The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Strategies for Autonomous Robot to Inspect Pavement Distresses

Y.H. Tseng, S.C. Kang, Y.S. Su, C.H. Lee, J.R. Chang

Abstract— **Distress inspection is an important task in pavement maintenance. Pavement inspection requires tremendous human resources, so many investigators start developing automatic and robotic inspection methods to increase the efficiency and accuracy. However, the systems they developed are applicable for network-level inspection (large areas, long-distance) but too expensive or too big for project-level inspection (small areas, short-distance). Dealing with this problem, some researchers have developed an autonomous vehicle for inspection. In this research, we specific focus on developing strategies for executing the inspection tasks using robots. We developed three strategies. The first strategy is random-walk. The second strategy is random-walk with map recording. The third one adds the vision capacity to the robot. To validate the three strategies, we developed a test field in a virtual environment. This test field includes 5 types of common distress, including an alligator crack, a patching, a breaking hole, a rectangular manhole and a circular manhole. We also developed a virtual robot which can autonomously navigate in the test field. We then implemented the three survey strategies in the robot and compare their performances with traditional longitudinal survey method. The results show that using the first strategy, we can increase frequency for passing the distresses; it means that robot can detect and collect data more times than traditional longitudinal survey. The results of the second strategy show that we can increase the repeatability by using map recording to guide random-walk. The results of third strategy show that the robot can find more distresses in a certain amount of time; it means that we can improve survey efficiency by adding vision capacity to adjust motion path when distresses detected. Comparing these strategies with conventional Longitudinal survey strategy(L strategy), the three proposed strategies have a higher possibility of revisiting distresses, and it means making the results more reliable.**

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

Distress inspection is an important task in pavement maintenance and rehabilitation (M&R) (Abaza et al., 2004). Currently, pavement distresses are detected and recorded by manual inspection (Miller & Bellinger, 2003). Previously, engineers have had to manually observer and record and pavement distress data on paper, which is difficult to store and analyze. In recent years, engineers have used various instruments, such as PDAs, to record and digitalize their findings. In the past 30 years, many pavement management agencies follow standards, such as pavement condition index (PCI), to evaluate the coverage and severity of pavement distresses (Darter & Shahin, 1980). The manual inspection approach, however, is still very costly, time-consuming and labor-intensive, and is often unable to be accepted in accordance to current regulations and best practices.

To accelerate the inspection process, previous researches in the past 30 years have developed automated inspection vehicles. During the 1970s, PASCO (Haas et al., 1994) started developing the first inspection vehicle. Fugro Roadware, a Canada-based company, then developed Automatic Road Analyzer (ARAN). This inspection vehicle was equipped with a panoramic camera to collect pavement images and used image processing software WiseCrax(Groeger et al., 2003) to detect the cracks automatically. Another inspection vehicle, Digital Highway Data Vehicle (DHDV) developed by WayLink Systems Corporation in the U.S. integrated a laser-scanner and was able to detect cracks in real time by processing the data collected from the laser scanners. Mandli's Pavement System (Lee, 2005) used downward line scan camera to collect pavement data and used customized software, Roadview Automated Distress Rating Software to detect pavement distresses automatically. Beside the industrial solutions, many academic researchers, such as Chou (1996), Cheng (2003) and Rababaah (2005) also developed inspection vehicles by integrating sensors and software to achieve the goal of automatic inspection.

Although the inspection vehicles significantly enhance the efficiency of pavement inspection, they can only be used when surveying in network-level inspection (large areas, long-distance). In many project-level inspection (small areas, short-distance), the vehicles are not applicable because it is expensive; furthermore, in some cases, the vehicles cannot reach the survey area because of licenses or its big volume. To inspect in those cases, we started developing an autonomous robot for inspection (Chang et al., 2007). Robots have greater mobility, are more convenient and have the ability to save on manpower, making it very suitable for small surveying tasks. Because robots can carry multiple types of sensors and integrate them the motion control, they are also able to perform the inspection tasks more intelligently. For example, an inspection robot can survey the road based on its sensor readings and alter its motion accordingly to complete

This work was supported by the National Science Council of Taiwan for their financial support provided under the project of NSC 96-2628-E-159-002-MY3.

Y.H. Tseng is with Graduate Student of Computer Aided Engineering (CAE), Department of Civil Engineering, National Taiwan University windanthem@caece.net

S.C. Kang is with Associate Professor of Computer Aided Engineering (CAE), Department of Civil Engineering, National Taiwan University sckang@caece.net

Y.S. Su is with Graduate Student of Computer Aided Engineering (CAE), Department of Civil Engineering, National Taiwan University operaticlife@caece.net

C.H. Lee is with Graduate Student of Computer Aided Engineering (CAE), Department of Civil Engineering, National Taiwan University piggyhoward@caece.net

J.R. Chang is with Associate Professor of Department of Environmental and Informatics, Minghsin University of Science and Technology jrchang@must.edu.tw

the inspection tasks efficiently. Although Gu et al. (2008) has already developed an inspection robot, they focused mainly on robot hardware architecture and its control. It is necessary to extend this idea further by developing motion strategies to allow it become an intelligent inspection robot.

In short, the previous research has done very little on the motion planning of autonomous robot specifically for the pavement inspection. We would like to focus on this fundamental problem and developing a simple and effective strategy for pavement inspection robots. To accomplish this task, we try to learn from the research targeting similar problems, such as autonomous mining sweeping robots and vacuum cleaning robots, such as Choset (1997) (2001) developed coverage path planning algorithms applying to vacuum, floor scrubbing and de-mining.

B. OBJECTIVE

The objective of this research is to develop intelligent behaviors for inspection robots. This is a novel research in pavement inspection. These behaviors can increase the efficiency and effectiveness of the robot when performing autonomous inspection. We focused on the framework rather than the detailed techniques. To be more specific, we would like to develop strategies to make the robots maneuver intelligently while performing inspection tasks. The strategies need to be simple and reliable, and must be easily implemented in a robot to ensure that they are able to follow pre-programmed motion patterns while gathering robust results at the same time.

II. SURVEY FUNCTIONS AND STRATEGIES

To achieve the goal of intelligent robot survey, we developed and implemented five computational functions to automate the motion of the robot. After that we developed three strategies to integrate these five functions to control the robot autonomously. The following section introduces these functions and strategies.

A. MAPBUILDING() FUNCTION

Obtaining the map of the survey area is the initial step of a survey procedure. The robots must know the survey region before they start. We employed the ray-crossing method (Foley et al., 1995) to set up the map.

B. DISTRESSFINDING() FUNCTION

The DistressFinding() function is designed to detect pavement distresses from images retrieved by the robot. The first step of the function is to monochromatize the collected image (i.e. make the image grayscale). The second step of the DistressFinding() function is to extract features from images using Fast Corner Detection (Rosten et al., 2006). In this research, we detected corner features in the images. The third step of the procedure is classification. To quickly classify pavement images, we use the results from the corner detection to build a classifier. Finally, we define a threshold value for the corner point to differential normal images from images with distresses such as alligator cracking, breaking hole and manhole.

C. RANDOMSURVEY() FUNCTION

This function is for motion control. We employ a random-walk algorithm, the most widely used algorithm in the robot research field. This method ensures simple implementation (even without sensor feedback) and reliable results. As long as the robot has sufficient survey time, it can cover the entire survey region.

D. MAPRECORDING() FUNCTION

The MapRecording() function is used to record the surveyed area by updating the robot's position continuously. Using this function to update the survey map, we can drive the robot to places that have not yet been surveyed. It can increase visit rates to any unvisited areas and reduce time wasted inspecting areas that had already been surveyed.

E. VISIONGUIDANCE() FUNCTION

The main idea behind the VisionGuidance() function is for it to quickly process the images captured from the camera and dynamically adjust the motion of the robot.

F. SURVEY STRATEGIES

The survey strategy is used by the autonomous robot to execute pavement inspections. According to the functions that we have designed, we developed three motion strategies: (1) Strategy I: random survey; (2) Strategy II: random survey with map recording; (3) Strategy III: random survey with map recording and vision guidance.

1) SURVEYSTRATEGIESI: The first strategy is random survey (R), the most basic motion planning strategies using to solve the classical robotics problems. Using this strategy, the robot can move randomly within a confined environment. This strategy is composed of a MapBuilding() function, a DistressFinding() function, and a RandomSurvey() function. In this strategy, the RandomSurvey() function is executed in the main computational thread and the DistressFinding() function is executed in a sub-thread parallel to the main thread. The two functions work simultaneously to allow the robot to survey and move at the same time.

2) SURVEYSTRATEGIESII: The second strategy is random survey with map recording (R+M). Using this strategy, the robot can randomly survey in the confined environment and record any data that it has collected. We use the MapBuilding() function to define survey map, DistressFinding() to detect pavement distresses, MapRecording() to record the surveyed area on map, and RandomSurvey() to navigate the robot. The RandomSurvey() function is executed in the main thread and the DistressFinding() and MapRecording() functions are executed in two sub-threads parallel to the main thread. The three functions execute simultaneously.

Using this strategy, robot can survey, move and record its path simultaneously during inspection. By adding the MapRecording() function into the strategy, the robot will have a better chance of moving to an area that has not already been surveyed. Since the robot always samples the goal point from regions that have not been visited, it can potentially reduce time wasted on traveling to regions that have already been inspected.

3) SURVEYSTRATEGIESIII: The third strategy is random survey with map recording and vision guidance (R+M+V). In this strategy, we add the vision feedback to the robot. We use MapBuilding()to define the survey area, Distress-Finding()to detect pavement distresses, MapRecording() to record the surveyed area, RandomSurvey() to navigate the robot, and VisionGuidance() to determine the inspection path as distresses are detected.

Same as R+M, RandomSurvey() is executed in the main thread; while DistressFinding() and VisionGuidance() are combined into a sub-thread. The MapRecording() function is still executed in an independent sub-thread parallel with the highest priority, if distresses are detected, the robot will be guided by these function, otherwise, the robot will navigate in the set area using random survey.

III. IMPLEMENTATION

To test the feasibility and efficiency of the proposed functions and strategies, we implemented a survey system, including both the hardware and software of a survey robot. We also implemented a virtual robot and an artificial test field to evaluate the robot's performance. The following sections introduce the implementation procedure.

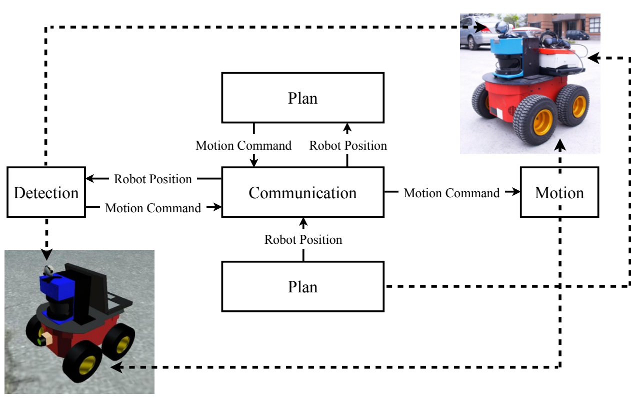

Fig. 1. The Architecture Of Robot Survey Program System

A. ARCHITECTURE OF THE SURVEY SYSTEM

The survey system includes 5 software modules, a robot and a simulator. The 5 software modules are: (1) communication module, (2) motion module, (3) plan module, (4) detection module and (5) position module. As shown in Figure 1. The Communication module is the hub of the system, connecting the other 4 modules. All the software modules are linked to both the robot and the simulator (the virtual robot) to enable autonomous inspection. Each module is described as follows:

1) Communication module: This module is actually a kernel of the whole system, and is in charge of collecting the information and sending motion commands to the robot. We employed inter-process communication (IPC) (Weisshaar, 1987) as the platform of data communication. It includes a set of techniques for the exchange of data among multiple threads in one or more processes.

2) Position module: This module is locates the robot in the survey map, and includes two of the aforementioned functions, MapBuilding() and MapRecording(). Because these two functions require the positional information to be retrieved from the robot, we had to implement two position modules, one for the physical one for the virtual world. The position module for the physical world is connected with a global position system (GPS), which continuously sends the position of the robot to the communication module. The position module for the virtual world is connected to a virtual encoder that simultaneously records the motion of the virtual robot, calculates its position and sends that information to the communication module.

3) Plan module: This module is the robot's motion planner, and includes functions such as RandomSurvey() to generate survey paths and strategies. As shown in Figure 3-1, the communication module continuously sends the robot's position to the plan module. The plan module then computes immediate movements of the robot and sends motion commands to the communication module to control the robot movement.

4) Vision module: This module is used to detect distresses on the pavement. Two functions, DistressFinding() and VisionGuidance() were implemented in the module. We used this system to monitor the pavement; if the system finds a distress, it records the data, such as distress positions and an accompanying image, and sends the command messages to execute reactive motion.

5) Motion module: This module is used to drive the robot. When it receives command messages from the communication module, this module sends the motion command to both the simulator and the robot. The robot will then move accordingly.

B. ROBOT HARDWARE

We implemented the robot by using four major hardware components. They are (1) motion platform, (2) GPS device, (3) image capturer and (4) computational processor. We selected Mobile Robot Pioneer P3-AT as the motion platform. P3-AT is a ready-made robot platform, suitable for field applications. P3-AT also integrates a laser rangefinder (SICK LMS-200) to detect obstacles and to localize the robot. These mechanisms are autonomous functions for pavement inspection.

We selected the Leica SR530 as the GPS device as it provides high-accuracy. We especially used Virtual Reference Station (VRS) to increase the accuracy of the real-time positioning system. We connected the VRS built and maintained by the Taiwanese government and distributed all over the country. By using 3.5G wireless network, the GPS device retrieves the position information from VRS, reducing any calculation errors (Huang, 2009). The real-time accuracy (data analysis can be done before the next data received) achieve centimeter-level which is acceptable in pavement inspection.

The third component is the image capturer. We selected Logitech QuickCam Pro 5000 webcam to retrieve the images in real time. The QuickCam Pro 5000 is very light-weight, approximate 400 grams and with a Universal Serial Bus (USB) interface. Its resolution is very high (1.3 Megapixels) and can automatically adjust the exposure according to lighting conditions for the best image quality.

Lastly, we selected a laptop computer, an IBM Lenovo ThinkPad X61 as the computational processor for motion planning, image processing and communicating with the hardware in real time.

map

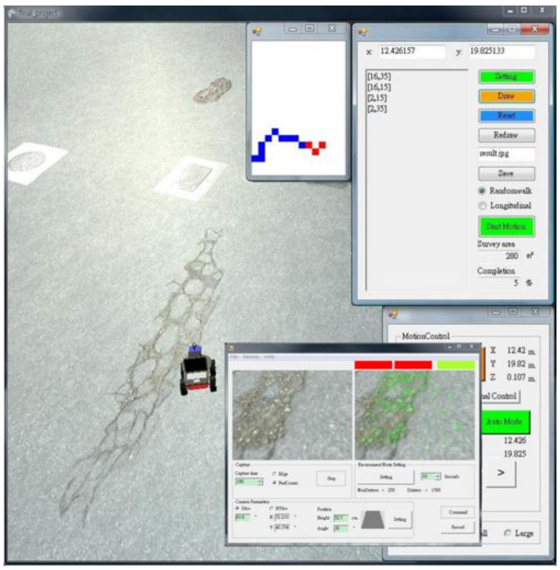

Fig. 2. Simulator and Virtual Environment

C. SIMULATOR (VIRTUAL ROBOT)

We built a virtual robot by following the actual robot for simulation purposes. Figure 2 shows a snapshot of the simulator. The appearance and dimension of the virtual robot is identical with the actual one, and includes a virtual platform, camera, GPS and computer. Because physical characteristics were simulated, the robot can move exactly like an actual robot in the virtual environment. The whole simulator was developed via three stages.

The first stage was to create a virtual robot. We used SketchUp, a well-known freeware for 3D modeling, to create the 3D model of the robot. We introduced the 3D model to a game environment constructed using XNA (Cawood & McGee, 2007). XNA is a set of tools with a managed runtime environment provided by Microsoft that facilitates computer game development and management. We created a virtual environment by using XNA, and then imported the virtual robot. Other than the robot, we also created two virtual actuators to allow users to control the wheels, similar to the control of an actual robot. Because we matched the capacity of the virtual and actual robot, the virtual robot was also set to reach speeds of 0.7 meters per second.

The second stage was to create virtual sensors. A virtual camera was created by attaching a viewport to the top of the virtual robot. We set the field of view (FOV) as 63 degrees (diagonal), matching the range of view angle of the QuickCam Pro 5000. By using this setting we are able to retrieve images as the physical camera on top of the real robot would. The other sensor was the virtual GPS, whose role is to send positional information to the robot. In the virtual world, this process is very straightforward. We only developed a short function to trace the center of the robot and then to continuous send the positional information.

The third stage was to create a virtual environment with all physical feedbacks found in the real world. We used a physics engine, PhysX (Rieffel, et al., 2009) to create an environment with physical feedbacks. We implemented various physical characteristics, such as friction, center of gravity and the resistance from collisions, to all the objects in the environment. These settings ensure realistic interactions between robots and objects in the virtual environment.

IV. TEST

We conducted multiple field tests to validate the system. We also designed a series of tests to evaluate the performance of the three proposed strategies. In particular, we focused on comparing the efficiency of finding distresses using a number of different survey strategies. The following sections describe the details of the test plan, the process and the results.

A. FIELD TESTS

The goal of the field tests is to validate the survey system. All the tests were conducted on Palm Ave. located on the campus of National Taiwan University (NTU). The pavement of the test field was asphalt pavement. As the test field is a major road on campus, servicing numerous vehicles and pedestrians day by day, many distresses, such as alligator cracks and breaking holes, occurred on the surface. The surface of the pavement also had many manholes and road signs.

From the tests, we found that the robotics survey system we developed is suitable for pavement inspection. We were able to successfully utilize the MapBuilding() and MapRecording() functions to create and update a survey map by using position information from the GPS. We were also able to use RandomSurvey() to configure the robot to move randomly within the allocated boundary. Lastly, we were able to successfully use the DistressFinding() function to detect distresses and the VisionGuidance() function to adjust the robot's path when a map has been found.

From these tests, we also discovered some limitations of the field test. First, it is very time consuming and labor-intensive to conduct testing in the field. The average testing time (including preparation time) was approximate 6 hours. The whole testing team consisted of three people, one for robot control, one for traffic control and another for video recording. It was also very difficult to find the right time to conduct testing as the road experienced heavy traffic throughout the day. To avoid the traffic problem, we were usually required to conduct the tests as early as 4 am. Furthermore, it is very difficult to find an "ideal" test field with all types of distresses. Therefore, we need to develop a virtual field for further testing.

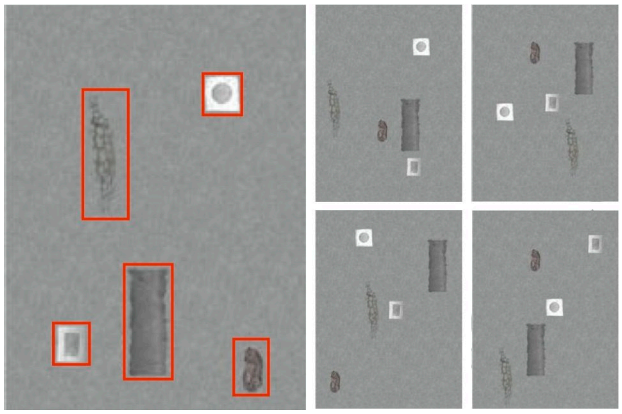

Fig. 3. The Figure Of Setting Distresses

B. VIRTUAL FIELD

In this research, we developed an artificial test field in a virtual environment. Due to the difficulties of conducting tests in the real world, as mentioned above, we further developed a virtual test field using computer graphic technologies. This test field includes a map with 5 typical pavement distresses and a virtual robot. The following section describes this virtual environment.

We create five field maps, each with uniform pavement materials and 5 types of distresses. In Figure 3, we show the figures of 5 types of distresses implemented in this research. They are alligator cracking, patching, breaking hole, rectangular manhole and circular manhole.

We randomly arranged the 5 distresses and created maps for each scenario (virtual test field), as shown in Figure 3. They each have a different distress distribution, which are randomly arranged on the map and do not overlap with each other. Each test field is formed within a boundary measuring 14.7 meters by 20 meters. The 5 distresses were then proportionally scaled down to fit in the map.

C. TEST PLAN

Our plan is to compare 4 strategies in the tests. The first strategy is longitudinal survey strategy (called L strategy). This is the strategy most commonly used in existing survey vehicles. L strategy begins by asking the robot to move along the pavement from one side of the survey boundary to the other side. This ensures that the robot is able to inspect the entire survey region. The others three strategies are the three strategies described earlier in this research. They are (1) random survey strategy (R); (2) random survey with map recording strategy (R+M); and (3) random survey with map recording and vision-guidance (R+M+V).

The goal of this test is to compare the performance of the robot survey using different strategies. Two factors influencing the performance need to be considered simultaneously. The first is whether the robot can travel efficiently and consistently to areas that need to be surveyed. The second is whether the robot's sensors (including cameras) can effectively retrieve the information from the field and maintain a high successful rate of identifying distresses.

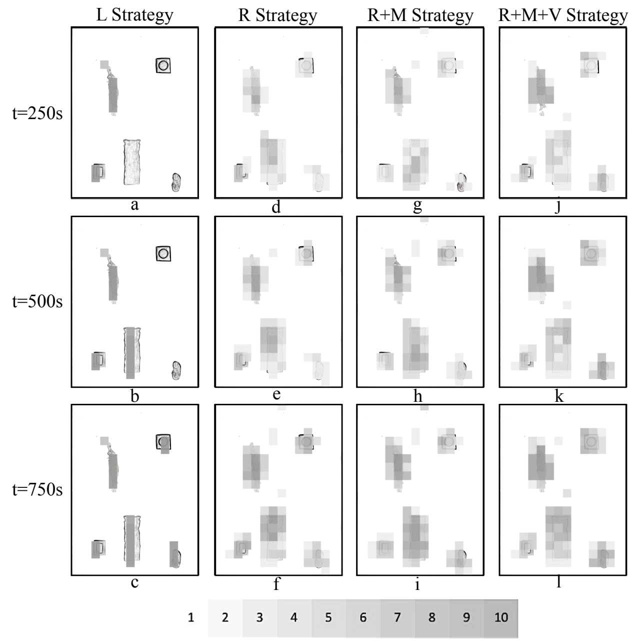

Fig. 4. Heat Maps

D. CONSISTENCY BETWEEN TESTS

Figure 4 shows the "heat maps" of the survey. The heat maps integrate the results of the 10 tests for each strategy. A ten-scaled color scheme was used to represent the number of distresses identified. The gray marker represents distresses that were detected in all 10 tests. The marker of lightest gray color represents distresses that were identified only once.

From Figure 4 (a), (b) and (c), we can see that the majority of the markers is dark gray, which means that the survey results are exactly the same in all 10 tests. There is one exception, a light gray mark in the top left of the image. This marker may be a result of the uncertainty of the DistressFinding() function. Although the robot moves along the same path, the image captured from the camera could be slightly different, resulting in the different classification.

Figure 4 (d) to (l) represent the heat maps of R, R+M, and R+M+V strategies. We can see a significant difference between the three strategies and the L strategy. The major difference is that using the three strategies, the robot may have a higher chance of identifying the distress. There are also more markers in Figure 4 (d) to (l) compared to Figure 4 (a) to (c), showing that the robot can identify distresses in more areas using these three strategies. This is because using the proposed strategies, the robot can retrieve images and data from different viewpoints and orientations. The variation increases the success rate of distress identification.

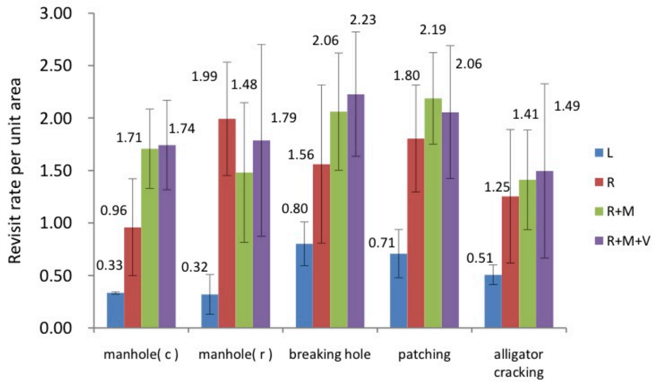

Fig. 5. Revisit rates for distresses

E. REVISIT RATE

Figure 5 shows the revisit rates of the 4 strategies. The higher the revisit rate the greater the chance that the robot visits a distress. This reduces the chance of erroneous identification by the image processors.

The results are shown in Figure 5. From the results, we found that random survey strategies, including R, R+M and R+M+V, performed much better than the determinative L strategy. As shown, it is able to increase the chance that the robot is able to visit individual distresses, leading to a higher success rate of distress identification.

V. CONCLUSIONS AND FUTURE WORKS

In this research, we have developed three motion strategies specifically for the survey robot. They are (1) random survey (R), which enables the robot to survey a confined area in a random manner; (2) random survey with map recording (R+M), which appends the MapRecording() function so the robot remembers areas that have already been surveyed; and (3) random survey with map recording and vision guidance (R+M+V), which integrates vision capabilities with random survey and MapRecording() functions.

To test the effectiveness of these strategies, we built a virtual robot and a virtual test field, which included five types of typical pavement distresses. We also implemented longitudinal survey strategy (L strategy), the most common survey method in current practice, for comparison. We repeated the pavement surveys by using the four strategies and measured the number of distresses found and also the number of times each distress was revisited. The findings are as follow:

(1) The proposed survey strategies, including R, R+M, R+M+V, are all capable of increasing the time taken to complete the survey process, identifying more pavement distresses but with less predictability.

(2) The MapRecording() function and VisionGuidance() functions can improve the efficiency of random survey. The efficiency of the three strategies we proposed is highest with R+M+V, lowest with R, with R+M in the middle.

Comparing all strategies, the three proposed strategies have a higher possibility of revisiting distresses. The higher the number of revisits, the greater the chance the robot may have of identifying and classifying the distresses, making the results more reliable.

In the future, the research results can be a reference to the future autonomous pavement inspection robots. Moreover, we plan to implement survey strategies in an actual robot, so that we can get the standards, such as PCI, by recording the inspection area and how many distresses are in this area. Furthermore, incorporating the knowledge of cracks may make the strategies more efficiency.

VI. ACKNOWLEDGMENTS

The authors would like to thank the National Science Council of Taiwan for their financial support provided under the project of NSC 96-2628-E-159-002-MY3.

REFERENCES

[1] K.A. Abaza, S.A. Ashur, I.A. Al-Khatib. Integrated pavement management system with a Markovian prediction model. *Journal of Transportation Engineering*, 130:24, 2004.

[2] J.R. Chang, Y.H. Tseng, S.C. Kang, C.H. Tseng, P.H. Wu. The study in using an autonomous robot for pavement inspection. In *Proceedings of International Symposium on Automation and Robotics in Construction, Kochi, Kerala, India*, 2003.

[3] S. Cawood, P. McGee. *Microsoft XNA game studio creator's guide: an introduction to XNA game programming*. McGraw-Hill Osborne Media, 2007.

[4] H.D. Cheng, X.J. Shi, C. Glazier. Real-time image thresholding based on sample space reduction and interpolation approach. *Journal of Computing in Civil Engineering*, 17:264, 2003.

[5] H. Choset. Coverage path planning: The boustrophedon cellular decomposition. *International Conference on Field and Service Robotics*, Citeseer, 1997.

[6] H. Choset, P. Pignon. Coverage for robotics–A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1):113–126, 2001.

[7] C.J. Chou, T.K. Liau. Development of automated algorithms for pavement condition survey. *Transportation Research Record: Journal of the Transportation Research Board*, pages 103–109, 1996.

[8] M.I. Darter, M.Y. Shahin. Pavement rehabilitation: Identifying the need. *Transportation Engineering*, 106(1):1–10, 1980.

[9] J. Foley, A. Van Dam, S. Feiner, J. Hughes. *Computer graphics: principles and practice*. Addison-Wesley Professional, 1995.

[10] J.L. Groeger, P. Stephanos, P. Dorsey. Evaluation of AASHTO cracking protocol: quantifying distress in asphalt pavement surfaces. In *82nd Annual Meeting of the Transportation Research Board, Washington, DC*, 2003.

[11] K.Y. Gu. Motion planning method for robotics pavement inspection. Taipei, Taiwan: Master's thesis, National Taiwan University Graduate School of Civil Engineering., 2008.

[12] R. Haas, W.R. Hudson, J. Zaniewski. Modern pavement management. *Road & transport research: a journal of Australian and New Zealand research and practice*, page 119, 1994.

[13] C. C. Lee. A study of applying line scan CCD camera on pavement distress image survey. Taiwan: Master's thesis, National Central University Graduate School of Civil Engineering.,2005.

[14] J.S. Miller, W.Y.Bellinger. Distress identification manual for the long-term pavement performance program. *Federal Highway Administration*, 2003.

[15] H. Rababaah, D. Vrajitoru, J. Wolfer. Asphalt pavement crack classification : a comparison of GA, MLP, and SOM. In *Proceedings of Genetic and Evolutionary Computing Conference, Late-breaking paper, Washington, D.C.*, pages 25–29., 2005

[16] J. Rieffel, F. Saunders, S. Nadimpalli, H. Zhou, S. Hassoun, J. Rife, B. Trimmer. Evolving soft robotic locomotion in PhysX. In *Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*, pages 2499–2504. ACM, 2009.

[17] E. Rosten, T. Drummond. Machine learning for high-speed corner detection. *Computer Vision–ECCV 2006*, pages 430–443, 2006.

[18] B.P. Weisshaar, F. Kolnick, A. Kun, B. Mansfield. Method of inter-process communication in a distributed data processing system, Sept. 15 1987. US Patent 4,694,396.