

Visibility-based Exploration in Unknown Environment Containing Structured Obstacles

Tirthankar Bandyopadhyay¹, Zheng Liu^{2,3}, Marcelo H. Ang Jr.¹,
Winston Khoon Guan Seah³

¹Dept. of Mechanical Engineering, ²Dept. of Electrical and Computer Engineering,
National University of Singapore, ³Institute for Infocomm Research, Agency for
Science, Technology and Research, Singapore.

Abstract:

How to enable a robot to explore in an unknown environment containing obstacles via local sensing is an important research issue for robot navigation, map building, and localization. In this paper, a reactive and visibility-based exploration algorithm is proposed. By line-of-sight sensors, the robot can find the occlusion points between covered and uncovered regions, and then approach these occlusion points to “sweep” the area. In addition, a prediction model based on local sensing is applied to refine the trajectory when the robot is moving; therefore as the robot tries to cover unknown area, it also tries to maximize the monitoring of covered area. Simulation experiment shows the efficacy of our approach.

Key Words:

Exploration, coverage, visibility, prediction, polygonal obstacles

1. Introduction

Exploration is one of the most basic capabilities for mobile robots to acquire information on unknown environments. It is the fundamental for robot localization and map building because exploration directly affects the input of the robot system. Without complete exploration, the robot cannot effectively find usable landmarks for localization or draw a complete map of the environment.

Exploration problem stems from the traditional coverage problem. However, the coverage problem mostly focuses on the sensible area that requires the optimal placement of static sensors to maximize the observation of the environment at one time point, while exploration concentrates on the sensed area after a period of movement of the robot. For example, as depicted in Figure 1.,

assuming the robot has panoramic line-of-sight sensor with infinite range, the robot in (1-a) is the winner of coverage because it stops at the end point that has much larger sensible area than in (1-b); however, from an exploration point of view (1-a) is inferior to (1-b) because the lower right region inside the small cell is not sensed.

The key problem for exploration can be elaborated as the following: “*Given what you know about the world, where should you move to gain as much new information as possible?*” [1]. A good exploration algorithm should have two properties, completeness and effectiveness. Completeness requires the robot to cover most of the environment; effectiveness emphasizes on the efficiency that the robot should achieve the completeness by minimal efforts (moves).

In this paper, we propose a visibility-based algorithm for robot exploration in unknown environments containing structured obstacles, i.e., polygonal obstacles. Assuming the robot has line-of-sight sensors with infinite detection range, e.g. ideal panoramic laser sensor or vision sensor, our exploration algorithm deploys the robot to move toward the occlusion points between covered and uncovered area. Therefore the robot can continuously search for unknown area to ensure the completeness of the exploration. In addition, a prediction model based on local sensing is used to adjust the motion of the robot; the aim is to maximize the coverage of the environment. By integrating these two approaches, we hope the robot can perform well in exploration and not lose the coverage if possible, e.g., the motion shown in Figure 2.

The remaining parts of this paper are organized as follows. Section 2 introduces the related work in robot exploration and coverage. Section 3 outlines

our visibility-based exploration algorithm including occlusion point approaching and prediction based refinement of exploration trajectory. Section 4 shows the efficacy of our approach by simulation. Finally, Section 5 concludes this paper and introduces our future work.

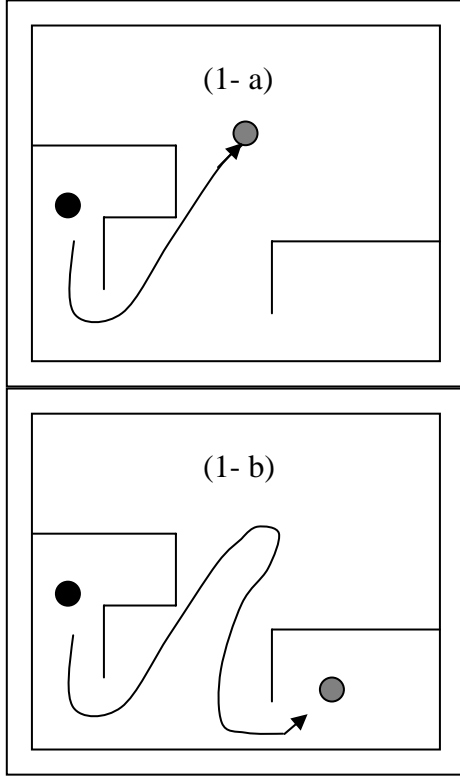


Fig. 1. Coverage and exploration
(Black dot: start point; gray dot: end point)

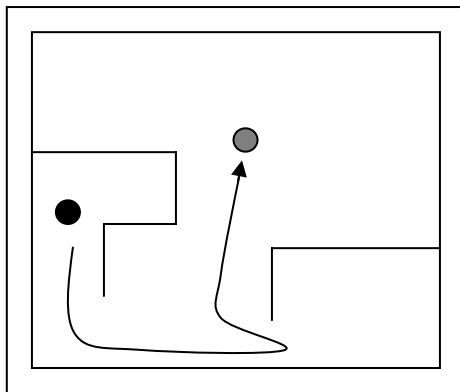


Fig. 2. Aim of our approach: to explore with maximal coverage
(Black dot: start point; gray dot: end point)

2. Related Work

The study on robot exploration is often integrated with the research on robot localization and map building, e.g. SLAM ([2, 3]). However, most of such work focuses on the processing of sensor data, e.g., fusion, filtering, etc. Few of them concern the exploration strategies or algorithms, which are in fact non-trivial especially for real applications. There are mainly two classes of approaches for robot exploration ranging from simple to complex, i.e., potential field based approach and frontier-based approach.

Potential field based approach concerns the obstacles inside the environment because they are usually the key factors affecting the observation of the environment. For example, in [4], Mataric et al introduce an artificial potential field based exploration algorithm for a team of robots. The idea is to map the sensed obstacles and robots as repulsive force sources and let the robots move under such forces; therefore the robots can be dispersed within the whole area. The potential field based approach can also be applied for tracking moving target, such as cooperative multi-robot tracking of multiple moving targets [5].

The simplicity of potential field approach makes it widely used in multi-robot systems. However, the simplicity also constrains the exploration performance because the robots only move by the received repulsive forces and they do not have target positions to approach to. This is somehow like a static deployment algorithm that is in fact a solution for coverage problem aiming to maximize the sensible area. For example, if the robots do not receive any repulsive forces (no obstacles or robots nearby) or the repulsive forces are balanced (in equilibrium states), they do not intend to move to further explore the environment. Therefore, the quality of exploration largely depends on the quantity of the robots. Without enough robots, a robot team cannot occupy the whole region for complete exploration even if they have infinite exploration time.

Frontier-based approach is derived from the exploration strategy that moves the robot to the most promising place to increase the sensed area. To achieve this, the robot needs to assess the covered space (sensed area) and unknown space to plan for the next move. A well known frontier-based exploration algorithm is proposed by Yamauchi [6]. With a grid represented map of the environment generated by local sensing, the robot can find the frontiers (boundary/edge) of the open space and then move toward the most promising

frontier according to some metrics. This approach can also be extended to multi-robot scenario [7], whereby the sharing of locally sensed information enables robots to explore more effectively. Recently, based on the concept of frontier, Gap Navigation Trees (GNT) is proposed to effectively represent the environment for better exploration [8]. Furthermore, frontier can also be applied for tracking moving targets. In [9], a robot can estimate the probability of losing a moving target via evaluating the target's distance to the frontier.

Compared with potential field based exploration, frontier-based exploration is more advanced because it can design motion trajectory for robots based on the map of the environment (drawn via local sensing or by other means). However, to map the environment, identify the frontiers and select the most promising frontier may require large memory space and lead to heavy computation burden.

In this paper, we propose a visibility-based approach for exploration. This approach inherits the advantage of frontier-based exploration that it can generate optimal movements for the robot to explore. In addition, our approach only requires the information of the occlusion points between covered and uncovered space. The details of our approach are to be introduced in next section.

3. Our Visibility based Exploration

As introduced previously, our research aim is to design an exploration algorithm that can enable the robot to explore in an unknown environment completely and effectively. Since the exploration strategy depends largely on the properties of the environment and robots, we clarify the assumptions of our approach first:

- Environment: bounded; two dimensions; contains polygonal obstacles. We can assume a normal size room that only has walls as the environment to be explored.
- Robot: equipped with panoramic line-of-sight sensor, the range of the sensor is infinity. The infinite sensing range may be realistic assumption because some vision sensor or laser sensors, e.g., SICK, can provides a view within 15 meters or more. Such range is comparable or even larger than the dimension of normal indoor environment.

Based on above assumptions, we may find that for the robot, the distance to unknown area is not the key issue for exploration because the sensor range

is infinite. The more important issue is the angle to the occlusion points of the obstacle that blocks sensing (eye-sight) of the sensors. As shown in Figure 3, the robot cannot see beyond the black obstacle therefore the gray region is uncovered. To solve this problem, a natural thought is to let the robot move along a leftward curve toward occlusion point A (small black point); therefore the robot can observe the uncovered region move efficiently.

To achieve this desired motion, the robot needs two basic behaviors, approaching to critical occlusion point, and adjusting motion to increase the coverage.

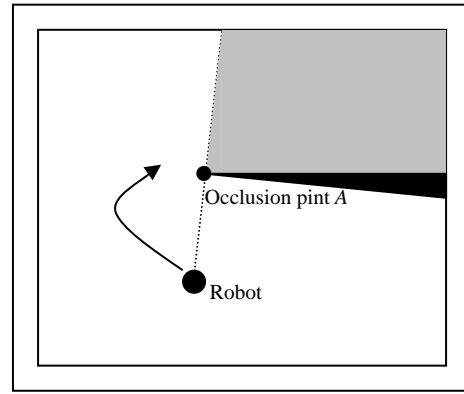


Fig. 3. A robot equipped with infinite line-of-sight panoramic sensor in an environment containing polygonal obstacles.

3.1 Approaching to the Critical Occlusion Points

First, we define the obstacle edge, free edge and occlusion point. The boundary of the visibility polygon of the robot consists of two types of edges: obstacle edges and free edges. Obstacle edges are part of obstacle boundaries. They directly block the sensor's line of sight. Free edges result from occlusion of the line of sight elsewhere. The occlusion points are the near ends of the free edges to the robot. The nearest occlusion point is termed as critical occlusion point. For instance, as depicted in Figure 4, the visible region of robot O is polygon $AA'B'DMNC'CEPF$; obstacle edges are $A'B'$, BD , DM , MN , NC' , CE , EP , PF , and FA ; free edges are AA' , $B'B$, and $C'C$; occlusion points are A , B , and C ; critical occlusion point is A .

The occlusion points indicate the edge/boundary between covered and uncovered region. If the robot wants to approaches to the unknown area, it needs to move toward the occlusion points. For

example, in Figure 4, if the robot can approach to critical occlusion point A , it may sense the region behind this point (the top-right gray region).

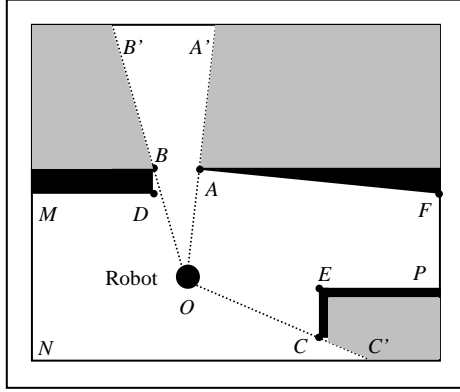


Fig. 4. Free edge, obstacle edge, and occlusion point

In our approach, the link between occlusion point and its projection (e.g., AA') is in fact the frontier between covered and uncovered space. However, we only let the robot move toward the occlusion point A , not the frontier AA' . This is based on the following considerations:

- In an environment containing structured polygonal obstacles, it is easier to use points to represent the obstacles. When the robot moves, the frontier of sensible region keeps changing but unless the robot across some specific region, the occlusion points do not change.
- Most frontier-based exploration algorithms need the robot to draw and store the map of the environment to detect and select the frontier; this may require large storage space of the sensor data and also demand heavy computation. However, for our occlusion point based approach, the robot only needs to find the end points of the obstacles as the occlusion points. By storing these occlusion point information, the robot does not have to build a detailed map of the environment.

In an environment with many polygonal obstacles, a robot may sense many occlusion points at a time. Each of the occlusion points calls the robot to search for the unseen region behind it. To effectively explore in the environment, the robot needs to visit these occlusion points sequentially. In our approach, we choose the nearest occlusion point to the robot as critical occlusion point to be explored first. When an occlusion point is finished, i.e., the region around it is all covered and it is no longer an occlusion point, the occlusion point will

be removed from the list of occlusion points. In addition, when a new occlusion point is found, it will be added to the list. By this means, the robot may explore in the environment effectively.

If the robot moves toward the critical occlusion point, it becomes nearer to the obstacle and the uncovered region. This is a necessary step for exploration; however, if the robot only moves directly to the critical occlusion point (e.g., point A in Figure 4), it can not increase the sensed area because the link (AA') does not move. Observing human behavior for exploration, a natural behavior is to approach to the occlusion point along a curve that can change the angle of the free edge of visible region, e.g., the trajectory shown in Figure 3. To mimic such behavior, the robot needs to take account for the increasing of sensed area by its local sensing. To solve this problem, we propose a prediction based refinement of the trajectory of the robot. This is to be introduced in next sub-section.

3.2 Adjusting Motion to Increase Coverage.

As introduced previously, approaching directly to the occlusion point cannot increase the sensed area. The robot needs to make a curve-like trajectory. We propose a prediction based method to solve this problem. The basic idea is to let the robot assume several possible new locations around its current location in the next step. Based on current local sensing, the robot predicts the change of sensed area in each “new location” and then can find the new location that can generate maximal increase of the sensed area.

Figure 5 shows how the prediction algorithm works. In (5-a), the black dot is the robot current location; the gray dot is the proposed new location. At current location O , the robot can detect the occlusion point A , its projection A' , and distance AA' . Assuming the robot will move to location P , the motion distance OP and angle AOP are known, then the area to be sensed can be predicted as sector $AA'A''$, whereas $AA' = AA''$. In (5-b), the black dot is the robot current location; the gray dot is the proposed new location. Just like in (5-a), occlusion point A , its projection A' , distance AA' , OP and angle AOP are known, then the area to be lost can be predicted as triangle $AA'A''$, whereas A'' is the projection of A if the robot moves to P . It worth to be noted that for uncovered region, the new projection point of the occlusion point has to be estimated (by assuming $AA'' = AA'$ in 5-a), while for covered region, the new projection is known (A'' in 5-b).

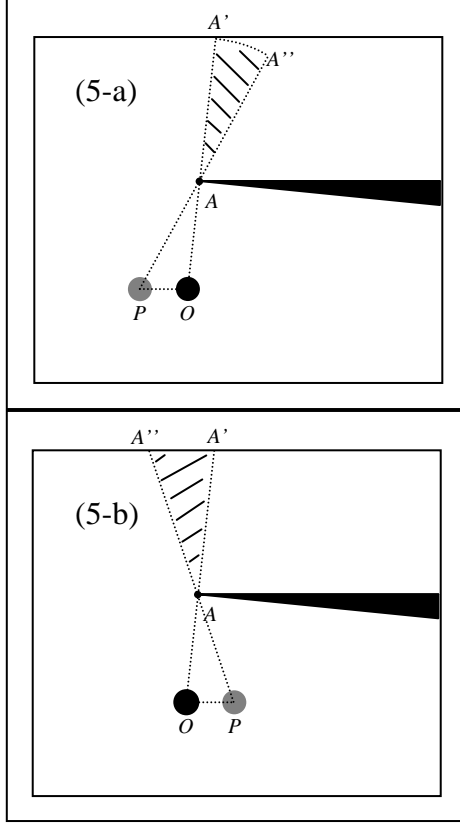


Fig. 5. Prediction of changes of sensing area
(Black dot: current location; gray dot: proposed new location)

In our approach, the robot moves to a position that locally maximizes the effect of current coverage and future exploration. Depending on the type of navigation function chosen, there can be different approaches to come up with the locally optimal position. e.g. If the function is smooth then a good approximation would be to take the local gradient direction for a small step.

3.3 Visibility-based Exploration

Above two sub-sections elaborate the two basic behaviors for our visibility-based exploration. Approaching to critical occlusion point helps the robot to explore the uncovered area. Prediction of changes of sensible area enables the robot to adjust its motion for better coverage. These two behaviors are integrated for the robot to explore.

In our approach, the robot works following the steps below:

- Step 1: Sense the environment, find all occlusion points. If an occlusion point has been covered previously, this point will not be added to the occlusion point list.

- Step 2: Choose the nearest occlusion point as critical occlusion point.
- Step 3: Predict the changes of sensible area each occlusion point in the list, not only for the critical occlusion point. For example, when the occlusion point list includes four occlusion points A , B , C , and D , while the robot plans to approach critical occlusion point A , it predicts the change of sensible area regarding all the occlusion points A , B , C , and D .
- Step 4: Select the best direction to maximize the coverage using equation (1).
- Step 5: To decide the moving direction, combine the direction to critical occlusion point and the direction to maximize coverage, as shown in Figure 6.
- Step 6: move along the decided direction, then examine the critical occlusion point, if it is covered, i.e., not an occlusion point in current visible region, remember this point as covered.
- Step 7: go to step 1.

The best step to move is taken by locally moving towards the gradient of V_{final} where

$$V_{final} = w_1 V_{exploration} + w_2 V_{coverage}.$$

Taking the gradient,

$$\nabla V_{final} = w_1 \nabla V_{exploration} + w_2 \nabla V_{coverage}.$$

The future exploration component is encoded as moving towards the critical occlusion points in order to explore the region beyond the current visibility in the quickest possible way. The quickest possible way to reach a critical occlusion point is the unit vector pointing towards the occlusion point.

$$F_{Critical\ Occlusion\ Point} = \text{tangential}(\text{robot position}, \text{critical occlusion point});$$

The maximal coverage component is encoded as the swinging behavior of the robot in order to maximize the current coverage. The direction that maximizes the coverage for one visibility edge is to swing the visibility edge around the occlusion point, i.e. to move perpendicular to the direction to the occlusion point.

$$F_{Maximal\ Coverage} = \text{normal}(\text{robot position}, \text{critical occlusion point});$$

These two components are added up to get F_{final} .

$$F_{final} = w_1 F_{Critical\ Occlusion\ Point} + w_2 F_{Maximal\ Coverage} \quad (1)$$

Observing equation (1), if there ~~has~~ a critical occlusion point, the robot will move toward it with the consideration of coverage; if there has no

critical occlusion point inside the visibility region, the robot will only move to the location to maximize the coverage of the environment. It should be noted that if there has no critical occlusion point and possible next location to maximize coverage, the robot will stop and cannot move to explore.

Compared with the frontier-based exploration, our approach does not require the robot to map the sensed environment in details, e.g., grid represented map of all the stuffs sensed. The robot only needs to remember the occlusion point information (step 2, 6). Such abstract information can enable the robot to explore in the environment with satisfactory performance.

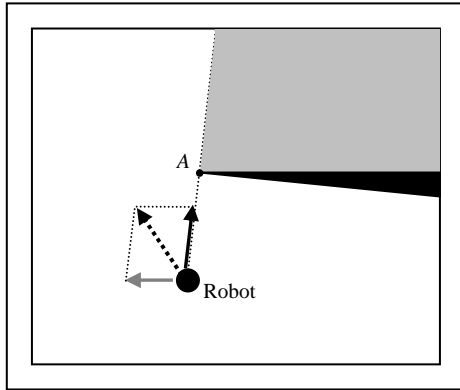


Fig. 6. Moving towards critical occlusion point with the consideration of coverage (Solid black arrow: direction to critical occlusion point; solid gray arrow: direction to the best location for coverage; dashed black arrow: final moving direction)

4. Simulation and Discussion

We test our exploration algorithm using LEDA [10], a C++ Library of Efficient Data types and Algorithms. To justify the efficacy of our approach, we compare it with the random exploration. The random exploration algorithm let the robot move forward until it detects an obstacle ahead, then it changes the direction and then moves forward until another obstacle is found ahead.

Figures 7 and 8 show the exploration trajectory of our visibility-based approach and the random exploration respectively. Comparing the trajectory in these figures, we may find that the two basic behaviors, approaching to critical occlusion point and refining the motion based on prediction, can effectively lead the robot to explore in the unknown environment. The performance is

satisfactory that the exploration result is complete and effective.

Figure 9 compares the exploration performance between our approach and random exploration. The x-axis is the simulation timeline; the y-axis is the accumulated percentage of covered area. At any time, higher y value indicates better performance because the robot has covered more area. The result shows that random exploration cannot cover the environment effectively compared with our approach.

In Figures 10, 11, and 12, our approach is compared with random exploration in another environment. The simulation results again show the efficacy of our approach.

As introduced previously, our visibility-based exploration algorithm only needs the robot to have abstract memory of the occlusion points, i.e., remember whether an occlusion point has been covered or not. This property brings great advantage compared with frontier-based approach that requires the robot to build a detailed map of the sensed environment. However, as introduced previously, to decide the moving direction by equation (1), the robot needs to have at least an occlusion point in its visible region ($F_{Critical\ Occlusion\ Point} \neq 0$), or have possible next step to increase the coverage ($F_{Maximal\ Coverage} \neq 0$). If both $F_{Critical\ Occlusion\ Point}$ and $F_{Maximal\ Coverage}$ are zero, the robot cannot move any more. This is a local minima problem depicted in Figure 13. If the robot has covered all the detected occlusion points inside its visibility region, it can only move to the location to maximize the coverage, therefore it cannot move out of this region to further explore the environment.

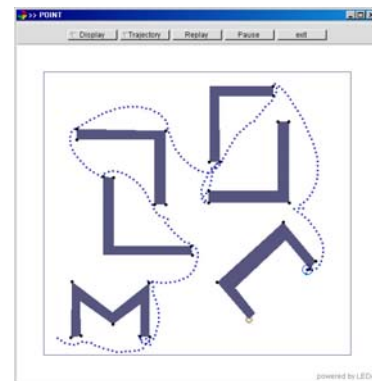


Fig. 7. Visibility-based exploration trajectory (start from left bottom)

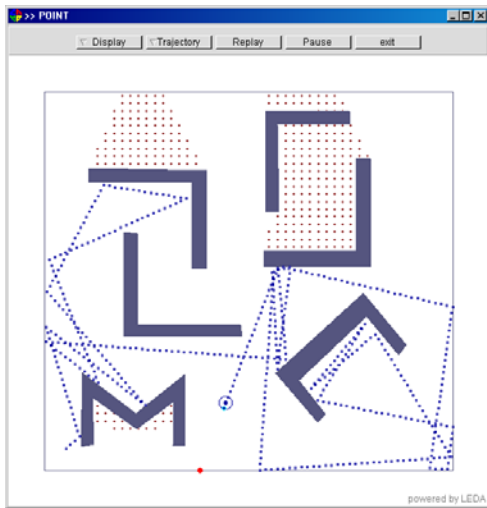


Fig. 8. Random exploration trajectory
(start from left bottom; dotted area: uncovered region)

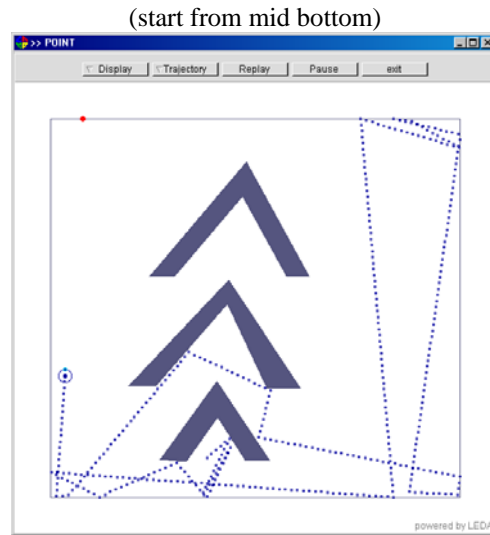
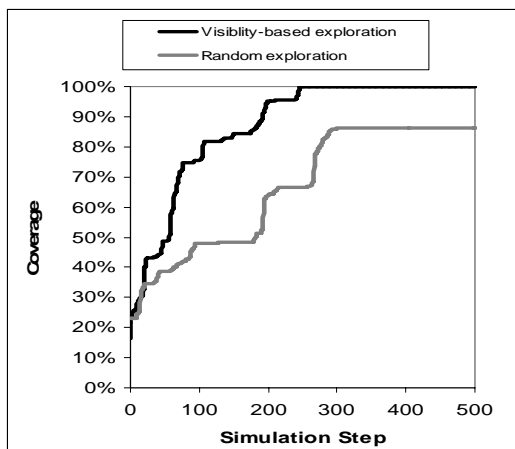


Fig. Fig. 11. Random exploration trajectory
(start from mid bottom)



ig. 9. Exploration performance

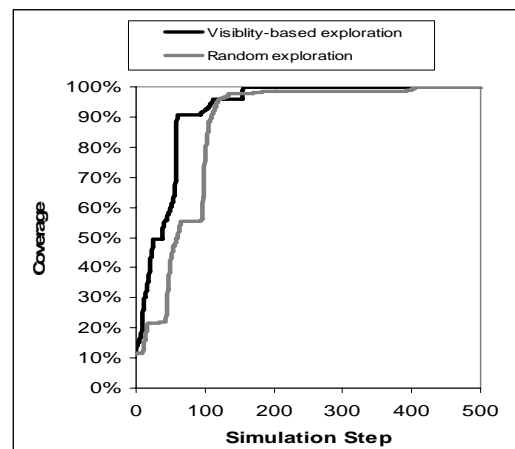
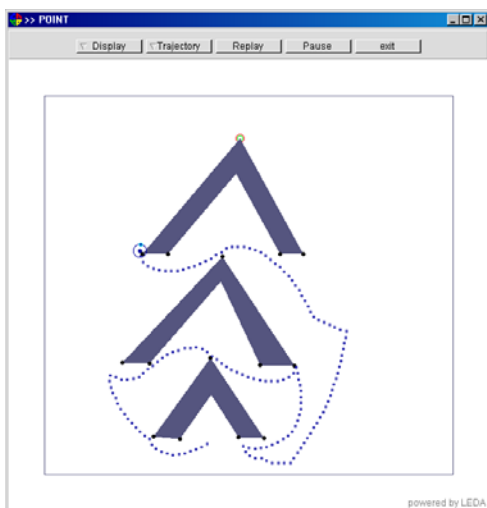


Fig. 12. Exploration performance



10. Visibility-based exploration trajectory

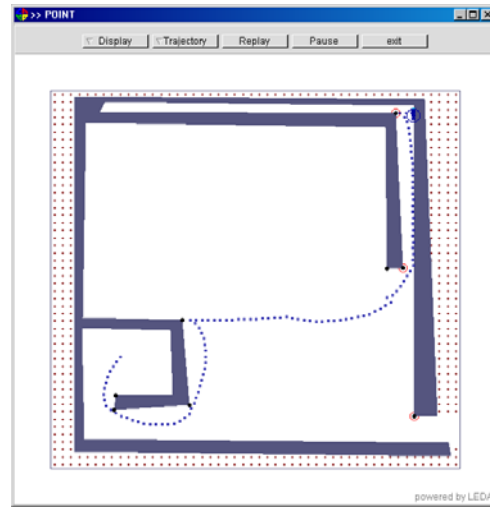


Fig. 13. Local minima problem

Simulation results (e.g., videos) can be found in <http://guppy.mpe.nus.edu.sg/~mpeangh/kevin/>.

5. Conclusion and Future Work

In this paper, our visibility-based exploration algorithm is introduced. Assuming the environment is structured that only contains polynomial obstacles, and the robot has infinite range line-of-sight sensors, we propose an exploration algorithm that lets the robot approach the critical occlusion point and modify the motion based on the prediction of sensible area. This reactive exploration algorithm can work based on local sensing without the need for detailed mapping. The efficacy is proved by simulations.

The advantage of our approach also leads to the limitation of our approach. While our visibility-based exploration algorithm greatly reduces the need for memory space and the complexity of computation, the exploration may not always be complete. This is due to the local minima problem that if the robot has no occlusion point in its visible region and possible next step to increase the coverage, it will stop exploration. To solve this problem, we propose two kinds of heuristics. One is to change the prediction model once the robot is captured by local minima. Enlarge the prediction circle radius or adjust the prediction direction may help the robot find better next step location to escape the local minima. Another solution is to let the robot move randomly until it finds a new occlusion point. We will implement and test these two heuristics in our future research to improve the performance of our visibility-based exploration algorithm.

Another meaningful work is to extend our visibility-based exploration algorithm to multi-robot systems. The requirement of exploration and the simplicity of our approach naturally satisfy the property of multi-robot systems. However, how to let the robots fuse their sensed data to cooperatively find occlusion points, and predict the changes of sensible area, is non-trivial research topic to be studied.

Reference

- [1] Brian Yamauchi, Alan Schultz and William Adams, Integrating Exploration and Localization for Mobile Robots, *Adaptive Behavior*, 7 (2), 217-229. 1999.
- [2] Sebastian Thrun, *Robotic Mapping: A Survey*, CMU-CS-02-111, February, 2002.
- [3] John W. Fenwick, Paul M. Newman, and John J. Leonard, Cooperative Concurrent Mapping and Localization, in proceedings of the International Conference on Robotics and Automation, pp.1810-1817, Washington, USA, May 2002.
- [4] Andrew Howard, Maja J. Mataric, and Gaurav S. Sukhatme, Mobile Sensor Network Deployment Using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem, in proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems, Fukuoka, Japan, June 25-27, 2002.
- [5] Zheng Liu, Marcelo H. Ang Jr., and Winston Khoon Guan Seah, Searching and Tracking for Multi-Robot Observation of Moving Targets, in proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS-8), March, 2004.
- [6] Brian Yamauchi, A Frontier-Based Approach for Autonomous Exploration, in proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, July 1997, pp. 146-151.
- [7] Brian Yamauchi, Frontier-Based Exploration Using Multiple Robots, in proceedings of the Second International Conference on Autonomous Agents (Agents '98), Minneapolis, MN, May 1998, pp. 47-53.
- [8] Benjamin Tovar, Luis Guilamo, and Steven LaValle, Gap Navigation Trees: A Minimal Representation for Visibility-based Tasks, in proceedings of the International Workshop on the Algorithmic Foundations of Robotics, 2004.
- [9] Tirthankar Bandyopadhyay, Yuanping Li, Marcelo H. Ang Jr., and David Hsu, Stealth Tracking of an Unpredictable Target among Obstacles, in proceedings of the International Workshop on the Algorithmic Foundations of Robotics, 2004.
- [10] Algorithmic Solutions Software GmbH. <http://www.algorithmic-solutions.com/index.htm>.