SCOUTING ALGORITHMS FOR FIELD ROBOTS

USING TRIANGULAR MESH MAPS

A Thesis

Submitted to the Faculty of Graduate Studies and Research
in Partial Fulfilment of the Requirement
for the Degree of

Doctor of Philosophy

In the Department of Agricultural and Bioresource Engineering,
University of Saskatchewan

By

Lifang Liu

Saskatoon, Saskatchewan

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a postgraduate degree from the University of Saskatchewan, I agree that the libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis. Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Agricultural and Bioresource Engineering

University of Saskatchewan

Saskatoon, Saskatchewan S7N 5A9

ABSTRACT

Labor shortage has prompted researchers to develop robot platforms for agriculture field scouting tasks. Sensor-based automatic topographic mapping and scouting algorithms for rough and large unstructured environments were presented. It involves moving an image sensor to collect terrain and other information and concomitantly construct a terrain map in the working field. In this work, a triangular mesh map was first used to represent the rough field surface and plan exploring strategies. A 3D image sensor model was used to simulate collection of field elevation information.

A two-stage exploring policy was used to plan the next best viewpoint by considering both the distance and elevation change in the cost function. A greedy exploration algorithm based on the energy cost function was developed; the energy cost function not only considers the traveling distance, but also includes energy required to change elevation and the rolling resistance of the terrain. An information-based exploration policy was developed to choose the next best viewpoint to maximise the information gain and minimize the energy consumption. In a partially known environment, the information gain was estimated by applying the ray tracing algorithm. The two-part scouting algorithm was developed to address the field sampling problem; the coverage algorithm identifies a reasonable coverage path to traverse sampling points, while the dynamic path planning algorithm determines an optimal path between two adjacent sampling points.

The developed algorithms were validated in two agricultural fields and three virtual fields by simulation. Greedy exploration policy, based on energy consumption outperformed other pattern methods in energy, time, and travel distance in the first 80% of the exploration task. The exploration strategy, which incorporated the energy consumption and the information gain with a ray tracing algorithm using a coarse map, showed an advantage over other policies in terms of the total energy consumption and the path length by at least 6%. For scouting algorithms, line sweeping methods require less energy and a shorter distance than the potential function method.

ACKNOLEDGEMENTS

I gratefully acknowledge the support and help of many people who assisted me in this project. I would like to thank my advisor, Dr. Trever Crowe, for his support and guidance during my Ph.D study. Trever allowed me to choose an interesting project and has given me encouragement during the research. His influence has contributed greatly to my development as a scientist. I would also like to thank the rest of my supervisory committee, including Dr. Martin Roberge, Dr. Venkatesh Meda, Dr. Claude Laguë, Dr. Erick Dupuis, and Dr. Daniel X. B. Chen for much helpful advice. My sincere thanks to Dr. Roberge for arranging my Canadian Space Agency (CSA) trip.

I am also indebted to the support of the CSA. Special note goes to Dr. Erick Dupuis, who kindly allowed me access to the facilities and software of the CSA and who has provided valuable ideas and advice on the research as my advisory committee member, Dr. J. Bakambu for his advice and help as my immediate supervisor in CSA, Mr. Pierre Allard, Mr. Sebastien Gemme, Dr. Ioannis Rekleitis, and other people in CSA who aided the research. The author also acknowledges the help of Dr. Guy Lafond, Agriculture and Agri-Food Canada, for providing GPS data of the field maps used in this work.

I also acknowledge the financial support from the Natural Sciences and Engineering Research Council (NSERC), the Canadian Space Agency (CSA), and the department of Agricultural & Bioresource Engineering.

Finally, I express my appreciation to my family for their understanding. My appreciation also extends to my friends for their support throughout my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY OF TERMS

accumulated path length	The total distance traveled from the start location to the current location; accumulated (summed) stepwise travel distances.
accumulated scan number	The number (count) of scans completed since the start location until the current location.
accumulated time	The total time required to progress from the start location to the current location; accumulated (summed) time for each of the steps in the overall path.
ASAE	American Society of Agricultural Engineering
CI	Cone Index: measured with a cone penetrometer as in ASAE S 313.2 (N/cm^2).
holes	Contiguous regions located entirely within the boundary of the field that are inaccessible and topographic data are not available.
NBV	Next best viewpoint: viewpoint chosen for a sensor scan based on a utility function at an iterative step during the exploration.
normalized relative energy requirement	Used in comparing algorithms, the nominal energy requirement is divided by the maximum value for each comparative set of conditions.
information gain	New terrain information which might be collected at a viewpoint by one sensor scan.
percentage of terrain explored	Fraction of the total terrain surface area; the explored surface area is divided by the total surface area.

1. INTRODUCTION AND PROBLEM STATEMENT

1.1 Introduction

Traditional agricultural production largely depends on application of pesticides and fertilizers. Canadian farmers alone consumed over 5 million tonnes of fertilizers in the year 2001/2002, while Saskatchewan accounted for 29% of total consumption (Korol, 2006). Economic incentive and public pressure to preserve and protect the environment have prompted agricultural producers to search for more efficient ways to manage chemical application. Site-specific crop management, including spatially-selective application of fertilizer and other chemicals, has the potential to increase profits and reduce the threat to the environment.

High-resolution variable maps, such as topographic, weed, and soil maps, play a critical role in precision agriculture. Field variable maps such as topographic maps or weed maps can be generated using aerial-based remote sensing methods (Bajwa and Tian, 2001) or ground-based measurement techniques. Air-based remote sensing techniques provide an efficient way to generate field variable maps over large-scale fields. However, the accuracy of maps generated from remotely sensed images can not achieve the required resolution for agricultural tasks (Bishop and McBratney, 2002).

The availability of the Global Positioning System (GPS) has prompted researchers to investigate vehicle-based methods, in which the GPS and other sensors are mounted on an agricultural vehicle to measure field variability while the vehicle moves within the field boundary (Westphalen et al., 2004; Saraswat et al., 2003; Schmidt et al., 2003; Yao and Clark, 2000; Clark and Lee, 1998; Adamchuk et al., 1999). However, all previous research required a driver to operate the vehicle through the fields, along a pre-defined path, such as a crop row or straight line direction. Human labor, which is relatively expensive in Canada, represents an input cost to the spatially-selective herbicide application. This will adversely affect the agricultural producer's motivation to implement this new technology.

Advances in new sensors, high-speed computers, and control technologies have prompted researchers to develop robot platforms for agricultural applications. A scouting robot equipped with an automatic guidance system and a sensor that can record geo-referenced field information would have the potential to alleviate the labor shortage problem. A few research groups have used robotic platforms to build a weed map or to conduct mechanical weeding at the same time as the weed map is created (Tillett and Hague, 1999; Nielsen et al., 2002; Åstrand and Baerveldt, 2002; Fontaine and Crowe, 2006); However, previous researchers have focused on finding a path relative to landmarks, such as crop rows (Tillett and Hague, 1999; Åstrand and Baerveldt, 2002; Fontaine and Crowe, 2006) or edges between cut and uncut crop (Ollis and Stentz, 1996) in an agricultural field. As far as the author knows, no study has considered exploration algorithms that focus on path planning in rough and unstructured environments for agricultural field scouting robots.

The intent of this work was to develop exploration and scouting algorithms to guide the robot to sample points while circumventing obstacles in the working field. Progress has been made towards developing surface coverage, scouting, and mapping algorithms in planetary exploration (Moorehead et al., 2001); exploring an office environment (Yamauchi, 1997); floor cleaning (Hofner and Schmidt, 1994); lawn mowing (Hicks and Hall, 2000), demining (Acar et al., 2003); and painting (Atkar et al., 2004). However, rough and large agricultural environments have enough peculiarities to make the proposed development project highly advanced and fraught with challenging research tasks.

Generally, a grid map is usually used to represent rough environments in robotic exploration. A triangular mesh map may be advantageous for such applications , because of its ability to represent a rich environment and generate a smoother path for robotic navigation. Efficiency is a critical aspect to exploration tasks, such as planetary exploration or agricultural scouting; therefore researchers have attempted to reduce the travel cost and the number of sensor readings to explore the whole environment. Current exploration algorithms tend to use travel distance as a factor to represent cost in exploration. However, energy cost might be a more suitable factor to represent travel cost in the rough environment. To reduce the number of sensor scans and overall cost, information gain, defined as new terrain information collected in one sensor scan, might be integrated in the utility function to improve the exploration efficiency.

Scouting tasks, such as automatic soil or weed map building, represent different types of exploration accompanied by topographic map building. It involves planning a reasonable path to reach all predefined sampling points and collect the related field information at each sampling point.

1.2 Objectives

The aims of this project were to develop reasonable automatic mapping algorithms for the construction of topographic maps for unknown or partially known rough agricultural fields and to develop scouting algorithms for field sampling. Specific objectives of the project were to

(1) develop a tool to simulate a 3D image sensor and exploration based on a triangular mesh map,
(2) develop a greedy mapping algorithm based on the minimum energy requirement,
(3) develop methods to estimate the new information gain based on the triangular mesh map and exploit the potential of considering the information gain in mapping strategies, and
(4) develop scouting algorithms for field sampling tasks.

The thesis is structured into eight chapters and is presented in a manuscript-style format. The first chapter is an introduction that presents the problem statement and objectives of the thesis project. Chapter 2 is a general literature review. Objective 1 is associated with Chapter 3, which is a manuscript on a two-stage greedy exploration algorithm based on the triangular mesh map and 3D image sensor model. Objective 2 is associated with Chapter 4, which focuses on developing the exploration strategy based on the energy function. Objective 3 is associated with Chapter 5, which is a manuscript on an information-based exploration strategy. The last objective is addressed in Chapter 6, which focuses on scouting algorithms. Chapter 7 and 8 present the general conclusions of the thesis project as well as recommendations for future work.

1.3 Hypothesis

This work focused on the exploration strategies, and it was assumed that there is zero error in localization estimation. In other words, the robot always knows exactly where it is within the field boundary. It was also assumed that fields are rough, large, and unstructured; there are no apparent landmarks, such as crop rows, in the fields. The field boundary is known, but there is no detailed terrain elevation information for the defined field.

A simulated four-wheel drive robot (mass: 16 kg; length: 50 cm; width: 49 cm; height: 26 cm) with four identical wheels (wheel diameter: 25.2 cm; wheel width: 7.5 cm) was used in the simulation. The robot's vision system was assumed to be a 3D image sensor (with a 90° field of view, 50 meter depth of field, and 1:1 aspect ratio). The image sensor could rotate 360˚ horizontally without any cost. The robot drives at a speed lower than 3 m/s.

1.4 References

Acar, E., H. Choset, Y. Zhang and M. Schervish. 2003. Path planning for robotic demining: robust sensor-based coverage of unstructured environments and probabilistic methods. *International Journal of Robotics Research* 22(7-8): 441-466.

Adamchuk, V. I., M. T. Morgan, and D. R. Ess. 1999. An automated sampling system for measuring soil pH. *Transactions of the ASAE* 42(4): 885-891.

Åstrand, B. and A. Baerveldt. 2002. An agricultural mobile robot with vision-based perception for mechanical weed control. *Automatic Robots* 13: 21-35.

Atkar, P., D. C. Conner, A. Greenfield, H. Choset and A. A. Rizzi. 2004. Uniform covreage of simple surfaces embedded in R^3 for auto-body painting. In *the Sixth*

Workshop on the Algorithmic Foundations of Robotics. Utrecht/Zeist, the Netherlands, July, 2004.

Bajwa, S. G. and L. F. Tian. 2001. Aerial CIR remote sensing for weed density mapping in a soybean field. *Transactions of the ASAE* 44(6): 1965-1974.

Bishop, T. F. A. and A. B. McBratney. 2002. Creating field−extent digital elevation models for precision agriculture. *Precision Agriculture* 3(1): 37-46.

Clark, R. L., and R. Lee. 1998. Development of topographic maps for precision farming with kinematic GPS. *Transactions of the ASAE* 41(4): 909-916.

Fontaine, V. and T.G. Crowe. 2006. Development of line-detection algorithms for local positioning in densely seeded crops. *Canadian Biosystems Engineering* 48: 7.19-7.29.

Hicks, R. W and E L. Hall. 2000. A survey of robot lawn mowers. In *Proceedings of the International Society for Optical Engineering* 4197: 262-269. San Diego, October 2000.

Hofner, C. and G. Schmidt. 1994. Path planning and guidance techniques for an autonomous mobile cleaning robot. In *Proceedings of the 1994 International Conference on Intelligent Robots and Systems (IROS)*: 610-617.

Korol, M. Accessed August 15, 2006. Canadian Fertilizer Consumption, Shipments and Trade 2001/2002. In http://www.agr.gc.ca/. Agriculture and Agri-Food Canada.

Moorehead, S., R. Simmons and W.L. Whittaker. 2001. Autonomous exploration using multiple sources of information. In *Proceedings of the IEEE International Conference on Robotics and Automation* 3: 3098- 3103. Seoul, Korea, May 21-26.

Ollis, M. and A. Stentz. 1996. First results in vision-based crop line tracking. In *Proceedings of the IEEE Robotics and Automation Conference* 951-956. Minneapolis, MN, April 22-28.

Saraswat, D., R. Ehsani, N. Watermeier and M. Sullivan. 2003. Potential application of yield data for creating topographic maps. In *2003 ASAE Annual Meeting.* Paper no: 031084. Las Vegas, Nev., July 27-30.

Schmidt, J. P., R. K. Taylor and R. J. Gehl. 2003. Developing topographic maps using a submeter accuracy global positioning receiver. *Applied Engineering in Agriculture* 19(3): 291-300.

Tillett, N. D. and T. Hague. 1999. Computer-vision-based hoe guidance for cereals — an initial trial. *Journal of Agricultural Engineering Research* 74(3): 225-236.

Westphalen, M. L., B. L. Steward and S. Han. 2004. Topographic mapping through measurement of vehicle attitude and elevation. *Transactions of the ASAE* 47(5): 1841-1849.

Yamauchi, B. 1997. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation:* 146-151. Monterey, CA. July 10-11.

Yao, H. and R. L. Clark. 2000. Development of topographic maps for precision farming with medium accuracy GPS receivers. *Applied Engineering in Agriculture* 16(6): 629-636.

2. LITERATURE REVIEW

2.1 Significance

As the brief introduction to field variable mapping and agricultural scouting robots presented in Chapter 1 reveals, it is necessary to develop automatic mapping algorithms for the construction of topographic maps and to develop scouting algorithms for field sampling. The large body of literature on robotic exploration and sensor coverage research forms the foundation of this project. The specific objective of this chapter is to review the details of the technologies exploited by automatic exploration, which include world models used in automatic exploration, coverage path planning, robotic exploration, topographic mapping of agricultural fields, and agricultural vehicle guidance.

2.2 World representation

Most exploring systems use a variety of terrain models to represent working environments and plan efficient coverage paths. An accurate map is important for robotic coverage and exploration because the coverage path can usually be generated more efficiently using a map (Choset, 2001). The map can also effectively store the related information about the covered areas (Moorehead et al., 2001). Several environmental models have been implemented, including certainty grids (Zelinsky et al., 1993; Moorehead et al., 2001; Sujan and Dubowsky, 2005; Thrun et al., 2005), polygonal layouts (Gonzalez-Banos and Latombe, 2002; Bourgault et al., 2002; Stachniss and Burgard, 2003; Taylor and Kriegman, 1998), topological maps (Choset and Nagatani, 2001; Wong and MacDonald, 2003), triangular grid maps (Oh et al., 2004; Dupuis et al., 2004), visibility graphs (Rao and Iyengar, 1990; Rao et al., 1988; Rao et al., 1993) and Voronoi diagrams (Rao et al., 1993).

Many studies used certainty grid maps due to the simplicity of this approach. A certainty grid map is composed of uniform grid cells, and a certainty value represents the probability of unknown, empty, and occupied status (Elfes, 1987). An unknown cell is a cell about which no information is available. Cells can be empty or occupied by an obstacle that is represented by a certainty value. A certainty grid map is a good choice for a system with noisy range sensors, such as sonar and laser range sensors, and has been used in a 2D environment (Yamauchi, 1997; Howard et al., 2006; Bourgault et al., 2002; Su and Tan, 2005; Simmons et al., 2000; Prestes et al., 2002). Several studies (Moorehead et al., 2001; Sujan and Dubowsky, 2005; Thrun et al., 2005) have investigated 3D grid maps for the exploration of rough environments.

The triangular grid map has advantages over square grid maps in its ability to generate a smoother path for navigation tasks (Oh et al., 2004). Every triangle has three edge neighbors and nine vertex neighbors, so it has twelve cell neighbors for a non-boundary triangle cell. The regular triangle mesh map provides twelve moving direction choices for each location, which results in a much smoother path compared to the traditional grid map. Oh et al. (2004) presented a triangular cell-based map representation that enabled a cleaning robot to explore 2D environments.

A triangle mesh is one of the most popular representations of object surfaces for computer graphics applications (Schroeder et al., 1996), and it is becoming a popular approach for modeling natural terrain. A triangle mesh map, a set of non-overlapping triangle cells where the vertices of the triangles are the input sample points, is generated by triangulation (Schroeder et al., 1996). Dupuis et al. (2004) used the triangular mesh map to represent a Mars-like environment and developed a start-goal path planning algorithm for a planetary exploration task. An algorithm to implement triangulation can be quite efficient and thus suitable for areas with a large number of samples. Furthermore, if further samples are obtained at a later date, they can be added to the already existing triangulation without having to triangulate all the previously existing samples in addition to the new samples. This makes it possible to efficiently perform a successive refinement on those areas where more detailed information is required.

Topological maps are more compact than grid maps; therefore, they have been extensively used as an alterative to the certainty grid in robotic exploration (Choset, 2001; Acar and Choset, 2002, Acar et al., 2002; Wong and MacDonald, 2003). Topological maps use a connectivity graph to represent the explored environment, where landmarks are marked as nodes in the graph and edges represent the connectivity relationship between neighbor landmarks. Accurate and consistent topological maps are often difficult to get and keep in large-scale environments, particularly if the sensor data are highly ambiguous.

The concept of a visibility graph was proposed by Rao and coworkers (Oommen et al., 1987; Rao and Iyengar, 1990; Rao et al., 1988; Rao et al., 1993). Visibility graph maps, as shown in Fig.2.1 (Rao et al., 1993), represent the environment using a graph, where all convex obstacle vertices are nodes and edges represent the line joining two nodes. The line may either correspond to an obstacle edge or connect two obstacles and does not intersect any obstacle polygon. The coverage problem can be implemented by traversing each edge in the graph. However, the visibility graph assumes that the obstacle edge is the only interest point and the sensor has the capability to see neighboring obstacles at anytime. In practice, this is unrealistic and cannot guarantee complete coverage of the whole area. Another problem with this method is that navigation along an obstacle boundary may be unreasonable and dangerous for robots.

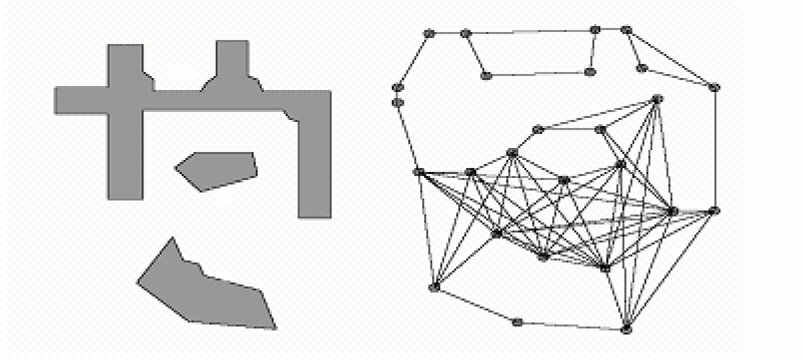

(a) (b)

Figure 2.1 Visibility graph: (a) world map; (b) visibility graph (Rao et al., 1993).

A Voronoi diagram of the terrain is the set of points that are closest to at least two points on the obstacle boundary (Rao et al., 1993). The Voronoi diagram as shown in Fig. 2.2 (Rao et al., 1993) is composed of straight and curved segments. The coverage path generated by a Voronoi diagram keeps the robot as far away from the obstacles as possible, so the path is safer than that generated by the visibility graph. Rao and Iyengar (1990) discussed an incremental approach to create a Voronoi diagram-like structure.

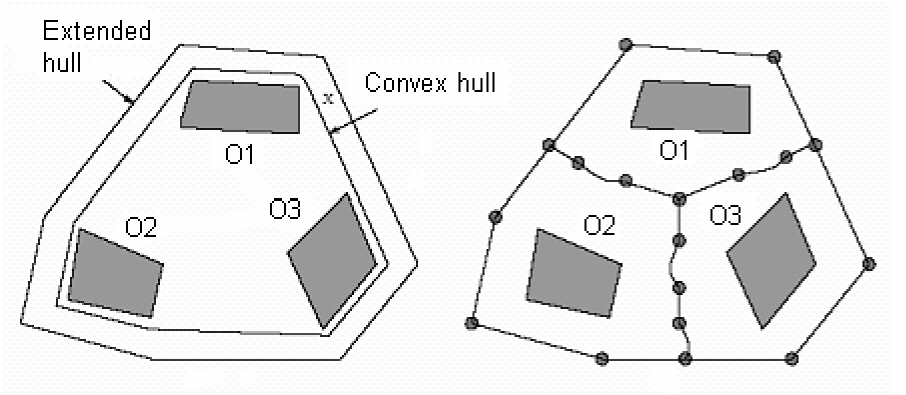


Figure 2.2 Voronoi diagram of the terrain (Rao et al., 1993).

2.3 Coverage path planning

A path for the purposes of surface coverage is a planned path in which a robot sweeps areas of free space, covering the greatest area possible, in an environment in an efficient manner. Coverage path planning has applications in floor cleaning (Hofner and Schmidt, 1994), lawn mowing (Hicks and Hall, 2000), demining (Acar et al., 2003), and painting (Atkar et al., 2004), and harvesting (Ollis and Stentz, 1996). Most of this work is limited to 2D environments. Pattern coverage and potential fields are the two most common approaches used to address complete coverage path planning.

One popular approach to field coverage has utilized pattern paths to explore the whole field (Choset, 2001; Hert et al., 1996; Huang, 2001). These algorithms take the following basic approach to generate a coverage path: the region to be covered is divided into subregions, a traveling-salesman algorithm (Choset, 2001) is applied to generate a sequence of subregions to visit, and a coverage path is generated from this sequence that covers each subregion in turn. All of these algorithms use a single line sweep in order to divide the coverage region into subregions, and these subregions are individually covered using a back and forth motion in rows perpendicular to the sweep direction. The limitation of such strategies is that the efficiency of these approaches is affected by the line sweeping direction.

A sensor-based line sweeping decomposition of unknown environments with convex obstacles was attempted by Cao et al. (1988) for a lawn mowing task. Extreme points of the convex obstacles were used to divide the working environment into small areas, which were individually covered using back and forth sweeping. Hert et al. (1996) proposed a line sweep terrain-covering algorithm for a planar underwater environment. The purpose of this study was to build a mosaic-like image of the ocean floor. The robot started at a point in the environment and moved along parallel straight lines to cover the given area. The depth-first order recursive procedure was used to guarantee smaller areas were covered only once. The algorithm was evaluated in terms of the path distance and the amount of memory required.

More recently, Choset and Pignon (1997) proposed the trapezoidal decomposition, as shown in Fig. 2.3(a) (Choset and Pignon, 1997), and the boustrophedon (the way of ox) cellular decomposition techniques, as shown in Fig. 2.3(b) (Choset and Pignon, 1997), to address the general environment with nonconvex obstacles. The robot's free space was divided into trapezoidal cells, so coverage of each cell could be achieved using simple back and forth motions. Coverage of the whole environment could be achieved by visiting each cell in the graph. Acar et al. (2002) introduced an on-line exact cellular decomposition using critical points. The critical point defined simple cells that could be covered by performing simple back and forth motions. A graph that had edges as the cells and nodes as the critical points was used to represent the topology of the Morse decomposition, therefore the sensor-based coverage was reduced to an incremental graph construction procedure. They verified their coverage algorithm and the critical point sensing method using a Nomad 200 robot equipped with a ring of 16 ultrasonic sensors.

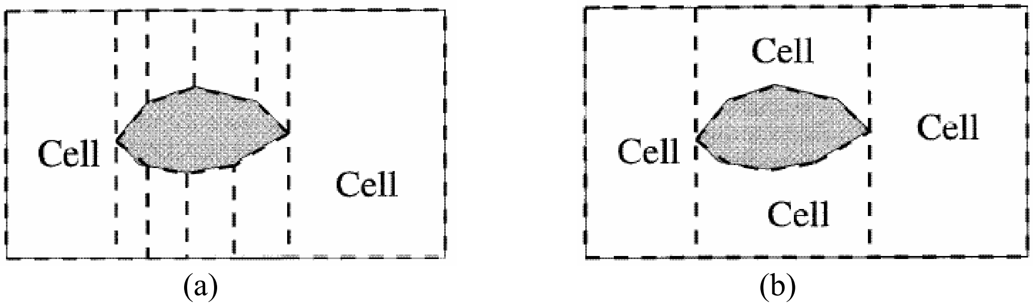


Figure 2.3 Cell decomposition: (a)Trapezoidal decomposition; (b)Boustrophedon decomposition (Choset and Pignon, 1997).

Oommen et al. (1987) proposed the visibility graph in their work. They presented an algorithm by which the robot moved along vertices of the obstacles and built the visibility graph of the terrain in a depth-first search manner. The visibility graph of terrain was defined as the union of vertices of all obstacles and edges such that the line joining two vertices did not intersect any obstacle. Rao and Iyengar (1990) extended the previous work to terrains with non-convex obstacles. Both the method used by Oommen et al. (1987) and the one used by Rao and Iyengar (1990) assumed that visibility was unlimited, which is rarely true in reality. Their algorithm was not verified by simulation nor was it implemented in actual robot navigation.

Two bug-style systematic techniques for exploring a 2D environment, the Sightseer and Seed Spreader strategies, were proposed by Lumelsky et al. (1990). They assumed that the robot was equipped with a tactile sensor and that localization was perfect. The sightseer algorithm, as shown in Fig. 2.4 (Lumelsky et al., 1990), required that all the obstacles be mutually visible from each other. The robot navigated towards the nearest visible obstacle and then circumnavigated it completely until no unvisited obstacles remained. The Seed Spreader algorithm, as shown in Fig. 2.5 (Lumelsky et al., 1990), divided the terrain into a number of equal-width strips, and the robot navigated around the strip to acquire information about the surrounding environment. The robot was required to deviate from the strip and circumnavigate an obstacle until the obstacle was completely defined.

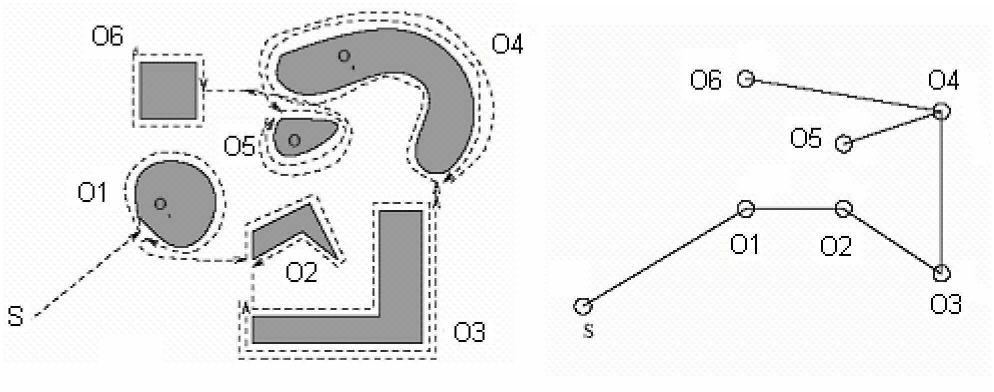


Figure 2.4 Sightseer strategy (Lumelsky et al., 1990).

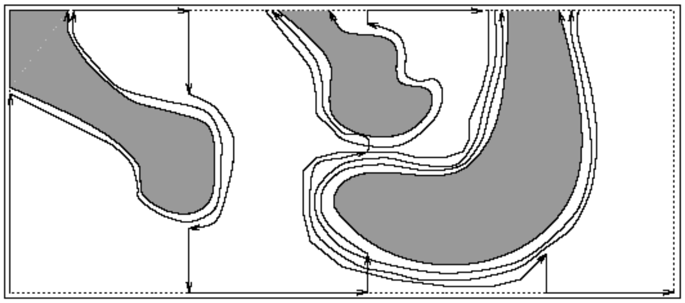


Figure 2.5 Seed spreader algorithm (Lumelsky et al., 1990).

To achieve optimal coverage, Huang (2001) adapted the planar line sweeping decomposition approach. It was assumed that minimizing the number of turns was the most important factor in an efficient solution. He showed that the optimal line sweeping decomposition, as shown in Fig. 2.6 (Huang, 2001), must use a sweep line that is parallel to an edge of the boundary, an obstacle, or a convex hull (the smallest convex set that includes the polygonal boundary) for a polygonal environment. Huang's algorithm required *a priori* knowledge of the environment and substantial computing ability for the robot.

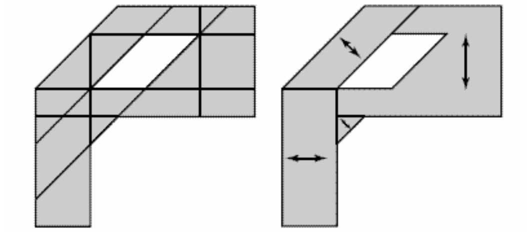


Figure 2.6 Huang's optimal line sweeping decomposition (Huang, 2001).

Gabriely and Rimon (1999) proposed a coverage algorithm based on traversing a minimum spanning tree of the coverage grid map. They assumed that localization was perfect and the range sensor could identify obstacles in the neighbor cells. The algorithm incrementally constructed a spanning tree for the grid, and then followed a subcell path that circumnavigated the spanning-tree edges. The result path was shown in Fig. 2.8 (Gabriely and Rimon, 1999). Their algorithm was evaluated in terms of both path length and memory requirement.

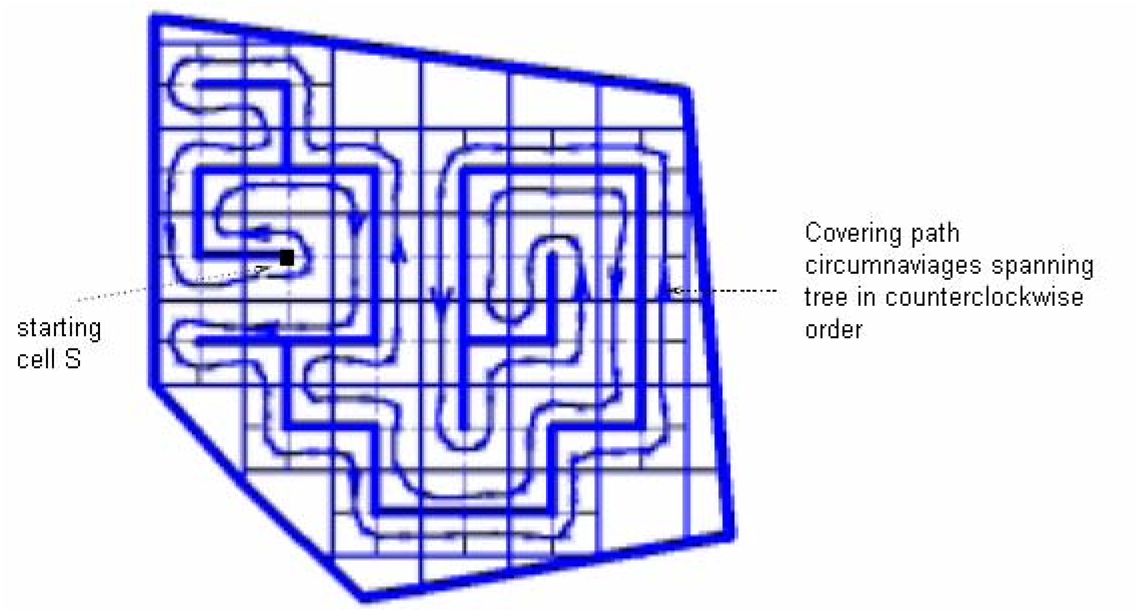

Covering path circumnaviages spanning tree in counterclockwise order
starting cell S

Figure 2.7 An execution example of the spanning tree coverage algorithm (Gabriely and Rimon, 1999).

Su and Tan (2005) proposed a robot exploration strategy following the spiral principle of a swirl expanse mode to explore a 2D indoor environment. The occupancy grid map was build with an omni-directional sonar system. At each step, the robot could choose from four possible directions. The priority of the four directions was set from high to low as follows: turning right 90°, keeping the direction unchanged, turning left 90°, turning left 180°. According to the principle of spiral motion, the robot started from its initial position, moved gradually surrounding the swirl center, and extended the detected “swirl area” step by step. Their algorithm was evaluated using both a single robot and multiple robots in simulation.

The potential field idea (Khatib, 1986; Koren and Borenstein, 1991) has been used to avoid obstacles in conventional start-goal path planning, in which the motion of the robot at any moment is determined by the potential function at its location. To avoid obstacles, Koren and Borenstein (1991) used the size of obstacles and the distance to the obstacles as a potential function to control the heading and velocity of a robot. The principle of a potential field has been applied in coverage path planning in many approaches (Zelinsky et al., 1993; Yang and Luo, 2004).

Zelinsky et al. (1993) investigated complete coverage based upon an extension to the distance transform path planning methodology with a known grid map. In their approach, a wave front was propagated that was a function of the distance from the goal. The coverage path, as shown in Fig. 2.8 (Zelinsky et al., 1993), was found using gradient descent based on this numeric potential field. Oh et al. (2004) proposed a similar distance-transform method to plan coverage paths for a cleaning robot. They used a triangular cell-based map representation that enabled a cleaning robot to have more navigation directions. The template method was combined with the distance-transform method to achieve complete coverage in an unknown environment.

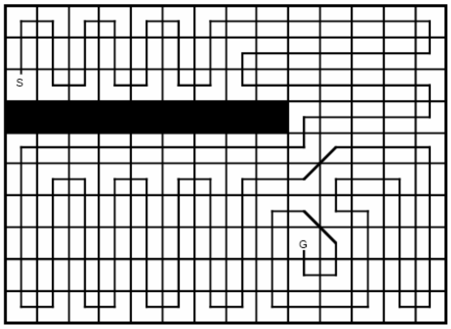


Fig. 2.8 Complete coverage path generated by Zelinsky’s algorithm (Zelinsky et al., 1993).

Yang and Luo (2004) developed a neural network approach, as shown in Fig. 2.9 (Yang and Luo, 2004), for complete coverage path planning of multiple cleaning robots in a 2D workspace. The workspace was divided into small grid cells called neurons and a neural network function decided the vehicle travel direction to achieve a coverage path. Unclean areas globally attracted the robot, whereas the obstacle areas just locally pushed the robot away to avoid collisions.

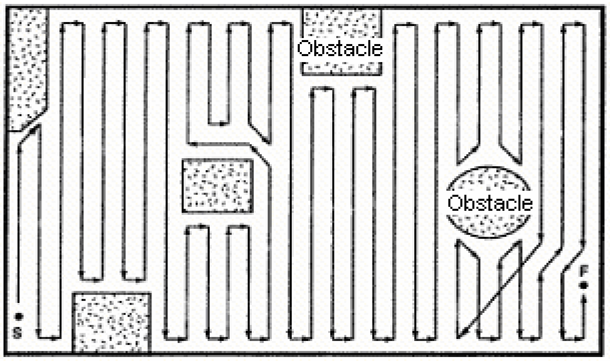


Fig. 2.9 Result of coverage path by Yang and Luo's neural network approach (Yang and Luo, 2004).

Acar et al. (2003) developed a probabilistic planner with a grid map for a demining task. For this planner, the highest priority area to visit was the area most likely to contain mines. The robot swept the region most likely to contain mines first. If the robot did not encounter a mine within a set time limit, the planner could then postulate that the cell was mine-free and direct the robot to another cell. The researchers developed a search algorithm to estimate the probable locations of remaining mines in order to determine where the search should proceed.

2.4 Robotic exploration and mapping

The main objective of exploration is to create an accurate map of an unknown area (Bourgault et al., 2002). To create an accurate map, a robot needs to know where it is, to navigate in the target environment, to collect terrain information, and to build the map that represents the environment. Therefore, exploration tasks require that robotic systems meet the objectives of both localization accuracy and exploration efficiency (Bourgault et al., 2002). Because the objectives of this thesis were to develop exploration algorithms and the ideal positioning was assumed, this review mainly concentrates on the exploration policies.

Many research groups have focused on structured indoor environments (Yamauchi, 1997; Howard et al., 2006; Bourgault et al., 2002; Kuipers and Byun, 1991; Sim and Dudek, 2003; Tovar et al., 2006; Thrun et al, 1998; Feder et al., 1999; Yamauchi, 1997; Simmons et al., 2000; Prestes et al., 2002; Choset and Nagatani, 2001; Gonzalez-Banos and Latombe, 2002). Other groups (Guivant et al., 2004; Moorehead et al., 2001; Thrun et al., 2005; Gerbaud et al., 2004) have attempted to develop exploration systems for outdoor environments.

To acquire a map, robot systems must possess sensors that enable them to perceive a variety of target environments. Many research groups used 2D laser range finders to explore a binary environment (Howard et al., 2006; Bourgault et al., 2002; Tovar et al., 2006; Victorino et al., 2003; Guivant et al., 2004; Yamauchi, 1997; Simmons et al., 2000; Gonzalez-Banos and Latombe, 2002). A few research groups used 2D laser range finders to explore a three dimensional environment (Thurn et al., 2005; Gerbaud et al., 2004), while others used 3D laser scanners (Surmann et al., 2003). There are also many systems equipped with sonar sensors (Kuipers and Byun, 1991; Su and Tan, 2005; Thrun et al., 1998; Feder et al., 1999; Prestes et al., 2002). Some researchers exploited image sensors such as stereoscopic cameras (Rocha et al., 2005; Lobo et al., 2003; Sujan and Dubowsky, 2005), a single camera (Sujan et al., 2006), and a combination of a camera and a sonar system (Sim and Dudek, 2003) to explore an indoor or outdoor environment.

One group of exploration strategies is to choose the closest next best viewpoint among frontiers (Yamauchi, 1997) extracted from the boundary between the known and unknown areas (Gonzalez-Banos and Latombe, 2002; Taylor and Kriegman, 1998; Thrun et al., 1998; Yamauchi, 1997). Yamauchi (1997) presented a frontier based robot explorer designed to explore an office environment using an occupancy grid map with a range sensor. The frontiers were defined by the known area close to the boundary between the familiar space and uncharted territory. The robot moved to the nearest frontier by the shortest path, collected a sensor reading, and updated the global map. The planner used a depth-first search on the grid to reach that frontier. The mapping exercise repeated this step until all the area was explored.

Other researchers chose the next best viewpoint using safety considerations. Prestes et al. (2002) proposed a frontier-based exploration approach based on a potential function using the harmonic functions (the solutions of the Laplace equation) method for path planning with a grid map. Obstacles generated repelling forces and frontiers attracted the robot. The robot planned the next best viewpoint using the harmonic potential calculation from an activation window, a local area with the size of sensing capacity. Following the gradient descent (similar to a greedy search) on the potential, and therefore minimizing the probability of collision, the robot will maximize knowledge gain while also avoiding obstacles. This method was implemented using a Nomad 200 platform with a sonar system and tested in a 400 in. × 330 in. 2D rectangular environment.

Gonzalez-Banos and Latombe (2002) introduced safe navigation constraints in an indoor environment exploration strategy. Their goal was to construct a 2D polygonal layout of the environment from the sets of points captured by a laser range sensor. The safe region was defined as the largest region that was guaranteed to be free of obstacles given a partially known map. The next best viewpoint was chosen within the safe region to avoid obstacles, under the constraint that the expected new information at this new position must have a minimal overlap with the current global map in order to maximize the expected gain of information. They showed that the next best viewpoint module produced strategies that cannot be easily out-done by a human operator.

A large body of study (Bourgault et al., 2002; Sujan et al., 2006; Rocha et al., 2005; Moorehead et al., 2001; Simmons et al., 2000; Sujan and Dubowsky, 2005) has centered on adaptive exploration policies based on the use of information as a measure of utility for taking exploration control actions. Bourgault et al. (2002) developed an exploration strategy based on adaptively selecting control actions that maximized both expected information gain and localization accuracy. Simmons et al. (2000) considered using multiple robots to explore using a utility based approach. The exploring robots chose the frontier that would provide the maximum new information gain and minimum driving cost.

Sujan et al. (2006) proposed an information-based iterative algorithm to plan the robot’s visual exploration strategy and efficiently build a graphic model of its environment. The mobile robot was equipped with a single camera to build a 2D panoramic image of the environment. Using a metric based on Shannon’s information theory, the algorithm determined potential locations of nodes from which to further image the environment. By tracing its path from node to node, a service robot can navigate around its environment.

Rocha et al. (2005) developed a frontier-based exploration strategy to build a 3D occupancy grid map for multi-robots equipped with stereo-vision sensors. A distributed architecture model was used to restrict the communication among robots to a minimum. An entropy concept representing uncertainty in the grid-based probabilistic map was defined as information gain to plan the next best viewpoint for the robots. The exploration strategy drives the robot to higher magnitudes of entropy gradient, where cells are more likely unoccupied.

Feder et al. (1999) proposed a feature-based stochastic mapping approach to the concurrent mapping and localization using a sonar system. An information metric named Fisher information was used to identify the next sensing positions, while maximizing the terrain information gain and minimizing expected dead-reckoning errors. This technique was validated by both simulations and physical experiments with land and underwater vehicles.

Tovar et al. (2006) developed optimal exploration strategies using a utility function to select sensing locations when exploring a 2D indoor environment with a laser range finder. A hybrid map constituted by polygons, landmarks, and a road-map was used in this work. The utility function integrated the travel distance, size of the unexplored space, robot configuration uncertainty, landmark identification probability, and ability to see features like corners. Unlike the traditional one step greedy method, this algorithm was designed to plan the ordering to visit all the current free edges. A decision tree was realized to facilitate the comparison of all possible paths. Their algorithm was evaluated with one robot and a team of robots. They concluded that generally traditional greedy exploration performs better in the path length, the total angle turned by the robot, the number of robot stops, and the number of sensing locations.

Generally, the previously described projects dealt with flat indoor environments that considered the traveling distance as the cost function. Much less work has been done in outdoor exploration. Thrun et al. (2005) developed a groundhog navigation system by acquiring 3D maps to explore abandoned mines. The Groundhog was equipped with two tiltable SICK laser range finders and employed the tilting mechanism to acquire 3D range scans of the area ahead of the robot.

Gerbaud et al. (2004) proposed a terrain exploration approach based upon Delaunay triangulation of the projections of acquired 3D points using a tilted laser range finder. Three exploration policies, reactive, goal-based and greedy, were developed in their system. In the reactive mode, the next best viewpoint was chosen in front of the platform. In the goal-based navigation mode, the next best viewpoint was selected by the supervisor. In the third exploration mode, the robot chose a frontier minimizing a cost function based upon its size and its distance to the platform.

Moorehead (2001) and Moorehead et al. (2001) proposed an exploration planner to integrate multiple sources of information in order to solve complex planetary exploration tasks. An information map was used to store multiple information sources within a 3D environment. This method enabled the robot explorer to maximize the total information gained while minimizing costs such as driving, sensing, and planning. An information map was used to store multiple information sources. The quality of the algorithm was demonstrated by creating traversability maps while exploring cliffs.

Surmann et al. (2003) presented an automatic system equipped with a 3D laser range finder for digitalization of 3D indoor environments. A next best viewpoint planner used an online greedy version based on the art gallery algorithm (Gonzalez-Banos and Latombe, 2002). The art gallery algorithm was modeled by several horizontal planes at different heights through the 3D scene. The next best viewpoint was chosen by maximizing the information gain while minimizing the path length.

Some researchers considered using multiple robots to map a large environment (Howard et al., 2006; Tovar et al., 2006; Rocha et al., 2005; Simmons et al., 2000). Howard et al. (2006) described a large heterogeneous mobile robot team, consisting of approximately eighty robots equipped with laser range sensors, that was employed to explore a 2D building environment using an occupancy grip map. In their system, each individual robot had an independent on-board localization and mapping algorithm to maintain an independent local pose estimate, while global pose estimates were generated by combining all the information from all robots through a second simultaneous localization and mapping (SLAM) algorithm. For exploration, a decentralized frontier-based approach with local occupancy grids and minimal communication between robots was used.

Although a variety of approaches have been developed, little work has focused on the problem of evaluating the relative performance of different strategies. The most common method is to present the planned path showing how a given environment might be covered by the algorithms (Zelinsky et al., 1993; Gonzalez-Banos and Latombe, 2002; Yang and, Luo, 2004; Gabriely and Rimon, 1999). Sim and Dudek (2003) examined several exploration trajectories to construct a visual map. They compared candidate robot exploration policies, which included SeedSpreader, Concentric, FigureEight, Random, Triangle, and Star polices. They found that the Star policy generated the most accurate map but was highly inefficient because it repeatedly traversed previously explored terrain, while the Random policy performed very well relative to the accumulated error.

In conclusion, most previous studies have focused on flat indoor or outdoor environments, where a region is simply regarded as traversable (free space) or unreachable (obstacle). The path distance has typically been used in the cost function to select the next best viewpoint to which the robot will move in order to explore new terrain. However, the exploration of agricultural fields is complicated by a number of issues. One of the most important issues is that agricultural field surfaces are usually rough. Travel distance alone is unsuitable to represent traveling cost in the rough terrain of an outdoor unstructured environment.

2.5 Topographic mapping of agricultural fields

The field topography affects soil characteristics, water flow, and crop yields (Westphalen et al., 2004; Saraswat et al., 2003). Improvements in sensing and computing technologies have enabled the development of digital representations of topography as a layer in geographic information systems (GIS) used for precision agriculture.

Incorporating topographic maps into site-specific management decisions will rely on cost-effective methods of obtaining a sufficiently accurate elevation map (Schmidt et al., 2003). For large land areas, aerial survey techniques have been used for some time as an economical method for developing topographic maps (Clark and Lee, 1998). However, aerial survey techniques may be less cost-effective for small areas and their accuracy may depend on the resolution of the images taken (Westphalen et al., 2004). Therefore, ground-based mapping methods, using a vehicle equipped with a GPS receiver, have become the topic of active research in recent years.

Clark and Lee (1998) described a system using real-time kinematic differential GPS to collect GPS data from a moving vehicle for the development of topographic maps. They used two sampling modes, stop-and-go (using a tripod GPS antenna mount) and kinematic (with a tractor-mounted GPS antenna). Sampling locations were defined by a systematic sampling method with 3-meter spacing. The collected data from both the rover and base station were downloaded to a computer and post processed with dedicated software. Results showed that kinematic GPS can quickly produce a large number of accurate data points from which an accurate topographic map can be developed.

To reduce the high cost of kinematic differential GPS, a few researchers attempted to utilize sub-meter accuracy GPS receivers to build topographic maps of fields. Yao and Clark (2000) proposed that sub-meter accuracy GPS receivers could be used, while repeatedly following the same path, to develop topographic maps. An all-terrain vehicle was driven across a field in straight parallel lines, with line spaces of approximately 3 to 5 m. The results showed that ten passes along the same path were required to develop an accurate topographic map. One particular difficulty was that it was not easy to repeatedly follow the same path. Schmidt et al. (2003) developed topographic maps using sub-meter GPS data that were collected during routine crop management passes during multiple years. Elevation data were collected from multiple passes using a typical DGPS receiver during routine planting, spraying, and harvesting field operations between 1999 and 2001. Elevation maps were created by a simple linear regression using the Spatial Analyst Extension in ArcView. Although this research demonstrated that an elevation map could be created using such a process, the assumption of a static environment may not always be true.

Westphalen et al. (2004) demonstrated that an inertial measurement unit (IMU) could be integrated with GPS to create an accurate digital elevation model (DEM) during typical field operations. They developed a self-propelled agricultural sprayer equipped with four RTK DGPS receivers and an IMU to build a field elevation map of a 2.3 ha area. Both stop-and-go and kinematic data collection modes were used to collect the elevation data in the field. The sprayer was driven along north-south paths to traverse the whole field and collect data in 3-m intervals. They concluded that the DEMs generated with attitude measurements had a higher accuracy than those generated without attitude measurements.

Yokota et al. (2004) proposed the concept of developing a method of 3D map generation using a robot tractor equipped with a laser range finder, real-time kinematics GPS, and a fiber-optic gyroscope. The robotic tractor moved to different locations and collected different local terrain information, which was integrated into a global map. Sugiura et al. (2004) introduced a different approach to represent terrain, using triangular polygons. The triangular polygon map was obtained, using a one-axis laser range finder mounted on an unmanned helicopter.

Besides topographic maps, other variables such as soil characteristic have been integrated into a field information map. Christy et al. (2004) proposed a mobile sensor platform to measure soil pH and electrical conductivity. The system automatically retrieved a soil sample and pressed it against an ion-selective electrode for analysis as the system moved through the field.

The vast majority of previous research requires a driver to drive the vehicle through the fields. Alternatively, research efforts (Christy et al., 2004; Adamchuk et al., 1999) have considered automatic mapping using a robotic platform, but several problems remain. One persistent problem involves developing the agricultural field coverage algorithm.

2.6 Agricultural vehicle guidance

The idea of automatic scouting is closely related with automatic agriculture vehicle guidance. The development of agricultural vehicle guidance before 2000 can be found in Japan (Torri, 2000), Europe (Keicher and Seufert, 2000), and the USA (Reid et al., 2000). Most research in this phase focused on identifying the landmarks such as crop rows or uncut/cut edges of the crops and guiding the vehicle follow the landmarks.

Reid and Searcy (1987) began development of a vision-based guidance system for steering a tractor through row crops. A camera was used to grab near infrared images and a Bayesian classifier was used to segment row crops into crop and soil. Regression equations representing crop row locations in image were found using an unsupervised classifier that clustered pixels based on the distance to a projection of the regression line passing through classified pixels. Ollis and Stentz (1996) proposed a vision-based guidance system for a hay windrower following the edge of the uncut crop. A color camera on either side of the vehicle was used to sensor the edge of cut and uncut vegetation. A classifier based on RGB features of the color images was used to segment images into cut and uncut regions. The researchers successfully harvested approximately one acre of alfalfa autonomously. Their average speed while harvesting was approximately 3 miles per hour. Tillett et al. (1998), Tillett and Hague (1999), Hague et al. (2000), Nielsen et al. (2002), and Åstrand and Baerveldt (2002) separately developed robotic weed mapping systems. In each of these systems, a charge-coupled device was located centrally at the front of the vehicle looking forward and down in order to provide images of crop rows. The images were utilized to provide guidance information.

Subramanian and Burks (2005) investigated automatic tractor guidance using a CCD camera and a laser finder for use in a citus grove. The camera and the laser sensor were mounted on the top of the tractor at an angle of 45 degrees looking forward in front of the vehicle. The vehicle was guided by a PID controller to follow the path between citrus rows identified by the vision sensors. They concluded that the laser sensor performed slightly better than the camera in terms of path tracking error when the vehicle traveled less than 3.1 m/s.

More recently, researchers have worked to integrate obstacle avoidance into the automatic guidance systems. Guo et al. (2002) explored detection of moving obstacles using two ultrasonic sensors. Their alert system had the ability to detect a moving object within 11 meters from the vehicle. Wei et al. (2005) developed an obstacle detection system using a binocular stereovision camera to improve the safety of the GPS navigation system. The camera was mounted on top of the tractor at a tilt angle of 15.5 degrees. A height threshold was used to classify the potential obstacles and the background field. The developed system could determine the relative moving speed and heading direction between the obstacle and the vehicle.

Other researchers have started to develop path planning and obstacle avoidance techniques, using a map. Gray (2000) attempted various obstacle-avoidance algorithms using a grid map for agricultural tractor navigation. He implemented the enhanced Vector Field Histogram (VFH) algorithm (Koren and Borenstein, 1991) as a local obstacle avoidance technique in agricultural tractor navigation. Rovira-Más et al. (2006) developed a 3D map, using stereovision for agricultural vehicle applications. A 3D density map, defined as the number of stereo-matched points per volume unit, was used to represent the field elevation information and detect obstacles. A * algorithm was used to plan an optimal path for the start-goal path planning application.

Coverage path planning is becoming a research topic for agricultural engineering researchers. Oksanen et al. (2005) explored the coverage path planning problem for agricultural applications. The problem was divided into two levels; in the higher level the complex shaped field was split into smaller parts based on trapezoidal split and merge (Choset 2001); in the lower level the path was planned in a line sweeping direction. The direction was chosen by using a search algorithm, which evaluated each possible direction based on the efficiency, area, and distance for each block. Jin and Tang (2006) optimized the sweeping direction by minimizing the number of turns and the cost of turns. The cost of turns included the wasted area in the headland and wasted distance of the turn.

2.7 Conclusions

In conclusion, numerous applications in automatic exploration and coverage path planning have been developed. Although many of these have been implemented under less demanding circumstances, it is reasonable to believe that similar solutions are relevant for agricultural field exploration. Agricultural field environments present many peculiarities and represent significant challenges for the development and use of autonomous vehicles.

Previous studies have focused on flat indoor or outdoor environments, where a region is simply regarded as traversable (free space) or unreachable (obstacle). A 2D or 3D occupancy grid map is the most common type of terrain model for exploration and coverage tasks. Previous study identified that a triangular mesh map might be a good choice to represent 3D rough environments and used for path planning because of its efficiency in representing large environments, and ability to generate a smoother path. However, there is no reported exploration algorithm based on a triangular mesh map. The attempt to develop an exploration algorithm based on a triangular mesh map should contribute to the engineering knowledge in the robotic community.

The path distance has typically been used in the cost function to select the next best viewpoint to which the robot will move in order to explore new terrain. However, travel distance alone is unsuitable to represent traveling cost in the rough terrain of an outdoor unstructured environment. Development of an energy cost function should be a step to address the knowledge gap within the robotic research community.

Although a variety of exploration approaches have been attempted, little work has focused on the problem of evaluating the relative performance of different strategies. The most common method was to present the planned path showing how a given environment might be covered by the algorithms.

Previous researchers in agricultural robotics focused on identifying landmarks such as crop rows or edges between cut and uncut crops, and guiding the vehicle to follow the landmarks in an agricultural field. Little research effort has been invested in guiding a vehicle to build a topographic map, or to scout a field, in an open, rough area without apparent landmarks but with obstacles.

2.8 References

Acar, E., H. Choset, A. Rizzi, P. Atkar and D. Hull. 2002. Morse decompositions for coverage tasks. *International Journal of Robotics Research* 21(4): 331-344.

Acar, E. and H. Choset. 2002. Sensor-based coverage of unknown environments: incremental construction of Morse decompositions. *International Journal of Robotics Research* 21(4): 345-366.

Acar, E., H. Choset, Y. Zhang and M. Schervish. 2003. Path planning for robotic demining: robust sensor-based coverage of unstructured environments and probabilistic methods. *International Journal of Robotics Research* 22(7-8): 441-466.

Adamchuk, V. I., M. T. Morgan and D. R. Ess. 1999. An automated sampling system for measuring soil pH. *Transactions of the ASAE* 42(4): 885-891.

Åstrand, B. and A. Baerveldt. 2002. An agricultural mobile robot with vision-based perception for mechanical weed control. *Automatic Robots* 13: 21-35.

Atkar, P., D. C. Conner, A. Greenfield, H. Choset and A. A. Rizzi. 2004. Uniform covreage of simple surfaces embedded in R^3 for auto-body painting. In *the Sixth*

Workshop on the Algorithmic Foundations of Robotics. Utrecht/Zeist, the Netherlands, July, 2004.

Bourgault, F., A.A. Makarenko, S.B. Williams, B. Grocholsky and H.F. Durrant-Whyte. 2002. Information based adaptive robotic exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* 1*:* 540-545. Lausanne, Switzerland, Sep 30-Oct 5, 2002.

Cao, Z. L., Y. Huang and E. Hall. 1988. Region filling operations with random obstacle avoidance for mobile robots. *Journal of Robotic System* (1988): 87-102

Choset, H. 2001. Coverage for robotics—A survey of recent results. *Annals of Mathematics and Artificial Intelligence* 31: 113-126.

Choset, H. and K. Nagatani. 2001. Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation* 17(2): 125-137.

Choset, H. and Pignon, P. 1997. Coverage path planning: the boustrophedon cellular decomposition. In *Proceedings of International Conference on Field and Service Robotics*. Canberra, Australia.

Christy, C., K. L. Collings, P. D. Drummond and E. D. Lund. 2004. A mobile sensor platform for measurement of soil pH and buffering. In *2004 ASAE Annual Meeting*. ASAE Paper No. 041042. St. Joseph, Michigan.

Clark, R. L. and R. Lee. 1998. Development of topographic maps for precision farming with kinematic GPS. *Transactions of the ASAE* 41(4): 909-916.

Dupuis, E., P. Allard, J. Bakambu, T. Lamarche and W·H. Zhu. 2004. Towards autonomous long-range navigation. In *the 8th ESA Workshop on Advanced Technologies for Robotics and Automation 'ASTRA 2004'*, ESTEC. Noordwijk, The Netherlands. November 2-4.

Elfes, A. Sonar-based real-world mapping and navigation. 1987. *IEEE Journal of Robotics and Automation*, RA3(3):249-265.

Feder,H.J.S., J.J. Leonard and C.M. Smith. 1999. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research* 18(7):650-668.

Gabriely, Y. and E. Rimon. 1999. Spanning-tree based coverage of continuous areas by a mobile robot. *Technical report*. Department of Mechanical Engineering, Technion, Israel Institute of Technology.

Gerbaud, T., V. Polotski and P. Cohen. 2004. Simultaneous exploration and 3D mapping of unstructured environments. In *Proceedings—IEEE International Conference on Systems, Man and Cybernetics*, *SMC 2004*: 5333-5337

Gray, K. W. 2000. Obstacle detection and avoidance for an autonomous farm tractor. Master's thesis, Utah State University, Logan, Utah

Gonzalez-Banos, H.H. and J.C. Latombe. 2002. Navigation strategies for exploring indoor environments. In *International Journal of Robotics Research* 21(10-11): 829-848.

Guivant, J., E. Nebot, J.Nieto and F. Masson. 2004. Navigation and mapping in large unstructured environments. In *International Journal of Robotics Research* 23 (4-5): 449-472

Guo, L, Q. Zhang, S. Han. 2002. Agricultural machinery safety alert system using ultrasonic sensors. *Journal of Agricultural Safety and Health* 8(4): 385-96.

Hague, T., J. A. Marchant and N. D.Tillett. 2000. Ground based sensing systems for autonomous agricultural vehicles. *Computers and Electronics in Agriculture* 25 (1-2): 11-28.

Hert, S., S. Tiwari and V. Lumelsky. 1996. A terrain covering algorithm for an AUV. *Autonomous Robots* 3: 91-119.

Hicks, R. W and E L. Hall. 2000. A survey of robot lawn mowers. In *Proceedings of the International Society for Optical Engineering* 4197: 262-269. San Diego, October 2000.

Hofner, C. and G. Schmidt. 1994. Path planning and guidance techniques for an autonomous mobile cleaning robot. In *Proceedings of the 1994 International Conference on Intelligent Robots and Systems (IROS)*: 610-617.

Howard, A., L. Parker and G.S. Gaurav. 2006. Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection. *International Journal of Robotics Research* 25(5-6): 431-447.

Huang, W. 2001. Optimal line-sweep-based decompositions for coverage algorithms. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*: 27- 32. Seoul, Korea, May 21-26.

Jin, J., L. Tang. 2006. Optimal path planning for arable farming. In *2006 ASAE Annual Meeting*. Paper no: 061158. Potland Oregon, July 9-12.

Keicher, R. and H. Seufert. 2000. Automatic guidance for agricultural vehicles in Europe. *Computers and Electronics in Agriculture* 25: 169-194.

Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research* 5(1):90-98.

Koren,Y., J. Borenstein. 1991. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the IEEE Conference on Robotics and Automation*: 1398-1404. Sacramento, CA.

Kuipers, B. and Y.T. Byun. 1991. Robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems* 8(1-2): 47-63.

Lobo, J., C. Queiroz and J. Dias. 2003. World feature detection and mapping using stereovision and inertial sensors. *Robotics and Autonomous Systems* 44(1): 69-81.

Lumelsky, V., S. Mukopadhyay and K. Sun 1990. Dynamic path planning in sensor-based terrain acquisition. *IEEE Transactions on Robotics and Automation* 6: 462-472.

Moorehead, S. 2001. Autonomous surface exploration for mobile robots. *Doctoral dissertation*, technical report. CMU-RI-TR-01-30. Robotics Institute, Carnegie Mellon University.

Moorehead, S., R. Simmons and W.L. Whittaker. 2001. Autonomous exploration using multiple sources of information. In *Proceedings of the IEEE International Conference on Robotics and Automation* 3: 3098-3103. Seoul, Korea, May 21-26.

Nielsen, K.M., Andersen, P., Pedersen, T.S., Bak, T., and Nielsen, J.D. 2002. Control of an Autonomous Vehicle for Registration of Weed and Crop in Precision Agriculture. In *IEEE Conference on Control Applications CCA/CACSD 2002*, Glasgow Scotland.

Oh, J., Y. Choi, J. Park and Y. F. Zheng. 2004. Complete coverage navigation using of cleaning robots using triangular cell based map. *IEEE Transactions on Industrial Electronics* 51(3): 718-726.

Oksanen, T., S. Kosonen, A. Visala, 2005. Path planning algorithm for field traffic. In *2005 ASAE Annual Meeting*. Paper no: 053087. Tampa FL, July 17-20.

Ollis, M. and A. Stentz. 1996. First results in vision-based crop line tracking. In *Proceedings of the IEEE Robotics and Automation Conference* 951-956. Minneapolis, MN, April 22-28.

Oommen, J.B., S.S. Iyengar, N.S.V. Rao, and R.L. Kashyap. 1987. Robot navigation in unknown terrains using learned visibility graphs. Part I: The disjoint convex obstacle case. *IEEE Journal of Robotics and Automation* 6(3): 672-681.

Prestes, E., P.M. Engel,M. Trevisan and M.A.P. Idiart. 2002. Exploration method using harmonic functions: *Robotics and Autonomous Systems* 40(1): 25-42.

Rao, N. S. V. and S. S. Iyengar. 1990. Autonomous robot navigation in unknown terrains: visibility graph based methods. *IEEE Transactions on Systems, Man and Cybernetics* 20(6): 1443-1449.

Rao, N. S., V. S. S. Iyengar, B. J. Oommen and R. L. Kashyap. 1988. On terrain acquisition by a point robot amidst polyhedral obstacles. *IEEE Journal of Robotics and Automation* 4(4): 450-455.

Rao, N. S., V. S. Kareti, W. Shi and S. S. Iyengar. 1993. Robot navigation in unknown terrains: introductory survey of non-heuristic algorithms. Technical report. ORNL/TM-12410. Oak Ridge National Laboratory, Oak Ridge, TN.

Reid, J.F. and S.W. Searcy. 1987. Vision-based guidance of an agricultural tractor. *IEEE Control Systems* 7 (12): 39-43.

Reid, J. F., Q. Zhang, N. Noguchi and M. Dickson. 2000. Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture* 25: 155-167.

Rocha, R., J. Dias and A. Carvalho. 2005. Cooperative multi-robot systems: a study of vision-based 3-D mapping using information theory. *Robotics and Autonomous Systems* 53(3-4): 282-311.

Rovira-Más, F., J. F. Reid, Q. Zhang. 2006. Stereovision data processing with 3D density maps for agricultural vehicles. Transactions of the ASABE 49(4): 1213-1222.

Saraswat, D., R. Ehsani, N. Watermeier and M. Sullivan. 2003. Potential application of yield data for creating topographic maps. In *2003 ASAE Annual Meeting.* Paper no: 031084. Las Vegas, Nev., July 27-30.

Schmidt, J. P., R. K. Taylor and R. J. Gehl. 2003. Developing topographic maps using a submeter accuracy global positioning receiver. *Applied Engineering in Agriculture* 19(3): 291-300.

Schroeder, W., K. Martin and B. Lorensen. 1996. The Visualization Toolkit: an object-oriented approach to 3D graphics. Prentice Hall PTR. Upper Saddle River, NJ.

Sim, R. and G. Dudek. 2003. Effective exploration strategies for the construction of visual maps. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*: 3224-3231. Las Vegas, NV, October 27-31.

Simmons, R., D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun and H. Younes. 2000. Coordination for multi-robot exploration and mapping. In *Proceedings National Conference on Artificial Intelligence.* Austin, TX.

Stachniss, C. and W. Burgard. 2003. Exploring Unknown Environments with Mobile Robots using Coverage Maps. In *Proceedings of the International Joint Conference on Artificial Intelligence(IJCAI)*: 1127-1134. Acapulco, Mexico. August 9-15.

Su, L. and M.Tan. 2005. A virtual centrifugal force based navigation algorithm for explorative robotic tasks in unknown environments**.** *Robotics and Autonomous Systems* 51 (4): 261-274.

Subramanian, V. and T.F. Burks. 2005. Autonomous path navigation in citrus groves using machine vision and laser radar. In *2005 ASAE Annual International Conference*. Paper no: 051142. Tampa FL, July 17-20.

Sugiura, R., K. Ishii and N. Noguchi. 2004. Simplification method of topographic map using triangle polygons. In *Proceedings of Automation Technology for Off-Road Equipment Conference*. Kyoto, Japan.

Sujan,V. A. and S. Dubowsky. 2005. Efficient information-based visual robotic mapping in unstructured environments. *International Journal of Robotics Research* 24 (4): 275-293.

Sujan, V.A., M.A. Meggiolaro and F.A.W. Belo. 2006. Information based indoor environment robotic exploration and modeling using 2-D images and graphs**.** *Autonomous Robots* 21(1): 15-28.

Surmann, H., A. Nüchter and J. Hertzberg. 2003. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems* 45: 181-198.

Taylor, C. and D. Kriegman. 1998. Vision-based motion planning and exploration algorithms for mobile robots. *IEEE Transactions on Robotics and Automation* 14(3):147-427.

Tillett, N. D. and T. Hague. 1999. Computer-vision-based hoe guidance for cereals—an initial trial. *Journal of Agricultural Engineering Research* 74(3): 225-236.

Tillett, N.D., T. Hague and J. A. Marchant. 1998. A Robotic system for plant-scale husbandry. *Journal of Agricultural Engineering Research* 69(2): 169-178.

Thrun, S., D. Fox and W. Burgard. 1998. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning* 31:29-53.

Thrun, S., S. Thayer. W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverte and W. Whittaker. 2005. Autonomous Exploration and Mapping of Abandoned Mines. *IEEE Robotics and Automation Magazine* 11(4): 79-91.

Torii, T. 2000. Research in autonomous agriculture vehicles in Japan. *Computers and Electronics in Agriculture* 25(1-2): 133-153

Tovar, B., L. Munoz-Gomez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy and S. Hutchinson. 2006. Planning exploration strategies for simultaneous localization and mapping. *Robotics and Autonomous Systems* 54 (4): 314-331.

Victorino, A., P. Rives and J.J. Borrelly. 2003. Safe navigation for indoor mobile robots. part II: exploration, self-localization and map building. *International Journal of Robotics Research* 22(12): 1019-1039.

Wei, J., F. Rovira-Mas, J. F. Reid, S. Han. 2005. Obstacle detection using stereo vision to enhance safety of autonomous machines. In *2005 ASAE Annual Meeting*. Paper no: 055006. Tampa FL, July 17-20.

Westphalen, M. L., B. L. Steward and S. Han. 2004. Topographic mapping through measurement of vehicle attitude and elevation. *Transactions of the ASAE* 47(5): 1841-1849.

Wong, S. C. and B. A. MacDonald. 2003. A topological coverage algorithm for mobile robots. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NEV.

Yamauchi, B. 1997. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation:* 146-151. Monterey, CA. July 10-11.

Yang, S.X. and C. Luo. 2004. A neural network approach to complete coverage path. In *IEEE Transactions on Systems, Man, and Cybernetics, Part B.* 34(1): 718-724.

Yao, H. and R. L. Clark. 2000. Development of topographic maps for precision farming with medium accuracy GPS receivers. *Applied Engineering in Agriculture* 16(6): 629-636.

Yokota, M., A. Mizushima, K. Ishii and N. Noguchi. 2004. 3-D map generation by a robot tractor equipped with a laser ranger finder. In *Proceedings of the Automation Technology for Off-Road Equipment Conference*. Kyoto, Japan, October 2004.

Zelinsky, A., R.A. Jarvis, J.C. Byrne and S.Yuta. 1993. Planning paths of complete coverage of an unstructured environment by a mobile robot. In *Proceedings of International Conference on Advanced Robotics*: 533-538. Tokyo, Japan.

3. VISION-BASED EXPLORATION ALGORITHMS FOR ROUGH TERRAIN MODELING USING TRIANGULAR MESH MAPS

3.1 Significance

This chapter relates to objective 1 of the thesis as stated in Section 1.2. The review of world models in the previous chapter shows that a triangular mesh map can be used to represent a rough terrain and it has several advantages over other world models. The specific objective of this chapter is to investigate exploration algorithms based on a triangular mesh map model and a 3D image sensor model. The research in this chapter provides a platform to investigate exploration algorithms based on a triangular mesh map and a 3D image sensor model.

In this chapter, a triangular mesh map representing the rough agricultural field surface is introduced. A 3D image sensor model was developed for the simulation. Finally, a two-stage exploring policy was used to plan the next best viewpoint by considering both the distance and the slope factor in the cost function. A comparison of the performance of the two-stage exploration strategy with that of a line sweeping method is discussed.

3.2 Introduction

High-resolution topographic maps play a critical role in precision agriculture. Elevation change within the agricultural field influences crop yields by affecting soil characteristics and water flow (Westphalen et al., 2004; Saraswat et al., 2003); therefore, many studies have been conducted on generating topographic maps for agricultural fields (Bishop and McBratney, 2002; Westphalen et al., 2004; Saraswat et al., 2003; Schmidt et al., 2003).

A topographic map can be generated using an air-based remote sensing method or a ground-based measurement method. However, the accuracy of maps generated from remotely sensed images is insufficient to meet the requirements of many agricultural tasks (Bishop and McBratney, 2002). The availability of the Global Positioning System (GPS) has prompted researchers to investigate the vehicle-based method, in which a GPS unit and other sensors are mounted on an agricultural vehicle to measure the field variability while the vehicle moves around the working field (Bishop and McBratney, 2002; Westphalen et al., 2004; Saraswat et al., 2003; Schmidt et al., 2003; Yao and Clark, 2000; Clark and Lee, 1998; Adamchuk et al., 1999). However, all previous studies required a driver to drive the vehicle through the fields along a pre-defined path such as a crop row or in a straight line in one direction, which is labor intensive work.

The advances in robotic exploration may provide useful tools for automatic mapping of agricultural fields. Past research in automatic building map has mainly focused on structured indoor environments (Yamauchi, 1997; Bourgault et al., 2002; Simmons et al., 2000; Choset and Nagatani, 2001; Gonzalez-Banos and Latombe, 2002). More recently, some groups (Moorehead et al., 2001; Thrun et al., 2005; Gerbaud et al., 2004) attempted to develop an exploration system for outdoor environments.

Thrun et al. (2005) developed a groundhog navigation system by acquiring 3D maps to explore abandoned mines. The Groundhog was equipped with two tiltable SICK laser range finders and employed its tilting mechanism to acquire 3D range scans of the area ahead of the robot, which were used to build 3D terrain maps. In their study, the next best viewpoint was chosen from a sequence based on a 2D map.

Gerbaud et al. (2004) proposed a terrain exploration approach based upon Delaunay triangulation of the projections of acquired 3D points using a tilted laser range finder. Three exploration policies, reactive, goal-based, and greedy, were developed using their system. In the reactive mode, the next best viewpoint in front of the platform was chosen. In the goal-based navigation mode, the next best viewpoint was selected by the supervisor. In the third exploration mode, the robot chooses a frontier by minimizing a cost function based upon the frontier's size and distance from the platform. In their paper, no details were presented about how the exploration policies were implemented.

Some other studies (Moorehead et al., 2001; Sujan and Dubowsky, 2005; Rocha et al., 2005; Surmann et al., 2003) attempted a frontier-based exploration policy based on 3D occupancy grid maps. The next best viewpoint was chosen by maximizing the information gain while minimizing the path length.

Moorehead et al. (2001) proposed a multiple information metrics exploration planner to integrate multiple sources of information in order to solve complex planetary exploration tasks. An information map was used to store multiple information sources in a 3D environment. This method enabled the robot explorer to maximize the total information gained while minimizing costs such as driving, sensing, and planning. An information map was used to store multiple information sources. The algorithm was demonstrated by creating traversability maps and exploring cliffs. In their simulation, path distance was used as the cost function and a circle was used as an ideal sensor model.

Rocha et al. (2005) attempted a frontier-based exploration strategy to build a 3D occupancy grid map for multi-robots equipped with stereo-vision sensors. An information entropy concept representing uncertainty in the grid-based map was introduced. The exploration strategy drove the robot in the direction of the higher magnitude entropy gradient that is least likely to be occupied by obstacles.

Sujan and Dubowsky (2005) developed an information-based visual robotic mapping approach based on a 3D occupancy grid map in an unstructured environment. The robot is driven to next best viewpoint by maximizing the new information it gains about its environment. New terrain information will be collected in this position and be combined into the environment map using a Kalman filter model (Sujan and Dubowsky, 2005).

The motivation for this work was to contribute to the development of an exploration algorithms to guide the robot to sample points in a rough field while circumventing obstacles in the working field. Complete exploration methods for agricultural robots in a totally unknown environment, using only sensor data, are proposed and simulated using computer software. A terrain map is incrementally built from 3D image sensor readings. A regular triangular mesh map was used to represent the agricultural field surface. A 3D image sensor model, with attributes similar to a camera or laser sensor, was used in the simulation. This work focused on the next best viewpoint algorithm, so it is assumed that positioning is ideal as stated in Section 1.3. A two-stage exploration policy was used to plan the next best viewpoint by considering both the distance and the slope factor in the cost function. In the first stage of exploration, the robot travels to the outer boundary between the explored and unexplored terrain, while in the second stage it fills in the hole left by the first stage. A line sweeping approach based on the bug concept is also presented to find a path for the complete coverage of terrain. Finally, a comparison of the performance of the proposed two-stage exploration strategy based on a triangular cell map with that of the bug-like line sweeping method is presented.

Consistent with previous algorithms, implementation of exploration algorithms using a triangular mesh map, and simulation of a 3D laser sensor are the main original contributions of this chapter.

3.3 Terrain model

Choosing an adequate model for the environment is a difficult task in mobile robotics (Gonzalez-Banos and Latombe, 2002). Different types of world models have been proposed to represent the target environment to facilitate the description of the environment and automatic navigation. There are several basic types of models, such as 2D occupancy grids (Elfes, 1987; Yamauchi, 1997; Bourgault et al., 2002; Simmons et al., 2000), 3D occupancy grids (Thrun et al., 2005; Rocha et al., 2005; Moorehead et al., 2001; Sujan and Dubowsky, 2005), polygon maps (Feder et al., 1999; Gonzalez-Banos and Latombe, 2002), voronoi graphs (Choset and Nagatani, 2001), triangular mesh maps (Gerbaud et al., 2004; Dupuis et al., 2004), hybrid maps (Tovar et al., 2006), and visual maps (Sim and Dudek, 2003).

This research was to design an algorithm that would guide a mobile robot to explore and map large unstructured rough agricultural fields. 2D polygonal maps, which have been extensively used in planar environments, are definitely not suitable to represent three-dimensional rough agricultural fields. Some research (Moorehead et al., 2001; Sujan and Dubowsky, 2005; Thrun et al., 2005) described 3D grid maps for representation of 3D outdoor environments. The triangular mesh map was chosen as a worthy option for modelling the environment and planning the exploration path. A regular triangular mesh map can maintain a very rich representation of the environment and allows a smoother path, compared with the square grid map. Previous studies (Dupuis et al., 2004; Gerbaud et al., 2004) used the triangular mesh map to represent 3D rough environments; however, as far as the author knows, there is no literature about how to plan exploration paths using triangular mesh maps.

A regular triangular mesh map was used to represent the agricultural field surface and plan the exploration path. The triangular mesh map is incrementally built using laser sensor readings based on Delaunay triangulation (Schroeder et al., 1996). The Visualization Toolkit (Kitware Inc., 2005), available freely on the World Wide Web, was used to implement the triangulation in this simulation. Figure 3.1 shows an example of a triangular mesh map of an agricultural field, which was generated from regular map topography data with a grid size of 10 meters. The triangular mesh map is stored in the computer as a directed weighted graph, the vertex of which is used to represent every triangle and the edge of which represents the relative difficulty to traverse the adjacent triangle.

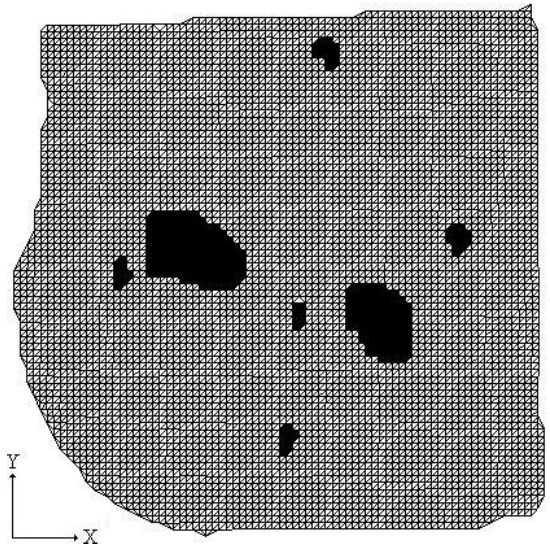


Figure 3.1 Triangular mesh map of an agricultural field environment (field size: 780 meters 800 by meters; the black blobs represent obstacles).

A triangular mesh map has special features. One feature involves the outer boundary and possible holes (contiguous regions located entirely within the boundary of the field that are inaccessible and topographic data are not available) that may exist inside that boundary. Boundary edges of both the outer boundary and holes do not have adjacent pairs of triangles. Another important feature of the boundary edges is that they are always continuous loops. Therefore, in order to find the boundary of a triangular mesh map, a connectivity graph is built by adding all the boundary edges and their connectivity relationships. The algorithm for finding one loop is given below:

(1) Find an extreme point, for example, V_1 shown in Figure 3.2, in the connectivity graph. This point will be the pivot, which is guaranteed to be on one of the boundary loops. In this work, the point with the smallest y coordinate was selected.

(2) Sort the edges connected with the pivot in the order of increasing angle relative the line parallel with the x axis about the pivot. Two outer edges with the smallest and largest angles, can be found. In this case, as much of the polygon as possible can be seen from the pivot. The first three vertices, V_n, V_1, and $V_{2,}$ are called the start vertex, previous vertex, and source vertex, respectively.

(3) Building the boundary loop by moving the pivot to V_{i+1} and adding the next edge that has the largest angle between the previous edge and all edges coincident at V_{i+1} vertex;

(4) Continue building the boundary loop until $V_{i+1} = V_i$ (return to start).

(5) The list of ordered edges generates a counter clockwise loop around the boundary.

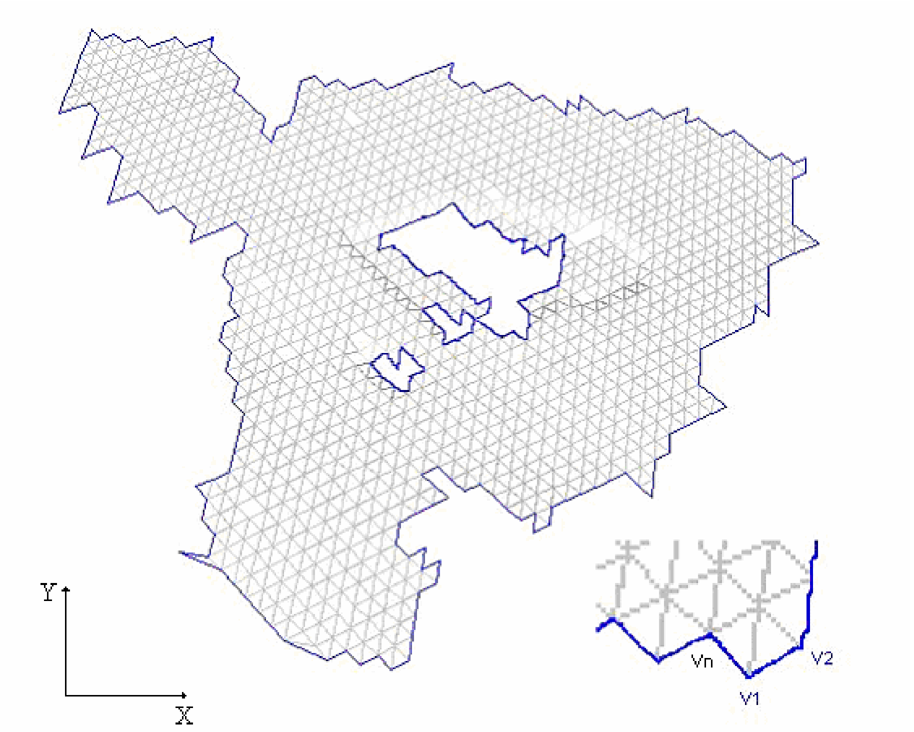


Figure 3.2 Outer boundary and holes found in a triangular mesh map.

After all the edges of the connectivity graph have been visited, all the loops, including the outer boundary and holes, will be found. The loop with the largest area is the outer boundary, while the holes inside the outer boundary have smaller areas. The outer boundary and three holes were found in a triangular mesh map as shown in Fig. 3.2.

3.4 Image sensor model

Most coverage and exploration algorithms only considered ideal sensors. They assumed that if the robot traverses a cell, then the whole cell is covered (Moorehead, 2001; Simmons et al., 2000). Most also assumed that every image reading is a rectangular area. It is difficult to calculate the coverage area of an image sensor such as a camera because it is affected by many factors. Sujan and Dubowsky (2005) first used the field of view of the camera to measure the new information gained in the exploration task. However, the camera model they used was a 2D model, which is not suitable for the rough terrain of an outdoor unstructured environment. To account for challenges associated with rough terrain and incomplete visibility, where one part of the terrain may occlude other parts, a 3D camera model was used in this task.

3.4.1 Frustum culling

The image sensor's capacity is constrained by its pose (position and orientation), field of view (angles between the left and right sides and top and bottom sides of the viewing capacity), and depth of field (sensor capacity in length, Z_{near} and Z_{far}). The viewing frustum shown in Fig. 3.3 is defined by six planes, which are named the near, far, left, right, top, and bottom planes. The viewing frustum defines the visibility of every triangle in the terrain for each viewpoint, and triangles inside the viewing frustum are visible to the viewer. Frustum culling was used to process the object level before the individual pixel was handled in the visibility analysis. Hence, the object level, the triangle, can be rejected quickly in the simulation. The procedure is much faster than a ray casting method (Hearn, 1994).

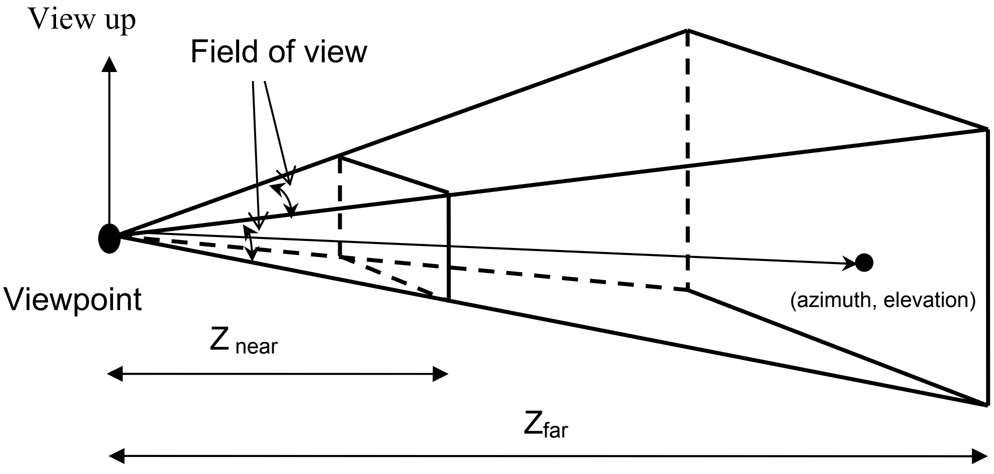


Figure 3.3 Image sensor viewing frustum.

To cull the models, the six planes of the viewing frustum were dynamically generated in accordance with the sensor's posture. These planes were calculated from the view and perspective projection matrices in the camera system (Gribb and Hartmann, 2001). In order to determine whether a triangle within the mesh is inside the frustum, it was necessary to check that all the vertices of the triangle were located inside the volume of the frustum.

This viewing frustum model assumed that the triangle size is much smaller than the viewing frustum. When terrain cells becoming larger, there will be cases where the frustum covers only part of a triangle, as shown in Fig 3.4. The large triangles represent the viewing frustum, while the smaller triangles represent terrain cells. The triangle T1 is seen from this viewpoint because it completely locates inside the frustum; while the triangle T4 is completely out of the frustum and is assumed not to be seen. Errors occur when the triangles, such as T2 and T3, are only partially contained in the viewing frustum. These errors will arise and have an impact on the simulation once the terrain cells become larger. The viewing frustum model also assumed a narrow field of view. A large field of view (near 180 degrees and above) will result in unacceptable error in simulating the laser sensor; therefore, the segments of the surface of a sphere should be used to represent the near field plane and the far field plane of the frustum for large field of view.

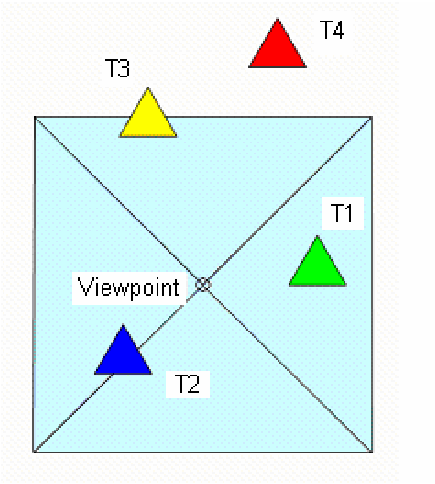


Figure 3.4 Viewing frustum sensor model limitaion.

3.4.2 Ray casting algorithm

For surface visibility calculations, the viewing frustum culling was used to clip the triangles bounded by the camera frustum in the first step. The next step was to check the visibility of every triangle contained in the frustum using a ray casting algorithm. Hearn (1994) described the ray casting algorithm in detail. The basic ray casting algorithm involves throwing a plethora of rays into the scene. A ray, shown in Fig. 3.3, is a straight line extending from the viewpoint to a pixel in the far plane of the viewing frustum. The algorithm begins, as in ray casting, by shooting a ray from the viewpoint to the screen (the far plane of the viewing frustum), then every object (triangle) inside the viewing frustum is tested to see if the given ray intersects any of them. One ray may intersect more than one triangle when a triangle is behind another. From the point of intersection (Z depth shown in Fig. 3.3), the triangle nearest to the viewpoint with a minimum Z depth is visible through this ray, while other triangles which intersect with the ray are shadowed. In this way, the visible triangles can be identified as those which intersect contiguous rays with the shortest distance to the viewpoint. In the simulation, the visible triangles are experimentally identified as those with fewer than 10 shadows. The ray casting algorithm is illustrated in Fig. 3.5.

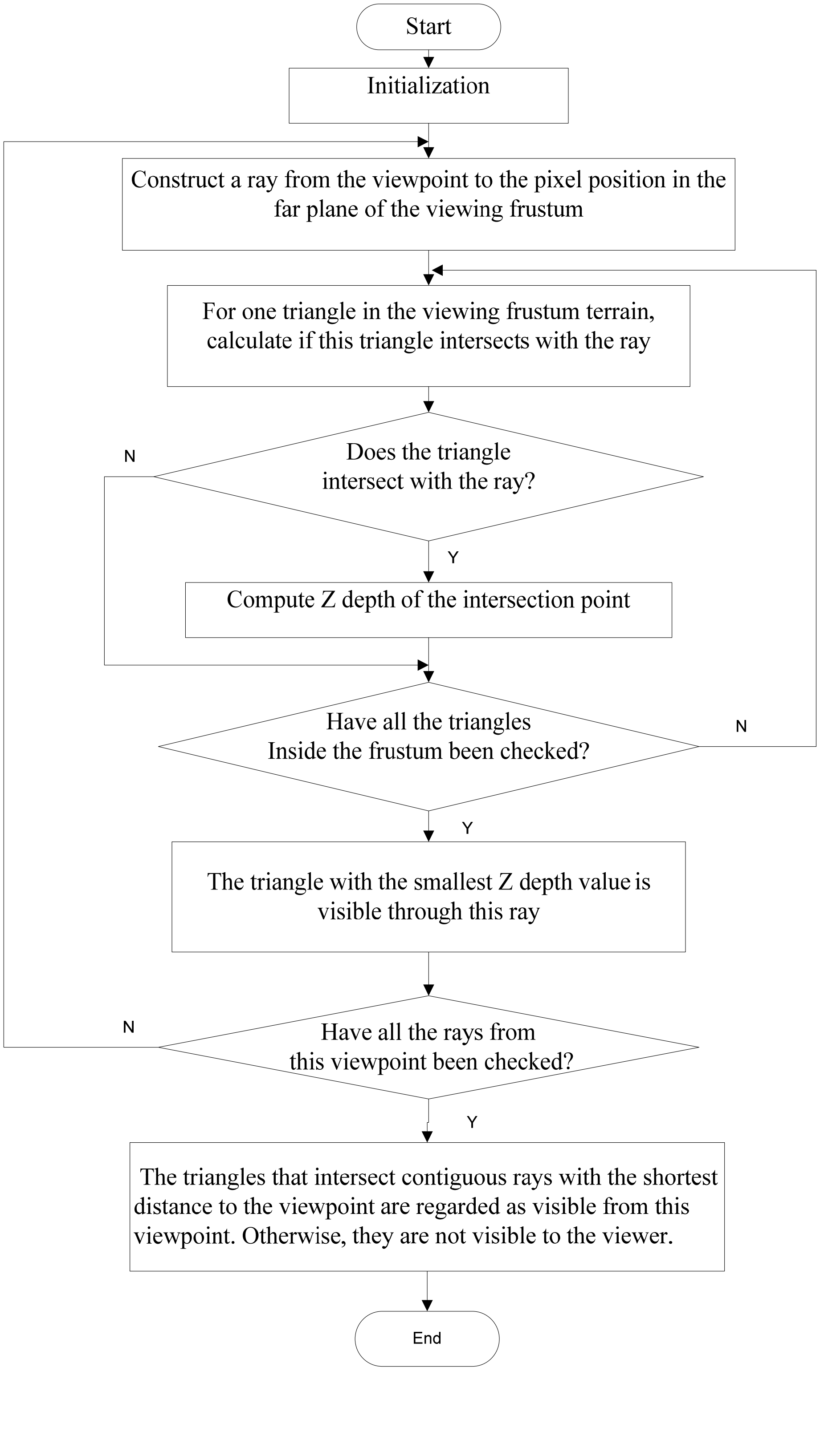

Start
Initialization
Construct a ray from the viewpoint to the pixel position in the far plane of the viewing frustum
For one triangle in the viewing frustum terrain, calculate if this triangle intersects with the ray
Does the triangle intersect with the ray?
N
Y
Compute Z depth of the intersection point
Have all the triangles Inside the frustum been checked?
N
Y
The triangle with the smallest Z depth value is visible through this ray
Have all the rays from this viewpoint been checked?
N
Y
The triangles that intersect contiguous rays with the shortest distance to the viewpoint are regarded as visible from this viewpoint. Otherwise, they are not visible to the viewer.
End

Figure 3.5 Ray tracing algorithm flow chart.

3.4.3 Vision sensor pose simulation

The pose (position and orientation) of the vision sensor is usually defined by the viewpoint, the gaze direction (X_1), and the view up vector (Z_1), as shown in Fig. 3.6. The viewpoint can be derived from its relative position to the robot reference point and the pose of the robot, and it can be transformed from the robot's position using a transform matrix thus, defining the relationship between the reference point and the viewpoint. The gaze direction is defined by the vector from the viewpoint. Considering the orientation of the vehicle, the target point is the projection of the next best viewpoint on the X_1Y_1 plane, which is parallel to the surface of the current triangle where the robot is located. The vertical upward view vector is normal to the current triangle surface. By this way, we can define the orientation of the vision sensor (roll, yaw, pitch) and its position.

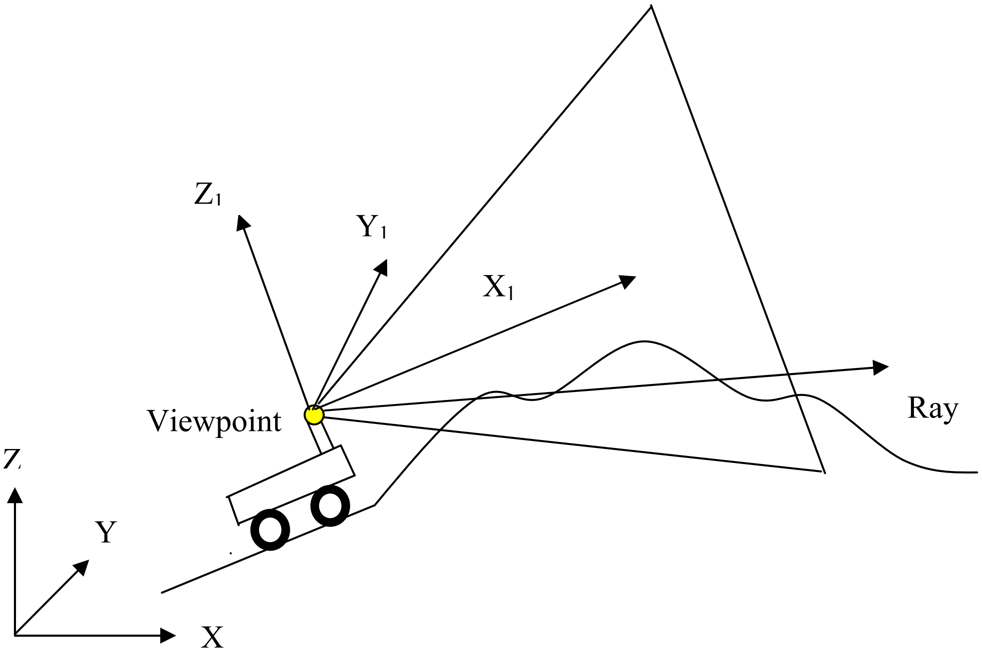


Figure 3.6 Image sensor posture.

3.5 Exploration algorithms

The goal of the exploration was to build the terrain map using the minimum travel distance in the least time. Candidate exploration policies were compared to find an efficient exploration policy for the specific agricultural field model. Previous work has proposed the greedy method or the pattern path method. In this work, two policies were compared. A two-stage next best viewpoint algorithm and a line sweeping pattern method were adopted to fit exploration in a triangular mesh map. The performance in terms of time, distance, and the number of scans are extensively compared.

3.5.1 Two-stage next best viewpoint algorithm

Garcia et al. (1998) presented a two-stage technique to determine the positions where a range sensor should be located to acquire surfaces of a 3D object. This idea was borrowed to perform mobile robotic exploration tasks. The algorithm consists of two stages. The first stage applies a voting scheme that only considers frontiers in order to expand the outer boundary. Most of the terrain is explored during the first stage. The second stage fills any remaining holes left by the first stage. By modeling the large area at the first stage and leaving small patches at the second stage, efficiency and flexibility can be achieved.

At the beginning of each stage, the robot extracts the frontiers (Yamauchi, 1997) from the triangles near the boundary of the current map, and then it constructs a voting scheme using the estimated energy cost for the rover to travel to the frontiers. Figure 3.7 shows the frontier extracted at a specific step in the first stage of exploration. The outer boundary and hole boundary edges are identified after all the loops in the connectivity graph are extracted from the triangular mesh map. The candidate frontiers in the map are defined as those triangles that are close to the boundary and have never acted as a viewpoint before. To reduce the number of the candidate frontiers, the distance between two candidate frontiers must satisfy the minimum distance requirement.

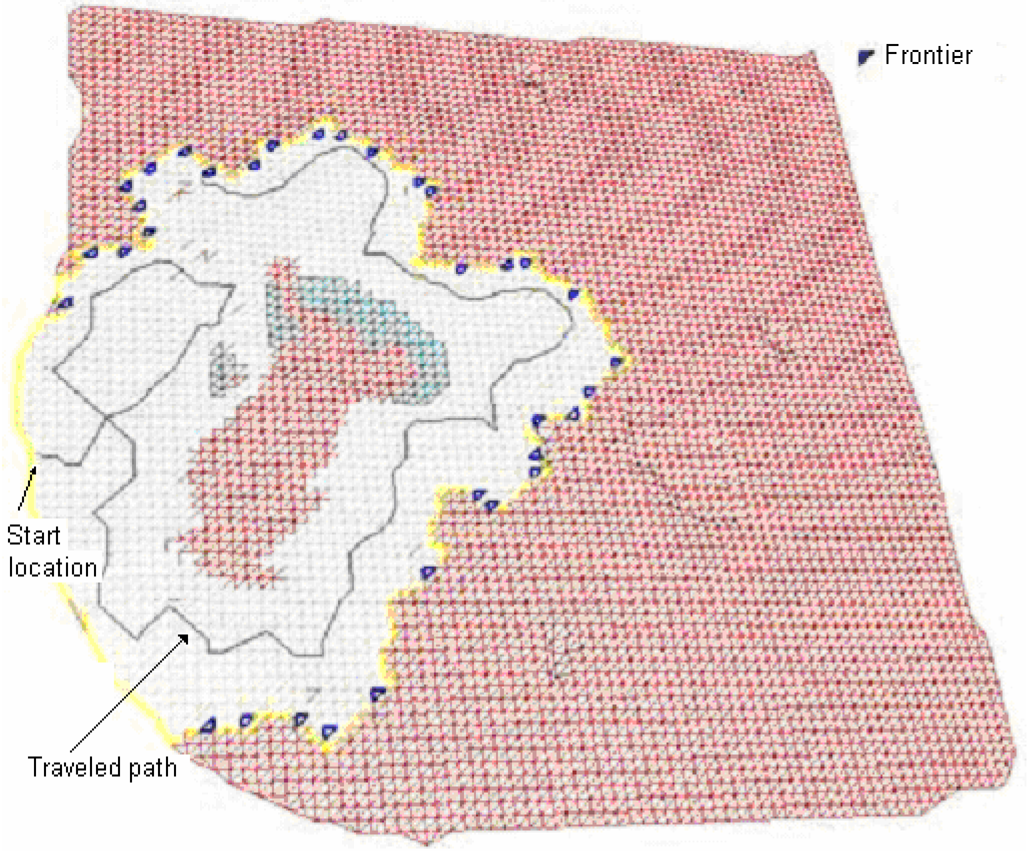


Figure 3.7 Frontiers extracted in an exploration step.

The robot visits the frontier with the minimum energy cost and takes a scan with its image sensor. The map is rebuilt by combining the new sensor reading. The robot plans the next best viewpoint with the new map until no other valuable frontier is available or no other frontier is reachable. Figures 3.8 and 3.9 show the flow diagram of this greedy algorithm. The objective of the greedy approach is to find an optimal path to minimize the travel cost.

The use of a triangular mesh to represent terrain allows the use of a graph search to easily find the next best viewpoint. The triangular mesh map is stored in the computer as a directed weighted graph. Once the graph is constructed, an optimal path between the current rover location and a destination can be planned by Dijkstra's shortest path algorithm (Cormen et al., 2001).

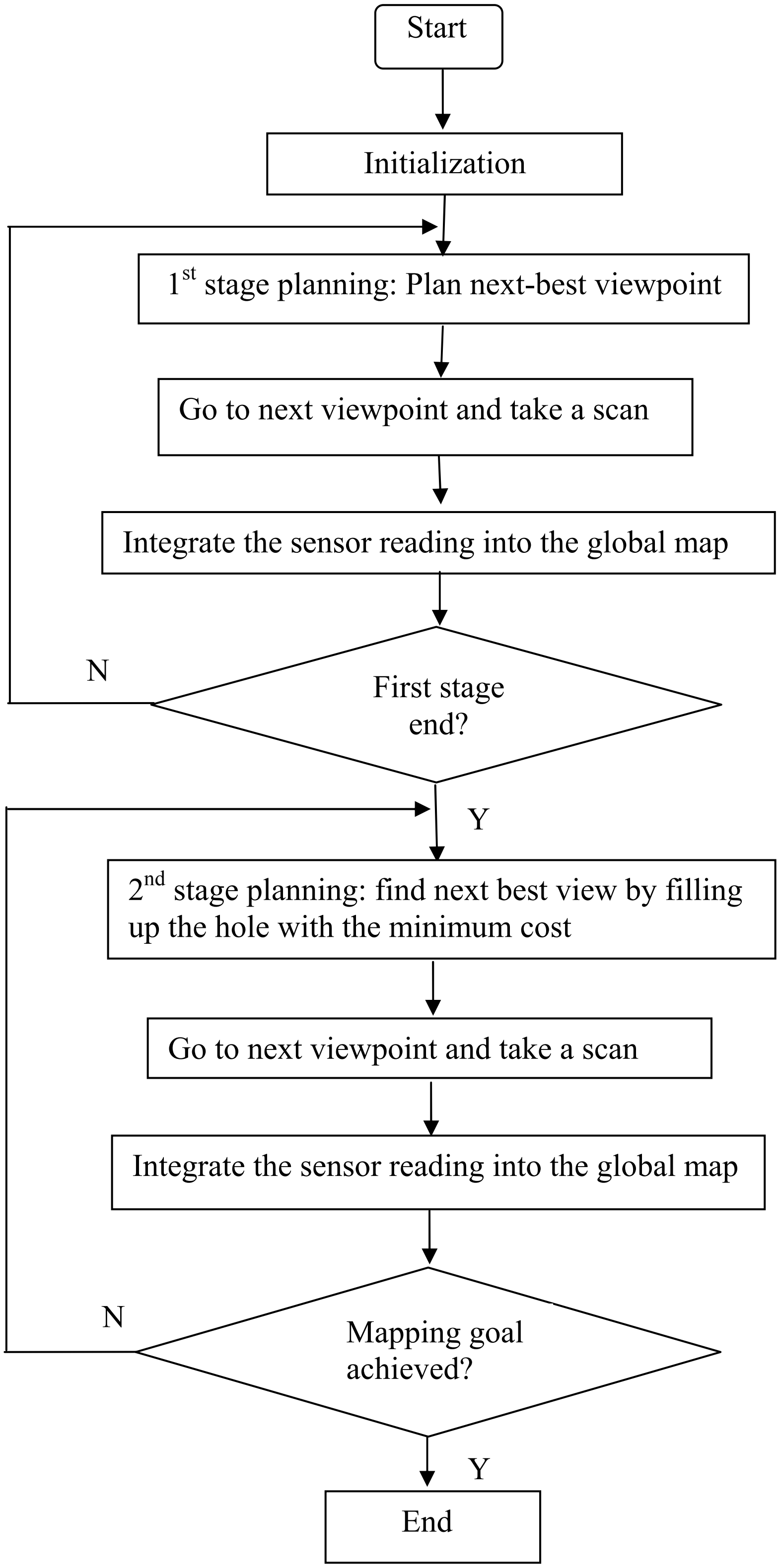

Start
Initialization
1st stage planning: Plan next-best viewpoint
Go to next viewpoint and take a scan
Integrate the sensor reading into the global map
N
First stage end?
Y
2nd stage planning: find next best view by filling up the hole with the minimum cost
Go to next viewpoint and take a scan
Integrate the sensor reading into the global map
N
Mapping goal achieved?
Y
End

Figure 3.8 Two-stage exploration algorithm.

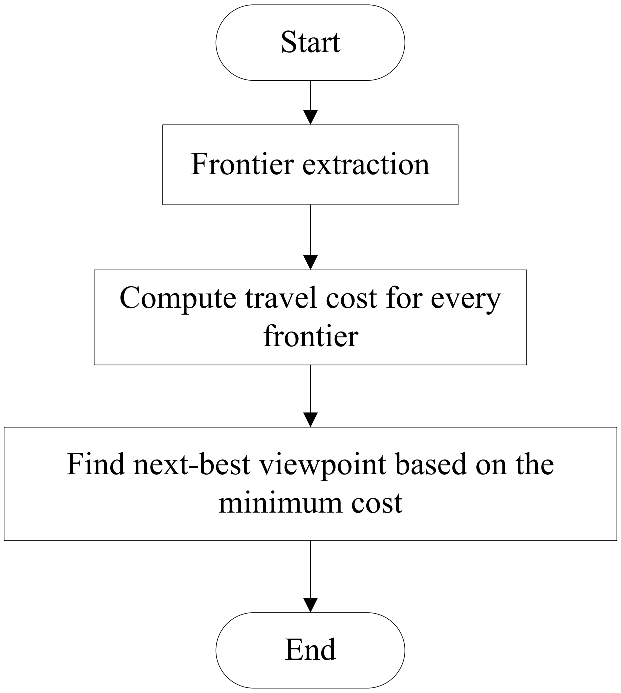

Start
Frontier extraction
Compute travel cost for every frontier
Find next-best viewpoint based on the minimum cost
End

Figure 3.9 Next best viewpoint algorithm.

The next best viewpoint was decided by a cost function which can by descriped by

$$D = \sum_{i=1}^{n} d_i * (1 + sf * \sin(\alpha_i)) \qquad (3.1)$$

where D = cost distance,

d_i = Euclidean distance of the i_{th} segment of a piecewise path,

n = the number of the segments of the path,

sf = slope factor, and

α_i = slope angle of the i_{th} segment of a piecewise path.

The cost function includes the distance traveled and the roughness of the terrain; the total cost requirement for the vehicle to move from a location to a goal location was estimated by the sum of the cost in the piecewise path. The cost is always set as a positive value.

The vehicle dynamics and the safety factor are also a concern in this work. In this chapter, the maximum climbing slope for the robot was set as 30º and the maximum downhill slope was 35º. When the uphill slope was more than 30 degrees or downhill slope was more than 35 degrees, the slope factor will be set as an infinity number because of the vehicle limitation. Because of safety concerns, the robot can not pass any triangle whose tilt angle is over 30º.

The terrain traversable capabilities of the vehicle were based on the triangle where the vehicle is located. The limitations of traversable assumption are potentially inconsistent with the small triangle assumption. If the triangles are kept small to avoid problems with the frustum, then the footprint of the vehicle could span more than one triangle. The footprint would then be more complicated to compute. The triangle size should be defined to satisfy the requirements of both traversable calculation and the viewing frustum simulation.

3.5.2 Line sweeping algorithm

Line sweeping strategies are usually used in a surface-coverage task with a tactile sensor. The Seed Spreader strategy proposed by Lumelsky et al. (1990) was a Bug-style systematic technique for exploring a 2D environment using a tactile sensor. The Seed Spreader algorithm divided the terrain into a number of strips with the same width and the robot navigated around the strip to acquire the information about the surrounding environment. However, if an obstacle was encountered, the robot moved off the strip and circumnavigated an obstacle until the obstacle was fully known.

In this research, a similar Bug-style strategy was used to construct a 3D terrain map based on a triangular mesh map. The concept of covering a planar field with a minimal number of circles (Guo and Qu, 2004) was used to define the sampling points in the field. The coverage problem was solved by finding a number of circles to completely cover the whole environment. Then a predefined path that connected the centers of the circles in sequence as shown in Fig. 3.10, following the line sweeping direction, was applied in the exploration procedure. It is noted that there is a minimum overlap between two sensor coverage circles to cover the whole field for this kind of line sweeping algorithm.

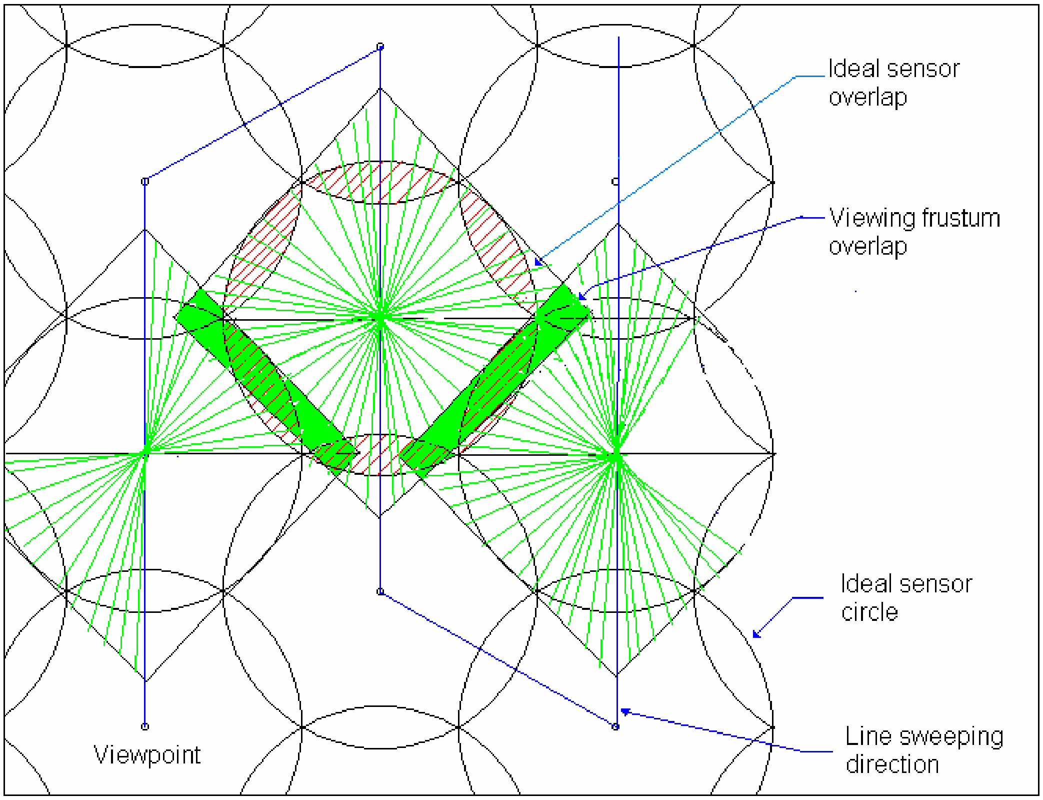


Figure 3.10 Line sweeping algorithm illustration.

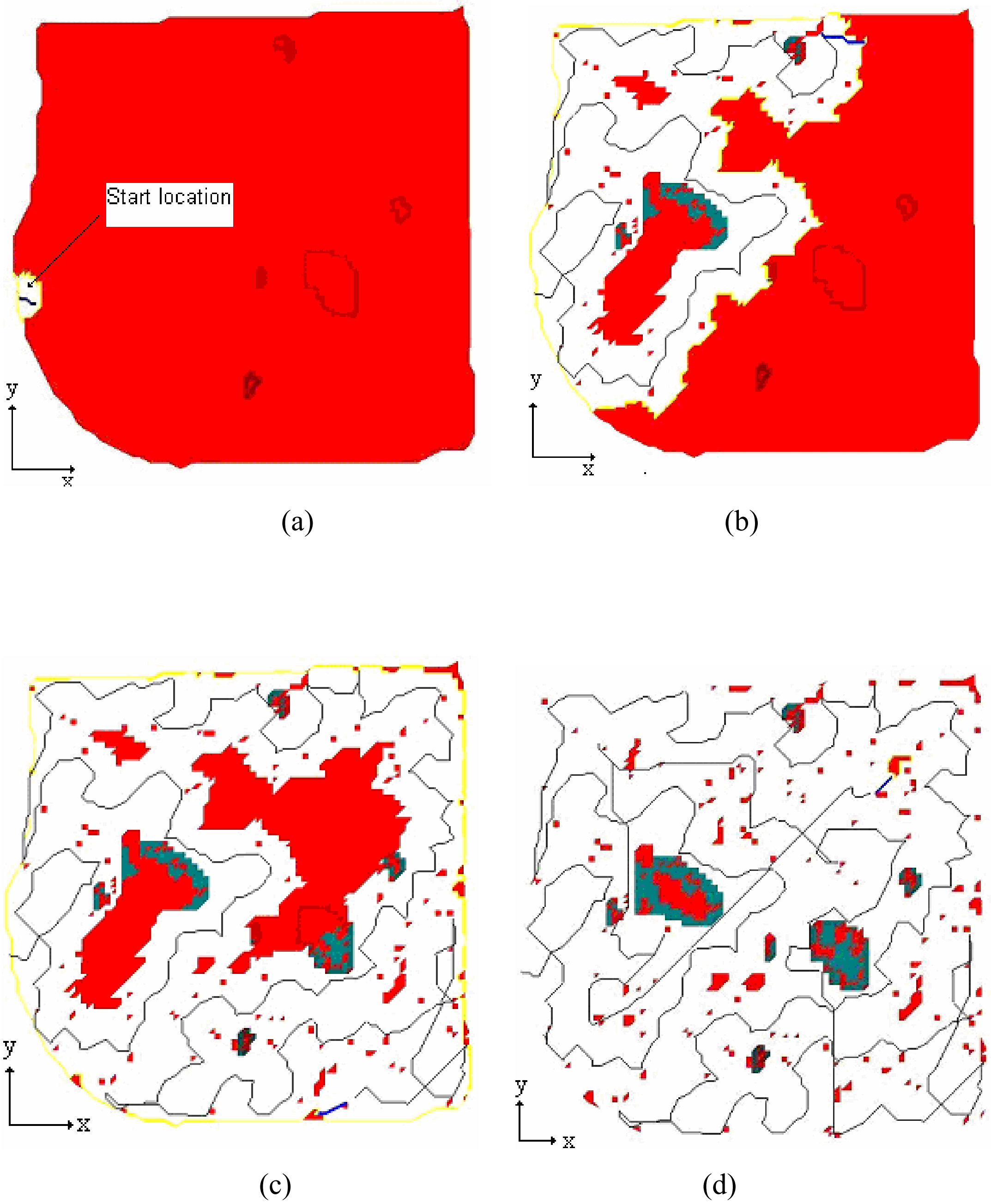
The robot starts at one location and selects the nearest center of the circle in the target environment as the first sampling point. Thereafter, it will choose the next sampling point in the sequence of the line sweeping direction. When the selected sampling point is located in the known environment, Dijkstra's shortest path algorithm is applied to plan an optimal path between the robot's current location and the destination. If the path exists, the robot will visit the destination and take a scan with its image sensor. The map is updated by combining the new sensor readings. The robot plans the path to the next sampling point using the new map, until it reaches the goal of the exploration. If the next sampling point is located in the unknown environment, the robot will choose the boundary edge that is closest to the selected sampling point as the destination and plan an optimal path to this destination using Dijkstra's shortest path algorithm. The robot will visit the triangular cell including this edge and take a scan. The map is updated by adding the new sensor reading. In this case, the robot always goes to the boundary cells closest to the sampling point until it finds a path to the sampling point or is blocked in any direction.

3.6 Results and discussion

A simulated four-wheel drive robot (mass: 16 kg; length: 50 cm; width: 49 cm; height: 26 cm) with four identical wheels (wheel diameter: 25.2 cm; wheel width: 7.5 cm) was used in the simulation. The simulated robot was placed in an unknown agricultural field 800 meters by 800 meters, as in Fig. 3.1. The robot's vision system was assumed to be a 3D image sensor (with a 90° field of view, 50 meter depth of field, and 1:1 aspect ratio). The image sensor could rotate 360˚ horizontally without any cost. The robot started near the lower left corner of the field pixel location (20 m, 300 m).

Captured intermediate screenshots of the simulation for the greedy method and the line sweeping method are shown in Fig. 3.11 and 3.12 respectively, where the traveled path (black lines) and updated map (white parts are explored terrain) are plotted at selected iteration times.

Figure 3.11 (c) shows the exploration result when the first stage of the greedy algorithm ended after 148 iterations; 79 percent of the terrain has been explored, with a travel distance of 7000 meters in 2800 seconds for the first stage of the exploration. The second stage accounted for the remaining 20 percent of the terrain, with a travel distance of 3000 meters in 1200 seconds, by filling up holes in the environment.



(a) (b)

(c) (d)

Figure 3.11 Traveled paths (black lines) and updated maps (white parts are explored terrain) using the greedy method are plotted at selected iteration times: (a) initialization, (b) after 75 iterations (40% of the terrain was explored), (c) after 148 iterations (first stage ended and 79% of the terrain was explored), (d) after 184 iterations (94.4% of the terrain was explored).

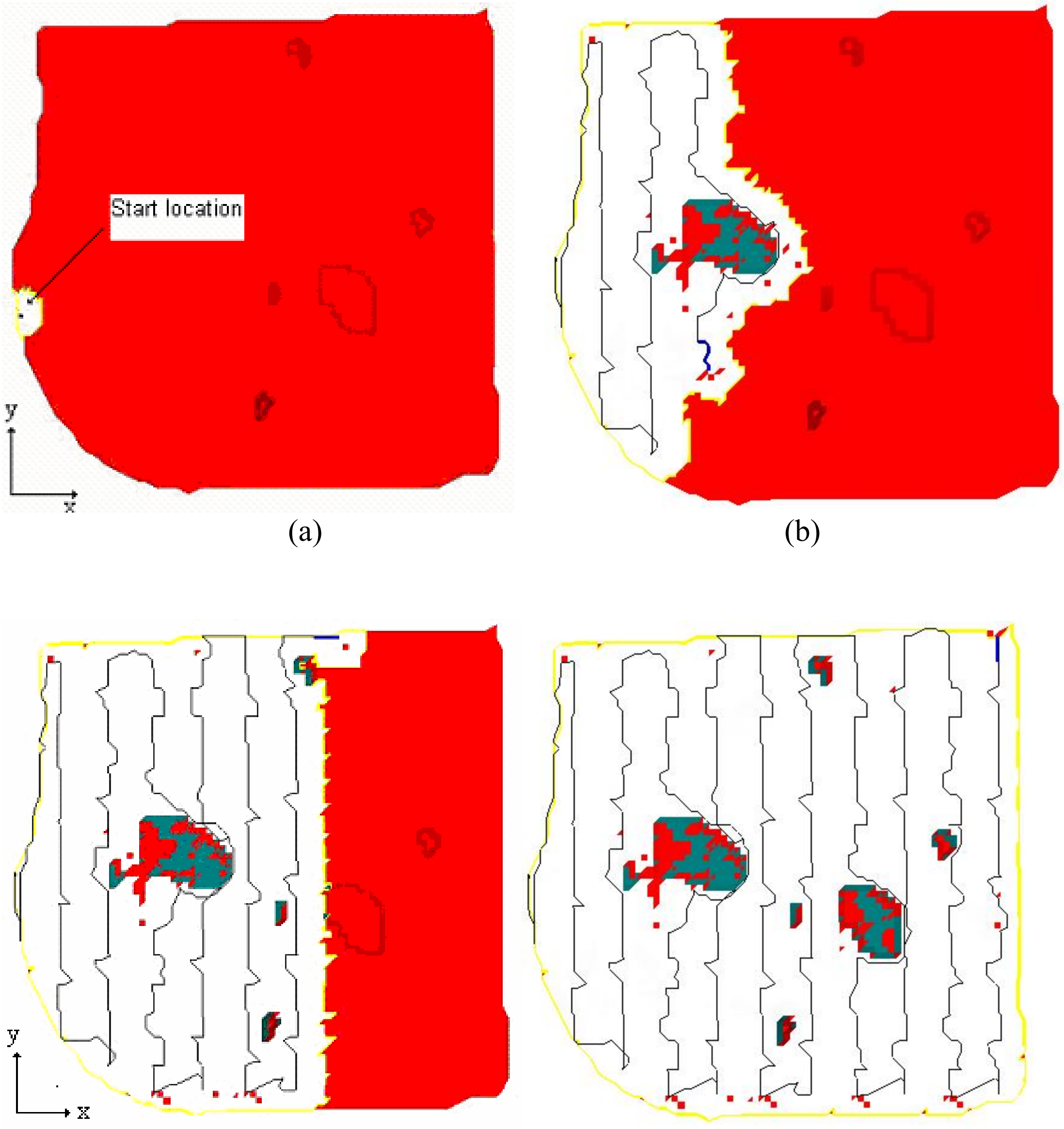
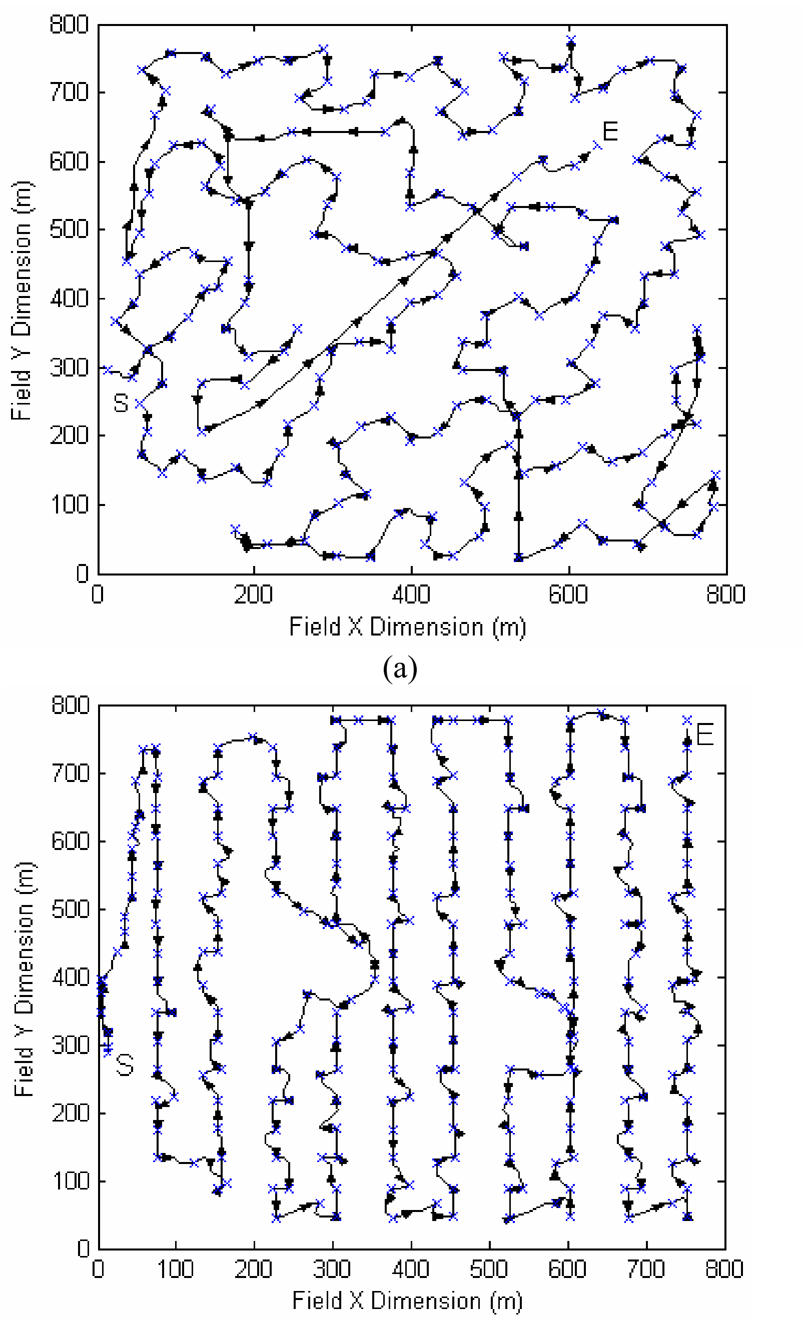


Figure 3.12 Traveled paths (black lines) and updated maps (white parts are explored terrain) using the line sweeping method are plotted at selected iteration times: (a) initialization, (b) after 79 iterations (30% of the terrain was explored), (c) after 160 iterations(59% of the terrain was explored), (d) after 259 iterations (96.7% of the terrain was explored).

Figure 3.13 shows the comparison of the trajectory paths and the viewpoints generated by the greedy method and the line sweeping method. The black lines represent the traveled path; the arrows show the vehicle's travel direction; the cross marks represent viewpoints where the robot stopped to take a scan. The distance between two viewpoints for the line sweeping method is approximately equal in the whole exploration task, while the distance between two viewpoints in some steps of the later stage of the greedy method increases drastically. The robot must travel greater distances to reach the next best viewpoint in the later stage of the exploration task.



(a)

(b)

Figure 3.13 Generated trajectories and selected viewpoints (arrows represents travel direction; cross marks represent viewpoints; start at S and end at E) from (a) the greedy method simulation, and (b) line sweeping method simulation.

Figures 3.14-16 show the relationships between the fraction of the environment mapped and the distance traveled, the time requirement (planning time plus navigation time), and the number of scans made by the robot for the two exploration strategies. It is desirable to map a large fraction of the environment with a short traveled distance, a short time, and few scans. The total path length, time required (including both the planning time and traveling time), and number of scans required for the greedy method were less than those required for the line sweeping method for a consistent target of 95% coverage.

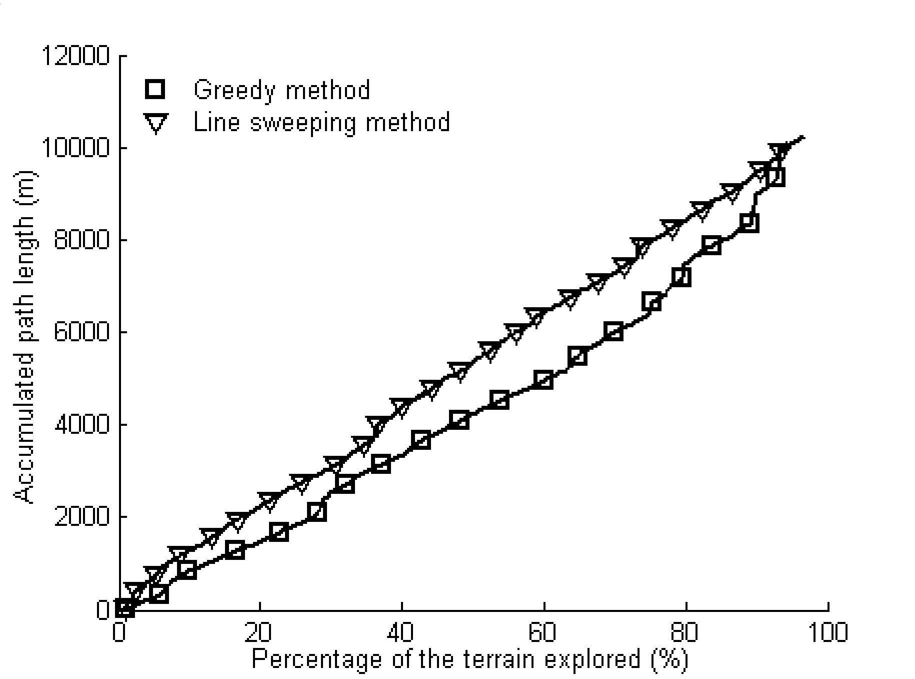


Figure 3.14 Result of the autonomous construction of an agricultural field map: path length traveled by the robot as a function of fraction of explored terrain.

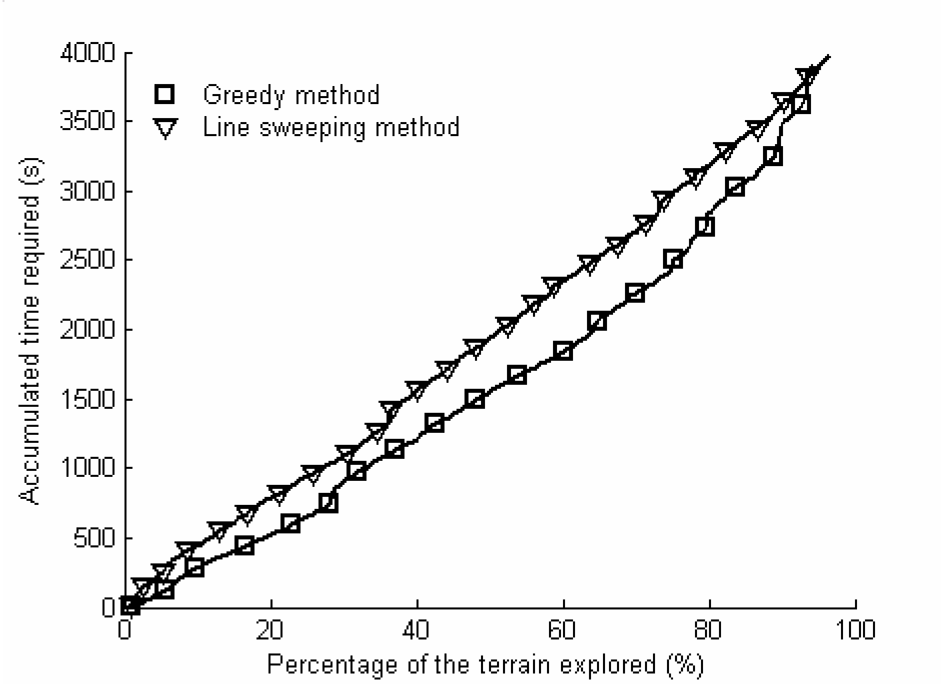


Figure 3.15 Result of the autonomous construction of an agricultural field map: time required for the exploration as a function of fraction of explored terrain.

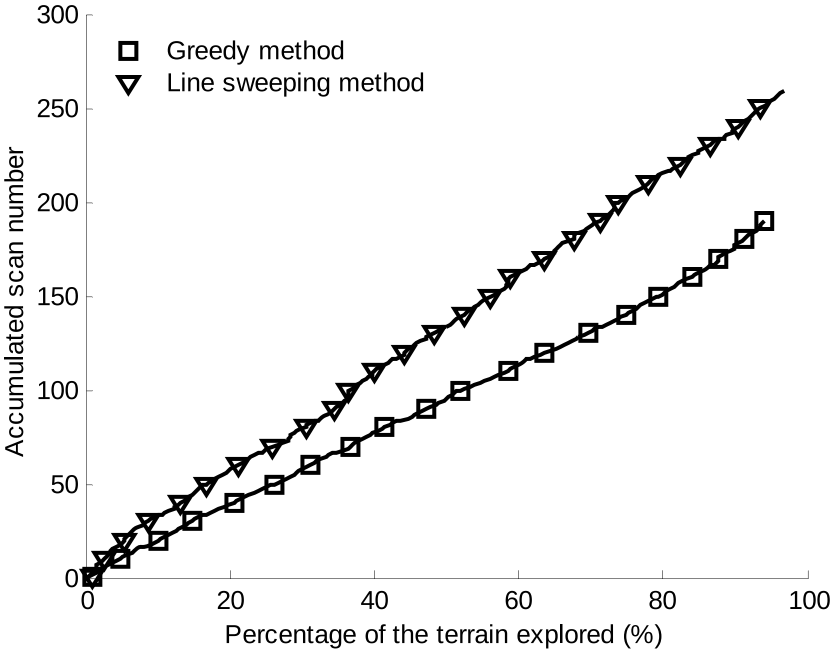


Figure 3.16 Result of the autonomous construction of an agricultural field map: the number of scans required for the exploration as a function of fraction of explored terrain.

For the line sweeping method, the relationships between the fraction of the environment mapped and the distance, time consumption, and scan number are linear. The rate of the distance, time consumption, and number of scans are almost constant for the entire exploration procedure. However, for the greedy method the relationship between the fraction of the environment mapped and the distance or time consumption is non-linear. The first part of the exploration requires less time and a shorter travel distance to explore the same fraction of the environment than the fraction explored in the last part of the exploration. In other words, the robot is more efficient in the first stage and requires more effort in the remaining portion of terrain. Both exploration techniques required almost the same total travel distance and time to complete the exploration. In terms of performance based on the number of scans, the greedy method showed a great advantage over the line sweeping method. It required 184 scans to complete the exploration using the greedy method, while it required 259 scans using the line sweeping method. The relationship between the fraction of the environment mapped and the number of scans is linear for both strategies.

These results illustrate the effectiveness of the two-stage greedy method in its ability to minimize number of scans, time consumption, and traveling distance in the early stage of the exploration. The performance difference between the greedy and the line sweeping methods might partly come from the viewing frustum overlap between two sensor readings for this kind of line sweeping method. The line sweeping method has the potential to be improved to reduce the overlap, thereby reduce the scan number and path length.

3.7 Conclusions

In this chapter, the problem of autonomous mapping was addressed in a large unstructured agricultural environment. A new triangular mesh map was presented that allowed the robot to maintain a very rich representation of the environment and to robustly perform exploration. A viewing frustum model and ray casting algorithm were described to facilitate a 3D image sensor simulation. Finally, the problem was addressed with the development of two exploration algorithms, which included greedy and pattern path methods, to map an unknown rough agricultural environment based on the triangular mesh map representation. For the greedy method, a two-stage strategy was used to find the next best viewpoint by considering both the distance and the slope factor in the cost function. In the first stage of exploration, the robot tended to visit frontiers near the outer boundary of the terrain, while in the second stage of exploration, the robot filled the holes left by the first stage. In comparison, bug-style line sweeping strategy was presented.

Simulation results for a typical western Canadian agricultural field are presented and demonstrate:

(1) the simulation tool performed very well to simulate the exploration based on the triangular mesh map and a 3D laser sensor,
(2) the greedy method was more efficient at early stages and required about 16% less than the line sweeping method in terms of travel distance and time to complete 75% of exploration,
(3) from the perspective of the whole exploration procedure, there is little difference in the traveled distance and time consumption required for either method, and,

3.8 References

Adamchuk, V. I., M. T. Morgan and D. R. Ess. 1999. An automated sampling system for measuring soil pH. *Transactions of the ASAE* 42(4): 885-891.

Bourgault, F., A.A. Makarenko, S.B. Williams, B. Grocholsky and H.F. Durrant-Whyte. 2002. Information Based Adaptive Robotic Exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* 1: 540-545. Lausanne, Switzerland. September 30-October 5.

Bishop, T. F. A. and A. B. McBratney. 2002. Creating field-extent digital elevation models for precision agriculture. *Precision Agriculture* 3(1): 37-46.

Choset, H. and K. Nagatani. 2001. Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation* 17(2): 125-137.

Clark, R. L. and R. Lee. 1998. Development of topographic maps for precision farming with kinematic GPS. *Transactions of the ASAE* 41(4): 909-916.

Cormen H.T., C.E. Leiserson, R.L. Rivest and C. Stein. 2001. *Introduction to algorithms*, second edition. MIT Press and McGraw-Hill. Cambridge/New York.

Dupuis, E., P. Allard, J. Bakambu, T. Lamarche and W·H. Zhu. 2004. Towards autonomous long-range navigation. In *8th ESA Workshop on Advanced Technologies for Robotics and Automation 'ASTRA 2004'*, ESTEC. Noordwijk, the Netherlands. November 2-4.

Elfes, A. Sonar-based real-world mapping and navigation. 1987. *IEEE Journal of Robotics and Automation*, RA3(3):249-265.

Feder,H.J.S., J.J. Leonard and C.M. Smith. 1999. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research* 18(7): 650-668.

Garcia, M., S. Velazquez, and A. Sappa. 1998. A two-stage algorithm for planning the next view from range images. In *Proceedings of ninth British Machine Vision Conference* 1998: 720-729. Southampton, UK.

Gerbaud, T., V. Polotski and P. Cohen. 2004. Simultaneous exploration and 3D mapping of unstructured environments. *Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, *SMC 2004*: 5333-5337. Hague, Netherlands. 10-13 Oct. 2004.

Gonzalez-Banos, H.H. and J.C. Latombe. 2002. Navigation strategies for exploring indoor environments. *International Journal of Robotics Research* 21(10-11): 829-848.

Gribb, G. and K. Hartmann. 2001. Fast extraction of viewing frustum planes from the world-view-projection matrix. *http://www2.ravensoft.com/users/ggribb/ plane%20extraction.pdf.*

Guo, Y. and Z. Qu. 2004. Coverage control for a mobile robot patrolling a dynamic and uncertain environment. In *Proceedings of World Congress on Intelligent Control and Automation.* China, June 2004.

Hearn, D. 1994. Computer graphics, second edition. Prentice-Hall. EngleWood Cliffs, New Jersey.

Kitware Inc. 2005. The Visualization Toolkit. *http://www.vtk.org.* Accessed June 26, 2005. New York.

Lumelsky, V., S. Mukopadhyay and K. Sun 1990. Dynamic path planning in sensor-based terrain acquisition. *IEEE Transactions on Robotics and Automaation* 6: 462-472.

Moorehead, S., R. Simmons and W.L. Whittaker. 2001. Autonomous Exploration Using Multiple Sources of Information. In *Proceedings of the IEEE International Conference on Robotics and Automation* 3: 3098- 3103. May 21-26.

Oh, J., Y. Choi, J. Park and Y. F. Zheng. 2004. Complete coverage navigation using of cleaning robots using triangular cell based map. *IEEE Transactions on Industrial Electronics* 51(3): 718-726.

Rocha, R., J. Dias and A. Carvalho. 2005. Cooperative multi-robot systems: A study of vision-based 3-D mapping using information theory. *Robotics and Autonomous Systems* 53(3-4): 282-311.

Saraswat, D., R. Ehsani, N. Watermeier and M. Sullivan. 2003. Potential Application of Yield Data for Creating Topographic Maps. In *2003 ASAE Annual Meeting*, Paper number: 031084.

Schmidt, J. P., R. K. Taylor and R. J. Gehl. 2003. Developing topographic maps using a sub−meter accuracy global positioning receiver. *Applied Engineering in Agriculture* 19(3): 291-300.

Schroeder, W., K. Martin and B. Lorensen. 1996. The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics. Prentice Hall PTR. Upper Saddle River, New Jersy.

Sim, R. and G. Dudek. 2003. Effective exploration strategies for the construction of visual maps. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*: 3224-3231. Las Vegas, NV, October 27-31.

Simmons, R., D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun and H. Younes. 2000. Coordination for multi-robot exploration and mapping. In *Proceedings National Conference on Artificial Intelligence.* Austin, TX.

Sujan,V. A., and S. Dubowsky. 2005. Efficient Information-based Visual Robotic Mapping in Unstructured Environments. *International Journal of Robotics Research* 24 (4): 275-293.

Surmann, H., A. Nüchter and J. Hertzberg. 2003. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems* 45: 181-198.

Thrun, S., S. Thayer. W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverte and W. Whittaker. 2005. Autonomous Exploration and Mapping of Abandoned Mines. *IEEE Robotics and Automation Magazine* 11(4): 79-91.

Tovar, B., L. Munoz-Gomez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy and S. Hutchinson. 2006. Planning exploration strategies for simultaneous localization and mapping. *Robotics and Autonomous Systems* 54(4): 314-331.

Westphalen, M. L., B. L. Steward and S. Han. 2004. Topographic mapping through measurement of vehicle attitude and elevation. *Transactions of the ASAE* 47(5): 1841-1849.

Yamauchi, B. 1997. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation:* 146-151. Monterey, CA. July 10-11.

Yao, H. and R. L. Clark. 2000. Development of topographic maps for precision farming with medium accuracy GPS receivers. *Applied Engineering in Agriculture* 16(6): 629-636.

4. AN EXPLORATION STRATEGY BASED ON THE MINIMUM ENERGY CONSUMPTION FOR AUTONOMOUS CONSTRUCTION OF AGRICULTURAL FIELD MAPS

4.1 Significance

This chapter relates to objective 2 of the thesis (Chapter 1.2). Chapter 3 demonstrated the feasibility of an exploration policy based on a triangular mesh map and the advantages of the greedy approach over the line sweeping method. The review of robotic exploration and mapping in Chapter 2 revealed the limitations of only considering the distance when determining the travel cost in the exploration task. The research in the Chapter 3 provided a platform to investigate different cost functions in exploration algorithms based on a triangular mesh map and a 3D image sensor model. The main aim of this chapter was to explore the possibilities of using the energy requirement as a cost function to select the next best viewpoint (NBV). To validate the developed next best viewpoint algorithm, a variety of strategies will be developed and their performance will be compared extensively in terms of energy requirement, time, traveled distance, and number of scans.

4.2 Introduction

Agricultural field topographic mapping has become an active topic of research in the last decade. The most popular approach for generating agricultural field maps is the vehicle-based method, in which a GPS receiver is mounted on an agricultural vehicle to measure the field variability while the vehicle moves around the working field (Bishop and McBratney, 2002; Westphalen et al., 2004; Schmidt et al., 2003; Yao and Clark, 2000; Clark and Lee, 1998). However, previous work required a driver, who must pass through the fields along a pre-defined path such as a crop row or straight line direction, which is labor intensive work.

Advances in new sensors, high speed computers, and control technologies have provided the potential to develop robot platforms for automatic mapping of agricultural fields. This manuscript addresses the automatic mapping problem of moving an image sensor to collect the terrain information and concomitantly construct a terrain map while circumventing obstacles in the working field. A terrain map is incrementally built using image sensor readings. At each step, the robot decides where to go to collect new information based on a partially-built map. This is a typical next best viewpoint problem (Gonzalez-Banos and Latombe, 2002) in robotic exploration.

Several exploration strategies have been developed to address autonomous construction of maps. One group of approaches utilizes pattern paths to explore the whole field. These algorithms (Choset, 2001; Hert et al., 1996; Huang, 2001) take the following basic approach to generate a coverage path: the region to be covered is divided into subregions, a traveling-salesman algorithm (Choset, 2001) is applied to generate a sequence of subregions to visit, and a coverage path is generated from this sequence that covers each subregion in turn. All of these algorithms use a single line sweep in order to divide the coverage region into subregions, and these subregions are individually covered using a back and forth motion in rows perpendicular to the sweep direction. The limitation of this group of strategies is that the efficiency of these approaches is heavily affected by the line sweeping direction. Another group of exploration strategies is to choose a next best viewpoint among frontiers (Yamauchi, 1997) extracted from the boundary between the known and unknown areas. Path distance is typically used in the cost function to select the next best viewpoint to which the robot will move to explore new terrain (Gonzalez-Banos and Latombe, 2002; Sujan and Dubowsky, 2005; Taylor and Kriegman, 1998; Thrun et al., 1998; Yamauchi, 1997). However, in the exploration of agricultural fields, the problem is complicated by a number of farming issues. One of the most important issues is that agricultural field surfaces are usually rough. Travel distance alone is unsuitable to represent traveling cost in the rough terrain of an outdoor unstructured environment. The ruggedness of the terrain influences the exploration strategy employed by a robot because the cost of driving is not the same for all traversable areas.

The specific objectives of this phase of the research were to develop

1) the concept of including energy consumption within the cost function to choose the next best viewpoint (the cost function not only considers the traveling distance, but also includes the energy required to change elevation, the rolling resistance of the terrain, and the vehicle tire slip during exploration)
2) spiral pattern strategies to choose the next best viewpoint , and
3) a comparison of the performances of the different strategies in terms of energy requirement, time, distance, and scan number.

This work assumed the zero positioning error. In other words, the robot always knows where it is precisely.

4.3 World model

Although different models such as occupancy grids, polygon maps, voronoi graphs, and hybrid maps have been proposed to represent the target environments in robotic exploration, few of them are fit for an outdoor rough field exploration task. 3D grid maps (Moorehead et al., 2001; Sujan and Dubowsky, 2005; Thrun et al., 2005) were usually used to represent outdoor uneven environments. The regular triangular mesh map was proposed in this research to model the agricultural field surface because of its ability to generate a smoother path for navigation tasks.

The triangular mesh map is incrementally built using laser sensor readings based on Delaunay triangulation (Schroeder et al., 1996). The Visualization Toolkit (Kitware Inc., 2005) has been used to implement the triangulation in this simulation. The detail description of the triangular mesh model is discussed in Chapter 3.3.

4.4 Image sensor model

To account for challenges associated with rough terrain and incomplete visibility, where one part of the terrain may occlude other parts, a 3D view model was used. Chapter 3 described, in detail, the approaches to simulating the 3D image in this project.

The image sensor's capacity is constrained by its pose (position and orientation), field of view (angles between the left and right sides and top and bottom sides of the viewing capacity), and depth of field (sensor capacity in length, Z_{near} and Z_{far}). The viewing frustum defines the visibility of every triangle in the terrain for each viewpoint, and triangles inside the viewing frustum are visible to the user. For surface visibility calculations, the viewing frustum culling is used to clip the triangles bounded by the camera frustum in the first step. The ray casting algorithm, which is described in detail in Chapter 3, is used to check the visibility of every triangle contained in the frustum using.

4.5 Exploration algorithm

The core of this research is to choose appropriate next best viewpoints. Energy cost is critical in farming due to its economic impact, so this chapter proposes the concept of including energy consumption within the cost function to choose the next best viewpoint. The goal of the exploration is to consume the minimum amount of energy to explore new terrain in each step. The energy cost function not only considers the traveling distance but also includes the energy required to change elevation, the rolling resistance of the terrain, and the vehicle tire slip during exploration.

4.5.1 Energy cost function

Suvinen et al. (2003) proposed the concept of generating a cost surface based on machine, terrain, tree coverage, road, and weather objects for GIS-based terrain mobility modeling and optimization of off-road routes. The energy cost function through a vehicle tractive function is derived below. The equation of motion along the longitudinal axis of the vehicle was expressed by Wong (1978) as

$$F_t = R_a + R_r + R_d + R_s + R_i, \qquad (4.1)$$

where

F_t = the tractive effort (N),

R_a = the aerodynamic resistance of the vehicle (N),

R_r = the rolling resistance of the vehicle (N),

R_d = the drawbar load (N),

R_s = the slope resistance (N), and

R_i = the inertial resistance (N).

Aerodynamic resistance is usually not a significant factor for off-road vehicles operating at speeds below 48 km/h (Wong 1978), and it was assumed to be zero in this research. Drawbar load, R_d, was also assumed to be zero for the exploration robot. When the velocity remains constant, inertial resistance, R_i, is zero. Slope resistance is calculated using the inclined plane equation,

$$R_s = W \sin\theta \,, \qquad (4.2)$$

where W is the weight of the vehicle (N) and θ is the inclination angle of the terrain. When driving uphill (θ is positive), slope resistance is in the opposite direction of the vehicle's tractive force, and it functions as a resistant force. When driving downhill (θ is negative), slope resistance is in the same direction as the vehicle's tractive force, and the slope resistance works as an active force.

On a slope at a constant low speed, the tractive effort, F_t, must overcome slope resistance and rolling resistance,

$$F_t = W * \sin\theta + R_r \,. \qquad (4.3)$$

The rolling resistance of a pneumatic tire is dependent on load, size, tread pattern, and inflation pressure as well as soil strength (Goering et al., 2003). For soils that are not very soft and tires with a width/diameter ratio of approximately 0.3, along with tires with a deflection/section height ratio (δ/h) limitation of 0.20, the rolling resistance for a single tire can be predicted using the equation (Goering et al., 2003)

$$R_t = W_t\left(\frac{1.2}{C_n} + 0.04\right), \tag{4.4}$$

where

R_t = the rolling resistance for a single tire (N),

$C_n = \frac{CI * b * d}{W_t}$, wheel numeric (dimensionless),

CI = cone index measured with a cone penetrometer as in ASAE S 313.2 (N/cm^2),

$W_t = \frac{W\cos\theta}{4}$ for a 4WD vehicle with identical tires, load on a single tire (N),

b = the tire width (cm), and

d = the tire diameter (cm).

The total rolling resistance for a 4WD vehicle on a slope, with 4 identical tires ($b/d \approx 0.3$), along with a δ/h limitation of 0.20, can be predicted using the formula

$$R_r = W\cos\theta\left(\frac{1.2}{C_n} + 0.04\right). \tag{4.5}$$

With the tractive force and the path length known, the energy requirement of the vehicle traveling along a straight line path can be derived using

$$E = \frac{F_t * 1}{1 - s}, \tag{4.6}$$

where

E = the energy requirement (N·m),

l = the Euclidean distance or the surface distance of the path (m),

$s = 1 - \frac{V_a}{V_t}$, the wheel slip rate,

$V_a =$ actual travel speed (m/s), and

$V_t =$ theoretical wheel speed (m/s).

Substituting Eqs. 4.3 and 4.5 into Eq. 4.6, the energy requirement can be represented by

$$E = \frac{W}{1-s} * \left[l * \sin\theta + l * \cos\theta * \left(\frac{1.2}{C_n} + 0.04 \right) \right]. \tag{4.7}$$

Substituting the wheel numeric definition and trigonometric functions into Eq. 4.7,

$$E = \frac{W}{1-s} * \left[\Delta z + d_h \left(\frac{d_h}{l * CI} A + 0.04 \right) \right], \tag{4.8}$$

where

$\Delta z = l * \sin\theta$, slope height of the path (m),

$d_h = l * \cos\theta$, horizontal distance of the path (m), and

$A = \frac{0.3W}{b * d}$, constant (N/cm^2).

Δz is positive on an uphill slope, while it is negative on a downward slope. In this chapter, the maximum uphill slope is 30 degrees, limited by vehicle capabilities. Similarly, the maximum downhill slope is 35 degrees.

The total energy requirement for the vehicle to reach a destination from a starting location is calculated by integration of energy with the piecewise path.

$$E_{total} = W * \sum_{i=1}^{n} \left[\left(\frac{1}{1-s_i} \right) * \left(\Delta z_i + d_{hi} \left(\frac{d_{hi}}{l_i * CI_i} * A + 0.04 \right) \right) \right], \tag{4.9}$$

where

E_{total} = total energy requirement ($N \cdot m$),

Δz_i = the slope height of the i_{th} segment of a piecewise path (m),

d_{hi} = the horizontal distance of the i_{th} segment of a piecewise path (m),

n = the number of the segments of the path,

s_i = the slip rate of the i_{th} segment of a piecewise path,

l_i = the Euclidean distance of the i_{th} segment of a piecewise path (m), and

CI_i = the cone index of the i_{th} segments of the path (N/cm^2).

The rolling resistance and the energy cost is inversely proportional to the cone index, CI, which depends on moisture content, specific weight and soil type. The soil hardness might be highly variable in one field considering soil type and moisture content can cause significant changes in CI values. When a soil strength map is not available, a uniform CI value for one whole field is assumed. The energy requirement can be calculated using

$$E_{total} = W * \sum_{i=1}^{n} \left[\left(\frac{1}{1-s_i} \right) * \left(\Delta z_i + d_{hi} * \mu_i \right) \right], \tag{4.10}$$

where

$\mu_i = \frac{d_{hi}}{l_i} * B + 0.04$, rolling resistance coefficient (dimensionless), and

$B = \frac{0.3W}{CI * b * d}$, constant (dimensionless).

Slip between the tire and soil surface during exploration affects the energy consumption by loosing traction to generate heat loss. When the slip factor is not considered, and the soil is hard and uniform for the entire field, the energy consumption can be described by

$$E_{total} = W * \sum_{i=1}^{n} \left[\Delta z_i + d_{hi} * \mu_i \right] \tag{4.11}$$

The energy function, Eq. 4.11, shows that the energy consumption is proportional to the elevation change Δz_i. When driving uphill, Δz_i is positive and slope resistance is in the opposite direction of the vehicle's tractive force. When driving downhill, Δz_i is negative and the slope resistance is in the same direction as the vehicle's tractive force. Assuming the vehicle travels at a constant speed in this chapter, brake energy will be required when the negative elevation change, Δz_i, is excessive.

The energy consumption is also proportional to the horizontal trip distance of the vehicle. The contribution of travel distance to energy consumption will vary considerably in relation to the rolling resistance coefficient, μ_i. A tire's rolling resistance coefficient depends on the soil hardness as indicated by Cone index readings, terrain slope, and wheel parameters. Load has a positive relationship with the energy in that the robot needs fuel to carry the load. Load also affects the energy by changing the magnitude of the rolling resistance coefficient.

4.5.2 Greedy method

The use of a triangular mesh to represent terrain allows the use of a graph search to easily find the next best viewpoint. The triangular mesh map is stored in the computer as a directed weighted graph. Once the graph is constructed, an optimal path between the current rover location and a destination can be planned by Dijkstra's shortest path algorithm (Cormen et al., 2001). The objective of the greedy approach is to find an optimal path to minimize the traveling cost.

To start, the robot extracts the frontiers (Yamauchi, 1997) from the triangles near the boundary of the current map, and then it constructs a voting scheme using the estimated energy cost for it to travel to frontiers. The robot visits the frontier with the minimum energy cost and takes a scan with its image sensor. The map is rebuilt by combining the new sensor reading. The robot plans the next best viewpoint with the new map until it reaches the goal of exploration. Figure 4.1 shows the flow diagram of this greedy algorithm.

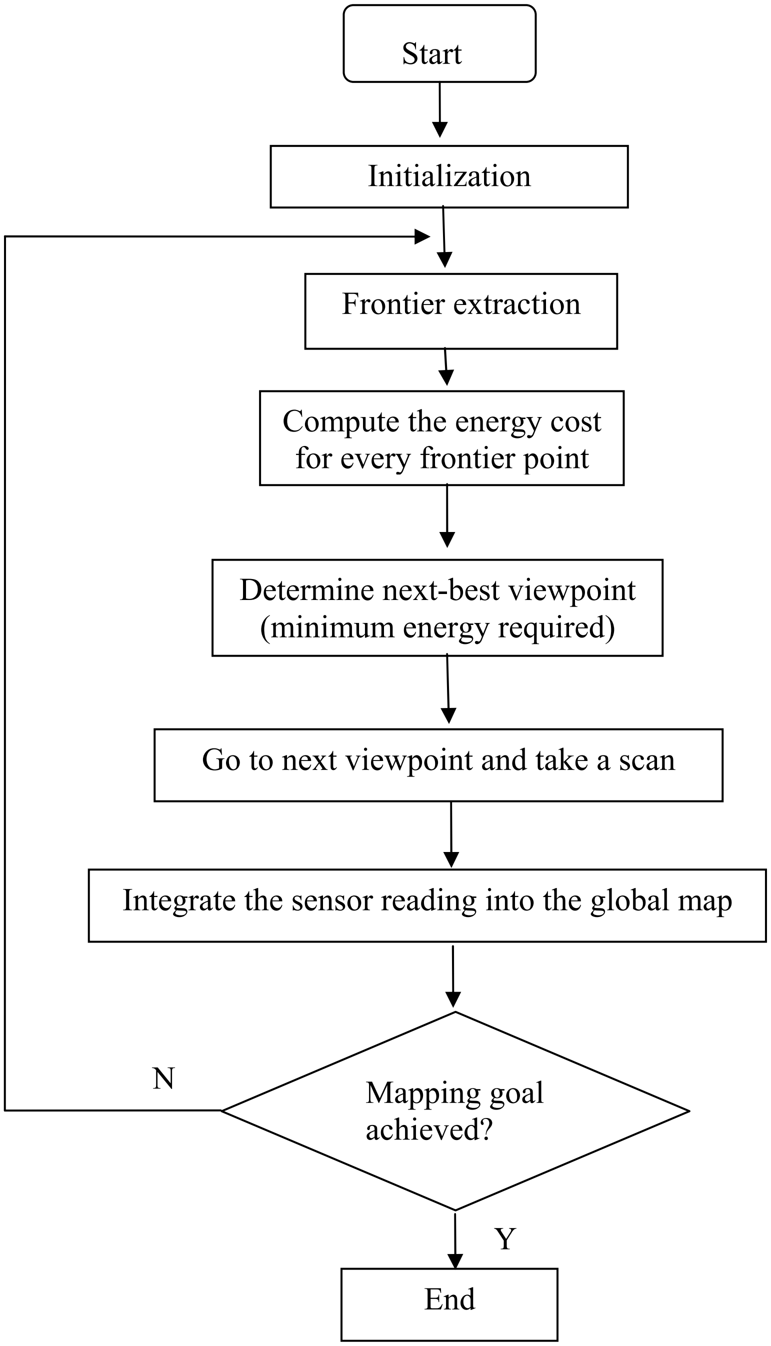

Start
Initialization
Frontier extraction
Compute the energy cost for every frontier point
Determine next-best viewpoint (minimum energy required)
Go to next viewpoint and take a scan
Integrate the sensor reading into the global map
Mapping goal achieved?
N
Y
End

Figure 4.1 The flow chart of the greedy method based on the minimum energy requirement.

4.6 Results and discussion

4.6.1 Algorithm validation

The efficiency of a mapping strategy is difficult to quantify because robotic map building is a complex task. Some researchers (Gonzalez-Banos and Latombe, 2002; Taylor and Kriegman, 1998; Thrun et al., 1998; Thrun et al., 2005) have shown the validity of algorithms by presenting paths generated to explore given environments. A few researchers (Moorehead et al., 2001; Sim and Dudek, 2003; Stachniss and Burgard, 2003; Sujan and Dubowsky, 2005) opted to show the efficiency of proposed exploration strategies by comparing their results with other strategies.

In this chapter, the performance of the algorithm will be compared with other strategies such as random selection, line sweeping/raster, and spiral pattern policies.

(1) *Random NBV selection*—the next best viewpoint is selected randomly within the frontiers of the environment in every step.

(2) *Line sweeping method*—the next best viewpoint is selected in a sequence along the line sweeping direction in the unknown environment. This algorithm is described in detail in Chapter 3. A number of circles with radius *Rc* are found to completely cover the whole environment. A path connecting the centers of circles in a sequence following the line sweeping direction is applied in the exploration procedure. When the selected next viewpoint is located in the known environment, Dijkstra’s shortest path algorithm is applied to plan an optimal path between the current robot location and the destination. Otherwise, the robot will choose the boundary edge that is closest to the selected sampling point as the destination and plan an optimal path to this destination using Dijkstra’s shortest path algorithm.

(3) *Spiral pattern method 1*—the next best viewpoint is selected in a sequence along the spiral sweeping direction within the frontiers of the known environment in every step. The robot first travels along the boundary of unknown fields to collect the boundary terrain information with the image sensor. Frontiers are defined as the points along the boundary between the

known area and unknown terrain. The predefined path is defined in a sequence along the spiral sweeping direction within the frontiers. Dijkstra's shortest path algorithm is applied to plan an optimal path between the current viewpoint and the next best viewpoint. After the robot has traversed all the frontiers for this step, it will define new frontiers based on the updated map for the next iteration until the mapping task is finished or the robot is blocked.

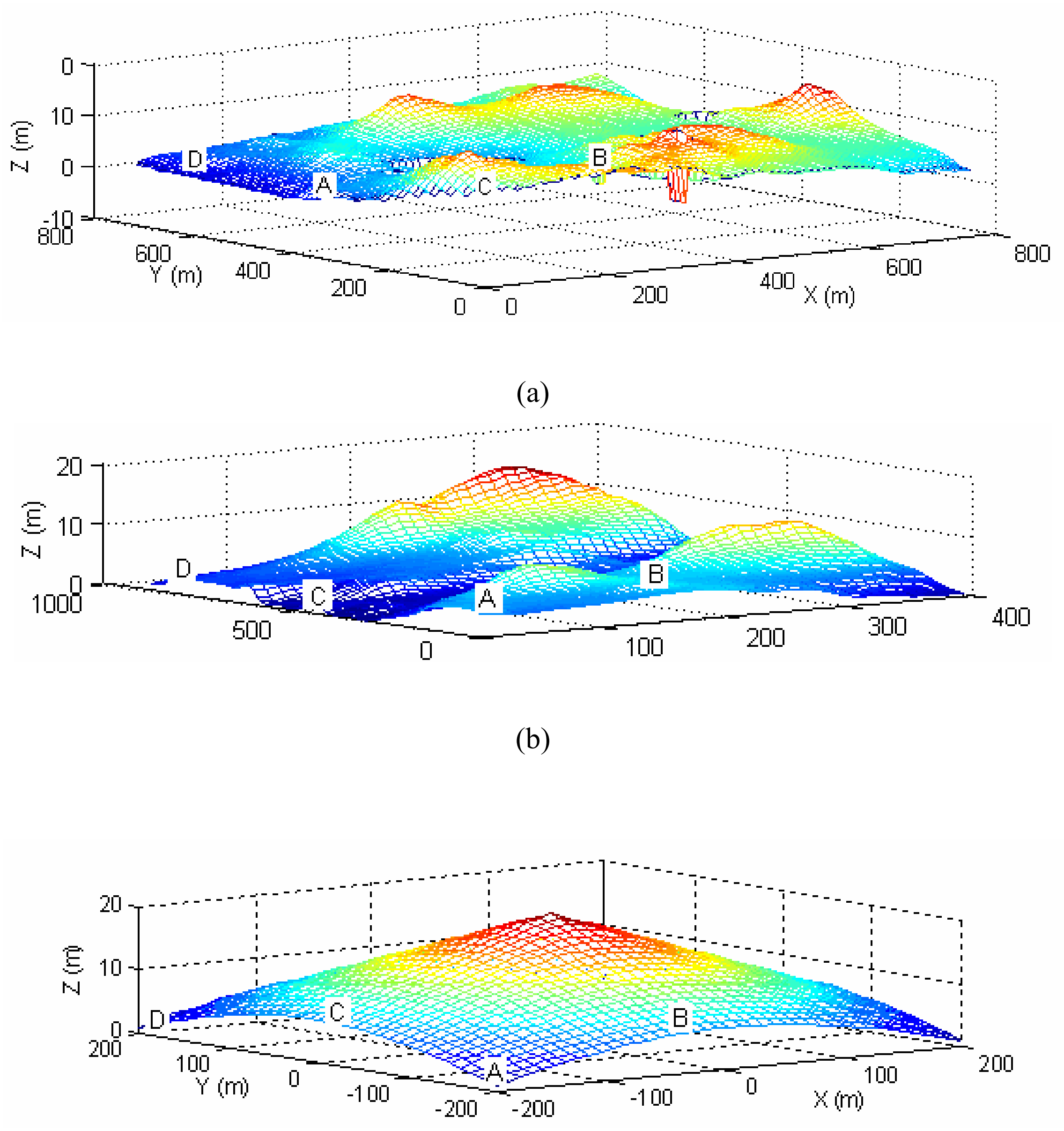
(4) *Spiral pattern method 2*—the next best viewpoint is selected in sequence along the spiral sweeping direction. A number of circles with radius *Rc* are found to completely cover the whole environment. A path that connects the centers of circles in a sequence following the spiral pattern sweeping direction is applied in the exploration procedure. Dijkstra's shortest path algorithm is applied to find a path between the current robot location and the destination when the next viewpoint is located in the known environment. Otherwise, the robot will choose the boundary edge that is closest to the selected sampling point as the destination.

4.6.2 Test results

A simulated four-wheel drive robot (mass: 16 kg; length: 50 cm; width: 49 cm; height: 26 cm) with four identical wheels (wheel diameter: 25.2 cm; wheel width: 7.5 cm) was used in the simulation. The simulated robot was assumed to have a 3D image sensor (with a 90° field of view, 50 m depth of field, 1:1 aspect ratio) installed as the vision system. The image sensor could rotate 360˚ horizontally without any cost.

The robot was placed in five unknown testing fields as shown in Fig. 4.2. Field 1 (780 meters by 800 meters) and field 2 (396 meters by 796 meters) are unknown agricultural fields, while the other 3 fields, included an ideal mountain environment, an ideal hole environment, and an ideal slope environment. The size for the three ideal fields is 400 meters by 400 meters. A uniform CI of 75 N/cm^2 is assumed for all the five silt fields (Goering et al., 2003). The robot started at points A, B, C, and D for each of the fields shown in Table 4.1.

It is assumed that the robot traveled at a steady speed of 3 m/s and the acceleration or deceleration state was not considered in order to simplify the problem. The exploration continued in the agricultural field until all reachable terrain was explored or expected new terrain information was negligible.

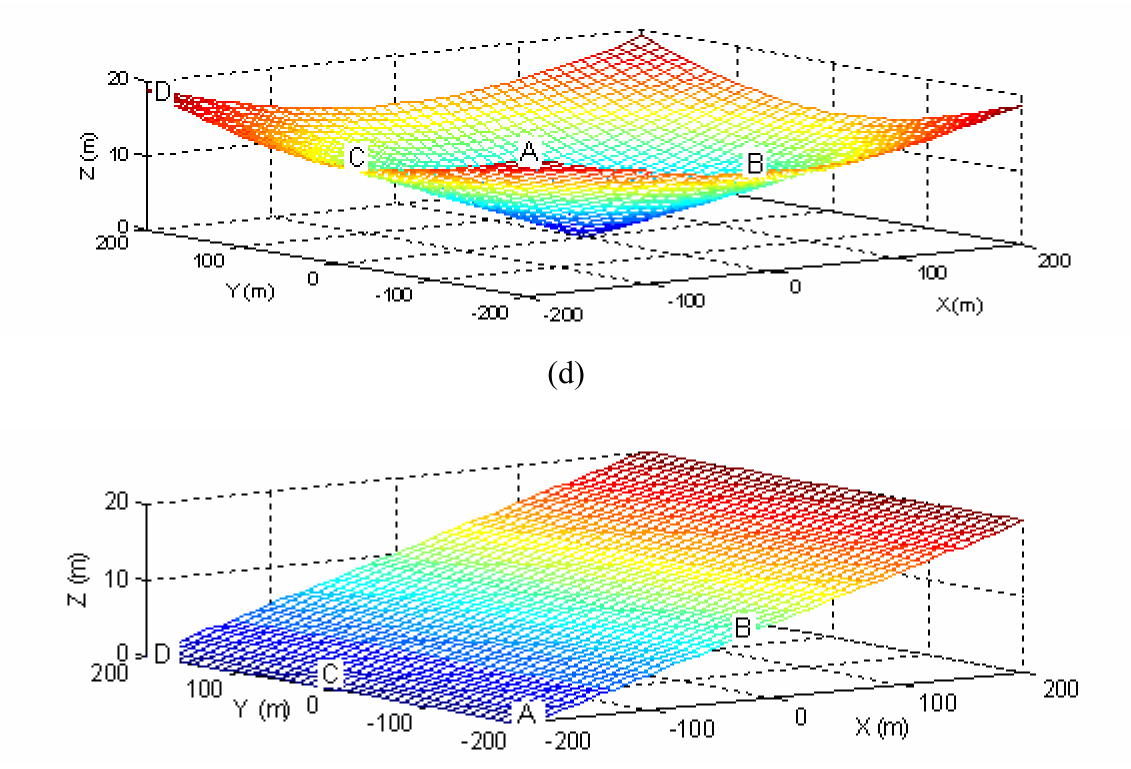


(a)

(b)

(c)

Figure 4.2 3D maps of test fields (an area with negative z values represents unreachable parts): (a) agricultural field 1, (b) agricultural field 2, (c) an ideal mountain environment.



(d)

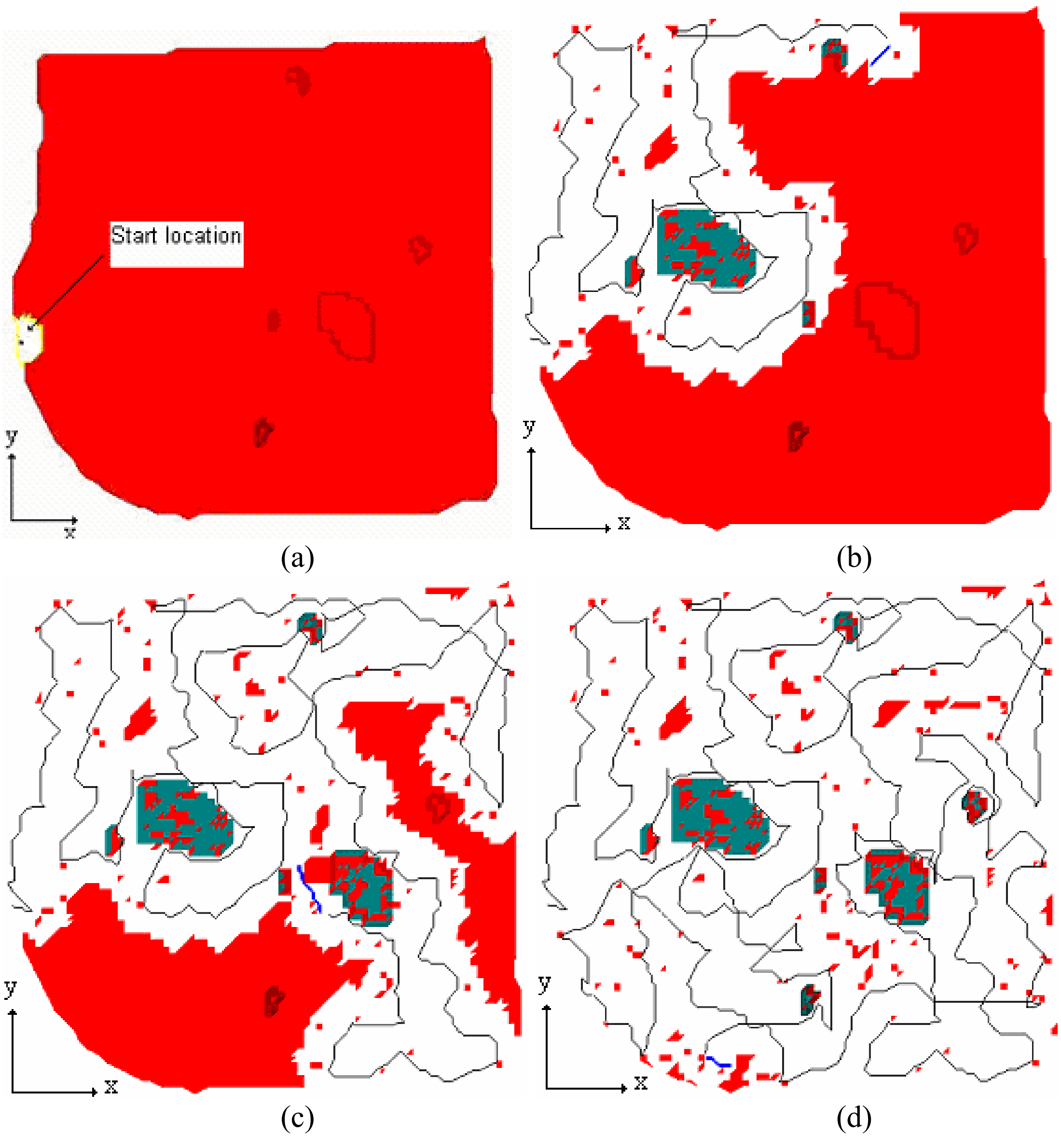
(e)

Figure 4.2 (Continued) 3D maps of test fields (an area with negative z values represents unreachable parts): (d) an ideal hole environment, (e) an ideal slope environment.

Table 4.1: Robot starting locations in 5 fields

Field	Starting A (x ,y) in units (m, m)	Starting B (x,y) in units (m, m)	Starting C (x,y) in units (m, m)	Starting D (x,y) in units (m, m)
Agricultural 1	(20,300)	(380,30)	(200,30)	(50,700)
Agricultural 2	(2,2)	(200,20)	(20,400)	(20,770)
Ideal mountain	(-198,-198)	(0,-198)	(-190,0)	(-190,190)
Ideal hole	(-198,-198)	(0,-198)	(-190,0)	(-190,190)
Ideal slope	(-198,-198)	(0,-198)	(-190,0)	(-190,190)

Captured intermediate screenshots of the simulation in agricultural field 1 (starting location A) are shown in Fig. 4.3, where the traveled path (black lines) and updated map (white parts are explored terrain) are plotted at selected iteration times. Figure 4.3(d) shows the exploration result when the exploration using the greedy algorithm ended after 208 iterations. In the end, 95 percent of the terrain has been explored with a travel distance of 11273 m in 72 minutes.

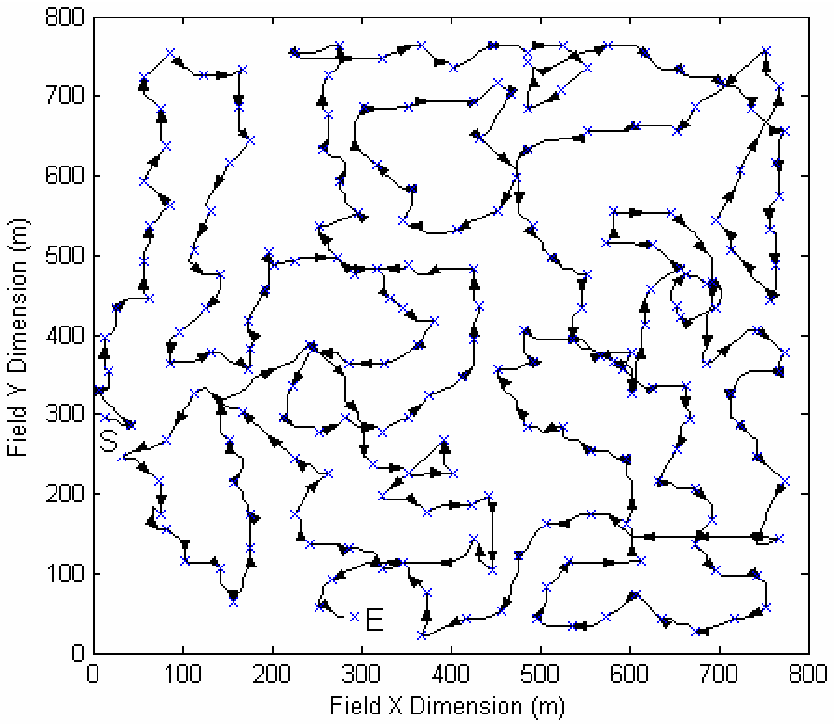


(a) (b) (c) (d)

Figure 4.3 The traveled paths (black lines) and updated maps (white parts are explored terrain) that resulted from the greedy method are plotted at selected iteration times: (a) initialization, (b) after 70 iterations(34.5% of the terrain was explored), (c)

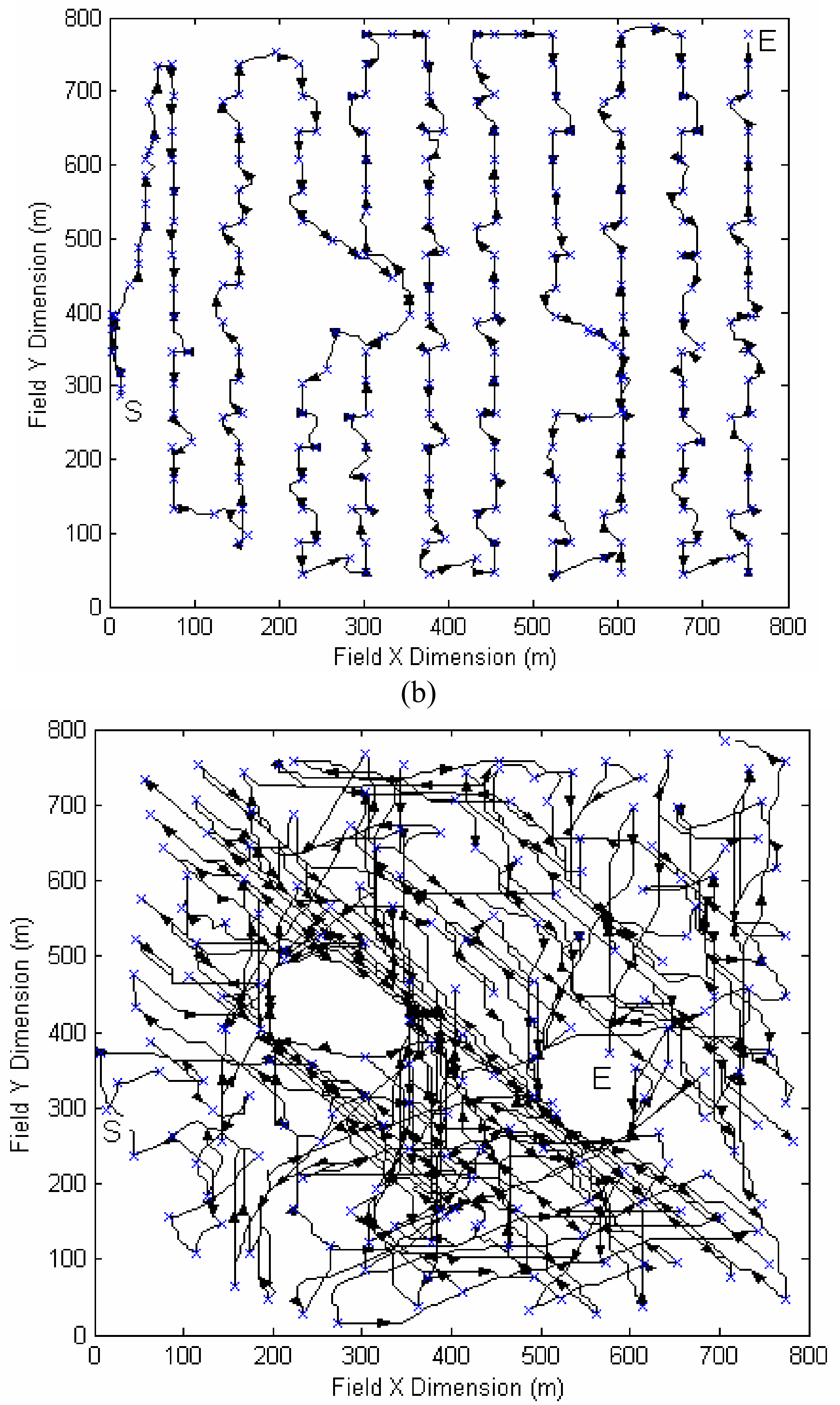
after 140 iterations(70% of the terrain was explored), (d) after 208 iterations(95% of the terrain was explored).

The trajectory paths generated by the greedy, line sweeping, random, spiral pattern 1, and spiral pattern 2 methods are given in Figs. 4.4(a) through 4.4(e), respectively. The black lines represent the path, the arrows show the vehicle's travel direction, and the cross marks represent viewpoints where the robot stopped to take a scan. They show that the path generated by the random method has much more overlap than those generated by other methods. It is also shown that the path generated by spiral pattern 2 has more overlap than that generated by spiral pattern 1, while both spiral patterns have a little more overlap than those generated by the line sweeping and greedy methods.



(a)

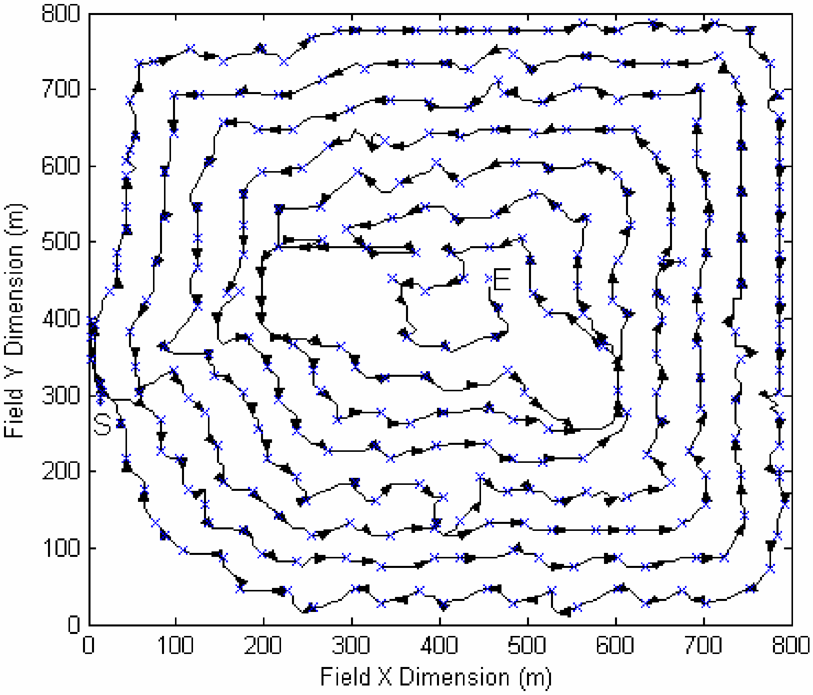
Figure 4.4 Trajectories generated in the simulation (arrows represent travel direction; cross marks represent viewpoints; started at S and ended at E): (a) greedy method.



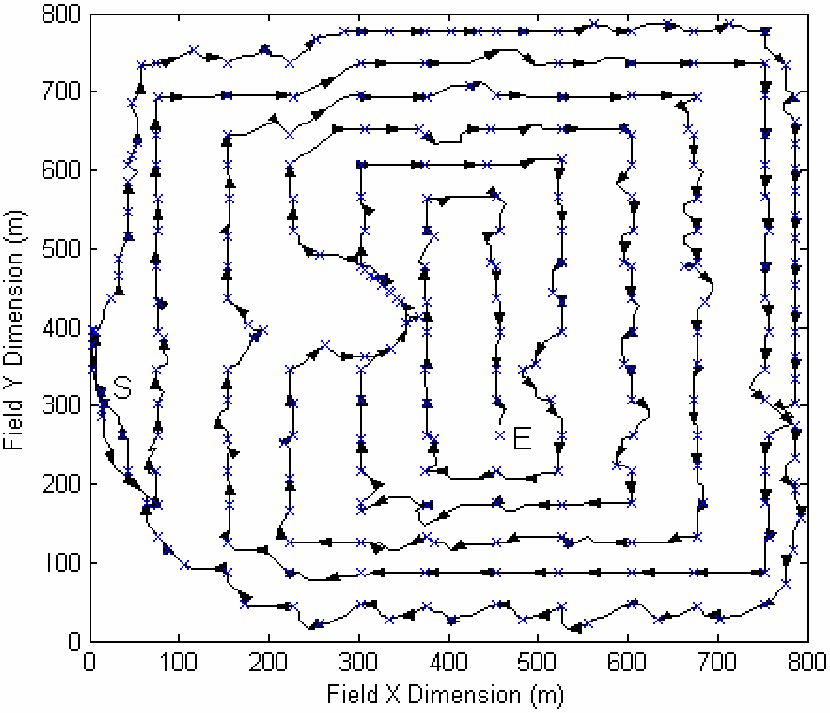
(b)

(c)

Figure 4.4 (continued) Trajectories generated in the simulation (arrows represent travel direction; cross marks represent viewpoints; started at S and ended at E): (b) line sweeping method, (c) random method.



(d)

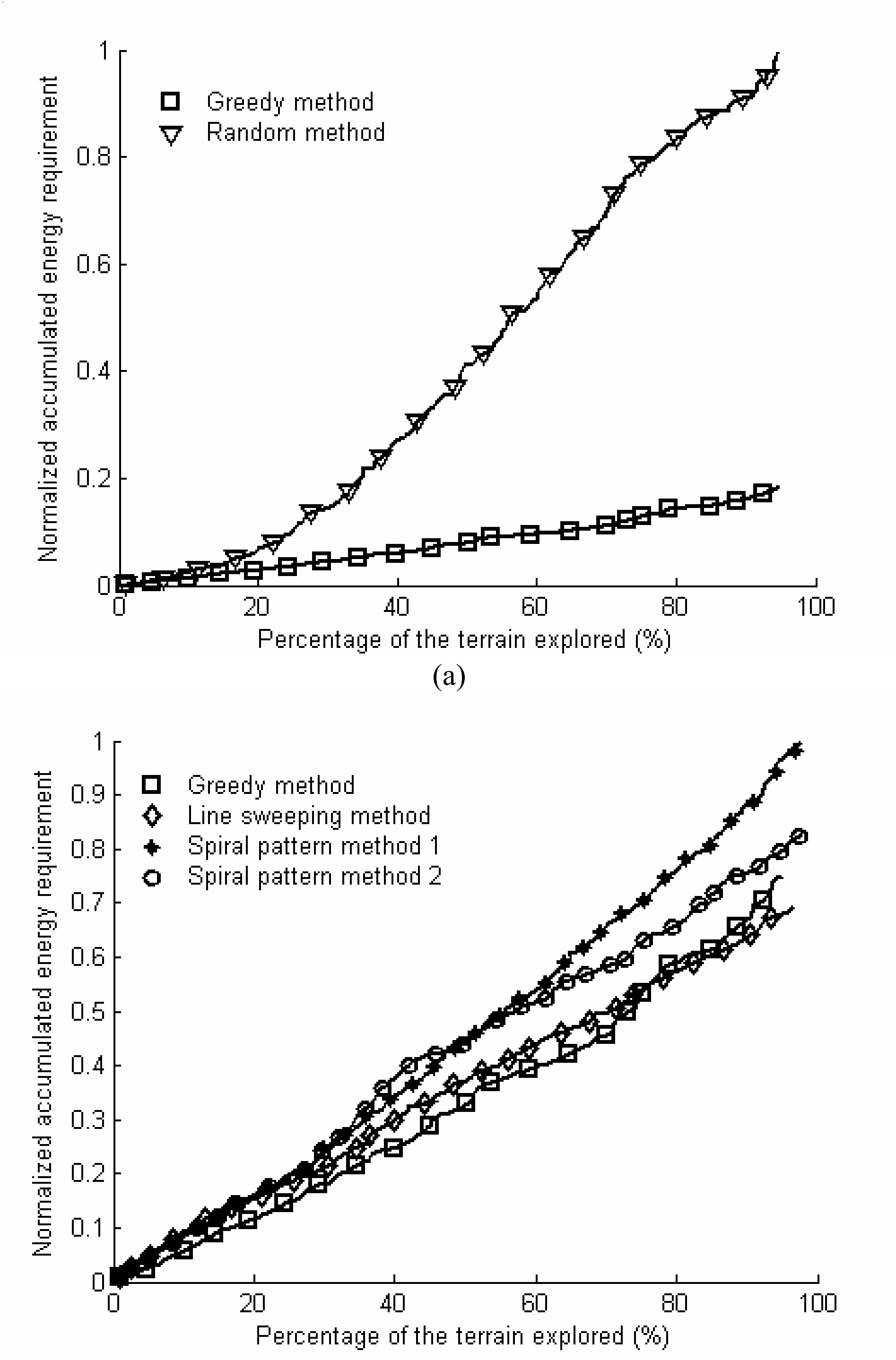


(e)

Figure 4.4 (continued) Trajectories generated in the simulation (arrows represent travel direction; cross marks represent viewpoints; started at S and ended at E): (d) spiral pattern 1, (e) spiral pattern 2.

Figures 4.5(a), 4.6(a), 4.7(a), and 4.8(a) show the relationships between the fraction of the environment mapped and the normalized relative energy requirement, distance traveled, time requirement (including planning time and navigation time), and scan number, respectively, for the robot's exploration of agricultural field 1 starting from location A for the random and greedy exploration strategies. It is desirable to have a large fraction of the environment mapped with a low amount of energy consumed, a short traveled distance, and a low number of scans in a shorter period of time. It is apparent that the path length, energy requirement, and time requirement (including both the planning time and traveling time) of the greedy method are substantially smaller than those for the random method, whereas the fraction of the terrain explored is greater in the former. These results show a stark contrast between the two methods and illustrate the effectiveness of the greedy method in its ability to minimize energy demand during exploration.

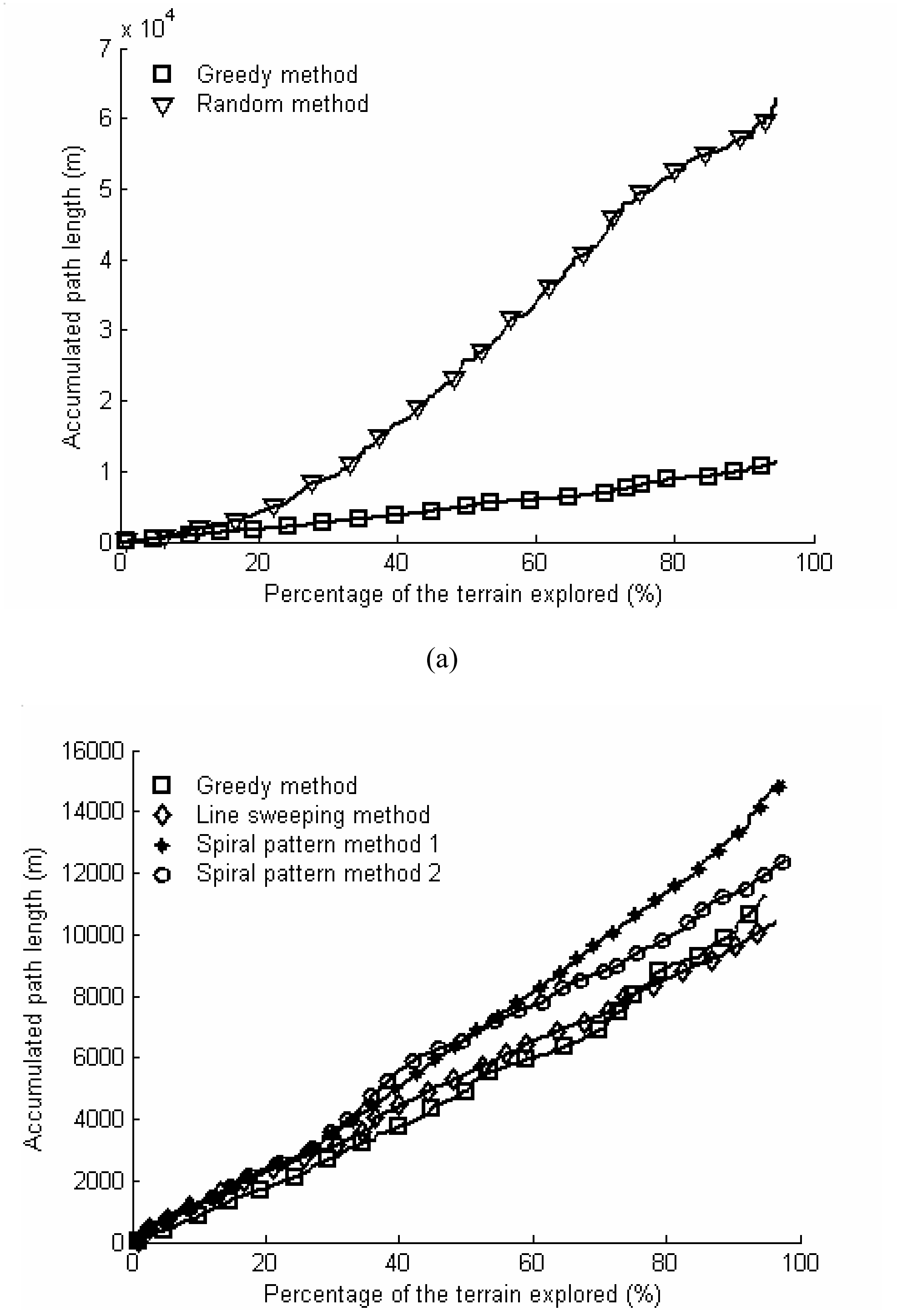
Figures 4.5(b), 4.6(b), 4.7(b), and 4.8(b) show the relationships between the fraction of the environment mapped and the energy requirement, distance traveled, time requirement, and scan number, respectively, for the robot's exploration of an agricultural field starting from location A for the four exploration strategies. It is also apparent that the path length, energy requirement, and time requirement of the greedy method are smaller than those for the line sweeping, spiral pattern 1, and spiral pattern 2 methods in the first 75% of the exploration task. However, during the final 25% of the exploration task, the energy consumption, path length, and time requirement of the greedy method increased drastically, so the line sweeping method overtook the greedy method in the last stage of the exploration task. Compared with the spiral pattern strategies, the greedy method still consumed less energy during the whole exploration. It required 0.32 units of energy to complete 50% of the exploration using the greedy method, while it required 0.37, 0.45, and 0.43 units to complete the same percentage of the exploration for the line sweeping, spiral pattern 1, and spiral pattern 2 methods, respectively. To complete 90 percent of the exploration, it took 0.66, 0.63, 0.88, and 0.75 units for the greedy, line sweeping, spiral pattern 1, and spiral pattern 2 methods, respectively. It is also showed that spiral pattern 2 consumed less energy, had a shorter path length, and required less time than spiral pattern 1.



(a)

(b)

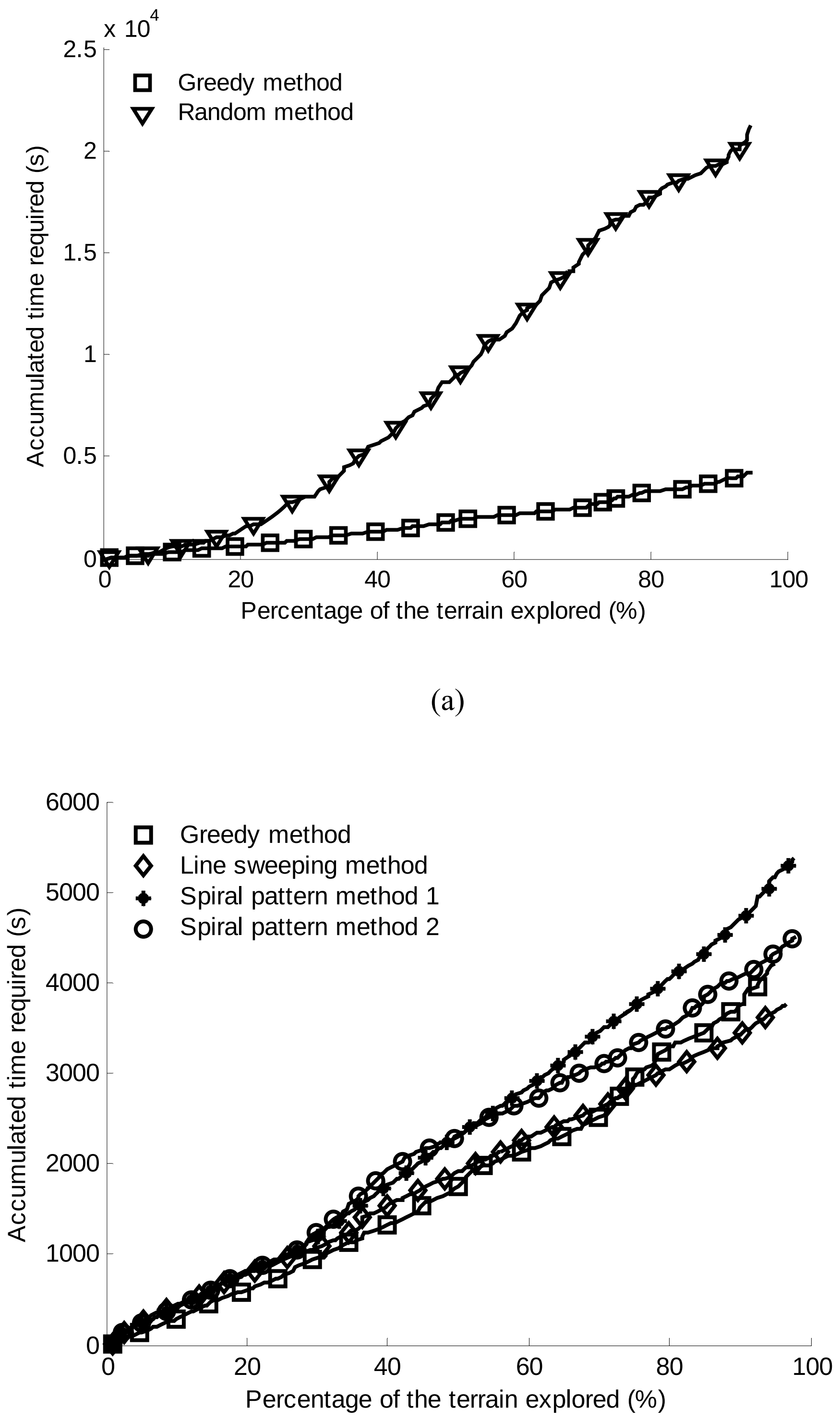
Figure 4.5 Result of the autonomous mapping of agricultural field 1 (start A): normalized relative energy requirement (the nominal energy is divided by the maximum value) of the exploration as a function of fraction of explored terrain.



(a)

(b)

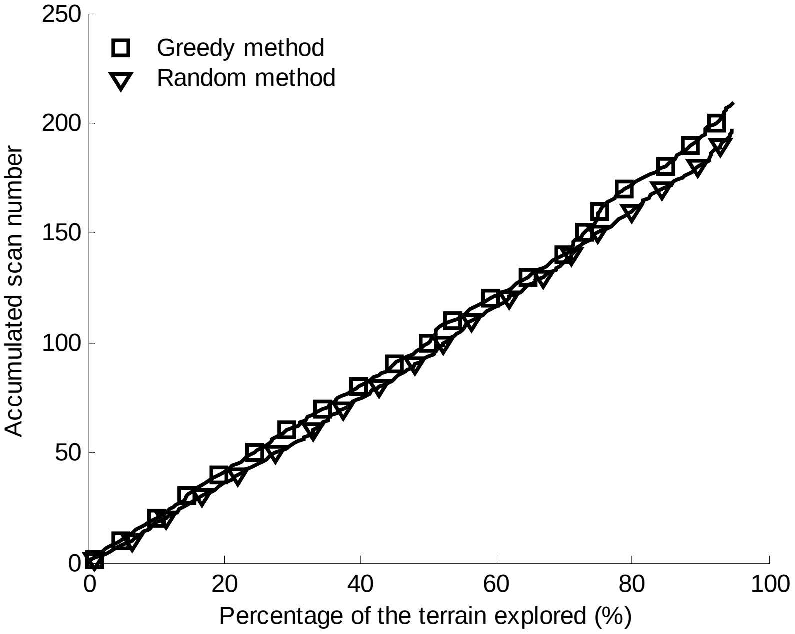
Figure 4.6 Result of the autonomous mapping of agricultural field 1 (start point A): accumulated path length traveled by the robot as a function of fraction of explored terrain.



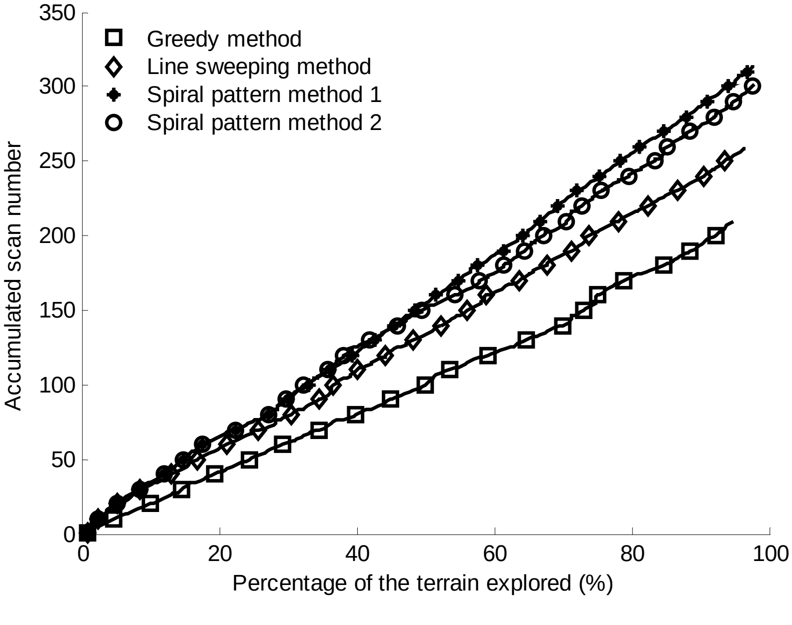
(a)

(b)

Figure 4.7 Result of the autonomous mapping of agricultural field 1 (start point A): accumulated time required for the exploration as a function of fraction of explored terrain.



(a)



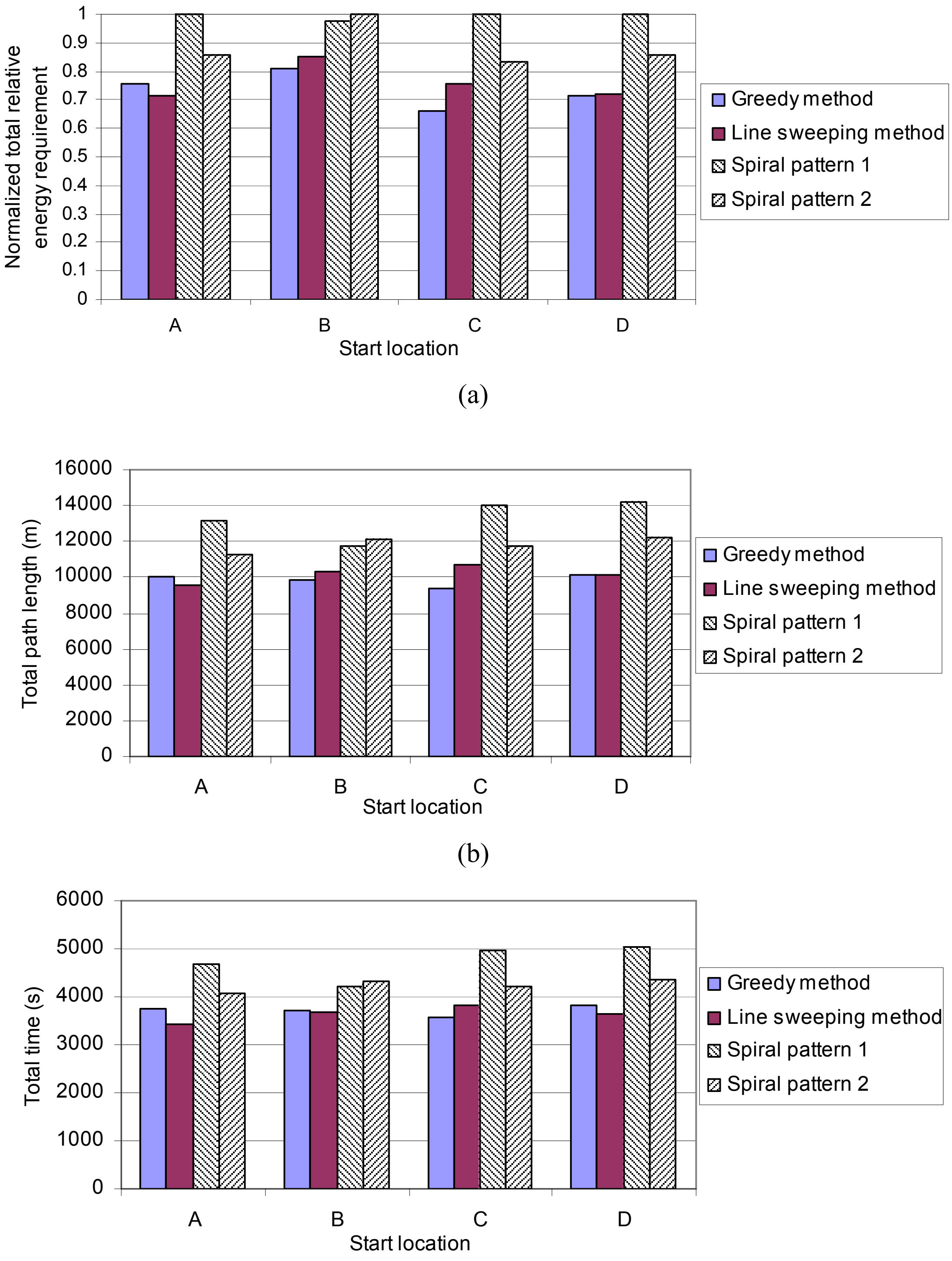
(b)

Figure 4.8 Result of the autonomous construction of agricultural field 1 map (starting location A): the number of scans as a function of fraction of explored terrain.

In terms of performance based on the number of scans, Fig. 4.8(b) shows that the greedy method has a great advantage over the line sweeping and spiral pattern methods. However, the random method performs a little better than the greedy method (Fig. 4.8(a)). It required 193 scans to complete 90% of the exploration using the greedy method, while it required 238, 287, and 274 scans using the line sweeping, spiral pattern 1, and spiral pattern 2 methods, respectively. The random method only needed 180 scans to explore the same portion of the terrain. The relationship between the fraction of the environment mapped and the scan number is linear for all strategies.

Figures 4.9(a), 4.9(b), 4.9(c), and 4.9(d) show the performance of the four exploration strategies on the normalized relative energy requirement, distance traveled, time requirement, and scan number, respectively, for exploring 90% of agricultural field 1 starting from 4 different locations. It was shown that the total path length, energy requirement, time requirement, and scan number of the spiral pattern methods were much larger than those for the line sweeping and the greedy methods in all the four starting locations. The spiral pattern 2 outperformed the spiral pattern 1 in 3 of 4 starting locations.

Figures 4.9(a) and 4.9(b) also show that the total path length and energy requirement of the greedy method are smaller than those for the line sweeping method when starting at locations B and C; however, the line sweeping method performs better than the greedy method when starting at A, and there was little difference for the two methods in starting at D. In terms of performance based on time, Fig. 4.9(c) showed that line sweeping matched or outperformed the greedy methods in most cases. The greedy method required much more calculation than the line sweeping method in every iterative step. In terms of performance based on the scan number, Fig. 4.9(d) shows that the greedy method has great advantage over the line sweeping and spiral pattern methods.

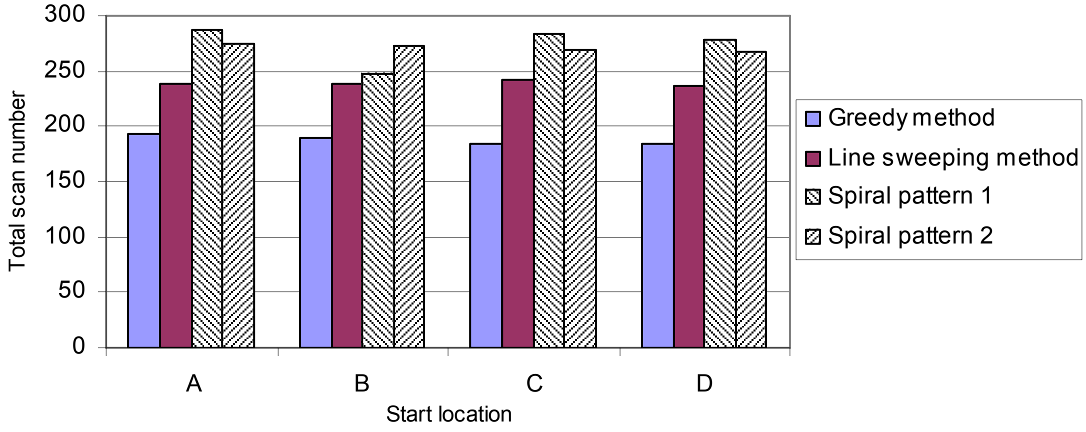


(a)

(b)

(c)

Figure 4.9 Performance for achieving 90% of the exploration task for the automatic mapping of agricultural field 1 and four different starting points: (a) normalized total relative energy requirement, (b) total path length, (c) total time.



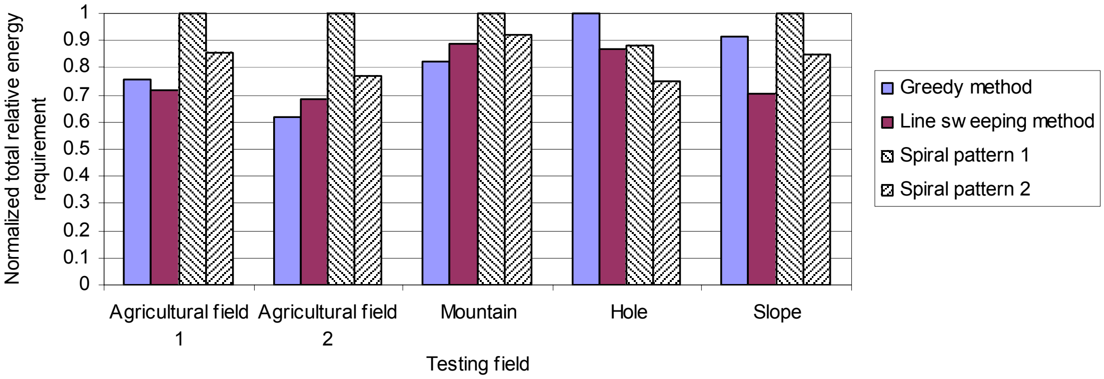
(d)

Figure 4.9 (continued) Performance for achieving 90% of the exploration task for the automatic mapping of agricultural field 1 and four different starting points: (d) total scan number.

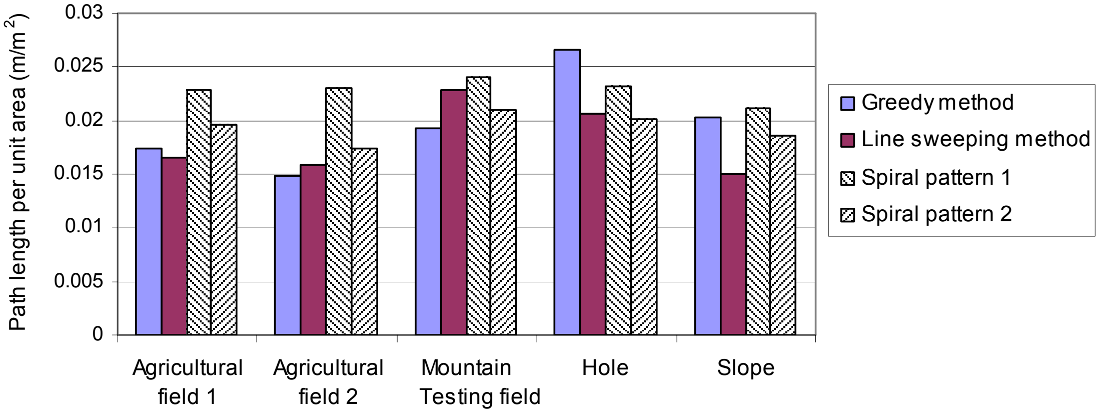
Figures 4.10(a), 4.10(b), 4.10(c) show the performance of the four exploration strategies on the normalized relative energy requirement, distance traveled, and time requirement per unit area respectively, for 90% of the exploration tasks with 5 testing fields and a single starting location (location A) for each field. It is showed that the total path length, energy requirement, and time requirement for the greedy method were much smaller than those for the other methods in agricultural field 2 and mountain environment. The line sweeping method performed best in agricultural field 1 and the ideal slope, while spiral pattern 2 outperformed other methods in the hole environment.

In terms of normalized relative energy requirement, the line sweeping method and spiral pattern 2 performed best in the ideal slope and hole environments, respectively. The result is consistent with expectations. It is surprising that the spiral pattern 2 did not perform best in the mountain environment; rather, the greedy method outperformed the spiral pattern 2, which can be explained by their trajectory paths. The paths generated (shown in Appendix C, Fig. C.29) by the spiral pattern 2 and the line sweeping methods show a similar pattern for the mountain environment. It is also demonstrated that the greedy method can adapt to different field patterns.

Figures 4.10(d) show the performance of the four exploration strategies on the number of scans for 90% of the exploration of 5 testing fields starting from location A for each field. The numbers of scans per unit area for the greedy methods were much smaller than other methods for all testing fields except for the slope environment, where the greedy method tied with the line sweeping method. This was consistent with the previous results for agricultural field 1 starting from different locations.

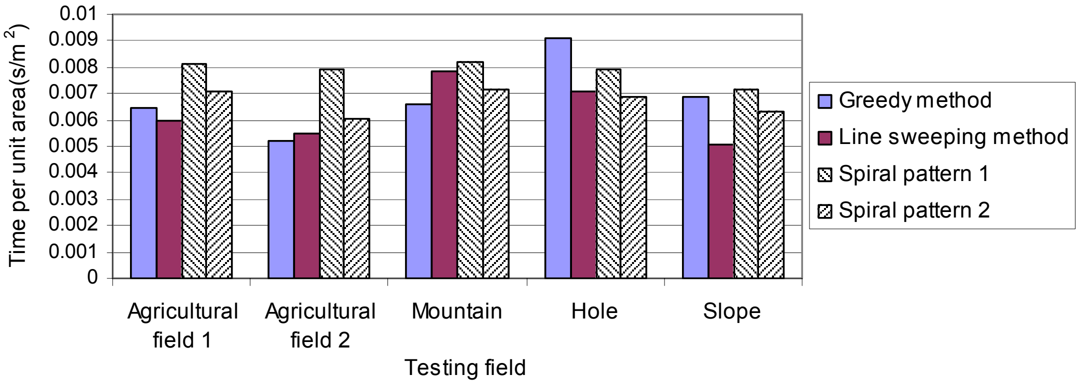


(a)

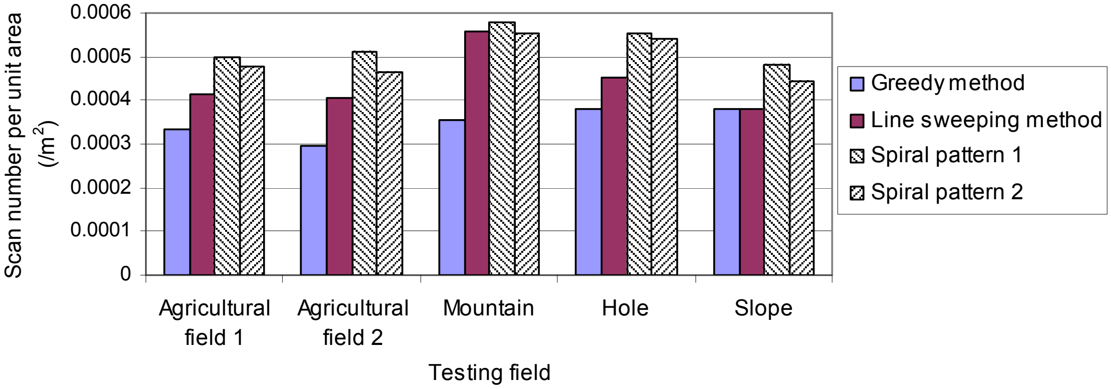


(b)

Figure 4.10 Performance for 90% of the exploration task of the automatic mapping of 5 testing fields starting location A: (a) normalized total relative energy requirement, (b) path length per unit area.



(c)



(d)

Figure 4.10 (continued) Performance for 90% of the exploration task for the automatic mapping of 5 testing fields, starting at location A: (c) time per unit area, (d) scan number per unit area.

4.7 Conclusions

In this chapter, the problem of autonomous mapping was addressed for large, rough, unstructured agricultural environments. A new triangular mesh map was used that allowed the robot to maintain a very rich representation of the environment and to robustly perform exploration. The problem was addressed with the development of a novel energy cost function to plan next best viewpoints. The energy cost function considered the distance, terrain elevation change, vehicle slip rate, and other vehicle parameters. Simulation results in two typical western Canadian agricultural fields and three virtual environments were presented to demonstrate the algorithms.

The greedy method required about 80% less energy, distance, and time than the random method for all 4 start points to complete 50% and the whole exploration task. The path length, energy requirement, and time requirement of the greedy method were smaller than those for the line sweeping, spiral pattern 1, and spiral pattern 2 methods in the earlier stage of the exploration task. The greedy method required between 12% and 48% energy, between 10% and 47% length, between 8% and 45% time less than the other three methods, including the line sweeping, spiral method 1, and spiral method 2, for all start points A, B, C, and D to complete 50% exploration task.

However, after completing 90% of the exploration task, the line sweeping method outperformed the greedy method by about 5% in terms of energy when starting A, while the greedy method exceeded the line sweeping method by 5%, 12%, and 0.5% in terms of energy efficiency for start location B, C, and D, respectively. The spiral pattern 2 performed better than the spiral pattern 1 in terms of energy consumption, distance, and time by about 15% for start locations A, C and D. However, the spiral pattern 1 outperformed spiral pattern 2 slightly by 2 % in terms of energy.

The number of scans took using the greedy method was smaller than the number of scans required by the line sweeping, spiral pattern 1, and spiral pattern 2 methods during the entire exploration, while there was not much difference between the number of scans taken using the greedy and random methods. It required 193 scans to complete 90% of the exploration using the greedy method, while it required 238, 287, 274, and180 scans using the line sweeping, spiral pattern 1, and spiral pattern 2, and random methods, respectively. The relationship between the fraction of the environment mapped and the scan number was linear for all strategies.

Finally the terrain types and starting location had influences on the performance of the exploration methods. The greedy method can adapt to different field patterns, and the minimum energy demand was achieved by minimizing the energy cost during the early stages of the exploration task.

4.8 References

Bishop, T. F. A. and A. B. McBratney. 2002. Creating field−extent digital elevation models for precision agriculture. *Precision Agriculture* 3(1): 37-46.

Choset, H. 2001. Coverage for robotics—A survey of recent results. *Annals of Mathematics and Artificial Intelligence* 31: 113-126.

Clark, R. L. and R. Lee. 1998. Development of topographic maps for precision farming with kinematic GPS. *Transactions of the ASAE* 41(4): 909-916.

Cormen H.T., C.E. Leiserson, R.L. Rivest and C. Stein. 2001. *Introduction to algorithms*, second edition. MIT Press and McGraw-Hill. Cambridge/NewYork.

Goering, C.E., M.L. Stone, D.W. Smith and P.K. Turnquist. 2003. *Off-Road Vehicle Engineering Principles*. St. Joseph, MI: ASABE. 474 pages.

Gonzalez-Banos, H.H. and J.C. Latombe. 2002. Navigation strategies for exploring indoor environments. In *International Journal of Robotics Research* 21(10-11): 829-848.

Hert, S., S. Tiwari and V. Lumelsky. 1996. A terrain covering algorithm for an AUV. *Autonomous Robots* 3: 91-119.

Huang, W. 2001. Optimal line-sweep-based decompositions for coverage algorithms. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*: 27- 32. Seoul, Korea, May 21-26.

Kitware Inc. 2005. The Visualization Toolkit. *http://www.vtk.org.* Accessed June 26, 2005. New York.

Moorehead, S., R. Simmons and W.L. Whittaker. 2001. Autonomous exploration using multiple sources of information. In *Proceedings of the IEEE International Conference on Robotics and Automation* 3: 3098- 3103. Seoul, Korea, May 21-26.

Schmidt, J. P., R. K. Taylor and R. J. Gehl. 2003. Developing topographic maps using a submeter accuracy global positioning receiver. *Applied Engineering in Agriculture* 19(3): 291-300.

Schroeder, W., K. Martin and B. Lorensen. 1996. The Visualization Toolkit: an object-oriented approach to 3D graphics. Prentice Hall PTR. Upper Saddle River, NJ.

Sim, R. and G. Dudek. 2003. Effective exploration strategies for the construction of visual maps. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*: 3224-3231. Las Vegas, NV, October 27-31.

Stachniss, C. and W. Burgard. 2003. Exploring Unknown Environments with Mobile Robots using Coverage Maps. In *Proceedings of the International Joint Conference on Artificial Intelligence(IJCAI)*: 1127-1134. Acapulco, Mexico. August 9-15.

Sujan,V. A. and S. Dubowsky. 2005. Efficient information-based visual robotic mapping in unstructured environments. *International Journal of Robotics Research* 24 (4): 275-293.

Suvinen, A., M. Saarilahti, and T. Tokola. 2003. Terrain mobility model and determination of optimal off-road route. In *Proceedings of the 9th Scandinavian Research Conference on Geographical Information Science*: 251-259. Espoo, Finland, July 4-6.

Taylor, C. and D. Kriegman. 1998. Vision-based motion planning and exploration algorithms for mobile robots. *IEEE Transactions On Robotics and Automation* 14(3):147-427.

Thrun, S., D. Fox and W. Burgard. 1998. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning* 31:29-53.

Thrun, S., S. Thayer. W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverte and W. Whittaker. 2005. Autonomous Exploration and Mapping of Abandoned Mines. *IEEE Robotics and Automation Magazine* 11(4): 79-91.

Westphalen, M. L., B. L. Steward and S. Han. 2004. Topographic mapping through measurement of vehicle attitude and elevation. *Transactions of the ASAE* 47(5): 1841-1849.

Wong, J.Y. 1978. Theory of ground vehicles, John Wiley & Sons. New York/Toronto.

Yamauchi, B. 1997. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation:* 146-151. Monterey, CA. July 10-11.

Yao, H. and R. L. Clark. 2000. Development of topographic maps for precision farming with medium accuracy GPS receivers. *Applied Engineering in Agriculture* 16(6): 629-636.

5. INFORMATION-BASED EXPLORATION ALGORITHMS FOR ROUGH TERRAIN MODELING USING TRIANGULAR MESH MAPS

5.1 Significance

This chapter is associated with objective 3 of the thesis, as stated in Section 1.2. The discussion in the previous chapter demonstrated that the scan number, path length, energy requirement, and time requirement of the greedy method based on energy consumption are less than those of the pattern exploration models. The review of automatic exploration in Chapter 2 revealed that information based exploration might be more efficient in terms of the number of scans and travel cost. The specific objective of this chapter was to exploit the potential of considering the information gain in greedy mapping strategies.

In this chapter, two methods to estimate possible new terrain in one spot using a 3D image sensor will be presented. For the first method, assuming a partly known environment, the information gain will be estimated by applying the ray tracing algorithm to the known part of the environment. For the second method, the new information gain will be calculated using polygon clipping in an unknown environment.

A complete comparison of the energy requirement, time consumption, and number of scans of the different methods will be presented to show the effectiveness of the information based exploration policy.

5.2 Introduction

The main objective of exploration is to create an accurate map of an unknown area (Bourgault et al., 2002). To create an accurate map, a robot needs to know where it is, plan where to go next, collect the terrain information with sensor readings, and build the map to represent the environment. Therefore, the exploration task requires robotic systems to meet the objectives of both localization accuracy and exploring efficiency (Bourgault et al., 2002). This work mainly focuses on the next best viewpoint algorithm.

Different exploration strategies have been used in the environmental modeling task. One group of exploration strategies is to choose the closest point as the next best viewpoint among frontiers (Yamauchi, 1997) extracted from the boundary between the known and unknown areas (Gonzalez-Banos and Latombe, 2002; Taylor and Kriegman, 1998; Thrun et al., 1998; Yamauchi, 1997). Yamauchi (1997) first presented a frontier based robot explorer designed to explore an office environment using an occupancy grid map with a range sensor. The frontiers were defined by the known area close to the boundary between open space and uncharted territory. The robot moved to the nearest frontier by the shortest path, took a scan and updated the environmental map. The planner used a depth-first search to reach that frontier goal. The mapping procedure repeated this iterative step until the entire area had been explored.

Another family of methods chooses the next best viewpoint using safety considerations. Prestes et al. (2002) proposed an exploration approach based on harmonic functions for path planning with a grid map. Obstacles act as a repelling force and frontiers attract the robot. The robot plans the next best viewpoint using the harmonic potential calculation from a local area with the size of sensing capacity. By greedily following the gradient descent on the potential the robot maximizes knowledge gain while avoiding obstacles. Gonzalez-Banos and Latombe (2002) introduced safe navigation constraints in an indoor environment exploration strategy to construct a 2D polygonal layout of the environment with a laser range sensor. The next best viewpoint is chosen within the safe region to avoid the obstacles and under the constraint that the expected new information must have a minimal overlap with the current global map. They showed that the next best viewpoint algorithm produced strategies that cannot be easily out-done by a human operator.

A large body of studies (Bourgault et al., 2002; Sujan et al., 2006; Rocha et al., 2005; Moorehead et al., 2001; Simmons et al., 2000; Sujan and Dubowsky, 2005) has centered on information-theoretic methods based on the use of information as a measure of utility for making exploration control actions. Bourgault et al. (2002) attempted to maximize both the expected Shannon information gain and localization accuracy. Simmons et al. (2000) investigated an explore algorithm based on next best viewpoints which will provide the maximum new information gain and minimum driving cost using multiple robots. Feder et al. (1999) proposed an information metric, named Fisher information, which was used to plan next sensing positions to maximum the terrain information gain and minimize expected dead-reckoning errors. Tovar et al. (2006) developed optimal exploration strategies using a utility function which integrated the travel distance, size of the unexplored space, robot configuration uncertainty, landmark identification probability, and ability to see features like corners.

Generally, the previously described projects dealt with flat environments that consider the travel distance as the cost function. Much less work has been done in outdoor exploration. Moorehead et al. (2001) proposed a multiple information metrics exploration planner to integrate multiple sources of information in order to solve complex planetary exploration tasks. An information map was used to store multiple information sources for a 3D environment. This method enables the robot explorer to maximize the total information gained while minimizing costs such as driving, sensing, and planning. The algorithm was demonstrated by creating traversability maps and exploring cliffs. Sujan and Dubowsky (2005) developed an information-based visual robotic mapping approach based on a 3D occupancy grid map in an unstructured environment. The robot was controlled to maximize geometric knowledge gained about its environment using an evaluation function based on Shannon's information theory. They began by using the field of view of the camera to measure the new information gained in the exploration task. However, the camera model they used is a 2D model, which is not suitable for the rough terrain of an outdoor unstructured environment.

The aim of this project was to develop a next best viewpoint algorithm for the construction of topographic maps for partially-known rough agricultural fields using a 3D image sensor. Agricultural environments have enough peculiarities to make the proposed development project challenging. In agricultural environments, there is usually a commercial low-resolution topographic map available. Agricultural fields are usually rough and large, unlike the indoor office environment or flat terrain explored by previous researchers (Gonzalez-Banos and Latombe, 2002; Yamauchi, 1997). The approach in this chapter to agricultural environmental modeling has two important contributions. First, a visibility analysis method based on frustum culling and ray casting is applied to estimate the new terrain information gain using a coarse triangular mesh map. Second, energy consumption is used to represent the travel cost rather than the path distance, which was used in previous work (Simmons et al., 2000; Sujan and Dubowsky, 2005; Moorehead et al., 2001).

5.3 Triangular mesh map

Several environmental models have been implemented, including certainty grids, polygonal layouts, topological maps, triangular mesh maps, visibility graphs, and Voronoi diagrams. The triangular mesh map was used in this research to model the agricultural field surface. The triangular mesh map allows a smoother path, compared with the square grid map (Oh et al., 2004).

The triangular mesh map is incrementally built using laser sensor readings based on Delaunay triangulation (Schroeder et al., 1996). The Visualization Toolkit (Kitware Inc., 2005) has been used to implement the triangulation in this simulation. Figure 3.1 is the triangular mesh map of an agricultural field. To find the boundary and holes in a triangular mesh map, a connectivity graph is built by adding all the boundary edges according to their connectivity relationships. By visiting all the edges of the connectivity graph, all the circles, including the outer boundary and holes, will be found (Chapter 3.3). The loop of edges with the largest area is the outer boundary, while holes inside the outer boundary have smaller areas.

5.4 Exploration algorithm

5.4.1 Overview

To create a topological map using a 3D image sensor, the robot uses the iterative greedy method to plan the next best viewpoints. The objective of the greedy approach is to find an optimal path to minimize the travel cost and maximize new terrain information gain. To begin each step, the robot extracts the frontiers (Yamauchi, 1997) from the triangles close to the boundary between the known and unknown area. It constructs a voting scheme using a utility function that includes the estimated energy cost for it to travel to the frontiers and the estimated new terrain information gain for the frontiers. The robot visits the frontier at the point of the maximum utility and takes a scan using its image sensor. The map is updated by combining the new data. The robot plans the next best viewpoint with the new map until it reaches the goal of exploration or it is blocked. Figure 5.1 shows the flow diagram of this iterative algorithm.

The use of a triangular mesh to represent terrain allows the use of a graph search to easily find the next best viewpoint. The triangular mesh map is stored in the computer as a directed weighted graph. Once the graph is constructed, an optimal path between the current rover location and a destination can be planned by Dijkstra's shortest path algorithm (Cormen et al., 2001).

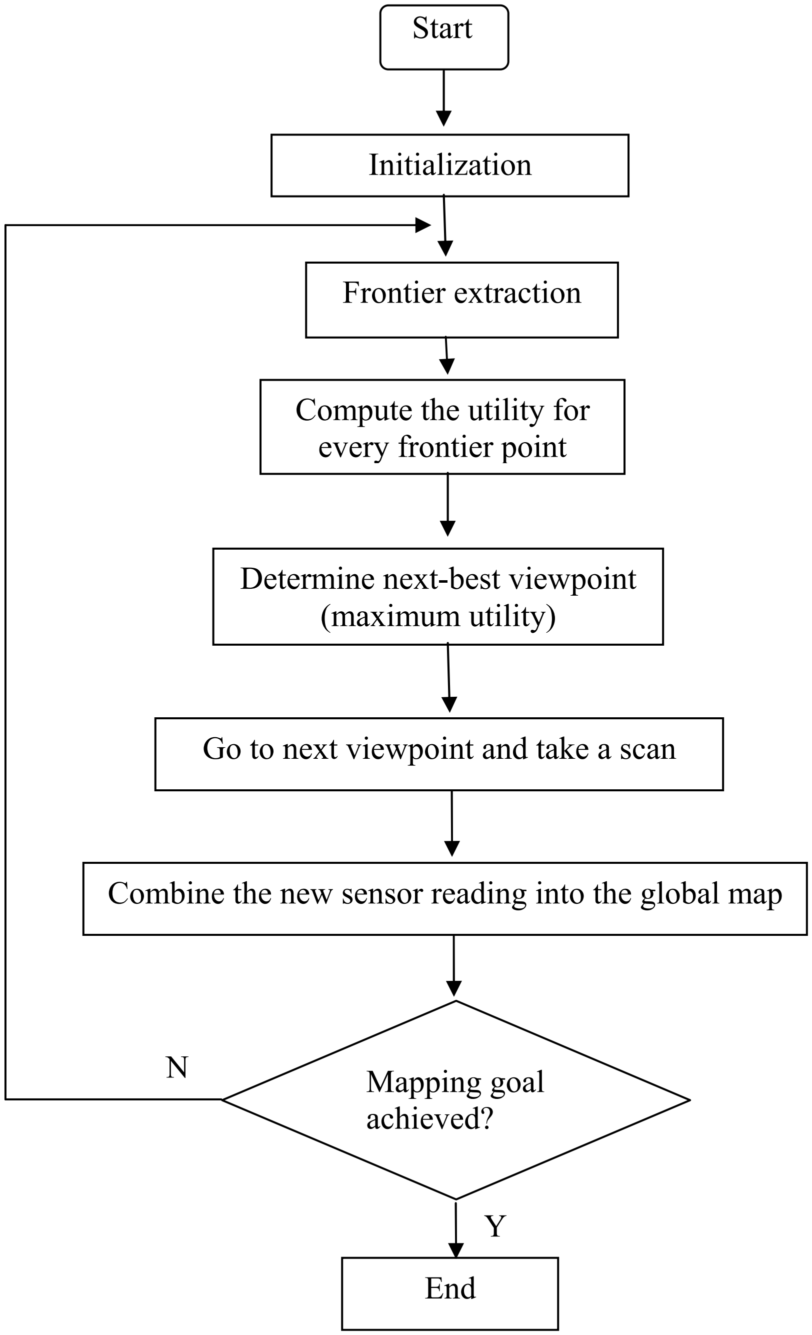

Start
Initialization
Frontier extraction
Compute the utility for every frontier point
Determine next-best viewpoint (maximum utility)
Go to next viewpoint and take a scan
Combine the new sensor reading into the global map
Mapping goal achieved?
N
Y
End

Figure 5.1 The flow chart of the information-based exploration algorithm.

5.4.2 Frontier extraction

The outer boundary and the hole boundary edges are identified after finding all the loops in the connectivity graph extracted from the triangular mesh map. The candidate frontiers in the map are defined as those triangles that have a specific distance from the boundary and have never acted as a viewpoint previously. To reduce the number of candidate locations on the frontiers, the distance between two candidate frontiers must satisfy the minimum distance requirement.

5.4.3 Utility function

The goal of this work was to maximize terrain information gain with the least energy consumption; therefore, the utility function will be chosen to reach this goal in each step. In this research, we consider new terrain information gain and the energy cost. The utility is constructed simply using a linear combination of new terrain information and the energy cost. The utility function can be described by the formula

$$\text{Utility} = \alpha_i \times \text{IG} - (1 - \alpha_i) \times k \times \text{Cost} + \omega_n \tag{5.1}$$

where

α_i = information gain weight (dimensionless decimal fraction),

IG = estimated new terrain information gain (m^2),

Cost = estimated energy consumption ($N \cdot m$),

k = information gain coefficient ($m^2/(N \cdot m)$), and

ω_n = utility offset (m^2).

5.4.4 Energy cost

Previous work focused on path distance as the only source of the cost for the exploration. However, travel distance alone is unsuitable to represent travel cost in the rough terrain of an outdoor unstructured environment. Energy consumption is very important for some exploration tasks such as planetary exploration and agricultural applications. In this work, the energy cost function not only considers the travel distance, but also includes the energy required to change elevation and the rolling resistance of the terrain during exploration. The total energy requirement for the vehicle to reach a goal location from a starting location will be predicted by the integration of energy (E) in the piecewise path. The total energy is always set as a positive value.

Assuming that soil hardness of the field is known and uniform, the energy requirement can by calculated by the following formula (Section 4.5.1):

$$E_{total} = W * \sum_{i=1}^{n} \left[\Delta z_i + d_{hi} * \mu_i \right] \tag{5.2}$$

where

E_{total} = the energy requirement (N·m),

Δz_i = slope height of the i_{th} segment of a piecewise path (m),

d_{hi} = horizontal distance of the i_{th} segment of a piecewise path (m),

$\mu_i = \frac{d_{hi}}{l_i} * B + 0.04$, rolling resistance coefficient (dimensionless),

n = the number of the segments of the path,

$B = \frac{0.3W}{CI * b * d}$, constant (dimensionless),

b = the tire width (cm),

d = the tire diameter (cm), and

W = the weight of the vehicle (N).

The energy consumption is proportional to the elevation change Δz_i. When driving uphill, Δz_i is positive and the slope resistance is in the opposite direction of the vehicle's tractive force. When driving downhill, Δz_i is negative, and the slope resistance is in the same direction as the vehicle's tractive force. This manuscript assumed the vehicle travels at a constant speed; therefore, brake energy will be required when the negative elevation change, Δz_i, is excessive.

The energy consumption is also proportional to the horizontal trip distance of the vehicle. The contribution of travel distance to energy consumption will vary considerably in relation to the rolling resistance coefficient, μ_i. A tire's rolling resistance coefficient depends on the soil hardness, terrain slope, and wheel parameters.

The vehicle dynamics and the safety factor are also of concern in this work. In this manuscript, the maximum climbing slope for the robot was set as 30º and the maximum downhill slope was 35º. When the uphill slope is greater than 30º or downhill slope is greater than 35º, the slope factor will be set as infinity because of the vehicle limitation. As the result of safety concerns, the robot can not pass any triangle whose tilt angle is more than 30º.

The soil hardness might be highly variable in one field considering soil type and moisture content can cause significant changes in CI values. When a soil strength map is available, a variable CI value can be used to calculate the rolling resistance.

5.4.5 Information gain

Two different methods of estimating the new terrain information gain have been developed to compare their performance in the exploration task. To address the case of exploration without a coarse map *a priori*, the overlap of the sensor's 2D footprint with the map was used to find how much new terrain area might be found in the next step. A 3D visibility analysis method based on frustum culling and ray casting is proposed to address exploration when a low-resolution map is available in advance.

5.4.5.1 Information gain estimation for an unknown environment

The Sutherland-Hodgman algorithm is a traditional polygon clipping algorithm (Hearn 1994). It uses a divide-and-conquer strategy to attack the problem. First, it identifies the intersection between the triangles within viewing frustum and triangles within the outer boundary. Any area within the viewing frustum that does not coincide with triangles within the boundary is neglected. The resulting polygon will include triangles about which information is known and those which are to be explored shown in Fig. 5.2. The new terrain information gained can be estimated by calculating unknown area of unexplored terrain within the polygon.

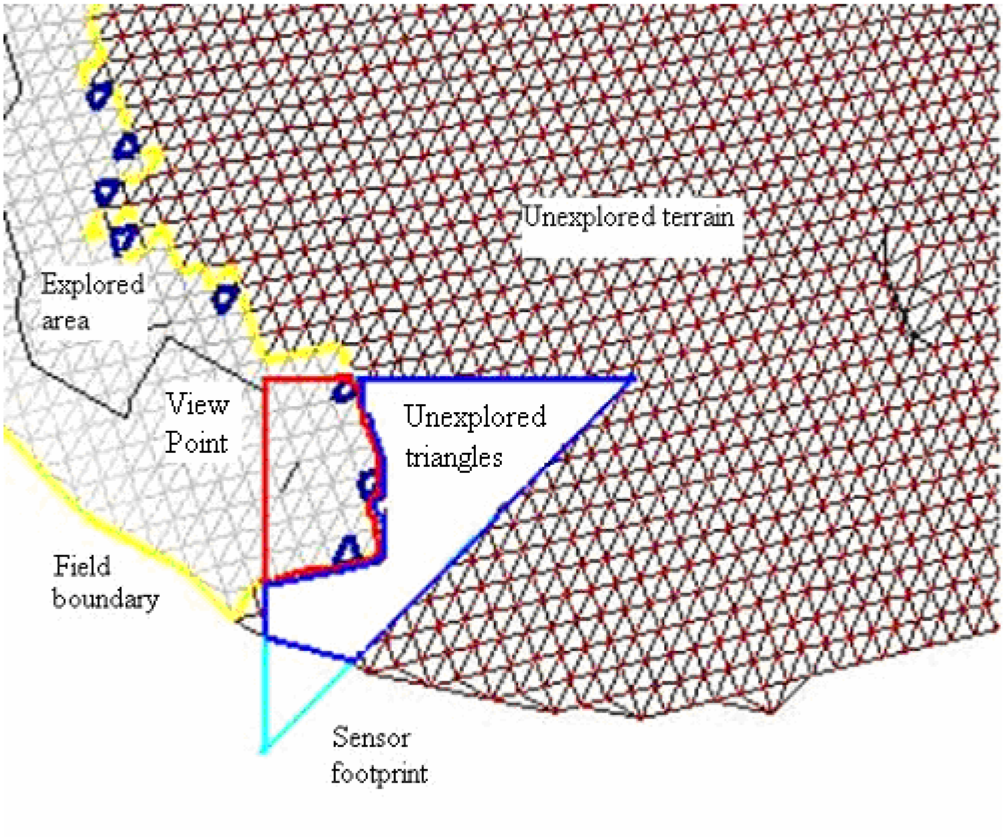


Figure 5.2 The sensor's footprint used to estimate the information gain.

5.4.5.2 The new terrain information gain estimation based on the visibility analysis

With a partially known map and 3D camera model, visibility analysis can be used to estimate which triangle is visible from every candidate viewpoint of the image sensor. The information gain can be calculated by applying frustum culling and a ray tracing algorithm to the known part of the environment. These visibility analysis algorithms are described in detail in Chapter 3.4.

The first step for visibility analysis is frustum culling. The image sensor's capacity is constrained by the viewing frustum which defines the visibility of every triangle in the terrain for each viewpoint, and triangles inside the viewing frustum are visible to the user. The viewing frustum is dynamically generated in accordance with the sensor's motion. The second step for visibility analysis is the ray casting algorithm. The algorithm starts by shooting a ray from the viewpoint of the camera and through the scene. Then, every triangle inside the viewing frustum is tested to see if the given ray intersects any of the elements. From the point of intersection, the triangle nearest to the viewpoint is visible through this ray. In this way, the visible triangles can be identified as those which intersect contiguous rays with the shortest distance from the viewpoint.

The information gain of a frontier point is calculated by the total area of the unvisited triangle that could be seen from this frontier point. Figure 5.3 shows that the coarse map used to estimate the information gain.

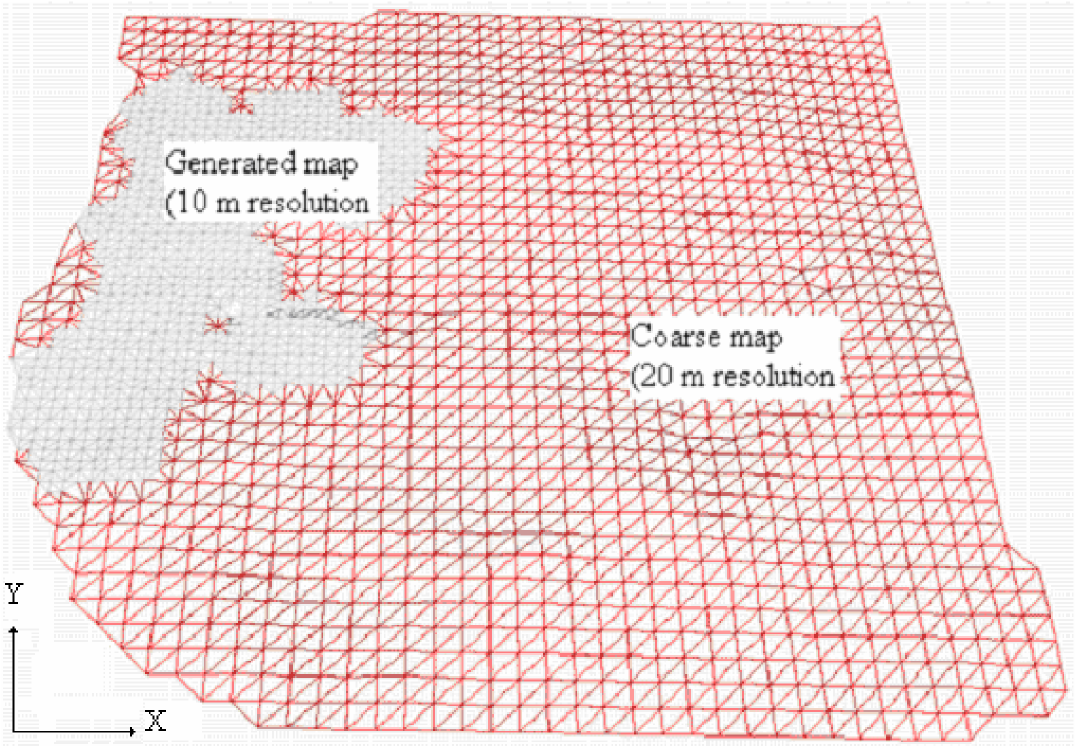


Figure 5.3 The coarse map used to estimate the information gain.

5.4.6 Simulation setup

A four-wheel drive robot (mass: 16 kg; length: 50 cm; width: 49 cm; height: 26 cm) with four identical wheels (wheel diameter: 25.2 cm; wheel width: 7.5 cm) was used in the simulation. The simulated robot was placed in 5 unknown testing fields as shown in Fig. 4.2. A uniform CI of 75 N/cm^2 is assumed for all the five silt fields (Goering et al., 2003). A 3D laser sensor model with a 90° field of view, 50 meter depth of field, and 1:1 aspect ratio was used as the vision system. It was assumed that the robot traveled at a steady speed of 3 m/s. The robot started at points A, B, C, and D for each fields shown in Table 4.1. The coarse map for agricultural filed 1 is shown in Fig. 5.4. Two different methods of estimating information gain were tested on these fields and start locations to compare the performance of the algorithms:

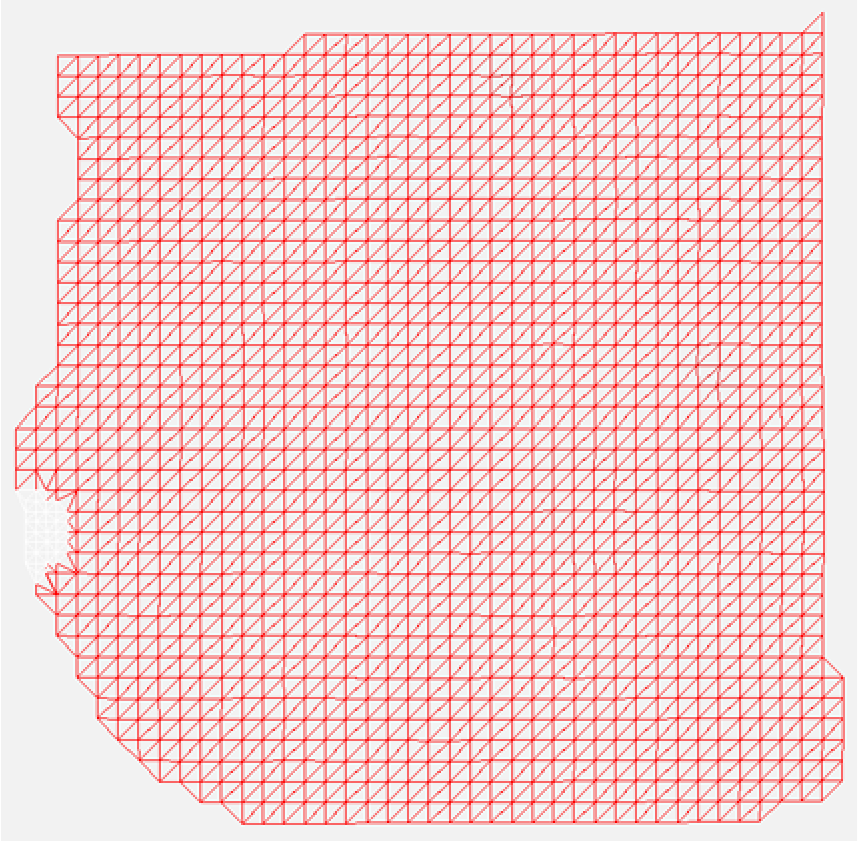


Figure 5.4 The 20 meter resolution coarse map for agricultural field 1.

(1) Information gain method 1—the new terrain information gain was estimated using polygon clipping. In this case, the scouting robot started with a completely blank map. The resolution of final maps in this simulation is 10 meters.

(2) Information gain method 2—it utilized a partially known map to estimate the information gain using frustum culling and ray casting algorithms. In this case, the exploring robot started with a 20 meter resolution coarse map rather than a blank map. The resolution of final maps in this simulation is also 10 meters.

Three different information gain weights for the utility function in Eq. 5.1 were chosen to test on each field and location. The weights are the α_i multipliers that are used to denote the relative importance of the information gain when computing the total expected utility for a frontier. The weights used for the three runs are found in Table 5.1.

Table 5.1 Information gain weights α_i for three runs.

Utility function	Run 1	Run2	Run 3
Exploration by Information gain method 1	0.0	0.5	1.0
Exploration by Information gain method 2	0.0	0.5	1.0

By combining three different information gain weights for two utility functions, five exploration strategies (α_i =0 results in identical functions) were got and listed below:

(1) Minimum energy consumption (α_i =0),

(2) Consider both energy requirement and information gain 1 (method 1, α_i =0.5),

(3) Maximum information gain 1 (method 1, α_i =1),

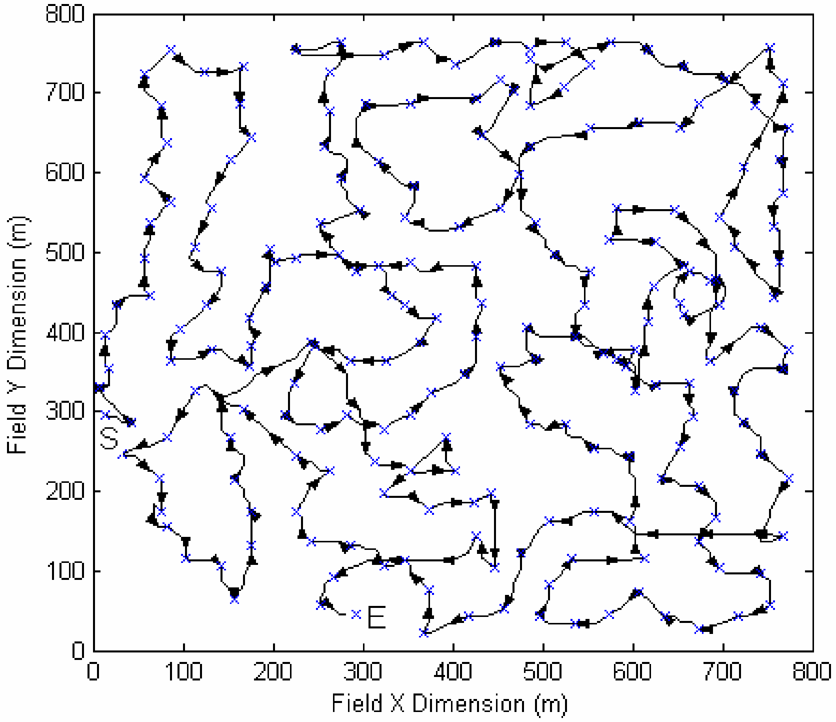
(4) Consider both energy requirement and information gain 2 (method 2, α_i =0.5), and

(5) Maximum information gain 2 (method 2, α_i =1).

5.5 Results and discussions

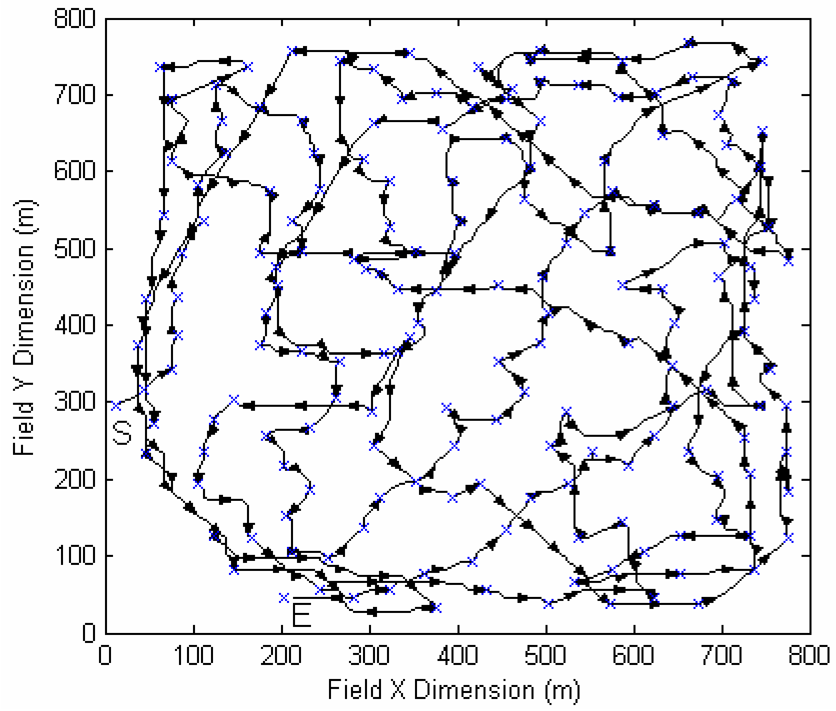
The complete results from the combination of all five fields and four starting positions for all methods can be found in Appendix C and D. Sample results for agricultural field 1, as shown in Fig. 3.1, from starting point A (20m, 300 m) using the five methods can be found in Figure 5.5 through Figure 5.6.

The trajectory paths generated using the five methods are given in Fig. 5.5(a) through Fig. 5.5(e), respectively. The black lines represent the path and the arrows show the vehicle’s travel direction, while the cross marks represent viewpoints where the robot stopped to take a scan. Figure 5.5 shows that strategies considering only information gain have more overlap in their trajectory plots. However, the other three strategies, which considered the minimum energy cost or integrated both the energy cost and information gain, have less overlap in the path generated in the simulation.

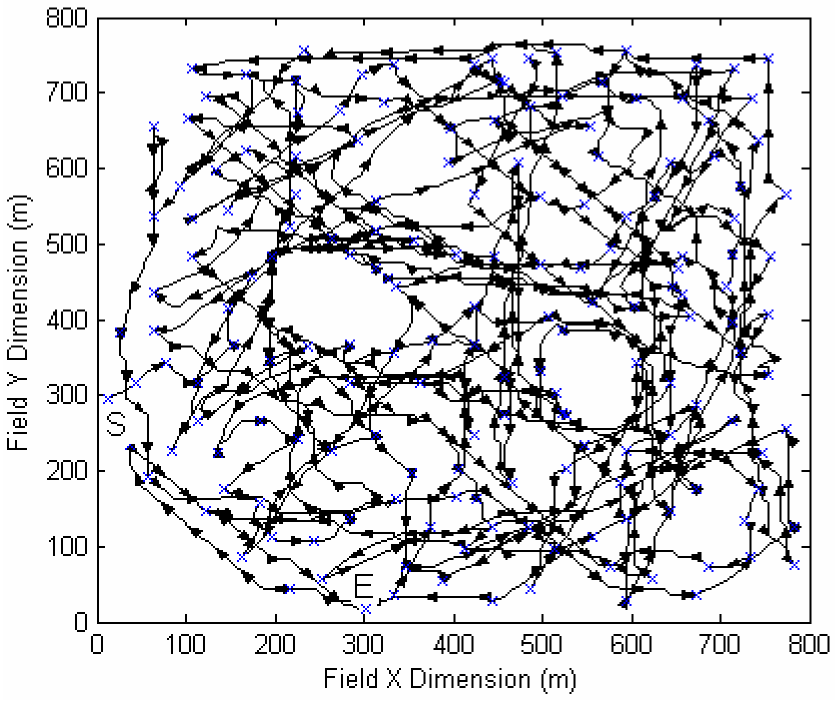


(a)

Figure 5.5 Trajectories generated in the simulation (arrows represent travel direction; cross marks represent viewpoints; started at S (starting point A) and ended at E): (a) minimum energy consumption.

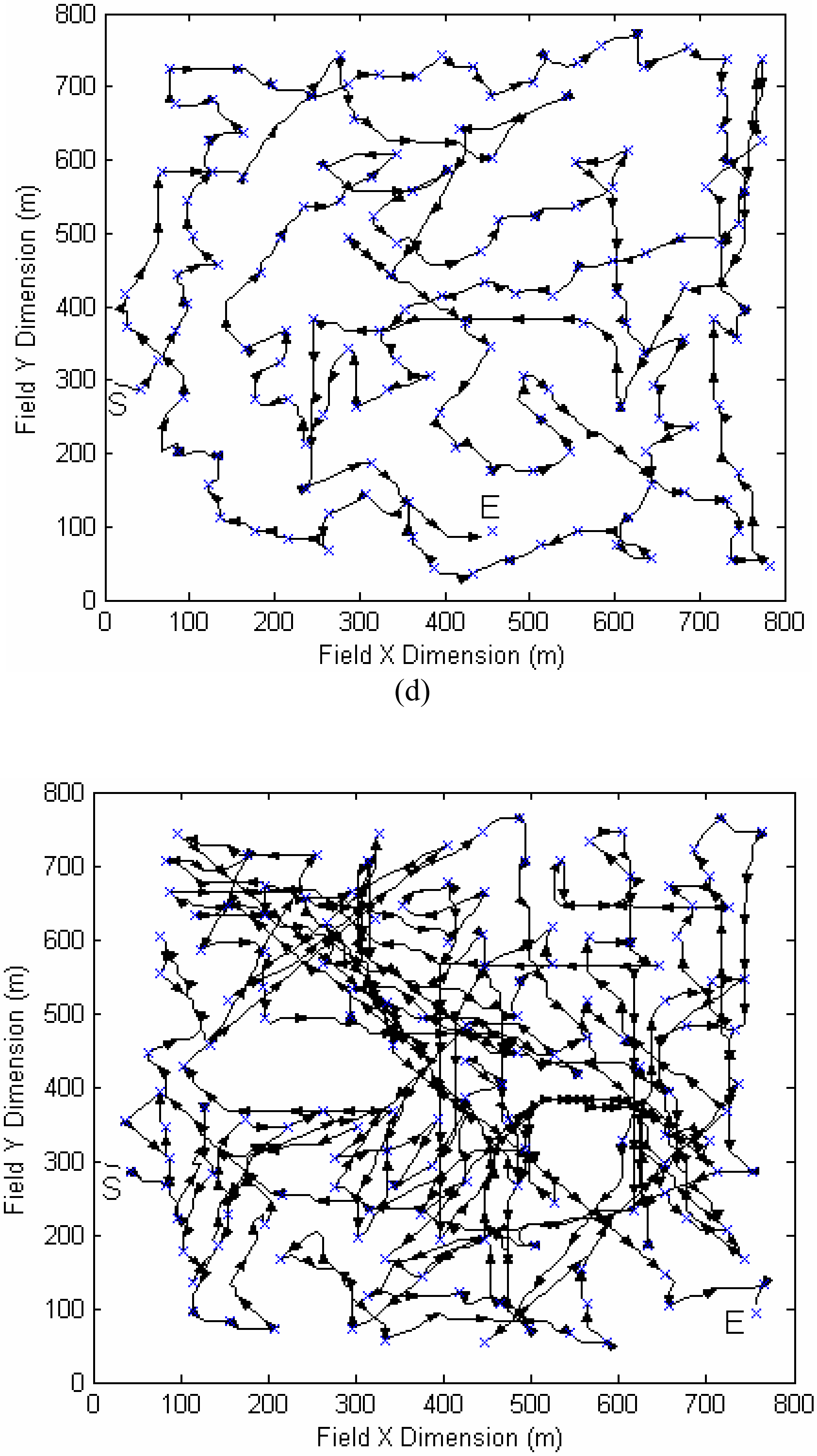


(b)



(c)

Figure 5.5 (continued). Trajectories generated in the simulation (arrows represent travel direction; cross marks represent viewpoints; started at S (starting point A) and ended at E): (b) consider both energy requirement and information gain 1, (c) maximum information gain 1.



(d)

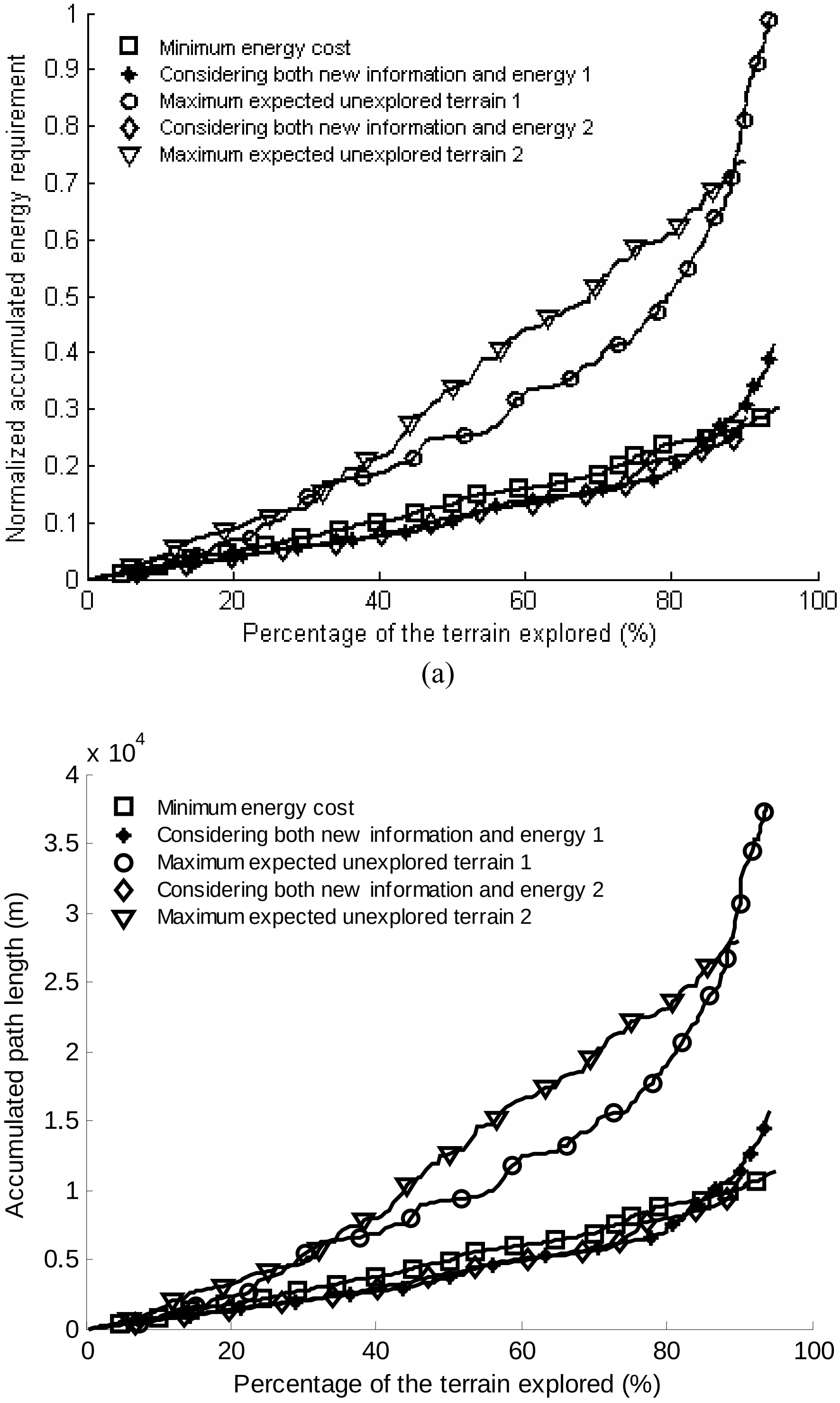
(e)

Figure 5.5 (continued). Trajectories generated in the simulation (arrows represent travel direction; cross marks represent viewpoints; started at S (starting point A) and ended at E): (d) consider both energy requirement and information gain 2, (e) maximum information gain 2.

Figures 5.6(a) – 5.6(e) show the relationship of the fraction of the environment mapped and the energy requirement, distance traveled, time requirement (including planning time and navigation time), and scan number by the robot exploration of agricultural field 1 starting from location A for the five exploration strategies. As shown in Figs. 5.6 (a) and 5.6(b), it is apparent that the energy requirement and path length of the methods that consider energy consumption are substantially smaller than those that consider only information gain, while the fraction of the terrain explored is greater in the former. Figure 5.6(e) shows that the scan number for the greedy method, which did not consider the information gain is drastically larger than those methods that consider the information gain.

For the three methods that considered the energy consumption and information gain, the strategies using both information gain and energy consumption resulted in more efficient energy usage and shorter path length during the beginning of the exploration tasks, while the minimum energy requirement method required a little larger energy and travel length in the earlier stage of the exploration task. The three methods resulted in almost the same path length and energy consumption for 90 percent of the exploration task for agricultural field 1 starting from location A (20 m, 300 m). After that, the efficiency of the exploration policy using information gain method 1 decreased greatly in terms of energy usage and path length.

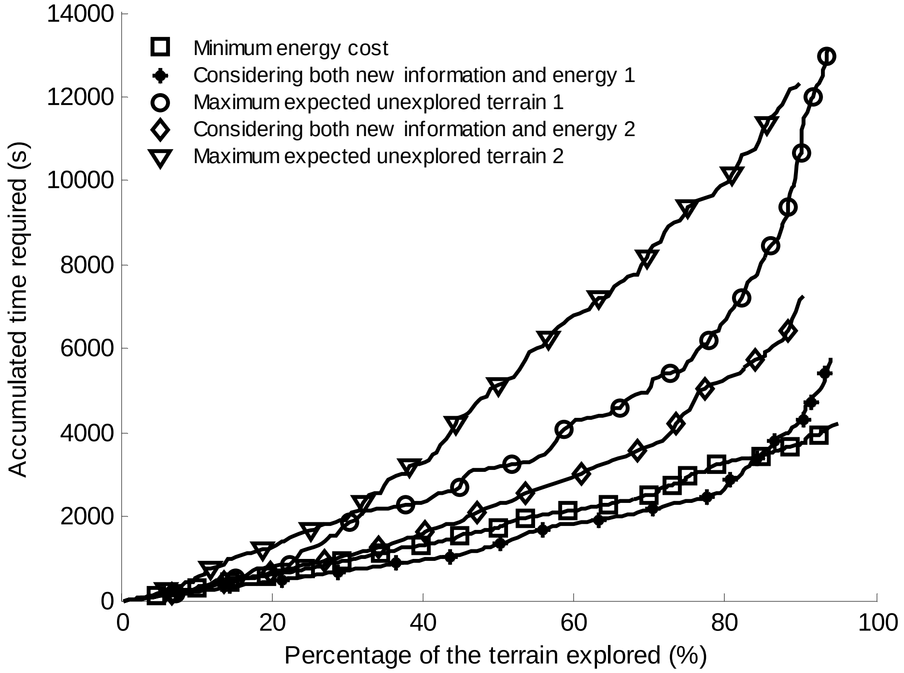
The greedy method using frustum culling and a ray casting algorithm required much more time than the method using the sensor footprint to estimate new information gain, because the ray casting algorithm needs a relatively large amount of time to do calculations.



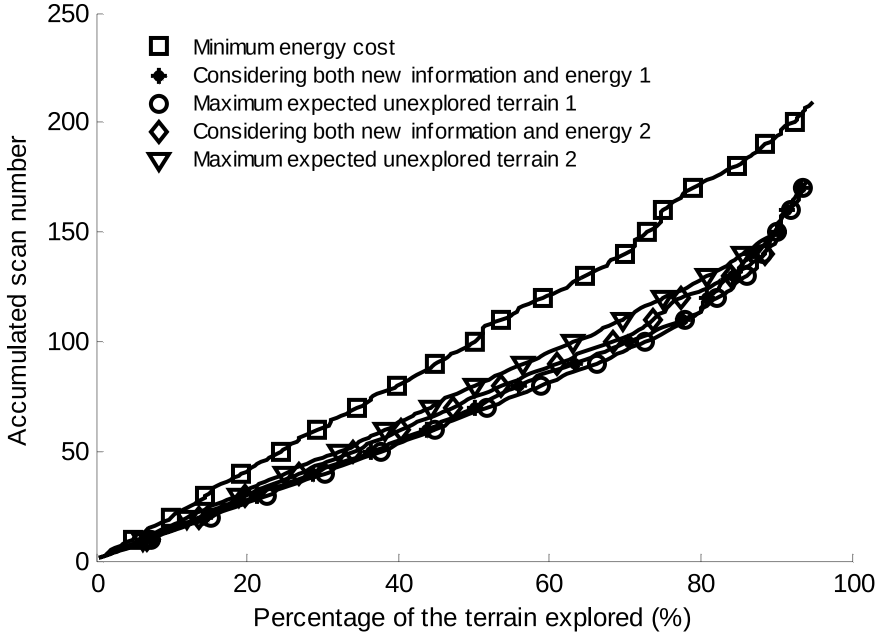
(a)

(b)

Figure 5.6 Result of the autonomous mapping of agricultural field 1 (start point A): (a) normalized accumulated energy requirement (the nominal energy is divided by the maximum energy) of the exploration as a function of fraction of explored terrain; (b) path length traveled by the robot as a function of fraction of explored terrain.



(c)



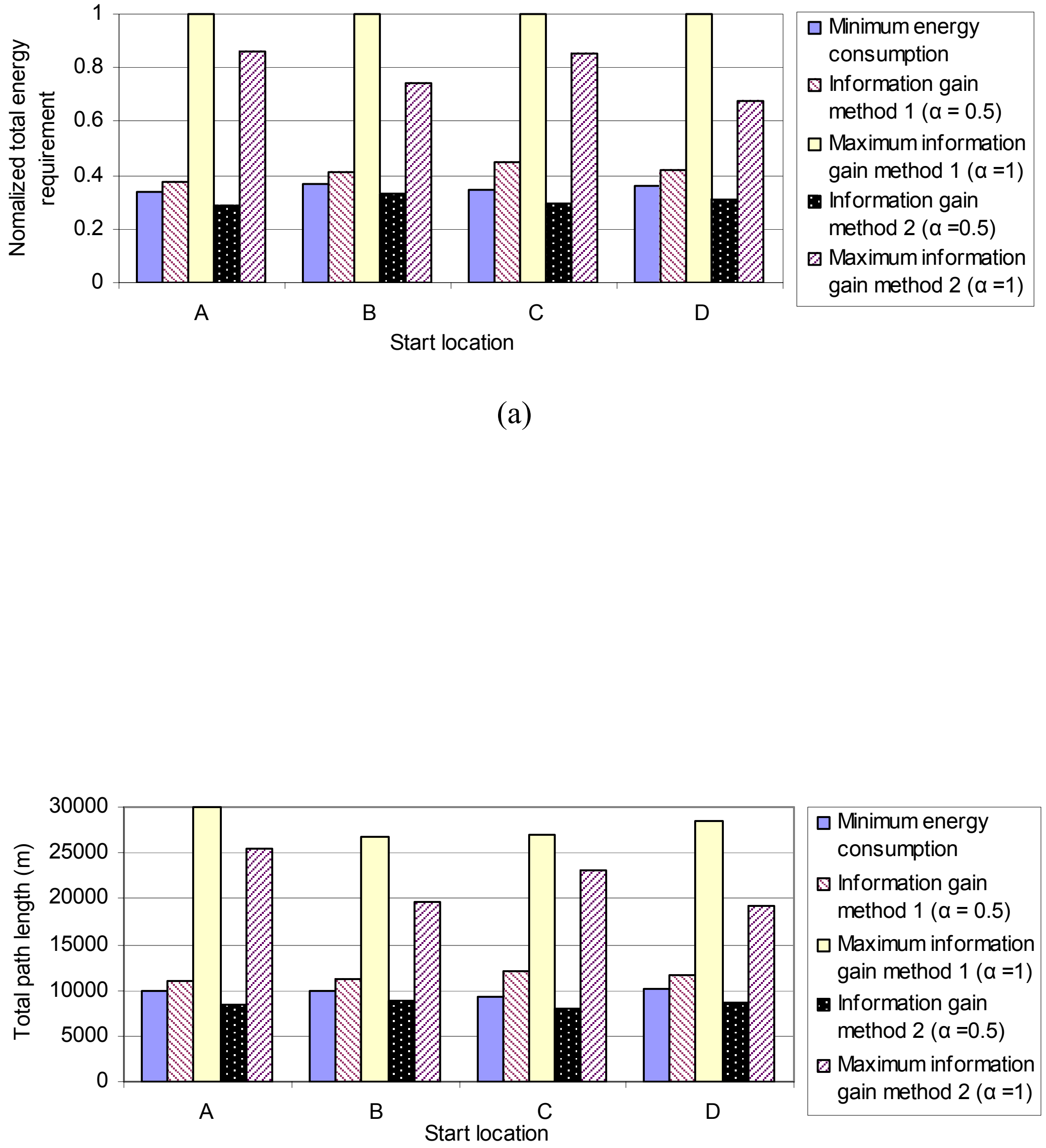
(d)

Figure 5.6 (continued) Result of the autonomous mapping of agricultural field 1 (start point A): (c) accumulated time required of the exploration as a function of fraction of explored terrain; (d) the scan number as a function of fraction of explored terrain.

Figures 5.7(a) to 5.7(d) demonstrated the performance of the five exploration strategies on the energy requirement, distance traveled, time requirement, and scan number, respectively, for 90% of the exploration of agricultural field 1 starting from 4 different locations. Figures 5.8(a) to 5.8(d) showed the performance of the five exploration strategies for 90% of the exploration of 5 testing fields.

The information gain method 2, considering both information gain ($\alpha = 0.5$) and energy consumption, performed the best among the 5 methods in terms of total energy requirement, path length for 90% of the exploration of all five fields and 4 starting locations (for agricultural field 1). Because there is a coarse map available for the information gain method 2, the robot can plan a better next best viewpoint by maximizing both the information gain and minimizing the travel cost in each step; therefore, the information gain method 2 performed best in terms of total energy consumption. The minimum energy function performed the second in the same tasks for agricultural field 1 (4 starting locations), agricultural field 2, and the hole environment, while the information gain method 1 ($\alpha = 0.5$) performed second best in the other tasks for the mountain and the slope environments. These results demonstrate a great advantage of coverage planning with the ray tracing using a coarse map over other situations, and the advantages of the ray casting algorithm over the simple sensor footprint method to estimate the new terrain information.

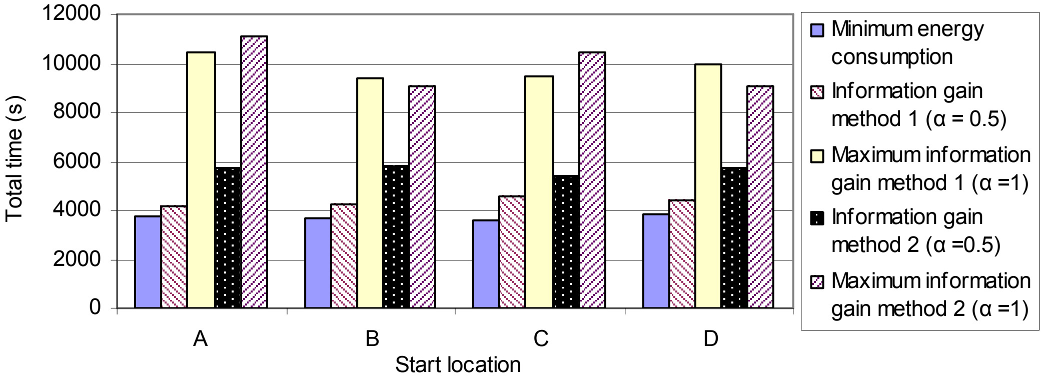
In terms of time requirement, the minimum energy requirement method greatly outperformed other methods for agricultural field 1 (4 starting locations) and agricultural 2, although it required more energy and path length to complete the same tasks. This is because that information gain method 2 ($\alpha = 0.5$) required much more time on planning the path.



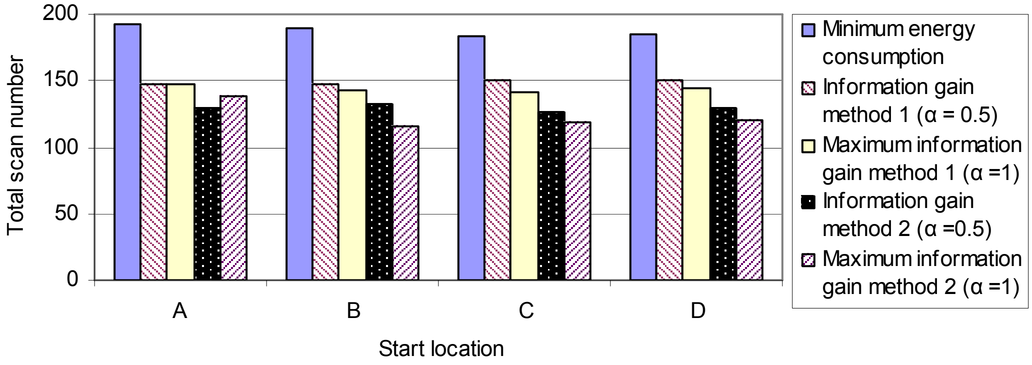
(a)

(b)

Figure 5.7 Performance for 90% of the exploration of the automatic mapping of agricultural field 1 for different starting points A, B, C, D as defined in Table 4.1: (a) normalized total energy requirement, (b) total path length.

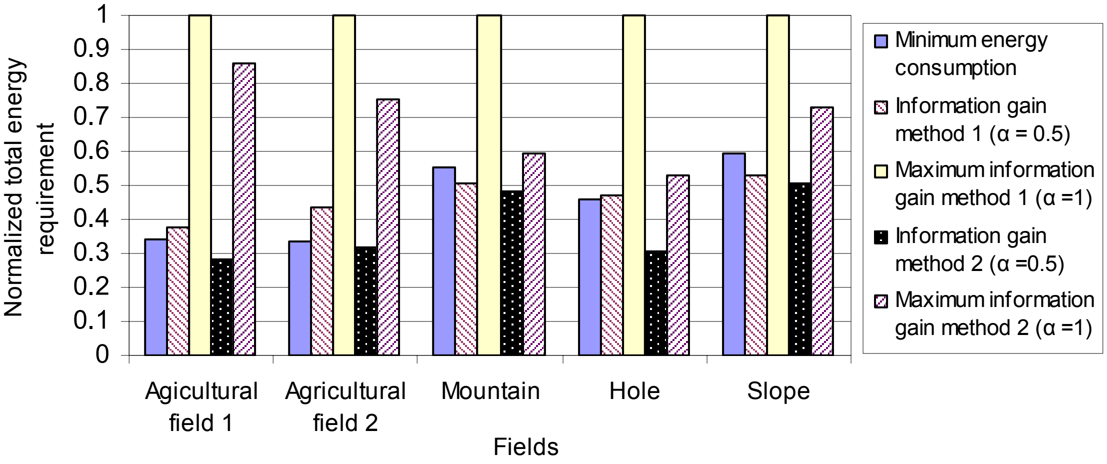


(c)

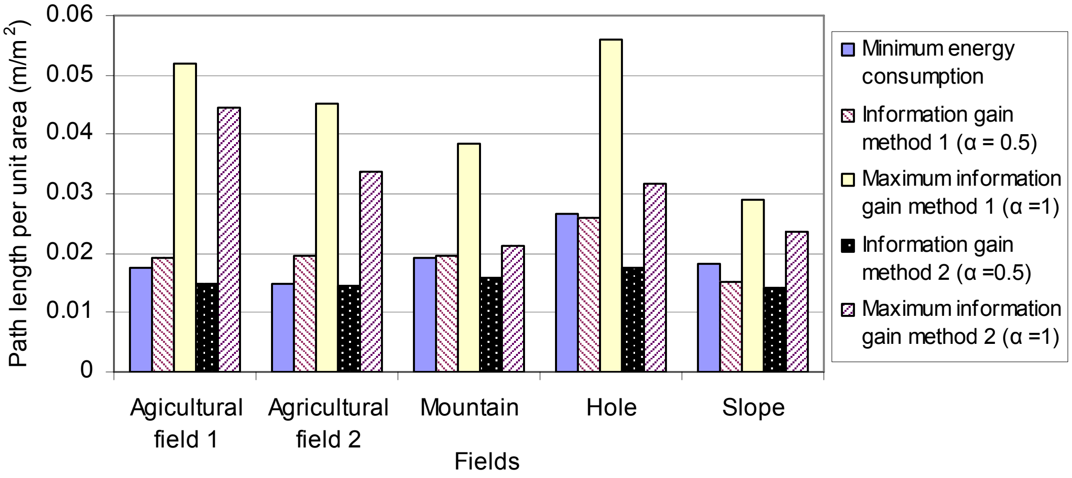


(d)

Figure 5.7 (continued) Performance for 90% of the exploration of the automatic mapping of agricultural field 1 for different starting points A, B, C, D as defined in Table 4.1: (c) total time, (d) total scan number.

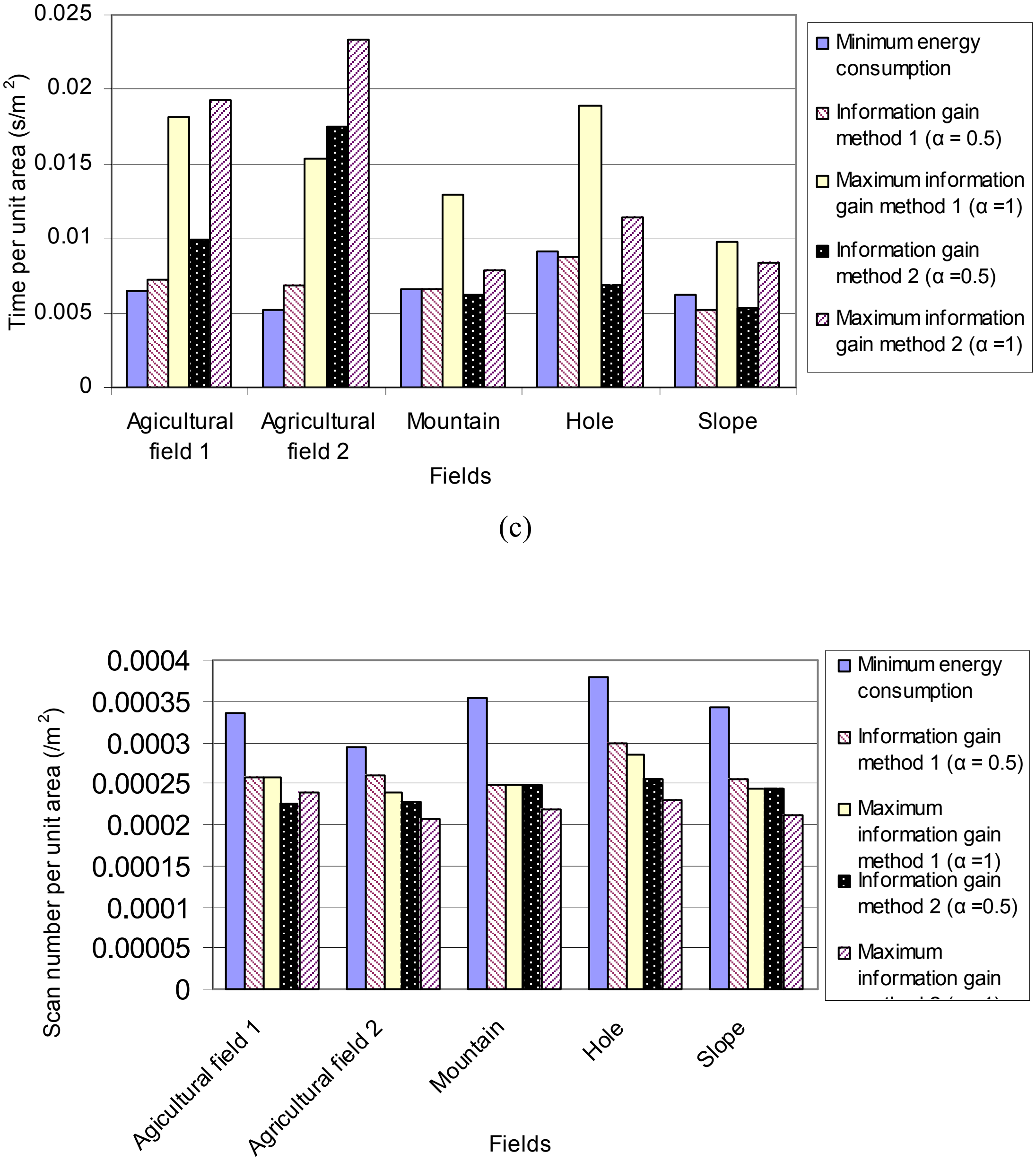


(a)



(b)

Figure 5.8 Performance for 90% of the exploration of the automatic mapping of testing fields when starting at point A as defined in Table 4.1: (a) normalized total relative energy requirement, (b) path length per unit area.



(c)

(d)

Figure 5.8 (continued) Performance for 90% of the exploration of the automatic mapping of testing fields when starting at point A: (c) time per unit area, (d) scan number per unit area.

5.6 Conclusions

In this chapter, information-based exploration algorithms were presented to address the problem of the next best viewpoint in modeling large rough unstructured environments. A triangular mesh map was used to represent a 3D rough environment. Two methods of estimating new terrain information gain were developed. The first method of estimating information gain involved polygon clipping. A terrain visibility analysis based on a viewing frustum model and ray casting algorithm was proposed in the second method to address the information gain estimation for exploration with an *a priori* coarse map. Simulation results in two typical western Canadian agricultural fields and three virtual fields were presented to demonstrate the algorithm.

The exploration strategy, which incorporated the energy consumption and the information gain with a ray tracing algorithm using a coarse map, had an advantage over other policies in terms of the total energy consumption and the path length by at least 4%.

The information gain 2, with a ray tracing algorithm using a coarse map, performed much better over the simple sensor footprint method to estimate the new terrain information; The information gain 2 method required 25%, 26%, 4%, 34% and 4% less energy than information gain 1 method for agricultural field 1, agricultural field 2, mountain, hole, and slope environment, respectively when starting at location A. However, the greedy method using the frustum culling and the ray casting algorithms required more planning time than the method that used the sensor footprint to estimate new information gain.

Path length and energy requirement of the methods that considered energy consumption were substantially smaller than those for the methods that consider only information gain, while the fraction of the terrain explored was greater in the former; The maximum information gain methods required between 8% and 251% more energy than the minimum energy and other two methods, considering both energy requirement and the information gain.

The scan number for the greedy method that did not consider information gain was at least 14% larger than the methods that considered information gain. These results show the effectiveness of the algorithm considering both the energy consumption and travel cost.

5.7 References

Bourgault, F., A.A. Makarenko, S.B. Williams, B. Grocholsky and H.F. Durrant-Whyte. 2002. Information based adaptive robotic exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* 1. 540- 545. Lausanne, Switzerland, Sep 30–Oct 5, 2002.

Cormen H.T., C.E. Leiserson, R.L. Rivest and C. Stein. 2001. *Introduction to algorithms*, second edition. MIT Press and McGraw-Hill. Cambridge/NewYork.

Feder,H.J.S., J.J. Leonard and C.M. Smith. 1999. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research* 18(7):650-668.

Goering, C.E., M.L. Stone, D.W. Smith and P.K. Turnquist. 2003. Off-Road Vehicle Engineering Principles. St. Joseph, MI: ASABE. 474 pages.

Gonzalez-Banos, H.H. and J.C. Latombe. 2002. Navigation strategies for exploring indoor environments. In *International Journal of Robotics Research* 21(10-11): 829-848.

Hearn, D. 1994. Computer graphics, second edition. Prentice-Hall. EngleWood Cliffs, New Jersey.

Kitware Inc. 2005. The Visualization Toolkit. *http://www.vtk.org.* Accessed June 26, 2005. New York.

Moorehead, S., R. Simmons and W.L. Whittaker. 2001. Autonomous exploration using multiple sources of information. In *Proceedings of the IEEE International Conference on Robotics and Automation* 3: 3098- 3103. Seoul, Korea, May 21-26.

Oh, J., Y. Choi, J. Park and Y. F. Zheng. 2004. Complete coverage navigation using of cleaning robots using triangular cell based map. *IEEE Transactions on Industrial Electronics* 51(3): 718-726.

Prestes, E., P.M. Engel,M. Trevisan and M.A.P. Idiart. 2002. Exploration method using harmonic functions: *Robotics and Autonomous Systems* 40(1): 25-42.

Rocha, R., J. Dias and A. Carvalho. 2005. Cooperative multi-robot systems: a study of vision-based 3-D mapping using information theory. *Robotics and Autonomous Systems* 53(3-4): 282-311.

Schroeder, W., K. Martin and B. Lorensen. 1996. The Visualization Toolkit: an object-oriented approach to 3D graphics. Prentice Hall PTR. Upper Saddle River, NJ.

Simmons, R., D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun and H. Younes. 2000. Coordination for multi-robot exploration and mapping. In *Proceedings National Conference on Artificial Intelligence.* Austin, TX.

Sujan,V. A. and S. Dubowsky. 2005. Efficient information-based visual robotic mapping in unstructured environments. *International Journal of Robotics Research* 24 (4): 275-293.

Sujan, V.A., M.A. Meggiolaro and F.A.W. Belo. 2006. Information based indoor environment robotic exploration and modeling using 2-D images and graphs**.** *Autonomous Robots* 21(1): 15-28.

Surmann, H., A. Nüchter and J. Hertzberg. 2003. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems* 45: 181–198.

Taylor, C. and D. Kriegman. 1998. Vision-based motion planning and exploration algorithms for mobile robots. *IEEE Transactions on Robotics and Automation* 14(3):147–427.

Thrun, S., D. Fox and W. Burgard. 1998. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning* 31:29-53.

Tovar, B., L. Munoz-Gomez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy and S. Hutchinson. 2006. Planning exploration strategies for simultaneous localization and mapping. *Robotics and Autonomous Systems* 54 (4): 314-331.

Yamauchi, B. 1997. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation:* 146-151. Monterey, CA. July 10-11.

6. SENSOR-BASED SCOUTING ALGORITHM FOR AGRICULTURAL ROBOTS

6.1 Significance

This chapter relates to Objective 4 of the thesis (Chapter 1.2). The review of coverage algorithms in Chapter 2 shows that there has not been previous research on the statistical coverage task, in which the robot visits all predefined sampling points. In statistical coverage, the distance between two sampling points is beyond the capability of a vision sensor, while in traditional coverage, algorithms assume that the robot can always view the neighboring sampling point from its current location. The aim of this chapter was to develop scouting algorithms to guide the vehicle to reach predefined sampling points for field sampling tasks.

Previous chapters discuss the exploration policies required to build a topographic map of a rough agricultural environment using a robot equipped with a laser sensor. The exploration policies developed in previous chapters can potentially be integrated into the path planning algorithm to find an efficient path between two sampling points. The scouting algorithm is composed of two parts: the coverage algorithm, which identifies a reasonable coverage path to reach all the sampling points, and the path planning algorithm, which determines an optimal path between two adjacent sampling points.

6.2 Introduction

Traditional agricultural production is largely dependent on the application of pesticides and fertilizers. Canadian farmers alone consumed over 5 million tones of fertilizers in the year 2001/2002, while Saskatchewan accounted for 29% of total consumption (Korol, 2006). Economic incentives and environmental pressure from the public have prompted agricultural producers to search for more efficient ways to manage chemical application. Site-specific crop management including spatially-selective application of fertilizer and other chemicals has the potential to increase profits and reduce the threat to the environment. Compared with the uniform method, site-specific application of herbicide allowed herbicide saving up to 30-40% (Baio and Balastreire, 2002).

High-resolution weed and soil maps play a critical role in variable-rate application of chemicals. Field variable maps such as a soil map or a weed map can be generated by air-based remote sensing methods (Goel et al., 2002; Bajwa and Tian, 2001) or by ground-based measurement methods (Fontaine and Crowe, 2006; Adamchuk et al., 1999). Air-based remote sensing techniques provide an efficient way to generate field variable maps over large scale fields. However, the accuracy of maps generated from remotely sensed images can not attain the requirement of some specific agricultural tasks (Bishop and McBratney, 2002). The availability of the Global Positioning System (GPS) has prompted researchers to investigate the vehicle-based method, in which a GPS and other sensors are mounted on an agricultural vehicle to measure the field variability in weeds or soil while the vehicle moves around the working field (Bishop and McBratney, 2002; Westphalen et al., 2004; Saraswat et al., 2003; Schmidt et al., 2003; Adamchuk et al., 1999). However, all previous work required a driver to drive the vehicle through the fields along a pre-defined path such as a crop row or straight line direction, and it is labor intensive. Human labor, which is rather expensive in Canada, will increase the inputs to the spatially selective herbicide application. This will adversely affect the agricultural producer’s motivation to implement this new technology.

A scouting robot equipped with an automatic guidance system and a sensor that can record geo-referenced data about the field would have the potential to alleviate the labor shortage problem. A few groups used robotic platforms to build a weed map or conduct mechanical weeding at the same time (Tillett and Hague, 1999; Nielsen et al., 2002; Åstrand and Baerveldt, 2002; Fontaine and Crowe, 2006). All of the above systems registered and mapped weed distribution between crop rows using computer vision to navigate scouting robots. The camera was located at the front of the vehicle looking forward in order to provide images of crop rows. The images were analyzed to identify crop rows to provide guidance information. In general, most previous researchers focused on finding a path using landmarks such as crop rows (Tillett and Hague, 1999; Nielsen et al., 2002; Åstrand and Baerveldt, 2002; Fontaine and Crowe, 2006) or edges between cut and uncut crop (Ollis and Stentz, 1996) in an agricultural field. However, few have considered exploration algorithms that focus on obstacle avoidance and path planning for agricultural field scouting robots.

The objective of this chapter was to develop a scouting algorithm to guide a robot to get to each sample point while circumventing obstacles in a working field. A number of interesting research problems need to be studied in order to make the system functional. The research described in this chapter concentrates on the following problem areas:

(1) coverage algorithm: it is important for a scouting robot to plan an efficient path to cover the whole environment, and
(2) dynamic path planning in the unknown environment: the robot would need to be effective in determining the optimal path to its goal at no risk of collision with obstacles in its surrounding environment; this is important for weed scouting applications without apparent landmarks to follow.

A triangular mesh map was used to represent the rough agricultural field surface because of its ability of planning smoother paths. The map is incrementally built using laser sensor readings. This chapter describes a system that integrates coverage and path planning algorithms during weed scouting and soil sampling tasks in an unknown or partially known unstructured agricultural field. The coverage algorithm identifies a reasonable coverage path to traverse, while the path planning algorithm determines an optimal path between two adjacent sampling points.

6.3 Related Work

The scouting algorithm presented in this chapter operates by sweeping every sampling point with a sensor, which has been strongly associated with coverage planning in the literature. The robot must pass through all reachable points in the target environment. Because determining an optimal coverage path is an NP-complete (verifiable in nondeterministic polynomial time) problem and may very well prove to be intractable (Choset, 2001), much work in coverage path planning has been developed to get a sub-optimal path in applications such as floor cleaning, lawn mowing, demining, painting, and environment model building. These existing algorithms take the following basic approach to generating a coverage path (Huang, 2001; Hert et al., 1996; Choset and Pignon, 1997): the region to be covered is divided into subregions, a traveling-salesman algorithm is applied to generate a sequence of subregions to visit, and a coverage path is generated from this sequence that covers each subregion in turn. These algorithms all use a single line sweep in order to divide the coverage region into subregions, and these subregions are individually covered using a back and forth motion in rows perpendicular to the sweep direction.

Traditional coverage path planning algorithms generally use uniform grid maps where the value for each cell represents the probability of an obstacle in the 2D environment (Zelinsky et al., 1993). They assumed that the travel cost value is the same for every grid cell and the sensor has the capability to sense its neighbor cells. In contrast, for weed scouting or soil sampling tasks in agricultural fields, the distance between two cells is usually so great that the imaging sensor could not know the traversability of its neighboring cells. For example, collecting 15-20 cores for the surface sample and six to eight cores for subsurface samples per 20 acres usually will give reliable mean values for the sampled area (Ferguson et al., 1998). The ruggedness of the terrain also influences the exploration strategy employed by a robot because the cost of driving is not the same for all traversable areas. This requires that the obstacle avoidance and path planning problem be addressed while the robot is going from one sampling location to another neighboring sampling location.

6.4 Scouting Algorithm

6.4.1 Triangular mesh map

A regular triangular mesh map was used to represent the rough agricultural field surface because it can maintain a very rich representation of the environment and allows a smoother path, compared with the square grid map. Every triangle has three edge neighbors and nine vertex neighbors, so it has twelve cell neighbors in a non-boundary triangle cell. The triangular mesh map provides twelve moving direction choices for each location, enabling the generation of a much smoother path compared to the path generated using a traditional grid map.

The triangular mesh map is incrementally built using laser sensor readings based on Delaunay triangulation. The Visualization Toolkit (Kitware Inc., 2005), available freely on the Web, has been used to implement the triangulation in our simulation. Figure 3.1 is the triangular mesh map of a simulation environment created using Delaunay triangulation. The triangular mesh map is stored in the computer as a simple weighted graph, the vertex of which is used to represent every triangle and the edge represents the relative difficulty to traverse the adjacent triangle.

6.4.2 Laser sensor model

Most coverage and exploration algorithms only consider the ideal sensors. They assume that if the robot traverses a cell, then the whole cell is covered (Moorehead, 2001). Very few researchers have investigated the coverage algorithm using image sensors. To deal with the rough terrain visibility problem, in which one part of the terrain may occlude other parts, we utilize a 3D view model, which is commonly used in computer graphics rendering. Field of view defines the visibility of every triangle in the terrain for each viewpoint. Triangles lying inside the view frustum may be visible to the user and vice versa. The laser sensor model is described in detail in Chapter 3.4.

6.4.3 Sampling points extraction

The systematic grid sampling problem was presumed in this research. An assumption is that the boundary of the field has been clearly defined. For example, it can be defined as a rectangular area by 4 vertices or by a polygon as shown in Fig. 6.1. The sampling pattern was defined by the intervals between two sampling centers in the longitude and latitude direction. Fig. 6.1 shows the result—the interval equals 100 meters in x direction and 87 meters in y direction in agricultural field 1 as shown in Fig. 4.2 (a).

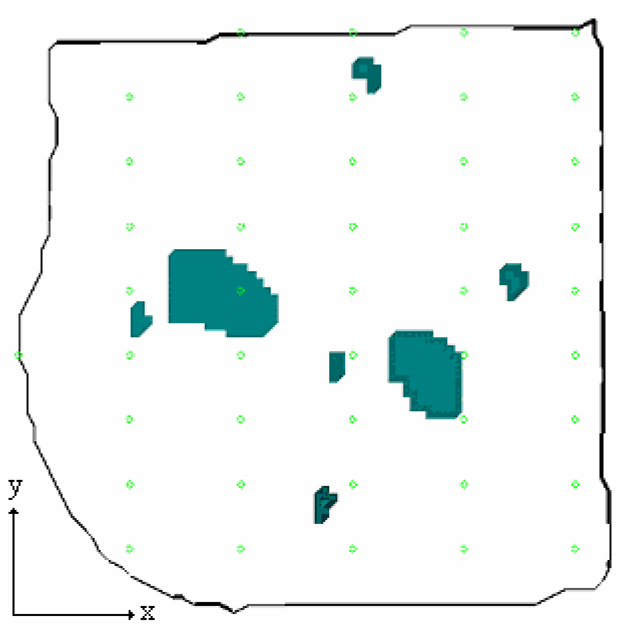


Figure 6.1 Sampling points extracted in a simulation environment.

6.4.4 Line sweeping + heuristic function method

Given sample points defined by the above method, the scouting algorithm is used to traverse every sample point in the field. To illustrate our algorithm, assume that the robot knows its precise location at anytime, whether through GPS or odometry. Another assumption in the simulation is that the environment is static. In other words, there are no moving obstacles in the fields. Finally the field boundary is precisely defined.

The approach divides the scouting problem into two parts: the coverage algorithm identifies a reasonable coverage path to traverse sampling points, while the path planning algorithm determines an optimal path between two adjacent sampling points. The main strategy for the coverage is the line sweeping method. The sampling points list is defined as those sample points in a sequence following the line sweeping direction. By traversing the sampling points (accessing points on the sampling points list), the covering task is completed. A path planner based on a heuristic function is used to find an optimal path between adjacent sampling points, circumventing obstacles between them. The core of the scouting algorithm is the path planning algorithm.

6.4.5 Dynamic path planning algorithm

The use of a triangular mesh to represent terrain allows us to use the path planning algorithm easily. In the known environment, a similar method to that used by Dupuis et al. (2004) was used. The triangular mesh map is stored in the computer as a simple weighted graph, the vertex of which is used to represent every triangle and the edge of which represents the relative difficulty to traverse the adjacent triangle. Once the graph is constructed, a path between the current rover location and a destination can be planned using Dijkstra's shortest path algorithm. In their approach, they only use three triangle edge neighbors to build the graph. In contrast, twelve adjacent neighbors, including three edge neighbors and nine vertex neighbors, were considered to build the graph. The method proposed in this manuscript allows a much smoother path to be generated compared to the former method.

Unfortunately in most cases, there are no such maps available, so the ability to deal with unknown terrain is very important. In an unknown environment, due to the limitation of the sensor's ability, the robot usually cannot see the next sampling point from the current sampling point. Therefore, the robot requires the intelligence to decide where to go in the next step and take a new scan. Combining the new sensor readings into a map, the robot eventually will find a path to the next sampling point. A next best viewpoint planning approach was developed to find an optimal path to reach the next sampling point by planning the next best viewpoint in each iterative step. In this application, a heuristic function is used to find the next best viewpoint, which is decided not only using the traveling cost from the current robot location to the next potential sensing location, but also using the estimated cost from the next sensing location to the target sampling point. In this way, the exploration strategy is integrated into path planning.

To begin the task, the robot extracts the frontiers (Yamauchi, 1997) from the triangles near the boundary of the current map, and then it constructs a heuristic function consisting of the cost for it to travel to various frontiers and the estimated cost from the frontiers to the target sampling point. The robot visits the frontier with the least traveling cost and takes a scan with its visual sensor. The map is rebuilt by combining the new sensor reading. The robot plans the next best viewpoint using the new map until it reaches the goal or the goal is unreachable. Figure 6.2 shows the flow diagram of this algorithm.

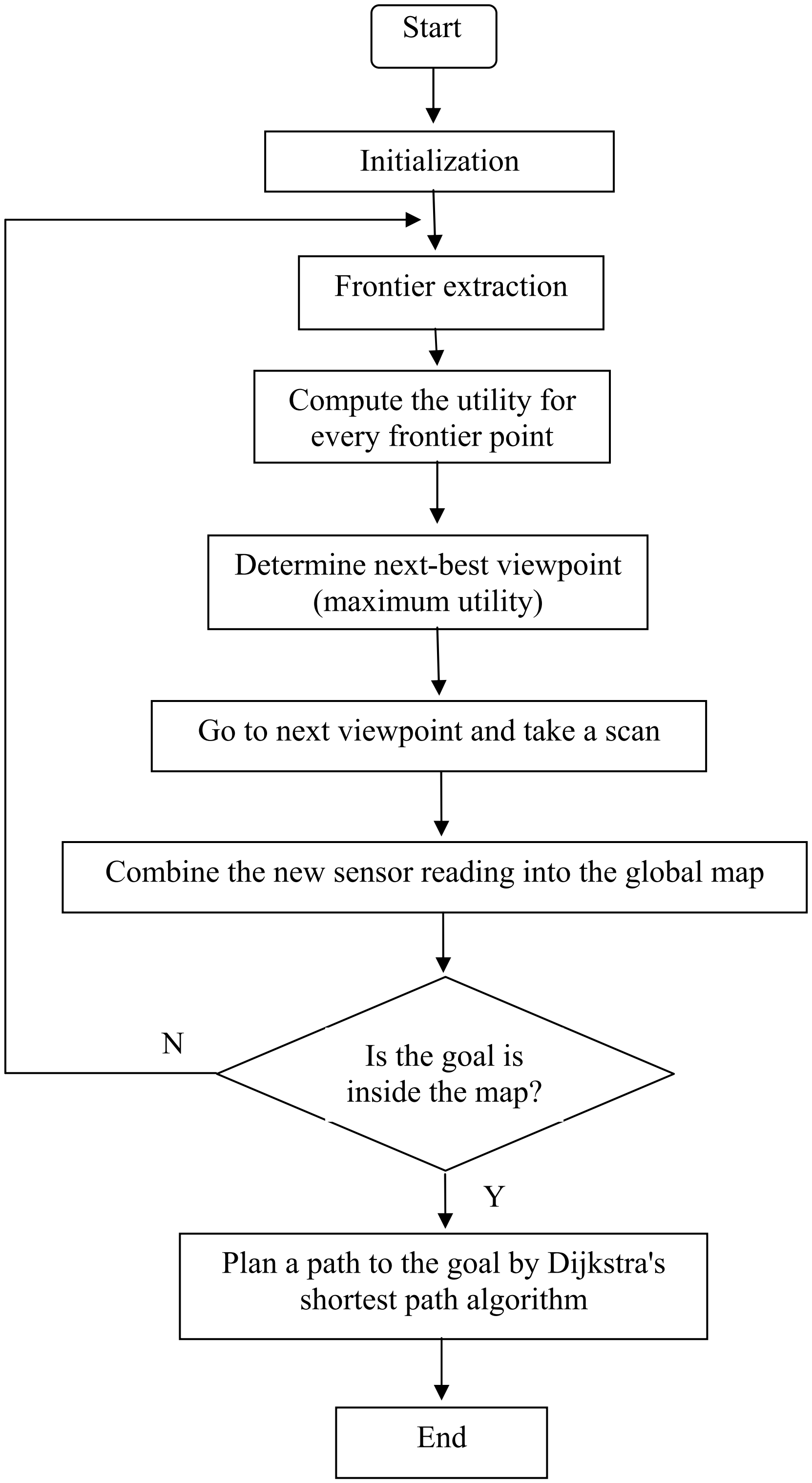

Start
Initialization
Frontier extraction
Compute the utility for every frontier point
Determine next-best viewpoint (maximum utility)
Go to next viewpoint and take a scan
Combine the new sensor reading into the global map
Is the goal is inside the map?
N
Y
Plan a path to the goal by Dijkstra's shortest path algorithm
End

Figure 6.2 Dynamic path planning algorithm flow diagram.

6.4.5.1 Energy cost function

Many exploration algorithms consider path distance as the source of the cost for the exploration. However, energy consumption is more important for agricultural tasks because of economic concerns. In this work, the energy requirement can be calculated by the formula (Chapter 4.5.1)

$$E = W * \sum_{i=1}^{n} [\Delta z_i + d_{hi} * \mu_i] \tag{6.1}$$

where

E = the energy requirement (N·m),

Δz_i = slope height of the i_{th} segment of a piecewise path (m),

d_{hi} = horizontal distance of the i_{th} segment of a piecewise path (m),

$\mu_i = \frac{d_{hi}}{l_i} * B + 0.04$, rolling resistance coefficient (dimensionless),

n = the number of the segments of the path, and

B = constant (dimensionless).

In this work, the energy cost function not only considers the traveling distance, but also includes the energy required to change elevation, and the rolling resistance of the terrain. A tire's rolling resistance coefficient depends on the soil hardness, terrain slope, and wheel parameters. The total energy requirement for the vehicle to reach a goal location from a start location will be predicted by the integration of energy (E) in the piecewise path.

The soil hardness might be highly variable in one field considering soil type and moisture content can cause significant changes in CI values. When a soil strength map is available, a variable CI value can be used to calculate the rolling resistance.

6.4.5.2 Next best viewpoint

The main idea of the next best viewpoint is to try to find the next best viewpoint with the least energy consumption among all the candidate frontiers. In this research, we consider both the energy cost of traveling from the current robot location to the next sensing point and the estimated cost of traveling from the next sensing point to the target sampling point. The heuristic function proposed in our research is

$$E_{total} = E_1 + E_2 , \qquad (6.2)$$

where

E_1 = the energy cost to move from the current robot location to the frontier, and

E_2 = the estimated energy cost to move from the frontier to the next sampling point.

Dijkstra's shortest path algorithm is used to find the shortest path to each frontier. Eq. 6.1 is used to calculate the traveling cost to each frontier, E_1. To estimate the cost from the frontier to the goal, we assume that the terrain pattern is the same for the path from the current robot location to the frontier as that from the frontier to the sampling point. Therefore, the estimated energy cost, E_2, can be derived by

$$E_2 = E_1 * d_2 / d_1 , \qquad (6.3)$$

where

d_1 = path distance from the current robot location to the frontier, and

d_2 = estimated distance from the frontier to next sampling point.

d_2 is the straight line Euclidean distance from the frontier to the target sampling point. We assume that no obstacles exist in this path. Actually, the straight path is not always the path of least energy, but this estimation is underestimated in most cases.

6.5 Results and Discussion

6.5.1 Simulation setup

In the simulation, a 4-wheel drive robot (mass: 16 kg; length: 50 cm; width: 49 cm; height: 26 cm) with 4 identical wheels (wheel diameter: 25.2 cm; wheel width: 7.5 cm) was used. A 3D laser sensor (with a 90° field of view, 50 m depth of field, and 1:1 aspect ratio) was used as the vision system. The scouting algorithms were tested in the five fields and four starting locations. A uniform CI of 75 N/cm^2 is assumed for all the five silt fields (Goering et al., 2003). It was assumed that the robot traveled at a steady speed of 3 m/s. Table 6.1 shows the robot start locations. In these cases, the scouting robot started with a completely blank map. The resolution of the maps in this simulation is 10m. In the systematic grid sampling problem, we set 150 m as the interval in x direction and 87 meters in the y direction.

Table 6.1 Robot starting locations in agricultural field 1

Field	Starting A (x,y) in units (m, m)	Starting B (x,y) in units (m, m)	Starting C (x,y) in units (m, m)	Starting D (x,y) in units (m, m)
Agricultural field 1	(20,300)	(380,30)	(200,30)	(50,700)

In this chapter, the performance of the algorithm was validated by comparing it with two other strategies—line sweeping + bug, and potential function methods.

(1) Line sweeping + bug method—after the sampling points are extracted, the robot will visit every sampling point in sequence along the line sweeping direction. Then the bug method (Lumelsky and Stepanov, 1987) is used to determine a path between two adjacent sampling points. In this method, when the next sampling point selected is located outside the known terrain, the robot will choose to go

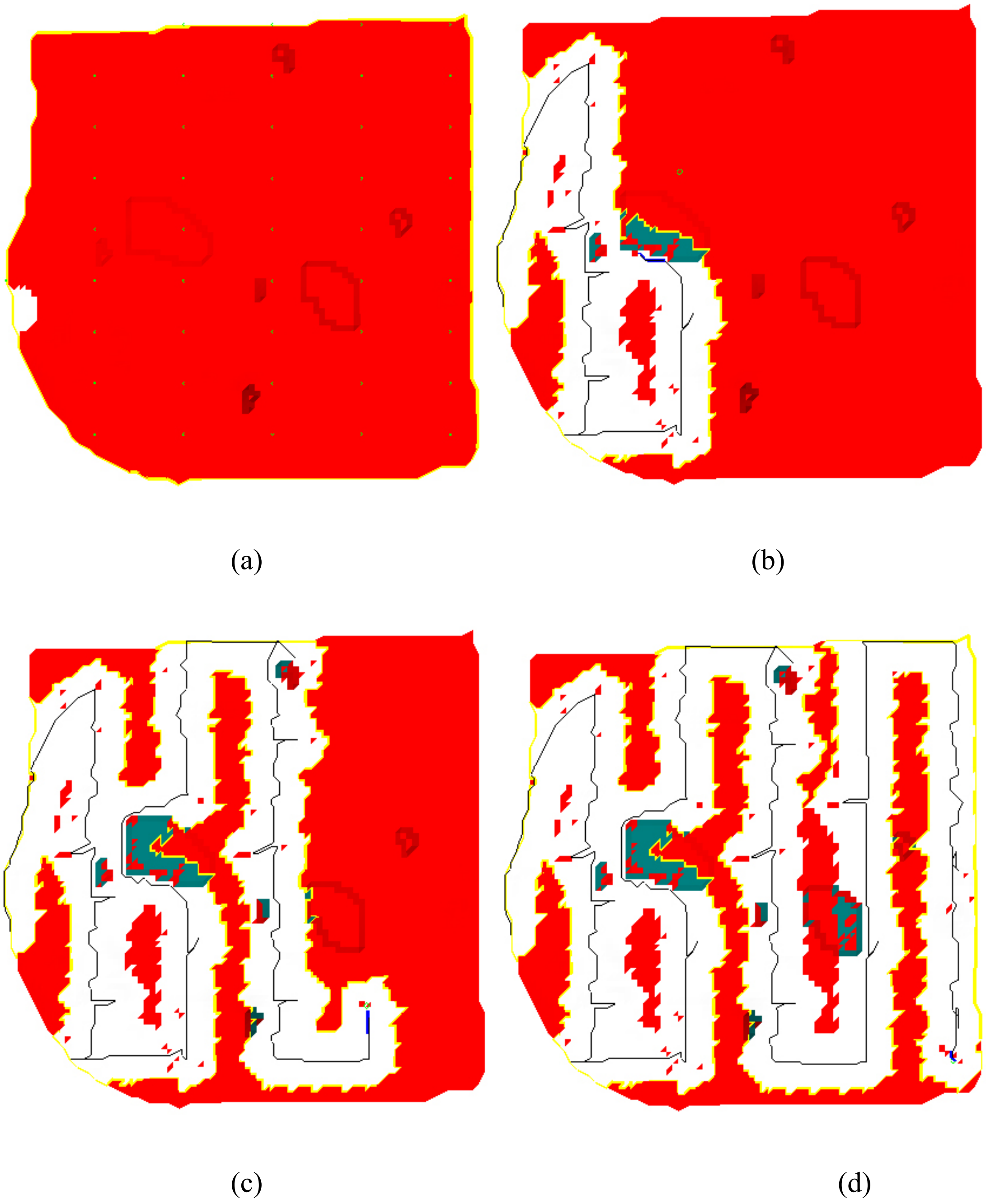
around the nearest boundary edge between the known and unknown terrain until the next sampling point is inside the known area or the robot is blocked. As the sampling point is detected by the sensor, a path between the current rover location and the sampling point can be planned using Dijkstra's shortest path algorithm.

(2) Potential function method—after the sampling points are extracted, the robot will be attracted to visit the sampling point that requires the least energy cost to reach. When the next sampling point selected in this way is located inside the unknown terrain, the heuristic function defined by Eq. 6.1 is then used to determine an optimal path between two adjacent sampling points.

6.5.2 Results

The complete results from the combination of all the five fields and four starting positions for all methods can be found in Appendix E and F. Sample results for agricultural field 1 from starting point A can be found in Figures 6.3, 6.4, 6.5, and 6.6.

Captured screenshots of a simulation in agricultural field #1 (start location A) for line sweeping + heuristic method are shown in Fig. 6.3, where the traveled path (black lines) and updated map (white indicates explored terrain) are plotted at selected sampling times. Figure 6.3 (d) shows the scouting result ended after forty-five sampling points were visited.

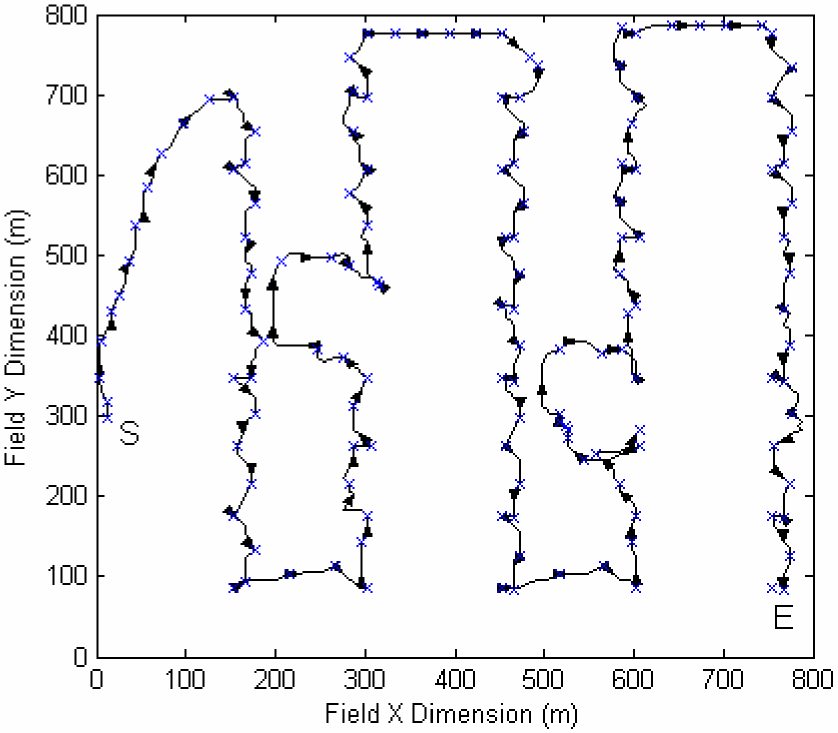


(a) (b)

(c) (d)

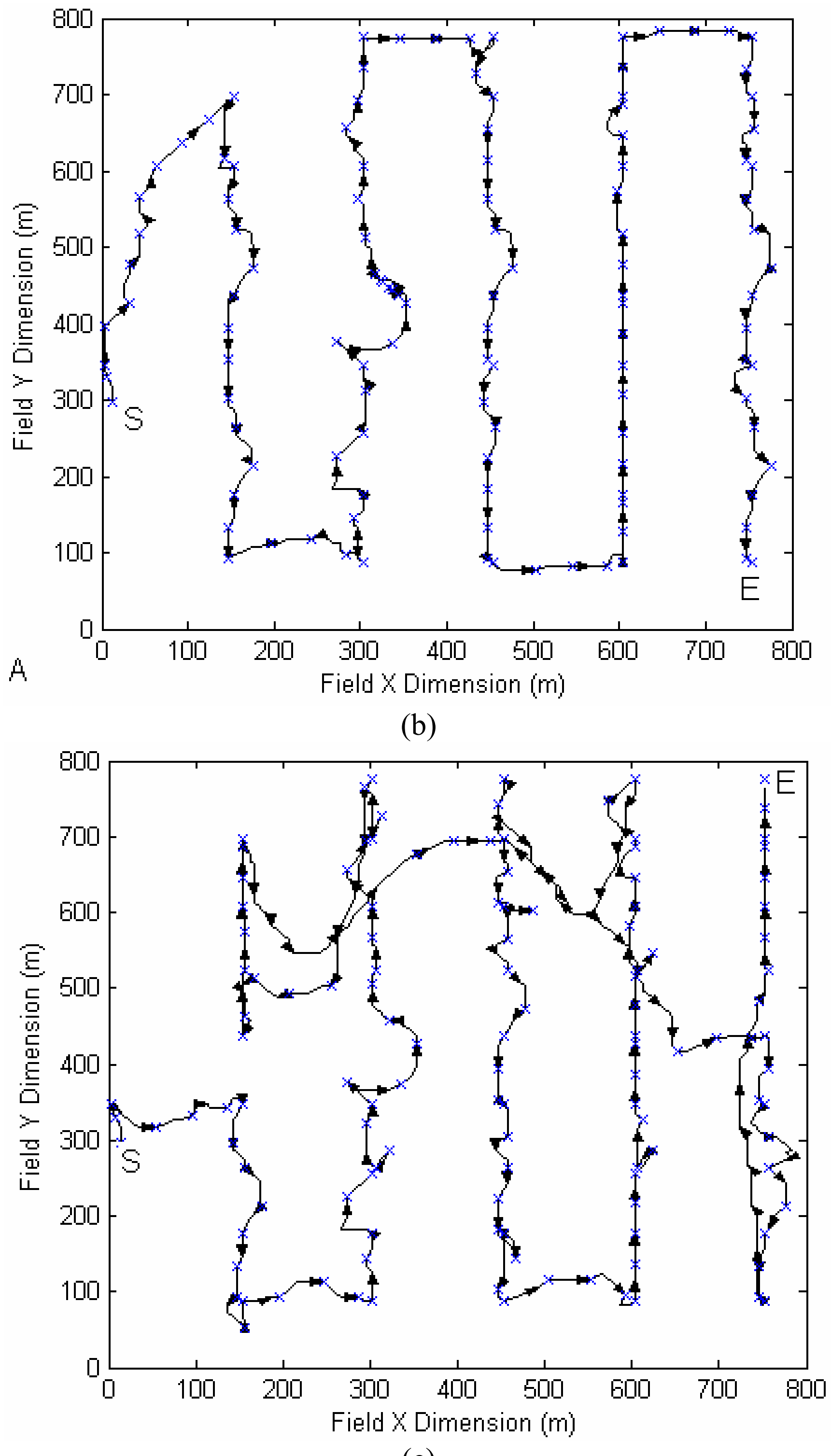
Figure 6.3 Traveled paths (black lines) by the scouting algorithm and updated maps (white indicates explored terrain) are plotted at selected sampling times: (a) initialization, (b) after visiting thirteen samples location, (c) after visiting twenty-nine sample points, and (d) after visiting forty-five sample points.

The trajectory path generated using the line sweeping + bug, line sweeping + heuristic function, and potential function methods are given in Figs. 6.4(a) through 6.4(c), respectively. The black lines represent the path; the arrows show the vehicle's travel direction; the cross marks represent viewpoints where the robot stopped to take a scan. It shows that the path generated using the potential function method has more overlap than the paths generated using the line sweeping methods.



(a)

Figure 6.4 Trajectories generated in the simulation (arrows represent the travel direction; cross marks represent viewpoints; started at S, and ended at E): (a) line sweeping + bug method.

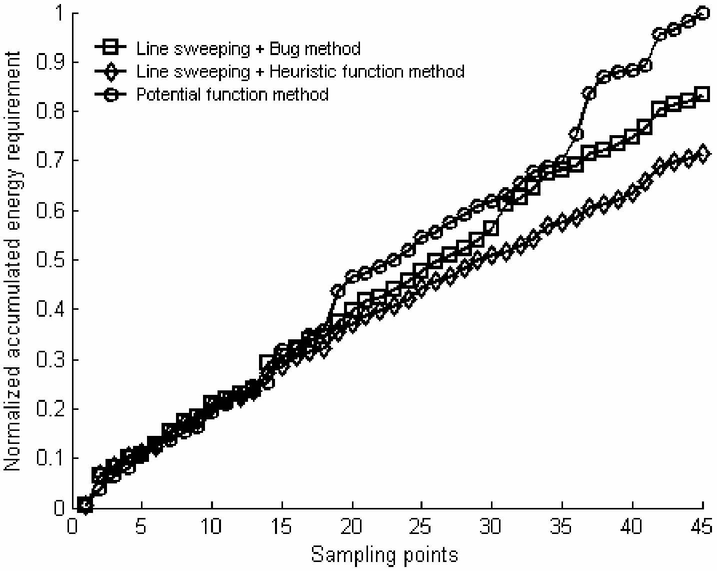


(b)

(c)

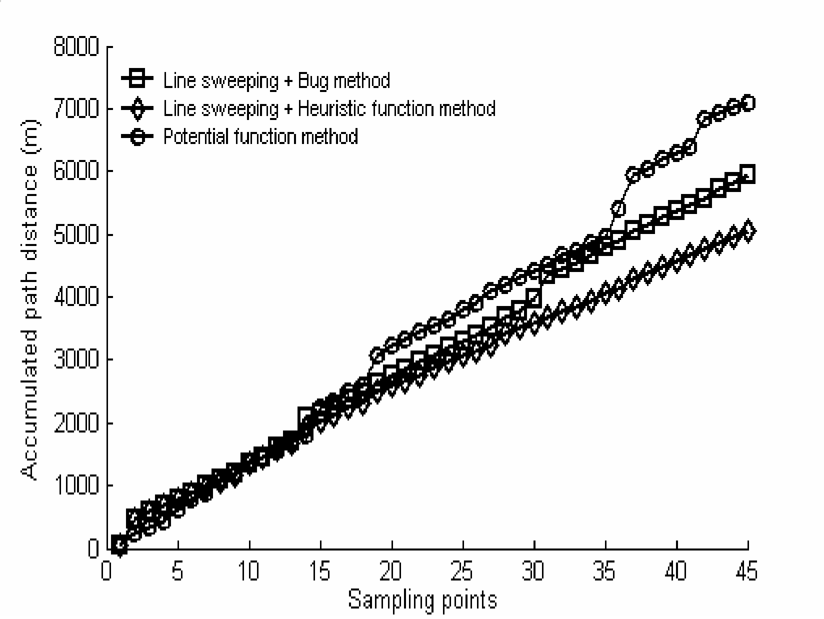
Figure 6.4 (continued) Trajectory generated in the simulation (arrows represent the travel direction; cross marks represent viewpoints; started at S and ended at E): (b) line sweeping + heuristic function method, and (c) potential function method.

Figure 6.5 shows the relationship of the number of sampling points explored and the energy requirement, the distance traveled, the time requirement (including planning time and navigation time), and the scan number for agricultural field 1 starting from location A for the three scouting algorithms. It is desirable to have all the samples collected with less energy consumption, a shorter traveled distance, a lower number of scans, and a shorter time. The relationship between the energy, path length, traveling time, and sampling point number were linear for the two line sweeping methods. That relationship was linear in the first part of exploration for the potential function method; however, it was highly nonlinear for last part of exploration. The performance of the three methods in the first 18 sampling points was almost the same for the energy requirement, path length, and traveling time. The cost in energy, distance, and scan number increased exponentially after thirty-five sample points for the potential function method. It is apparent from Fig. 6.5(d) that the scan number of the line sweeping + bug method was greater than that for the potential function and line swiping + heuristic function methods. The relationship between the number of scans and the sampling point explored was linear for all three methods.

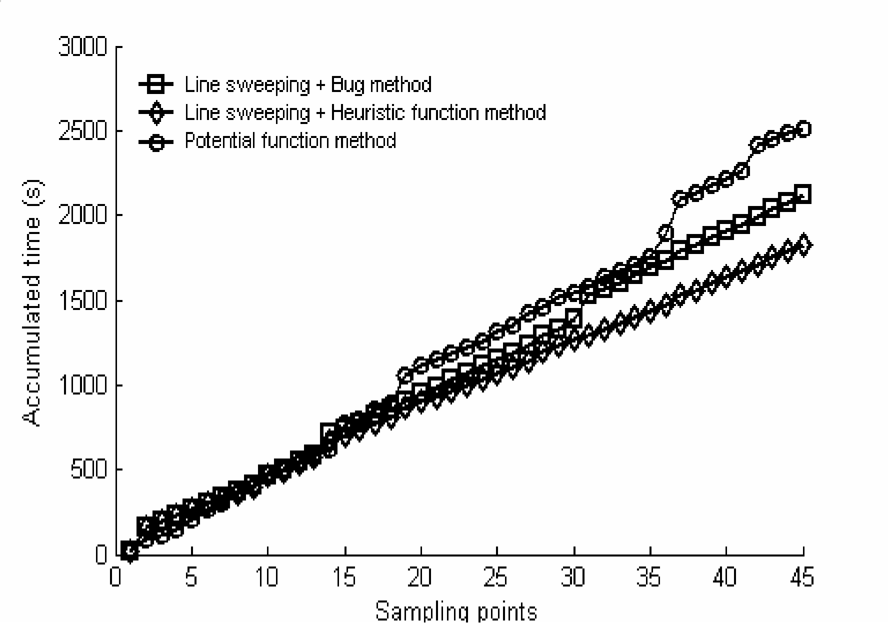


(a)

Figure 6.5 Result of the scouting algorithms for agricultural field 1 (start point A): (a) normalized accumulated relative energy requirement as a function of sample points traversed.

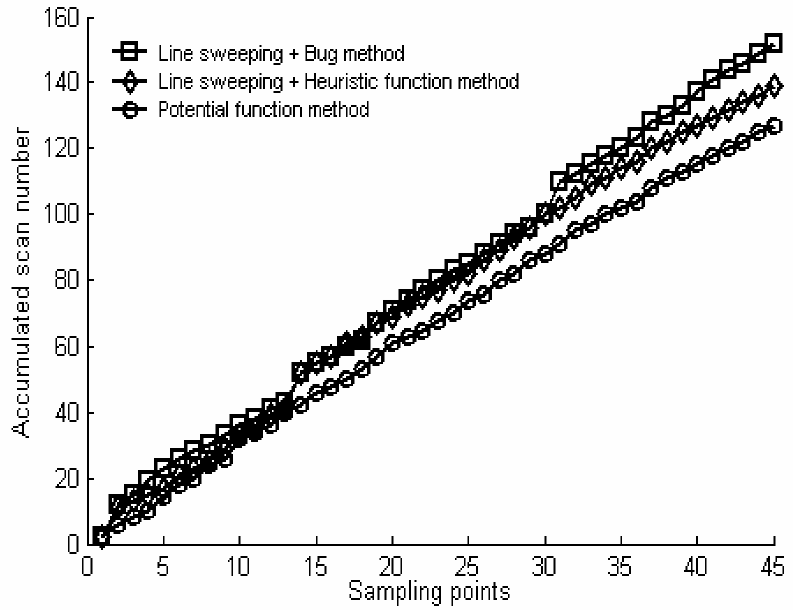


(b)



(c)

Figure 6.5 Result of the scouting algorithms for agricultural field 1 (start point A): (b) accumulated path length traveled by the robot as a function of sample points traversed, and (c) accumulated time required as a function of sample points traversed.

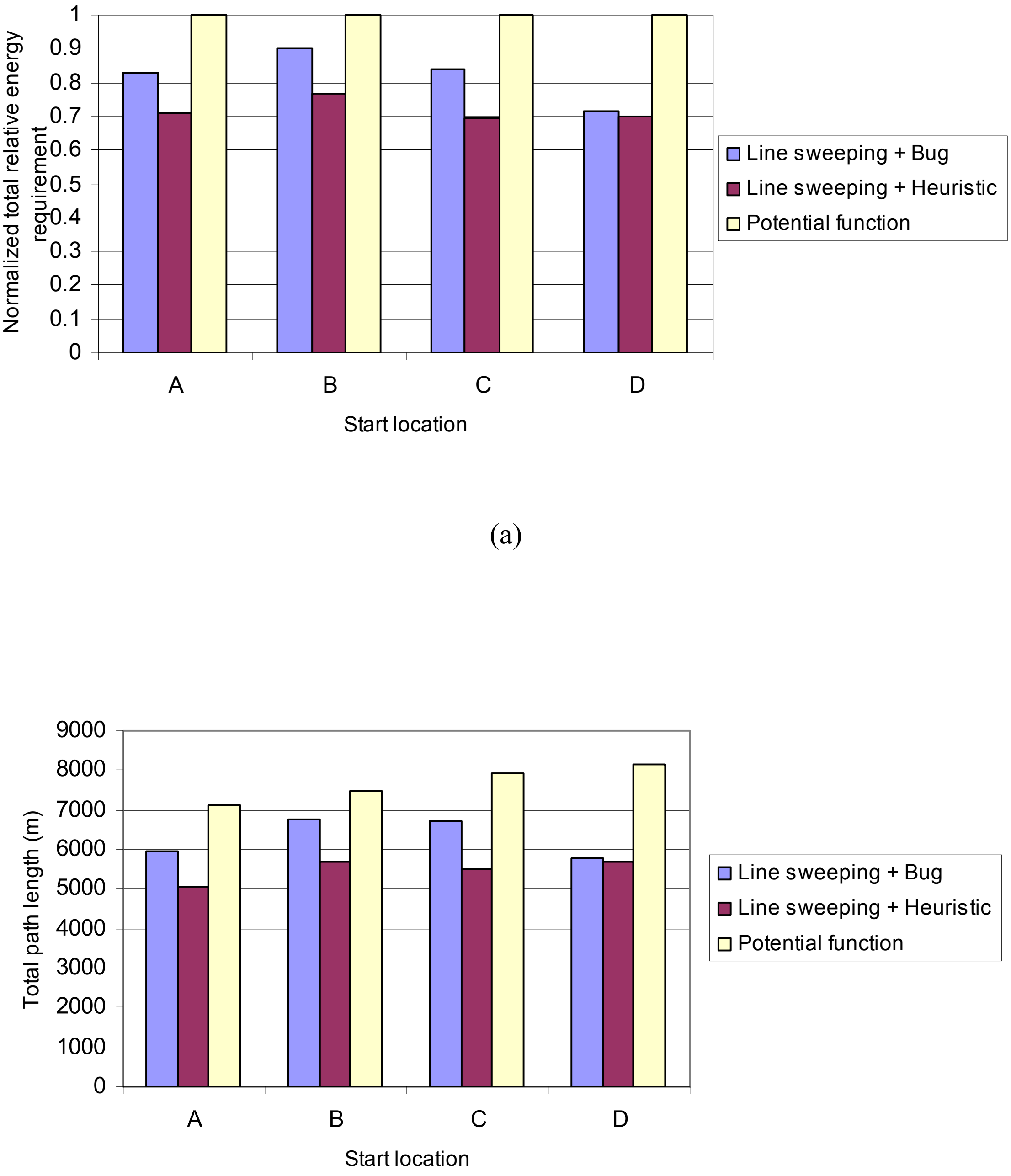


(d)

Figure 6.5 (continued) Result of the scouting algorithms for agricultural field 1 (start point A): (d) accumulated scan number as a function of sample points traversed.

A comparison of the performance of the three methods for energy consumption, path length, time, and the number of scans for agricultural field 1 starting four locations were given in Fig. 6.6. In Fig. 6.6(a), it shows that the energy requirements for the potential function method were higher than the energy requirements for the line sweeping methods. The line sweeping + heuristic function method outperformed the line sweeping + bug method in terms of the energy cost for all four starting locations. Figures 6.6(b) and 6.6(c) show that the performances of these methods for distance and traveling time was similar to the performance for the energy cost.

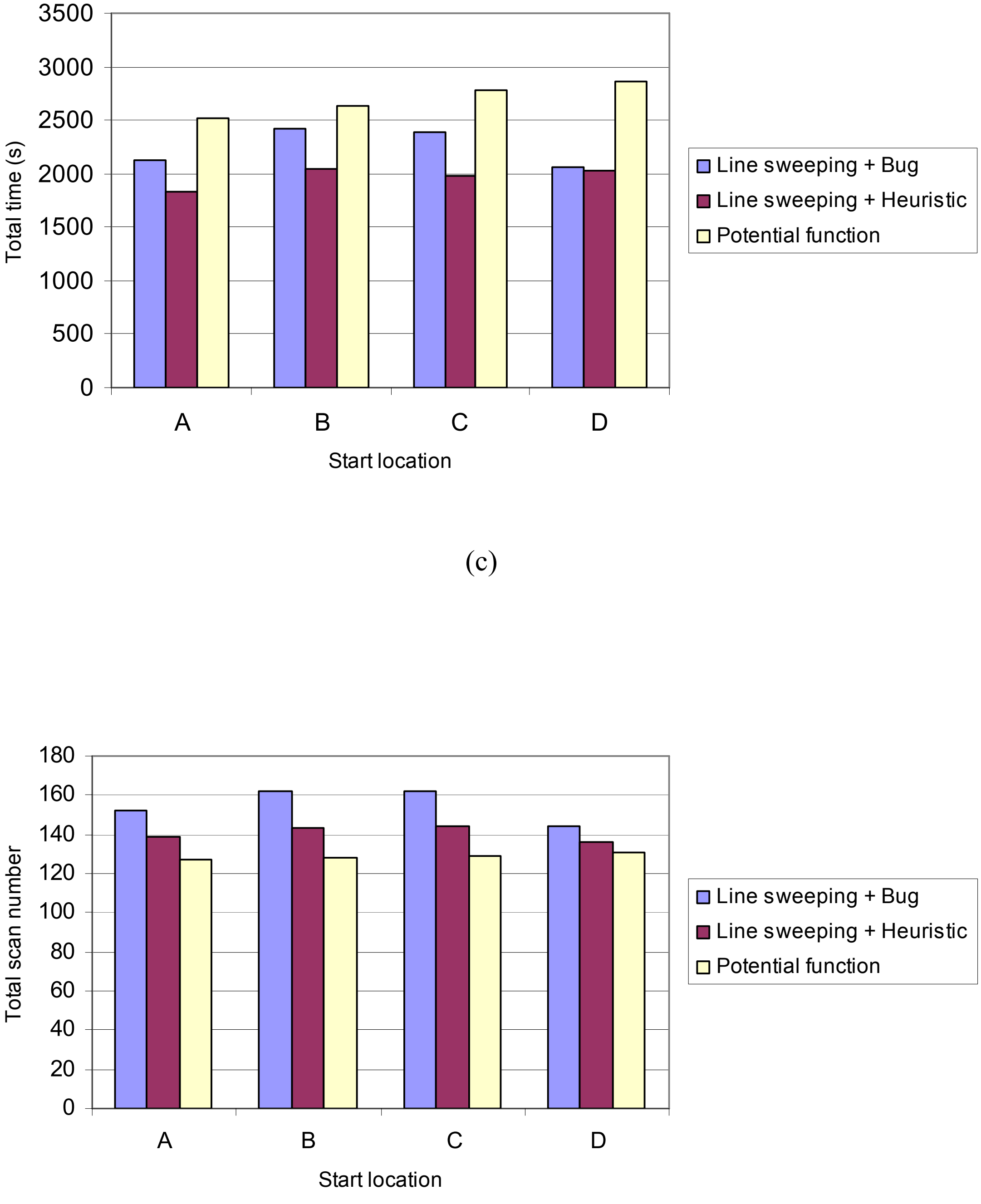
It is apparent from Fig. 6.6(d) that the number of scans took when using the line sweeping methods was greater than the number of scans taken when using the potential function method for all four start locations. The line sweeping + heuristic function method greatly outperformed the bug method in the scan number for start locations A, B and C, while the heuristic function method only slightly exceeded the bug method for start location D.



(a)

(b)

Figure 6.6. Result of the scouting algorithms for 90% of the exploration for agricultural field 1 at four starting locations: (a) normalized total relative energy requirement, (b) path length traveled by the robot.



(c)

(d)

Figure 6.6 (continued) Result of the scouting algorithms for 90% of the exploration for agricultural field 1 at 4 starting locations: (c) time required, and (d) the number of scans.

6.6 Conclusions

This manuscript has presented a next best viewpoint field scouting algorithm to address the agricultural field coverage problem using a 3D image sensor. This sensor-based weed scouting algorithm can deal with different situations with or without an *a priori* map or in a partly known environment. From the simulation results, we can conclude:

(1) The path length, energy requirements, and time requirements (including both the planning time and traveling time) of the line sweeping methods were less than those for the potential function method for all four start points. The line sweeping + heuristic function method required over 23%, 23%, and 22% less, in terms of the path length, energy requirements, and time requirements respectively, than the potential function; the line sweeping + bug method required over 9%, 9%, and 8% less in terms of the path length, energy requirements, and time requirements respectively, than did the potential function.

(2) The line sweeping + bug method required over 17% more energy that did the line sweeping + heuristic function method for start point A, B and C, while there was only 3% difference of energy requirement between the two line sweeping methods for start point D.

(3) Furthermore, the scan number of the potential method was less than the scan numbers for the line sweeping related methods. The bug method required over 10% more scans than did the potential function method, while the potential method outperformed the heuristic method by over 3% when considering scan numbers.

6.7 References

Adamchuk, V. I., M. T. Morgan and D. R. Ess. 1999. An automated sampling system for measuring soil pH. *Transactions of the ASAE* 42(4): 885-891.

Åstrand, B. and A. Baerveldt. 2002. An agricultural mobile robot with vision-based perception for mechanical weed control. *Automatic Robots* 13: 21-35.

Baio, F. H. R. and L. A. Balastreire. 2002. Evaluation of a Site Specific Chemical Application System Based on the Spatial Variability of Weeds. In *Proceedings of the World Congress of Computers in Agriculture and Natural Resources* 225-231. Iguacu Falls, Brazil. March 13-15.

Bajwa, S. G. and L. F. Tian. 2001. Aerial CIR remote sensing for weed density mapping in a soybean field. *Transactions of the ASAE* 44(6): 1965-1974.

Bishop, T. F. A. and A. B. McBratney. 2002. Creating field-extent digital elevation models for precision agriculture. *Precision Agriculture* 3(1): 37-46.

Choset, H. 2001. Coverage for robotics—A survey of recent results. *Annals of Mathematics and Artificial Intelligence* 31: 113-126.

Choset, H. and Pignon, P. 1997. Coverage path planning: The boustrophedon cellular decomposition. In *International Conference on Field and Service Robotics*. Canberra, Australia.

Dupuis, E., P. Allard, J. Bakambu, T. Lamarche and W·H. Zhu. 2004. Towards autonomous long-range navigation. In *the 8th ESA Workshop on Advanced Technologies for Robotics and Automation 'ASTRA 2004'*, ESTEC. Noordwijk, The Netherlands. November 2-4.

Ferguson,R., K. Frank, G. Hergert, E. Penas and R. Wiese. 1998. Guidelines for soil sampling. *http://ianrpubs.unl.edu/Soil/.* Lincoln NE.

Fontaine, V. and T.G. Crowe. 2006. Development of line-detection algorithms for local positioning in densely seeded crops. *Canadian Biosystems Engineering* 48: 7.19-7.29.

Goering, C.E., M.L. Stone, D.W. Smith and P.K. Turnquist. 2003. Off-Road Vehicle Engineering Principles. St. Joseph, MI: ASABE. 474 pages.

Goel, P.K., S.O. Prasher, P.M. Patel, D.L. Smith and A. DiTommaso. 2002. Use of airborne multi-spectral imagery for weed detection in field crops. *Transactions of the ASAE* 45: 443-449.

Hert, S., S. Tiwari and V. Lumelsky. 1996. A terrain covering algorithm for an AUV. *Autonomous Robots* 3: 91-119.

Huang, W. 2001. Optimal line-sweep-based decompositions for coverage algorithms. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*: 27- 32. Seoul, Korea, May 21-26.

Keicher, R. and H. Seufert. 2000. Automatic guidance for agricultural vehicles in Europe. *Computers and Electronics in Agriculture* 25: 169-194.

Kitware Inc. 2005. The Visualization Toolkit. *http://www.vtk.org.* Accessed June 26, 2005. New York.

Korol, M. Accessed August 15, 2006. Canadian Fertilizer Consumption, Shipments and Trade 2001/2002. In *http://www.agr.gc.ca/*. Agriculture and Agri-Food Canada.

Lumelsky, V. and A. Stepanov. 1987. Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica* 2: 403-430.

Moorehead, S. 2001. Autonomous surface exploration for mobile robots. *Doctoral dissertation*, technical report. CMU-RI-TR-01-30. Robotics Institute, Carnegie Mellon University.

Nielsen, K.M., Andersen, P., Pedersen, T.S., Bak, T., and Nielsen, J.D. 2002. Control of an Autonomous Vehicle for Registration of Weed and Crop in Precision Agriculture. In *IEEE Conference on Control Applications CCA/CACSD 2002*, Glasgow Scotland.

Ollis, M. and A. Stentz. 1996. First results in vision-based crop line tracking. In *Proceedings of the IEEE Robotics and Automation Conference* 951–956. Minneapolis, MN, April 22-28.

Reid, J.F. and S.W. Searcy. 1987. Vision-based guidance of an agricultural tractor. *IEEE Control Systems* 7 (12): 39-43.

Reid, J. F., Q. Zhang, N. Noguchi and M. Dickson. 2000. Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture* 25: 155-167.

Saraswat, D., R. Ehsani, N. Watermeier and M. Sullivan. 2003. Potential application of yield data for creating topographic maps. In *2003 ASAE Annual Meeting.* Paper no: 031084. Las Vegas, Nev., July 27-30.

Schmidt, J. P., R. K. Taylor and R. J. Gehl. 2003. Developing topographic maps using a submeter accuracy global positioning receiver. *Applied Engineering in Agriculture* 19(3): 291-300.

Tillett, N. D. and T. Hague. 1999. Computer-vision-based hoe guidance for cereals — an initial trial. *Journal of Agricultural Engineering Research* 74(3): 225-236.

Torii, T. 2000. Research in autonomous agriculture vehicles in Japan. *Computers and Electronics in Agriculture* 25(1-2): 133-153.

Westphalen, M. L., B. L. Steward and S. Han. 2004. Topographic mapping through measurement of vehicle attitude and elevation. *Transactions of the ASAE* 47(5): 1841-1849.

Yamauchi, B. 1997. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation:* 146-151. Monterey, CA. July 10-11.

Zelinsky, A., R.A. Jarvis, J.C. Byrne and S.Yuta. 1993. Planning paths of complete coverage of an unstructured environment by a mobile robot. In *Proceedings of International Conference on Advanced Robotics*: 533-538. Tokyo, Japan.

7. SUMMARY AND CONCLUSIONS

The goals of this project were to develop coverage path planning algorithms for automatic construction of topographic maps for unknown or partially known rough agricultural fields and to develop scouting algorithms for agricultural field sampling. A new triangular mesh map was presented that allowed the robot to maintain a very rich representation of the environment and allowed a smooth path. A viewing frustum model and ray casting algorithm were described to facilitate the 3D image sensor simulation and estimate new terrain information.

Three methods have been attempted to find an optimal next best viewpoint to address the automatic mapping problem. A two-stage strategy was first used to find the next best viewpoint by considering both the distance and the slope factor in the cost function.

In the second method, the mapping problem was addressed with the development of a novel energy cost function to plan next best viewpoints. The energy cost function considered the distance, terrain elevation change, and other vehicle parameters.

Finally, information-based exploration algorithms were presented to address the problem of next best viewpoint in modeling large rough unstructured environments with an *a priori* coarse map. A terrain visibility analysis based on a viewing frustum model and a ray casting algorithm was proposed to estimate the new information gain.

In addition to the strategy proposed to address the automatic mapping problem, a 3D frontier-based scouting algorithm was presented to address the agricultural field coverage problem raised by weed mapping or soil sampling tasks.

Simulation results in two typical western Canadian agricultural fields and three ideal fields are presented to demonstrate the algorithms. To evaluate the developed algorithms, extensive comparison of a variety of algorithms in terms of path length, energy requirement, number of scans, and time requirement was discussed in four chapters. Specific chapter conclusions have been summarised in the following sections.

7.1 Chapter 3: Vision-based exploration algorithms for rough terrain modeling using triangular mesh maps

Chapter 3 was related to objective 1 of the project and was dedicated to the development of a simulation platform for the project. A two-stage exploration strategy was developed to exploit the potential of developing an exploration algorithm based on a triangular mesh map model and a 3D image sensor model. In the first stage of the exploration, the robot tended to visit frontiers near the outer boundary of the terrain, while the robot attempted to fill any holes left by the first stage in the second stage.

By comparing the performances between the two-stage algorithm and the line sweeping method, specific results from a case study can be concluded as:

1) the greedy method was more efficient at early stages and required about 16% less than did the line sweeping method in terms of travel distance and time to complete 75% exploration;
2) from the perspective of the whole exploration procedure, there was little difference in the traveled distance and time consumption required for either method, and,

7.2 Chapter 4: An exploration strategy based on the minimum energy consumption for autonomous construction of agricultural field maps

This chapter relates to objective 2 of the project. This chapter was dedicated to develop an energy cost function and explore the possibilities of using the energy requirement as a cost function to select the next best viewpoint. The energy cost function considered the distance, terrain elevation change, and other vehicle parameters.

To validate the developed next best viewpoint algorithm, a variety of strategies were developed and their performances were compared extensively in terms of energy requirement, time, traveled distance, and number of scans. Major findings can be concluded as:

1) The greedy method required about 80% less energy, distance, and time than the random method for all 4 start points to complete 50% and the whole exploration task;
2) The greedy method required between12% and 48% energy, between 10% and 47% length, between 8% and 45% time less than the other three methods, including the line sweeping, spiral method 1, and spiral method 2, for all start points A, B, C, and D to complete 50% of the exploration task;
3) After completing 90% of the exploration task, the line sweeping method outperformed the greedy method by about 5% in terms of energy when starting A, while the greedy method exceeded the line sweeping method by 5%, 12%, and 0.5% in terms of energy efficiency for start location B, C, and D, respectively;
4) The spiral pattern 2 performed better than the spiral pattern 1 in terms of energy consumption, distance, and time by about 15% for most cases;
5) There was not much difference between the number of scans taken using the greedy and random methods;
6) Although the terrain type and starting location had influences on the performance of the exploration methods, the greedy method could adapt to different field patterns, and the minimum energy demand was achieved by minimizing the energy cost during the exploration task in the earlier stage;
7) The greed method was shown to be the best choice for partial coverage tasks, while the line sweeping method might be a better choice to explore the whole environment.

7.3 Chapter 5: Information-based exploration algorithms for rough terrain modeling using triangular mesh maps

This chapter is associated with objective 3 of the thesis. It focused on the investigation of the information-based exploration algorithms. Specific results can be concluded as:

1) The exploration strategy, which incorporated the energy consumption and the information gain with a ray tracing algorithm using a coarse map, showed an advantage over other policies in terms of the total energy consumption and the path length by at least 4%;
2) The information gain 2 method required 25%, 24%, 4%, 34%, and 4% less energy that information gain 1 method for agricultural field 1, agricultural field 2, mountain, hole, and slope environment, respectively when starting at location A;
3) The greedy method using the frustum culling and the ray casting algorithms required more planning time than the method that used the sensor footprint to estimate new information gain;
4) The maximum information gain methods required between 8% and 251% more energy and path length than the minimum energy and the other two methods, which incorporated both energy requirement and the information gain;
5) The scan number for the greedy method that did not consider information gain was at least 14% larger than the methods that considered the information gain.

7.4 Chapter 6: Sensor-based scouting algorithm for agricultural robots

Chapter 6 was related to objective 4 of the thesis and was focused on the development of field scouting algorithms. In this chapter, three scouting algorithms, including line sweeping + heuristic function, line sweeping + bug, and potential function methods, were developed and their performances were compared. Following specific conclusions have been drawn:

1) The line sweeping + heuristic function method required over 23%, 23%, and 22% less in terms of the path length, energy requirements, and time, respectively, than the potential function;
2) The line sweeping + bug method required over 17% more energy than the line sweeping + heuristic function method when starting at point A, B and C, while there was only 2% difference of energy requirement between the two line sweeping methods;
3) The scan number of the potential method was less than the scan numbers for the line sweeping related methods. The bug method required over 10% more scan numbers than the potential function method, while the potential method outperformed the heuristic method by 3% in terms of scans.

8. CONTRIBUTIONS AND RECOMMENDATIONS

8.1 Contributions to knowledge

The contributions of this research to engineering knowledge can be stated in four major aspects: development of a simulation tool based on a 3D sensor model and 3D visualization, design and development of exploration algorithms based on a triangular mesh map and a nominal energy cost equation which can be used as a weight factor in exploration algorithms, design and development of scouting algorithms for agricultural field scouting tasks, and an extensive comparison of the different algorithms.

Simulation tool: A simulation tool was developed and used to facilitate the development of exploration, path planning algorithms. A 3D sensor model was developed based on a frustum culling and ray tracing algorithm. Over ten thousand of lines of code was developed using Java and the Visual Tool Kit (VTK) library to deal with the triangular mesh map, visualize and record the intermediate simulation results.

Exploration algorithms: As far as the author knows, this is the first recorded attempt of developing exploration algorithms based on a triangular mesh map in the field robotics research community. A two-stage greedy algorithm was developed based on the triangular mesh map. An energy cost function was developed. The energy cost function considered the distance, terrain elevation change, vehicle slip rate, and other vehicle parameters. As far as the author knows, this is also the first attempt to choose the next best viewpoints using the energy requirement in the robotic research community.

Scouting algorithms: As far as the author knows, this is the first recorded attempt of developing sensor-based field scouting algorithms to address the automatic sampling in an open area without apparent landmarks but with obstacles in the agricultural robotic research society. Three scouting algorithms, including line sweeping + heuristic function, line sweeping + bug, and potential function methods were developed, tested, and compared.

Algorithm comparison: As far as the author knows, this is the first recorded attempt to evaluate different algorithms in terms of the energy requirement, travel distance, time, and the number of scans in the robotic research community. A variety of algorithms were designed or imported to incorporate a triangular mesh map. Most algorithms were tested and evaluated in two typical western Canadian agricultural fields and 3 ideal fields, and 4 start locations for each field.

8.2 Recommendations for future research

Many more simulation runs, including more fields and starting locations, should be considered in the future research. Although a physical experiment was considered during the planning stage of this project, it was not conducted. To evaluate the algorithms, extensive physical experiment action should be performed in a real-world agricultural field. This will require significant research effort to complete the whole robotic navigation system. A robotic platform equipped with a 3D image sensor and a position sensor (a GPS or a relative position sensor) should be developed. A control system which can guide the robot to move along a planned trajectory path should be implemented. A map stitching algorithm which combines a variety of sensor readings into a triangular mesh map should be developed.

As part of the current simulation work, a frustum consisting of six planes was used to model the vision sensor. Segments of the surface of a sphere should be used to represent the near field plane and the far field plane of the frustum for large field of view (near 180 degrees and above) in future research. Due to limitation of computing ability, the triangle size was not considered.

An irregular triangular mesh map should be used in the future research. Although the regular triangular mesh map used in this work was easy to implement for the sensor coverage task, it required a large amount of memory. An irregular triangular mesh map capable of representing a large-scale environment with variable-size triangles would be used to reduce the memory requirement in the future system. The exploration algorithms should be adapted to the irregular mesh map. The algorithm should be put in light of the difficulties with the violation of the small triangle assumption, which could result from an irregular mesh.

The energy cost function developed in this thesis assumed that the soil hardness is uniform in the whole filed. The soil hardness might be highly variable in one field considering soil type and moisture content can cause significant changes in CI values. The soil hardness map should be integrated in the energy cost function if a soil strength map is available in future work.

Further experimental study should be conducted to verify the energy function. The energy function used in this work was derived from the tractive equations. It was assumed that the energy was used to do only useful work, including energy required to overcome the elevation change and the rolling resistance, and the vehicle was simplified as a rigid body. A vehicle model and any energy loss should be considered in future research. It is noted that the tractive equations were experimentally developed for specific tires and vehicle systems. Variations in the tire size may cause large deviations in estimating the energy consumption. Experiments should be considered to verify the energy function in further study.

APPENDICES

APPENDIX A - DIRECTORY OF DATA AND PROGRAM FILES

The following is an annotated directory of the primary data and programs on the accompanying DVD-ROM.

A.1 Root directory

Folder Name	Description
Farm map	Contains the elevation data, generated vtp mesh map, and related programs. Refer to A.1.1
Java programs	Contains Java programs and documents. Refer to A.1.2
Matlab programs	Contains Matlab codes for drawing figures in this thesis. Refer to A.1.3
Results - figures and tables	Contains figures and tables. Refer to A.1.4
Results - movies	Contains movies and programs used to generate movies. Refer to A.1.5
Results_raw_data	Contains all the raw data generated by simulation

A.1.1 Farm map

Folder Name	Description
farm map txt	Contains the elevation data (xyz information)
farm map vtp	Contains Java programs and documents.
farm_map_IndiaHead	Contains the original GPS data from IndianHead Farm and intermediate files using ArcGIS.
Ideal map	Contains Matlab codes to generate the ideal maps and the elevation data for ideal maps.

A.1.2 Java programs

Folder Name	Description
coverage planner documents	Contains the documents generated by Javadoc
coverage planner program	Contains Java programs. Refer to A.1.2.1

A.1.2.1 coverage planner program

Item Type	Name	Description
folder	Doc	Related documents
folder	Samples	Test package for mars environment
folder	samples_ag	Test package for agricultural environment
folder	samples_aroundboundary	Test package for the around boundary method

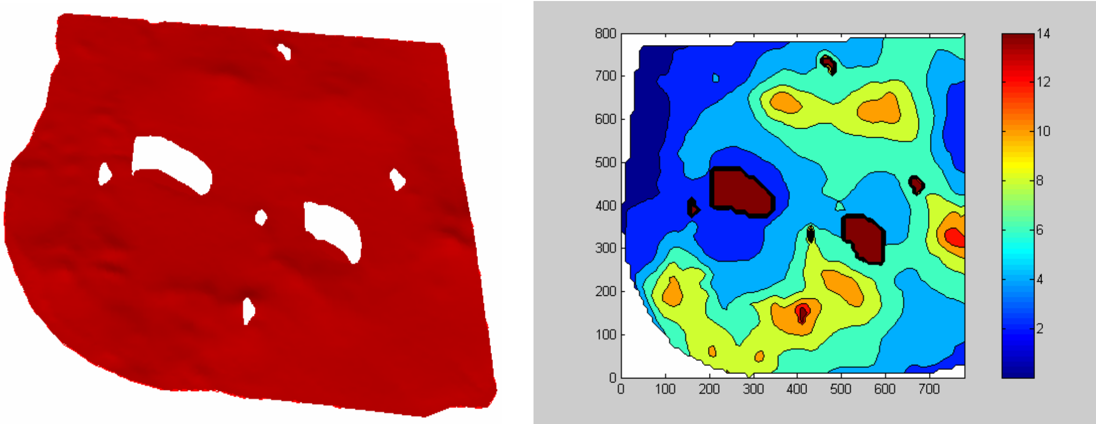
folder	samples_linesweeping	Test package for the line sweeping method
folder	samples_objective1_nostage	Test package for the greedy method (no information gain)
folder	samples_objective2	Test package for the greedy method (information gain)
folder	samples_random	Test package for the random method (not used)
folder	samples_randomfinal	Test package for the random method
folder	samples_scouting	Test package for scouting algorithms
folder	samplesfield1	Test package for manuscript 1
folder	Visibility	Package used for visibility analysis
Java file	Camera.java	Camera class
Java file	CoverageMap.java	CoverageMap class
Java file	CoveragePlanner.java	CoveragePlanner class
Java file	CoveragePlannerRayTracer.java	CoveragePlannerRayTracer class
Java file	CycleDetector.java	CycleDetector class
Java file	DijkstraCoveragePath.java	DijkstraCoveragePath class
Java file	EnergyCostFunction.java	EnergyCostFunction class
Java file	Frontier.java	Frontier class
Java file	GVertex.java	GVertex class
Java file	ICostFunction.java	ICostFunction class
Java file	IImagingSensor.java	IImagingSensor class
Java file	ImagingSensor.java	ImagingSensor class
Java file	IterativeMap.java	IterativeMap class
Java file	LinearRegress.java	LinearRegress class
Java file	MobilityEdge.java	MobilityEdge class
Java file	MyPolygon.java	MyPolygon class
Java file	MyTriangle.java	MyTriangle class
Java file	MyVertex.java	MyVertex class
Java file	RayTracer.java	RayTracer class
Java file	SamplingPoint.java	SamplingPoint class
Java file	StatisticalCoveragePlanner.java	StatisticalCoveragePlanner class
Java file	XYZIO.java	XYZIO class

A.1.3 Matlab plot thesis

plot_objective1	Matlab programs for objective 1
plot_objective2	Matlab programs for objective 2
plot_objective2_5 methods	Matlab programs for objective 2 (5 methods)
plot_objective3	Matlab programs for objective 3

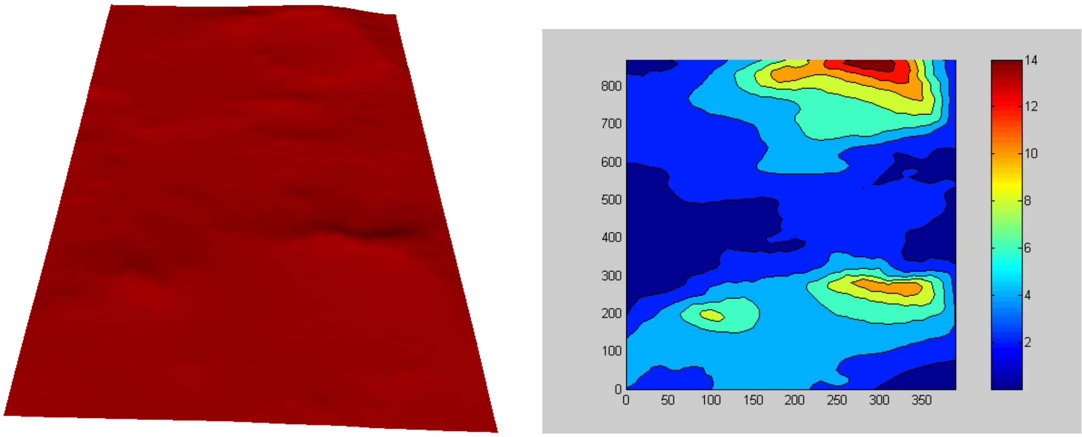
APPENDIX B – TEST FIELDS AND STARTING LOCATIONS

B.1 Test fields



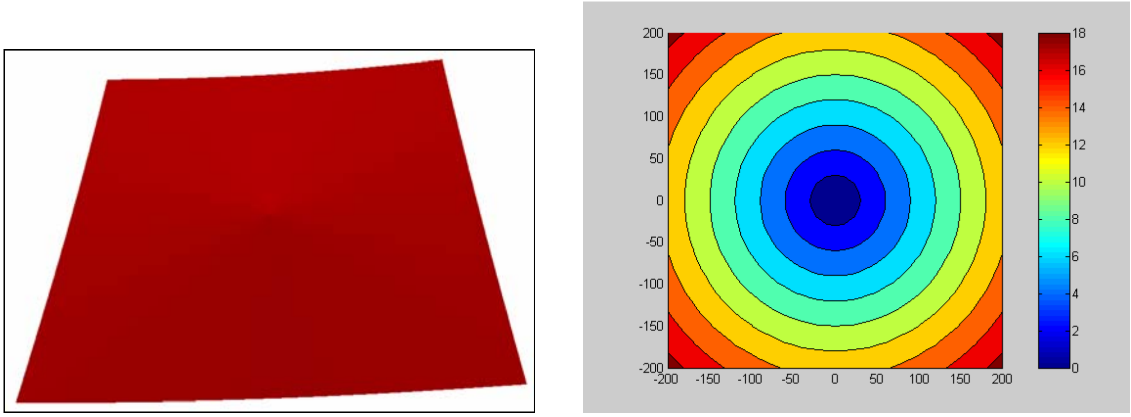
(a) (b)

Figure B.1 Maps of agricultural field 1 in Indian Head Research Farm: (a) triangular map of agricultural field 1, (b) topographic map of agricultural field 1.



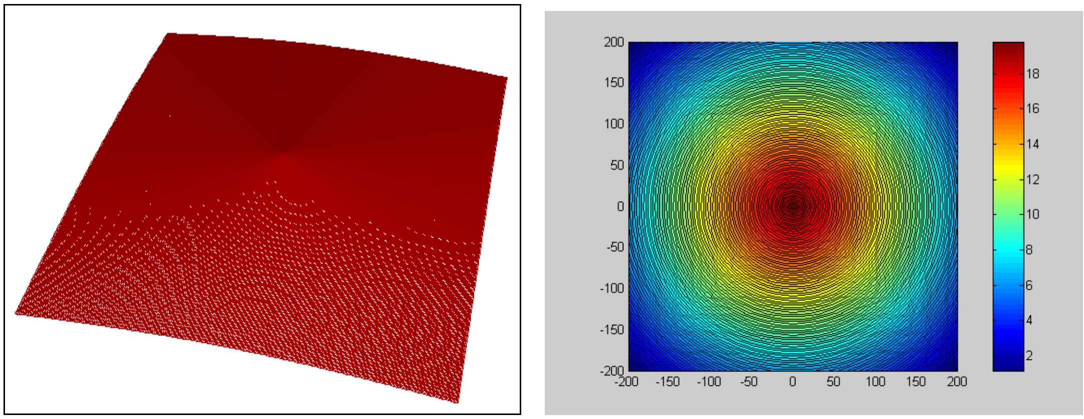
(a) (b)

Figure B.2 Maps of agricultural field 2 in India Head Research Farm: (a) triangular map of agricultural field 2, (b) topographic map of agricultural field 2.



(a) (b)

Figure B.3 Maps of a hole-like virtual field: (a) triangular map of a hole-like virtual field, (b) topographic map of a hole-like virtual field.



(c) (d)

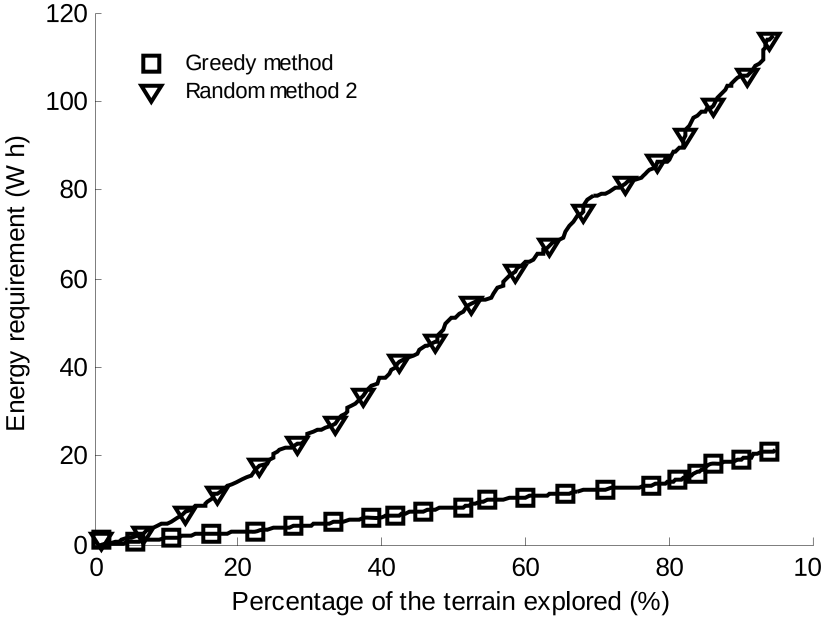
Figure B.4 Maps of a mountain-like virtual field: (a) triangular map of a mountain-like virtual field, (b) topographic map of a mountain-like virtual field.

APPENDIX C - RESULTS OF THE AUTONOMOUS CONSTRUCTION OF AGRICULTURAL FIELD MAPS BASED ON MINIMUM ENERGY COST FUNCTION

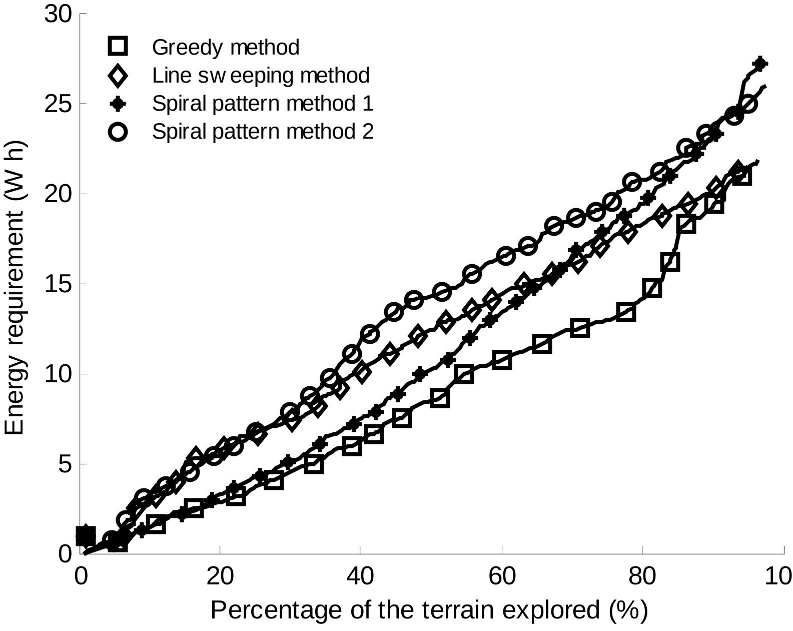
C.1 Figures generated for objective 2

(1) Figures C.1 – C.12 Results of the autonomous mapping of agricultural field 1 for objective 2,

(2) Figures C.13 – C.28 Results of the autonomous mapping of agricultural field 2 for objective 2.

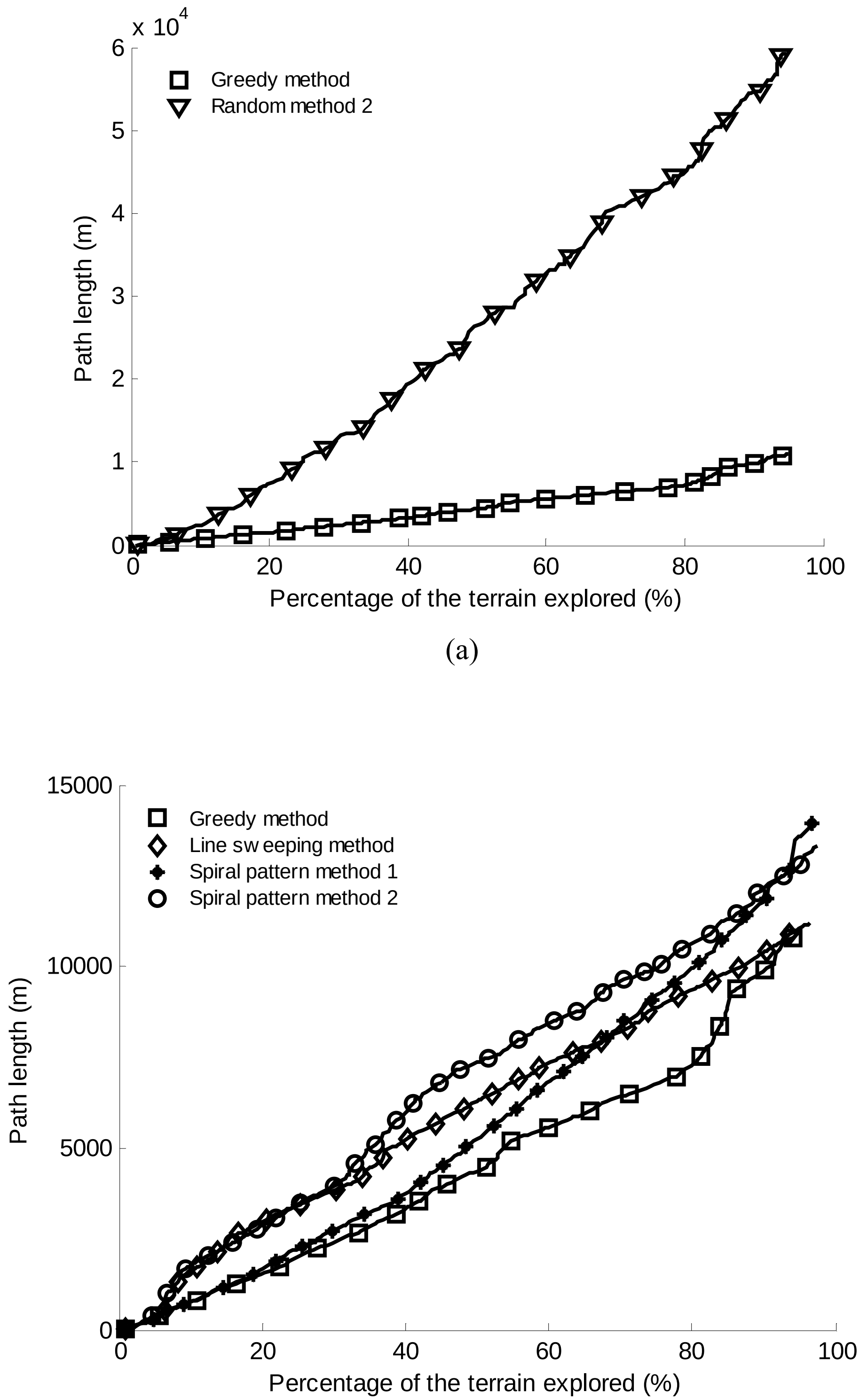


(a)



(b)

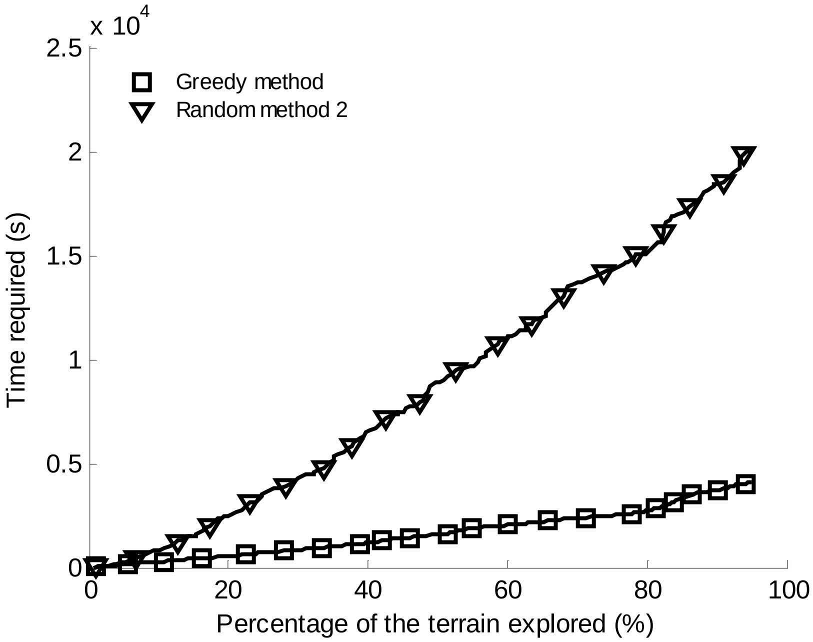
Figure C.1 Result of the autonomous mapping of agricultural field 1 for objective 2 (start B): energy requirement of the exploration as a function of fraction of explored terrain.



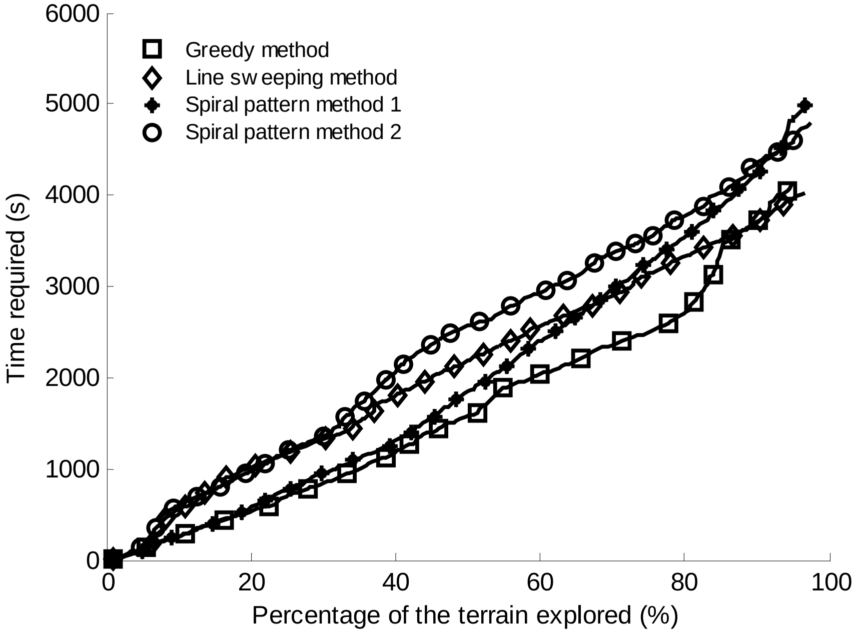
(a)

(b)

Figure C.2 Result of the autonomous mapping of agricultural field 1 for objective 2 (start B): path length traveled by the robot as a function of fraction of explored terrain.

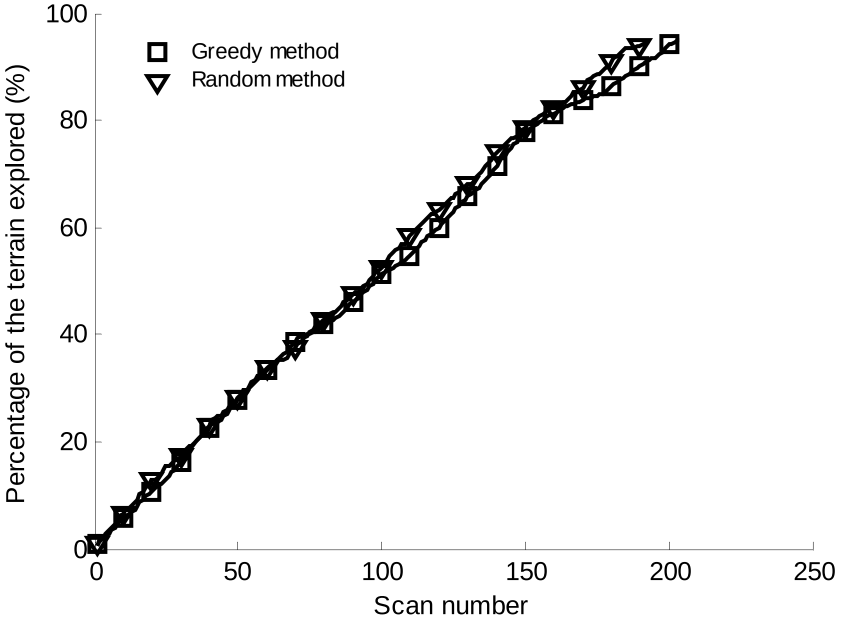


(a)

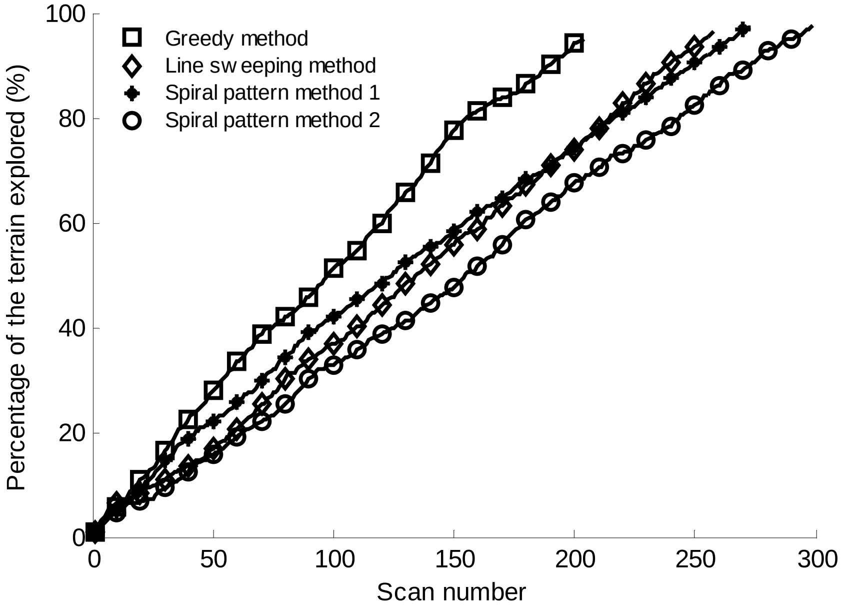


(b)

Figure C.3 Result of the autonomous mapping of agricultural field 1 for objective 2 (start B): time required of the exploration as a function of fraction of explored terrain.

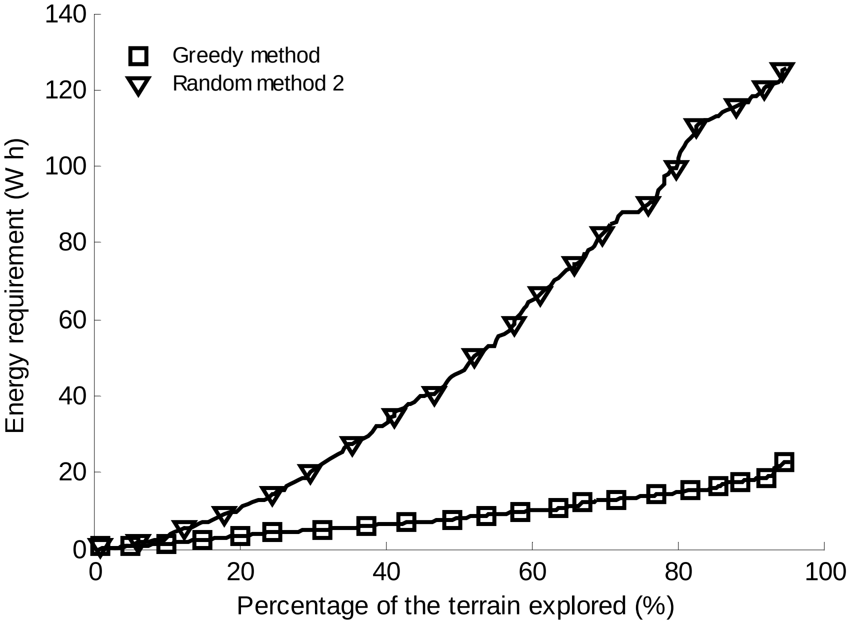


(a)

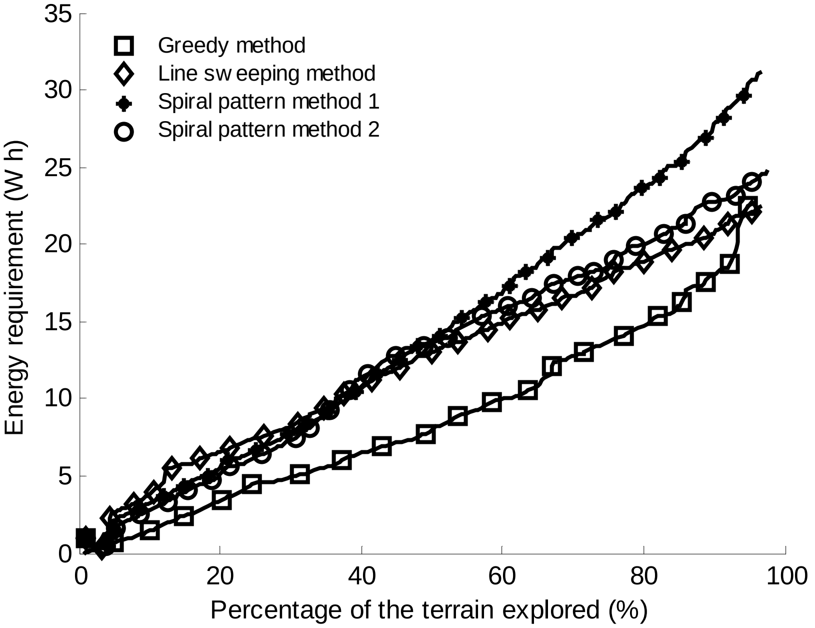


(b)

Figure C.4 Result of the autonomous mapping of agricultural field 1 for objective 2 (start B): the fraction of explored terrain as a function of scan number.

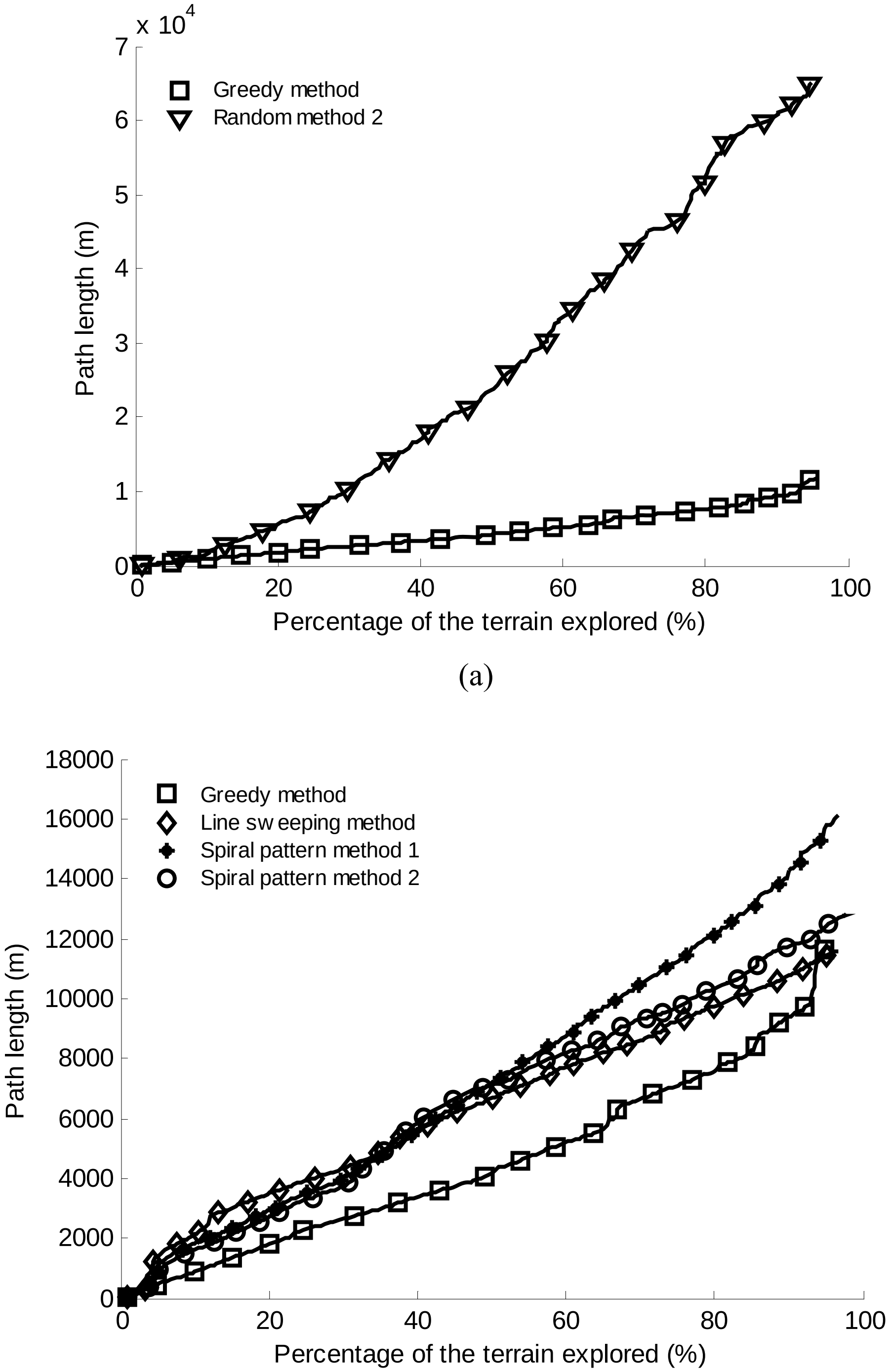


(a)



(b)

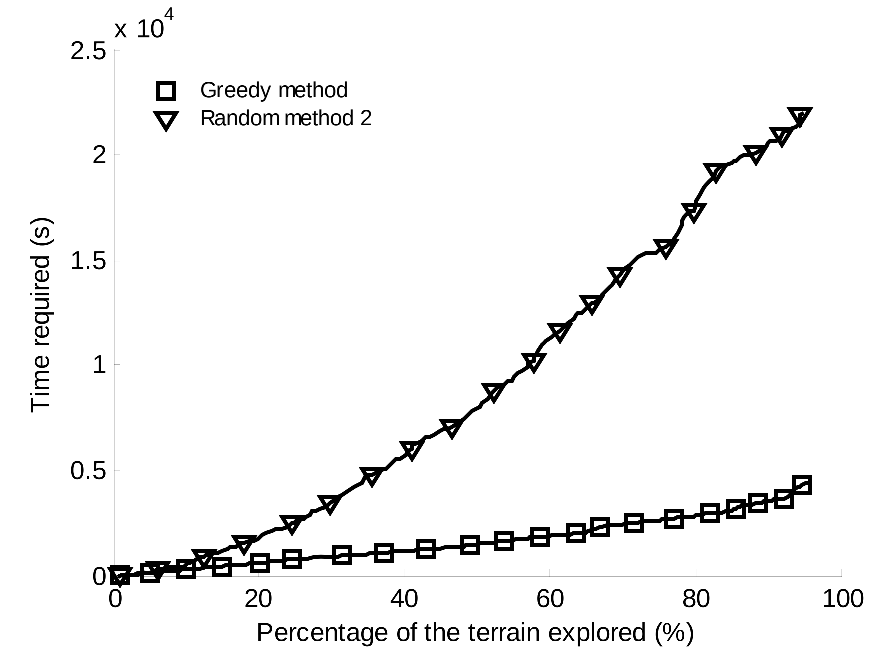
Figure C.5 Result of the autonomous mapping of agricultural field 1 for objective 2 (start C): energy requirement of the exploration as a function of fraction of explored terrain.



(a)

(b)

Figure C.6 Result of the autonomous mapping of agricultural field 1 for objective 2 (start C): path length traveled by the robot as a function of fraction of explored terrain.



(a)

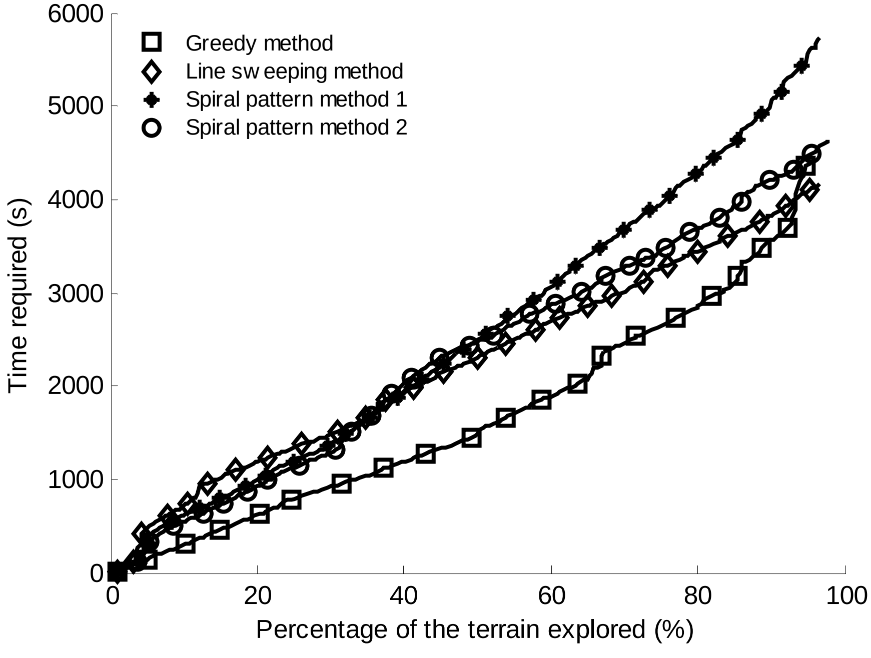
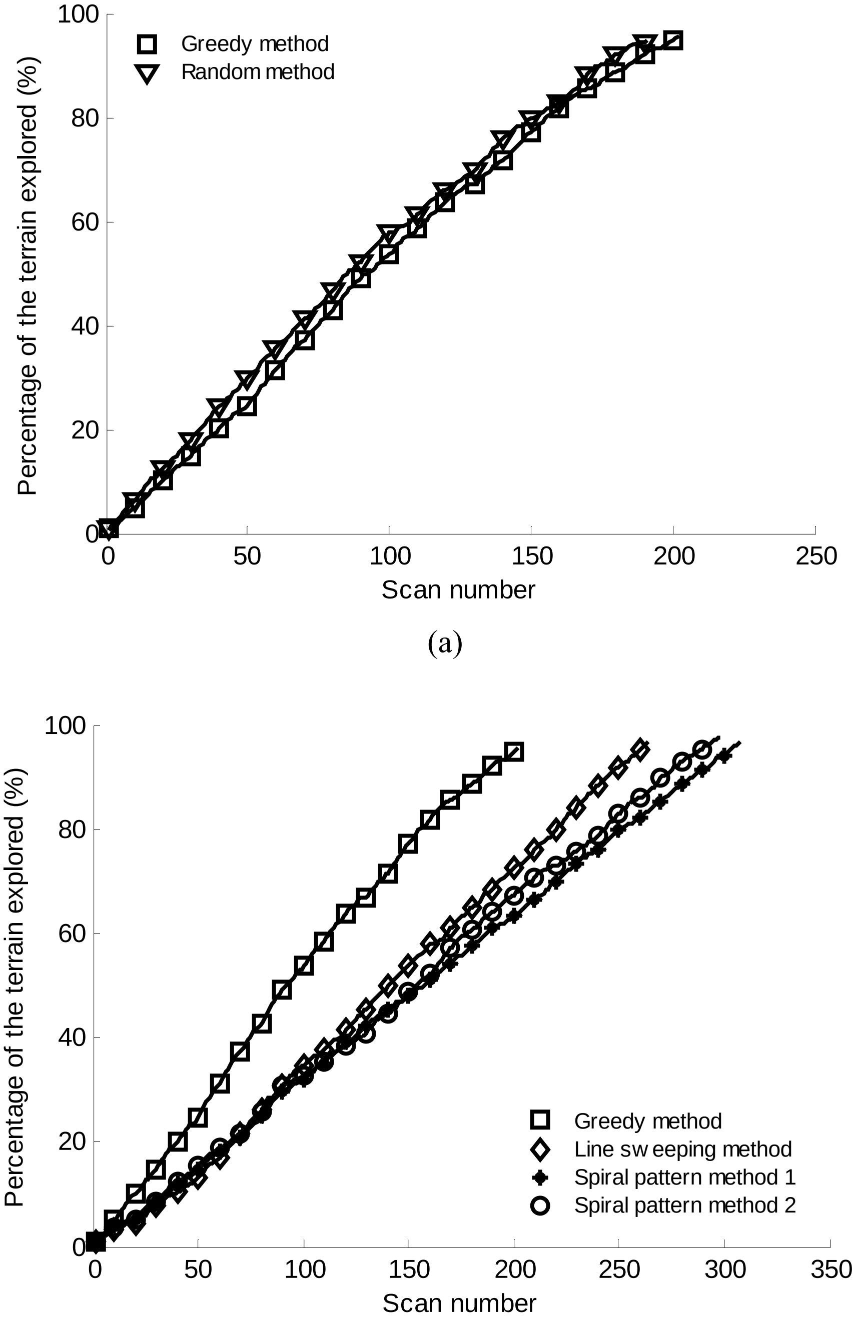


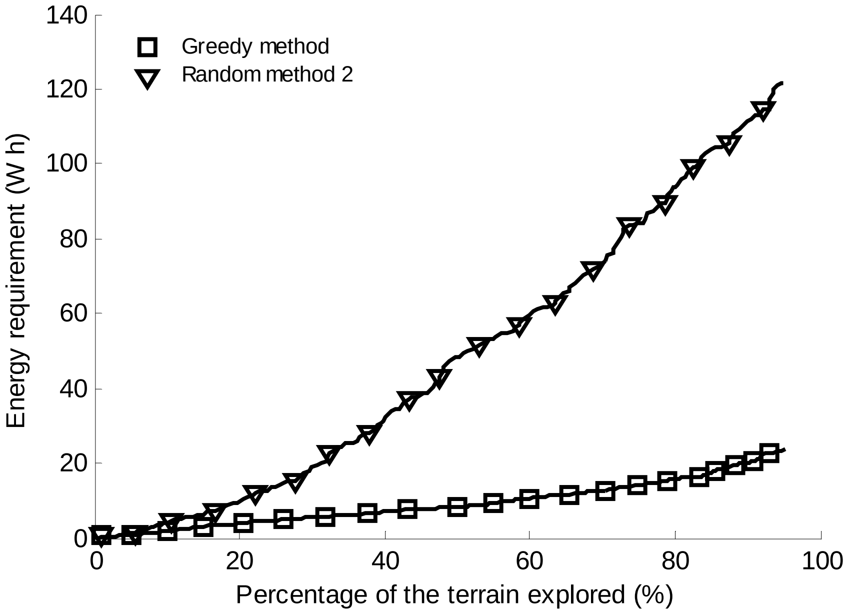
Figure C.7 Result of the autonomous mapping of agricultural field 1 for objective 2 (start C): time required of the exploration as a function of fraction of explored terrain.



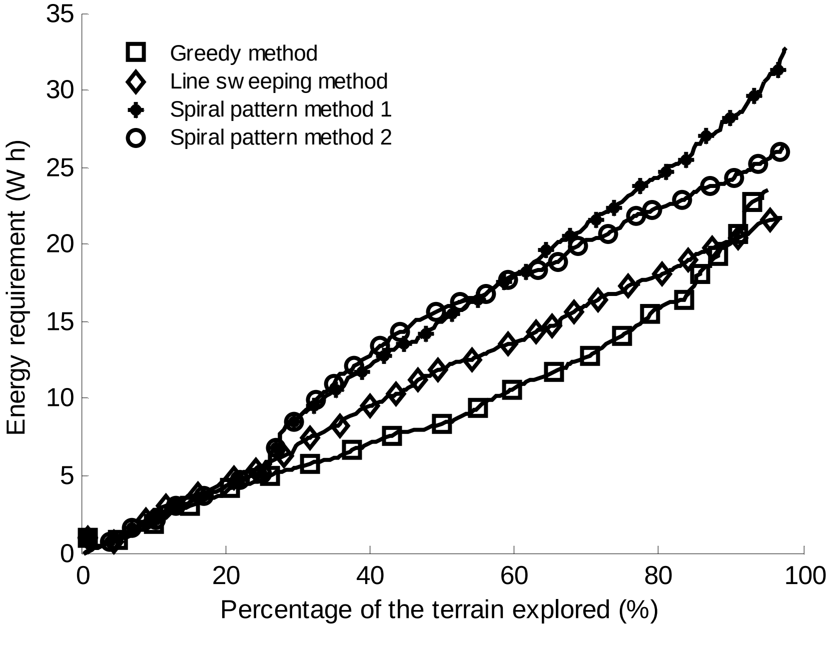
(a)

(b)

Figure C.8 Result of the autonomous mapping of agricultural field 1 for objective 2 (start C): the fraction of explored terrain as a function of scan number.

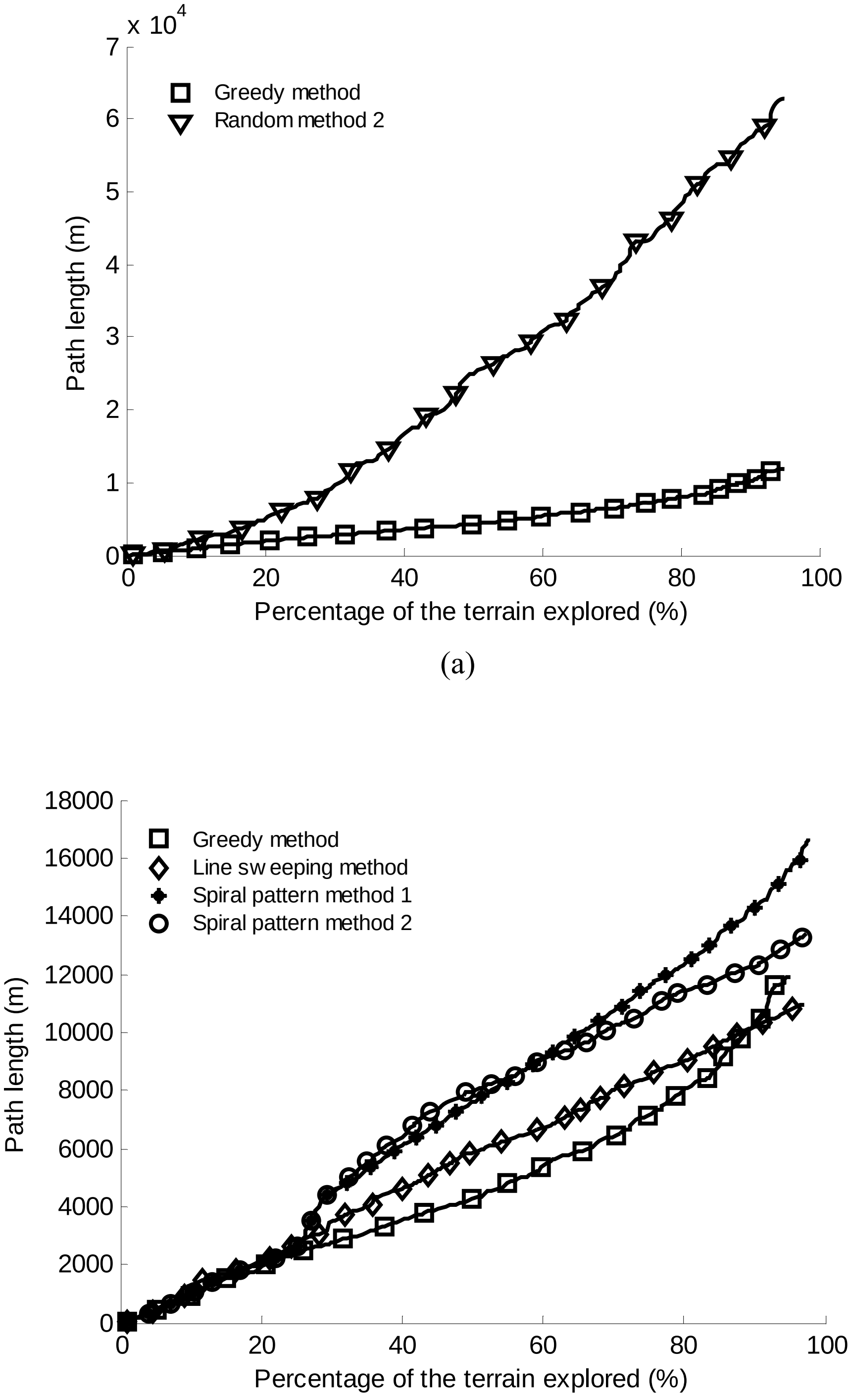


(a)



(b)

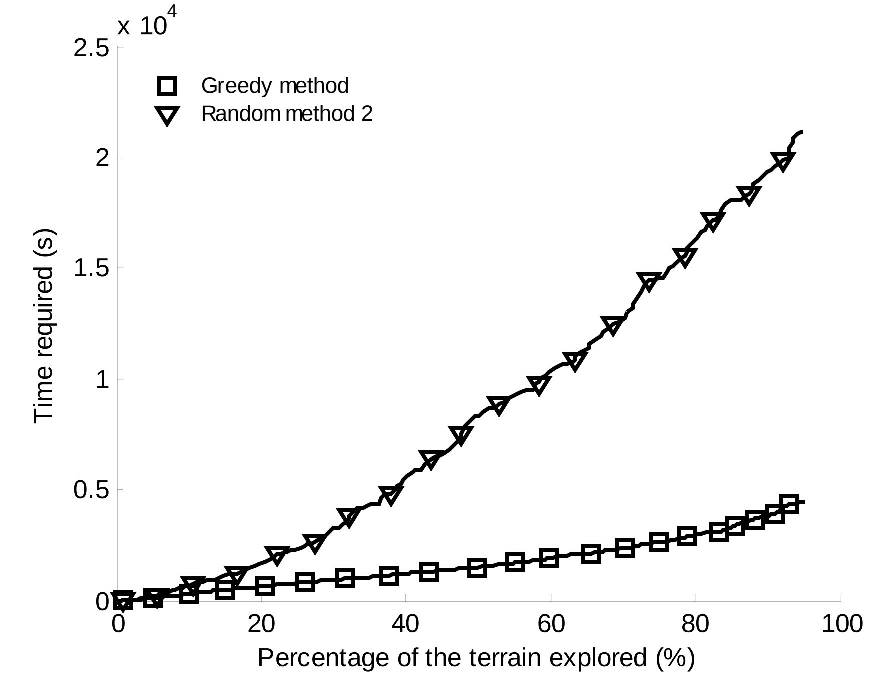
Figure C.9 Result of the autonomous mapping of agricultural field 1 for objective 2 (start D): energy requirement of the exploration as a function of fraction of explored terrain.



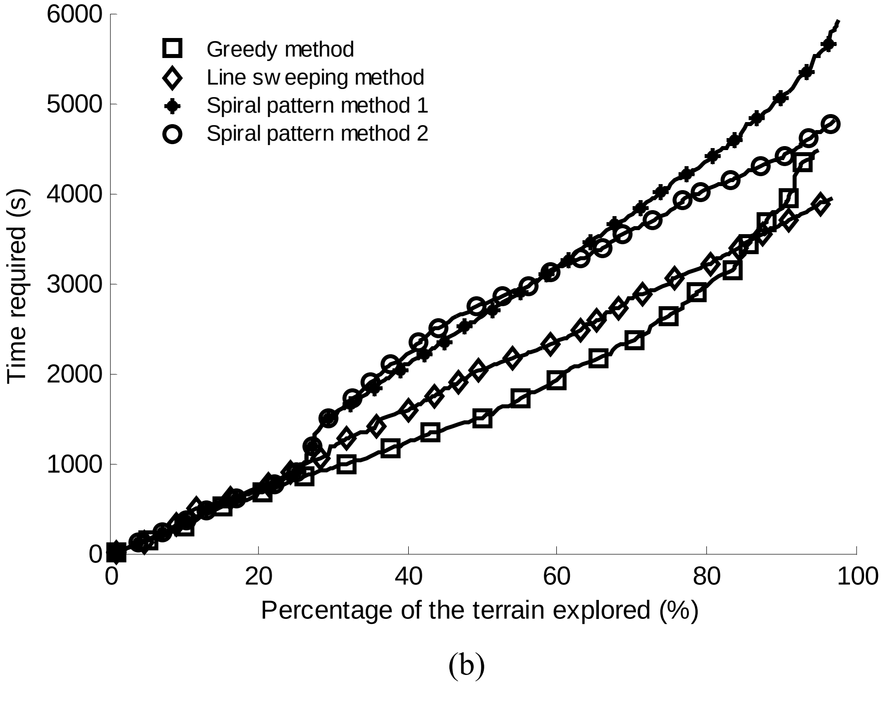
(a)

(b)

Figure C.10 Result of the autonomous mapping of agricultural field 1 for objective 2 (start D): path length traveled by the robot as a function of fraction of explored terrain.

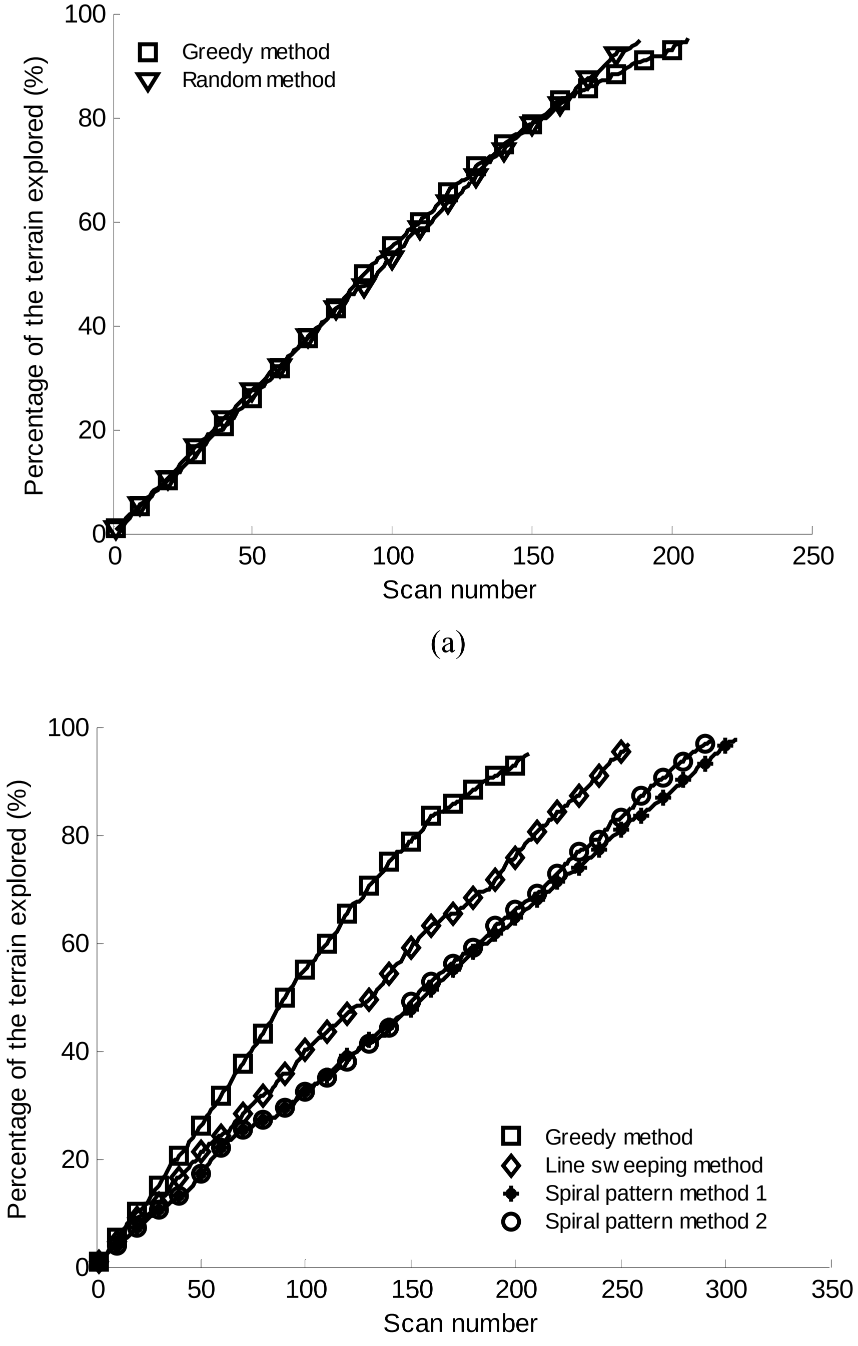


(a)



(b)

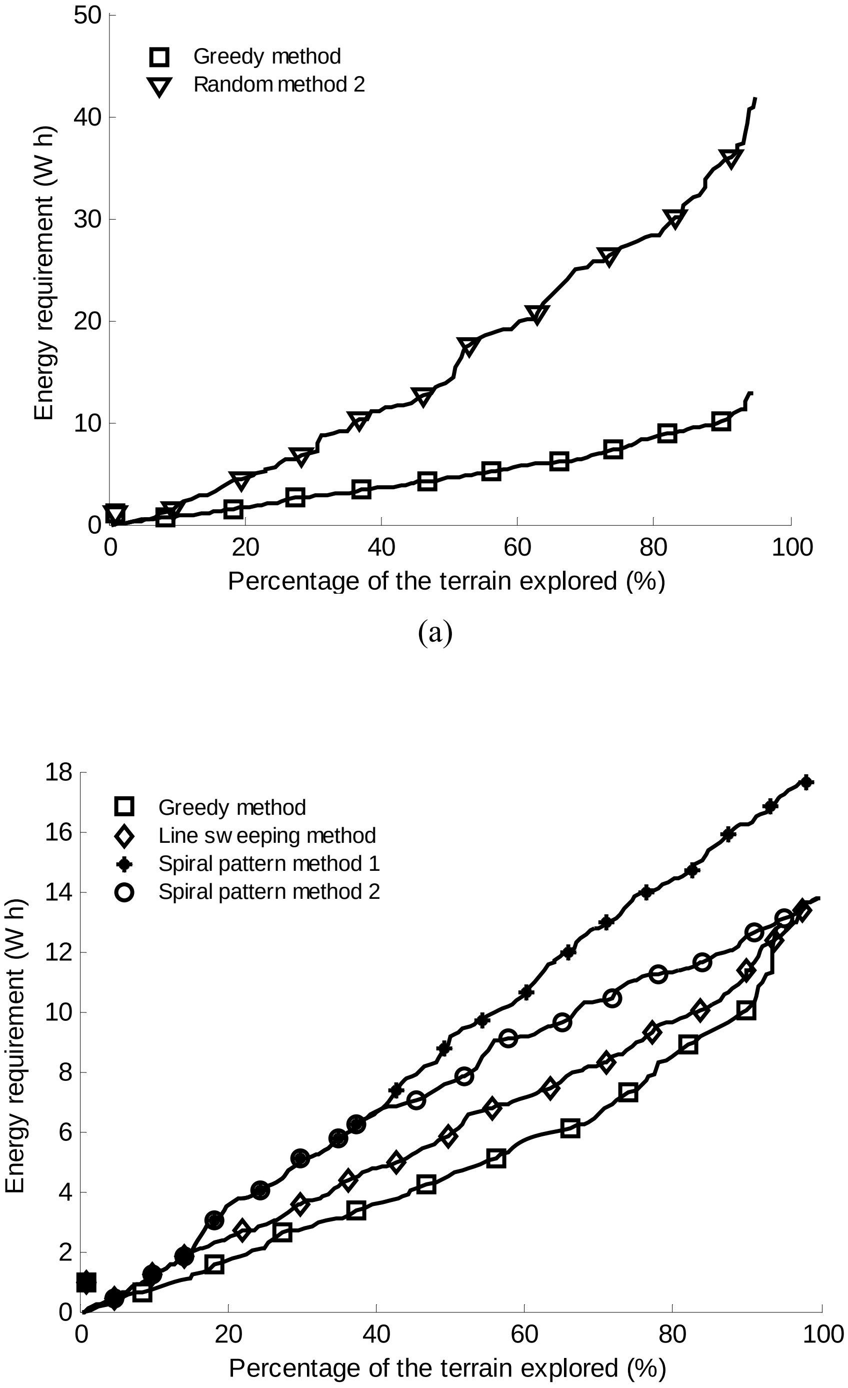
Figure C.11 Result of the autonomous mapping of agricultural field 1 for objective 2 (start D): time required of the exploration as a function of fraction of explored terrain.



(a)

(b)

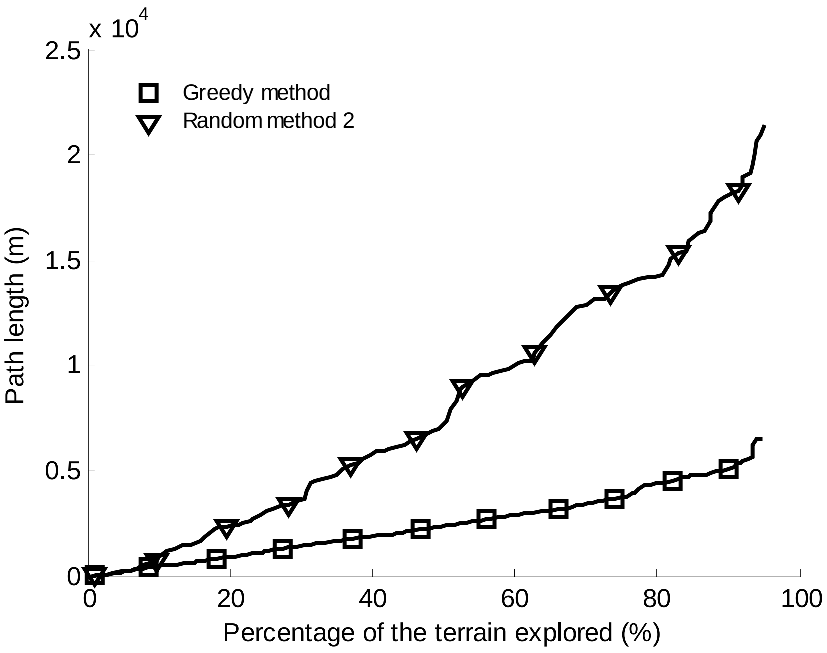
Figure C.12 Result of the autonomous mapping of agricultural field 1 for objective 2 (start D): the fraction of explored terrain as a function of scan number.



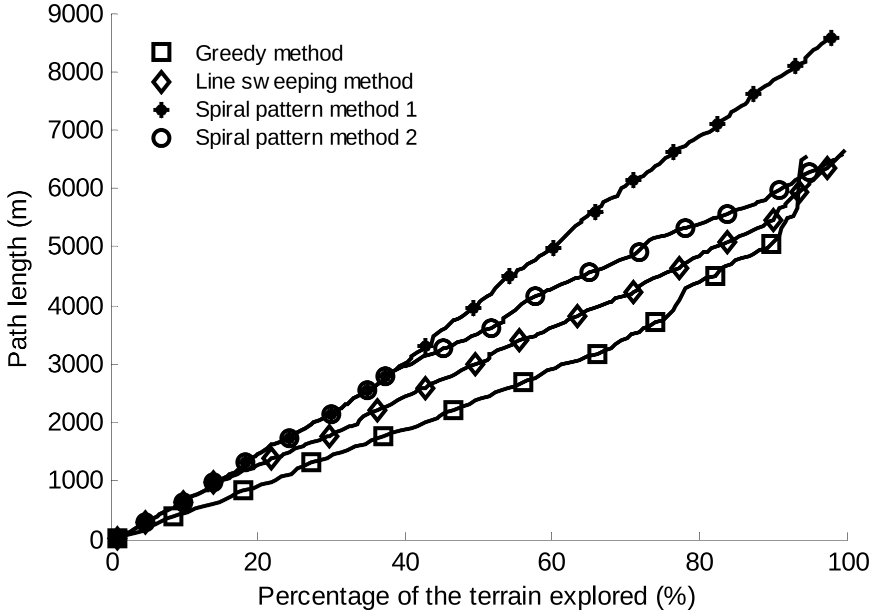
(a)

(b)

Figure C.13 Result of the autonomous mapping of agricultural field 2 for objective 2 (start A): energy requirement of the exploration as a function of fraction of explored terrain.

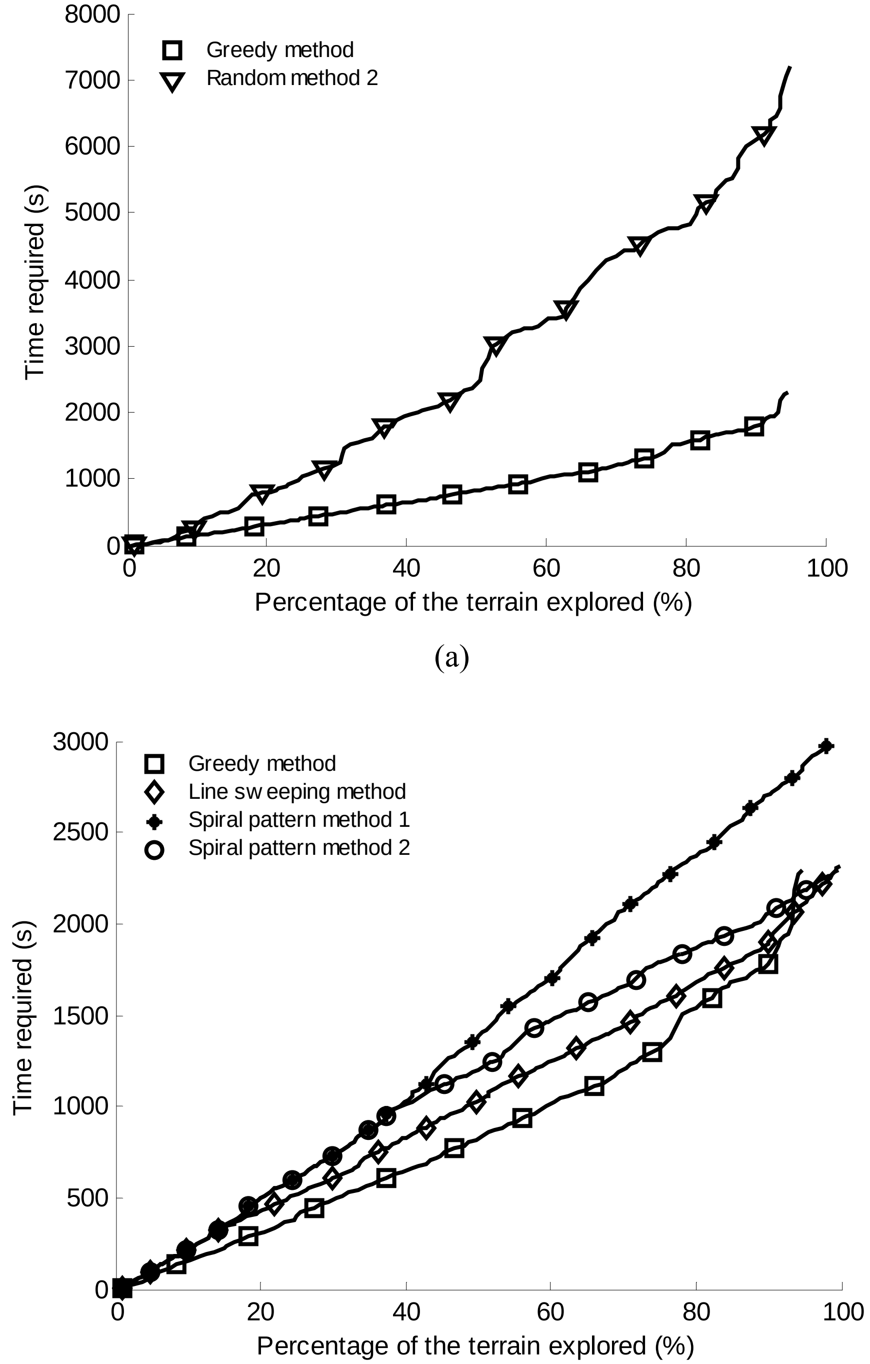


(a)



(b)

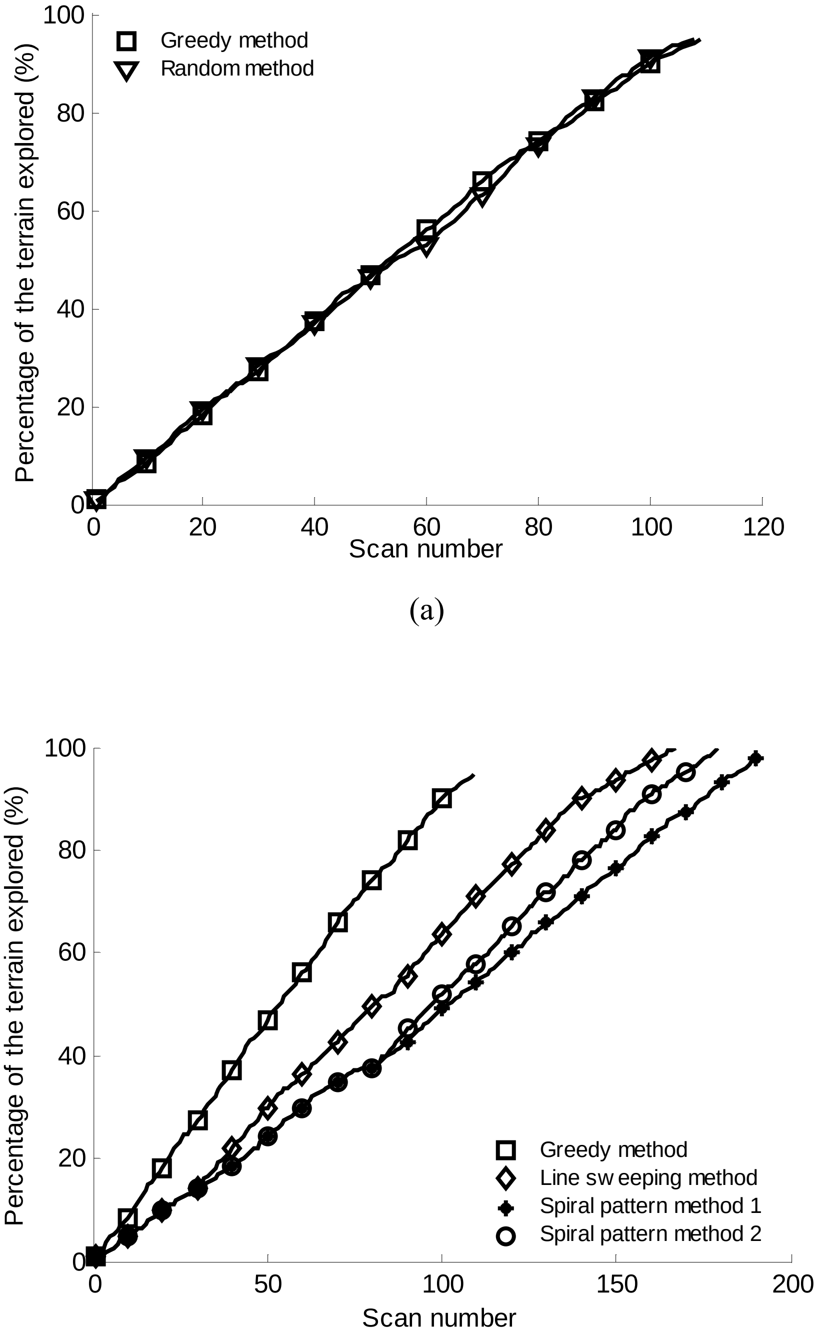
Figure C.14 Result of the autonomous mapping of agricultural field 2 for objective 2 (start A): path length traveled by the robot as a function of fraction of explored terrain.



(a)

(b)

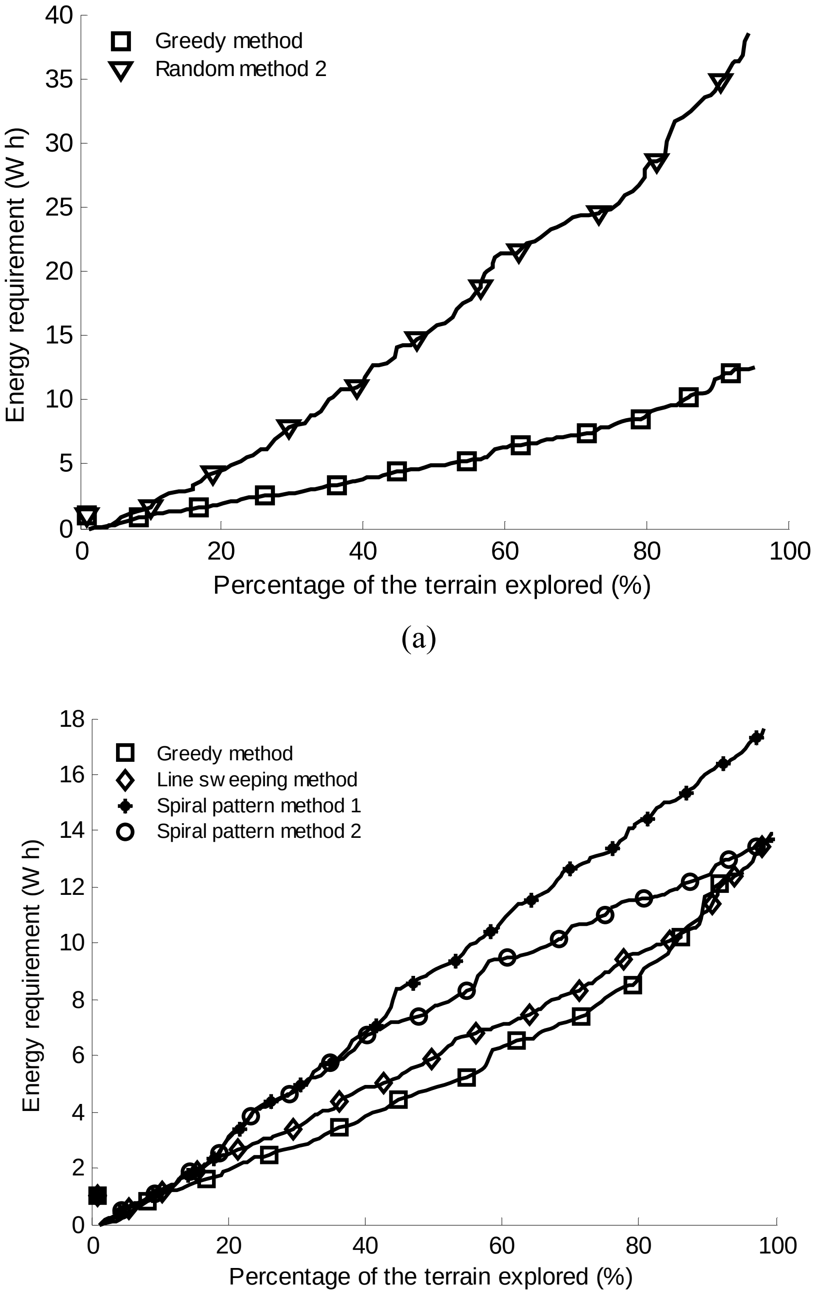
Figure C.15 Result of the autonomous mapping of agricultural field 2 for objective 2 (start A): time required of the exploration as a function of fraction of explored terrain.



(a)

(b)

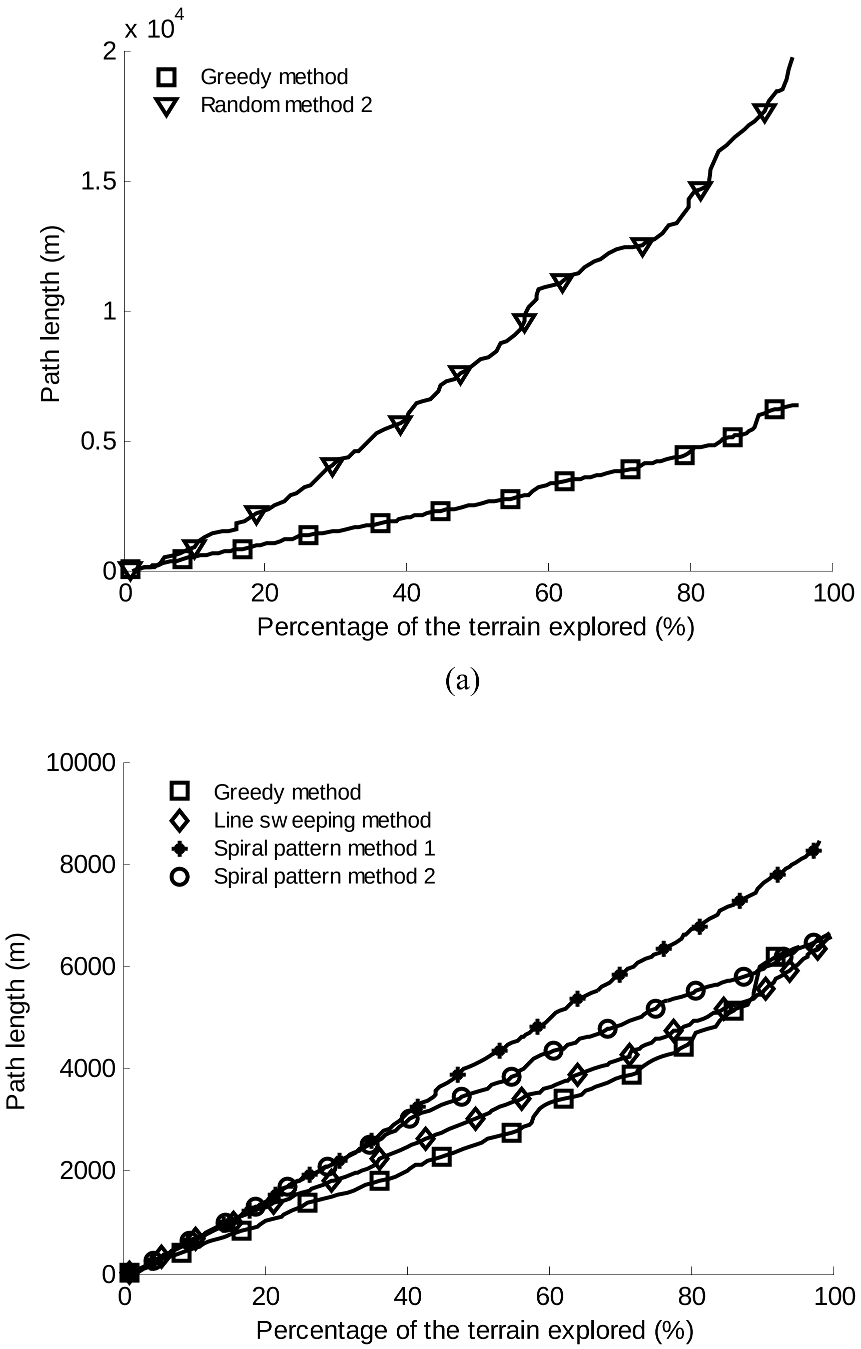
Figure C.16 Result of the autonomous mapping of agricultural field 2 for objective 2 (start A): the fraction of explored terrain as a function of scan number.



(a)

(b)

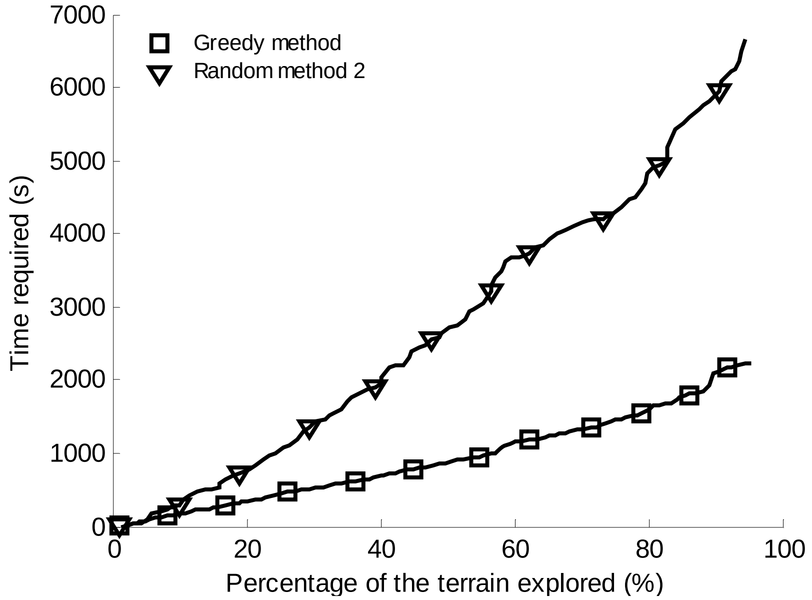
Figure C.17 Result of the autonomous mapping of agricultural field 2 for objective 2 (start B): energy requirement of the exploration as a function of fraction of explored terrain.



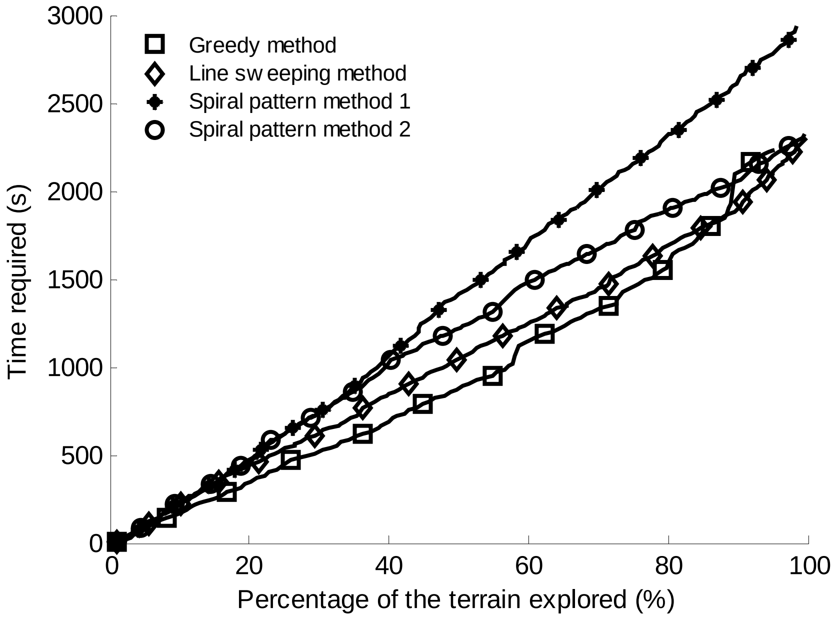
(a)

(b)

Figure C.18 Result of the autonomous mapping of agricultural field 2 for objective 2 (start location B): path length traveled by the robot as a function of fraction of explored terrain.

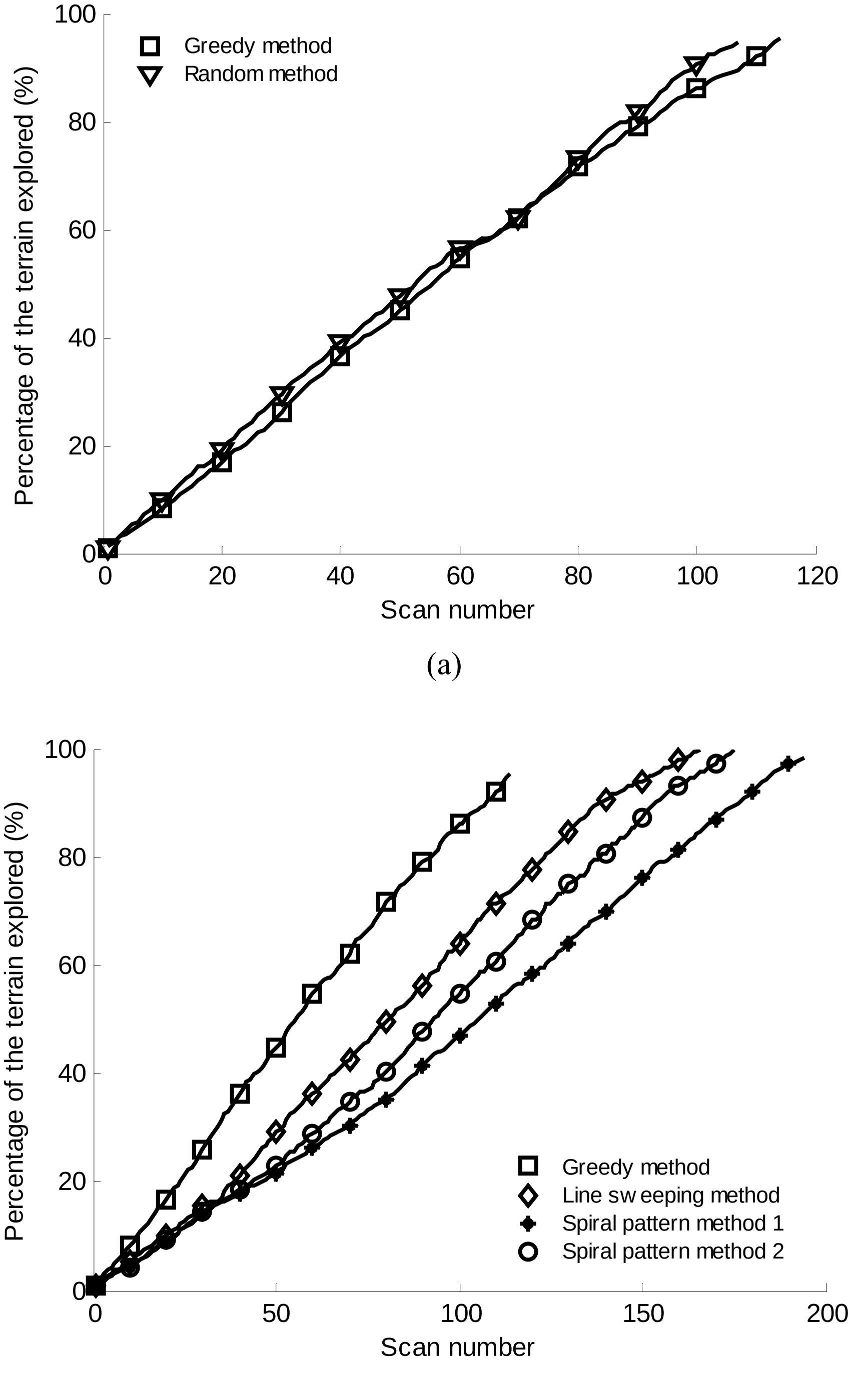


(a)



(b)

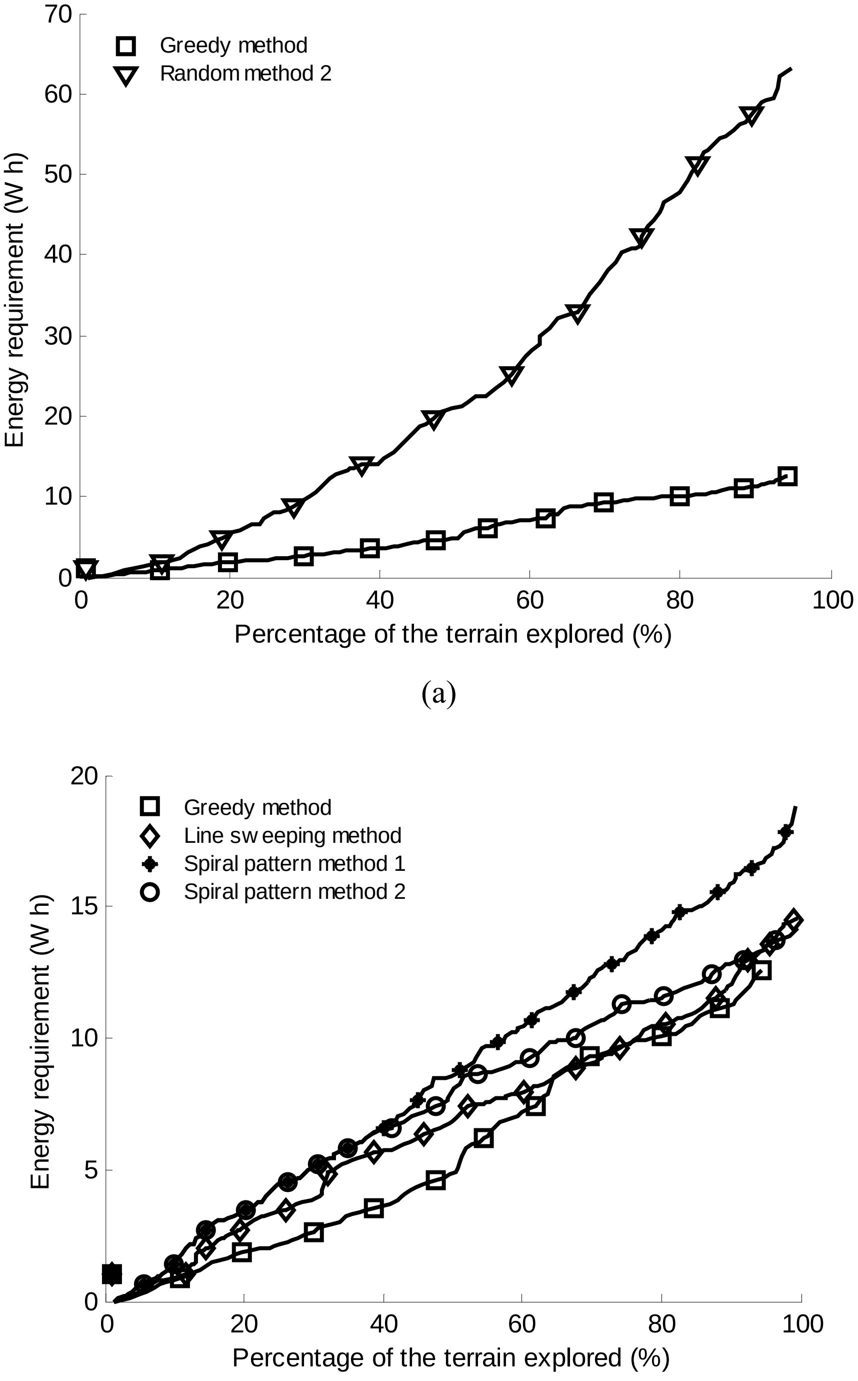
Figure C.19 Result of the autonomous mapping of agricultural field 2 for objective 2 (start B): time required of the exploration as a function of fraction of explored terrain.



(a)

(b)

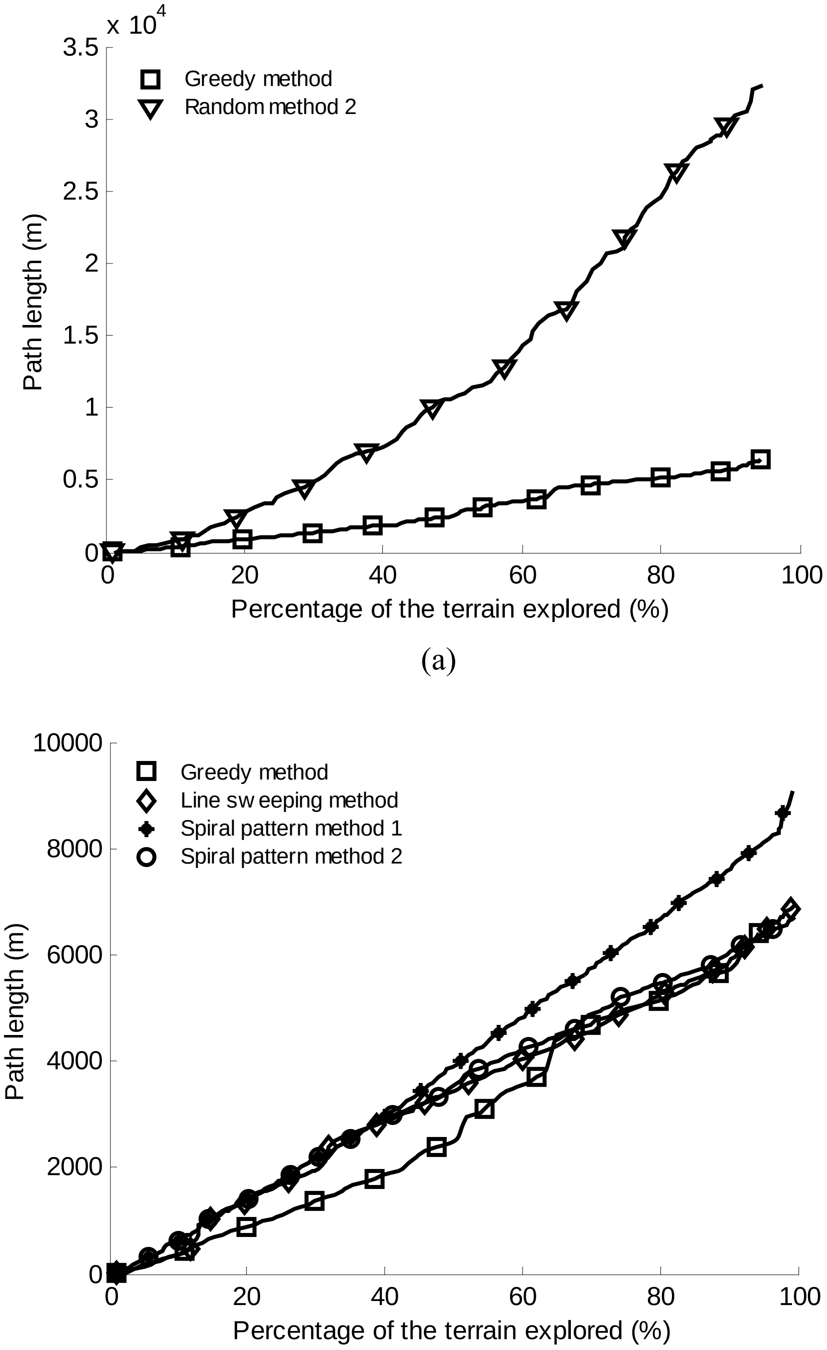
Figure C.20 Result of the autonomous mapping of agricultural field 2 for objective 2 (start location B): the fraction of explored terrain as a function of scan number.



(a)

(b)

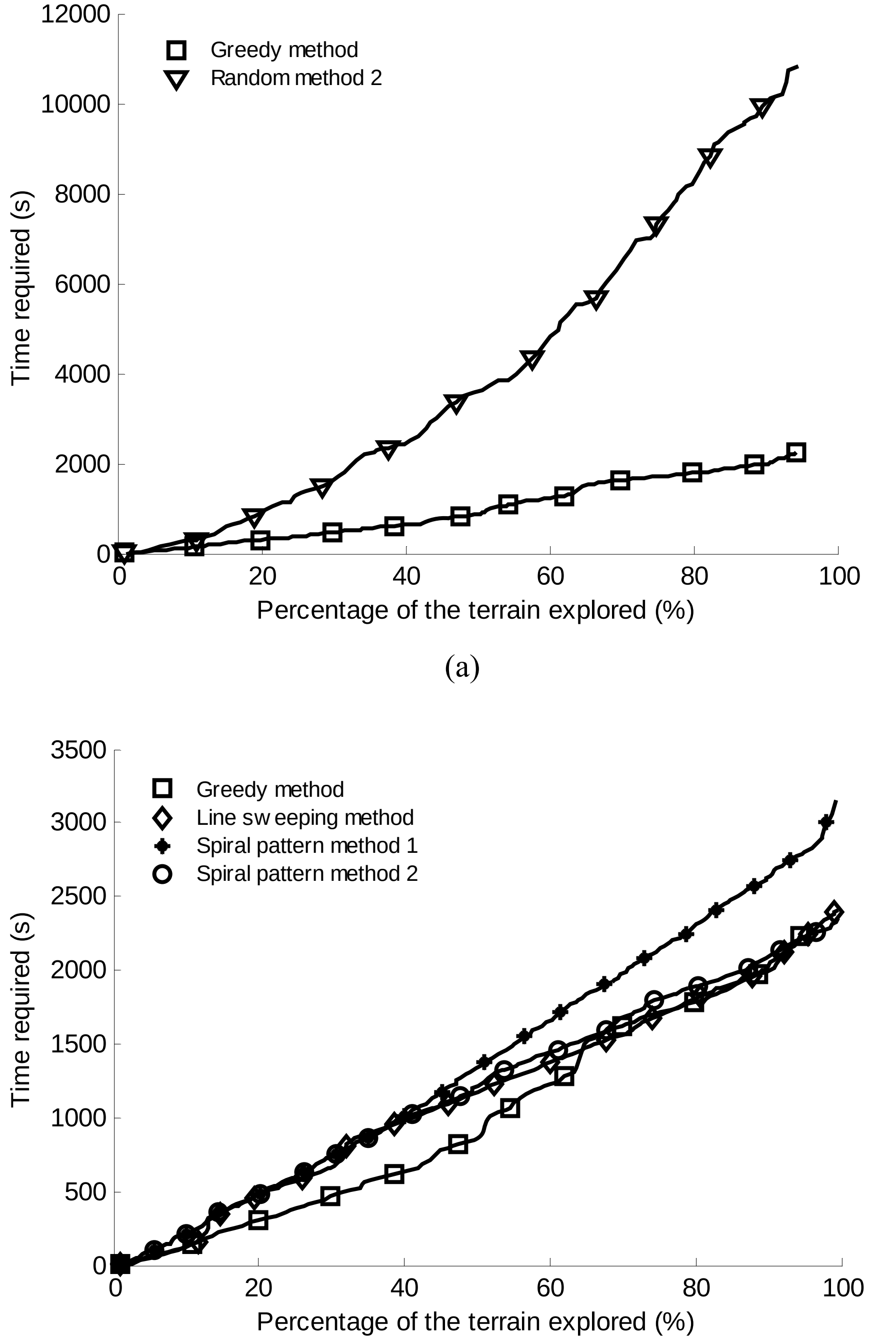
Figure C.21 Result of the autonomous mapping of agricultural field 2 for objective 2 (start C): energy requirement of the exploration as a function of fraction of explored terrain.



(a)

(b)

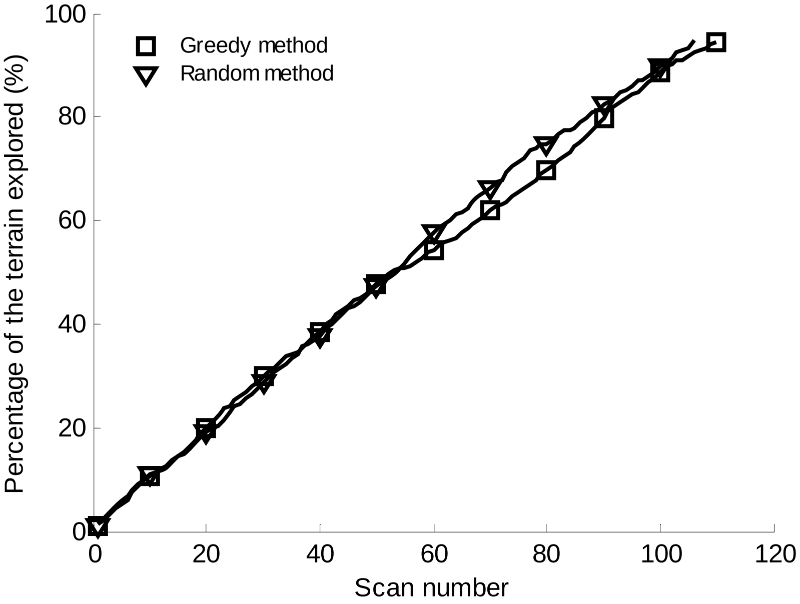
Figure C.22 Result of the autonomous mapping of agricultural field 2 for objective 2 (start C): path length traveled by the robot as a function of fraction of explored terrain.



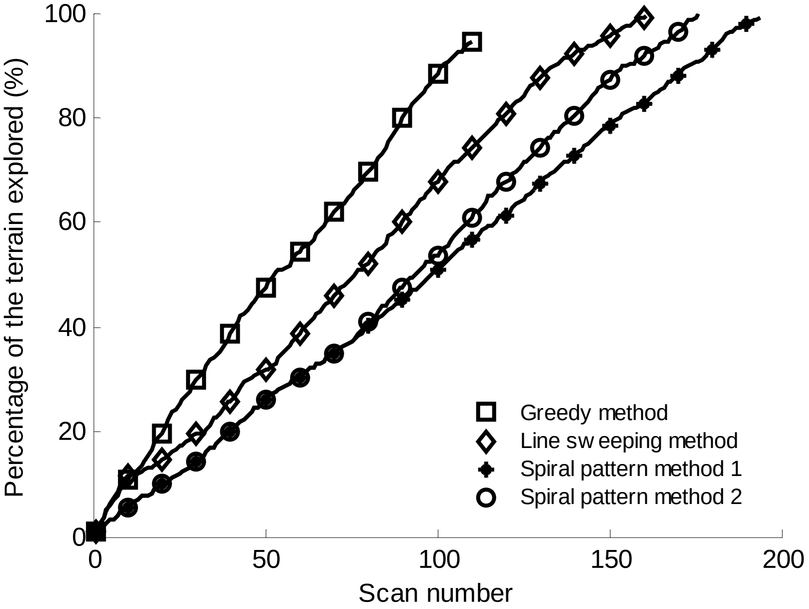
(a)

(b)

Figure C23. Result of the autonomous mapping of agricultural field 2 for objective 2 (start C): time required of the exploration as a function of fraction of explored terrain.

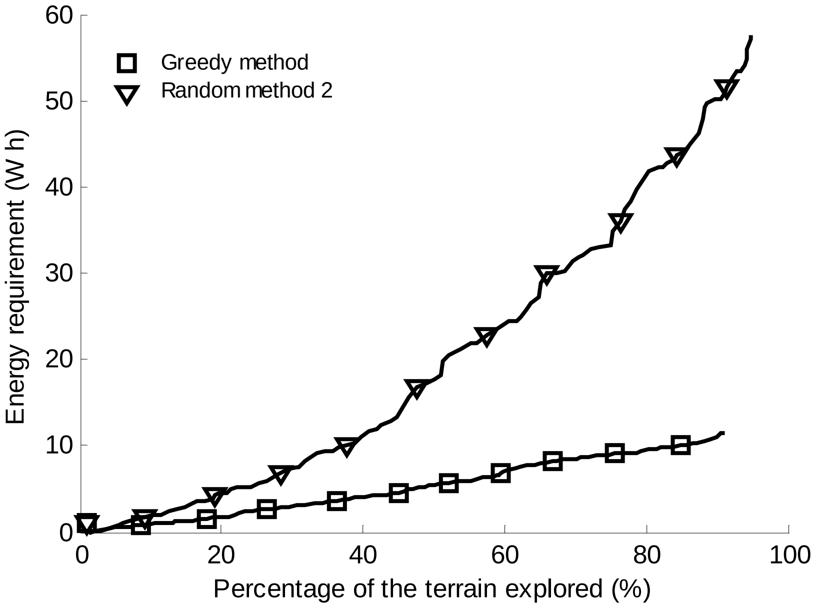


(a)

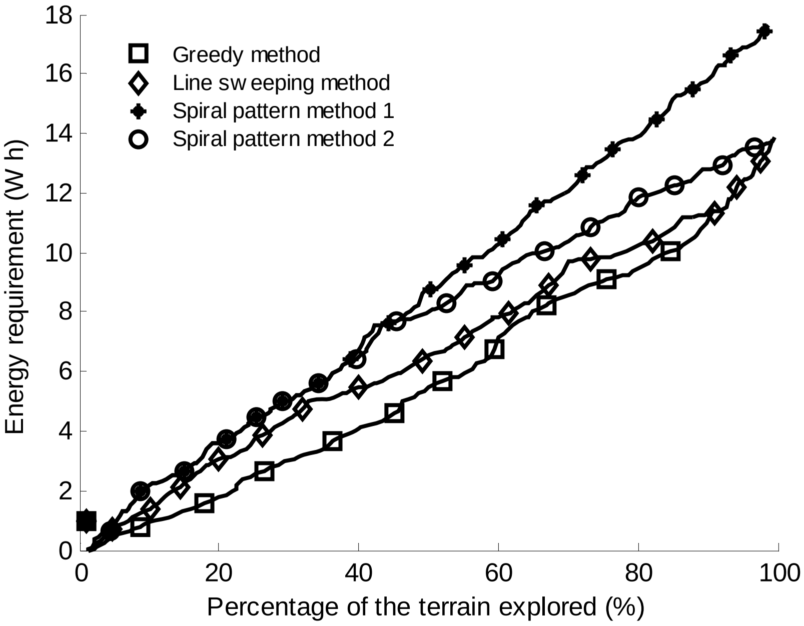


(b)

Figure C24. Result of the autonomous mapping of agricultural field 2 for objective 2 (start C): the fraction of explored terrain as a function of scan number.

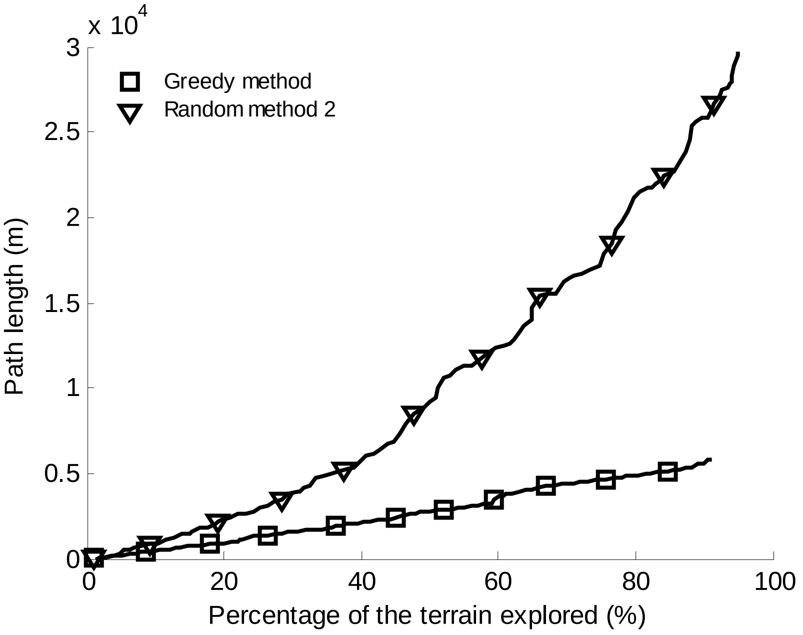


(a)

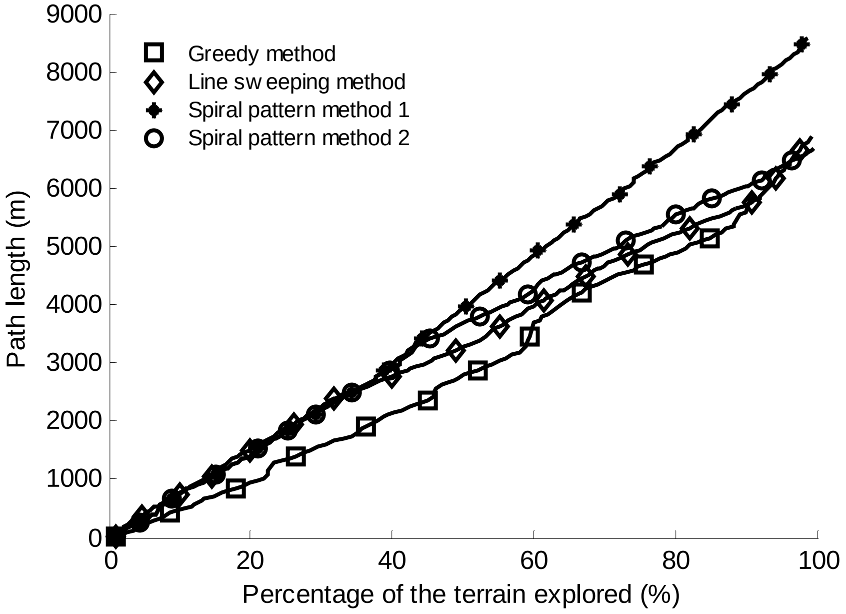


(b)

Figure C.25 Result of the autonomous mapping of agricultural field 2 for objective 2 (start D): energy requirement of the exploration as a function of fraction of explored terrain.

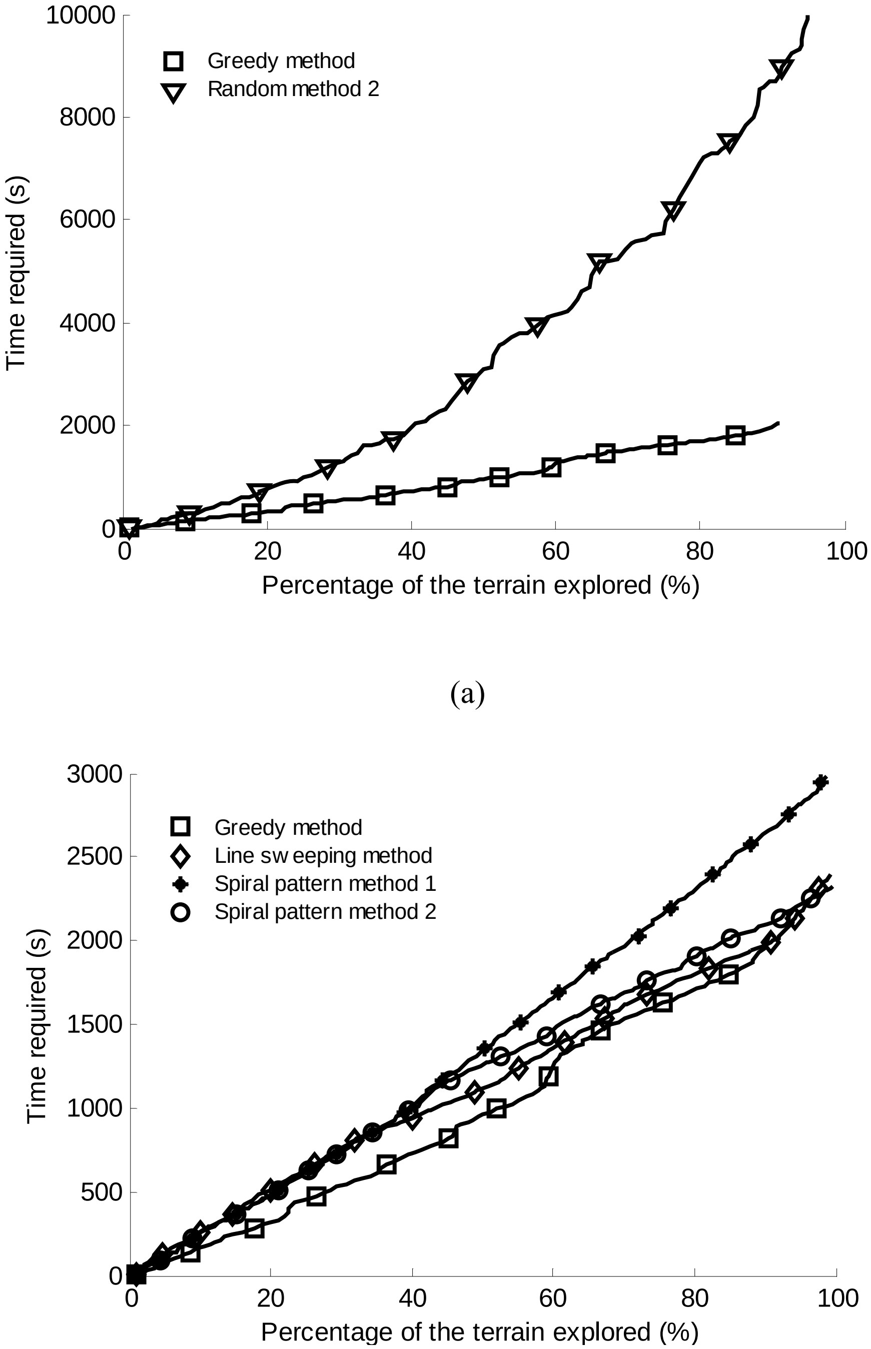


(a)



(b)

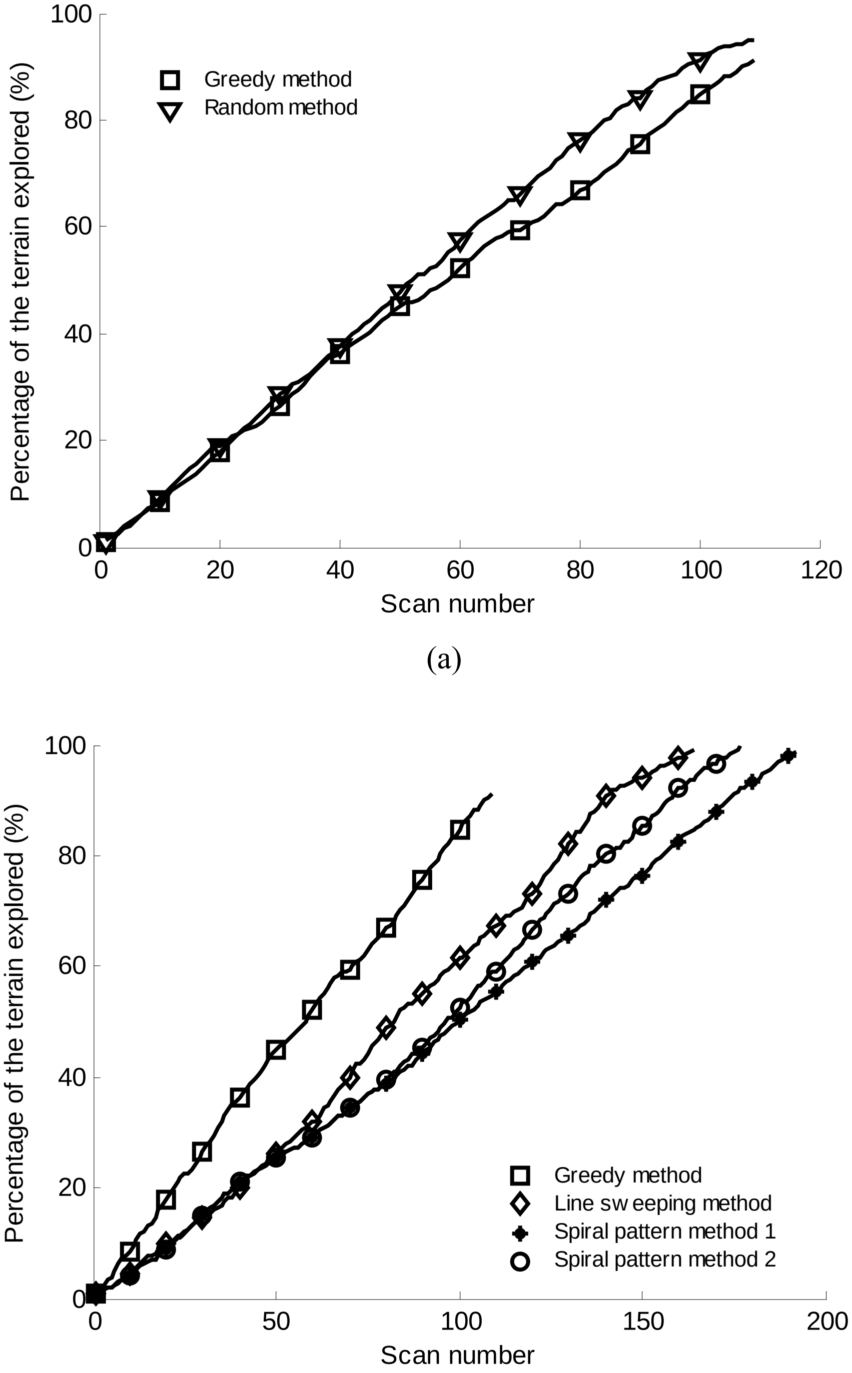
Figure C.26 Result of the autonomous mapping of agricultural field 2 for objective 2 (start D): path length traveled by the robot as a function of fraction of explored terrain.



(a)

(b)

Figure C27. Result of the autonomous mapping of agricultural field 2 for objective 2 (start D): time required of the exploration as a function of fraction of explored terrain.



(a)

(b)

Figure C.28 Result of the autonomous mapping of agricultural field 2 for objective 2 (start D): the fraction of explored terrain as a function of scan number.

C.2 Tables generated in objective 2

(1) Tables C.1– C.2: Results of the autonomous construction of agricultural field 1 map for objective 2,

(2) Tables C.3– C.4: Results of the autonomous construction of agricultural field 2 map for objective 2,

(3) Tables C.5 – C.6: Results of the autonomous construction of an ideal mountain map for objective 2,

(4) Tables C.7 – C.8: Results of the autonomous construction of an ideal hole map for objective 2,

(5) Tables C.9 – C.10: Results of the autonomous construction of an ideal slope map for objective 2.

Table C.1: Results for 90% of the exploration of the automatic mapping of agricultural field 1.

Start Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	19.7	10030	3738	193
	Line sweeping method	18.6	9530	3419	238
	Spiral pattern #1	26.0	13171	4695	287
	Spiral pattern #2	22.3	11307	4065	274
	Random	111.0	57124	19289	180
B	Greedy method	19.1	9863	3698	189
	Line sweeping method	20.1	10351	3695	238
	Spiral pattern #1	23.1	11771	4208	248
	Spiral pattern #2	23.6	12125	4337	272
	Random	105.2	54442	18385	178
C	Greedy method	18.0	9381	3560	184
	Line sweeping	20.6	10717	3817	243
	Spiral pattern #1	27.2	14030	4981	284
	Spiral pattern #2	22.7	11722	4203	270
	Random	117.6	60752	20509	174
D	Greedy method	20.0	10164	3817	185
	Line sweeping method	20.1	10155	3635	236
	Spiral pattern #1	28.0	14222	5035	279
	Spiral pattern #2	24.0	12208	4361	267
	Random	111.4	57402	19359	176

Table C.2: Results for 50% of the exploration of the automatic mapping of agricultural field 1.

Start Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	9.5	4870	1730	99
	Line sweeping method	10.8	5413	1889	133
	Spiral pattern #1	13.2	6616	2307	154
	Spiral pattern #2	12.9	6547	2279	150
	Random	50.1	25606	8609	93
B	Greedy method	8.4	4333	1564	97
	Line sweeping method	12.4	6282	2182	134
	Spiral pattern #1	10.1	5257	1838	124
	Spiral pattern #2	14.2	7326	2543	154
	Random	51.1	26465	8896	95
C	Greedy method	7.9	4090	1467	91
	Line sweeping method	12.8	6602	2285	139
	Spiral pattern #1	13.7	7131	2481	156
	Spiral pattern #2	13.6	7101	2472	154
	Random	45.8	23664	7957	85
D	Greedy method	8.3	4239	1509	90
	Line sweeping method	11.9	5852	2040	131
	Spiral pattern #1	15.0	7572	2631	156
	Spiral pattern #2	15.8	7969	2760	152
	Random	48.2	24842	8346	94

Table C.3: Results of 90% exploration of agricultural field 2 for objective 2.

Starting Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	10.0	5046	1780	100
	Line sweeping method	11.1	5408	1876	138
	Spiral pattern #1	16.2	7795	2694	174
	Spiral pattern #2	12.5	5902	2050	158
	Random	35.3	18010	6064	98
B	Greedy method	11.7	5996	2095	107
	Line sweeping method	11.2	5459	1895	138
	Spiral pattern #1	16.0	7545	2611	176
	Spiral pattern #2	12.4	5900	2047	153
	Random	33.9	17423	5869	99
C	Greedy method	11.2	5706	1984	101
	Line sweeping method	12.0	5796	2006	133
	Spiral pattern #1	15.8	7577	2625	173
	Spiral pattern #2	12.8	6013	2088	155
	Random	57.5	29536	9912	100
D	Greedy method	11.0	5579	1952	107
	Line sweeping method	11.2	5702	1971	139
	Spiral pattern #1	15.8	7598	2630	173
	Spiral pattern #2	12.8	6003	2082	156
	Random	50.1	25839	8678	97

Table C.4: Results of 50% exploration of agricultural field 2 for objective 2.

Starting Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	4.5	2368	820	53
	Line sweeping method	5.8	2986	1022	80
	Spiral pattern #1	8.9	3984	1363	101
	Spiral pattern #2	7.6	3507	1202	97
	Random	13.7	7030	2365	54
B	Greedy method	4.8	2528	871	55
	Line sweeping method	5.9	3023	1035	80
	Spiral pattern #1	9.0	4124	1411	105
	Spiral pattern #2	7.6	3532	1208	92
	Random	15.2	7833	2632	52
C	Greedy method	4.8	2477	854	52
	Line sweeping method	6.7	3386	1155	75
	Spiral pattern #1	8.5	3894	1334	98
	Spiral pattern #2	7.7	3488	1194	93
	Random	20.9	10606	3557	53
D	Greedy method	5.4	2744	946	57
	Line sweeping method	6.4	3229	1102	81
	Spiral pattern #1	8.8	3891	1332	99
	Spiral pattern #2	8.0	3662	1253	96
	Random	17.2	8882	2982	51

Table C.5: Results of 90% exploration of an ideal mountain for objective 2.

Starting Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	7.5	3088	1059	57
	Line sweeping method	8.1	3679	1259	90
	Spiral pattern #1	9.1	3867	1322	93
	Spiral pattern #2	8.4	3373	1154	89
	Random	16.7	7720	2591	47
B	Greedy method	7.8	3529	1207	59
	Line sweeping method	8.1	3742	1297	80
	Spiral pattern #1	8.8	3840	1312	90
	Spiral pattern #2	7.9	3335	1142	88
	Random	18.6	8952	3002	45
C	Greedy method	7.5	3370	1153	58
	Line sweeping method	8.3	3878	1326	87
	Spiral pattern #1	11.9	5356	1817	92
	Spiral pattern #2	9.7	4138	1409	89
	Random	17.7	8551	2869	49
D	Greedy method	8.1	3543	1209	58
	Line sweeping method	8.4	3732	1272	81
	Spiral pattern #1	9.1	3923	1339	90
	Spiral pattern #2	9.3	3902	1331	96
	Random	18.7	8657	2905	48

Table C.6: Results of 50% exploration of an ideal mountain for objective 2.

Starting Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	3.1	1273	433	26
	Line sweeping method	4.4	1866	635	45
	Spiral pattern #1	4.8	1966	670	56
	Spiral pattern #2	4.9	2031	692	60
	Random	5.5	2135	719	24
B	Greedy method	3.0	1300	443	27
	Line sweeping method	4.6	1917	659	46
	Spiral pattern #1	4.2	1855	632	53
	Spiral pattern #2	4.3	1911	651	56
	Random	7.0	3201	1075	23
C	Greedy method	2.9	1271	433	27
	Line sweeping method	4.4	2048	696	42
	Spiral pattern #1	6.7	3044	1029	56
	Spiral pattern #2	5.4	2507	849	52
	Random	6.4	3067	1029	24
D	Greedy method	4.0	1684	573	32
	Line sweeping method	5.2	1936	658	46
	Spiral pattern #1	4.8	2015	686	54
	Spiral pattern #2	5.4	2415	820	62
	Random	6.0	2454	825	24

Table C.7: Results of 90% exploration of an ideal hole for objective 2.

Starting Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	8.5	4288	1463	61
	Line sweeping method	7.4	3330	1137	73
	Spiral pattern #1	7.5	3731	1274	89
	Spiral pattern #2	6.4	3239	1109	87
	Random	20.3	9991	3350	49
B	Greedy method	7.1	3458	1185	58
	Line sweeping method	7.6	3275	1120	74
	Spiral pattern #1	7.4	3603	1229	87
	Spiral pattern #2	6.9	3328	1137	82
	Random	18.4	8780	2947	50
C	Greedy method	8.0	3958	1353	59
	Line sweeping method	7.9	3461	1181	72
	Spiral pattern #1	7.7	3606	1234	91
	Spiral pattern #2	6.5	3230	1104	80
	Random	23.4	11188	3749	49
D	Greedy method	8.4	4150	1416	60
	Line sweeping method	7.0	3310	1129	70
	Spiral pattern #1	7.5	3670	1255	91
	Spiral pattern #2	6.9	3407	1164	82
	Random	19.4	9426	3161	49

Table C.8: Results of 50% exploration of an ideal hole for objective 2.

Starting Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	2.8	1621	552	27
	Line sweeping method	3.5	1748	593	39
	Spiral pattern #1	3.6	1652	561	48
	Spiral pattern #2	3.9	1712	583	52
	Random	6.8	3527	1183	24
B	Greedy method	2.4	1320	451	25
	Line sweeping method	3.5	1684	572	39
	Spiral pattern #1	4.0	1837	624	53
	Spiral pattern #2	3.8	1655	563	48
	Random	6.5	3338	1120	26
C	Greedy method	3.1	1639	557	26
	Line sweeping method	4.0	1834	623	37
	Spiral pattern #1	3.8	1693	576	50
	Spiral pattern #2	3.8	1693	576	49
	Random	8.9	4399	1474	24
D	Greedy method	2.9	1676	569	26
	Line sweeping method	3.2	1812	615	39
	Spiral pattern #1	3.7	1683	573	49
	Spiral pattern #2	3.7	1683	573	48
	Random	6.0	2958	993	24

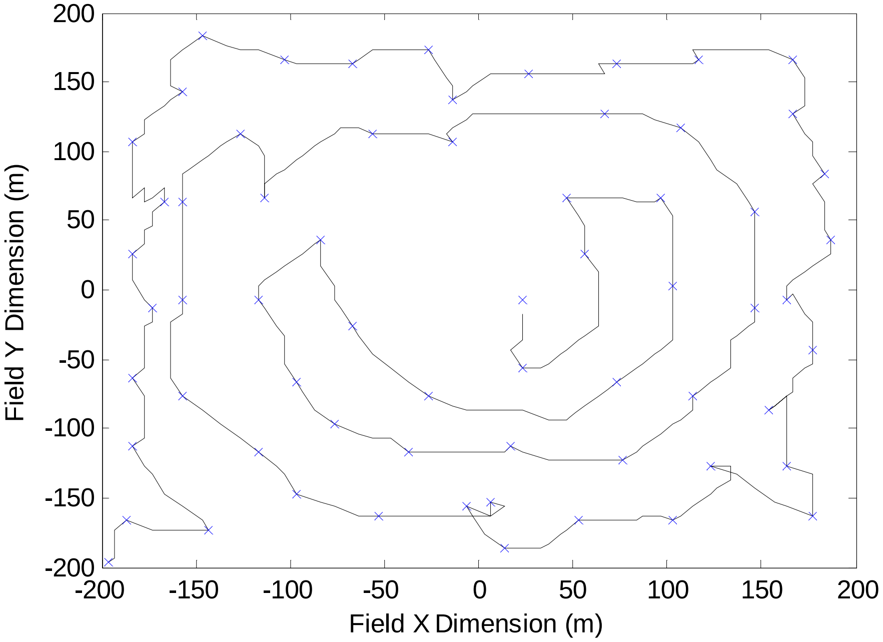
Table C.9: Results of 90% exploration of an ideal slope for objective 2.

Start Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	8.4	3255	1106	61
	Line sweeping method	6.5	2391	810	61
	Spiral pattern #1	9.2	3376	1146	77
	Spiral pattern #2	7.8	2988	1012	71
	Random	23.2	8944	2995	52
B	Greedy method	7.7	3113	1057	54
	Line sweeping method	6.2	2525	856	62
	Spiral pattern #1	8.6	3634	1228	76
	Spiral pattern #2	7.5	3063	1035	67
	Random	29.0	11354	3798	50
C	Greedy method	7.6	2871	977	59
	Line sweeping method	6.9	2589	877	61
	Spiral pattern #1	9.6	3455	1171	80
	Spiral pattern #2	8.3	3126	1058	69
	Random	25.6	9767	3272	56
D	Greedy method	8.4	3233	1100	62
	Line sweeping method	6.2	2269	769	58
	Spiral pattern #1	9.8	3551	1203	80
	Spiral pattern #2	8.1	3063	1036	68
	Random	25.6	9296	3113	53

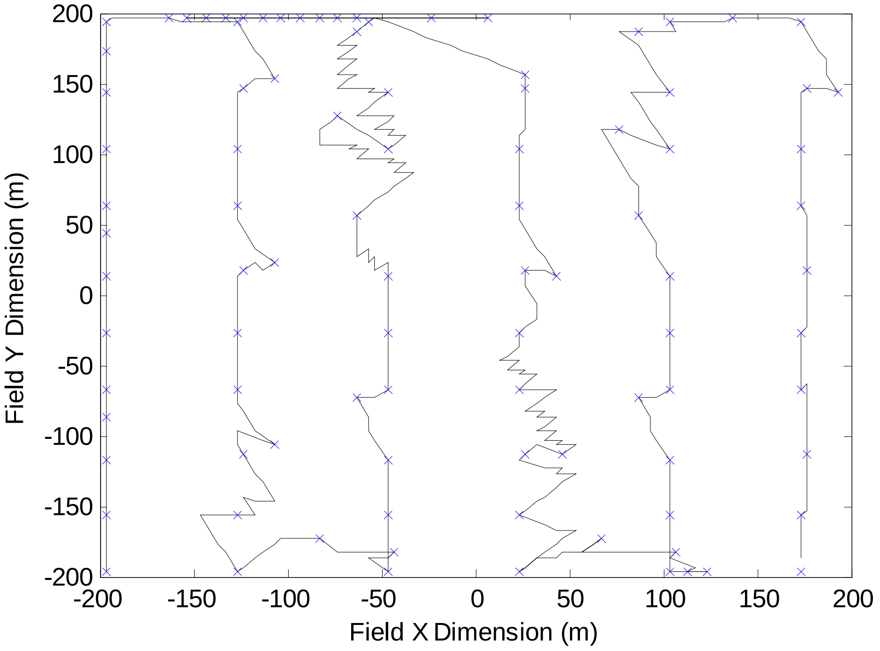
Table C.10: Results of 50% exploration of an ideal slope for objective 2.

Start Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	4.1	1541	521	29
	Line sweeping method	3.6	1349	455	36
	Spiral pattern #1	4.2	1438	487	39
	Spiral pattern #2	4.2	1438	486	39
	Random	6.1	2350	788	26
B	Greedy method	2.9	1453	491	25
	Line sweeping method	3.4	1527	515	37
	Spiral pattern #1	4.3	1537	518	41
	Spiral pattern #2	4.4	1500	506	40
	Random	10.3	3840	1284	23
C	Greedy method	3.8	1487	502	29
	Line sweeping method	4.0	1547	522	36
	Spiral pattern #1	4.0	1504	508	39
	Spiral pattern #2	4.0	1492	504	38
	Random	8.7	3367	1127	25
D	Greedy method	3.9	1421	481	29
	Line sweeping method	3.4	1282	432	34
	Spiral pattern #1	4.1	1593	538	41
	Spiral pattern #2	4.1	1598	539	40
	Random	9.3	3237	1084	25

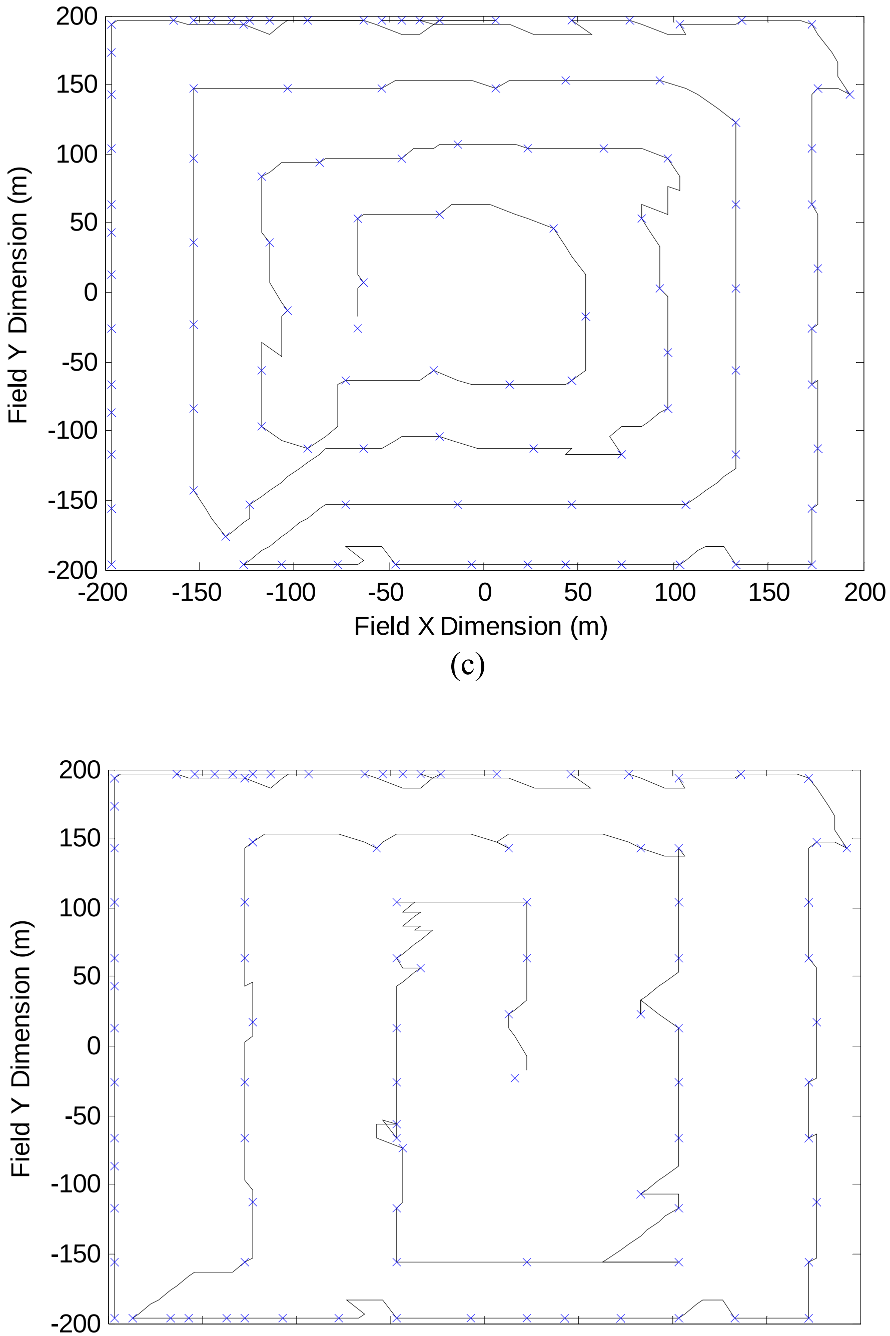
C.3 Trajectory paths generated in objective 2



(a)



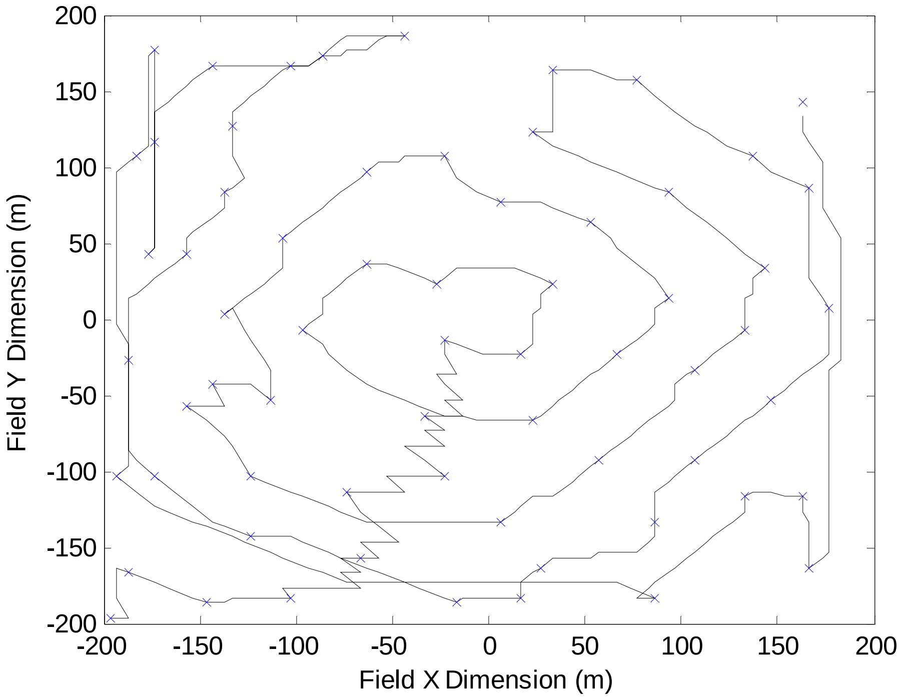
(b)



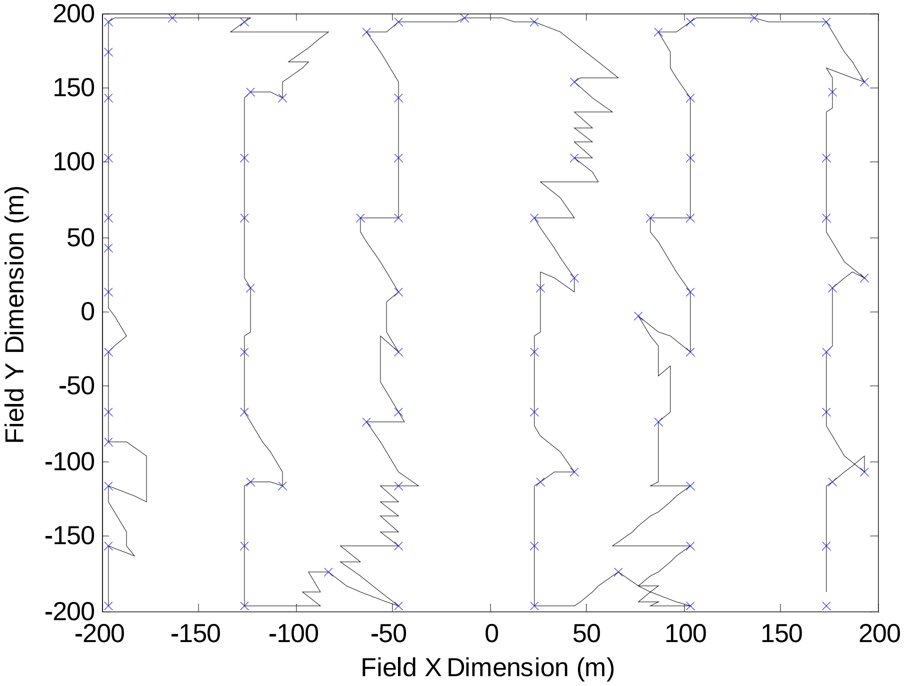
(c)

(d)

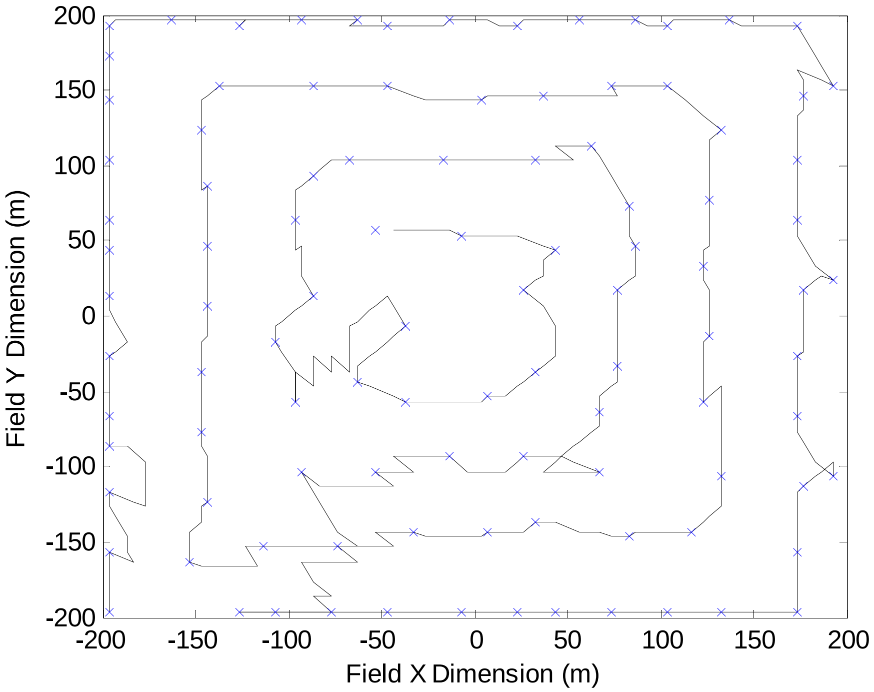
Figure C.29 Trajectory generated for the mountain environment (cross marks represent viewpoints): (a) greedy method, (b) line sweeping method, (c) spiral pattern 1, (d) spiral pattern 2.


200
150
100
50
0
-50
-100
-150
-200
-200
-150
-100
-50
0
50
100
150
200
Field Y Dimension (m)
Field X Dimension (m)

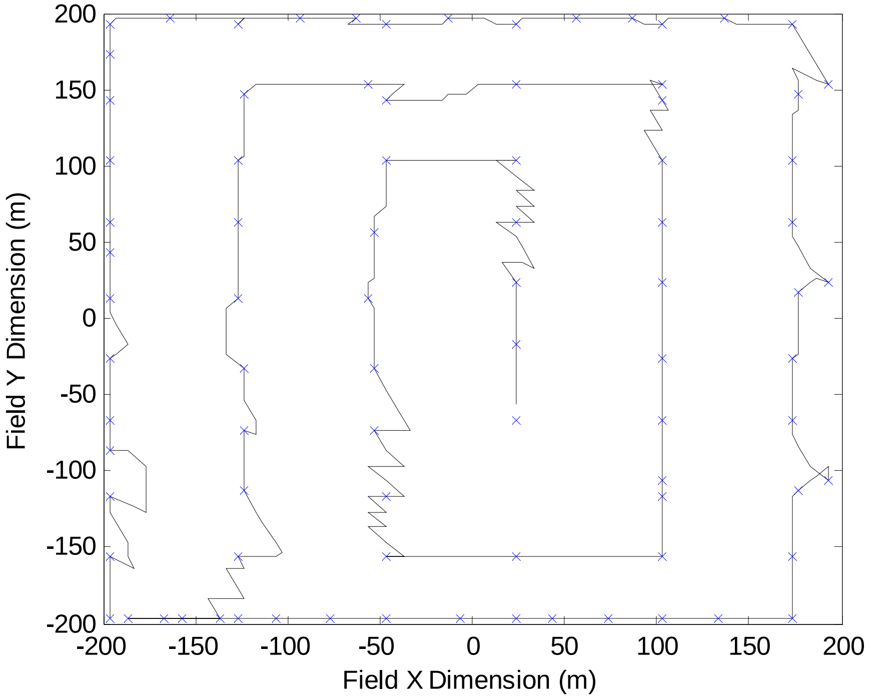
(a)


200
150
100
50
0
-50
-100
-150
-200
-200
-150
-100
-50
0
50
100
150
200
Field Y Dimension (m)
Field X Dimension (m)

(b)

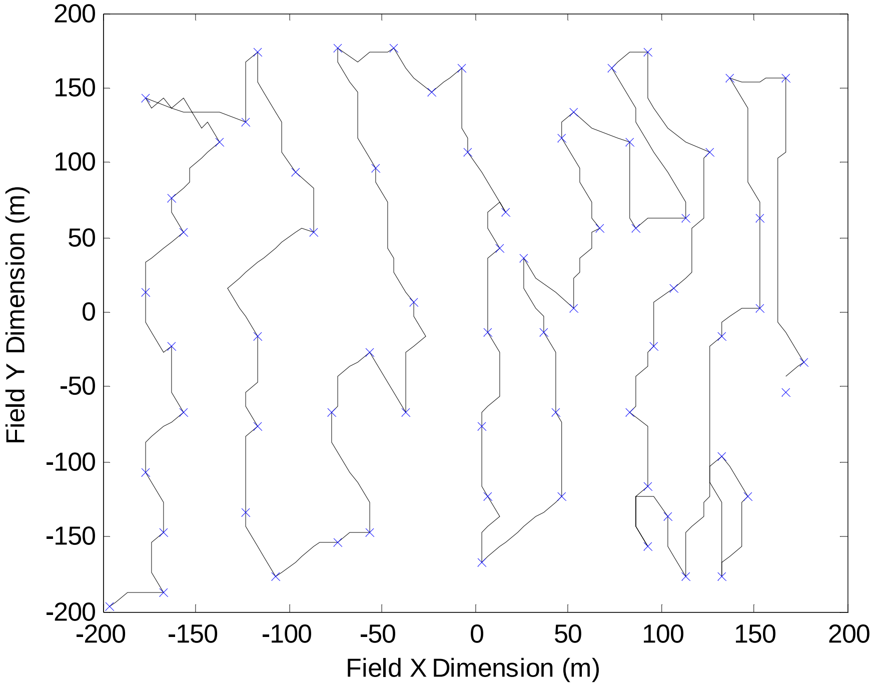


(c)

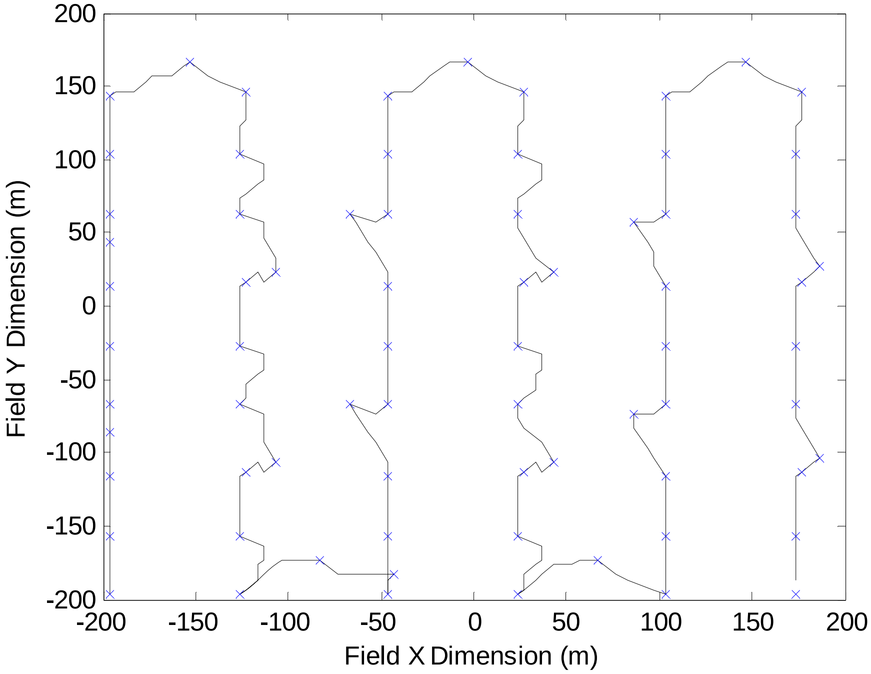


(d)

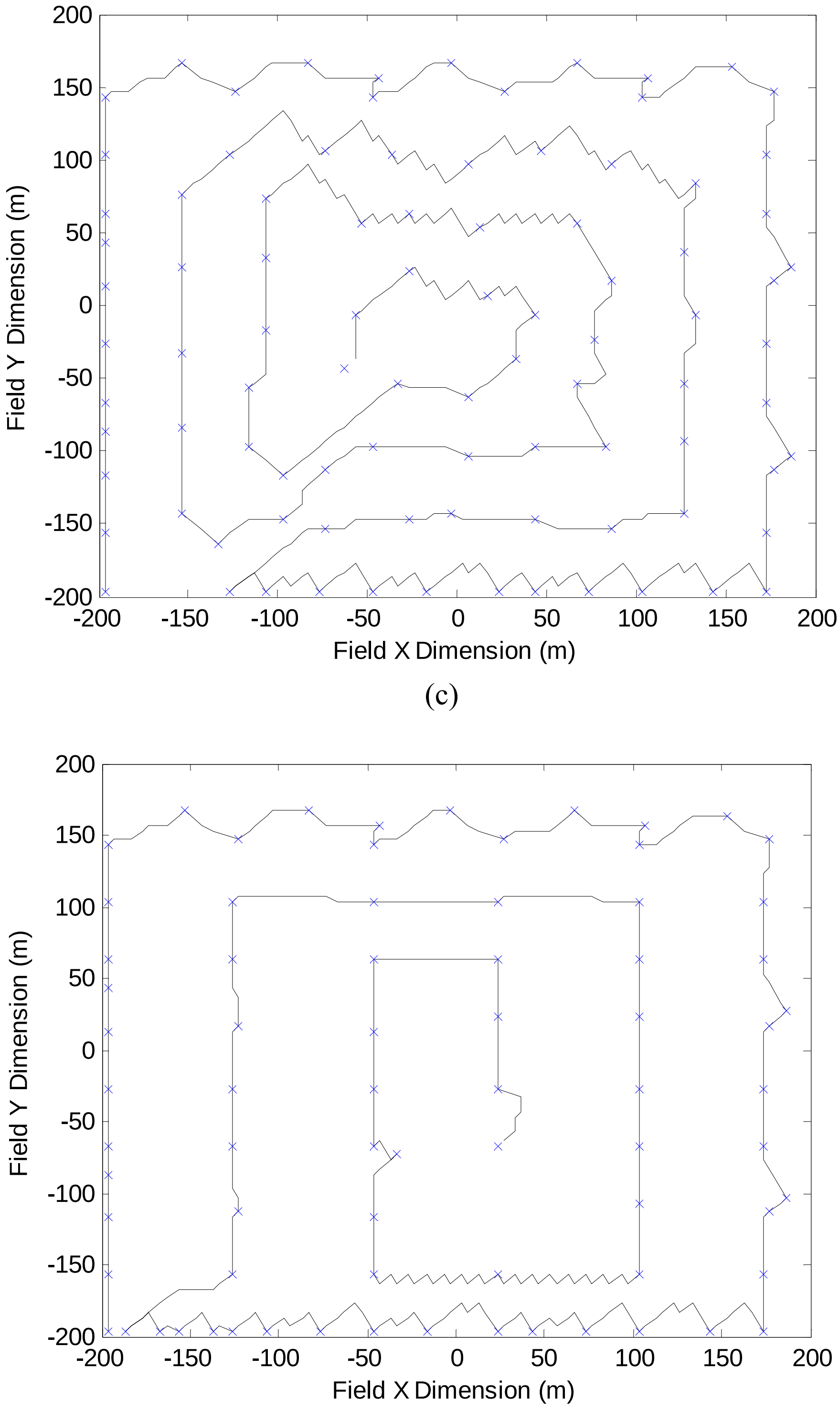
Figure C.30 Trajectory generated for the hole environment (cross marks represent viewpoints): (a) greedy method, (b) line sweeping method, (c) spiral pattern 1, (d) spiral pattern 2.


200
150
100
50
0
-50
-100
-150
-200
Field Y Dimension (m)
-200
-150
-100
-50
0
50
100
150
200
Field X Dimension (m)

(a)


200
150
100
50
0
-50
-100
-150
-200
Field Y Dimension (m)
-200
-150
-100
-50
0
50
100
150
200
Field X Dimension (m)

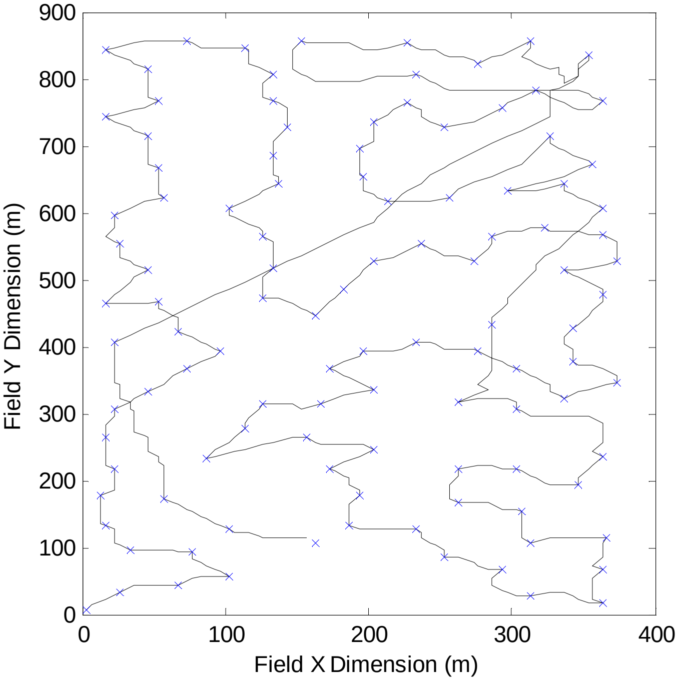
(b)



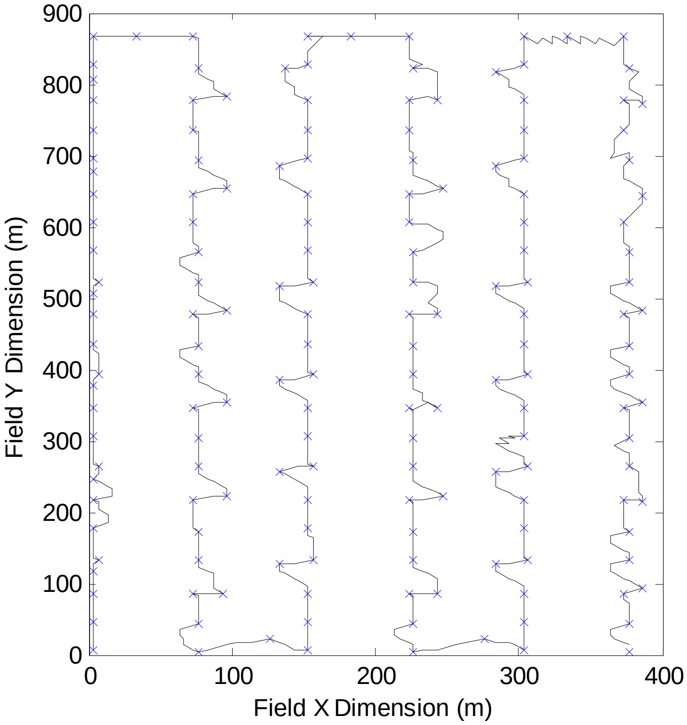
(c)

(d)

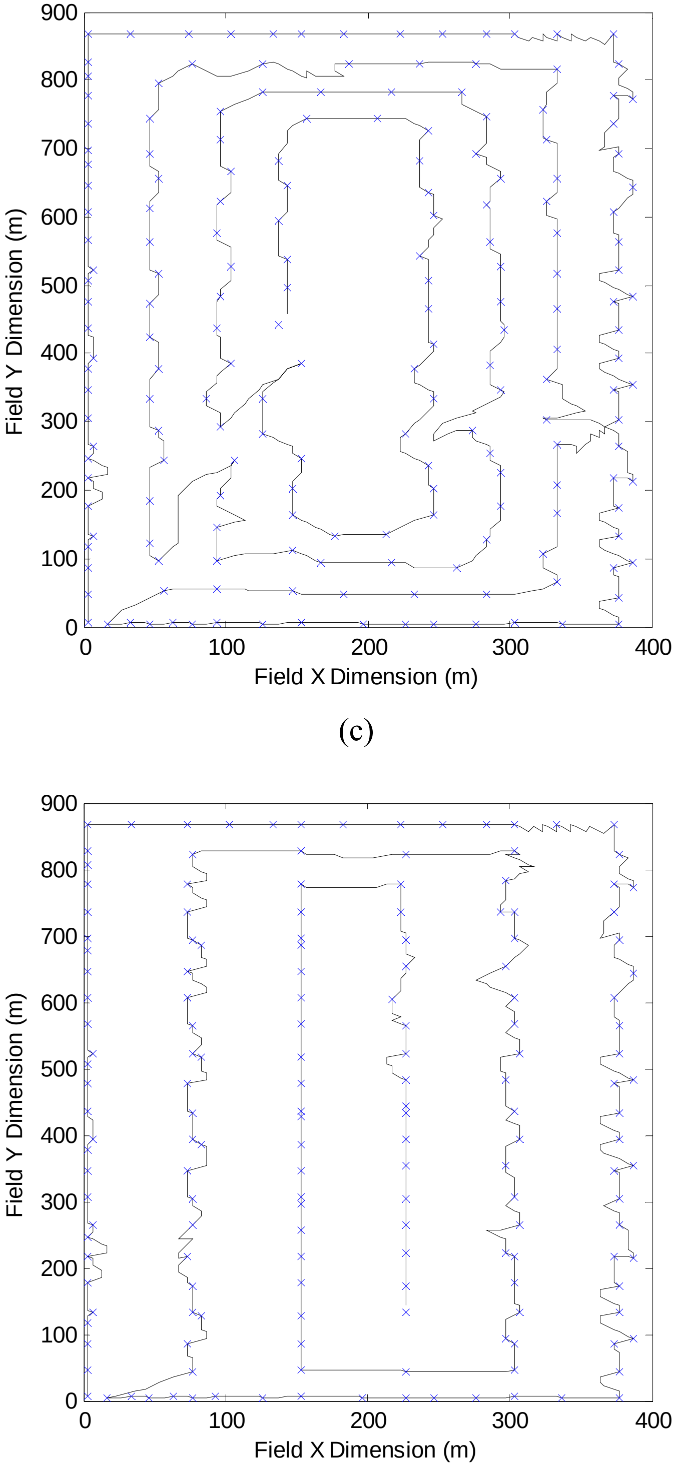
Figure C.31 Trajectory generated for the slope environment (cross marks represent viewpoints): (a) greedy method, (b) line sweeping method, (c) spiral pattern 1, (d) spiral pattern 2.


Field Y Dimension (m)
Field X Dimension (m)
0
100
200
300
400
500
600
700
800
900
0
100
200
300
400

(a)


Field Y Dimension (m)
Field X Dimension (m)
0
100
200
300
400
500
600
700
800
900
0
100
200
300
400

(b)



(c)

(d)

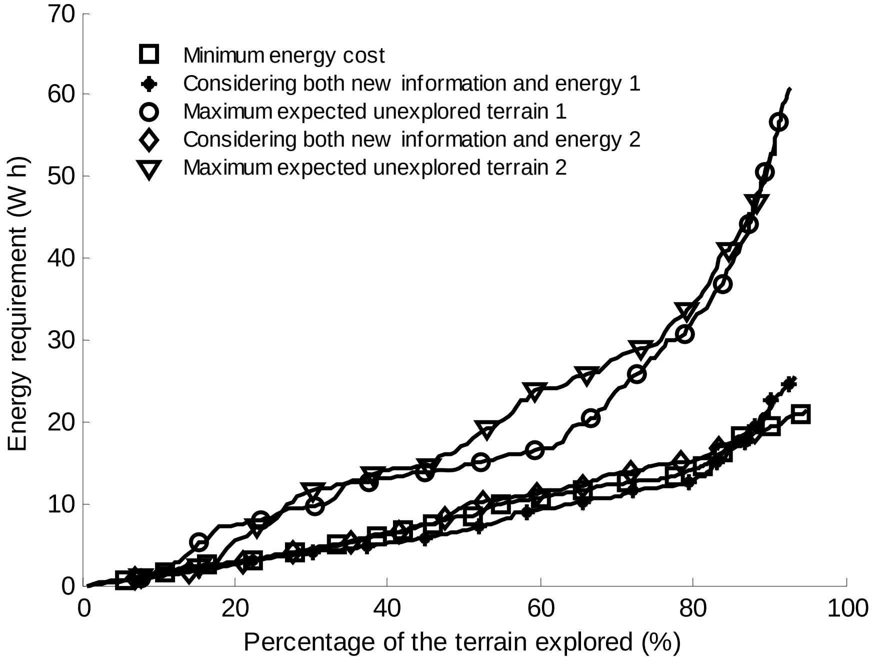
Figure C.32 Trajectory generated for agricultural field 2 (cross marks represent viewpoints): (a) greedy method, (b) line sweeping method, (c) spiral pattern 1, (d) spiral pattern 2.

APPENDIX D - PLOTS OF THE AUTONOMOUS CONSTRUCTION OF MAPS WITH A PRIORI MAP

D.1 Figures generated for objective 3

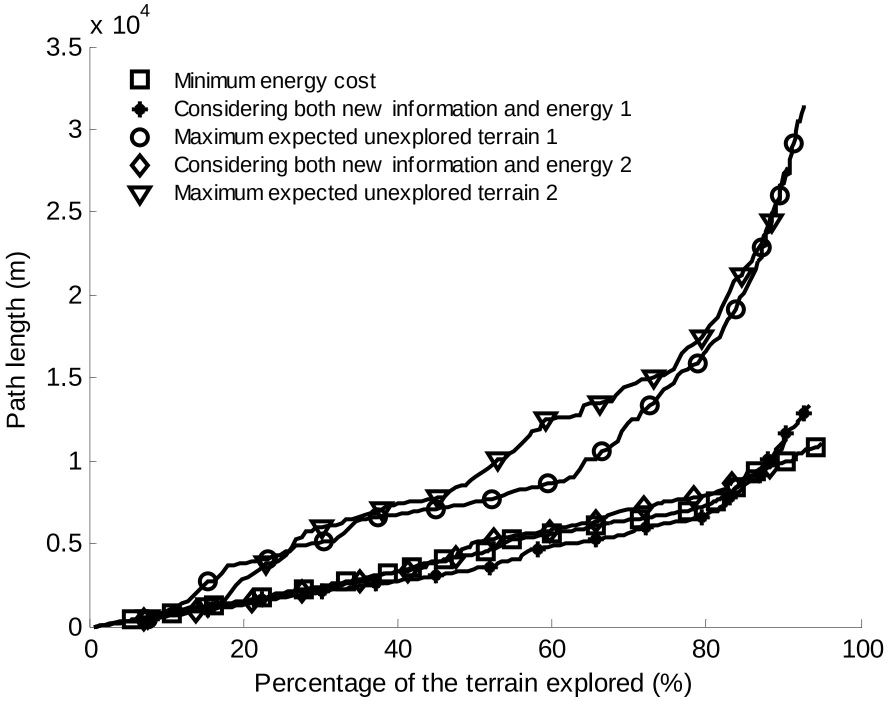
(1) Figures D.1 – D.3: Results of the autonomous construction of agricultural field #1 map for objective 3,

(2) Figures D.4 – D.7: Results of the autonomous construction of agricultural field #2 map for objective 3.



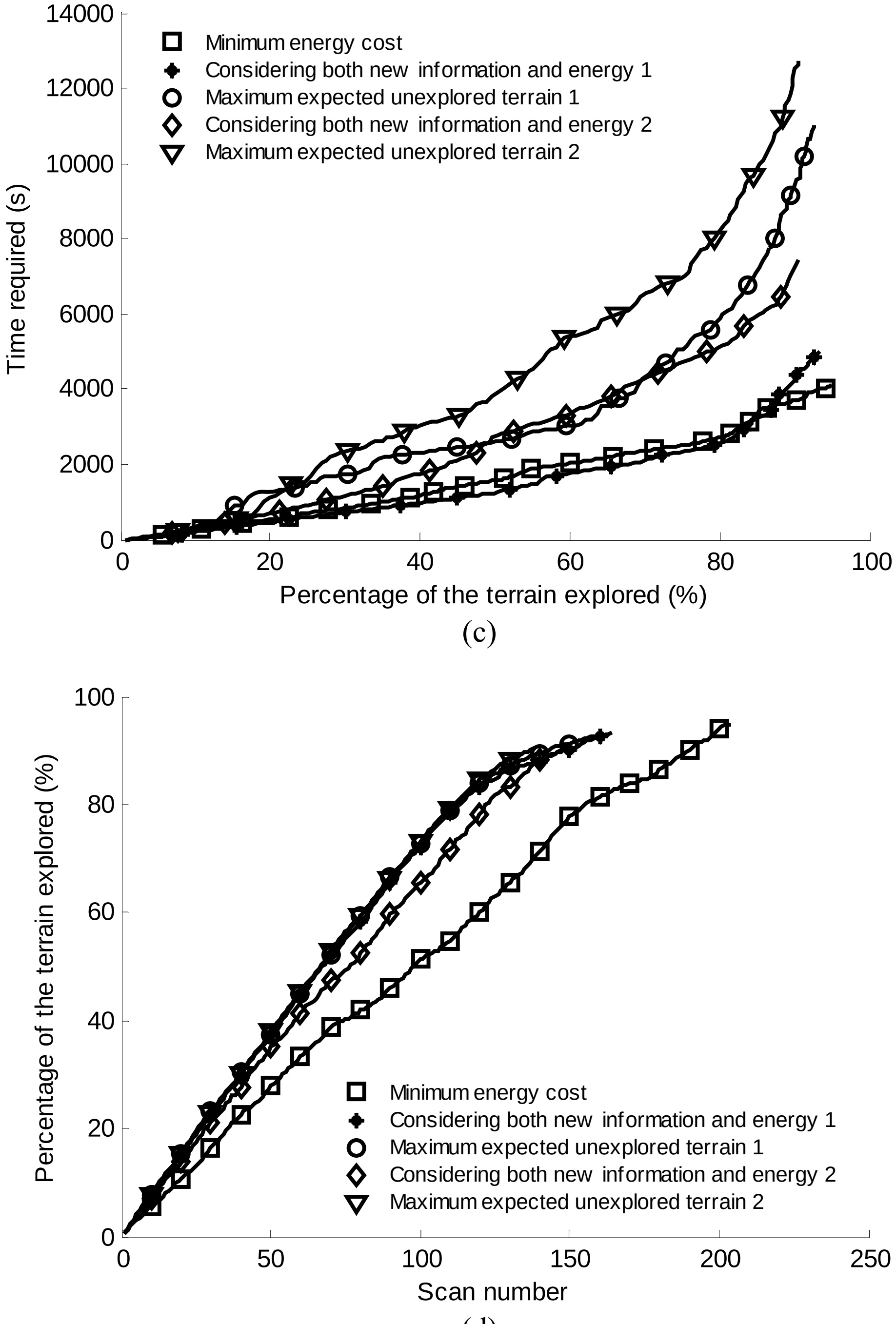
70
60
50
40
30
20
10
0
0
20
40
60
80
100
Energy requirement (W h)
Percentage of the terrain explored (%)
□ Minimum energy cost
✚ Considering both new information and energy 1
○ Maximum expected unexplored terrain 1
◇ Considering both new information and energy 2
▽ Maximum expected unexplored terrain 2

(a)



x 10^4
3.5
3
2.5
2
1.5
1
0.5
0
0
20
40
60
80
100
Path length (m)
Percentage of the terrain explored (%)
□ Minimum energy cost
✚ Considering both new information and energy 1
○ Maximum expected unexplored terrain 1
◇ Considering both new information and energy 2
▽ Maximum expected unexplored terrain 2

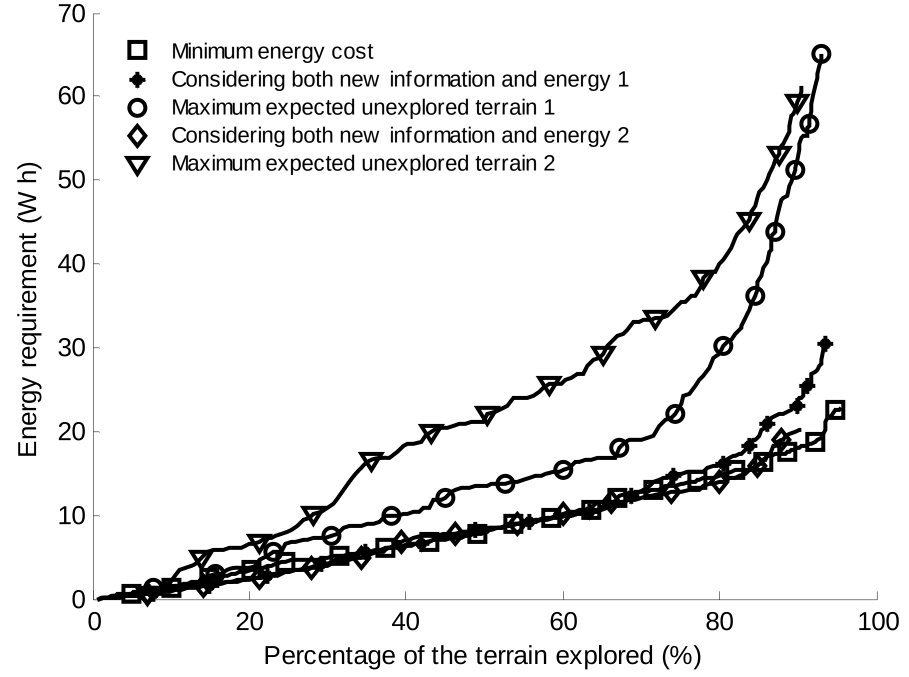
(b)



(c)

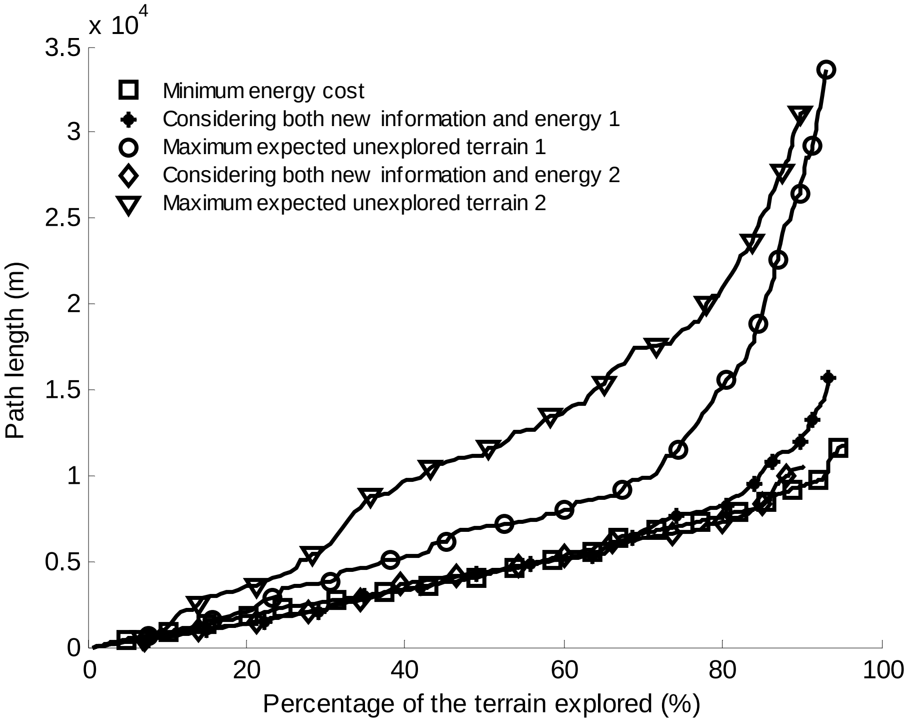
(d)

Figure D.1 Result of the autonomous mapping of agricultural field 1 for objective 3 (start B): (a) energy requirement of the exploration as a function of fraction of explored terrain, (b) path length traveled by the robot as a function of fraction of explored terrain, (c) time required of the exploration as a function of fraction of explored terrain, (d) the fraction of explored terrain as a function of scan number.



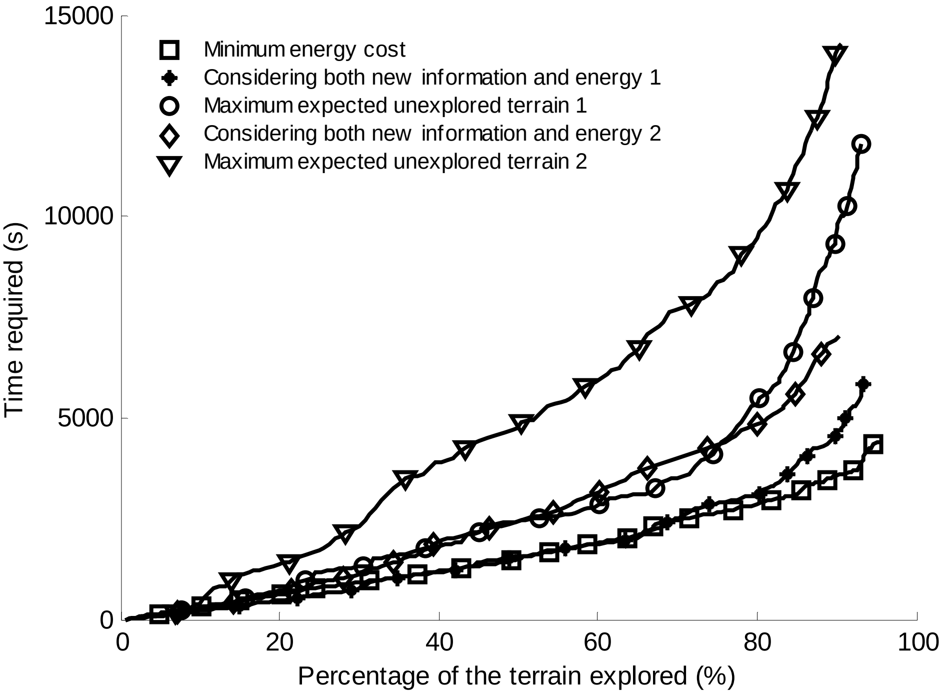
70
60
50
40
30
20
10
0
0
20
40
60
80
100
Energy requirement (W h)
Percentage of the terrain explored (%)
□ Minimum energy cost
✚ Considering both new information and energy 1
○ Maximum expected unexplored terrain 1
◇ Considering both new information and energy 2
▽ Maximum expected unexplored terrain 2

(a)

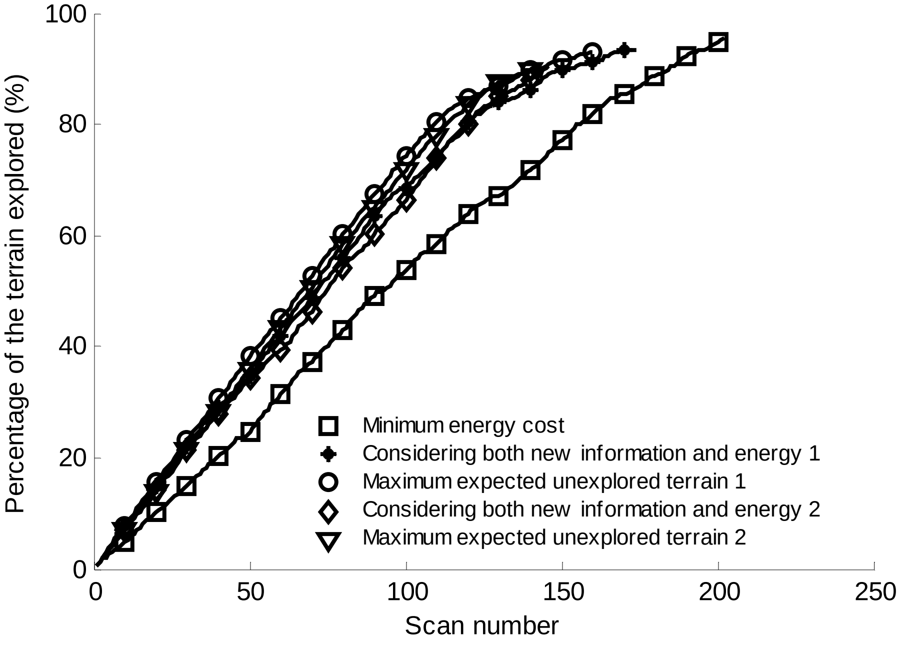


x 10^4
3.5
3
2.5
2
1.5
1
0.5
0
0
20
40
60
80
100
Path length (m)
Percentage of the terrain explored (%)
□ Minimum energy cost
✚ Considering both new information and energy 1
○ Maximum expected unexplored terrain 1
◇ Considering both new information and energy 2
▽ Maximum expected unexplored terrain 2

(b)

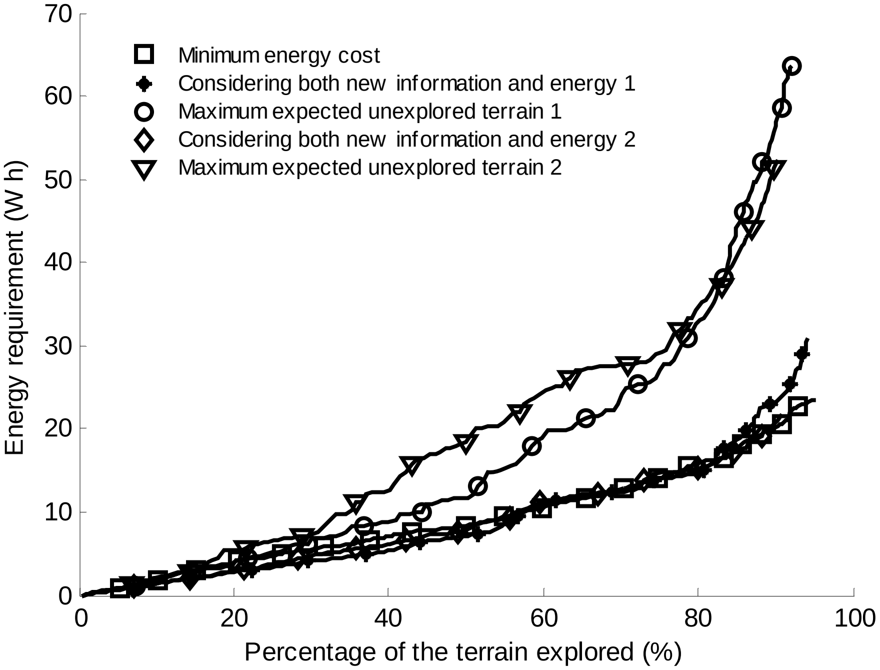


(c)

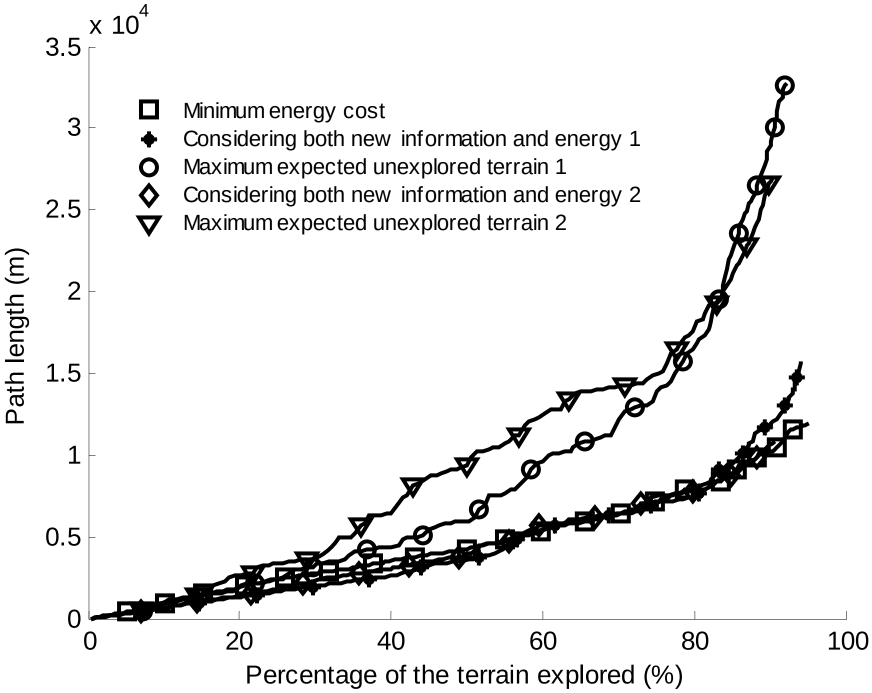


(d)

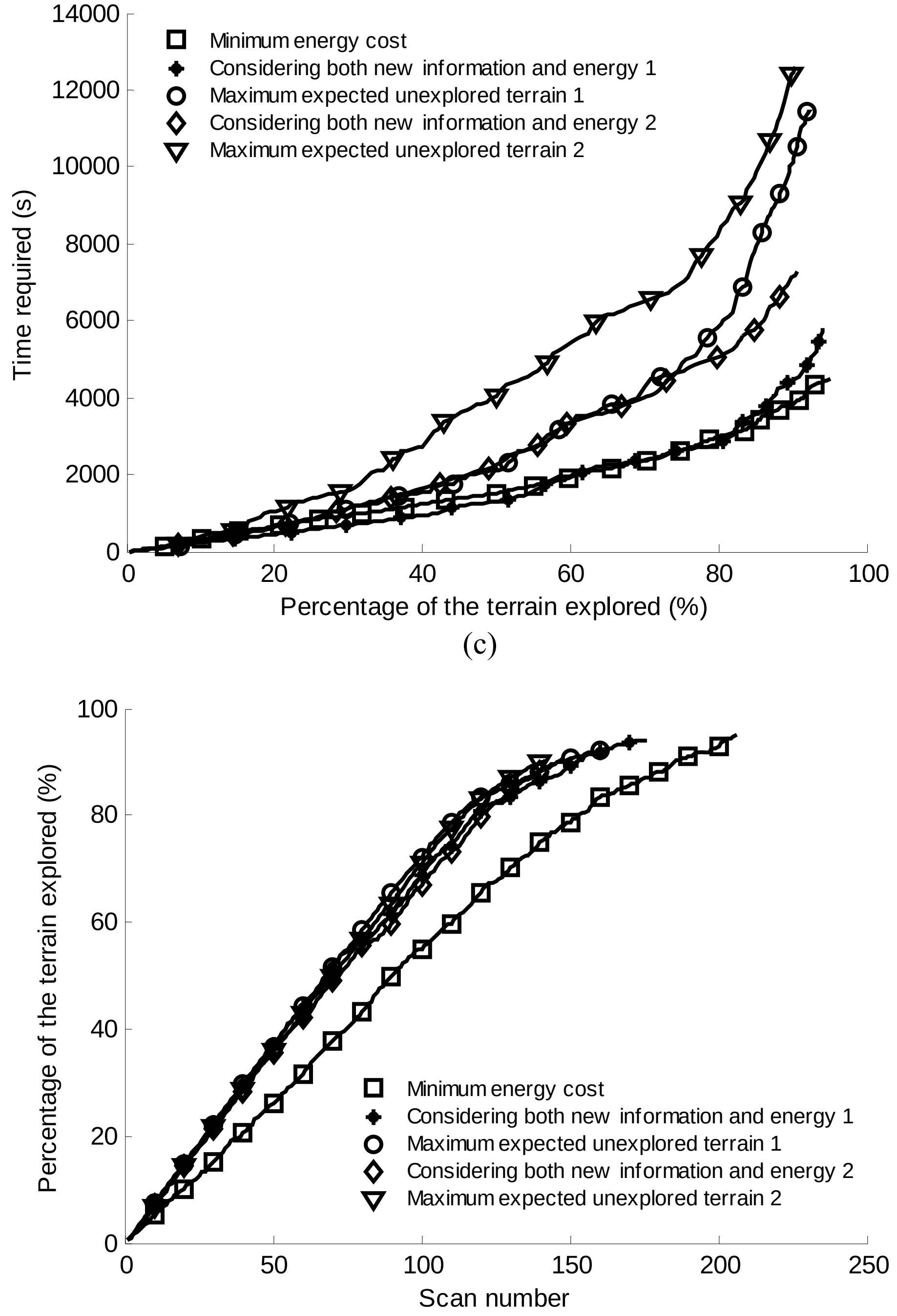
Figure D.2 Result of the autonomous mapping of agricultural field 1 for objective 3 (start C): (a) energy requirement of the exploration as a function of fraction of explored terrain, (b) path length traveled by the robot as a function of fraction of explored terrain, (c) time required of the exploration as a function of fraction of explored terrain, (d) the fraction of explored terrain as a function of scan number.


70
60
50
40
30
20
10
0
Energy requirement (W h)
0
20
40
60
80
100
Percentage of the terrain explored (%)
□ Minimum energy cost
✚ Considering both new information and energy 1
○ Maximum expected unexplored terrain 1
◇ Considering both new information and energy 2
▽ Maximum expected unexplored terrain 2

(a)


x 10^4
3.5
3
2.5
2
1.5
1
0.5
0
Path length (m)
0
20
40
60
80
100
Percentage of the terrain explored (%)
□ Minimum energy cost
✚ Considering both new information and energy 1
○ Maximum expected unexplored terrain 1
◇ Considering both new information and energy 2
▽ Maximum expected unexplored terrain 2

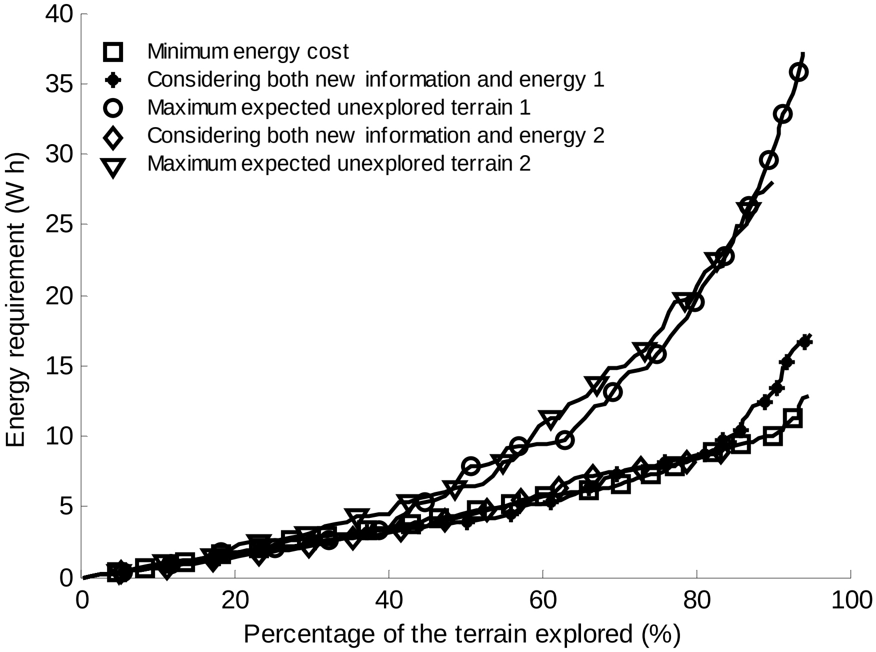
(b)



(c)

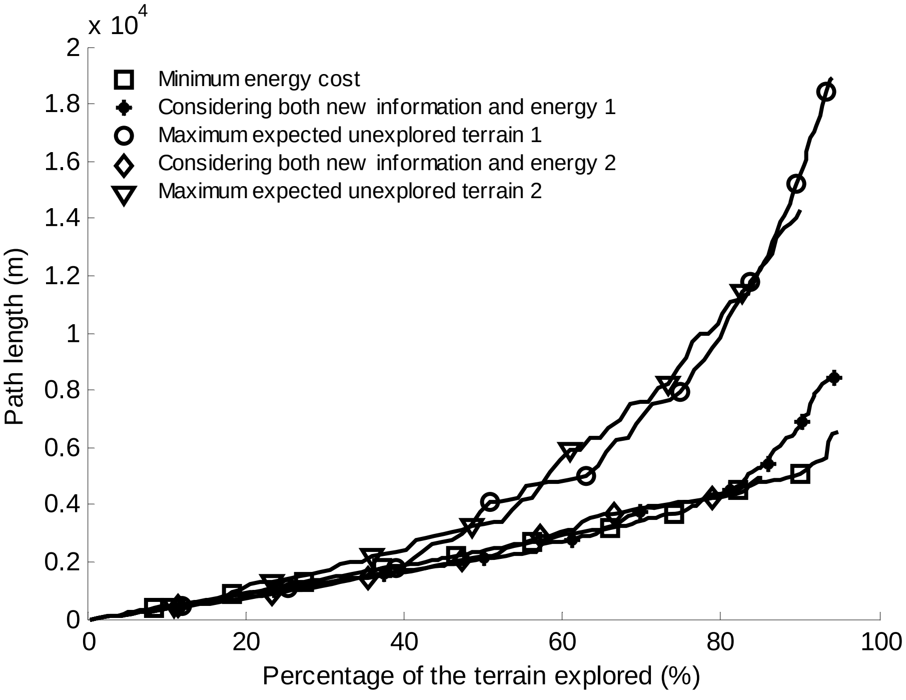
(d)

Figure D.3 Result of the autonomous mapping of agricultural field 1 for objective 3 (start D): (a) energy requirement of the exploration as a function of fraction of explored terrain, (b) path length traveled by the robot as a function of fraction of explored terrain, (c) time required of the exploration as a function of fraction of explored terrain, (d) the fraction of explored terrain as a function of scan number.



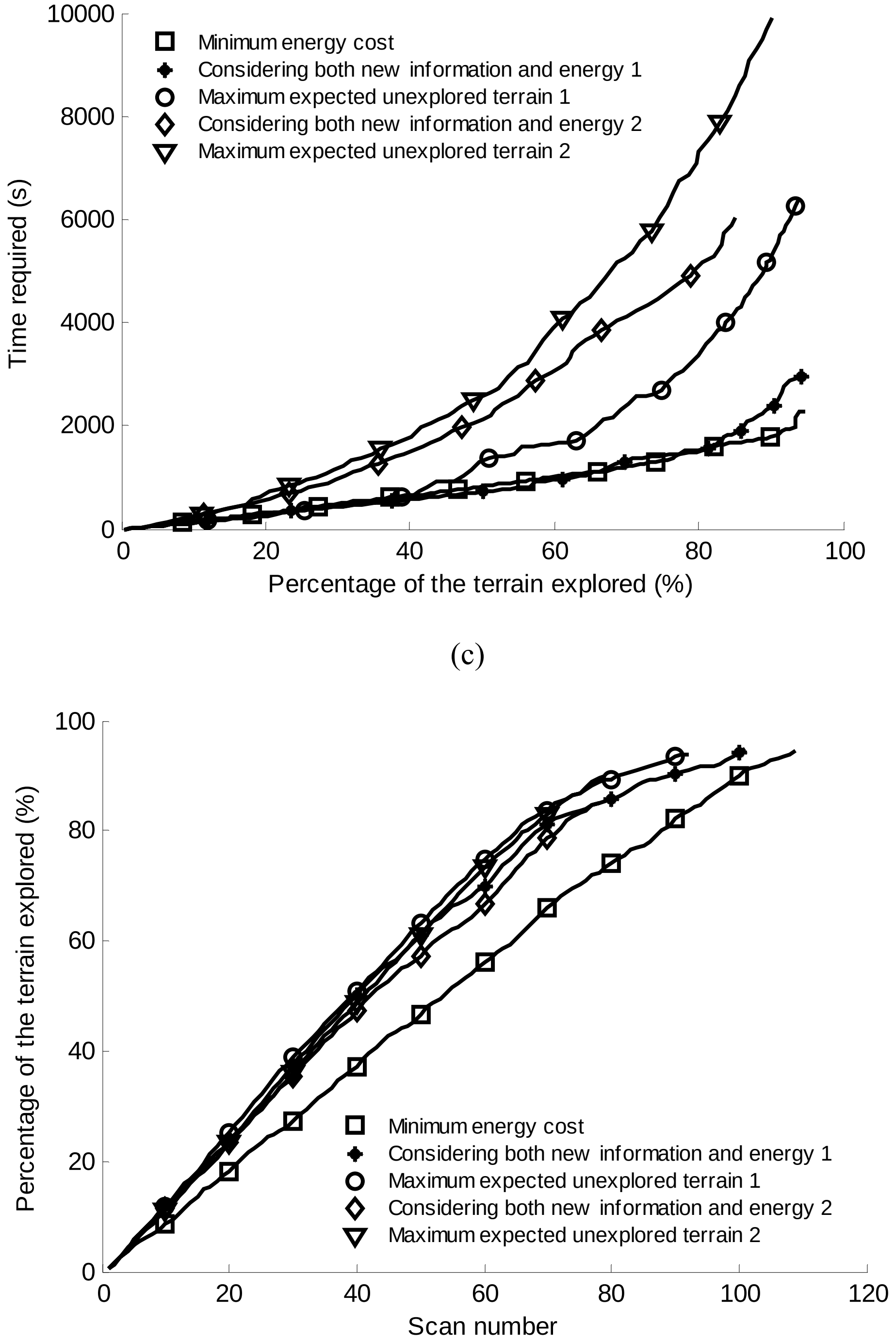
40
35
30
25
20
15
10
5
0
Energy requirement (W h)
□ Minimum energy cost
✚ Considering both new information and energy 1
○ Maximum expected unexplored terrain 1
◇ Considering both new information and energy 2
▽ Maximum expected unexplored terrain 2
0
20
40
60
80
100
Percentage of the terrain explored (%)

(a)



x 10^4
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Path length (m)
□ Minimum energy cost
✚ Considering both new information and energy 1
○ Maximum expected unexplored terrain 1
◇ Considering both new information and energy 2
▽ Maximum expected unexplored terrain 2
0
20
40
60
80
100
Percentage of the terrain explored (%)

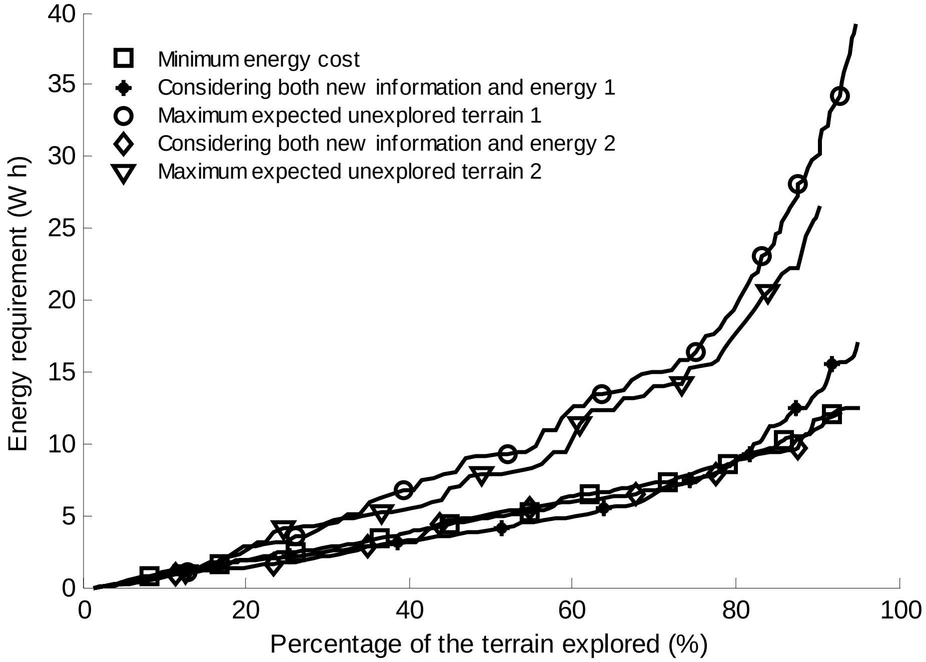
(b)



(c)

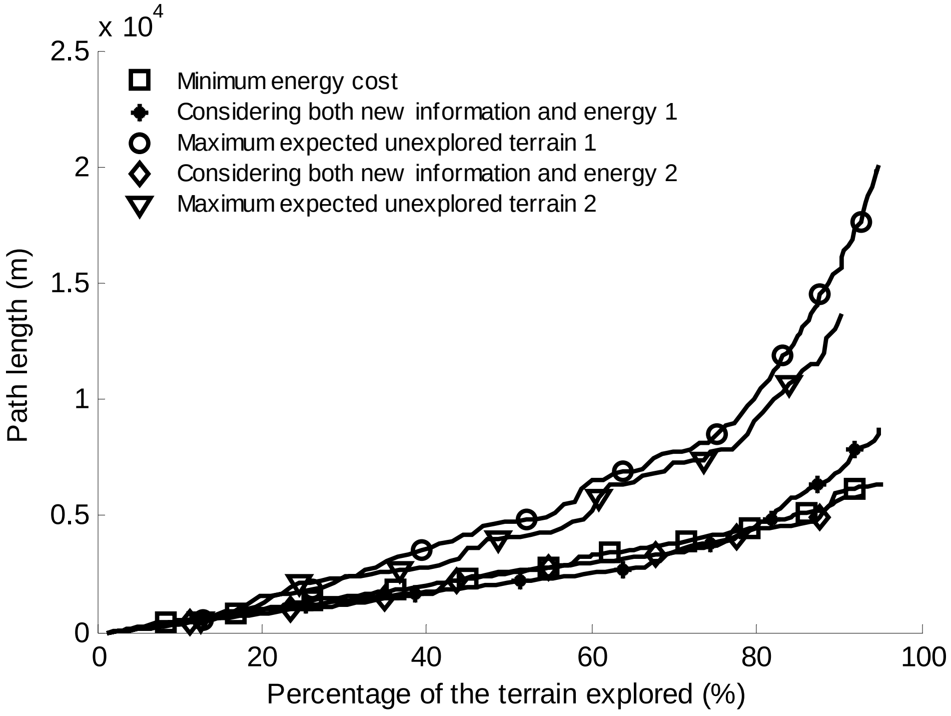
(d)

Figure D.4 Result of the autonomous mapping of agricultural field 2 for objective 3 (start A): (a) energy requirement of the exploration as a function of fraction of explored terrain, (b) path length traveled by the robot as a function of fraction of explored terrain, (c) time required of the exploration as a function of fraction of explored terrain, (d) the fraction of explored terrain as a function of scan number.



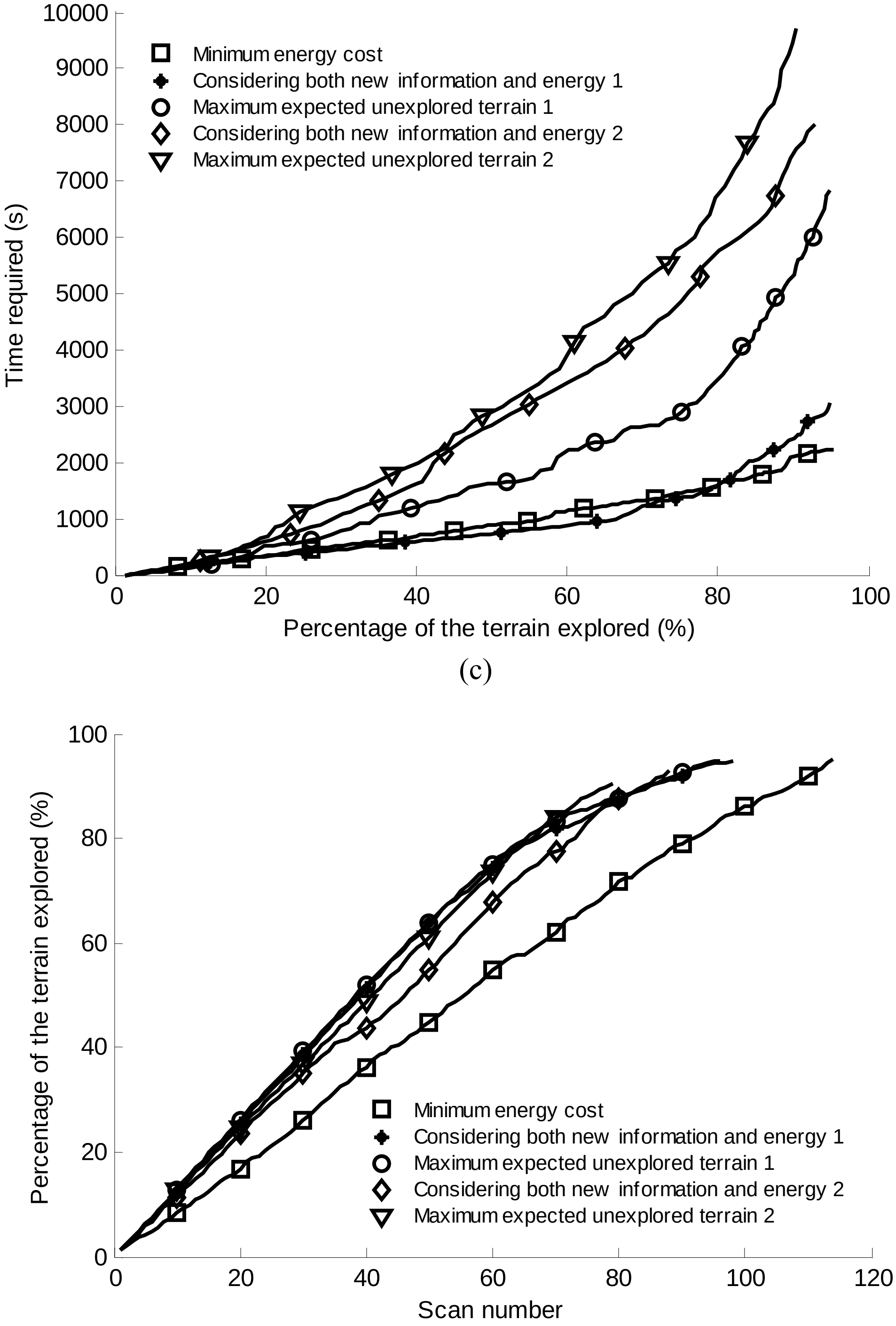
□ Minimum energy cost
✦ Considering both new information and energy 1
○ Maximum expected unexplored terrain 1
◇ Considering both new information and energy 2
▽ Maximum expected unexplored terrain 2
Energy requirement (W h)
0
5
10
15
20
25
30
35
40
0
20
40
60
80
100
Percentage of the terrain explored (%)

(a)



x 10^4
□ Minimum energy cost
✦ Considering both new information and energy 1
○ Maximum expected unexplored terrain 1
◇ Considering both new information and energy 2
▽ Maximum expected unexplored terrain 2
Path length (m)
0
0.5
1
1.5
2
2.5
0
20
40
60
80
100
Percentage of the terrain explored (%)

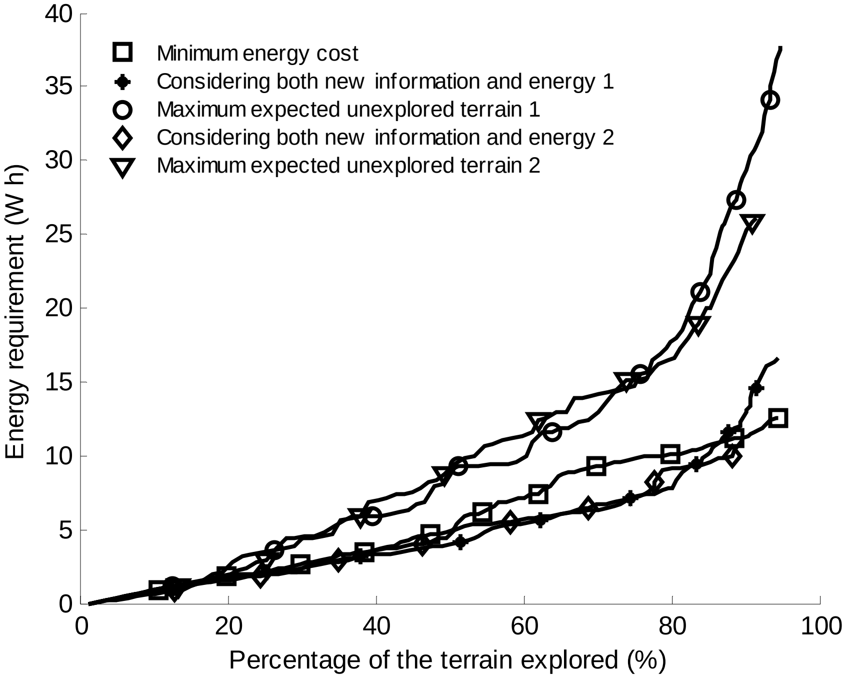
(b)



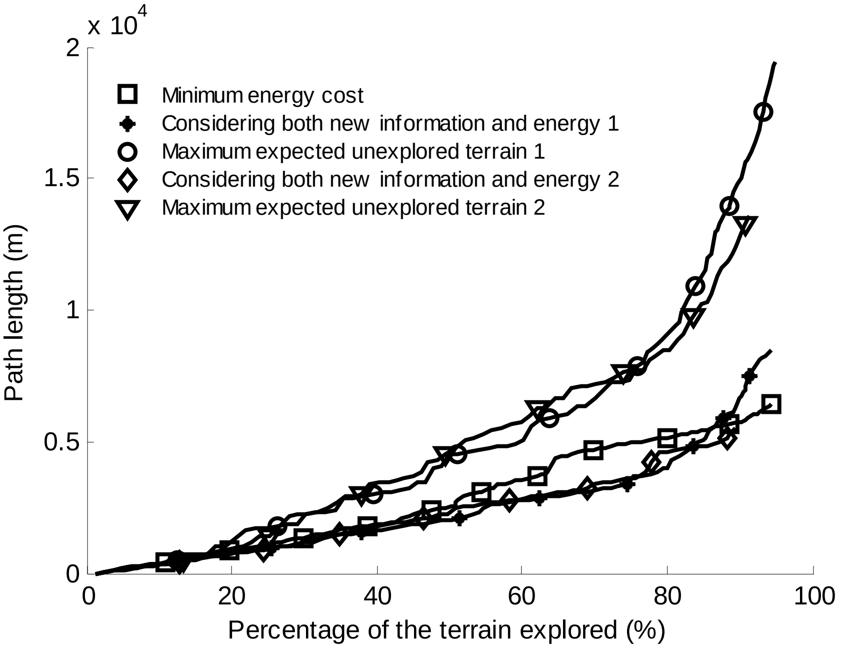
(c)

(d)

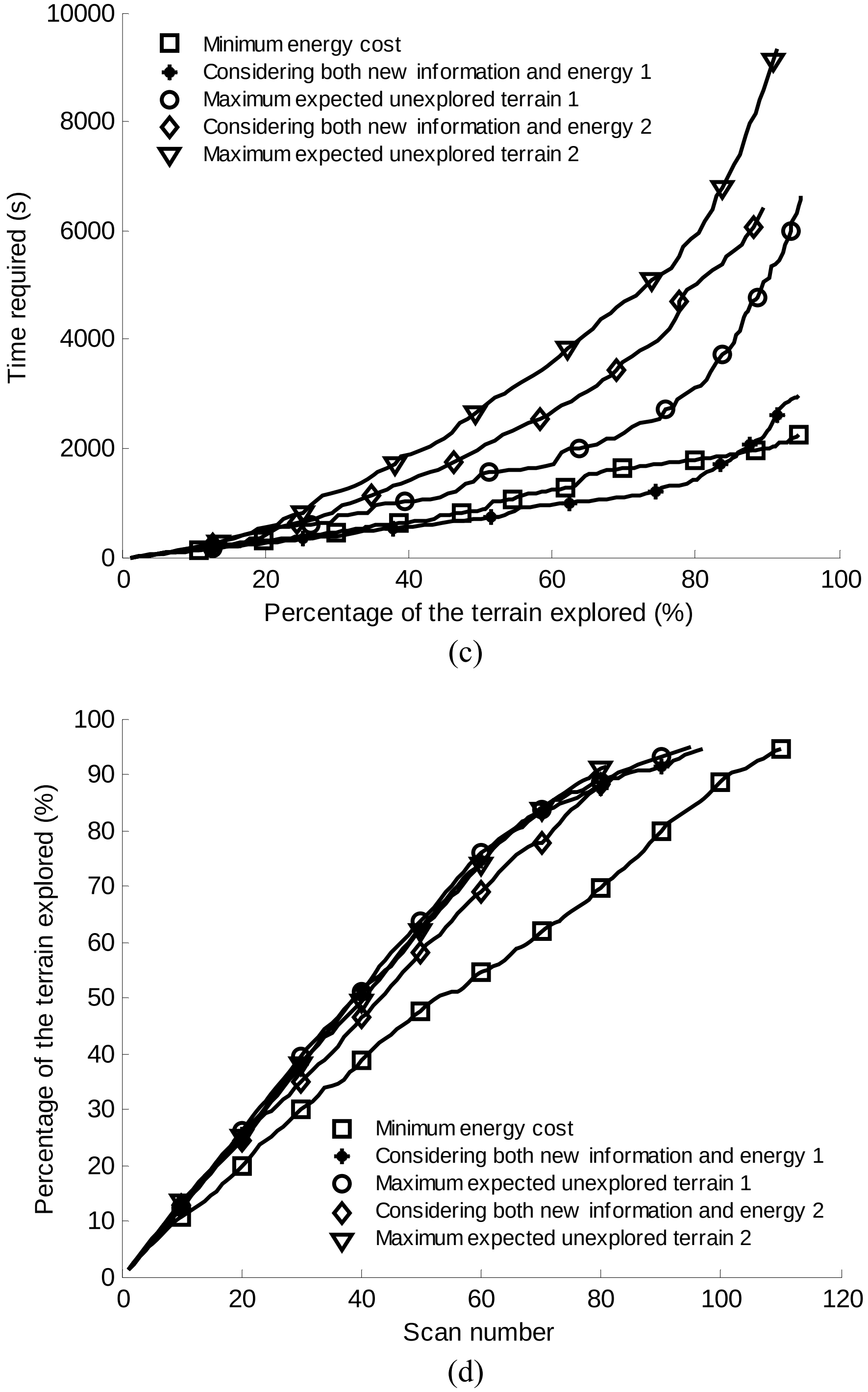
Figure D.5 Result of the autonomous mapping of agricultural field 2 for objective 3 (start B): (a) energy requirement of the exploration as a function of fraction of explored terrain, (b) path length traveled by the robot as a function of fraction of explored terrain, (c) time required of the exploration as a function of fraction of explored terrain, (d) the fraction of explored terrain as a function of scan number.


40
35
30
25
20
15
10
5
0
0
20
40
60
80
100
Energy requirement (W h)
Percentage of the terrain explored (%)
Minimum energy cost
Considering both new information and energy 1
Maximum expected unexplored terrain 1
Considering both new information and energy 2
Maximum expected unexplored terrain 2

(a)


x 10^4
2
1.5
1
0.5
0
0
20
40
60
80
100
Path length (m)
Percentage of the terrain explored (%)
Minimum energy cost
Considering both new information and energy 1
Maximum expected unexplored terrain 1
Considering both new information and energy 2
Maximum expected unexplored terrain 2

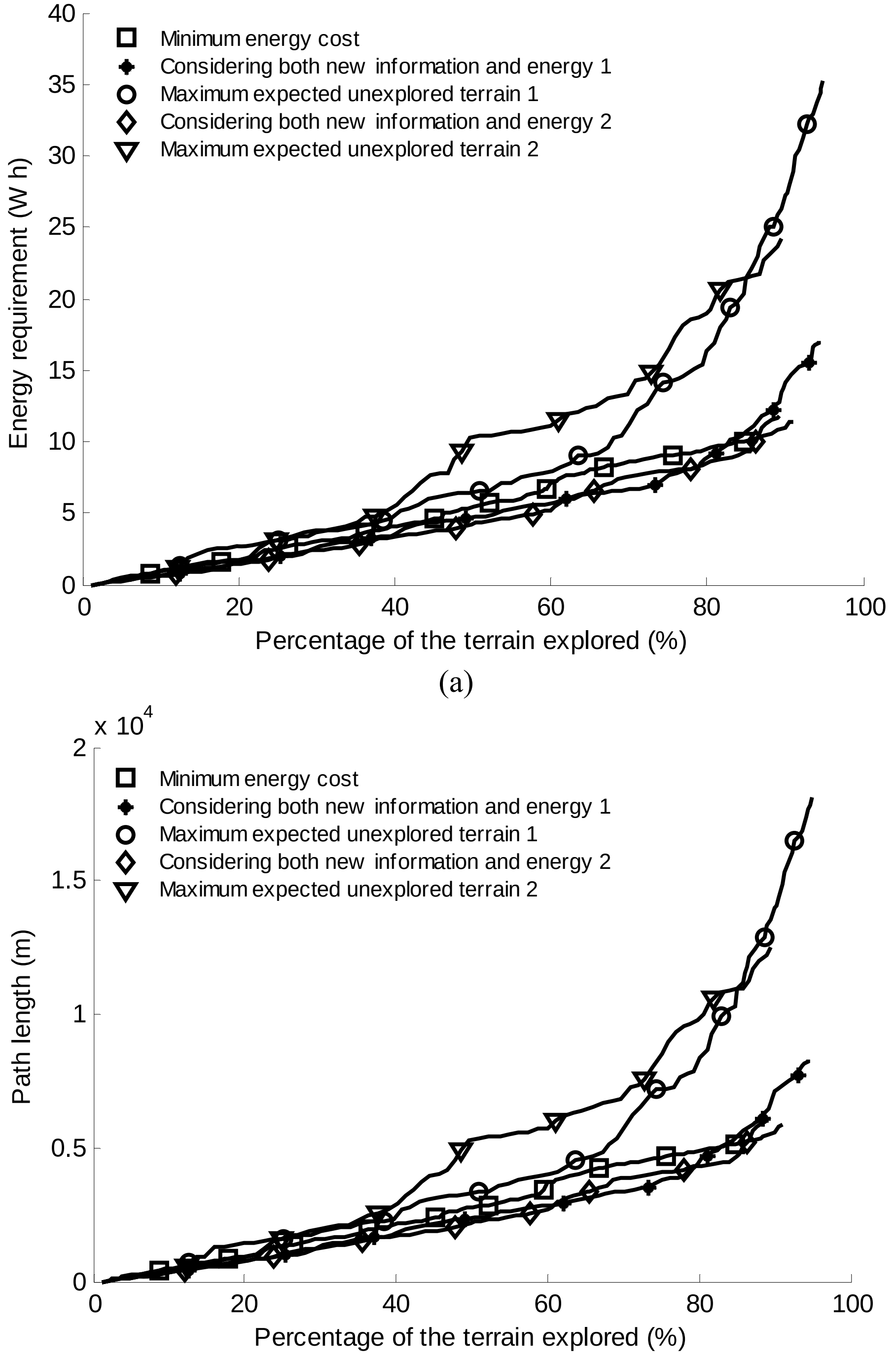
(b)



(c)

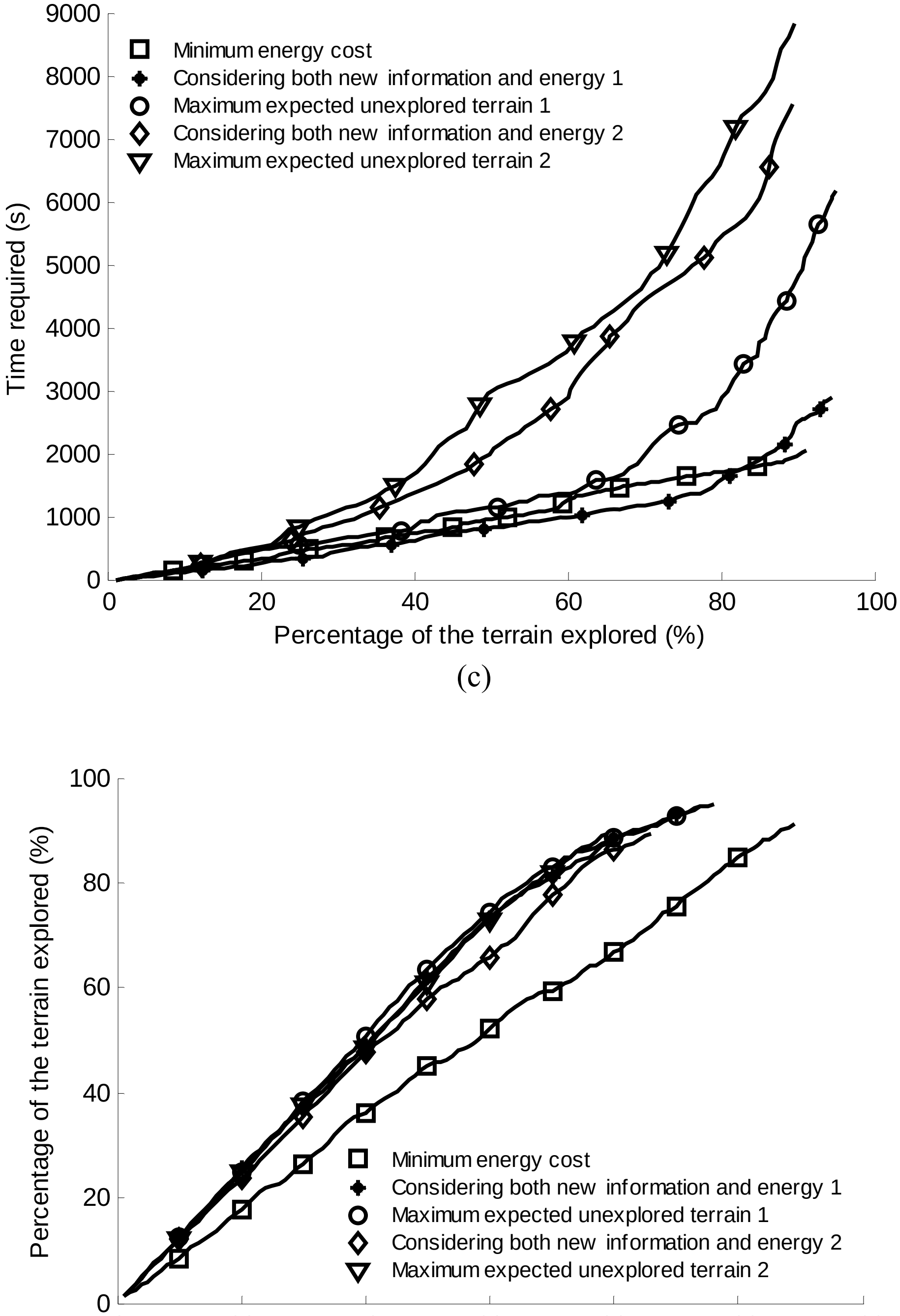
(d)

Figure D.6 Result of the autonomous mapping of agricultural field 2 for objective 3 (start C): (a) energy requirement of the exploration as a function of fraction of explored terrain, (b) path length traveled by the robot as a function of fraction of explored terrain, (c) time required of the exploration as a function of fraction of explored terrain, (d) the fraction of explored terrain as a function of scan number.


Minimum energy cost
Considering both new information and energy 1
Maximum expected unexplored terrain 1
Considering both new information and energy 2
Maximum expected unexplored terrain 2
Energy requirement (W h)
0
5
10
15
20
25
30
35
40
0
20
40
60
80
100
Percentage of the terrain explored (%)
x 10^4
Path length (m)
0
0.5
1
1.5
2
Percentage of the terrain explored (%)

(a)

(b)



(c)

(d)

Figure D.7 Result of the autonomous mapping of agricultural field 2 for objective 3 (start D): (a) energy requirement of the exploration as a function of fraction of explored terrain, (b) path length traveled by the robot as a function of fraction of explored terrain, (c) time required of the exploration as a function of fraction of explored terrain, (d) the fraction of explored terrain as a function of scan number.

D.2 Tables generated for objective 3

(1) Tables D.1– D.2: Results of the autonomous construction of agricultural field 1 map for objective 3,

(2) Tables D.3– D.4: Results of the autonomous construction of agricultural field 2 map for objective 3,

(3) Tables D.5 – D.6: Results of the autonomous construction of an ideal mountain map for objective 3,

(4) Tables D.7 – D.8: Results of the autonomous construction of an ideal hole map for objective 3,

(5) Tables D.9 – D.10: Results of the autonomous construction of an ideal slope map for objective 3.

Table D.1: Results for 90% exploration of automatic mapping of agricultural 1.

Start Location	Method	Energy (W h)	Path length (m)	Plan time (s)	Scan number
A	Greedy method	19.7	10030	3730	193
	Information Gain 0.5	22.0	11064	4170	148
	Information Gain 1.0	58.2	29917	10456	148
	Coarse map 0.5	16.6	8475	5720	130
	Coarse map 1.0	50.0	25573	11097	138
B	Greedy method	19.1	9863	3695	189
	Information Gain 0.5	21.4	11120	4221	148
	Information Gain 1.0	51.7	26788	9395	143
	Coarse map 0.5	17.2	8856	5835	132
	Coarse map 1.0	38.3	19714	9037	116
C	Greedy method	18.0	9381	3558	184
	Information Gain 0.5	23.5	12143	4575	151
	Information Gain 1.0	52.5	26963	9472	142
	Coarse map 0.5	15.5	7965	5372	127
	Coarse map 1.0	44.6	23028	10419	119
D	Greedy method	20.0	10164	3817	185
	Information Gain 0.5	23.3	11751	4387	151
	Information Gain 1.0	55.7	28555	9998	145
	Coarse map 0.5	17.2	8704	5712	129
	Coarse map 1.0	37.5	19290	9053	120

Table D.2: Results for 50% exploration of automatic mapping of agricultural field 1.

Start Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	9.5	4870	1726	99
	Information Gain 0.5	7.1	3606	1308	69
	Information Gain 1.0	18.2	9215	3173	67
	Coarse map 0.5	7.3	3689	2083	70
	Coarse map 1.0	23.2	11855	4813	76
B	Greedy method	8.4	4333	1564	97
	Information Gain 0.5	6.6	3402	1237	67
	Information Gain 1.0	14.2	7387	2563	66
	Coarse map 0.5	8.0	4159	2275	69
	Coarse map 1.0	15.8	8088	3454	62
C	Greedy method	7.9	4090	1467	91
	Information Gain 0.5	8.3	4292	1547	71
	Information Gain 1.0	13.5	6949	2429	66
	Coarse map 0.5	8.0	4179	2291	71
	Coarse map 1.0	20.9	10945	4514	64
D	Greedy method	8.3	4239	1508	90
	Information Gain 0.5	7.0	3534	1283	67
	Information Gain 1.0	11.6	5940	2083	67
	Coarse map 0.5	7.4	3642	2052	67
	Coarse map 1.0	17.4	8868	3733	65

Table D.3: Results of the autonomous mapping of agricultural field 2 (90%) for objective 3.

Start location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	10.0	5046	1778	100
	Information Gain 0.5	12.9	6602	2314	88
	Information Gain 1.0	29.8	15337	5221	81
	Coarse map 0.5	9.5	4909	5935	77
	Coarse map 1.0	22.4	11466	7931	70
B	Greedy method	11.7	5996	2093	107
	Information Gain 0.5	13.2	6801	2385	84
	Information Gain 1.0	29.8	15393	5250	83
	Coarse map 0.5	9.2	4557	6125	75
	Coarse map 1.0	20.1	10265	7476	69
C	Greedy method	11.2	5706	1984	101
	Information Gain 0.5	12.3	6296	2214	83
	Information Gain 1.0	28.8	14811	5055	82
	Coarse map 0.5	9.4	4817	5617	76
	Coarse map 1.0	19.0	9751	6806	70
D	Greedy method	11.0	5579	1950	107
	Information Gain 0.5	14.1	7113	2481	85
	Information Gain 1.0	27.2	13949	4774	83
	Coarse map 0.5	9.0	4501	5864	76
	Coarse map 1.0	21.3	10826	7356	71

Table D.4: Results of the autonomous mapping of agricultural field 2 (50% exploration) for objective 3.

Start location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	4.5	2368	819	53
	Information Gain 0.5	3.9	2038	705	39
	Information Gain 1.0	7.4	3763	1281	39
	Coarse map 0.5	4.1	2081	1978	40
	Coarse map 1.0	5.9	3027	2306	38
B	Greedy method	4.8	2528	870	55
	Information Gain 0.5	4.0	2102	728	39
	Information Gain 1.0	9.1	4738	1607	38
	Coarse map 0.5	4.7	2440	2483	43
	Coarse map 1.0	7.0	3648	2595	38
C	Greedy method	4.8	2477	853	52
	Information Gain 0.5	3.9	1987	687	38
	Information Gain 1.0	8.1	4093	1391	38
	Coarse map 0.5	4.0	2063	1762	40
	Coarse map 1.0	7.7	3856	2250	37
D	Greedy method	5.4	2744	946	57
	Information Gain 0.5	4.6	2330	805	40
	Information Gain 1.0	6.5	3292	1126	39
	Coarse map 0.5	3.8	1974	1740	39
	Coarse map 1.0	7.8	4028	2314	37

Table D.5: Results of the autonomous mapping of an ideal mountain (90% exploration) for objective 3.

Start Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	7.5	3088	1058	57
	Information Gain 0.5	6.9	3131	1066	40
	Information Gain 1.0	13.6	6192	2088	40
	Coarse map 0.5	6.6	2533	996	40
	Coarse map 1.0	8.1	3425	1265	35
B	Greedy method	7.8	3529	1205	59
	Information Gain 0.5	7.3	3317	1132	43
	Information Gain 1.0	14.5	7051	2375	41
	Coarse map 0.5	6.3	2595	1031	43
	Coarse map 1.0	8.1	3707	1381	34
C	Greedy method	7.5	3370	1152	58
	Information Gain 0.5	7.3	3347	1143	43
	Information Gain 1.0	13.1	6032	2034	39
	Coarse map 0.5	5.8	2414	957	40
	Coarse map 1.0	12.2	5922	2106	34
D	Greedy method	8.1	3543	1208	58
	Information Gain 0.5	7.1	2886	984	40
	Information Gain 1.0	11.7	5349	1806	40
	Coarse map 0.5	5.8	2255	900	39
	Coarse map 1.0	9.1	4045	1498	35

Table D.6: Results of the autonomous mapping of an ideal mountain (50% exploration) for objective 3.

Start location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	3.1	1273	433	26
	Information Gain 0.5	3.0	1145	389	20
	Information Gain 1.0	4.5	1817	614	20
	Coarse map 0.5	3.5	1376	515	23
	Coarse map 1.0	5.0	2064	734	21
B	Greedy method	3.0	1300	442	27
	Information Gain 0.5	2.3	968	330	19
	Information Gain 1.0	3.7	1784	603	19
	Coarse map 0.5	3.0	1343	498	23
	Coarse map 1.0	3.5	1434	524	20
C	Greedy method	2.9	1271	432	27
	Information Gain 0.5	2.5	1004	342	19
	Information Gain 1.0	3.2	1264	429	20
	Coarse map 0.5	2.7	1250	466	22
	Coarse map 1.0	6.6	2966	1037	20
D	Greedy method	4.0	1684	572	32
	Information Gain 0.5	2.6	948	323	20
	Information Gain 1.0	3.6	1314	445	19
	Coarse map 0.5	3.0	1177	445	22
	Coarse map 1.0	3.5	1439	530	19

Table D.7: Results of the autonomous mapping of an ideal hole (90% exploration) for objective 3.

Start location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	8.5	4288	1462	61
	Information Gain 0.5	8.7	4159	1413	48
	Information Gain 1.0	18.6	9018	3035	46
	Coarse map 0.5	5.7	2832	1110	41
	Coarse map 1.0	9.8	5089	1837	37
B	Greedy method	7.1	3458	1184	58
	Information Gain 0.5	8.8	4170	1418	48
	Information Gain 1.0	16.9	8047	2710	44
	Coarse map 0.5	6.3	3046	1204	43
	Coarse map 1.0	10.5	5296	1915	36
C	Greedy method	8.0	3958	1352	59
	Information Gain 0.5	9.2	4128	1406	49
	Information Gain 1.0	18.0	8638	2906	45
	Coarse map 0.5	6.2	2981	1172	42
	Coarse map 1.0	12.0	5548	2012	38
D	Greedy method	8.4	4150	1416	60
	Information Gain 0.5	8.4	4009	1369	51
	Information Gain 1.0	19.0	9148	3078	46
	Coarse map 0.5	6.9	3287	1256	40
	Coarse map 1.0	9.6	4567	1665	36

Table D.8: Results of the autonomous mapping of an ideal hole (50% exploration) for objective 3.

Start location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	2.8	1621	552	27
	Information Gain 0.5	3.0	1653	560	21
	Information Gain 1.0	4.5	2453	826	20
	Coarse map 0.5	2.4	1408	524	21
	Coarse map 1.0	3.7	1988	707	20
B	Greedy method	2.4	1320	451	25
	Information Gain 0.5	2.5	1162	395	20
	Information Gain 1.0	3.3	1814	614	21
	Coarse map 0.5	2.5	1323	492	21
	Coarse map 1.0	3.7	1865	667	19
C	Greedy method	3.1	1639	557	26
	Information Gain 0.5	2.8	1410	479	21
	Information Gain 1.0	5.3	2488	837	21
	Coarse map 0.5	2.6	1445	533	20
	Coarse map 1.0	5.1	2447	864	19
D	Greedy method	2.9	1676	570	26
	Information Gain 0.5	2.6	1369	468	21
	Information Gain 1.0	4.1	2025	683	20
	Coarse map 0.5	2.3	1357	497	19
	Coarse map 1.0	4.8	2398	849	20

Table D.9: Results of the autonomous mapping of an ideal slope (85% exploration) for objective 3.

Start location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	7.5	2931	996	55
	Information Gain 0.5	6.7	2442	830	41
	Information Gain 1.0	12.6	4624	1559	39
	Coarse map 0.5	6.4	2274	844	39
	Coarse map 1.0 (85%)	9.2	3763	1332	34
B	Greedy method	7.1	2934	995	51
	Information Gain 0.5	6.4	2455	833	39
	Information Gain 1.0	10.7	4585	1545	36
	Coarse map 0.5	7.5	2948	1067	39
	Coarse map 1.0	8.6	3264	1171	35
C	Greedy method	6.9	2650	899	54
	Information Gain 0.5	6.6	2703	916	40
	Information Gain 1.0	12.7	4938	1662	38
	Coarse map 0.5	6.4	2230	828	38
	Coarse map 1.0	9.1	3336	1195	34
D	Greedy method	7.8	2980	1012	57
	Information Gain 0.5	7.9	2781	942	41
	Information Gain 1.0	11.3	4437	1495	37
	Coarse map 0.5	6.8	2395	883	40
	Coarse map 1.0	12.4	4599	1609	34

Table D.10: Results of the autonomous mapping of an ideal slope (50% exploration) for objective 3.

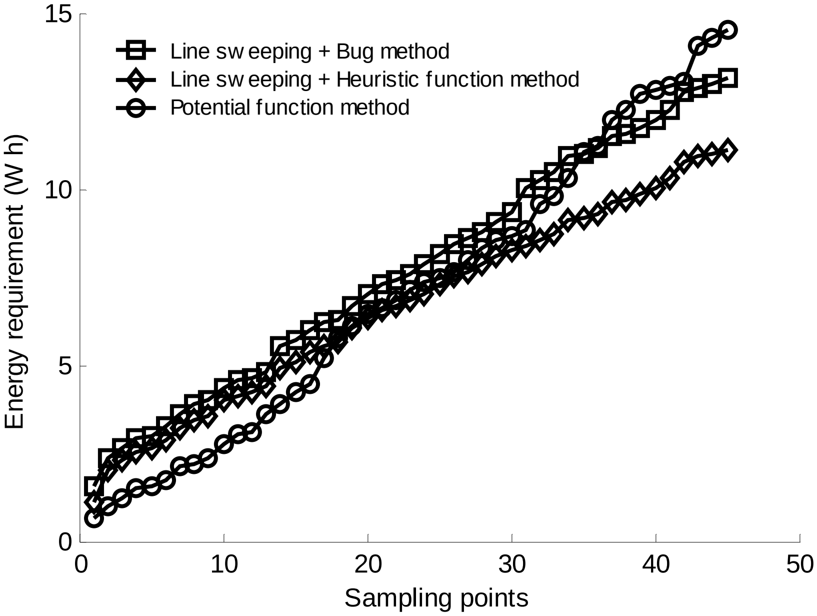
Start Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Greedy method	4.1	1541	521	29
	Information Gain 0.5	2.8	1069	362	21
	Information Gain 1.0	3.7	1223	414	20
	Coarse map 0.5	3.3	1250	444	22
	Coarse map 1.0	3.8	1651	576	19
B	Greedy method	2.9	1453	491	25
	Information Gain 0.5	3.2	1320	445	21
	Information Gain 1.0	4.3	1637	552	19
	Coarse map 0.5	2.6	1123	402	22
	Coarse map 1.0	3.6	1555	544	19
C	Greedy method	3.8	1487	502	29
	Information Gain 0.5	3.4	1116	377	21
	Information Gain 1.0	5.0	1775	597	19
	Coarse map 0.5	3.4	1084	389	21
	Coarse map 1.0	5.8	1888	656	19
D	Greedy method	3.9	1421	481	29
	Information Gain 0.5	4.6	1657	559	24
	Information Gain 1.0	3.7	1242	419	19
	Coarse map 0.5	3.6	1230	436	22
	Coarse map 1.0	5.8	2152	740	19

APPENDIX E - RESULTS OF THE SCOUTING ALGORITHMS

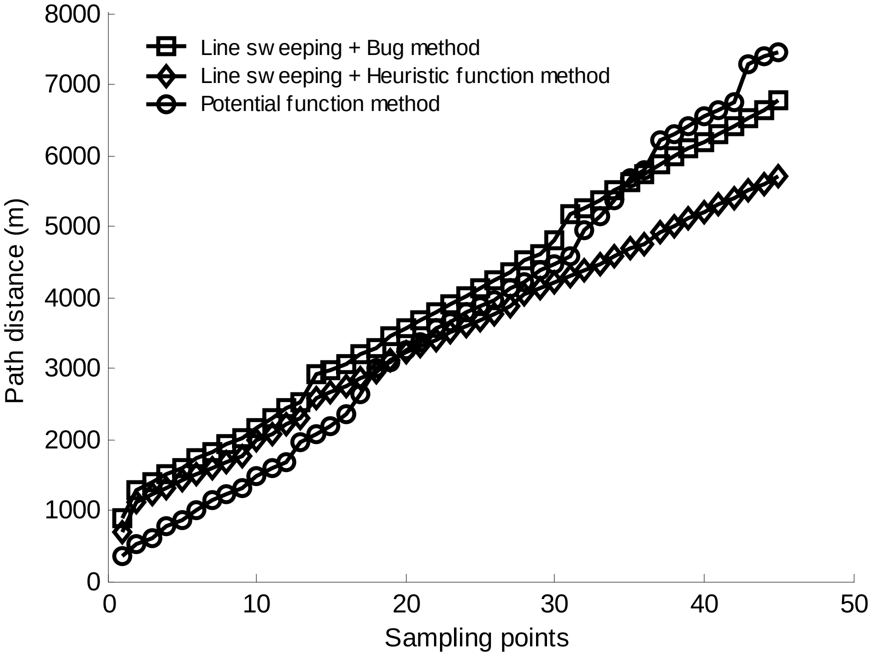
E.1 Plots of scouting algorithms

(1) Figures E.1 – E.3: Results of scouting algorithms for agricultural field 1,

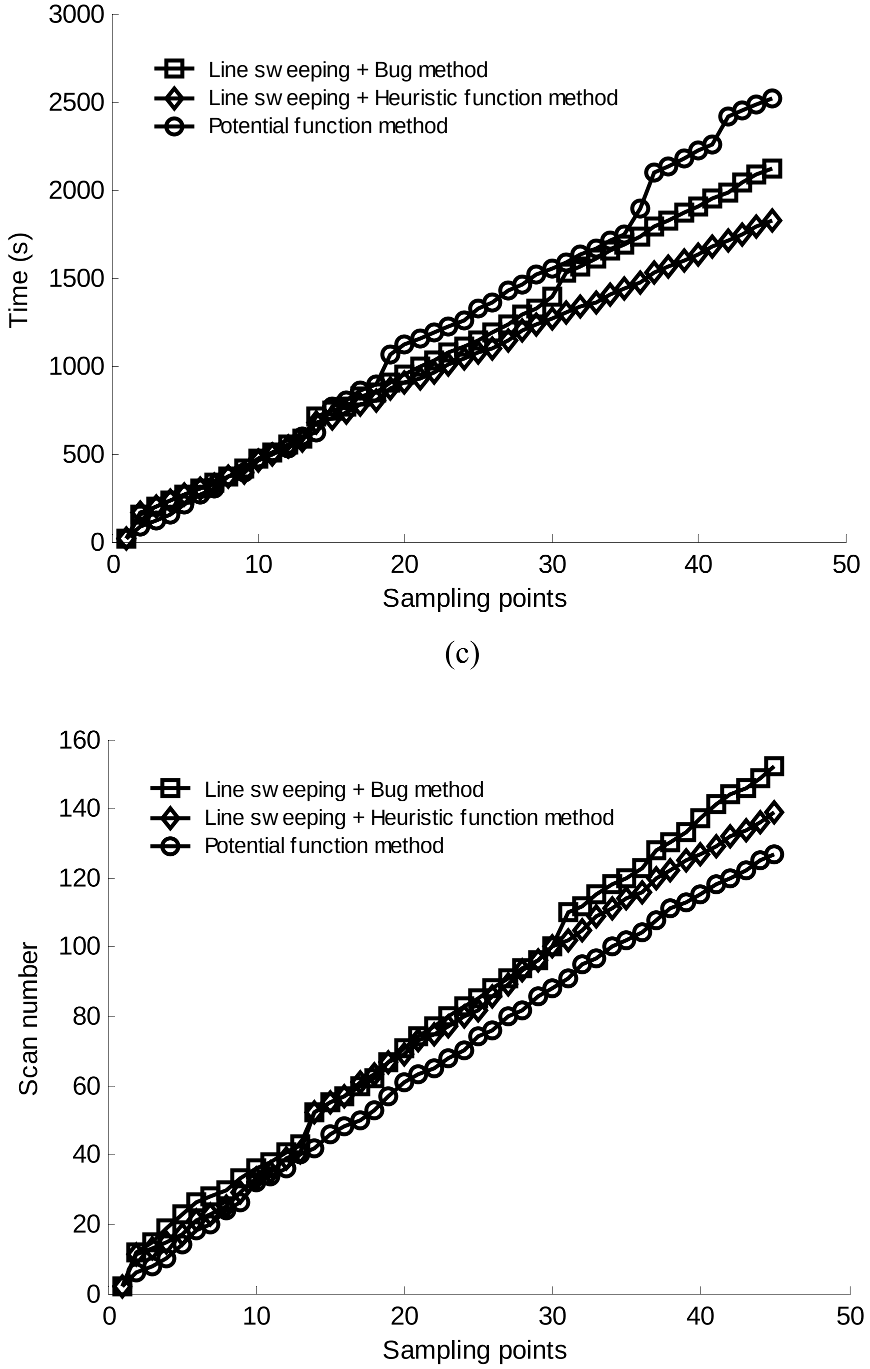
(2) Figures E.4 – E.7: Results of scouting algorithms for agricultural field 2.


15
10
5
0
0
10
20
30
40
50
Energy requirement (W h)
Sampling points
Line sw eeping + Bug method
Line sw eeping + Heuristic function method
Potential function method

(a)


8000
7000
6000
5000
4000
3000
2000
1000
0
0
10
20
30
40
50
Path distance (m)
Sampling points
Line sw eeping + Bug method
Line sw eeping + Heuristic function method
Potential function method

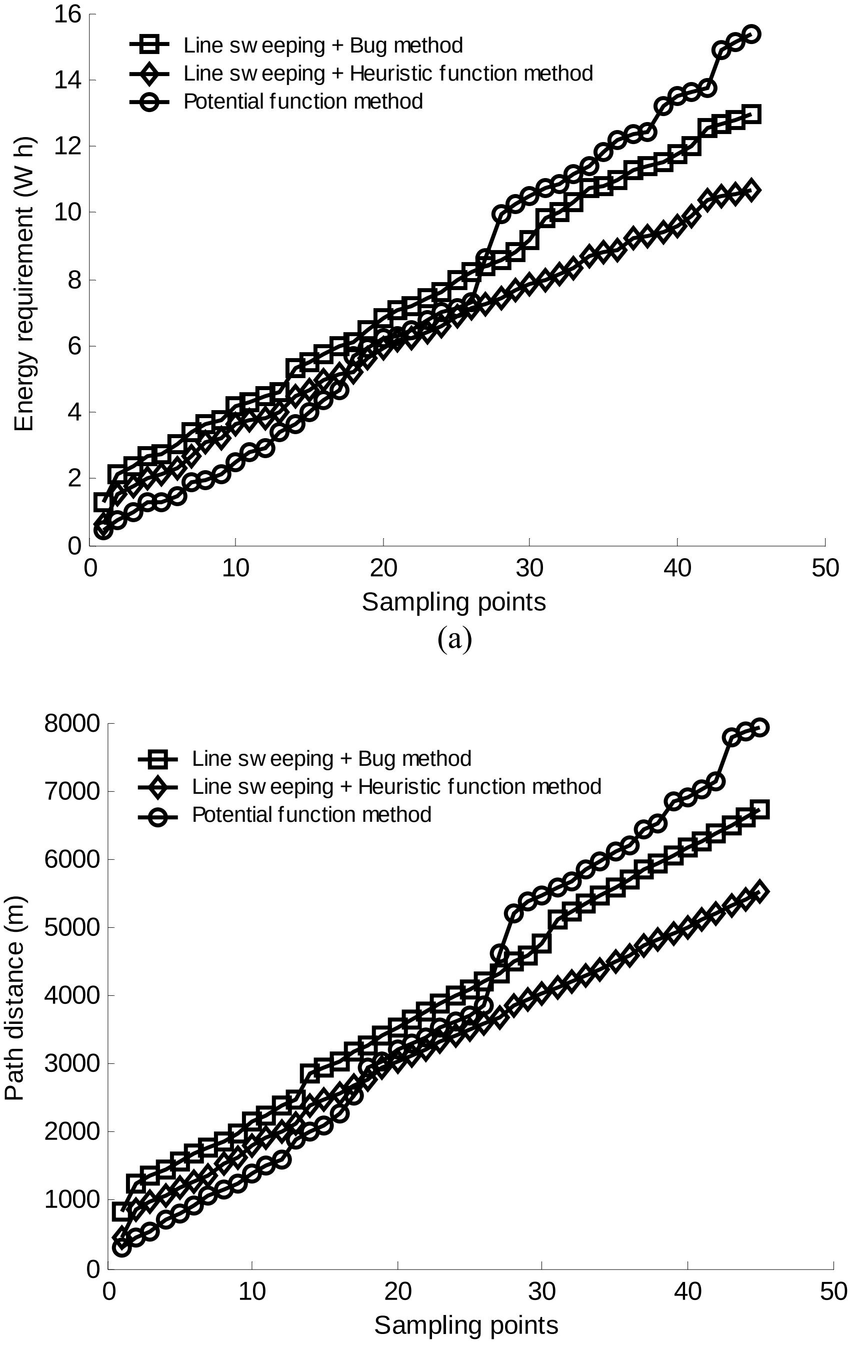
(b)



(c)

(d)

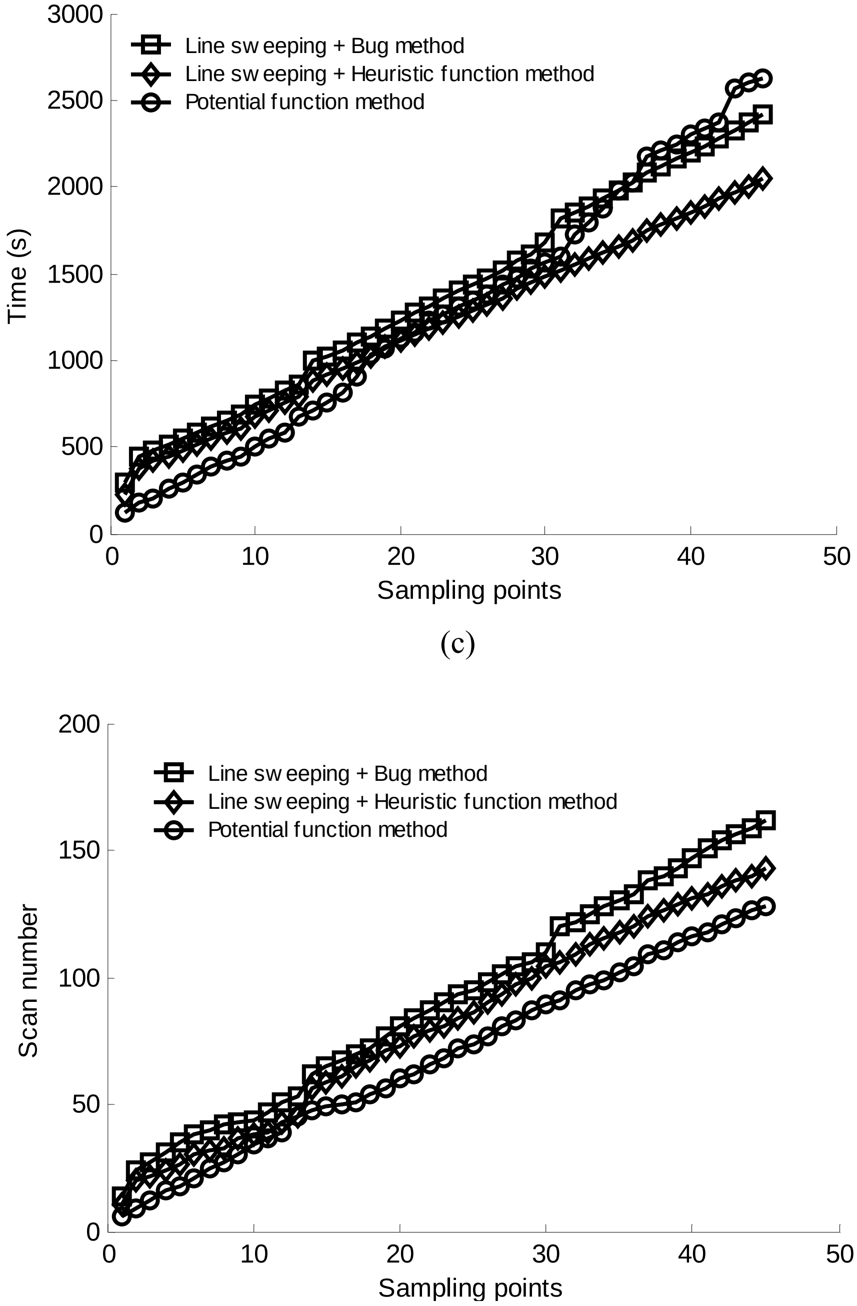
Figure E.1 Result of the scouting algorithm for agricultural field 1 (start B): (a) energy requirement as a function of sample points traversed, (b) path length traveled by the robot as a function of sample points traversed, (c) time required as a function of sample points traversed, (d) scan number as a function of sample points traversed.


16
14
12
10
8
6
4
2
0
Energy requirement (W h)
0
10
20
30
40
50
Sampling points
Line sw eeping + Bug method
Line sw eeping + Heuristic function method
Potential function method

(a)

8000
7000
6000
5000
4000
3000
2000
1000
0
Path distance (m)
0
10
20
30
40
50
Sampling points
Line sw eeping + Bug method
Line sw eeping + Heuristic function method
Potential function method

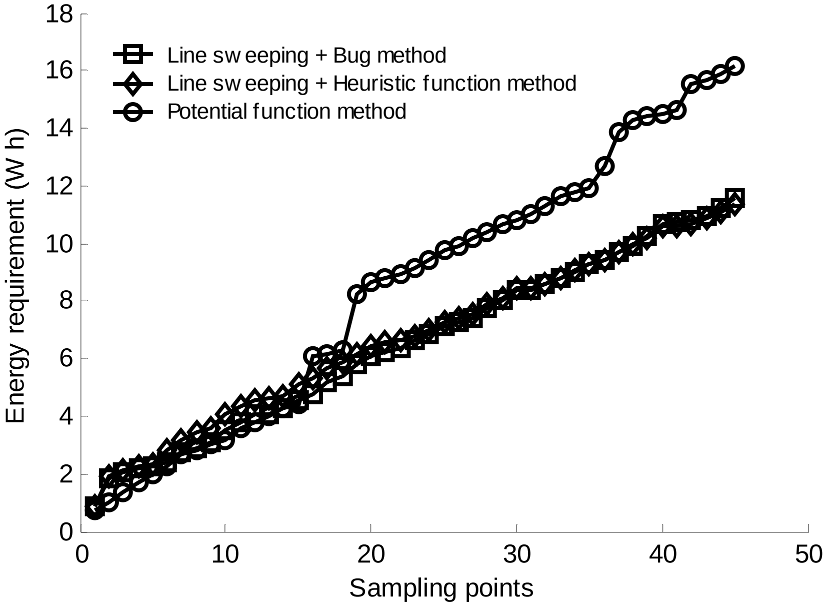
(b)



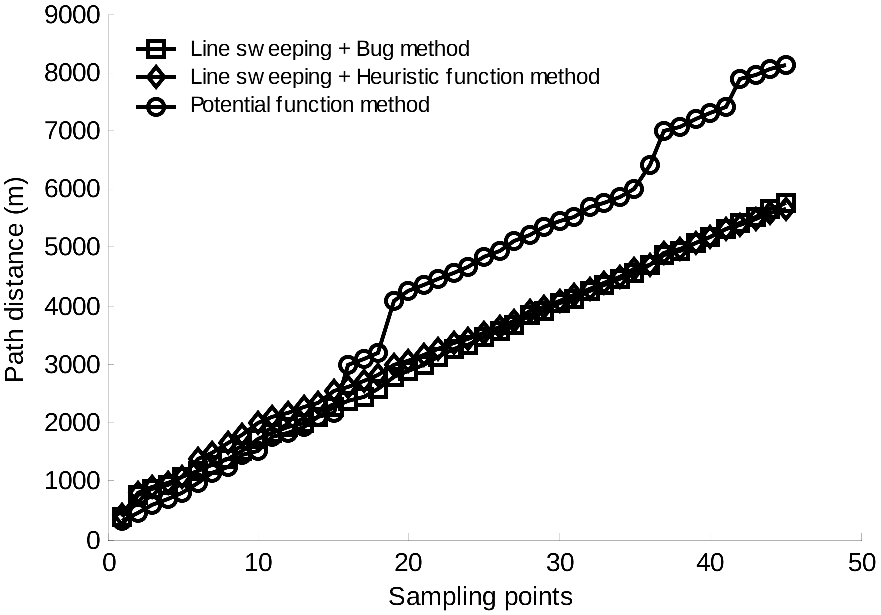
(c)

(d)

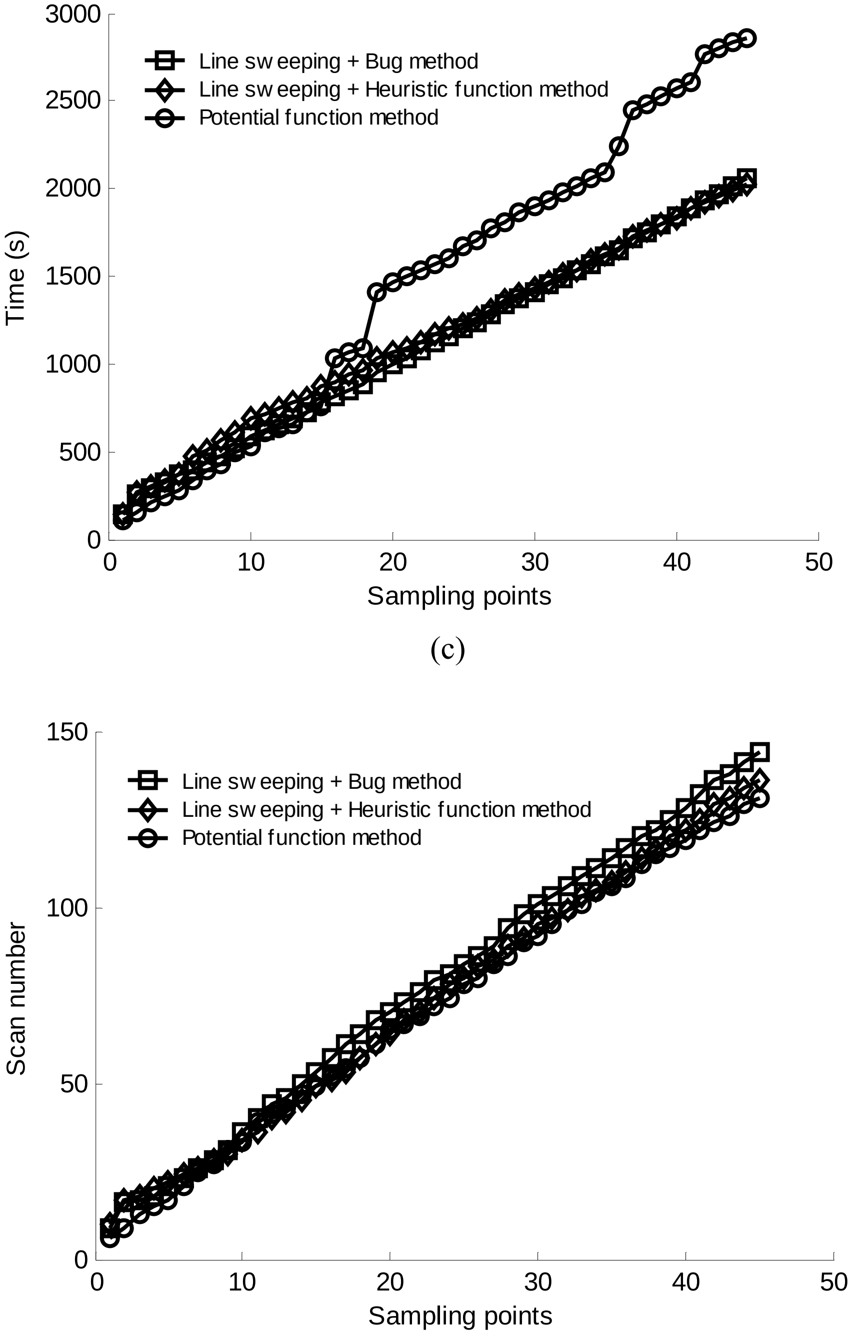
Figure E.2 Result of the scouting algorithm for agricultural field 1 (start B): (a) energy requirement as a function of sample points traversed, (b) path length traveled by the robot as a function of sample points traversed, (c) time required as a function of sample points traversed, (d) scan number as a function of sample points traversed.


18
16
14
12
10
8
6
4
2
0
Energy requirement (W h)
0
10
20
30
40
50
Sampling points
Line sw eeping + Bug method
Line sw eeping + Heuristic function method
Potential function method

(a)


9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Path distance (m)
0
10
20
30
40
50
Sampling points
Line sw eeping + Bug method
Line sw eeping + Heuristic function method
Potential function method

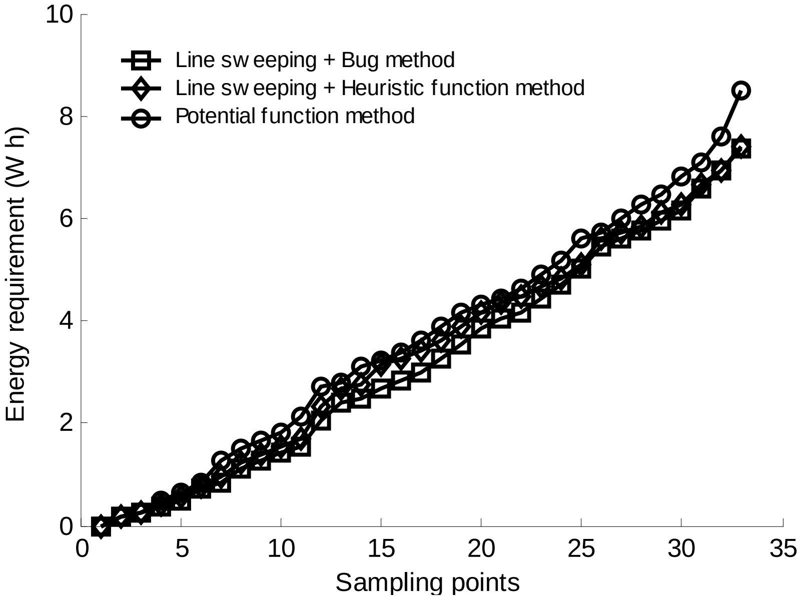
(b)



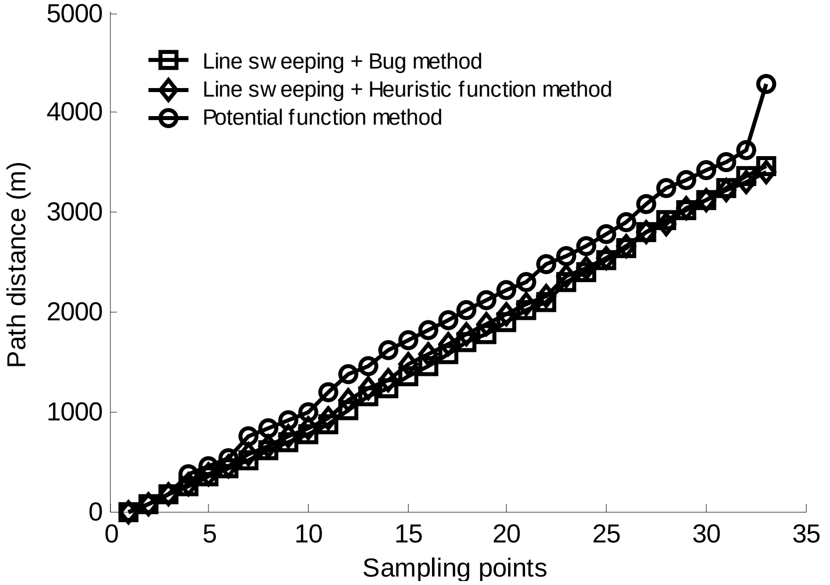
(c)

(d)

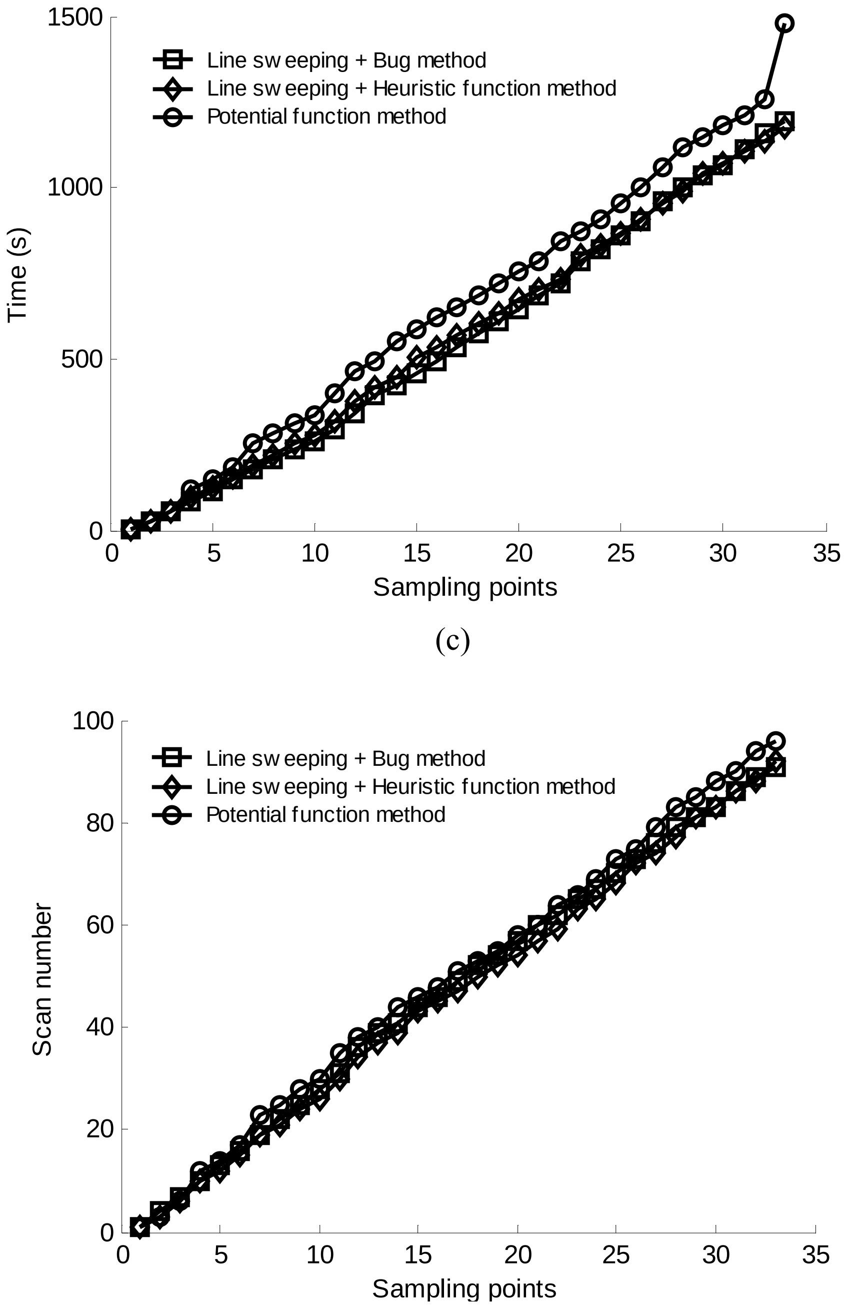
Figure E.3 Result of the scouting algorithm for agricultural field 1 (start D): (a) energy requirement as a function of sample points traversed, (b) path length traveled by the robot as a function of sample points traversed, (c) time required as a function of sample points traversed, (d) scan number as a function of sample points traversed.


10
8
6
4
2
0
Energy requirement (W h)
0
5
10
15
20
25
30
35
Sampling points
Line sweeping + Bug method
Line sweeping + Heuristic function method
Potential function method

(a)


5000
4000
3000
2000
1000
0
Path distance (m)
0
5
10
15
20
25
30
35
Sampling points
Line sweeping + Bug method
Line sweeping + Heuristic function method
Potential function method

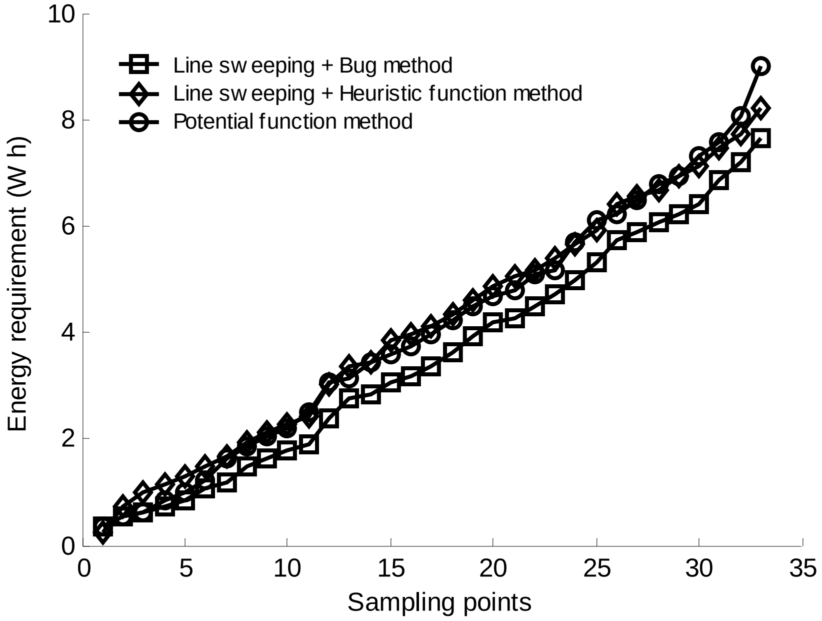
(b)



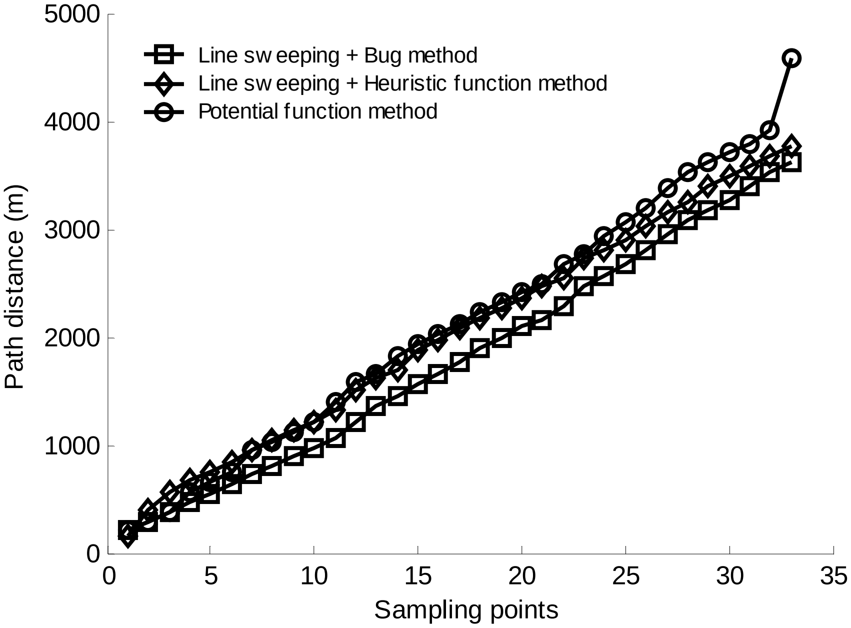
(c)

(d)

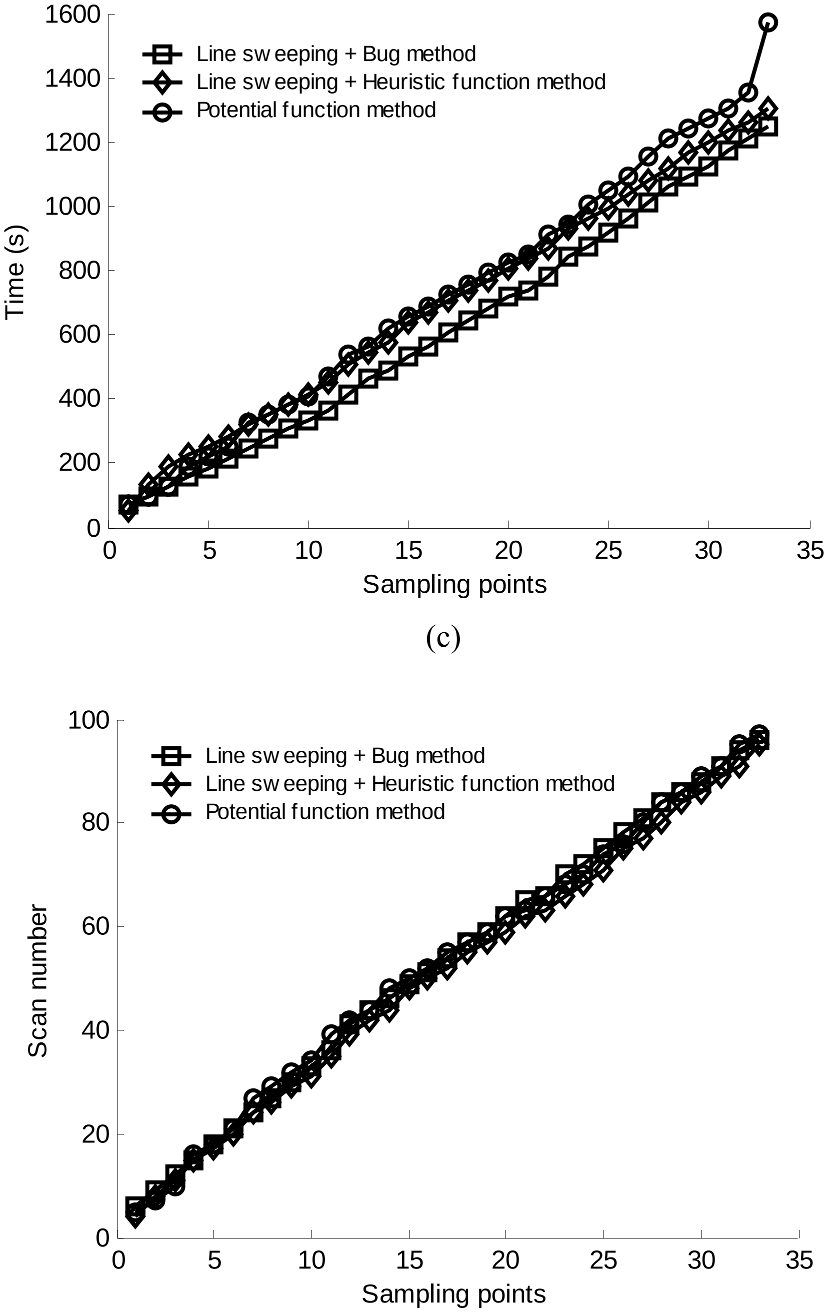
Figure E.4 Result of the scouting algorithm for agricultural field 2 (start A): (a) energy requirement as a function of sample points traversed, (b) path length traveled by the robot as a function of sample points traversed, (c) time required as a function of sample points traversed, (d) scan number as a function of sample points traversed.


10
8
6
4
2
0
Energy requirement (W h)
0
5
10
15
20
25
30
35
Sampling points
Line sw eeping + Bug method
Line sw eeping + Heuristic function method
Potential function method

(a)


5000
4000
3000
2000
1000
0
Path distance (m)
0
5
10
15
20
25
30
35
Sampling points
Line sw eeping + Bug method
Line sw eeping + Heuristic function method
Potential function method

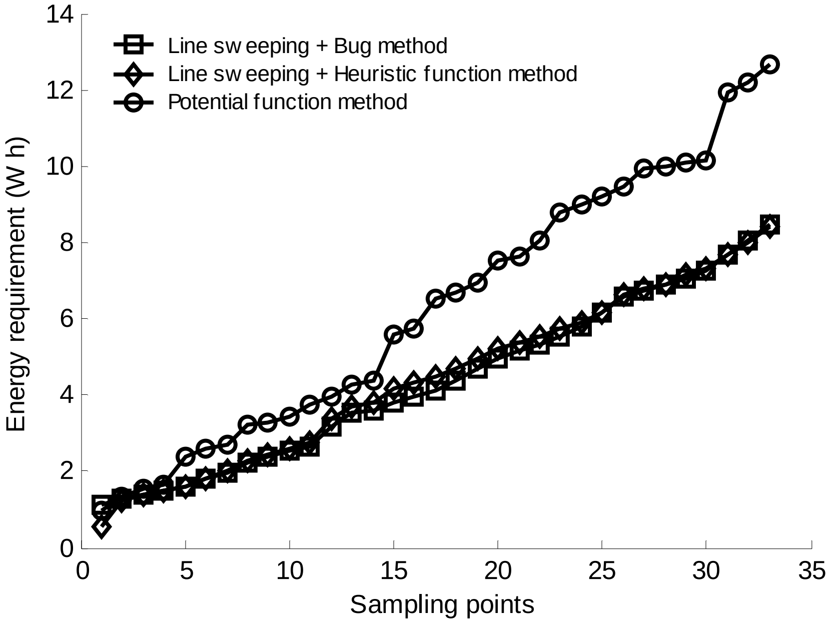
(b)



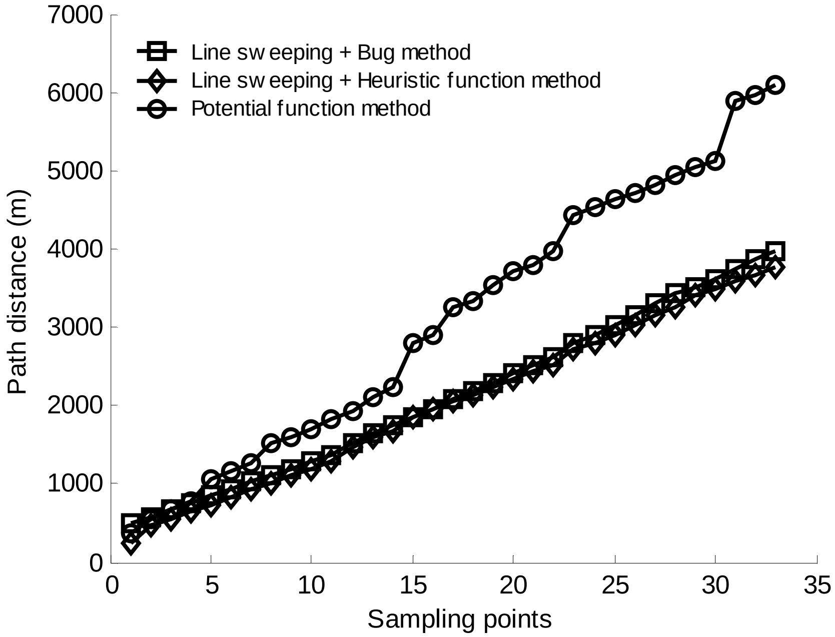
(c)

(d)

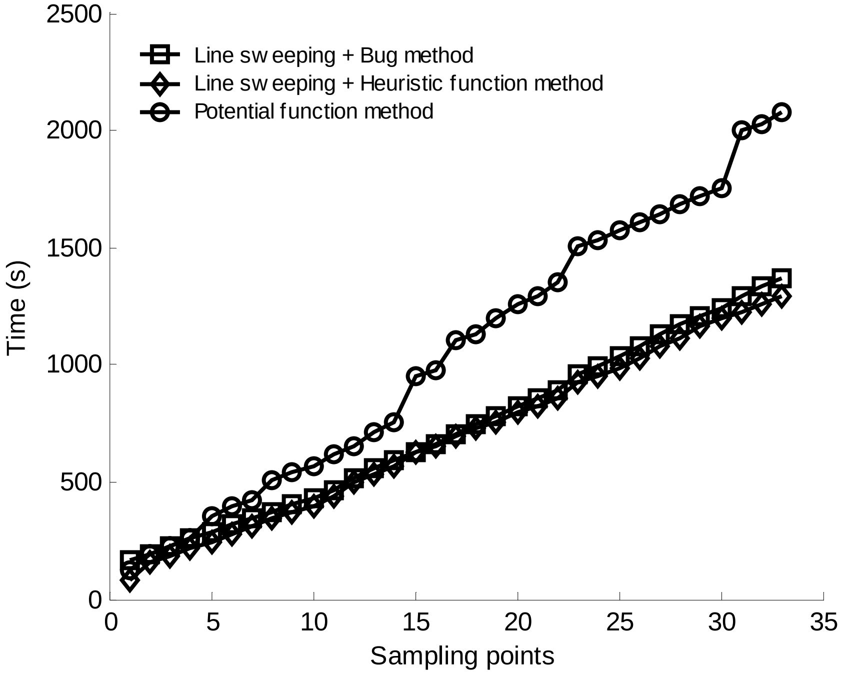
Figure E.5 Result of the scouting algorithm for agricultural field 2 (start B): (a) energy requirement as a function of sample points traversed, (b) path length traveled by the robot as a function of sample points traversed, (c) time required as a function of sample points traversed, (d) scan number as a function of sample points traversed.


Line sweeping + Bug method
Line sweeping + Heuristic function method
Potential function method
Energy requirement (W h)
Sampling points
0
2
4
6
8
10
12
14
0
5
10
15
20
25
30
35

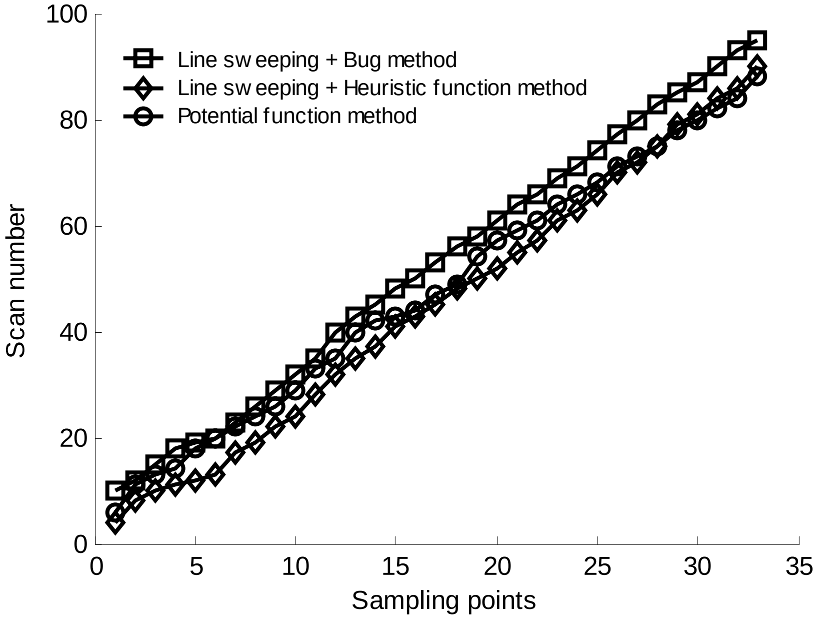
(a)


Line sweeping + Bug method
Line sweeping + Heuristic function method
Potential function method
Path distance (m)
Sampling points
0
1000
2000
3000
4000
5000
6000
7000
0
5
10
15
20
25
30
35

(b)

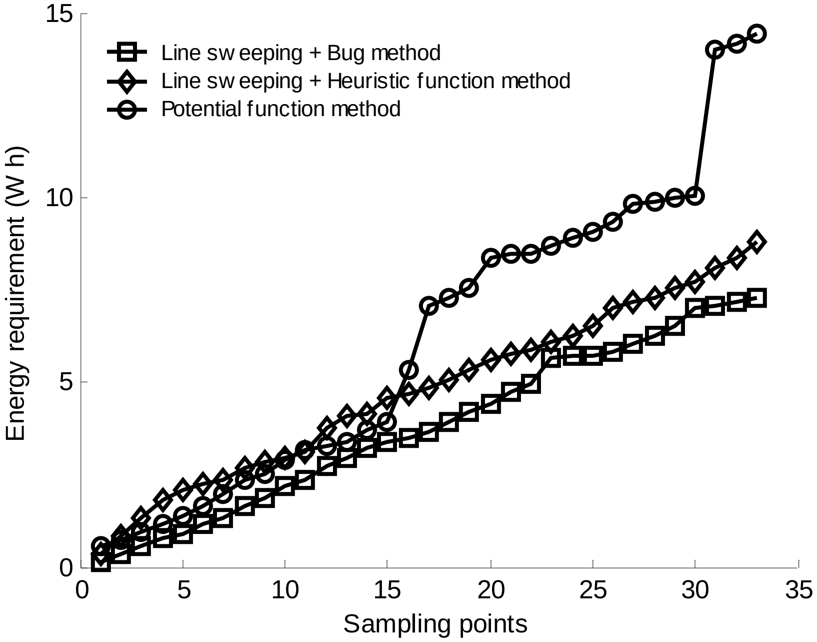


(c)

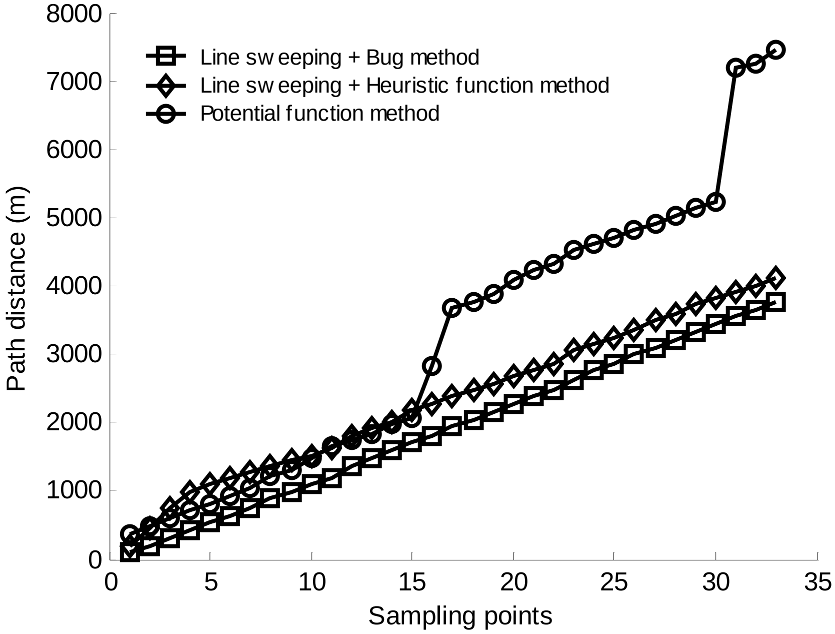


(d)

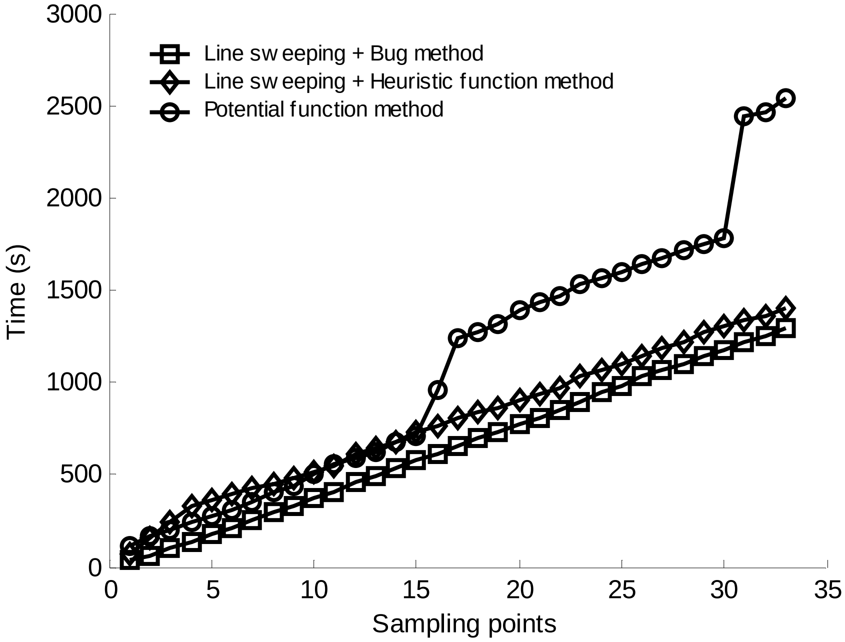
Figure E.6 Result of the scouting algorithm for agricultural field 2 (start C): (a) energy requirement as a function of sample points traversed, (b) path length traveled by the robot as a function of sample points traversed, (c) time required as a function of sample points traversed, (d) scan number as a function of sample points traversed.


15
10
5
0
Energy requirement (W h)
0
5
10
15
20
25
30
35
Sampling points
Line sw eeping + Bug method
Line sw eeping + Heuristic function method
Potential function method

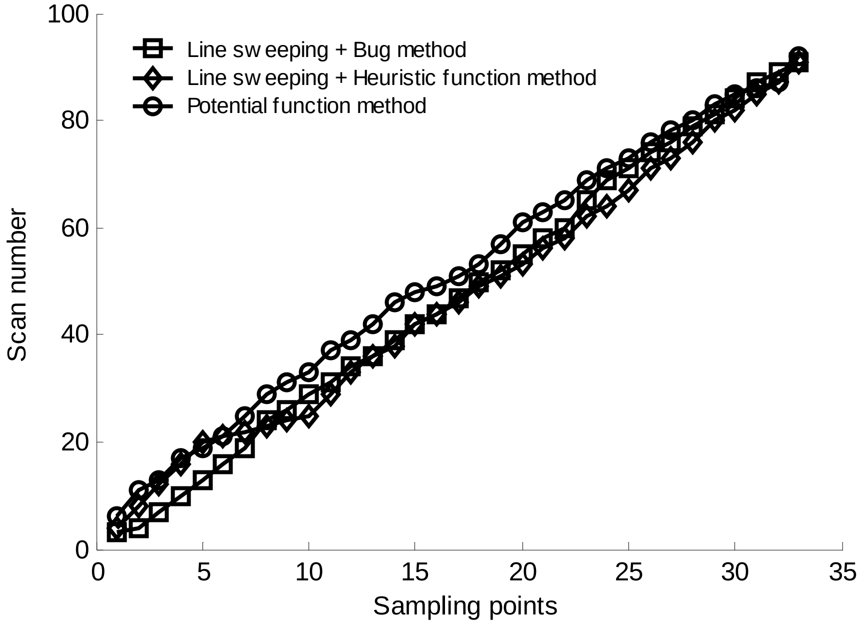
(a)


8000
7000
6000
5000
4000
3000
2000
1000
0
Path distance (m)
0
5
10
15
20
25
30
35
Sampling points
Line sw eeping + Bug method
Line sw eeping + Heuristic function method
Potential function method

(b)



(c)



(d)

Figure E.7 Result of the scouting algorithm for agricultural field 2 (start D): (a) energy requirement as a function of sample points traversed, (b) path length traveled by the robot as a function of sample points traversed, (c) time required as a function of sample points traversed, (d) scan number as a function of sample points traversed.

E.2 Tables of scouting algorithms for objective 4

(1) Table E.1: Results of scouting algorithms for agricultural field 1,

(2) Table E.2: Results of scouting algorithms for agricultural field 2,

(3) Table E.3: Results of scouting algorithms for an ideal mountain,

(4) Table E.4: Results of scouting algorithms for an ideal hole environment,

(5) Table E.5: Results of scouting algorithms for an ideal slope environment.

Table E.1 Results of scouting algorithms for agricultural field 1

Start Location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Bug	11.7	5940	2123	152
	Heuristic	10	5066	1825	139
	Potential field	14.1	7102	2512	127
B	Bug	13.1	6765	2416	162
	Heuristic	12.4	6226	2350	150
	Potential field	14.5	7462	2630	128
C	Bug	12.9	6732	2393	162
	Heuristic	12	6096	2302	155
	Potential field	15.4	7939	2785	129
D	Bug	11.6	5783	2059	144
	Heuristic	12.5	5995	2278	158
	Potential field	16.2	8131	2862	131

Table E.2: Results of scouting algorithms for agricultural field 2

Starting location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Bug	7.4	3461	1191	91
	Heuristic	7.4	3406	1174	92
	Potential field	8.5	4295	1480	96
B	Bug	7.6	3625	1248	96
	Heuristic	8.2	3774	1302	95
	Potential field	9.0	4580	1575	97
C	Bug	8.5	3966	1366	95
	Heuristic	8.4	3769	1295	90
	Potential field	12.7	6085	2077	88
D	Bug	7.3	3753	1291	91
	Heuristic	8.8	4103	1403	91
	Potential field	14.4	7453	2535	92

Table E.3: Results of scouting algorithms for an ideal mountain environment.

Starting location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Bug	4.3	1856	630	39
	Heuristic	5.0	2198	744	45
	Potential field	6.5	2889	977	45
B	Bug	5.4	2491	846	62
	Heuristic	4.7	2401	813	47
	Potential field	6.9	3248	1095	44
C	Bug	4.6	2147	727	36
	Heuristic	4.8	2325	786	41
	Potential field	7.4	3094	1046	44
D	Bug	4.5	1975	669	36
	Heuristic	5.5	2346	793	41
	Potential field	8.4	3852	1297	38

Table E.4: Results of scouting algorithms for an ideal hole environment.

Starting location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Bug	3.7	1840	623	37
	Heuristic	4.5	2157	730	40
	Potential field	5.0	2537	858	40
B	Bug	4.4	2035	690	41
	Heuristic	5.3	2388	808	43
	Potential field	5.9	2767	935	41
C	Bug	4.5	2089	706	34
	Heuristic	5.1	2265	766	38
	Potential field	7.5	3498	1179	37
D	Bug	4.2	1975	671	34
	Heuristic	5.1	2487	839	37
	Potential field	6.3	3059	1031	37

Table E.5: Results of scouting algorithms for an ideal slope environment.

Start location	Method	Energy (W h)	Path length (m)	Time (s)	Scan number
A	Bug	4.8	1632	551	40
	Heuristic	5.5	1763	595	45
	Potential field	6.0	1868	632	45
B	Bug	4.7	1875	632	43
	Heuristic	5.7	2497	841	50
	Potential field	5.5	2091	706	44
C	Bug	5.3	1847	623	40
	Heuristic	5.9	1914	644	42
	Potential field	7.8	3460	1161	39
D	Bug	5.2	1705	575	38
	Heuristic	6.2	2006	675	43
	Potential field	7.2	2972	999	39

APPENDIX F - LIST OF JAVA CLASSES DEVELOPED

All classes, attributes and methods are commented out with valid Javadoc Code.

F.1 Package coverageplanner

F.1.1 Class summary

Camera	This class is used to simulate an image sensor such as a camara or lindar.
CoverageMap	This class is used to represent the triangular mesh map.
CoveragePlanner	This class is used to plan the next best viewpoints for the exploration task.
CoveragePlannerRayTracer	This class is used to plan the next best viewpoints for the exploration task by using the ray tracing algorithm.
DijkstraCoveragePath	This class is to implement the search algorithm
Frontier	This class is used to deal with the frontier.
GVertex	This class is used to represent the points in the coarse map known a priori.
ImagingSensor	Replaced by camera class.
IterativeMap	This class is used to deal with the iterative map by integrating every sensor reading.
LinearRegress	This class is used to deal with linear regress with points. It is not used in the final project.
MobilityEdge	This class represent a triangle to trangle “neighour” relationship in the graph representation of the map used by the path planner (imported from CSA).
MobilityEdgeFactory	This class is used in the graph to instanciate MobilityEdge for the path planning (imported from CSA).
MyPolygon	This class is used to represent a polygon in double Data format
MyTriangle	This class is used to represent basic triangle cell in a triangular mesh map.
MyVertex	This class is used to represent a vertix for a triangle cell.
p2atMobilityCostFunction	Provide traversability cost calculation (imported from CSA).
RayTracer	This class is used to deal with ray tracing algorithm.
SamplingPoint	This class is used to represent sampling point for the statistical coverage planner.
StatisticalCoveragePlanner	This class is used to plan next best viewpoints for the sampling task.
XYZIO	This class is used to deal with data input for coverage planner (imported from CSA).

F.1.1.1 Camera class

Camera(Point3d _eye, Point3d _center, double _vertical_fov, double _aspect, double _znear, double _zfar)	Camera constructor
Camera(Pose3D _pose, Point3d _center)	Camera constructor
drawFrustum()	Display the viewing frustum
estimateNewSeenCellsByFrustum(CoverageMap _map)	Estimate the visible triangles by the camera frustum
estimateNewSeenCellsByRayTracer(CoverageMap _map)	Estimate the new visible triangles by ray tracing algorithm
getFootPrint()	Get 2D footprint
getFrustum()	Get the viewing frustum
getFrustum2D()	Get the projected viewing frustum
getHorizontalFOV()	get the horizontal field of view
getLocation()	Get the camera location
getLookAt()	Get the view center
getMaxRange()	Get the maximum range of the camera
getMinRange()	Get the minimum range of the camera
getOldInformation(CoverageMap _map)	Get the 2D visited area
getPose()	Get the camera pose
getPosition()	Get the camera position
getTransform()	Get the transform matrix
getVerticalFOV()	Get the vertital field of view
getVisibleTriangles(CoverageMap _map)	Get the visible triangle list
LookAt(javax.vecmath.Point3d _aPoint)	Take a view action at the specific location
projectionToWorld(javax.vecmath.Point3d _pt)	Tranform the perspective view to the world point
setFocalPoint(javax.vecmath.Point3d _pt)	Set the view center of the camera
setHorizontalFOV(double _horizontal_fov)	Set the horizontal field of view
setMaxRange(double _max_ra	Set the maximum range of the camera

nge)	
setMinRange(double _min_range)	Set the minimum range of the camera
setPitch(double _angle)	Set the pitch angle of the camera pose
setPose(ca.gc.space.mrt.devices.utils.Pose3D _pose)	Set the camera's pose
SetPosition(javax.vecmath.Point3d _aPoint)	Set the camera position
setRoll(double _angle)	Set the roll angle of the camera pose
setVerticalFOV(double _vertical_fov)	Set the vertical field of view
setYaw(double _angle)	Set the Yaw angle of the camera pose
updatePose(Pose3D _pose, Point3d _center)	Update the pose of the camera
worldToProjection(Point3d _pt)	Tranform the world point to the perspective view

F.1.1.2 CoverageMap class

CoverageMap(vtkPolyData _mesh)	Constructor
ccw(Point3d p, Point3d q,Point3d r)	A function to test whether three successive points are in a counter-clock wise direction
createGlobalTriangleList()	Create global triangle list
display_vtkBoundary(UndirectedGraph _graph)	Display the boundary of the map
displayBoundaryVertex(Vector _boundary_list)	Display the boundary vertex
displayFrontierLineList(java.util.Vector<Frontier> _frontier_list)	Display the frontier list
displayFrontierList(java.util.Vector<Frontier> _frontier_list)	Display the frontier list
displayPathList(java.util.List _path_list)	Display the path list
displayTriangleList(java.util.Vector _triangle_list)	Display the triangle list
find_Next_Vertex(MyVertex _previous, MyVertex _pivot, MyVertex[] _v)	Search next vertex having the largest angle with the previous edge
GetCellNeighbors(int _cell_id)	Returns the neighbors of the given cell: the

	neighbor cell here is a cell that shares at least one common vertex with a given cell
getFieldBoundary()	Get the field bounday defined a priori
getFrontierList()	Get the frontier list
getGlobalCellList()	Get the global triangle list
getHolesList()	Get holes list of the map
getMapTriangleForPosition (Point3d _point)	Returns the trangle in the map for a specified (X,Y) position
getNumberofHoles()	Get the number of the holes in the map
getOuterBoundaryVertex()	Get the outer bounday vertex list in this map
getSeenCellList()	Get the known cells in the map
line_LineAngle(double[] pt1, double[] pt2, double[] pt3)	return the angle between line p2p1 and line p2p3
setCostFunction(ICostFunction _ cost_function)	Sets the cost function to be used by the planner.
setCoverageBoundary(MyPolygon _boundary)	Set the boundary of this map
setFieldBoundary(Point3d[] _boundary)	Set field bounday
simulationCreateGraph()	Create the graph for simulation
simulationCreateGraph12Neighbors()	Create 12-neighbors graph for simulation
vertex_points_orientation(int n, Point3d[] v)	A function to test whether three successive points are in a counter-clock wise direction or not

F.1.1.3 Coverageplanner class

CoveragePlanner()	Constructor
CoveragePlanner(CoverageMap _coverage_map, IImagingSensor _sensor, ICostFunction _cost_function, ILogStorage _log_storage, java.lang.String _hierarchy)	Constructor
clrOccludedState()	Clear the occluded state
computeTraj(Pose3D _start, Pose3D _end)	Compute the path between two locations
contains(int[] array, int j)	Check wheather the index has appeared in the array
createGraph()	Create Graph fom the current coverage map
extractOuterFrontierList()	Extract the frontier list in the first stage
getAlgStage()	Get the coverage planner stage
getCoveredPercentage()	Return the percentage of the coverage area
getFrontier()	Return the current frontier
getFrontierList()	Get the frontier list

getFrontierPath()	Get the path list to next frontier
getFrontierTriangle()	Get the frontier triangle
getHolesList()	Get all the hole found in the map
getHoleVertexList(int _index)	Return one hole by its index number
getInformationGain(Pose3D _camera_pose, Point3d _view_center)	Estimate the information gain for the specific frontier
getMapTriangleForPosition(Point3d _point)	Returns the trangle in which contains the specified point.
getMaxUtilityHoleIndex()	Get the current frontier hole index
getNavigationTrajectory(Pose3D _current_location, Pose3D _destination)	Gets a navigation trajectory that the robot can follow between two points
getOuterBoundary()	Return outer boundary
getPath(Pose3D _current_location, Pose3D _destination)	Get the path from one location to another location
getPathCost()	Get the path cost fron current robot's position to next frontier
getPointPath(java.util.Vector _path)	Return point path from triangle path list
getRobotPosition()	Return the robot's position
getRobotTriangle()	Get the triangle where the robot locates
getSecondStageFrontierPathV2()	Get path list to next frontier in new version
getSimulationMapBoundary()	Update the map boundary
getTrajectory(Vector _path)	Return trajectory from path list
getUtility()	Get the utility of the frontier
getViewCenter()	Get view center for next scan
initMap(double _radius)	Initialize the map before the exploration task starts
MobilityCostFunction(ICostFunction _mobility_cost_function)	Set cost function
retractInnerFrontierList()	Extract frontier list in the second stage
retractOuterFrontierList()	Extract frontier list in the first stage
rotate2D(Point3d _pt, Point3d _pivot)	Rotate one point by 90 degrees around a pivot point
rotate2DByDegree(Point3d _pt, Point3d _pivot, double _angle)	Rotate one point by an angle around a pivot point
scan()	Scan with current sensor and update the map
setAlgStage(int _stage)	Set the algorithm stage
setCostFunction(ICostFunction _cost_function)	Set the cost function used by the planner
setFrontier(Frontier _fr)	Set frontier
setFrontierTriangle(MyTriangle _triangle)	Set Frontier triangle

setImagingSensor(IImagingSensor _imaging_sensor)	Set the current sensor
setInformationWeight(double _a)	Set the information weight factor
setPathList(List _path)	Set the path list for next step
setPathPlanner(IPathPlanner _path_planner)	Set path planner
setRobotMobiltyCostFunction(ICostFunction _mobility_cost_function)	Sets the cost function.
setRobotPosition(Point3d _pt, MyTriangle _triangle)	Set the robot's position
setTerrainMap(CoverageMap _terrain_map)	Set the terrain map
setViewCenter(Point3d _pos)	Set the view direction of coverage planne
updateSubGraph()	Update the graph when new information is integrated into the map

F.1.1.4 CoveragePlannerRayTracer class

get2DArea(Vector<MyTriangle> _triangle_list)	Get the projected area in the triangle list
getFrontierPath()	Get the path list to next frontier
getNewInformationGain(Pose3D _camera_pose, Point3d _view_center)	Estimate the information gain for the frontier
getSecondStageFrontierPathV2()	Get path list to next frontier in new version
initMap(double _radius)	Initialize the map before the task start
setCoarseMap(IterativeMap _map)	Set the initial coarse map
updateCoarseMap()	Update the coarse map

F.1.1.5 DijkstraCoveragePath class

DijkstraCoveragePath()	Constructor
findFrontierListShortestPath(Graph graph, Object _start_vertex, Vector _end_list)	Find the shortest paths from start vertex to frontiers
findNearestFrontier(Graph graph, Object _start_vertex, Vector end_list)	Find the shortest path between two vertices, represented as a List of Edges
findPathBetween(Graph graph, Object startVertex, Object endVertex)	Find the shortest path between two vertices, represented as a List of Edges
findShortestPath(Graph graph, Object _start_vertex, Vector _end_list)	the shortest path from start vertix to the nearest frontier

F.1.1.6 Frontier class

compareTo(Frontier _other)	Compare with another frontier
getCenter()	Get the center of this frontier
getCost()	Get the travel cost to this frontier
getDirection()	Get the view direction of this frontier
getPathList()	Get the path list to this frontier
setDirection(Point3d _pt)	Set the view direction for this frontier
setInfo(double _info)	Set the information gain in this frontier
setPathList(Vector _path)	Set the path list to this frontier
setPolygonIndex(int _index)	Set the polygon index

F.1.1.7 Gvertex class

GVertex()	GVertex default constructor
GVertex(double[] x, int _id)	GVertex constructor
GVertex(double x, double y, double z)	GVertex constructor
GVertex(double x, double y, double z, int _id)	GVertex constructor
clearVisitState()	Clear visit state
getVisitState()	Get the visit state
setVisitState()	Set the visit state as true
toString()	Get the string for display

F.1.1.8 IterativeMap class

IterativeMap()	Creates a new instance of IterativeMap
IterativeMap(vtkPolyData _mesh)	Creates a new instance of IterativeMap
createGlobalTriangleList()	Create the global triangle list
getCoarseVertices()	Return the points in the original coarse map
getGVertex(MyVertex _vertex)	Return the global vertex id
getMesh()	Reutturn the mesh
getSeenTriangleList()	Get the seen cell list
getUnkownVertices()	Get the unvisited vertex list
getVertexID(MyVertex _vertex)	Return the global vertex id
initVertexList(vtkPolyData _mesh)	Create global vertex list
update(CoverageMap _map)	Insert a new sensor reading

F.1.1.9 MyPolygon class

MyPolygon()	Creates a new instance of MapPolygon
MyPolygon(double[] xpoints, double[] ypoints, double[] zpoints, int npoints)	Constructor
MyPolygon(Point3d[] _pt)	Constructor
MyPolygon(Vector _polygon)	Constructor
area()	Compute signed area of polygon
boundingBoxContains(double _x, double _y)	Return whether the 2D piont locates inside the bounding box
centroid()	Compute the centroid of the polygon
contains(double x, double y)	Determines if the specified coordinates are inside this Polygon
getBoundary()	Return the polygon boundary
perimeter()	Return the polygon 's perimeter

F.1.1.10 MyTriangle class

MyTriangle()	Constructor
MyTriangle(int _cell_id)	Constructor
MyTriangle(int _cell_id, vtkPolyData _mesh)	Constructor
clearHitState()	Clear the hit state of this triangle
clearOccludedState()	Clear the occluded state
compareTo(java.lang.Object o)	Compare a triangle with another triangle
containsEdge(MyVertex _vertex1, MyVertex _vertex2)	Test if a specified vertex is equal to one of the triangle vertices.
containsVertex(MyVertex _vertex)	Test if a specified vertex is equal to one of the triangle vertices.
decreaseShadowNumber()	Decrease the shadowed times by the raytracer algorithm
equals(Object _other)	Compare with another triangle
getAbsoluteCenter()	Return the triangle's center point in absolute coordinates.
getAbsoluteNormal()	Returns the triangle's normal in absolute coordinates
getArea()	Get the area of this triangle
getId()	Get the id of this triangle
getArea2D()	Get the area projection in xy plane
getCenter()	Get the triangle center

getDistance(MyTriangle _triangle)	Get the distance to another triangle
getShadowNumber()	Get the shadowed times of this triangle
getVertex(int _vertex_index)	Get one vertex of this triangle
getVisitState()	Get the visit state of this triangle
hashCode()	Get the hashCode of this class
increaseShadowNumber()	Count the shadowed times by the raytracer
intersect(Vector3d _start, Vector3d _end)	Calculate the intersection point with a segment
setHitState()	Set the hit state
setMap(vtkPolyData _mesh)	Set the mesh which the triangle belongs to
setOccludedState()	Set the occluded state
setVisitState(int _state)	Set the visit state of this triangle
toString()	Get the string of this class for display

F.1.1.11 MyVerex class

MyVertex()	Default constructor
MyVertex(double[] p)	Constructor
MyVertex(double x, double y, double z, int _id)	Constructor
equals(MyVertex _other)	Compare two MyVertex objects
hashCode()	Get the hashCode of this class
toString()	Get the string for display

F.1.1.12 Ray tracer class

RayTracer(Transform3D _transform, Point3d _eye, double _z)	Constructor: Initializes an instance of the ray tracer with the window size and the world description
RayTracer(Transform3D _transform, Point3d _eye, Vector _trianglelist, double[] _range, double _step, double _z)	Constructor
estimateSeenCells()	Compute the visible triangle cells
getRenderPixel(int i, int j)	Get the world coordinate of the specifix pixel in the screen
projectionToWorld(Point3d _pt)	Transform the pixel to world coordinate
render()	Establish the viewing matrix, set up the transfer from screen space to world space, and call the render pixel function

F.1.1.13 SamplingPoint class

SamplingPoint()	Constructor
SamplingPoint(int _row_id, int _colomn_id, Point3d _pt)	Constructor
equals(Object _other)	Compare two SamplingPoint objects
getLocation()	Get the location(x,y,z) of this sampling point
getColomnId()	Get the column id of this sampling point
getRowId()	Get the row id of this sampling point
getVisitState()	Get the visit state of this triangle
setVisitState(int _state)	Set the visit state of this triangle

F.1.1.14 StatisticalCoveragePlanner class

StatisticalCoveragePlanner()	Constructor
StatisticalCoveragePlanner(CoverageMap _coverage_map, ICostFunction _cost_function)	Constructor
StatisticalCoveragePlanner(CoverageMap _coverage_map, IImagingSensor _sensor, ICostFunction _cost_function, ILogStorage _log_storage, String _hierarchy)	Constructor
getNextBestView()	Return next best viewpoint
getNextSubGoal()	Get next sampling point in the sample list
getOuterPoints()	Get the outer sampling cell list
getPath(Point3d _current_location, Point3d _destination)	Get the path between two points
getPath2NextBextView(SamplingPoint _subgoal)	Get the path to next best viewpoint
getSampleCell(int _row, int _column)	Return the specific sampling cell
getSamplePoints()	Return sampling points by statistical coverage circle
getStatCoverageRadius()	get statistical coverage radius
setStatCoverageRadius(double _radius)	Set statistical coverage radius