Navigating a Miniature Crawler Robot for Engineered Structure Inspection

Weihua Sheng, Hongjun Chen, and Ning Xi

Abstract—This paper addresses the problem of how to navigate a miniature Crawler robot in a typical engineered structure inspection application—aircraft rivet inspection. First, a novel vision-assisted localization algorithm is developed to find the heading and position of the Crawler robot. Second, a new algorithm is developed to solve the path planning problem so that the Crawler robot can navigate through all the rivets. Experimental results validate the localization and path planning algorithms. This inspection system can be extended to other similar engineered structure inspection applications.

Note to Practitioners—This paper was motivated by the problem of automated aircraft rivet inspection. Currently, a new nondestructive inspection technology has been developed for deep layer crack detection around the rivet area on an aircraft body. However, the efficiency and accuracy of the inspection system can not achieve the maximum if the inspection is manually conducted by human operators. Therefore, there is a need for automated rivet inspection. This paper develops a mobile sensor that integrates the sensor and a micromobile robot. The control, localization and path planning problems are solved so that the micro mobile robot can navigate on the airplane body surface to carry out rivet inspection without human involvement. This mobile robot based inspection system can be extended to different engineered structure inspection applications by integrating different sensors. The use of vision-assisted localization method requires that the position information of the inspected subject, such as rivets, should be known from a source such as a CAD model. Also, the surface of the structure to be inspected should be reasonably smooth to enable the robot to stick to it. Our future work will integrate multiple mobile sensors to speed up the inspection process.

Index Terms—Climbing robot, localization, mobile sensor, nondestructive inspection (NDI), path planning.

I. INTRODUCTION

A. Motivation

For many civil, mechanical and aerospace structures, structural health monitoring (SHM) [1], [2] is very critical to maintaining their performance and preventing potential failures or disasters. For example, the body of most commercial aircraft has multiple rivets connecting multilayer skins. Because of the fatigue stress caused by the typical application of a commercial aircraft, cracks can be induced in the vicinity of rivet holes. These cracks should be detected as early as possible for necessary corrections [3].

In recent years, various sensing technologies and sensors for structural deformation or damage detection have been developed [4]. For example, a new nondestructive inspection (NDI) technique called flat geometry remote field eddy current (FG-RFEC) has been developed by Innovative Materials Testing Technologies, Inc. (IMTT, Inc.) [5] to accurately detect deep cracks around the rivets in multilayer aircraft structures with high sensitivity. However, these probes are mainly used in a manual way, where human operators carry the probes to conduct inspection. Therefore, the accuracy and efficiency of the inspection are greatly affected by the experience of the operator. To solve these problems, we propose to develop an automated inspection system using mobile crawler robots.

B. Related Work

Previous research in structure inspection using robotic devices has resulted in several prototypes [6]–[9]. In recent years, interests in the development of miniature climbing robots are growing rapidly [10], [11]. Siegel [12] gives a survey on the climbing robots used in aerospace structure inspection. These robots are usually small and have suction cups that enable them to climb on aircraft skins. The ROSTAM series robots developed by Benham et al. [13] aim to carry NDI sensors and video cameras for aircraft skin inspection. Seemann et al. [14] developed a tank-like, multisuction-cup-tracked robot called AutoCrawler for aircraft skin inspection. NASA JPL’s (Jet Propulsion Laboratory) MACS series robots [15] are small, lightweight mobile platforms that use suction cups for attachment and ultrasonic motors for motion. The Automated Nondestructive Inspector (ANDI) robot developed by Siegel et al. [12] was used for commercial aircraft skin inspection.

Most of the robotic devices developed so far are remotely controlled. The lack of effective navigation capability greatly limits the scope of their applications. Our goal is to develop a mobile sensor capable of autonomous navigation for structure inspection applications. The focus of this paper is on how to navigate such a mobile sensor on an aircraft body. This paper is organized as follows. In Section II, we introduce the Crawler robot used in the inspection and its pose error model. Section III investigates the localization, or pose estimation problem. Section IV studies the path planning problem raised in the autonomous navigation. Experimental results are presented in Section V and Section VI concludes this paper.

II. THE CRAWLER ROBOT AND ITS POSE ERROR MODEL

A. The Crawler Robot

The mobile sensor consists of a miniature FG-RFEC probe and a Crawler robot. Fig. 1 shows the prototype of the mobile sensor. The Crawler [16] is a bipedal robot with an underactuated mechanism to achieve a good balance between compactness and maneuverability. It is supported by two suction cups that can hold on anticipated smooth and nonporous surfaces. Each of its three joints is driven by a DC servomotor. Onboard touch sensors and pressure sensors are used for suction cup control. For localization, a CCD camera is mounted under the belly of the Crawler. There are two wireless communication channels. One is a low data rate channel to receive motion command from and return motion status to a remote host computer. The other is a high data rate channel to transfer camera view data to the remote PC. The dimension of the Crawler is about 80 mm × 50 mm, and its weight is about 450 g. Due to the limited space, readers are referred to [16] for the dynamic model and control of Crawler.

B. Robot Pose Error Model

Before studying robot localization, we should analyze the error introduced in the rotation and translation movement of the Crawler robot. As shown in Figs. 2 and 3, the amount of rotation and translation in each step is limited. We assume the step size for rotation is \( \theta_s \) and the step size for translation is \( d_s \). For each rotation step, the rotation error can
be modeled as a random error with a zero-mean Gaussian distribution. i.e., $\Delta \theta_s \sim N(0, \sigma_{\theta_s}^2)$. To turn an angle of $\theta$, the Crawler needs to make a total of $\lceil \theta / \theta_s \rceil$ independent rotations, where $\lceil x \rceil$ represents the minimum integer that is greater than or equal to $x$. Therefore, we have $\Delta \theta \sim N(0, \lceil \theta / \theta_s \rceil \cdot \sigma_{\theta_s}^2)$. For each translation step, the translation error can also be modeled as a zero-mean Gaussian distribution, i.e., $\Delta t_s \sim N(0, \sigma_{t_s}^2)$. To translate a distance of $d$, the crawler needs to make a total of $\lceil d / d_s \rceil$ independent translation steps. Similarly, we have $\Delta t \sim N(0, \lceil d / d_s \rceil \cdot \sigma_{t_s}^2)$.

As shown in Fig. 4, when the Crawler moves from rivet $a$ to rivet $b$, the position error caused by the rotation error and translation error is

\[
\Delta d_{ab} = \sqrt{(\Delta x_{ab})^2 + (\Delta y_{ab})^2 - 2(\Delta x_{ab} \Delta y_{ab}) \cos(\Delta \theta)}.
\]

As shown in Fig. 5, the robot position error at any rivet along a path can be calculated as

\[
\Delta d_i = \sum_{j=0}^{i} \left\{ \left( d_{ij+1} + 3 \sqrt{\frac{d_{ij+1}}{d_s}} \sigma_t \right)^2 + d_{ij+1}^2 - 2 \left( d_{ij+1} + 3 \sqrt{\frac{d_{ij+1}}{d_s}} \sigma_t \right) d_{ij+1} \cos \left( \frac{\sum_{j=1}^{i} \left| \theta_{ij} \right| \sigma_{\theta}}{\sigma_{\theta}} \right) \right\}^{1/2}.
\]

The above equation shows that the position error significantly increases as the Crawler robot makes more turns and translations. The longest straight line distance that a Crawler can move without getting lost as the threshold distance is designated as $d_r$.

III. VISION-ASSISTED ROBOT LOCALIZATION

As shown in the above section, the Crawler robot has limited accuracy in positioning itself over a long distance because the position error grows as the travel distance increases. If the robot makes any turn, the position error will be significantly larger. Such position errors are undesirable. Therefore, a pose estimation and correction process is needed to find the position and heading of the Crawler robot. As in many engineered structures, each rivet has its exact geometric coordinate stored in the aircraft CAD model. Therefore, we develop an effective vision-assisted pose estimation approach. In this method, the motor encoders on the Crawler are used to provide rough pose estimation, while the onboard camera is used to provide fine pose estimation using the rivets as landmarks. Consider the movement of the Crawler between two consecutive rivets $a$ and $b$. Initially, the center of the camera is right above rivet $a$ and the heading of the Crawler is along the line connecting $a$ and $b$. The pose estimation process consists of two steps: heading estimation and position estimation. Depending on the distance between

![Fig. 1. The prototype mobile sensor. Note that the FG-RFEC probe is not mounted yet in this picture.](image1)

![Fig. 2. The angle error in the rotation movement.](image2)

![Fig. 3. The position error in the translation movement.](image3)

![Fig. 4. Crawler moves from rivet $a$ to rivet $b$ but ends at $b'$ due to the angle error and the position error.](image4)

![Fig. 5. Position error accumulated over a sequence of rivets.](image5)
the two consecutive rivets, we consider the following two cases: 1) two
or more rivets are in a camera view and 2) only one rivet is in a camera
view.

1) Two or More Rivets in a Camera View: When the distance be-
tween two consecutive rivets is short, two or more rivets can be viewed
by the camera, as shown in Fig. 6, where \((x_1, y_1)\) and \((x_2, y_2)\) are the
coordinates of the center of two rivets \(a\) and \(b\) in the camera view. The
rivets can be recognized through image segmentation. It is straightforward
to estimate the heading of the Crawler. The angle between the line
connecting the two rivets and the horizontal line is the heading angle
which can be calculated as \(\theta = \arctan\left(y_2 - y_1/x_2 - x_1\right)\). The posi-
tion of the Crawler can be easily calculated based on the image and the
known positions of the rivets.

2) One Rivet in a Camera View: When the distance between two
consecutive rivets is bigger than the camera view, only one rivet is cap-
tured in the camera view. In this case, the distance from the encoders
can help. To estimate the heading, the Crawler captures two images
when it reaches rivet \(b\). One image is captured when the Crawler stops
and another image is captured one small step before it stops. When the
two images are captured, the Crawler also records the corresponding
encoder readings. In Fig. 7, \(c_1\) and \(c_2\) are the center of the images; \(L_{ab}\)
is the distance between the two rivets; \(L_e\) is the distance calculated
from the encoder reading corresponding to \(c_1; \theta\) is the heading angle;
\(\alpha\) and \(\beta\) are the internal angles as indicated in the figure. The heading
angle \(\theta\) can be calculated as follows:

\[
\theta = \arccos \frac{\left| b_{c_1} \right|^2 + \left| c_1, c_2 \right|^2 - \left| b_{c_2} \right|^2}{2\left| b_{c_1} \right| \left| c_1, c_2 \right|} - \arccos \frac{\left| b_{c_1} \right|^2 + L_{ab}^2 - L_e^2}{2\left| b_{c_1} \right| L_{ab}}.
\]

Once the heading angle \(\theta\) is known, the position of the Crawler can
be calculated based on the image and the known positions of the rivet.
Then, the heading and position can be used to correct the pose of the
Crawler. It can be seen that the heading and position error is limited
between two rivets and will not accumulate to the next rivet along the path.

IV. PATH PLANNING

A. Problem Statement

The vision-assisted pose estimation approach discussed above re-
quires that the distance between two consecutive rivets on the Crawler’s
path be less than the threshold value \(d_r\), beyond which the Crawler
robot may not be able to find the target rivet. A typical \(d_r\) is 15 in.
It also requires the number of turns that the Crawler makes along the
path that should be minimized, because it usually takes much longer
time for a Crawler to make a turn than to move straight forward [17].
A formal statement of the single Crawler robot path planning problem
is as follows.

Given a set of \(n\) rivets, \(P = \{y_1, y_2, \ldots, y_n\}\), in a 2-D space
with known locations \(\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}\), find a
sequence of the rivets so that a Crawler, starting from rivet \(s\), can
cross the local-ization threshold \(d_r\); 2) the number of turns is small; and 3) the
overall distance traveled is short.

We name the above path planning problem the Crawler Traveling
Salesman Problem (Crawler-TSP). The problem of traveling a set
of points with turn concerns has been studied by several researchers
[18]–[20]. Recently, Stein and Wagner [21] developed approximation
algorithms for the Minimum Bends Traveling Salesman Problem
(MBTSP), which minimizes the number of turns in the tour. However,
in their work, it is assumed that there exists a path between any pair
of points. This assumption does not hold in our Crawler-TSP be-
because the Crawler is not able to travel between a pair of rivets that are more
than distance \(d_r\) away. Their algorithm does not minimize the overall
length either.

B. The Find-Crawler-TSP Algorithm

The basic idea of our Find-Crawler-TSP algorithm is as follows.
First, the minimum number of line segments that partition all the rivets
are obtained, with no rivet covered by more than two line segments.
It is clear that the number of line segments determines the number of
turns a Crawler makes. Second, the minimum set of line segments are
connected with a goal to minimize the overall traveling distance.
The minimum set of line segments that partition all the rivets can be
be found by modeling it as a set-partition-problem [22]. To connect
the resulted line segments, we first construct a segment graph, where
each vertex represents a line segment, while the edge between two ver-
tices represents the existence of connecting lines between the two cor-
responding line segments. The edge weight is the arithmetic average of
the length of all the connecting lines between the two line segments.
Christofides’ algorithm [23] is adopted to find the shortest tour on the
segment graph. Once the order of line segments is determined, starting
from rivet \(s\), a rivet sequence can be found, which enables the Crawler
to tour all the unvisited rivets in the segment that covers \(s\) and end up at
a rivet in the next line segment with a minimum tour distance. This can
be done by enumerating the different traveling possibilities between
two consecutive line segments. The entering rivet in the next line seg-
moves the starting rivet for the next round of this process, until
all the rivets are visited and the Crawler goes back to rivet \(s\). By apply-
ing a shortcut on the generated tour, a final tour can be obtained.
Fig. 8 shows an example path generated by the Find-Crawler-TSP al-
gorithm.

V. EXPERIMENTAL TESTS

The prototype rivet inspection system is shown in Fig. 9. The CMOS
camera is black and white with a resolution of 380 TV lines and a field
Fig. 8. The final tour.

Fig. 9. The rivet inspection experiment system.

Fig. 10. Picture used in robot pose estimation: two rivets in camera view.

Fig. 11. Pictures taken at the first rivet, at the second rivet before and after Crawler stops.

of view of 45 mm × 34 mm. The host computer first runs the path planning algorithm, then it sends the position information of the next rivet to the Crawler robot. The Crawler robot receives the next target information and moves to it. Upon arriving at the target rivet, the Crawler sends the images back for vision-assisted localization.

A. Localization

We conducted the robot localization experiment in two cases: two rivets in camera view and one rivet in camera view. In the first case, the camera can see two rivets, as shown in Fig. 10. The experiment is independently conducted on a set of four rivets. The heading estimation results are shown in Table I, while the position estimation results are shown in Table II. It can be seen from this table that the heading and position can be estimated with high accuracy.

In the second case, there is only one rivet in the camera view. Three images are shown in Fig. 11. The first one is taken when the robot is about to leave the starting rivet. The other two images are taken before and after the robot stops at the target rivet, respectively. The heading estimation results are shown in Table III and the position estimation results are shown in Table IV. From these tables it is clear that both the heading error and the position error based on two one-rivet images are greater than those based on a two-rivet image.

B. Path Planning

To test the Find-Crawler-TSP algorithm, we run it on simulated rivet distributions. The set-partition-problem is solved using the software IPLOG Studio™. The test data is a collection of 2-D coordinates which represent the rivet locations. All the coordinate values are integer numbers between 0 and 10 (1 unit 10 in) and the data set forms a connected graph under this threshold.

We compare three algorithms. The first is Stein and Wagner’s Find-MBTSP [21]. The second one is called minimum-spanning-tree tour (MST-Tour) algorithm, which constructs the minimum spanning tree on all the rivets and then uses depth-first method to tour the minimum spanning tree. The third one is our Find-Crawler-TSP algorithm. Figs. 12–14 show the tours generated by the three algorithms. It can be seen that the Find-MBTSP algorithm generates the least number of turns. However, since it does not consider the distance
between two consecutive rivets, the tour is not feasible. The MST-Tour algorithm generates a feasible tour but it has too many turns. Our Find-Crawler-TSP algorithm obtains less turns and shorter travel distance, compared to the MST-Tour algorithm. We also calculated the total time cost due to the turns and straight movement. As we observed in our tests, the time cost for one turn is roughly equal to the time cost for 10 inch straight movement. Therefore, this cost is set to 1. The total time cost of the three paths can be calculated according to the following formula: total time cost = number of turns + total distance (in 10 inches). Hence, the total time cost for the three paths are 95.6, 117.6, and 94.6, respectively.

We also conducted tests on other random sets of rivets using the three algorithms. Figs. 15 and 16 show the number of turns and the total distance in different tests. Figs. 17 shows the total time cost in different tests.
total distance obtained from tests on data sets with varying sizes, respectively. The total time cost for each of the algorithms is shown in Fig. 17. Since the Find-MBTSP algorithm cannot guarantee feasible tours, we are more interested in comparing the MST-Tour algorithm and our Find-Crawler-TSP algorithm. From the above figures, it is easy to see that our Find-Crawler-TSP algorithm outperforms the MST-Tour algorithm.

VI. CONCLUSION

In this paper, a Crawler robot is adopted to conduct the engineered structure inspections. The autonomous navigation of the Crawler robot is studied, with a focus on the localization and path planning problems. A vision-based algorithm is developed for localization and a Find-Crawler-TSP algorithm is developed for the path planning. Simulation and experimental results are provided. The proposed solution can be extended to other similar engineered structure inspection applications that involve autonomous robots such as the inspection of bridges, buildings etc. Our future work will investigate the integration of multiple Crawler robots so that they form a robotized sensor network for faster inspection.

REFERENCES


