ROCIM system provides an overall solution to bridge deck crack inspection. Moreover, we propose robot path planning based on the camera path (RPP-CP) to ensure the mobile robot collects all the images efficiently. Through the experiment and simulation evaluation, the crack detection algorithm works well for real cracks. The RPP-CP algorithm proves to be better than other algorithms in terms of robot turns, traveling distance and inspection time. In the future work, we will research the

VIII. ACKNOWLEDGMENT

This project is partially supported by NSF grant CISE/CNS MRI 0923238 and Oklahoma Transportation Center grant OTCREOS10.1-43.

REFERENCES

- [1] Wikipedia. http://en.wikipedia.org/wiki/i-35w_mississippi_river_bridge. 2007.
- [2] C. R. Farrar, H. Sohn, and S. W. Doebling. Structural health monitoring at los alamos national laboratory. *The 13th International Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management (COMADEM 2000)*, 2000.
- [3] H. Sohn, C. R. Farrar, M. L. Fugate, , and J. J. Czarnecki. Structural health monitoring of welded connections. *The First International Conference on Steel and Composite Structures*, 2001.
- [4] E. Sazonov, K. Janoyan, and R. Jha. Wireless intelligent sensor network for autonomous structural health monitoring. *Proceedings of the SPIE - The International Society for Optical Engineering*, 5384(1):305–314, 2004.
- [5] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. In *SenSys'04*, 2004.
- [6] D. R. Huston. Adaptive sensors and sensor networks for structural health monitoring. *Proceedings of SPIE - The International Society for Optical Engineering*, 4512:203–211, 2001.
- [7] S. N. Yu, J. H. Jang, and C. S. Han. Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. *Automation in Construction*, 16:255–261, 2007.
- [8] S. K. Sinha and P. W. Fieguth. Automated detection of cracks in buried concrete pipe images. *Automation in Construction*, 15:58–72, 2006.
- [9] P. C. Tung, Y. R. Hwang, and M. C. Wu. The development of a mobile manipulator imaging system for bridge crack inspection. *Automation in Construction*, 11:717–729, 2002.
- [10] J. H. Lee, J. M. Lee, J. W. Park, and Y. S. Moon. Efficient algorithms for automatic detection of cracks on a concrete bridge. *The 23rd International Technical Conference on Circuits/Systems, Computers and Communications*, pages 1213–1216, 2008.
- [11] J. K. Oh, G. Jang, S. Oh, J. H. Lee, B. J. Yi, Y. S. Moon, J. S. Lee, and Y. Choi. Bridge inspection robot system with machine vision. *Automation in Construction*, 18:929–941, 2009.
- [12] H. G. Sohn, Y. M. Lim, K. H. Yun, and G. H. Kim. Monitoring crack changes in concrete structures. *Computer-Aided Civil and Infrastructure Engineering*, 20:52–61, 2005.
- [13] A. Ito, Y. Aoki, and S. Hashimoto. Accurate extraction and measurement of fine cracks from concrete block surface image. *Proceedings IECON2002*, pages 77–82, 2002.
- [14] S. Thrun. Simultaneous localization and mapping. *Robotics and Cognitive Approaches to Spatial Mapping*, 38:113–41, 2008.
- [15] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots, 2001.
- [16] D. A. Forsyth and J. Ponce. *Computer vision: A modern approach*. Prentice Hall, 2003.
- [17] http://www.vision.caltech.edu/bouguetj/calib_doc.
- [18] J. Heikkil. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1066–1077, 2000.
- [19] Y. Mao, L. Dou, J. Chen, H. Fang, H. Zhang, and H. Cao. Combined complete coverage path planning for autonomous mobile robot in indoor environment. *Proceedings of the 7th Asian Control Conference*, pages 1468–1473, 2009.
- [20] T. Oksanen and A. Visala. Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, 26:651–668, 2009.
- [21] H. Choset and P. Pignon. Coverage path planning: The boustrophedon decomposition. *Proceedings of the International Conference on Field and Service Robotics*, 1997.
- [22] H. Choset. Coverage for robotics. *Annals of Mathematics and Artificial Intelligence*, 31:113–126, 2001.
- [23] H. Choset. Coverage of known spaces: the boustrophedon cellular decomposition. *Autonomous Robots*, 9:247–253, 2000.
- [24] Mobile Robot Inc. <http://www.mobilerobots.com/>.

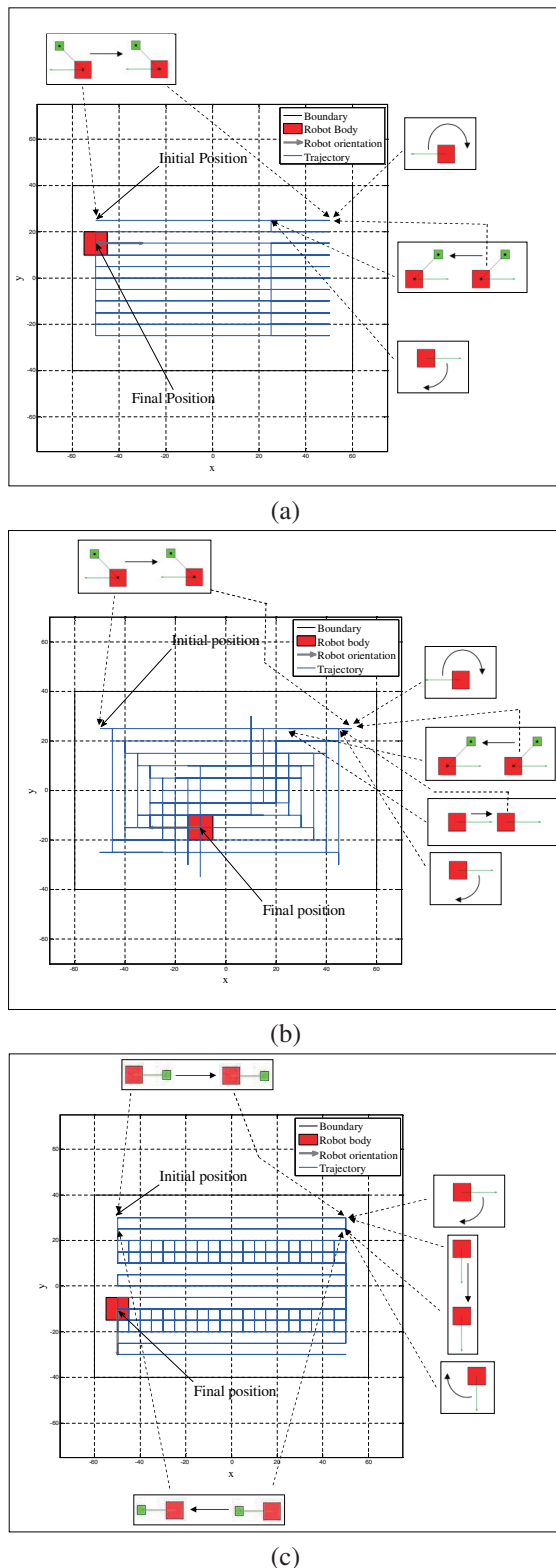


Fig. 11. The trajectory of the mobile robot for path planning algorithm type I, II, and III is shown in (a), (b), and (c), respectively

alignment problem and increase the accuracy of the crack map.