Autonomous Surface Exploration for Mobile Robots

Stewart John Moorehead

CMU-RI-TR-01-30

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Robotics

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, Pennsylvania 15213

August 13, 2001

Abstract

Exploration gathers information about the unknown. This information can come in many forms, from knowledge of new terrain, to rock geology, to lifeforms. The value of these different information forms to an explorer is determined by a set of information metrics, one for each form of information, that depend on the goal of the exploration task. As explorations become more complex, increasing numbers of information metrics must be considered in order to succeed. These multiple information metrics must be considered simultaneously during exploration and often conflict with each other to compete for the finite resources of the explorer. Exploration also involves making decisions, based on the collected information, to test hypotheses and collect more information in an efficient manner.

This thesis introduces a new exploration technique which actively considers how much information can be gained from taking sensor readings as well as the cost of collecting this information. The methodology can consider multiple metrics of information simultaneously — such as finding new terrain and identifying rock type — as it explores and these information metrics can be easily changed to perform new and different exploration tasks. The method considers the costs, such as driving, sensing and planning times, associated with collecting the information. Exploration plans are produced which maximize the utility, information gain minus exploration costs, to the exploring robot.

The multiple information metric exploration planner is used to solve two exploration problems: creating traversability maps and exploring cliffs. These tasks are performed in simulation and the information gain and exploration path lengths are compared as the information metrics are changed. The multiple information metric exploration planner is further demonstrated in a field experiment to explore a cliff, starting at the cliff top the planner found a route to the bottom and collected sensor information from the face of the cliff.

Acknowledgements

I would like to thank my advisor, Dr. William Whittaker for the support and guidance he has given me during my Ph.D. Red has given me fantastic opportunities to work on interesting projects at the ends of the Earth and has opened my mind to a great vision of what is possible. I would also like to thank the rest of my thesis committee, Drs. Reid Simmons, Tony Stentz and Richard Volpe for providing useful comments and guidance. I would especially like to thank Reid for treating me like one of his students and spending many hours talking about autonomy, both for projects and for my thesis research.

Special thanks go to Dr. Nicolas Vandapel. Nicolas has always been available to discuss and guide the direction of my research. He also provided me with premiere technical and logistical support collecting the data found in this thesis. The field data would not have been possible without his tireless efforts on my behalf.

I would also like to thank Alex Foessel who has been a close friend through thick and thin and has always been there to provide sage advice when needed. Another good friend, Daniel Huber, provided the code to read the Z+F laser data, joined me in the field and was always there for lunch with myself and Liang Zhao. Thanks to Paul Tompkins who provided the code to let me generate animations of my exploration paths.

The real (as opposed to simulation) cliff exploration found in Chapter 5 was done at a coal strip mine in Somerset County, PA. My thanks go out to Fieg Brother's Coal, the site owners, for giving me permission to collect data there. Frank Bowman deserves special thanks for spending many of his weekends showing me sites and then making sure everything was alright while I was collecting the data. I must also thank Dr. Martial Hebert who provided me with the laser scanner and Dr. Omead Amidi who provided much of the field equipment.

Over the course of my Ph.D I have made many friends here at CMU who helped my research and made my stay more enjoyable. It is impossible to name them all, but I would

like to thank Bernardine Dias, my office mate over the final years of my thesis; Nick Roy for organizing both the robotics floor hockey team and many cultural excursions; and Mike Wagner who spent countless hours at the slag heaps testing Nomad with me.

Of course I need to thank my family who have steadfastly supported me throughout my long education and have always been there when needed. And finally, I wish to thank Dot Marsh who guided me through life at FRC and was always there to lend a helping hand.

Table of Contents

List of Tables

List of Figures

CHAPTER 1 *Introduction*

Exploration gathers information about the unknown. This information can come in many forms from knowledge of new terrain, to rock geology, to lifeforms. The value of these different information forms to an explorer is determined by a set of information metrics, one for each form of information, that depend on the goal of the exploration task. Exploration also involves making decisions, based on the collected information, to test hypotheses and collect more information in an efficient manner.

As explorations become more complex, increasing numbers of information metrics must be considered in order to succeed. These multiple information metrics must be considered simultaneously during exploration and often conflict with each other to compete for the finite resources of the explorer. The explorer must know the relative importance of these information metrics and weight them appropriately when making decisions about what to discover next.

The ruggedness of terrain also influences the exploration strategy employed by robotic explorers. The cost of driving is not the same for all traversable areas. Some traversable areas are near the limits of mobility, making them difficult to drive over, others may be smooth and easy to drive through. The ruggedness of terrain can also affect the sensor coverage and thus the amount of information gained by taking a sensor reading. Terrain can shadow sensor readings, creating holes in the viewed area. Conversely, high ground can offer better views of large areas. Explorers must consider the effects of terrain on both information gain and exploration cost.

An example of the type of exploration task considered in this thesis is the search for signs of life, both past and present, on Mars. In this scenario, the explorer may have information metrics for the presence of water (both past and present), finding cliff faces (where water runoff has been detected [33]), the geology of rocks (some rock types may be more likely to have fossils), the presence of organic material and drilling conditions (to assess the difficulty of obtaining subsurface samples). Each of these items is an information metric for the explorer.

Another exploration task which might be imagined is the search for water ice in permanently dark regions of the Moon, Mercury and asteroids. In this case the explorer needs to consider lighting information (to identify permanently dark regions where water ice can exist [41]), drilling conditions, age of crater and the location within a crater (near the rim, in the center, etc.) [12]. Except for the drilling conditions information metric, this water ice example has different information metrics than the Martian life scenario. The two examples also have different numbers of information metrics. Thus to be generally applicable to many exploration problems, an exploration strategy must be capable of handling different types and numbers of information metrics.

Each information metric may compete with the others. For example some places may have good drilling conditions, making them favorable to take a core sample but be in a rock type that is unlikely to have fossils or water. The explorer must balance these disparate information metrics and weigh their relative importance to the exploration task. Figure 1-1 illustrates the kinds of decisions an explorer must make when exploring complex outdoor environments. An explorer is continually making these types of trade offs. It might decide to examine terrain that was occluded by a rock or to drive further and examine a crater. The explorer might also weigh the benefits of driving in a gully, where the sensor view is restricted but there is a great chance of finding signs of water or life, versus driving on a ridge top where wide panoramic views are possible.

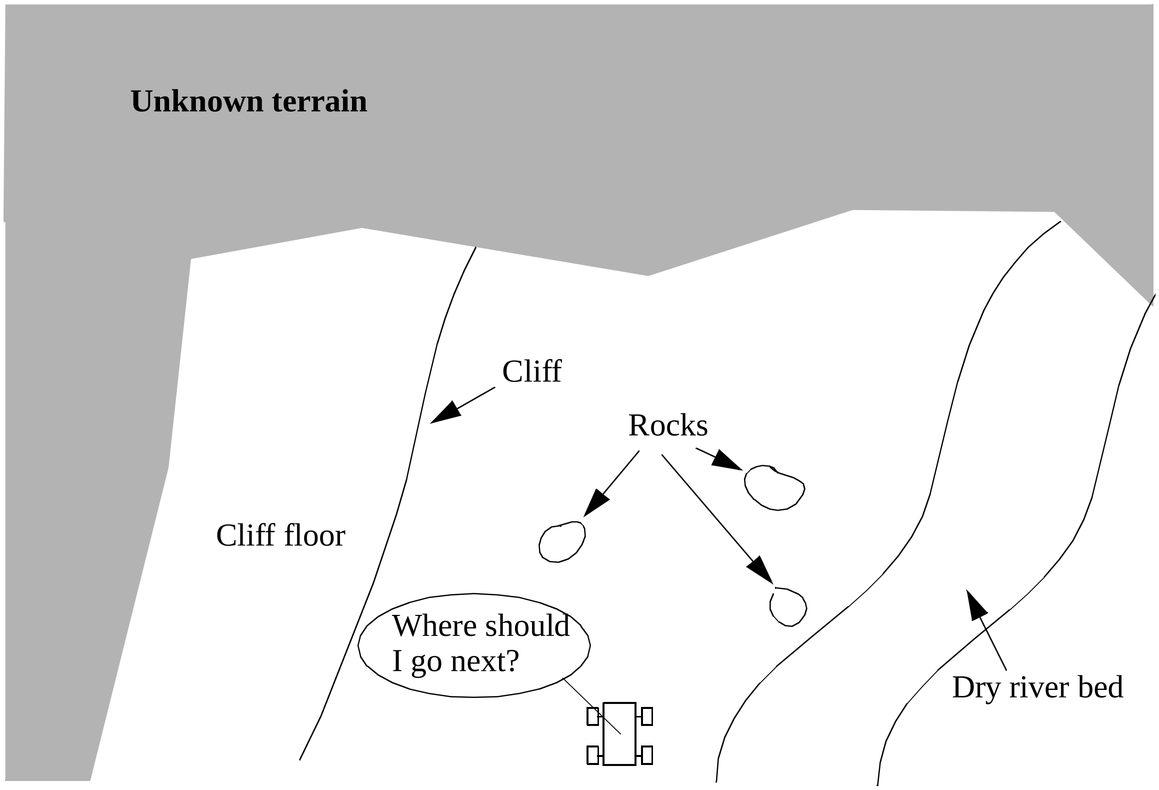


Figure 1-1: Robot’s decisions. Exploring robot must decide where to go next. Should it go look at the rocks more closely? They are close by so the cost is small but how much information will the robot gain? It could drive into the dry river bed and follow it. This has the potential of finding water or signs of life but the steep slopes of the banks make it risky to enter and once inside the banks block the robot’s view of other terrain potentially reducing the amount of information collected. It might start looking for a path to the cliff floor, to gain information from seeing the face of the cliff. Finally it might decide to view some of the unknown terrain at the top of the image. The explorer must make these types of decisions continuously as it explores a region.

While an explorer is considering the information gain from possible actions, it must also consider the cost of collecting this information. These costs might include the time spent driving to the site, sensing and planning. They may also include the power used for drilling. Therefore the explorer must maximize the information gain while minimizing the costs of collecting it. This quantity is referred to as the utility of an action.

Current exploration methods assume that a remote expert, usually a human, will analyze the current data and then specify which regions are important. This model for exploration has a few shortcomings. First, it increases the costs of exploration by requiring a large remote infrastructure to exist. This might limit the number of robots exploring at any one time since a human operator can only control so many robots at one time. Secondly, especially with planetary and undersea explorations, a significant amount of time is required to exchange the data between the explorer and the remote expert due to communication limitations. During this time no exploration is being performed. To overcome these obstacles, this thesis looks at increasing the autonomy of robot explorers by allowing them to make their own decisions on the importance of regions and where to travel.

1.1 Thesis Statement

This thesis asserts that exploration which considers multiple information metrics, and the costs of collecting the information, provides a robust, general and intelligent method of autonomously performing complex exploration tasks. By investigating exploration planning techniques that maximize information gain from multiple metrics while minimizing driving and computation costs, this research will show the benefits of multiple information metric exploration.

1.1.1 Thesis Assumptions

Several assumptions have been made in the development and implementation of the multiple information metrics exploration planning methodology. The exploration strategy has been designed for surface explorations of rugged, outdoor terrain. That is not to say that the methodology presented in this thesis cannot be applied to other domains such as aerial or indoor explorations, but many of the decisions made have been made to improve performance for outdoor surface explorations.

In addition to being a rugged, outdoor terrain, it has been assumed that the environment is static — no moving obstacles — and projectively planar — there are no overhanging structures such as caves or ledges. It is also assumed that the robot is told the extent of the region to explore. For example a robot might be commanded to explore a region 1km by 3km centered at position x. This removes the need to dynamically reallocate the robot’s map.

Finally some assumptions about the robot used with the exploration planner are made. First it is assumed that the robot has good localization — either through DGPS or landmark based position estimation. This means that the robot knows where it is in its map with good precision. It is also assumed that the planner is operated in conjunction with a local obstacle avoidance program. This lets the planner ignore the robot dynamics, such as rate of change of steering angle, and still maintain safety.

1.2 Autonomous Exploration Architectures

Most exploration methodologies fit into one of five exploration architectures: patterned search, human goal selection, active vision, find new terrain and autonomous exploration. These architectures are listed in order of their consideration of the environment; from patterned search, which defines an *a priori* path to travel regardless of the environment, to autonomous exploration which uses environmental information, sometimes with multiple metrics, to determine the exploration path.

Patterned search seeks to view or cover an entire area by using a fixed path which is determined at the start of the search. Some methods require a map of the environment at the start [10][78] to determine a coverage path, while others are able to modify the *a priori* path as obstacles are detected [1][9][23]. Typically, a raster pattern is used, but other patterns such as spirals have been attempted [55]. Essentially, patterned search methods have determined where they will drive before seeing the environment. Thus, these methods are not truly exploration but rather search methods.

In the human goal selection architecture, the exploring robot relays the information it has gathered about the environment to a remote human operator. The human then decides what tasks have the greatest utility for the robot and uplinks the appropriate commands. This type of architecture was successfully used by Sojourner on Mars [36]. One can consider basic path planners [31][61], where the shortest path to a human defined goal is found, to belong in this category of exploration. Planners such as Generalized Robotic Autonomous Mobile Mission Planning System (GRAMMPS) [8], which directed multiple robots to multiple goals are also in this category as are the planner/scheduler architectures Coupled Layer Architecture for Robotic Autonomy (CLARAty) [67][18] and Contingent Rover Language (CRL) [70][6]. These methods have been largely successful because they take the most difficult parts of the exploration problem, deciding what is important and where to go, and placing it in the hands of a knowledgeable human operator. These methods are not restricted to human generated goals. They reformulate plans based on perceived terrain and modify human generated plans as circumstances change. However, these methods focus on the creation, modification and execution of plans rather than on the generation of goals.

The field of active vision approaches the exploration problem from a different perspective. Generally, active vision approaches create a model of the environment and then plan the next sensor view to maximize the expected improvement in the model fit to the data [35][72]. These methods actively consider where the most information to be gained is but because of the need for an environment model they are restricted to very simple environments such as polygons on a table. Also, they usually do not consider the cost associated with collecting the information.

In the finding new terrain architecture the robot decides where to drive based on the single information metric of finding previously unseen terrain. Generally, these techniques divide the world into two classes: seen and unseen and do not consider how well a place has been viewed by the robot. Some methods do not consider how much information will be gained, but simply drive to the closest frontier of seen and unseen [75][76]. Others consider how much unseen terrain will be viewed as well as the costs of collecting the information [58].

The final exploration architecture reviewed here, autonomous exploration, is characterized by robots which autonomously decide where to go, based on a perceived gain, and generally consider more than just finding new terrain as a gain. The most common form of this exploration architecture is for the solution of the indoor mapping problem. These techniques try to find unknown terrain but are also very interested in finding places where they can localize themselves with high accuracy [65][66]. However, these techniques tend to focus on the localization part of the problem rather than the exploration part. Some techniques look at items other than localization such as science goals [17] or integrate the planning with robot perception and motor actions to solve a task [15][16].

1.3 Exploration using Multiple Information Metrics

This thesis introduces a new exploration technique which actively considers how much information can be gained from taking sensor readings as well as the cost of collecting this information. The methodology can consider multiple, metrics of information simultaneously — such as finding new terrain and identifying rock type — as it explores and these information metrics can be easily changed to perform new and different exploration tasks.

For each information metric, the robot computes the expected amount of information to be gained from taking a sensor reading in a particular location. The computed expected information gain is ideally the information theoretic expected information gain and is in the units of bits. By computing a measure for each different information metric that is in the same units, these values can be summed to get a total expected information gain for a location without the need for many heuristically determined relation constants. Each information metric can be weighted to specify its relative importance but ultimately the sum is a summation of items with the same units.

The exploration methodology also considers the costs of collecting the information. Again, to avoid the need for heuristic unit conversions, the same units are used for all the costs. In the examples contained in this thesis, the costs of driving, sensing and planning were considered. Units of time were the most appropriate choice here since sensing and planning are easily expressed in terms of time and driving distance can be converted into time by the robot’s speed. The effect of terrain on driving costs can be taken into account by assuming slower driving speeds over rough terrain. Other units, such as energy, could be used depending on the application.

Next, the exploration methodology must decide where to go. This is done by finding paths which maximize the utility to the exploring robot. Utility is a concept from decision theory which quantifies how useful an action is to the decision maker and allows many different decisions to be meaningfully compared [48]. In the exploration problem, the utility along a possible exploration path is computed as:

$$Utility = ValueOfInformation \cdot ExpectedInformation - Cost \tag{1.1}$$

where the expected information and cost are along the entire path being considered. Since expected information and cost are in two different units (bits and seconds) a unit conversion must be performed to compare them. This is done with the *value of information* metric. Essentially this determines how many seconds the explorer is willing to spend to collect one bit of new information. This conversion factor may change over the course of an exploration indicating that the explorer is more willing to take risks to collect information and spend more time as information becomes scarce.

The multiple information metric exploration planner developed in this thesis is able to handle the complex exploration tasks, such as searching for water ice on the Moon or signs of life on Mars, postulated earlier. Using information theory to assign expected information gains to each exploration criteria or, information metric, allows the planner to weigh the benefits of competing goals and also allows the exploration criteria to be easily changed for different exploration scenarios. It also actively considers the costs of collecting information and ultimately finds exploration paths which maximize the utility to the explorer as defined in equation (1.1).

The methodology actively reasons about the environmental effect on the expected information gains, calculating expected visibility from potential sensor readings. This is essential in the complex, outdoor terrains considered in this thesis, where sensor occlusions are commonplace. Further, the method considers the effect of terrain on the costs involved in collecting information, computing the traversability of the terrain and how that affects driving times.

The major disadvantage with the methodology is the planning complexity. The planner must find paths which maximize utility, however with no destination cell to plan to, the planner must search over all possible paths to all possible destinations. The planning is further complicated by the fact that information gains in the same neighborhood are dependent — every time a sensor reading is taken, the explorer’s map changes and the information gains in the region are also changed. This makes long term planning an expensive option. This problem can be mitigated by only planning to the next sensor reading, which is the approach used in this thesis. However, that can produce myopic plans which may miss some efficiencies found in long term plans.

In this thesis, the multiple information metric exploration planner is used to solve two exploration problems: creating traversability maps and viewing cliff faces. These tasks are performed in simulation and the information gain and exploration path lengths are compared as the information metrics are changed. The multiple information metric exploration planner is further demonstrated in real life to solve the view cliff faces problem on a cliff in a coal strip mine.

1.4 Contributions

This thesis makes several contributions to the field of autonomous exploration. While the focus is on exploring complex outdoor worlds, the results may be applied to many other domains of exploration.

The first contribution is the presentation of a general methodology for performing autonomous exploration. This methodology allows the exploring robot to consider more than one thing while exploring. Further, the criteria of exploration are easily changed in the presented framework. This allows the method to be easily adapted and applied to many diverse exploration problems. Finally, the costs of collecting the information are explicitly accounted for.

The second contribution is the application of information and decision theories to the problem of exploration. By expressing the information metrics in terms of the expected information gain a common unit, the bit, is applied to the very different criteria used in the exploration. By also expressing all of the costs in units of seconds, the expression for robot utility only contains two different units. These units are compared with the *value of information* parameter which has the physical interpretation of how many seconds the robot is willing to spend to gain one bit of information. This use of information theory has significantly reduced the potentially large number of hand tuned constants that might be required.

The final contributions of this work are the two exploration examples presented. These examples not only demonstrate the validity of the approach but they are both important exploration problems. The creation of a traversability map of an unknown area is a classic problem in robotics, particularly in field robotics where traversability is not just a binary — traversable or not — quantity. The introduction of the concept of having greater certainty near poor terrain is an interesting twist which provides maps that are more useful than conventional traversability maps. The second example, finding cliff faces, has great potential in the search for water and life on Mars.

CHAPTER 2

Background

2.1 Exploration Domains

Much work has been done in the area of robot exploration and several methodologies have been developed. In general the work can be broken down into two problem domains: indoor exploration and outdoor exploration. A great deal of research has been done in indoor exploration and much of it has focused on exploration for mapping the interiors of buildings, often with an emphasis on localization while mapping. Much less work has been done in outdoor exploration which is characterized by more complex terrain and exploration tasks than found in indoor exploration.

2.1.1 Indoor Exploration

Much of the work in robot exploration has considered indoor environments. Most of this work used the structure of indoor worlds, such as parallel and perpendicular walls, to help solve the exploration problem [11][28][34][64]. Since this structure is not found in outdoor terrains, these techniques are largely inapplicable.

Another environmental aspect which affects many of the indoor exploration techniques is the binary nature of the world. A region is either traversable (part of the floor) or not traversable (above or below the plane of the floor). Since it requires the same amount of energy to drive the same distance over any traversable area, the cost of driving is only dependent on the distance of the path. In an outdoor situation, traversability is continuously valued and driving costs depend on the distance and the traversability. However, unlike the parallel and perpendicular wall assumptions, indoor algorithms making the binary world assumption are easily converted to more complex terrain by modifying the driving cost to include traversability.

Yamauchi [75] and Yamauchi et al. [76] present a frontier based robot explorer designed to explore complex environments typically found in office buildings. The premise of this method is that “To gain the most information about the world, move to the boundary between open space and uncharted territory” [75]. The robot examines the map it has created so far and classifies each cell as unknown, occupied or empty. It then extracts all the regions, larger than the robot, where empty cells are adjacent to unknown cells. These are called frontiers (Figure 2-1). The robot then plans the shortest path to the nearest frontier, takes another sensor reading, finds new frontiers and repeats the process over again until no new frontiers are discovered. They present their method on an indoor mobile robot. While this method makes few assumptions about the environment it has several shortcomings for outdoor exploration. First, it assumes that detecting and driving to frontiers is the only source of information. For example, it makes no attempt to improve the certainty of detected obstacles. The robot uses a certainty grid for its map but when determining frontiers it reduces this world to unknown, occupied and empty. So as soon as a cell has a probability of being occupied that is greater than the initial probability, it is occupied as far as the exploration goes. It is possible that this was a bad sensor reading or that one would like to have a higher certainty of obstacles. Finally, the algorithm always chooses the closest frontier for exploration. It may in fact be better to go to a slightly farther frontier if it is larger or less bumpy than a closer one.

Simmons et al. [58] look at using multiple robots to explore using a similar frontier strategy. However, each frontier is evaluated based on the expected number of unknown cells the robot can see from the frontier as well as the distance from the robot. Thus the exploring robots choose the frontier which will provide the highest utility (information gain minus driving cost) rather than simply the closest frontier. The multiple robots also coordinate their efforts to reduce overlap while exploring.

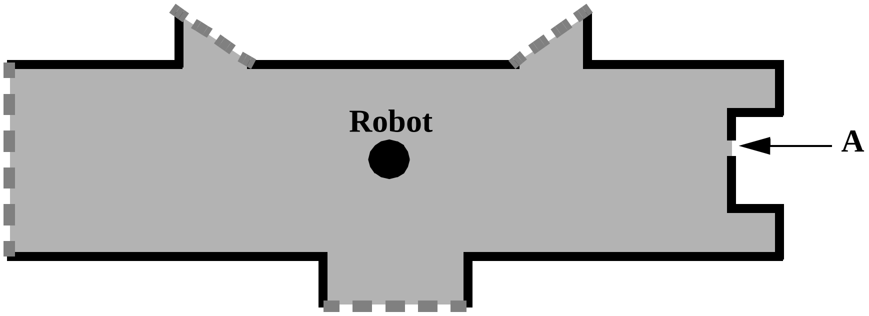


Figure 2-1: Frontiers. In this scene of a hallway, the black lines are walls or occupied, grey is empty and white is unknown. The dashed grey lines are frontiers - areas between empty and unknown regions. Opening A, which is also an area between empty and unknown regions, is not a frontier, as defined by Yamauchi, because it is too narrow for the robot to pass through.

The previous papers all looked at exploring as simply finding new terrain. However the exploration tasks considered in this thesis require more from the robot than just finding new terrain. The following are some methods developed for indoor robots which consider more than one criteria in their explorations.

Roy et al. [52] look at including the ability to localize the robot, using natural landmarks, when planning paths. This produces paths which remain close to walls. While [52] was not trying to explore new regions it is considering two criteria, shortest distance and localization ability, when planning paths. A similar problem is the simultaneous localization and mapping problem where the robot is trying to map an unknown area and at the same time maintain good localization. Thus, finding new terrain and finding good localization places can be considered the two criteria of these explorations. Many researchers have looked at this problem, [65][66] are two good methods, however more attention is usually given to the problem of localization than to deciding which paths to take in the exploration.

In [15][16], Elfes presents an integrated approach to robot navigation which incorporates task specific information needs, perception sensor capabilities and robot knowledge into the motion planning process. An inference grid [15] is used to represent the robot’s knowledge and task needs. The integrated architecture allows the robot to plan both motor and perceptual actions to solve a given task.

2.1.2 Outdoor Exploration

The capabilities of outdoor robots have increased dramatically over the last five years with robots being used in agriculture, mining and excavation [73]. Exploration robots have gone to extreme environments such as Mars [36] and Antarctica [37]. A large part of the research into outdoor and planetary robots has focused on terrain sensing and path planning for driving in rugged, outdoor terrains [8][13][22][31][37][59][61][68]. Despite the progress, little work has been done in creating truly exploring robots which reason about the environment as it is discovered to determine interesting areas.

Much of the outdoor exploration work has centered on space robotics and the exploration of other planets, particularly Mars [24][68]. Most of this work has focused on getting from point A to point B safely and, as such, much of this research has involved path planners such as RoverBug [32] and D* [61], local navigation such as Morphin [57] and determining traversability of the terrain [37][53]. Typically though, point B is chosen by a human operator. Recently, more work has been done in planning sequences of science goals and actions to maximize the utility of the rover and replan as new information is gathered [6][18]. However, it is still a human scientist who is determining the basic plan and more importantly what is interesting and what is not.

Estlin et al. [17] present a planning system for science exploration with multiple robots. The system clusters spectrometer readings to determine rock types. Goal points are set at the two mutually most distant points in physical space for each rock type seen so far. The planner is thus biased to explore towards the edges of its known world. The approach of Estlin et al. differs from that of this thesis in that they only consider one type of information, rock types, in their goal selection. Further, they do not explicitly consider how much information will be gained when selecting goal points, choosing instead to use a heuristic based on relative rock positions.

Pedersen [44][45] used a Bayesian classifier to autonomously classify rock and meteorite types from camera and spectrometer data collected by the robot Nomad. While Pedersen does not consider path planning his classifier produces valuable information that could be used to specify science goals in a planner such as CLARAty [18] or CRL [70]. Further, since the result is produced as a probability, this information could be very easily incorporated into the multiple information metric exploration planner presented in this thesis.

2.2 Exploration Methodologies

While the previous section looked at exploration work done in the domains of indoor and outdoor robotics, three methodology classes, which have received a great deal of attention, deserve to be examined in detail. The first, patterned search, is concerned with seeing an entire region and uses predetermined paths to complete a task. The second, active vision, tries to determine the next best place to take a sensor reading to improve its model of the environment. The third, behavior based techniques considers robot actions based on primitive behaviors.

2.2.1 Patterned Search

The use of coverage patterns provides a technique to allow the robot to search an area. These techniques assume that the robot has a constant width of interest, either from a sensor or implement such as a broom. The coverage pattern algorithm drives the robot in such a way that the entire environment is seen or covered by this width of interest. Typically the coverage patterns assume that all parts of the environment are equally interesting and as such it is common to use an *a priori* defined pattern which may be modified when obstacles are detected. However, Gage [20] examines the use of random paths to get coverage.

Many of the patterned search methods require that the robot starts with a map of the area to be covered. For example, Zelinsky et al. [78] present coverage planners based on the distance transform and path transform path planners. To get complete coverage the robot takes the longest path to the goal instead of the shortest. The distance transform based coverage pattern spirals into the goal point and so causes the robot to turn excessively, adding large errors to dead reckoning positioning accuracy. Therefore, the authors prefer the coverage planner based on the path transform which produces coverage paths that follow the contours of obstacles and have fewer sharp turns. In Choset and Pignon [10], obstacles inside the environment must be known *a priori*.

Cao et al. [9], developed a coverage pattern for use in a lawn mowing application. Their method takes a map of the region to be covered and decomposes this into regular regions. A regular region is one where a horizontal line will only intersect the region boundary in two places. The planner then covers each regular region using a back and forth pattern with horizontal rows. To handle unknown obstacles in the environment, the robot splits the current regular region into two, one on the current side of the obstacle and one behind the obstacle. It then covers each regular region separately. Only convex obstacles are considered. In Acar and Choset [1] a similar technique is introduced which can handle non-convex obstacles.

One of the most general coverage planners is presented in Hert et al. [23]. This coverage planner was designed for an autonomous underwater vehicle mosaiking images of the ocean floor. It requires no *a priori* information of the environment, except for a description of the boundary region. It is general enough to handle any terrain that is projectively planar (so no caves or overhangs). The paper also proves that the generated path lengths are linear in the size of the area to cover.

In general these coverage planners are concerned with search or coverage and are not well suited to outdoor exploration. Many of them were developed for lawn mowing or floor sweeping type applications and assume that a map of the environment is known at the start. Even [23], which requires no knowledge of the environment, is limited in three ways for outdoor surface exploration. First, it assumes a binary world - traversable and not traversable and so does not take terrain factors into account. This means that they assume obstacles will not interfere with sensing and that all paths are equally easy to drive along. Second, they consider discovering unknown terrain as the only gain of information and that all unknown terrain is equally interesting to discover, Finally, as illustrated in Figure 2-2, this coverage planner may cause excessive turning when the *a priori* pattern is incompatible with the environment. In Figure 2-2(a) a typical raster pattern coverage path is shown. The orientation of the raster is determined before any information about the environment is known and cannot be changed during the exploration. When the two black objects in Figure 2-2(a) are discovered, the raster pattern is shortened to cover the area between the rocks. However, in this case the raster pattern, which was chosen before the two black objects were known, is not the most efficient way to cover the area between the objects. It forces the robot to turn excessively and makes very short straight row lengths. A more intelligent way to cover the area between the two objects is shown in Figur e2-2(b). Unfortunately, pattern search techniques are intimately tied to the *a priori* defined pattern.

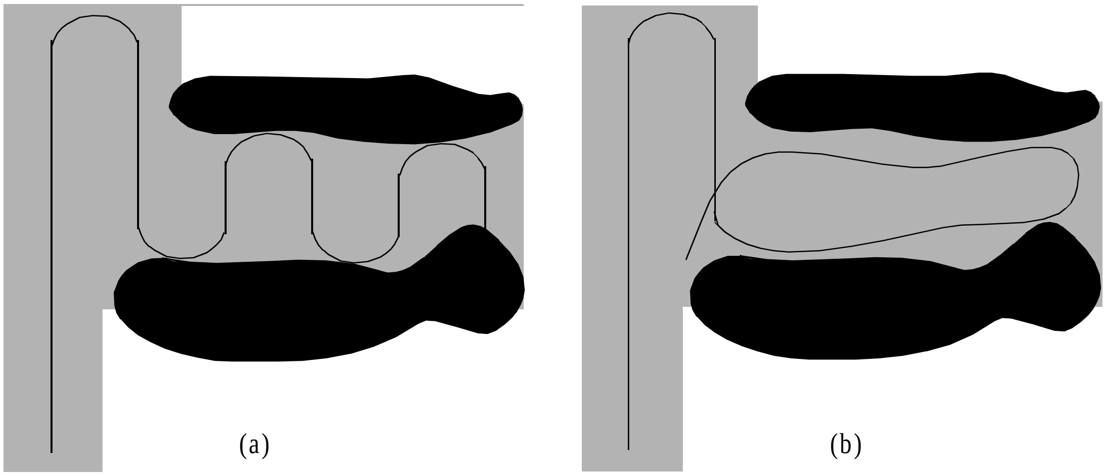


Figure 2-2: Problems with patterned search. (a) Fixed pattern is incompatible with environment features causing excessive turning and short rows. (b) Covering the same area with fewer turns.

Shillcutt et al. [55] are concerned with energy generation and consumption for solar or wind powered robots performing coverage patterns. The paper looks at different robot configurations and coverage patterns and evaluates how much energy was produced, consumed and what percentage of the area was covered. This methodology is different from most coverage or search pattern methods in that it does not just consider coverage as a requirement for success but also looks at power consumption and generation. However, the methodology uses fixed *a priori* patterns, evaluates energy produced and consumed based on prior knowledge of the environment, not information collected by the robot, and assumes sparse obstacles in the environment making it inappropriate for the outdoor exploration tasks considered here.

2.2.2 Active Vision

The field of active vision is concerned with actively controlling a perception sensor's parameters, such as position and zoom, to accomplish a specific task and in response to the environment. Several common applications of active vision are attention, foveal sensing, gaze control and hand-eye coordination [63].

One branch of active vision, planning the next view, has many similarities to the outdoor exploration problem. Generally the planning the next view problem looks at where to take the next sensor reading to provide the most information for a specific task. One example of this is filling in the holes caused by object occlusions when mapping a scene [35]. Another looks at fitting a superellipsoid model to a scene mapped with noisy sensors and planning the next sensor reading to maximize the decrease in uncertainty [72]. Both of these techniques have similarities to the exploration problems considered in this thesis. However, they are concerned with much smaller and simpler scenes, collections of polyhedral objects on a table, and both methods must model the environment to be able to plan the next view. In a similar manner, Okamura and Cutkosky [42] explore an object, locating specific features such as bumps and steps, with a robotic finger. The methodology guides the exploration to verify which feature model best describes the discovered feature. These techniques all consider quite simple scenes and guide the exploration to improve the fit of a model to that scene. It is unlikely that these methods will scale well to the complex outdoor environments considered in this thesis.

Another concept to consider is the art gallery problem [43] where the positions to be visited so that an entire 2D scene can be viewed are calculated. A recent implementation of this type of algorithm is [21] where the need for overlap to ease the registration of the multiple scenes is considered. However, these problems require a 2D map of the region of interest to work and are not applicable to exploration of an unknown environment.

2.2.3 Behavior Based Techniques

In behavior based robotics systems, a collection of primitive capabilities — the behaviors — are used to decide what action the robot will take at any given time. Since these behaviors are simple, they can process sensor data quickly, letting the robot operate in a real, dynamic environment at high speeds. Also, because the behaviors are generally independent, self contained processes, the suite of behaviors used can be easily modified. The disadvantage of behavior based techniques is that the often competing desires of these behaviors must be combined or selected into just one action which the robot will take. How best to do this is not an easy choice and several prominent methods are discussed below.

In the subsumption architecture [7] a priority-based arbitration technique is used. Each behavior operates asynchronously operating on sensor data as it arrives and producing a vote for how the robot should act when the behavior becomes active. Each behavior is assigned a priority and the active behavior with the highest priority controls the robot, the other behaviors are ignored. This arbitration scheme provides is very fast but does not allow multiple behaviors to affect the robot’s actions simultaneously.

Another method for combining multiple behaviors is to use command fusion. Instead of the all-or-nothing strategy of subsumption, command fusion techniques combine the requested actions of all behaviors. In motor schemas [2], each behavior generates a vector indicating which direction to travel and these vectors are summed to decide which direction the robot should travel. Another approach is the DAMN architecture where each behavior votes for or against each of a set of actions [50]. Fuzzy logic has also been used to perform command fusion from multiple behaviors [53]. A major drawback of command fusion is that unless each behavior is given substantial knowledge of the current robot state, kinematics and dynamics, commands which are not physically realizable may be generated.

The final approach considered here is utility fusion [51]. In the method proposed by Rosenblatt, each behavior decides on the utility, or usefulness to the behavior, of the robot being in a certain state. The behaviors generate a small local map with utilities which is sent to the arbiter. The arbiter can then combine the utilities in a local map and evaluate actions based on the kinematics and dynamics of the robot. Thus state, kinematic and dynamic information needs to only reside in the arbiter and not the individual behaviors. Further, utility theory provides a method of dealing with uncertainty.

This thesis uses a similar method to Rosenblatt's utility fusion. The combination of information metrics is similar to utility fusion except that expected information gain is being combined rather than utility. Further, a global map is maintained by the exploration planner, allowing the information metrics to make decisions on information gain based on the entire environment.

2.3 Information Theory

The theory of information was first presented by Claude Shannon in the late 1940's as a way to mathematically characterize communication systems [54]. One of the principle problems considered by Shannon was how much information was present in a transmitted message and how to optimally encode the message so that it could be reproduced at the receiver. Since then information theory has been applied to many diverse areas [49].

This section introduces the concept of entropy which is important in the calculation of information gain. Entropy will be used later in this thesis to compute the expected information gains for each information metric used by the explorer.

Imagine a set of mutually exclusive events, $\{E_0 \ldots E_n\}$. The probability of event E_i occurring is p_i and the p_i's sum to one. Together the E's and p's form a sample space for this example problem which we'll call Ω. Now, we define a measure of uncertainty called the *entropy* of this sample space as[1][49]:

[1] Unless otherwise noted all logarithms used in this thesis are base 2

$$H(X) = -\sum_{i=0}^{n} p_i \log(p_i) \tag{2.1}$$

where *X* is a random variable drawn from the sample space, Ω. The entropy of the sample space quantifies how surprised we should be at the event assumed by *X*. The entropy, *H(X)*, is a value between 0 and 1 and represents the average amount of information contained in an event.

For example, consider the following two binary sample spaces:

A. $\{E_0, E_1\}$	$P = \{1, 0\}$	$H(X) = 0$
B. $\{E_0, E_1\}$	$P = \{0.5, 0.5\}$	$H(X) = 1$

In sample space A, there is no surprise as to which event will occur. *X* will be the event E_0 with probability of 1. Thus on average there is no information gained by observing *X* and the entropy of sample space A is 0. However in sample space B, both events have the same probability of occurring. Therefore, on average, we gain a lot of information by observing *X* when it is drawn from sample space B and the entropy is one which is the maximum possible value for entropy.

Equation (2.1) defines the entropy of a discrete valued variable. The concept of entropy can be extended to variables with continuous distributions. If we let *X* be a random variable drawn from a distribution with a probability density function *f(x)*, then the entropy is defined as [49]:

$$H(X) = -\int_{-\infty}^{\infty} f(x)\log(f(x))dx \tag{2.2}$$

When applied to a continuous variable, two problems may occur with the entropy. First, the entropy may be negative. Probability only requires that

$$\int_{-\infty}^{\infty} f(x)dx = 1 \tag{2.3}$$

and this restriction does not require *f(x)* to be less than 1 for all values of *x*. If *f(x)* does exceed 1 for some values of *x* a negative entropy may result. In this case the concept of average self information cannot be associated with the continuous entropy [49]. Secondly, again based on the *f(x)* used, the entropy could become infinitely large. When using the concept of entropy for continuous variables care must be taken in choosing *f(x)*.

CHAPTER 3

Methodology

This chapter presents a general methodology called the *Multiple Information Metrics Exploration Planner,* which can be used to solve complex exploration tasks. The method stores its knowledge in a map and, for all areas of the map, computes the expected information to be gained, for multiple information metrics, from taking a sensor reading there. Finally, it plans paths which maximize the utility of the robot, information gained minus the cost of collecting it. By carefully choosing the appropriate information metrics and the knowledge recorded in the map, the multiple information metrics exploration planner can solve many complex exploration tasks.

3.1 Exploration Framework

This research models the explorer's knowledge of the world or its map as a uniform grid, ***M(t)***, which takes the robot's pose, ***X***, and maps it to a unique cell. The map, ***M(t)***, is the robot's current knowledge of the world and is a function of time since with each sensor reading the explorer learns a little bit more about the world and its map changes to reflect this. Each cell, *m*, in the map contains two vectors: $\boldsymbol{A^m}$ a vector of cell properties or attributes and $\boldsymbol{G^m}$ a vector of expected information gains (see Figure 3-1). The size and composition of $\boldsymbol{A^m}$ and $\boldsymbol{G^m}$ are the same for all cells, *m*, in the map and depend on the exploration task, the number of information metrics and their type. However, $\boldsymbol{A^m}$ does not need to be the same size as $\boldsymbol{G^m}$. Some typical elements of $\boldsymbol{A^m}$ and $\boldsymbol{G^m}$ are detailed at the end of this chapter and two specific implementations are found in Chapter 4.

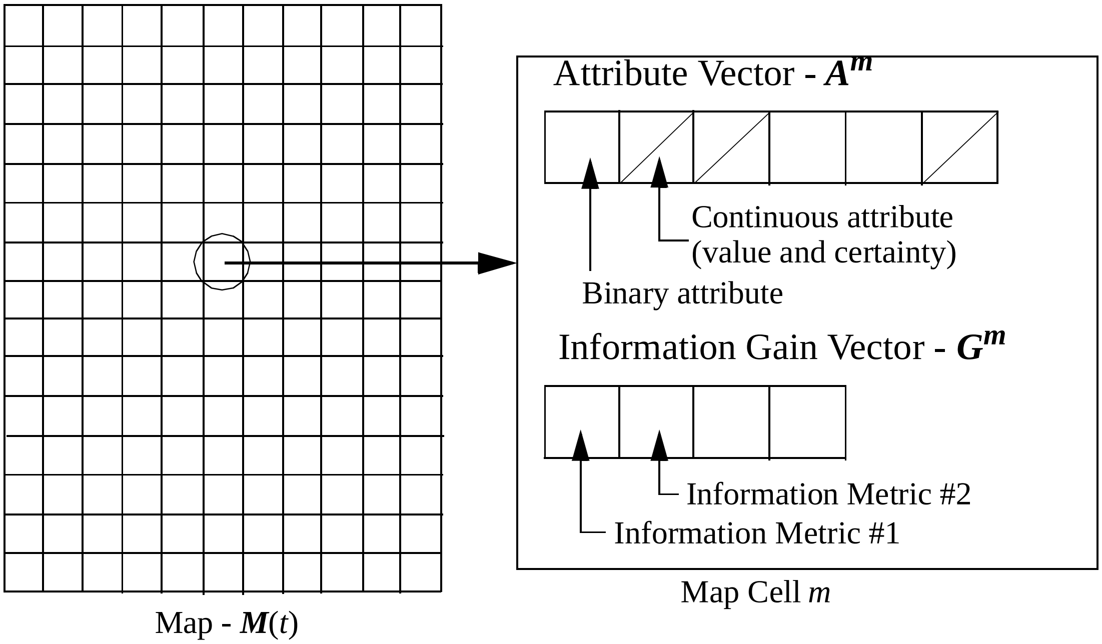

Attribute Vector - $\boldsymbol{A^m}$
Continuous attribute (value and certainty)
Binary attribute
Information Gain Vector - $\boldsymbol{G^m}$
Information Metric #2
Information Metric #1
Map Cell m
Map - $\boldsymbol{M}(t)$

Figure 3-1: Map Cell Composition. Shows the composition of one cell in the map, cell *m*, indicating the attribute vector with binary and continuous elements and the information gain vector. The diagonal line in the attribute vector indicates that an element is continuous and has two numbers, value and certainty, associated with it.

The cell attribute vector, $\boldsymbol{A^m}$, contains the information the explorer knows about that cell. Some typical types of information stored in the elements of $\boldsymbol{A^m}$ are cell height and traversability. Each element of $\boldsymbol{A^m}$ represents a different type of information. For binary valued variables this number is the probability of the state being true. This is the approach used in inference grids, which is an extension of the more common occupancy grids [15]. For non-binary valued variables, such as cell height, two elements of $\boldsymbol{A^m}$ are used. The first indicates the property value and the second is the explorer's certainty or confidence in that property value. Typically the certainty is based on the number of sensor readings received for that cell. This is similar to the goodness maps successfully employed in outdoor navigation [37]. A more rigorous method would be to compute and store the probability distribution of these non-binary variables, however, this would require significantly more computation and storage space than the value/certainty system used in this thesis.

The expected information gain vector, $\boldsymbol{G^m}$, has one element for each metric of information being considered by the explorer. Each element in $\boldsymbol{G^m}$ represents the expected amount of information to be gained, over the entire sensor footprint area, by taking a sensor reading in cell, *m*, for a particular information metric (see Figure 3-2). It is the expected information gain since it is predicting how much new information will be received by taking a sensor reading in this cell. Each element in $\boldsymbol{G^m}$ has the same units, that of information gain, which from Information Theory is the unit bits [49].

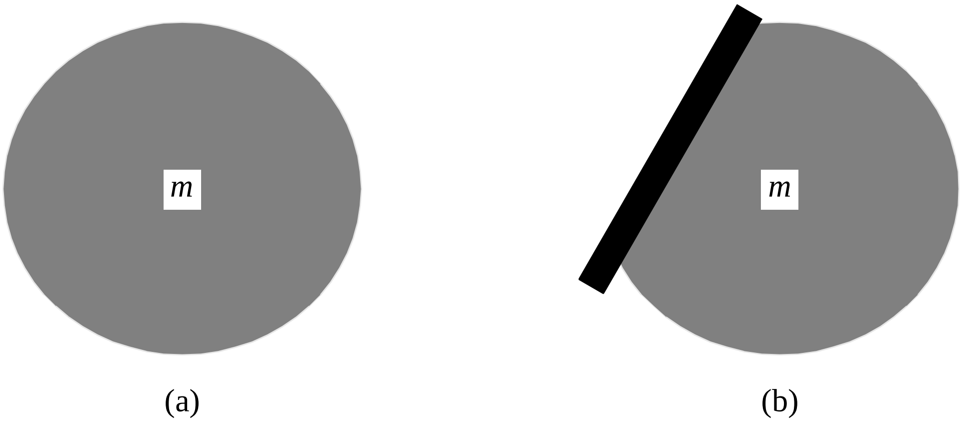


Figure 3-2: Expected Information Gain. (a) The expected information gain in cell *m* is the expected information to be gained over the entire sensor footprint (shown as grey circle). (b) Here a wall obstructs part of the sensor view so the information gain collection region is smaller (again shown as a grey circle).

The total expected information to be gained by taking a sensor reading in map cell *m*, is the sum of the expected information gained from each information metric. The total expected information is computed by performing a weighted sum of the elements in cell *m*’s $\boldsymbol{G^m}$ vector:

$$E[I(m, M)] = \sum_i \alpha_i g_i^m \tag{3.1}$$

where the α multipliers are chosen to weight the relative importance of the various information metrics.

When a sensor reading is taken, the map, $\boldsymbol{M}(t)$, changes reflecting the new information gathered. From (3.1) $\boldsymbol{G^m}$, and the expected information, $E[I]$ computed from $\boldsymbol{G^m}$, are dependent on $\boldsymbol{M}(t)$. Therefore, the $\boldsymbol{G}$ vectors and $E[I]$'s in nearby cells change with each sensor reading and are not independent. In this case, nearby is defined by the sensor footprint. If the sensor footprint at cell m overlaps with the sensor footprint in cell n then $\boldsymbol{G^m}$ and $\boldsymbol{G^n}$ are dependent. If there is no overlap, the vectors are independent (see Figure3-3).

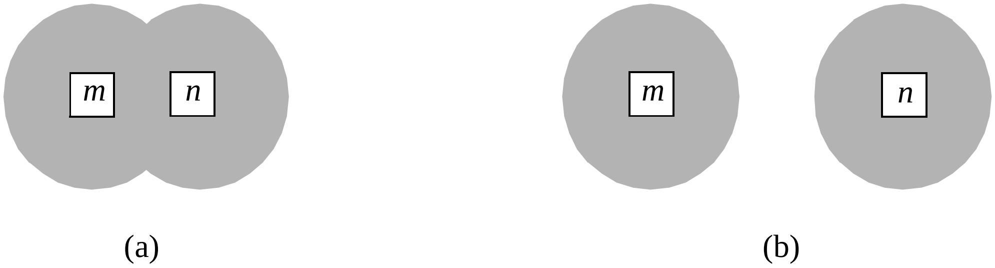


Figure 3-3: Region of dependency for $\boldsymbol{G}$ vectors. (a) Sensor footprints of cells m and n overlap - $\boldsymbol{G^m}$ and $\boldsymbol{G^n}$ are dependent. (b) Sensor footprints do not overlap - $\boldsymbol{G^m}$ and $\boldsymbol{G^n}$ vectors are independent.

3.2 Planning Exploration Paths

An autonomous explorer must decide where to drive and where to take sensor readings to maximize the amount of information it collects. At the same time it must minimize the costs of collecting this information such as driving time, sensing time and planning time. The goal of the exploration planner is to find a path which maximizes the utility to the explorer. The path is an ordered set of cells that the explorer must drive through or take sensor readings in. The utility of a path, p, is defined as:

$$U(p) = k\sum_{m \in p} S(m)E[I(m, M)] - \sum_{m \in p} C_C(m) - C_G \tag{3.2}$$

$E[I(m, M)]$ is the expected information gain of a cell as computed in equation (3.1) and has units bits. It is a function of cell, m and the current map, $\boldsymbol{M}$. $S(m)$ is one if a sensing action is to be performed in cell m and zero otherwise — indicating that no information is gained unless a sensor reading is taken. $C_C(m)$ is the per cell cost which is the amount of time, in seconds, spent in cell m. In this thesis $C_C(m)$ includes the driving time and sensing time. C_G includes any global costs, such as planning time and is also in units of seconds. Finally, k is the *value of information* and is used to set the relative importance of information and cost. It has units of seconds per bit and represents how much time we are willing to spend to get that next bit of information. The value of information, k, does not need to remain fixed throughout the exploration. It might be low at the start of the mission when new information is easy to obtain. As the mission progresses the robot knows more about its environment and new information is harder to find. At this point, k can be increased allowing the robot to spend more time, and take greater risks, to collect new information.

In traditional path planning problems the robot knows where it is and where it wants to go. Knowing the start and end points limits the number of possible paths and reduces the search space. Many techniques exist to solve this problem and some, such as [61] and [31], have been tested in outdoor conditions and shown to work well. Further, it is possible to find the optimal path given the robot’s knowledge of the world and map resolution.

Unlike traditional path planning problems, the exploration path problem does not have a destination cell to plan a path to. Instead, the exploration planner must maximize $U(p)$ over all possible paths to all possible destinations. Thus the number of possible paths to search through is much greater than in the traditional path planning problem. In fact, the exploration path problem is similar to the prize collecting travelling salesman problem (PCTS) [4]. In the PCTS problem a salesman must visit a set of cities. In each city he receives a prize but he incurs a cost for travel between cities. The goal is to find the optimal route which maximizes the prizes but minimizes the travel cost. If each map cell in the exploration path problem is equated with a city in the PCTS problem, the expected information to be gained, $E[I(m, M)]$, is the prize and the driving, sensing and planning costs are the travel cost then the similarity between the exploration path problem and the PCTS problem is apparent. Since the PCTS problem is NP-complete [4] finding an optimal path for the exploration path problem is intractable.

In the PCTS problem, the prizes in each city are independent, collecting one prize does not alter the value of prizes in other cities. In the exploration problem the prizes are the $E[I]$'s which depend on the current map, $\boldsymbol{M}(t)$. To collect a prize the explorer must take a sensor reading. This changes the map, $\boldsymbol{M}(t)$, which changes the $E[I]$'s in the neighborhood of the sensing action. Therefore the $E[I]$'s in equation (3.2) depend not only on the cell location but also on the path used to get there. Thus, if the planner gets to cell *m* from path A and also from path B, it cannot treat the two as one for the remainder of the path plan.

Figure 3-4 illustrates the problems with path dependency. In this example the robot receives information by taking sensor readings on the boundary of known and unknown regions much like the frontier exploration strategies discussed in [58] and [75]. Figure 3-4(a) shows the robot near a doorway. It expects to receive high information gain by taking sensor readings in the doorway because this is the boundary of its known region. While the map grid is not shown on the figures, the doorway is wide enough to encompass several map cells, all of which have high information gain.

Figure 3-4(b) shows a path where the robot visits and takes a sensor reading in each doorway cell and then proceeds into the next room. This would be the optimal path if the information gains in the doorway were independent — the robot could collect a lot of information for very little travel cost.

However, after every sensor reading, the map changes. Figur e3-4(c) shows the map after the robot takes its first sensor reading in the doorway. Now, the boundary between known and unknown is inside the next room and the remaining cells in the doorway no longer have any information gain. Therefore, if the robot travels to the left side of the doorway, takes a sensor reading then travels to the right side of the doorway and takes a sensor reading it will not gain any information from reading on the right side. However, if the robot went to the right side first and then the left, it would collect information from the right cell and not the left. This demonstrates the dependency of the information gains on the previous path. This dependency on path greatly increases the search space required to find *p*.

Finally, Figure 3-4(d) shows that the correct path to take is to visit one cell in the doorway, take a sensor reading and then proceed into the next room.

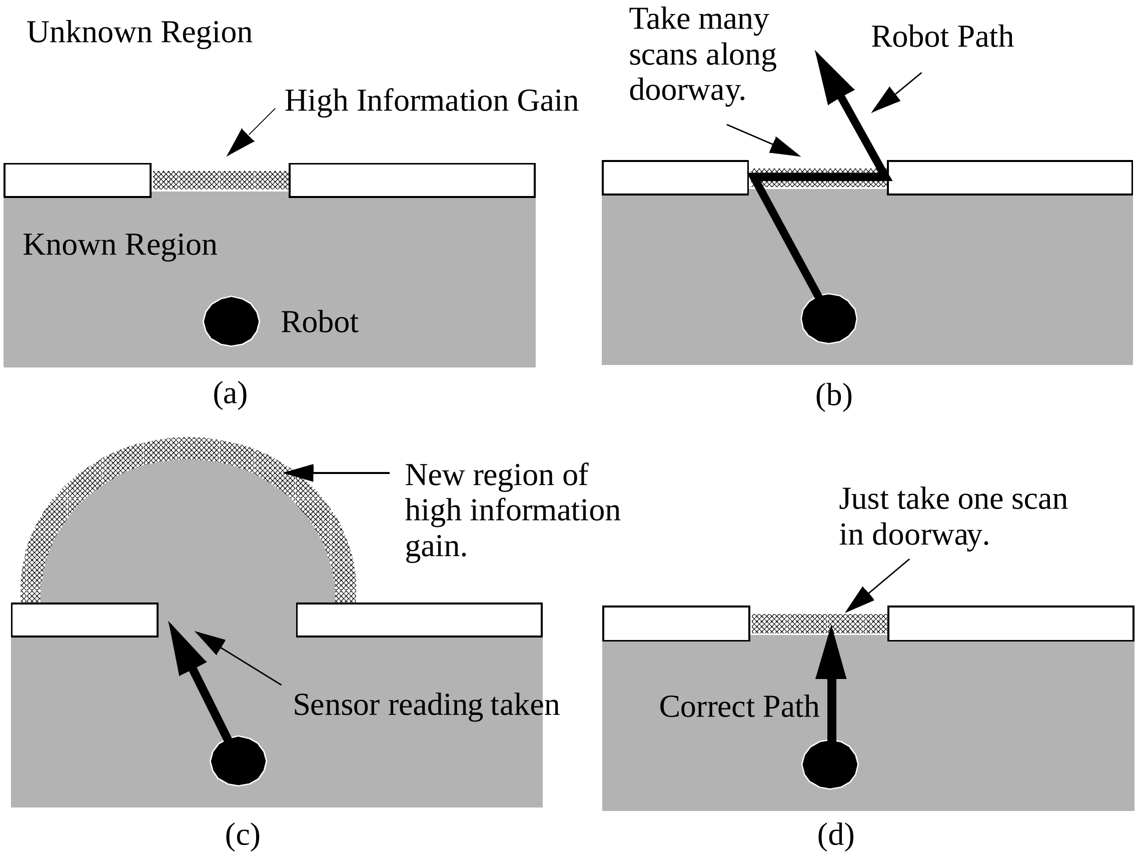


Figure 3-4: Path dependency of information gains.

3.2.1 Greedy Search Planner

The lack of a destination cell and the dependency of information gains makes planning exploration paths difficult. Once the planner calls for a sensor reading to be taken, it must estimate what the map will look like after the sensor reading and use this estimated map to plan the remainder of the path. In a simple and structured environment this may be possible, but in the complex outdoor terrain considered in this thesis it is unlikely that an accurate or meaningful estimate of a future map can be created.

A solution to this problem is to only plan to the first sensor reading. In essence the planner is asking the question “Where is the best place to take the next sensor reading so that utility is maximized?”. This is the greedy search algorithm and is used frequently in robot mapping tasks. In the implementation used in this thesis, the planner first propagates the driving costs through the map using a wave-front propagation technique [30]. Then it chooses the cell with the maximum utility as computed by equation (3.2). A similar planning method was successfully used for indoor exploration in [58]. The costs of sensing and planning can be ignored in the planning process since they will be the same for all paths. This greedy planner generates paths which are a series of cells to drive through and end with a sensor reading.

While the greedy search algorithm will not produce globally optimal paths, it will be shown in following chapters that this planner does produce reasonable exploration paths. Further, one of the requirements set out in the design of this exploration planner was to gain the greatest amount of utility in the shortest amount of time. Thus it is not obvious that taking a less greedy path in order to improve the global optimality of the path (which in essence is what more complex planners would do) is the right thing to do. Finally, Koenig et al. [27] show that greedy search, as used in the map making problem, is in fact not so bad after all.

3.2.2 Random Walk Planner

Another planning strategy investigated was a random walk planner. For this planner the expected information gains were normalized so that the total expected information gain in a cell was always between 0 and 1. The planner then generated a random number between 0 and 1. If the random number was less than or equal to the total expected information gain then the robot took a sensor reading. If not, the robot moved randomly to an adjacent cell (each cell had equal probability except the previous cell occupied by the robot — this cell had zero probability). This planner performed poorly compared to the greedy planner so results are not included in this document. If interested, results comparing the random walk planner to the greedy planner for the multiple information metrics exploration problem can be found in [38].

3.3 Map Attributes

The methodology presented above is very general and can be applied to many problems by creating and using the appropriate map attributes, $\boldsymbol{A^m}$, and information gains, $\boldsymbol{G^m}$, in the explorer's map. The composition of $\boldsymbol{A^m}$ and $\boldsymbol{G^m}$ must be designed based on the exploration task's goals and requirements. This section presents the four map attributes used in this thesis: height, traversability, reachability and cliff, to demonstrate the form of map attributes.

3.3.1 Height

Height (a_h, a_{ch}) is a continuous valued variable so it has a value and a certainty in the attribute vector. The height value (a_h) is the maximum height that the robot has perceived in the cell. It is relative to some global, fixed reference point. The height certainty (a_{ch}) is a number from 0 to 1 which is proportional to the number of sensor readings received in a cell.

3.3.2 Traversability

Traversability (a_t, a_{ct}) is also a continuous valued variable which represents how easy or safe it is for the robot to occupy the cell. The traversability value (a_t) of a cell is computed by fitting a plane, centered at the cell in question, to the cell height data in a region equal to the size of the robot. The traversability is determined by the roll and pitch of the plane as well as the residual from fitting the plane [37]. Using a plane the size of the robot produces a traversability score that is in configuration space. If the origin of the robot is in a cell with good traversability, this means that all parts of the robot are in good traversability. Alternately, if the cell has poor traversability some part of the robot is on dangerous terrain, perhaps a wheel would be in a deep hole. This use of configuration space traversability means that the planner can consider the robot to be a point robot [30]. The certainty in the traversability (a_{ct}) is related to the certainty in the height data used to calculate traversability. A traversability computed over a region of low height certainty would be less certain than one computed over a region with high height certainty. To maximize robot safety, the worst case scenario was chosen and the traversability certainty is set to the minimum, or worst, height certainty over the region fitted with the plane.

3.3.3 Reachability

The reachability of a cell is a binary valued quantity — the cell is reachable or it is not. Therefore, the reachability element in $\boldsymbol{A}$, a_r, denotes the probability that a cell is reachable. A cell is reachable if the robot can drive to that cell from any other cell in the set of reachable cells. By definition a reachable cell is traversable, however, a traversable cell may not be reachable if no traversable path from the start location to it exists. The set of reachable cells is connected in that every reachable cell has at least one adjacent cell which is also reachable.

The reachability of a cell, a_r, is computed by assigning $a_r = 1$ for any cell previously visited, $a_r = 0.5$ for unknown cells and $a_r = 0$ for untraversable cells. The remaining cells are set using a decaying exponential based on the cost of driving to that cell from a cell where $a_r = 1$, the higher the driving cost the lower the value of a_r.

3.3.4 Probability of Cliff

The final map attribute considered in this thesis is the cliff attribute (a_c). Like reachability this is a binary attribute so a_c denotes the probability that a cell is part of a cliff. The cliff attribute is computed using the plane fit for the traversability attribute. If there is a large discontinuity in the heights (a_h) in this plane region then the cell is considered as a potential cliff. The value of a_c is set proportionally to the slope of the plane - the larger the slope the greater the probability of the cell being a cliff. As will be seen in the following chapter, the cliff attribute will be used in an exploration problem that attempts to view the face of the cliff. Thus a_c is only non-zero for the cells which are below the cliff top.

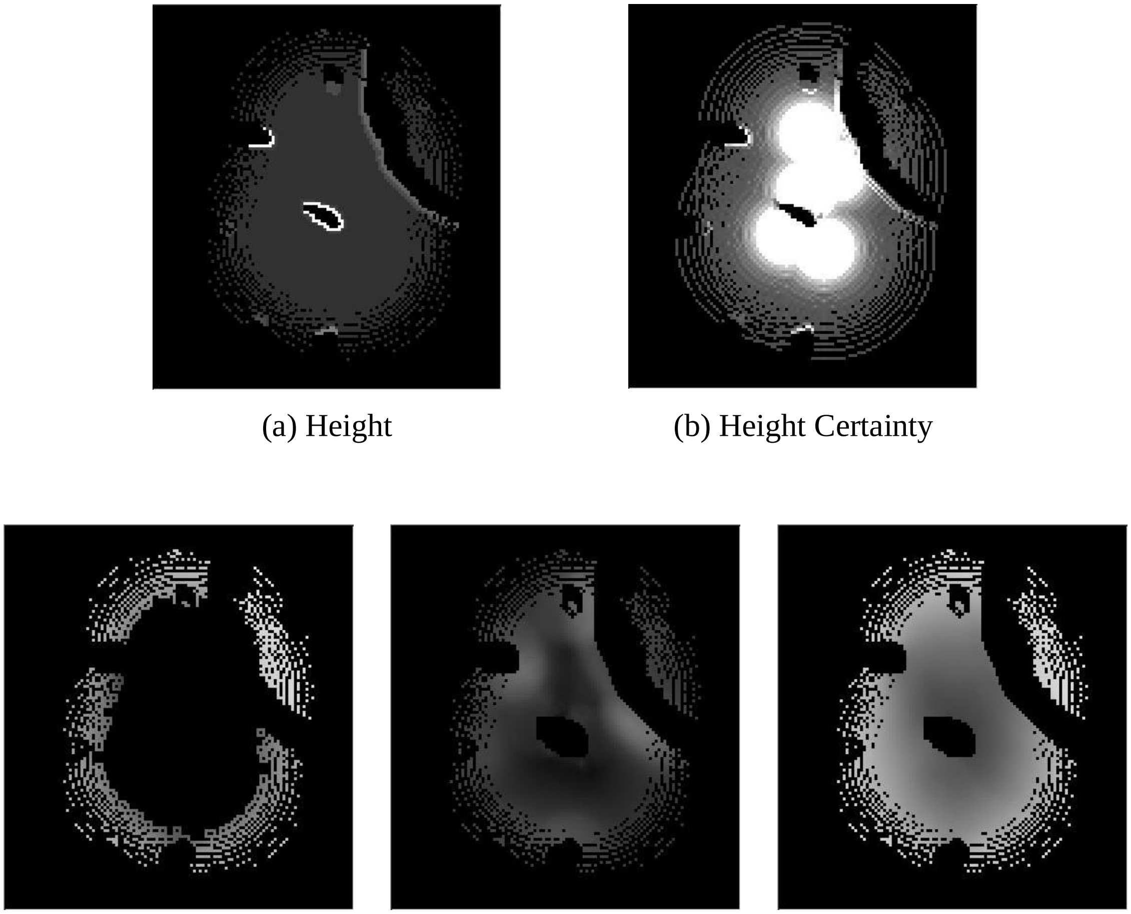
3.4 Information Metrics

As with the attribute vector, the information gain vector, $\boldsymbol{G^m}$, helps determine what exploration task will be performed. The elements in the information gain vector are the information metrics that have been chosen for the exploration problem. These metrics quantify what the robot will find interesting during the exploration. Without loss of generality, the elements of the vector are normalized to be in the range of zero to one. By knowing that the maximum information to be gained is one, the path planner can stop considering paths with costs that would yield negative utility given the value of information parameter and an information gain of one. This limits the distance of paths the planner needs to consider.

The actual sensor being used to collect information is not important for the exploration methodology, however some assumptions have been made in the development of the information metrics presented. First, it is assumed that the sensor has a 360 degree field of view in azimuth. This removes the need to plan the heading of the robot. Further, it is assumed that the height and range of the sensor is known.

The expected information gain in a cell is the amount of information that would be collected by taking a sensor reading in that cell. Therefore it is necessary for the information metrics to know, or speculate, which map cells will be seen from a given location. Let V_m be the set of cells visible to the sensor from cell m. To compute V_m we first get the set of cells W_m which contains all the cells inside a circle centered at m with radius equal to the maximum range of the sensor. The number of cells in W_m is denoted $\#W_m$. For each cell, n, in W_m a ray is traced back to cell m using the efficient Bressenham's Algorithm from computer graphics [19]. If the height of this ray is lower than the height in any cell it passes through then cell n is not visible and not in V_m. Otherwise cell n is added to the set V_m. Thus the shadowing effects caused by known obstacles are taken into account. Unknown cells are assumed to contain no features which obstruct sensor viewing. This assumption was made for two reasons. First, the robot must assume either obstructing or not obstructing and since it knows nothing about these cells either option is equally likely.

Secondly, if unknown cells were assumed to obstruct sensor views, the robot would never expect to see any unknown cells limiting its use as an explorer of unknown worlds.



(a) Height

(b) Height Certainty

(c) Frontier

(d) Increase Map Certainty Weighted by Traversability

(e) Determine Reachable

Figure 3-5: First Three Information Metrics. (a) Height values of the robot’s map. The higher a cell’s height the whiter its color. Black is unknown. (b) The certainty in the height values. Black is zero, white is one. (c, d, e) The expected information gains for the first three information metrics. Black is zero, white is one.

This thesis has implemented five information gain metrics, or information metrics, which are described below. As in the attribute section, these are not the only information metrics possible, nor are they required by all exploration tasks. These metrics are simply the ones needed in the examples that follow and are presented here to illustrate how information metrics are defined.

3.4.1 Frontier

The frontier information metric indicates how much unseen terrain the explorer can expect to see. This information metric is used to attract the robot explorer to the boundary of its known and unknown world and fill in the blank spots in its map. The greater the number of unknown cells expected to be viewed from cell *m*, the greater the expected frontier information gain. For a given cell, *m*, in the map the expected frontier information gain is calculated as:

$$g_f^m = \begin{cases} \dfrac{\sum\limits_{\forall n \in V_m} \mathrm{UNKNOWN}(a_{ct}^n)}{\# W_m} & ;\ \text{if m is a frontier cell} \\ 0 & ;\text{otherwise} \end{cases} \tag{3.3}$$

where a_{ct}^n is the traversability certainty in cell *n* and UNKNOWN(a_{ct}^n) is 1 if a_{ct}^n is less than a fixed threshold (set to 0.3 in this thesis) and 0 otherwise. A frontier cell is one which is traversable and the traversability is known and has at least one cell adjacent (in an 8 connected sense) to it which has an unknown traversability. Figure 3-5(c) shows the value of the expected frontier information gain for a partial map. Notice that the non-zero information gains are on the edge of the known and unknown world.

3.4.2 Increase Map Certainty Weighted by Traversability

The increase certainty weighted by traversability information metric rewards the explorer for increasing the density of sensor readings in a cell and thus increasing the height certainty. The information metric computes the expected increase in height certainty due to a sensor reading. It is more important to have high certainty about the terrain near obstacles so the information metric weights the expected increase in height certainty by the traversability. Height certainty is used because it can be predicted with a sensor model and traversability certainty is derived from height certainty. The equation for g_c is:

$$g_c^m = \frac{\sum_{\forall n \in V_m} w(a_t^n)(E[a_{ch}^n] - a_{ch}^n)}{\# W_m} \tag{3.4}$$

where $w(a_t)$ is a parabola which in this thesis had a value of 1 for zero traversability and 0.05 for traversability of one. $E[a_{ch}^n]$ is the expected value of the height certainty in cell$n$ after taking a sensor reading in cell *m*. The computation of $E[a_{ch}^n]$ depends on the sensor being used. This thesis assumes a sensor which takes range measurements with fixed angular increments in both azimuth and elevation. A sensor such as a laser scanner would fit in this class. For this type of sensor the number of readings in a cell (which is proportional to the cell's height certainty) is inversely proportional to the cube of the range [26]. The function of $E[a_{ch}^n]$ used in this thesis is shown in Fi gure3-6 and was computed using the kinematics of a laser scanner derived in [26] and a linear relation between the number of readings per cell and certainty. Since the height certainty cannot be greater than one Figure 3-6 has been limited to one indicating a region around the sensor yielding perfect certainty.

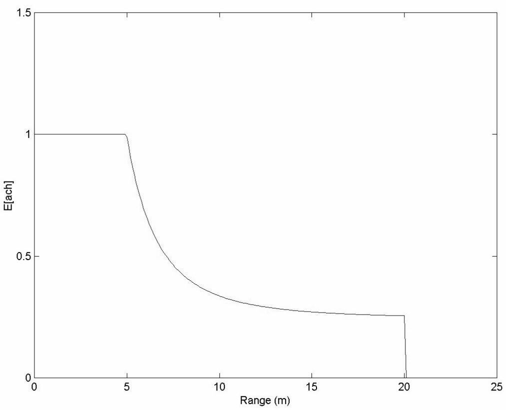


Figure 3-6: Graph of expected height certainty versus distance from sensor. For the sensor used in this thesis.

An example of the increase certainty weighted by traversability information metric can be found in Figure 3-5(d). Note how the values around the rocks and wall at the edges of the scene are high. Also note that the rock in the middle of the scene yields low increase certainty weighted by traversability scores because it is already well known as can be seen in the height certainty map shown in Figure 3-5(b). Thus the increase certainty weighted by traversability information metric encourages the robot explorer to see the world with greater care, particularly in regions that could be dangerous for the robot.

3.4.3 Determine Reachable

The reachability information metric rewards the robot for going to places which will most strongly impact its knowledge of reachability. It is calculated as:

$$g_r^m = \frac{\sum_{\forall n \in V_m} (-a_r^n \log(a_r^n) - (1 - a_r^n)\log(1 - a_r^n))}{\# W_m} \tag{3.5}$$

Since a_r^n is the probability that cell *n* is reachable, the numerator is the sum of the entropy over the sensor footprint [49]. This rewards the robot for viewing areas where the reachability is most uncertain (high entropy). See Figure 3-5(e).

3.4.4 Viewing Cliff Faces

The viewing cliff faces information metric rewards the robot for seeing the face of a cliff. This will be an important ability in exploration robots as scientists now believe they have found evidence of past water flow down the faces of some Martian cliffs [33]. In general it is not possible to view the cliff face from the top of the cliff, so the viewing cliff faces information metric only has non-zero values at the bottom of the cliff. The view cliff face information metric is defined as:

$$g_{cliff}^m = \begin{cases} \dfrac{\sum_{\forall n \in V_m} (E[a_{ch}^n] - a_{ch}^n) \cdot a_c^n}{\# W_m} & \text{; if not at the top of cliff} \\ 0 & \text{; else} \end{cases} \tag{3.6}$$

where $E[a^n_{ch}]$ is the expected height certainty in cell n as used in the increase map certainty weighted by traversability information metric.

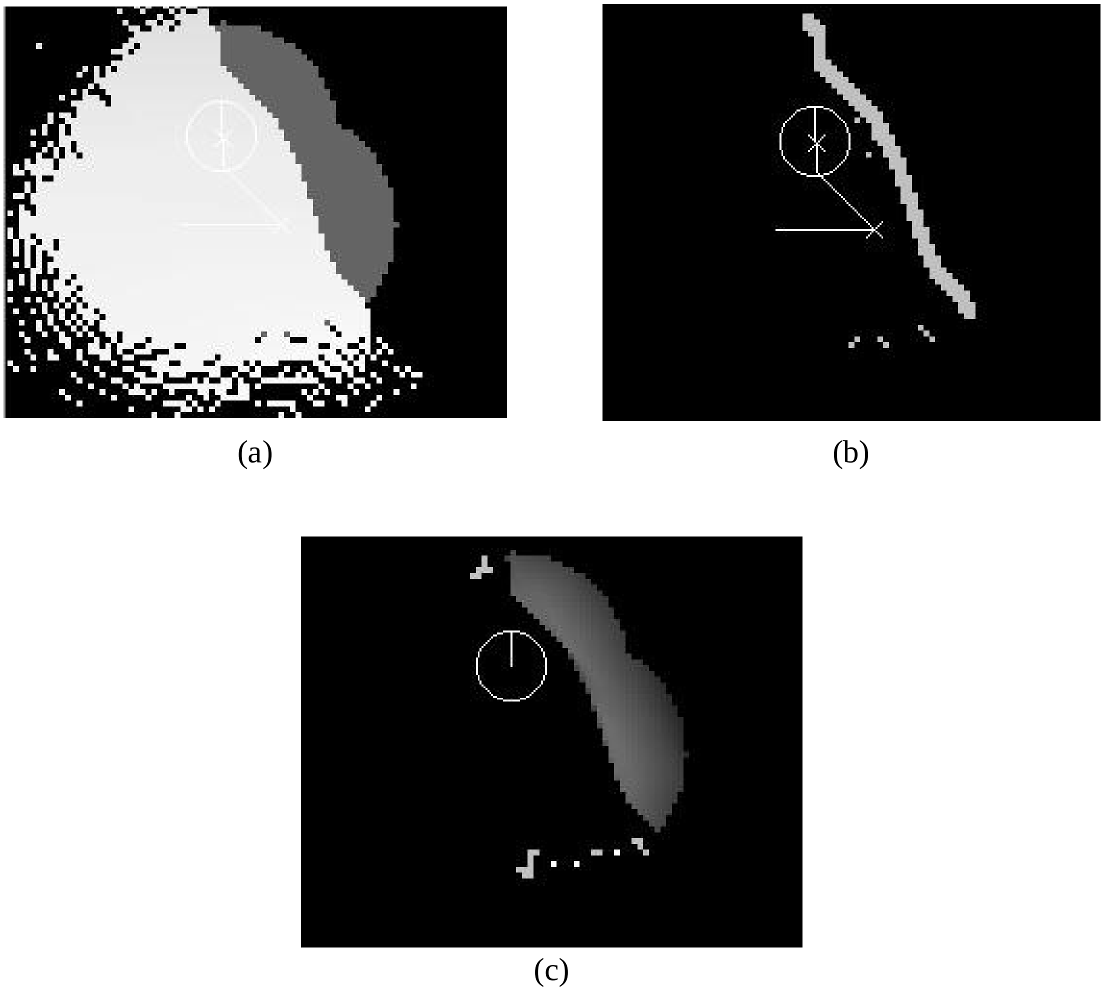
When an explorer is at the top of a cliff, it is unlikely that its sensor will be able to see the bottom of the cliff (unless it is a very short cliff). In this case the cells where g_{cliff} are nonzero have unknown heights. However, the sensor did view these cells, it just did not detect anything. Therefore, these cells are referred to as *perceived* cells and while the height is not defined, the height certainty is set using the sensor model found in Figure 3-2. This allows equation (3.6) to be defined even in places which are not yet known to the robot.

The intended exploration task for the viewing cliff faces information metric involves the robot starting at the top of a cliff and finding a way to the bottom so that it can view the cliff face. However, while it is at the top the regions of high information gain will be in the perceived cells. Since the heights and traversability of the perceived are not known, these high information gains are not seen by the path planner. They are not exerting a force bringing the robot down to see the cliff face. To overcome this problem, the view cliff faces information metric is modified slightly to find a way to the border of the cliff. Therefore, the view cliff faces information metric will also have a high value on cells at the top of the cliff which are near the cliff and the frontier as defined by the frontier information metric (equation (3.3)). This will serve to pull the robot along the cliff edge and down to the cliff base. This modification is illustrated in Figure 3-7(c).

3.4.5 Seek Lower Elevations

The seek lower elevations information metric rewards the robot for travelling to cells that have a lower elevation than the cell currently occupied by the robot. Many potentially important exploration tasks, such as looking at cliff faces from the base of a cliff or entering craters to search for water ice, can benefit from this simple information metric.

The seek lower elevations information metric is defined as:



(a)

(b)

(c)

Figure 3-7: Viewing cliff faces information metric. (a) Height attribute of map. The white cells are part of the cliff top. The grey cells are perceived cells, the laser passed through these cells but did not hit anything. This is because the cliff is here. (b) Shows the probability of a cell being a cliff. The main line is the actual cliff, the small grey blocks at the bottom are due to small dips in the terrain. (c) Viewing cliff faces information metric. Notice the high information gain in the perceived region. Also note the high information gain by the edge of the cliff and on the frontier of known world. This will help the robot follow the cliff and get to the perceived area.

$$g_{low}^{m} = \begin{cases} MIN(1.0, k \cdot (a_h^r - a_h^m)) & ;\text{if } (a_h^m < a_h^r) \\ 0 & ;\text{else} \end{cases} \tag{3.7}$$

where a_h^r is the height of the cell currently occupied by the robot, a_h^m is the height of the cell being assigned the information metric and k is a chosen constant. In this thesis k was empirically chosen to be 0.25.

CHAPTER 4 Simulation Results

The exploration methodology presented in the previous chapter can be applied to many different exploration tasks by the careful construction of the attribute and information gain vectors in the explorer's map. Changing the elements of the attribute vector changes what information the explorer records about the environment. Changing the information gain vector changes the metrics used by the explorer to determine what is interesting. Ultimately, the behavior of the explorer and the exploration task it performs is determined by the composition of these two vectors.

This chapter applies the methodology of Chapter 3 to two different exploration tasks: creating traversability maps and finding and viewing cliff faces. The compositions of the attribute and information gain vectors are presented along with the exploration paths generated in a simulation.

4.1 Software Architecture

A block diagram of the software architecture used for the simulation examples is shown in Figure 4-1. Each block is a separate process and the different processes exchange data using the Task Control Architecture (TCA) through the central module [56].

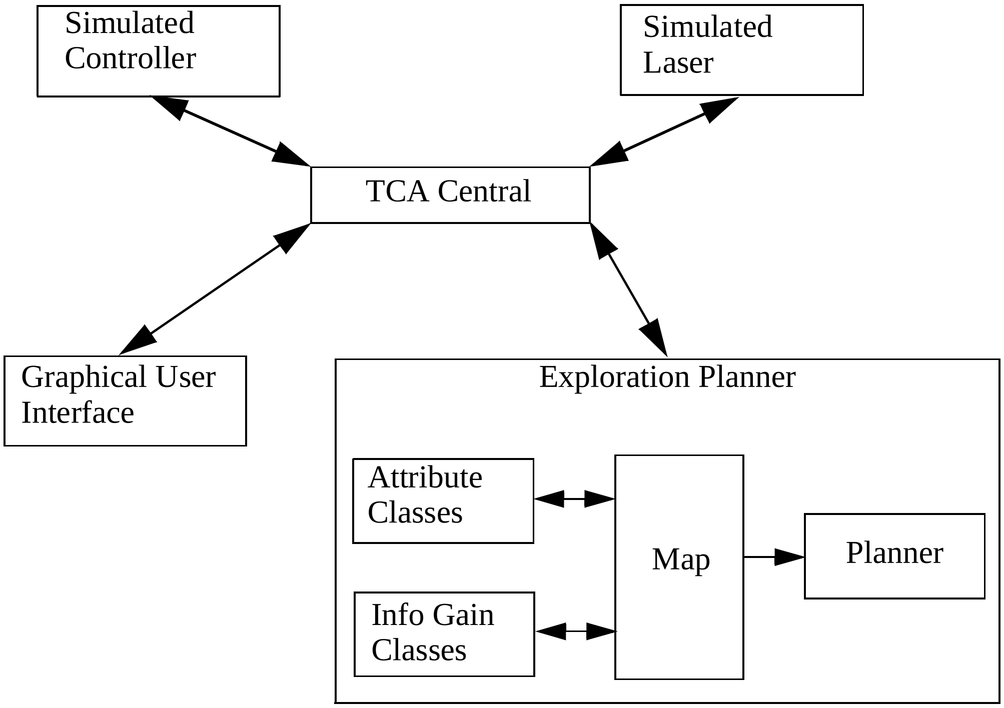


Figure 4-1: Block Diagram of Simulation Components.

The simulated controller process receives the current plan from the exploration planner. The plan consists of a sequence of positions to drive through. The final element in the plan is a scan request. The simulated controller executes the plan, stepping the robot through each desired location and then requesting a scan be taken by the simulated laser.

The simulated laser process simulates a sensor reading from a laser scanner. The simulated laser has a map of the actual environment being explored (as opposed to the exploration planner's map which represents the robot's knowledge of the world). The simulated laser is capable of reading maps in the United States Geological Service (USGS) Digital Elevation Map (DEM) format or as a two dimensional array of heights. The maps can be real locations mapped by the USGS or can be created using a stand alone graphical map drawing utility developed for this thesis.

The simulated laser process uses the sensor range, height above the ground, field of view in azimuth and elevation and the angular resolution (number of readings per degree) in azimuth and elevation. To take a scan a ray is projected from the current position of the sensor head onto the environment for each reading in the scan. The intersection point of each ray with the environment is found and converted into global Cartesian coordinates. The set of intersection points are then converted into an elevation map where each cell of the map holds the maximum elevation seen in that cell by the laser scan. The elevation map also contains a certainty reading for each cell which is proportional to the number of laser readings contained in the cell.

The graphical user interface is used to visualize the information contained in the exploration planner. Maps of the current values of each element in the attribute and information gain vectors can be displayed. The exploration path planner can be changed and the exploration paths are displayed.

The exploration planner block in Figur e4-1 is the process which implements the exploration methodology of Chapter 3. This process contains the robot's map which has the attribute and information gain vectors in each cell. The actual values in the attribute and information gain vectors are computed using attribute and information gain classes. By adding and deleting classes it is easy to re-compile the exploration planner and use it for a different exploration task. The exploration path planner also interacts with the map to generate paths which maximize the utility to the robot. The path planner is also implemented as a class allowing different planners to be used.

The software architecture of Figure 4-1 can easily be modified to use the exploration planning system on a real robot. The simulated laser module needs to be replaced with a driver for the actual sensor being used. This driver needs to convert the raw sensor data to the elevation map with certainties expected by the exploration planner. The simulated controller can be replaced by two processes on a real robot system. The first is just a process to broadcast, using a TCA message, the current pose of the robot. The second process would convert the generated exploration plans into robot driving commands. These commands could be combined with commands from a local navigation system to maintain robot safety. This type of architecture has been successfully employed to combine the D* grid based planner with Morphin a local navigation planner [59]. The exploration planner process would remain unchanged. The graphical user interface would also remain unchanged and using the power of TCA could also be run off board the robot.

4.2 Traversability Map Creation

The first exploration task presented is the creation of traversability maps. Traversability maps are maps which indicate how easy it is to drive over an area. They tell the robot where it is safe to drive and where it is not. The goal of the exploration robot in this task is to create a traversability map which is useful for other robots that might operate in this region at a later date. The traversability map created should have the traversability of a cell as well as how certain the robot is about that traversability. It is more important to know the traversability in rough terrain and near obstacles than in flat, benign terrain, both for path planning — if we think a narrow passage is traversable and it is not, this could drastically alter a plan — as well as robot safety — it is easier to damage the robot in rugged terrain than in flat terrain. Thus the explorer should be rewarded for increasing its certainty in low traversability regions more so than in high traversability regions. It would also be useful for another robot visiting the area to know which cells are reachable and which cells are not reachable. Finally, the explorer should record the height in each cell. This will allow robots to calculate sensor visibility regions.

Now that the exploration task is known, the elements of the attribute and information gain vectors must be chosen so that the explorer can succeed in its mission. The attributes are chosen so that all the data available from the sensors can be recorded. Also, attributes can be added that represent derived information (variables computed from the data provided from the sensors) that adds value to the finished map and may be useful in calculating the information gains. The information gain vector elements are chosen to represent each and every information metric which is important in solving the exploration task.

4.2.1 Problem Setup

For the traversability map exploration problem the attribute vector has the following elements: height, traversability and reachability. Both height and traversability are continuous valued variables and thus have certainties associated with them. The reachability is a binary valued attribute and is the probability that a cell is reachable. These attributes have been described in detail in section 3.3.

The information gain vector is determined by the goals of the exploration task. The exploration task is to create a traversability map of a region. Therefore, the robot needs to continually view new, unseen terrain. The robot also needs to spend extra time and determine the traversability of poor traversability areas with greater certainty. Finally, the robot should determine which areas in the region can be reached from the starting point. To fulfill these three goals, the information gain vector for this example uses three information gain metrics: frontier, increase map certainty weighted by traversability and determine reachable. These information gain metrics have been described in section 3.4. The composition of the attribute and information gain vectors is summarized in Ta ble4-1.

Table 4-1: Attribute and Information GainVectors

Attribute Vector	**Information Gain Vector**
Height, value and certainty	Frontier
Traversability, value and certainty	Increase Map Certainty, weighted by traversability
Reachability	Determine Reachable

The create traversability maps exploration task was performed on the crater world shown in Figure 4-2. The crater world is 300m x 300m and has a flat ground with discrete rocks protruding from the surface. Some rocks are small enough to be driven over, others are not. The crater world also has a large, shallow crater in the north west corner of the map. The crater is surrounded by an untraversable, low mound of debris. The mound is large enough to prevent the robot from driving over it but low enough that the robot’s sensor can see over top of it. A small opening in the northern part of the mound allows access to the interior of the crater which gradually slopes down for two meters. A five meter tall spire is located in the center of the crater.

The exploration was performed in the crater world using four different sets of weights for the three information metrics. The weights are the α_i multipliers from equation (3.1) that are used to denote the relative importance of the information metrics when computing the total expected information gain in a map cell. The weights used for the four runs are found in Table 4-2. Each of the weight sets were run from one of five different start locations indicated by the letters A through E on Figure 4-2.

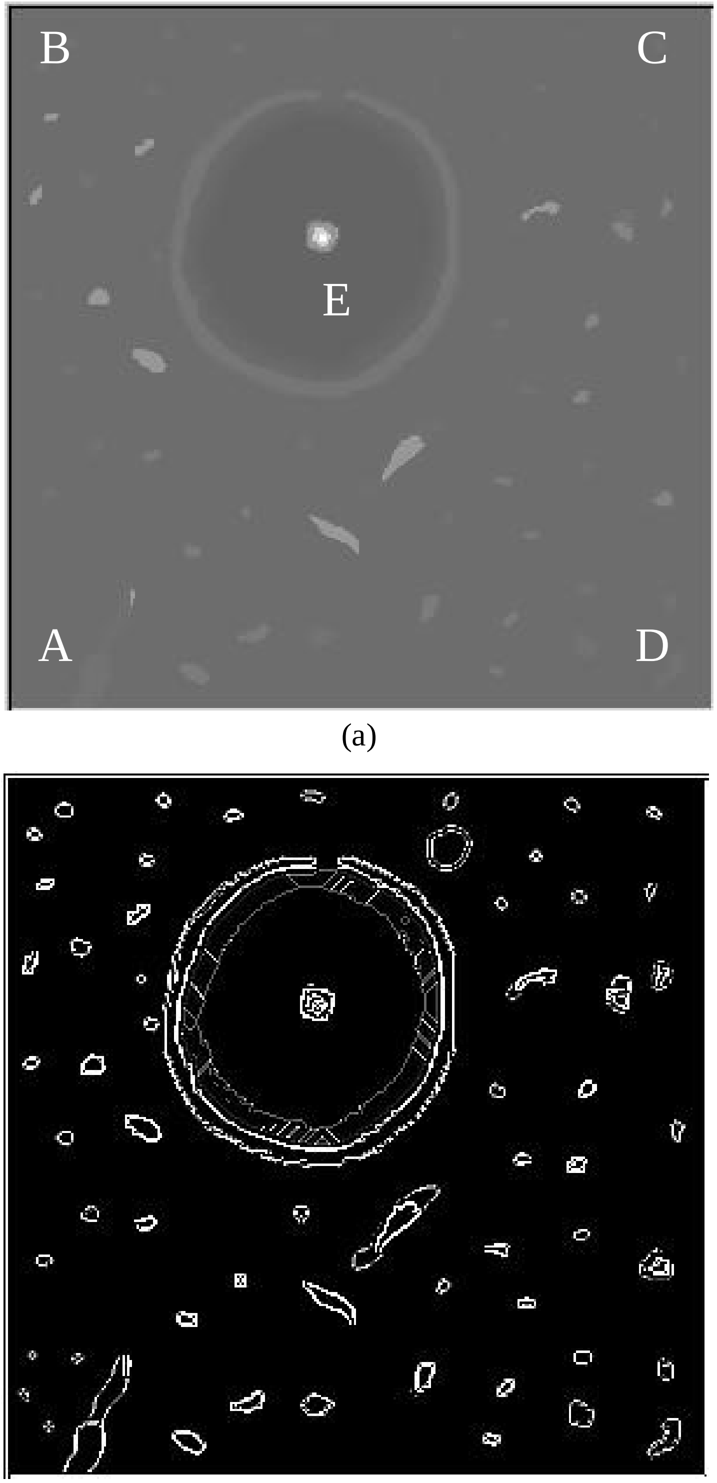
Table 4-2: Information Metric Weights for Four Runs

Information Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**
Frontier	0.33	1.0	0.0	0.85
Increase Map Certainty weighted by Traversability	0.33	0.0	1.0	0.15
Reachability	0.33	0.0	0.0	0.00

4.2.2 Results

The complete results from all four exploration runs from all five starting positions can be found in Appendix A. Sample results from starting point A of the four exploration runs can be found in Figure 4-3 through Figure 4-6. In all simulation runs the exploring robot started with a completely blank map, i.e. no *a priori* information about the environment. The figures show the values of the map attributes height (both its value and its certainty), traversability and reachability. The traversability certainty is not included in the figures since it is very similar to the height certainty map. Each pixel in the image is a single map cell. On the height image, the path taken by the exploring robot is drawn in white. The small white x’s on the map indicate where a sensor reading was taken.

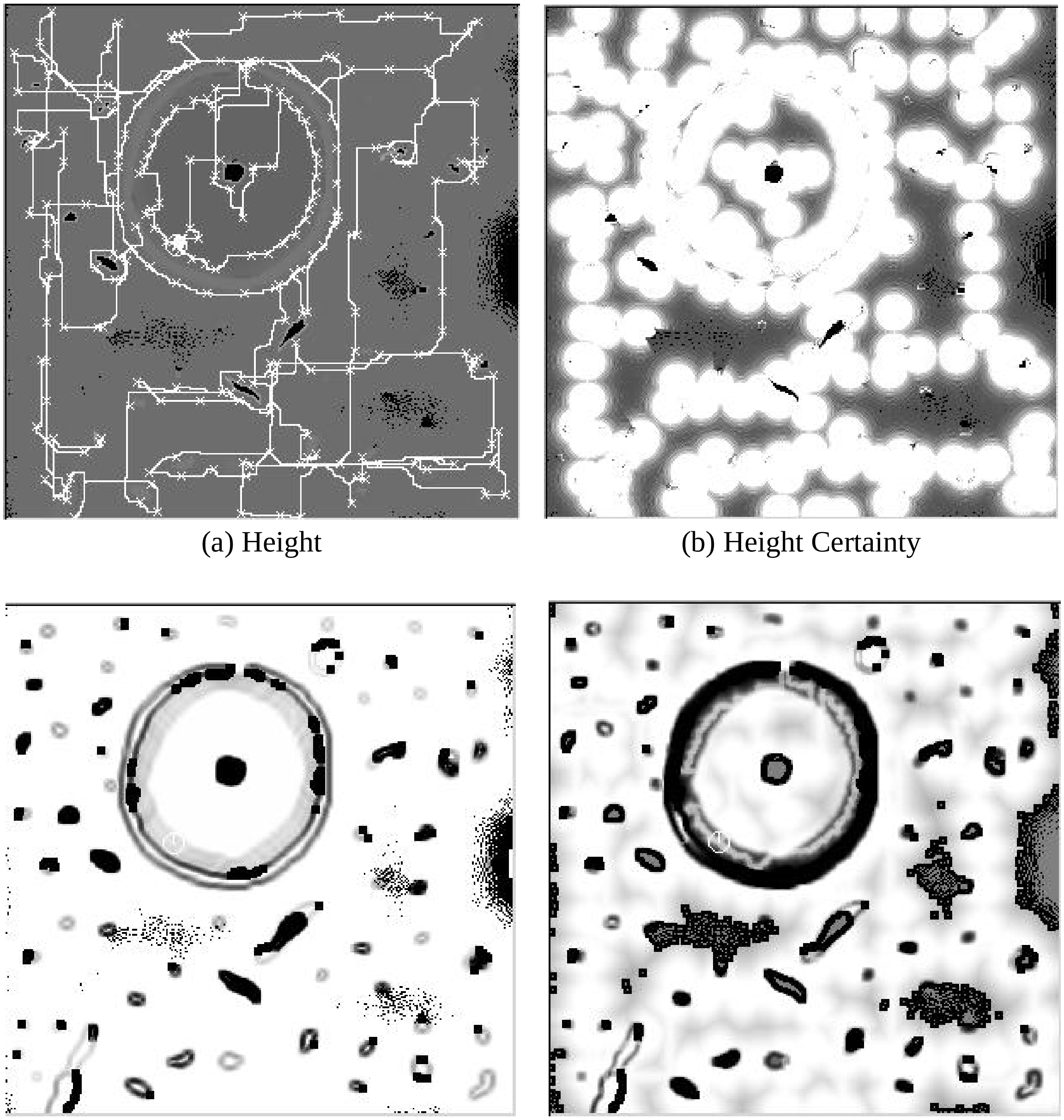
On the height and height certainty figures, the centers of some large rocks are black. This is because the rocks are taller than the sensor head and thus the robot cannot see the tops of the rocks. The black indicates no data in these cells. These cells are also black on the traversability maps, indicating that the terrain is untraversable. On the reachability maps however, the cells in the center of these large rocks are grey. Since the robot has no knowl-



(a)

(b)

Figure 4-2: Crater world simulation environment. (a) Elevation map - whiter cells are higher elevations. The letters indicate the five starting positions used in the simulations. (b) Edge map. It is difficult to see the low rocks in (a) so this edge map is included to help the reader understand the environment.



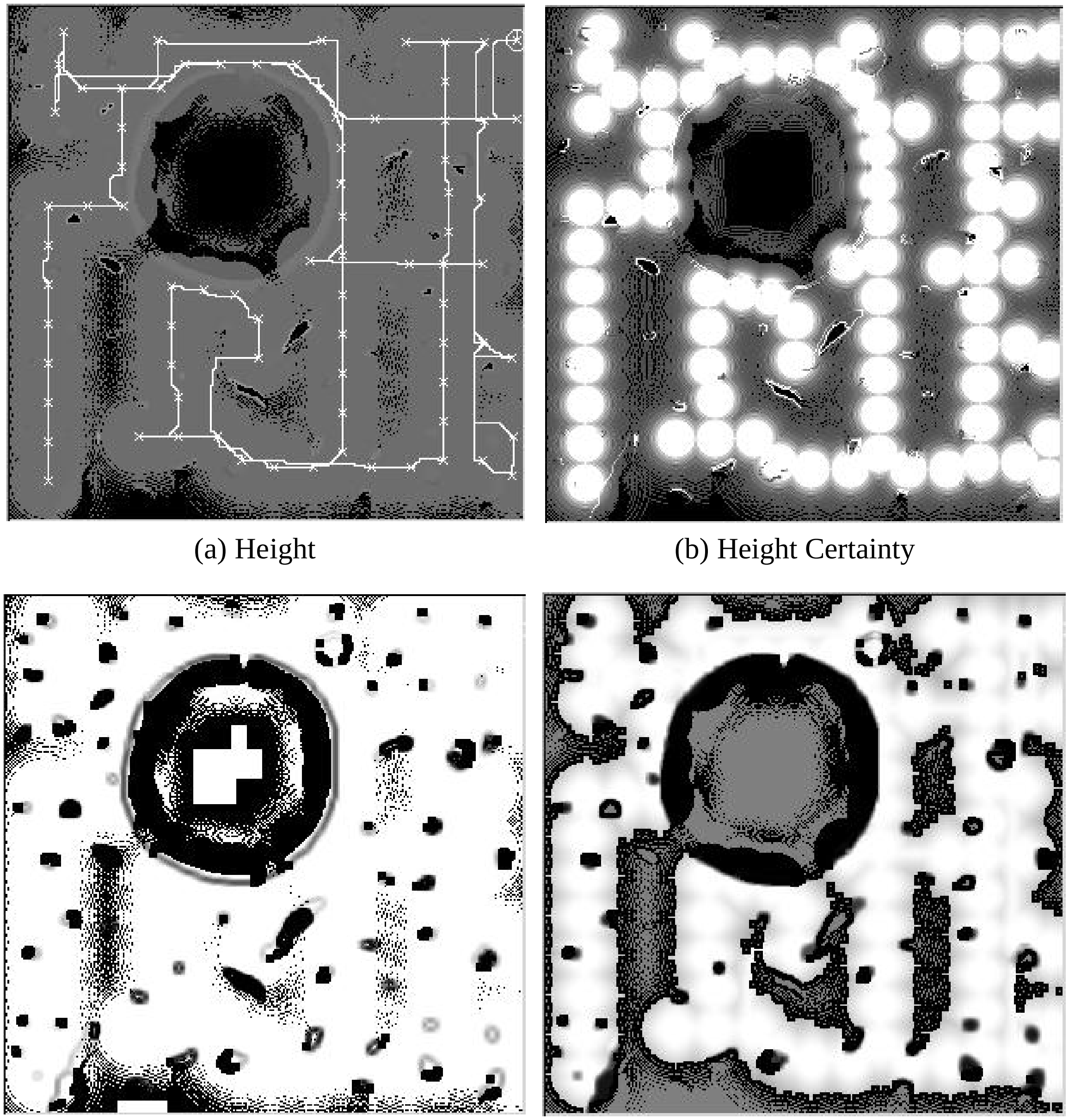
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure 4-3: Run 1a Path Results. Results of the traversability map creation exploration in the crater world simulation environment with all three information metrics weighted equally. In all images larger values are whiter. In the height and height certainty only, black indicates no data. Traversability certainty is similar to the height certainty image. Notice the high certainty around the crater rim and most rocks.



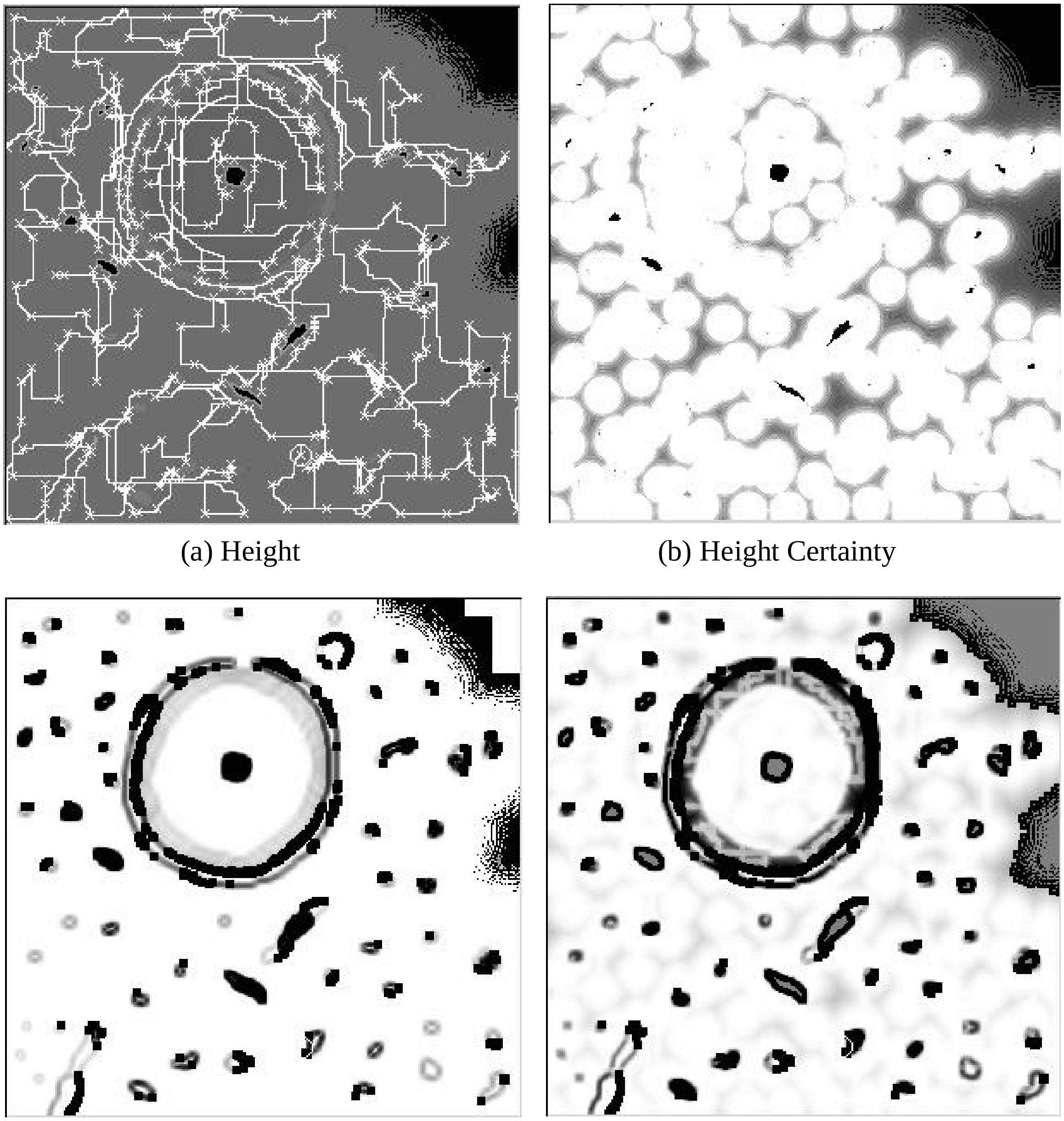
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure 4-4: Run 2a Path Results. Results of traversability map creation using only the frontier information metric. In all images larger values are whiter. In the height and height certainty only, black indicates no data. Traversability certainty is similar to the height certainty image. Compared to exploration with all three information metrics equally weighted, sensor readings are taken at fairly regular intervals. Certainty around the crater rim and rocks not as high as in previous example.



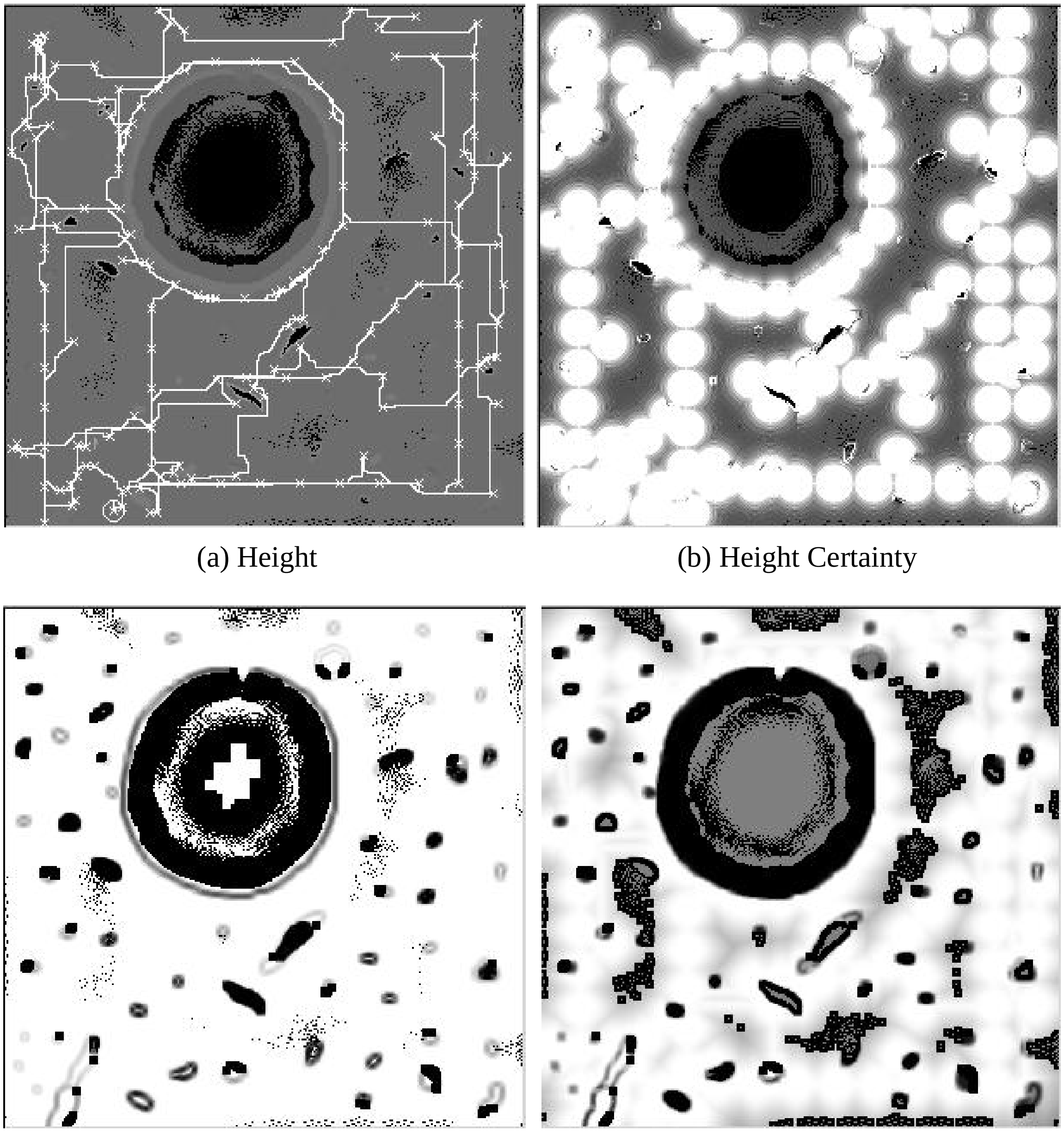
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure 4-5: Run 3a Path Results. Results of create traversability map exploration using only the information metric increase map certainty weighted by traversability. In all images larger values are whiter. In the height and height certainty only, black indicates no data. Traversability certainty is similar to the height certainty image. Notice the much higher density of sensor readings than previous two examples. Also notice the lower sample density in the benign region to the south west of the crater (north is up).



(d) Reachability

Figure 4-6: Run 4a Path Results. The addition of a small weight to the Increase Map Certainty Weighted by Traversability causes the explorer to examine obstacles more closely than in run 2 which only used frontier. However it does not take samples as often as runs 1 or 2.

edge of these cells, the probability of these cells being reachable is 0.5 which is drawn as grey.

The results of the run 1a experiment, where all three information metrics are weighted equally, is shown in Figure 4-3. This presents a very balanced exploration of the crater world. The robot covers a lot of ground quickly, viewing most of the environment. By comparing the black areas (very low traversability) in the traversability image with the height certainty image it is apparent that most low traversability regions have been sensed with high certainty (the white regions in the height certainty map). The robot is too short to see the tops of some of the larger rocks which is why the centers of these rocks appear as no data in the height certainty map. This effort to see poor terrain well is very visible around the large rock immediately south west of the crater and the two large rocks to the south of the crater. These rocks were too tall for the robot to see over and so the robot took sensor readings on all sides of the rocks to view their extent.

Another point of interest in the run 1a experimental results is the path taken by the robot around the crater rim. The robot follows the crater rim, both on the inside and outside and takes many sensor readings. There are two reasons for the robot’s interest in the crater rim. First, the rim is an obstacle and as such the robot wants to be certain about it because of the increase map certainty weighted by traversability information metric. The second reason is the determine reachability information metric. The robot first followed the outer side of the rim. The rim is an obstacle, but is low enough to see over and into the crater. The robot can thus see some traversable terrain inside the crater which it thinks is unreachable. The robot’s knowledge of the crater rim ends at the edge of its sensor footprint so the robot explorer does not know if the crater rim continues or ends. Therefore the determine unreachable information metric draws the robot along the crater rim to where the robot’s knowledge of the rim ends in the hope of finding a path to the traversable terrain inside the crater.

The results of the run 2a experiment (Figure4-4) shows how the robot explored the crater world with only the frontier information metric. This exploration would be similar to that found in the methods of Yamauchi [75] and Simmons et al. [58]. Since the frontier information metric has non-zero values only on the boundary of the known world, the exploration took sensor readings at fairly regular intervals corresponding to the senor range. As with run 1a which used three information metrics, run 2a did explore most of the map (although it failed to enter the crater). However, comparing the height certainty images for runs 1a and 2a shows a major difference both in the general quality of knowledge of the environment but run 2a does not make any effort to get high quality data of poor traversability regions. In contrast to run 1a, run 2a only sees the large rock immediately south west of the crater from a distance and does not even see the entire crater rim.

The run 3a experiment (Figure 4-5) goes to the other extreme and shows an exploration run which only considers increasing map certainty weighted by traversability. As in run 1a, extra attention is given to the crater rim and the large rocks. However, without the frontier information metric trying to push the robot to the edge of the known, run 3a takes many densely packed sensor readings.

The run 4a experiment (Figure 4-6) uses primarily the frontier information metric but adds a small amount of the increase map certainty weighted by traversability information metric. As expected, this exploration covers ground more quickly than runs 1a or 3a and the addition of a small amount of increase map certainty weighted by traversability causes the explorer to examine some obstacles in more detail than was found in run 2a. Note how the explorer pays special attention to the outside of the crater rim and the large rocks to the south of the crater. However, unlike run 1a, run 4a does not enter the crater.

To quantitatively compare the experimental runs it is useful to look at the amount of information contained in the final maps of each of the exploration runs as well as to compare the exploration path lengths and number of samples taken. To analyze the information content in the final maps, three information measures are used: height information, I_h, height information weighted by traversability, I_{tw}, and reachability, I_r. These three information measures roughly correspond to the three information metrics used in the exploration.

The equations used to compute the map information contents are:

$$I_h = \sum_{\forall m \text{ in map}} a_{ch}^m \tag{4.1}$$

$$I_{tw} = \sum_{\forall m \text{ in map}} w(a_t^m) \cdot a_{ch}^m \tag{4.2}$$

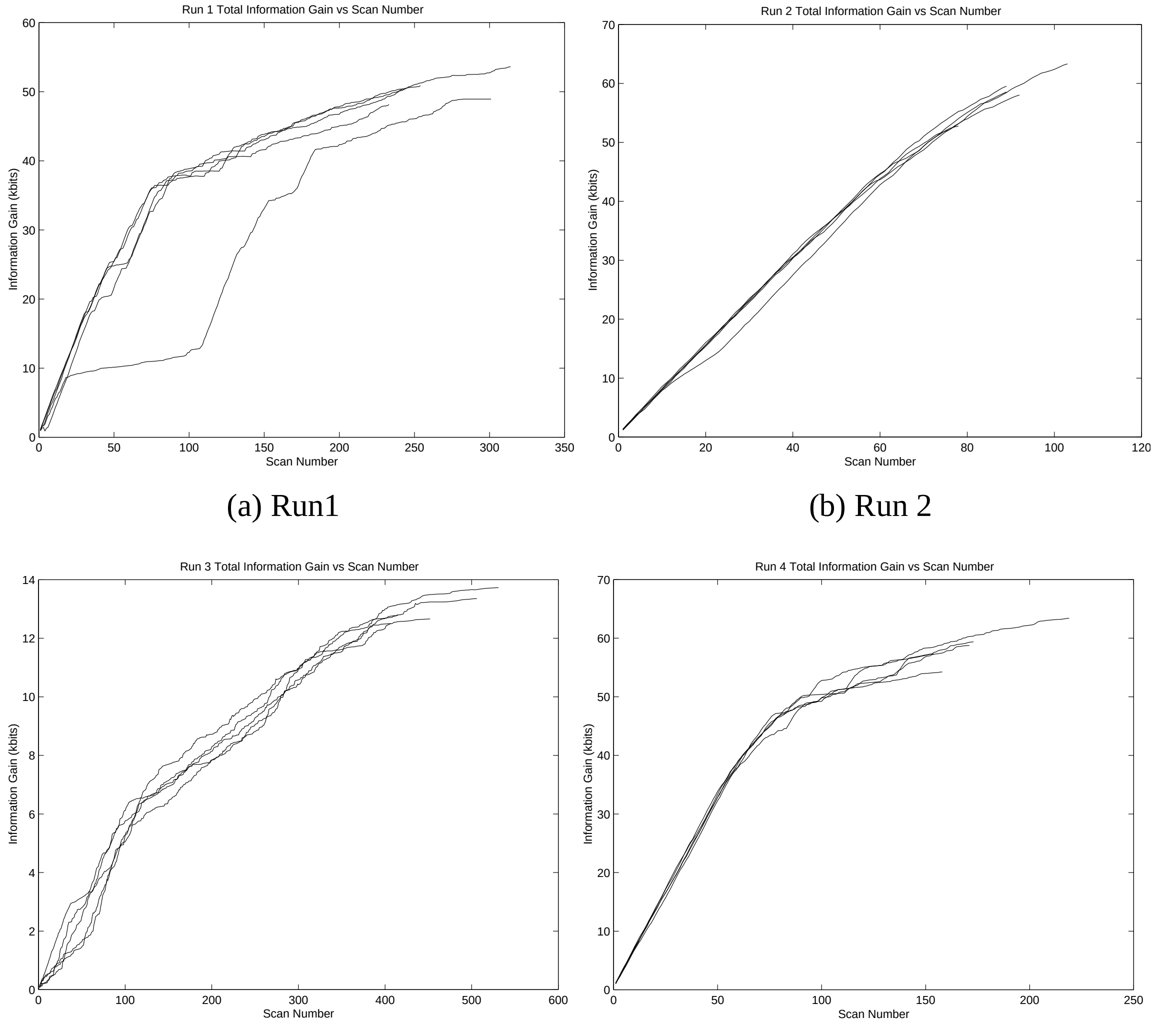
$$I_r = \sum_{\forall m \text{ in map}} (1 - ENTROPY(a_r^m)) \tag{4.3}$$

where a_{ch}^m, a_t^m and a_r^m are the height certainty, traversability and reachability of cell m as defined in section 3.3. *ENTROPY*( ) is the information theoretic entropy of a binary valued variable as defined in equation (2.1). Finally, *w*( ) is the same traversability weight parabola used in the increase map certainty weighted by traversability information metric (see section 3.4.2).

The map information contents for the four experimental runs at each starting location can be found in Appendix A. Table 4-3 summarizes this data by providing the average information content for each experimental run. The average for each run is computed over the five staring location. The graphs in Figure 4-7 show the total map information content for each run where the total information content is computed by summing the three information scores (equations (4.1)-(4.3)) with the same α_i weights used in summing the information gains (seeTabl e4-2). Finally, the costs of each exploration can be found in Appendix A and the average costs for each experimental run are presented in Table 4-4 below. The normalized path length is the total path length of the explorations divided by the largest dimension of the environment size, in this case 300 m.

Table 4-3: Create Traversability Map Average Exploration Gains

Final Map Information	**Run 1 (kbits)**	**Run 2 (kbits)**	**Run 3 (kbits)**	**Run 4 (kbits)**
Height Information	74.4	58.4	82.0	67.3
Height Info weighted by Traversability	11.14	8.1	13.0	9.76
Reachability Information	67.3	64.8	74.5	65.1



(a) Run1

(b) Run 2

(c) Run 3

(d) Run 4

Figure 4-7: Information gained vs. scan number. Graphs show how the total information contained in the robot’s map increases with each sensor scan. The four graphs correspond to the four information metric weights used. Each graph has five plots corresponding to the five starting locations (A through E). The total information is computed to reflect the weights given to the information metrics during the exploration. (a) Total information is computed as 0.33*I_h + 0.33*I_{tw} + 0.33*I_r. Note that the plot which is different than the other four corresponds to the starting location E, inside the crater. In this case, the robot gets bogged down looking at the crater instead of quickly finding new terrain. However, once outside the crater, the exploration recovers. (b) Total information is computer as I_h. (c) Total information computed as I_{tw}. (d) Total information is computed as 0.85*I_h + 0.15*I_{tw}.

Table 4-4: Create Traversability Map Average Exploration Costs

Exploration Costs	**Run 1**	**Run 2**	**Run 3**	**Run 4**
Path Length (meters)	4006	2084	6303	3146
Normalized Path Length (path length / site side length)	13.4	6.9	21.0	10.5
Number of sensor readings	266	90	462	175
Avg. Distance between Sensor Readings (m/scan)	15.1	22.9	13.7	18.0

4.2.3 Conclusions

The above results show that the multiple information metrics exploration planner is capable of performing the Create Traversability Map exploration problem. The qualitative results obtained by looking at the final exploration paths between the exploration runs demonstrate that the information metrics are being used concurrently and are performing their intended role. The frontier information metric is pulling the robot to the edge of the unknown and increasing the distance between sensor scans. The increase map certainty weighted by traversability, on the other hand, causes the robot to focus its attention on trouble spots and take scans more frequently. This claim is also supported by the average distance between sensor readings for the runs found in Table 4-4 and in Appendix A.

The quantitative results found in Tables 4-3 and 4-4 and in Appendix A support the qualitative analysis of the exploration results. On average, run 2 collected the least amount of information in its map for all three information categories but expended much less effort that the other three to collect it. Run 3 expended the greatest amount of effort, driving almost three times as far as run 2, but it also collected the most information. While run 3 collected more information in the height information category than run 1, it also gained significantly more information in the increase map certainty weighted by traversability. On average run 4 produces results between run 2 and run 1.

The graphs in Figure 4-7 how the information contained in the explorer’s map increases with each sensor scan. It is remarkable how similar the rates of information gain are for each starting location in an experimental run. The one exception is for starting location E in run 1 (Figure 4-7(a)). This is where the robot starts inside the crater. For the first 25 or so sensor scans, this run gains information at a similar rate to the other starting locations of run 1. However, at this point the robot has viewed most of the crater. It then spends the next 75 or so sensor scans examining the rough terrain inside the crater more carefully. This causes the rate of information gain to decrease since it is not seeing new terrain but rather examining previously seen terrain more closely. At around scan 100, the explorer finally leaves the crater and resumes collecting information at a high rate. It is interesting to note that this pattern did not occur for starting location E on run 4. In this case, the higher weighting given to the frontier information metric causes to the explorer to leave the crater more quickly.

The graphs also show that the rate of information gain decreases with time, although it is a slight decrease for run 2 (Figure 4-7(b)). What is interesting is that runs 1 and 4 gain information at essentially two fixed rates. While the reason for this is not certain, it is possible that the initial high rate of information gain occurs when unknown areas of the map are being viewed. The lower rate then takes over when the map is more or less known and the explorer is examining rough terrain more closely.

4.3 Finding and Viewing Cliff Faces

To demonstrate the flexibility of the multiple information metrics exploration planner to solve varied exploration tasks, a second exploration problem will be studied. In this case, the explorer is asked to find a cliff and then to view the face of that cliff. To view the face of the cliff, the robot must take sensor scans from the base of the cliff. In the example illustrated below, the exploration robot will start at the top of the cliff. It must determine where the cliff is, find a way to the bottom and then take sensor readings along the base of the cliff to view its face.

4.3.1 Problem Setup

The first step is to determine the composition of the attribute and information gain vectors needed to solve the specified problem. The attribute vector needs to contain the information collected which is pertinent to the problem. Clearly, the height is important to record as this is the raw data collected by the sensor. Height is a continuously valued variable and so it has a value and a certainty in the attribute vector. Another important variable to record is the probability that a cell is part of a cliff. Since the exploring robot is trying to find cliffs, it should record where it sees them. While this robot is not trying to create a traversability map as in the previous example, traversability will be stored in the attribute vector, not so much as a final product but to be used during the exploration when planning paths. These three attributes are discussed in detail in section 3.3.

The information gain vector is determined by the goals of the exploration. Clearly, the robot needs to view unseen terrain and it needs to view cliff faces. Thus two elements in the information gain vector are the frontier and the viewing cliff faces information metrics. Another information metric which may prove to be useful is the seek lower elevations metric. Since the robot is trying to view the face of a cliff and it must do this from the base of that cliff, there is value to the robot to go to locations which are lower in height or elevation than its current position. Therefore the information gain vector used in this problem contains the elements: frontier, viewing cliff faces and seek lower elevations. These metrics are detailed in section 3.4. The weights of each information metric (the α_i multipliers from equation (3.1)) used were: 0.2 frontier, 0.7 viewing cliff faces and 0.1 seek lower elevations.

The finding and viewing cliff faces exploration was performed on the cliff world shown in Figure 4-8. The world is 300m x 300m in extent with a cliff ranging from 30 to 15m high winding from the north east corner to the southern edge of the map. A path to the cliff base is found approximately in the center of the cliff. As in previous images, the whiter a pixel the higher its elevation. The cliff is not perfectly vertical and the distance from the top of

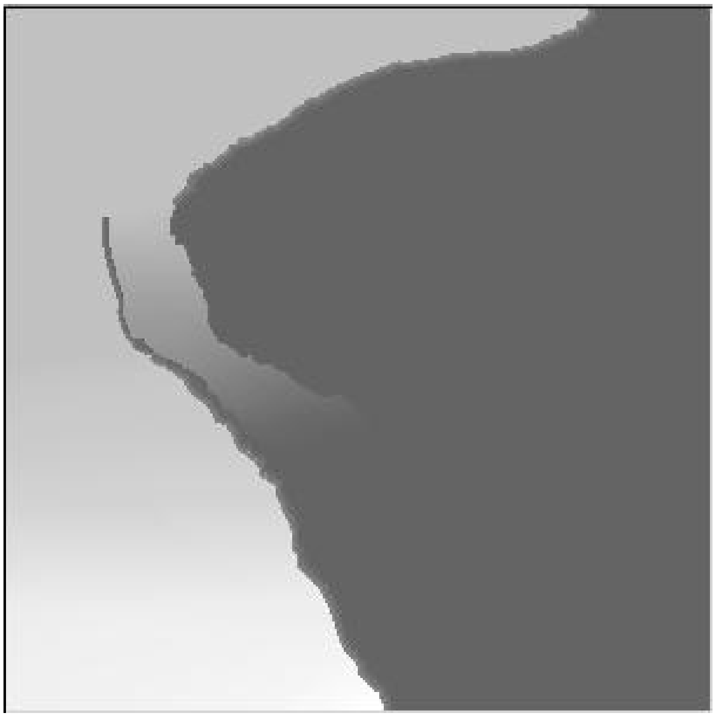


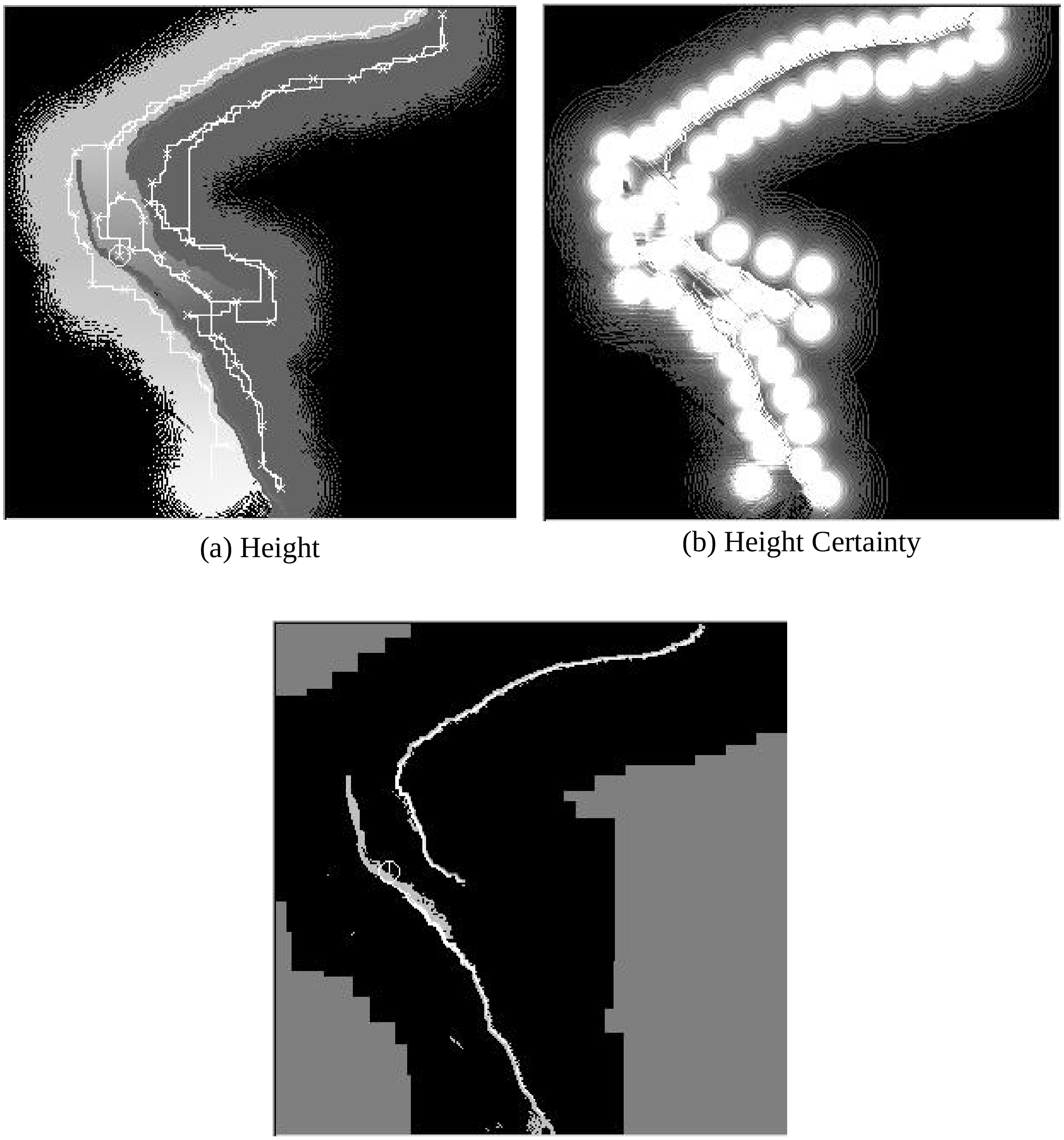
Figure 4-8: Cliff world. Elevation map of the cliff used in simulation tests of the cliff exploration.

the cliff to the base of the cliff, in a direction normal to the cliff face, varies from one to three meters.

4.3.2 Results

The results of the exploration of the cliff are shown in Figure 4-9. Each pixel in the images corresponds to one map cell. The robot started its exploration on the southern edge of the map and is at the top of the cliff.

From the top of the cliff, the robot cannot see the bottom. Instead it notices a large region where it thinks it should see something (i.e. nothing in its map is blocking or shadowing this region) but it does not. The robot calls this type of terrain, perceived terrain, its sensor has gone through that region but no returns or measurements were made. The most likely cause of this perceived terrain is a negative obstacle or hole. The robot then hypothesizes that the reason for this perceived region is a cliff and that the robot is on the top. Next, the robot follows along the cliff edge due to high information gains from the cliff information



(a) Height

(b) Height Certainty

(c) Probability of being a Cliff

Figure 4-9: Cliff exploration results. (a) Height map with the exploration path superimposed. The small x’s indicate sensor scan locations. (b) Height certainty. Notice that the exploration has high certainty (whiter cells) along the cliff. (c) Cliff attribute. Probability of a cell being part of the cliff. White cells are high probability, black cells low probability, grey is 50% probability, indicating no information.

metric. Eventually, the robot finds the path down and takes it, aided by the seek lower elevations information metric.

4.3.3 Conclusions

The robot explorer successfully explored the cliff world, viewing the entire face of the cliff. The robot was able to postulate the existence of the cliff from the top, without being able to see the bottom and follow this potential cliff until it found a route to the base. Once at the base the robot then proceeded to drive along the cliff face and take sensor readings.

This example, which is much different in purpose and structure than the create traversability map example above, also illustrates the flexibility of the multiple information metrics exploration planner to perform different exploration tasks by appropriate choices of the attribute and information gain vectors.

CHAPTER 5

Field Results

The previous chapter showed the capabilities of the multiple information metric exploration planner in simulation. While the simulations demonstrated the functionality of the approach, they were performed in hand drawn worlds and did not consider sensor or position noise. This chapter demonstrates the ability of the multiple information metric exploration planner to explore a real cliff using real sensor data.

5.1 Experimental Setup

The terrain sensor used in the experiment was a laser scanner from Zoller + Frohlich (Z+F) [29]. The Z+F laser sensor can scan 360 degrees in azimuth and ±30 degrees in elevation with a reading taken every 0.045 degrees in azimuth (for a total of 7999 readings) and 0.043 degrees (for a total of 1400 readings) in elevation. The Z+F laser has a maximum range of 25.2m with a range resolution of 0.38mm.

The laser was placed on a cart which was instrumented with differential GPS for position information and a compass and inclinometers for heading, roll and pitch information (see Figure 5-1). Using the pose information, each laser scan was converted to a global reference frame and combined into a single global map used for planning the exploration paths. The cart also held a computer to read the laser scan and a second computer to run the

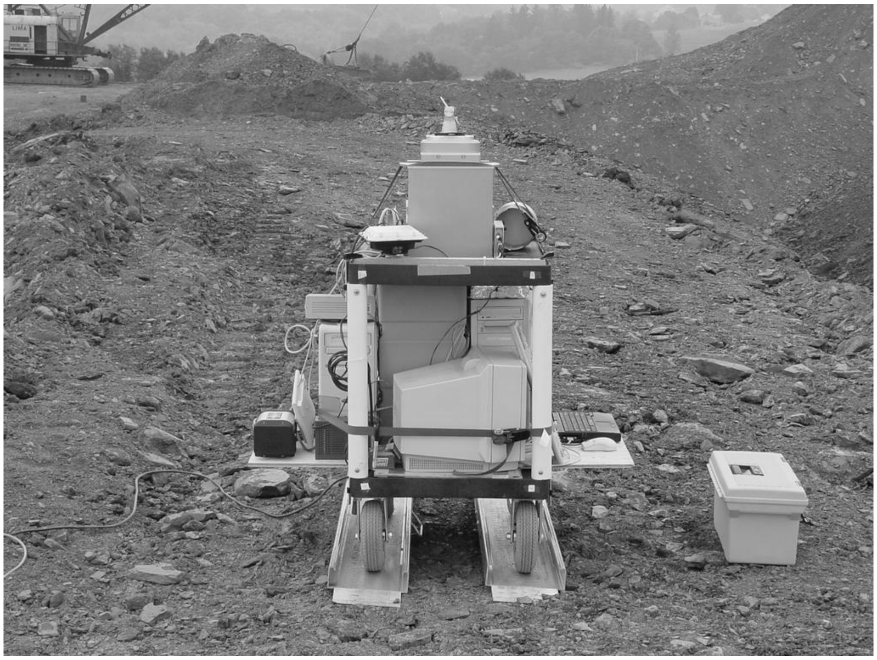


Figure 5-1: Experiment Platform. The brass colored cube on top is the Z+F laser.

exploration planner. All processing was done on-board the cart and only power was off-board.

A push cart provided an ideal platform to test the exploration planner. The cart was large enough to hold all of the computers and sensors needed for the exploration. Using the cart rather than a full robot, such as Nomad [71], allows the exploration planner to be tested without the complications of low level motor controllers and local navigation planners such as Morphin [57]. Several other investigations have shown that it is possible to combine a grid based planner with a local navigator so the simplification of using a cart is valid for the purpose of testing the exploration planner [22][59].

The exploration planner used three information metrics: frontier, viewing cliff faces and seek lower elevations which are described in more detail earlier in this document. The weights of each information metric (the α_i multipliers from equation (3.1)) were: 0.2 frontier, 0.7 viewing cliff faces and 0.1 seek lower elevations. The attribute vector used for the experiment had elements for probability of cliff, height and traversability. Height and traversability were continuous valued variables with a value and a certainty. It should be

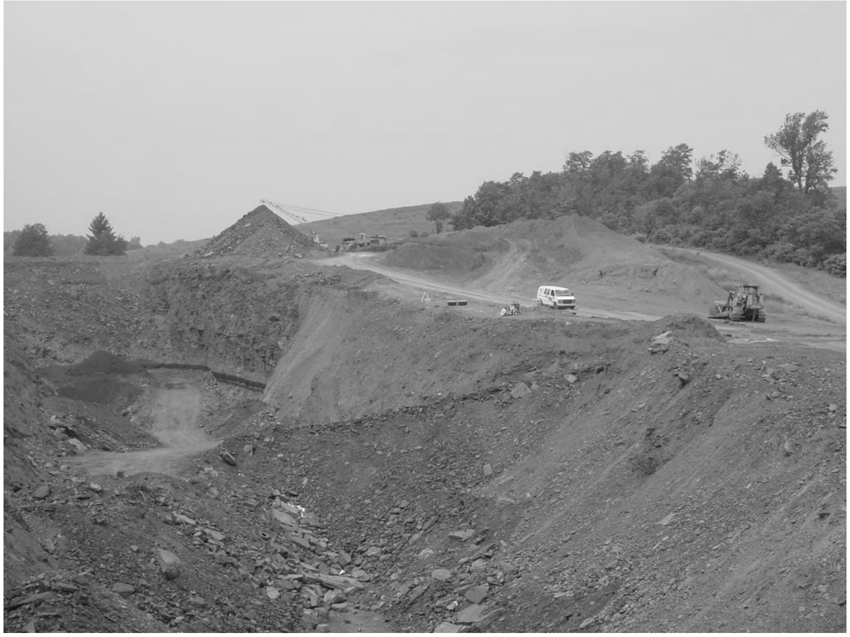


Figure 5-2: Test site.

noted that the information metrics and weights used here are identical to those used in the simulation experiment. In fact the exploration planner code used for the field test is identical to that used in the simulation. The only changes needed in the architecture of Figur e4-1 was a laser driver for the Z+F laser to replace the simulated laser and a process to enter the cart pose to replace the simulated controller.

The experiment was performed at a coal strip mine operated by Fieg Brothers Coal in Somerset Co. Pennsylvania. The mine is a large hole approximately 15m deep and 35m wide (see Figure 5-2). The diagonal line of dark rock in the lower middle part of Figure 5-2 is the route to the bottom of the cliff. In general the ground was comprised of hard packed soil, except along the top of the cliff and right along the base of the cliff where the soil was soft, making it difficult to move the cart.

5.2 Experimental Procedure

The procedure used to explore the cliff was as follows. First a scan was taken with the Z+F laser. The scan was tagged with the cart’s pose and converted into a digital elevation map (DEM) based in the global coordinate frame. The DEM was then merged into the existing map of the exploration planner with the elevations in the DEM corresponding to the height variable of the attribute vector. As in the simulation experiments the height certainty was determined by the number of laser readings from the current scan falling in the map cell.

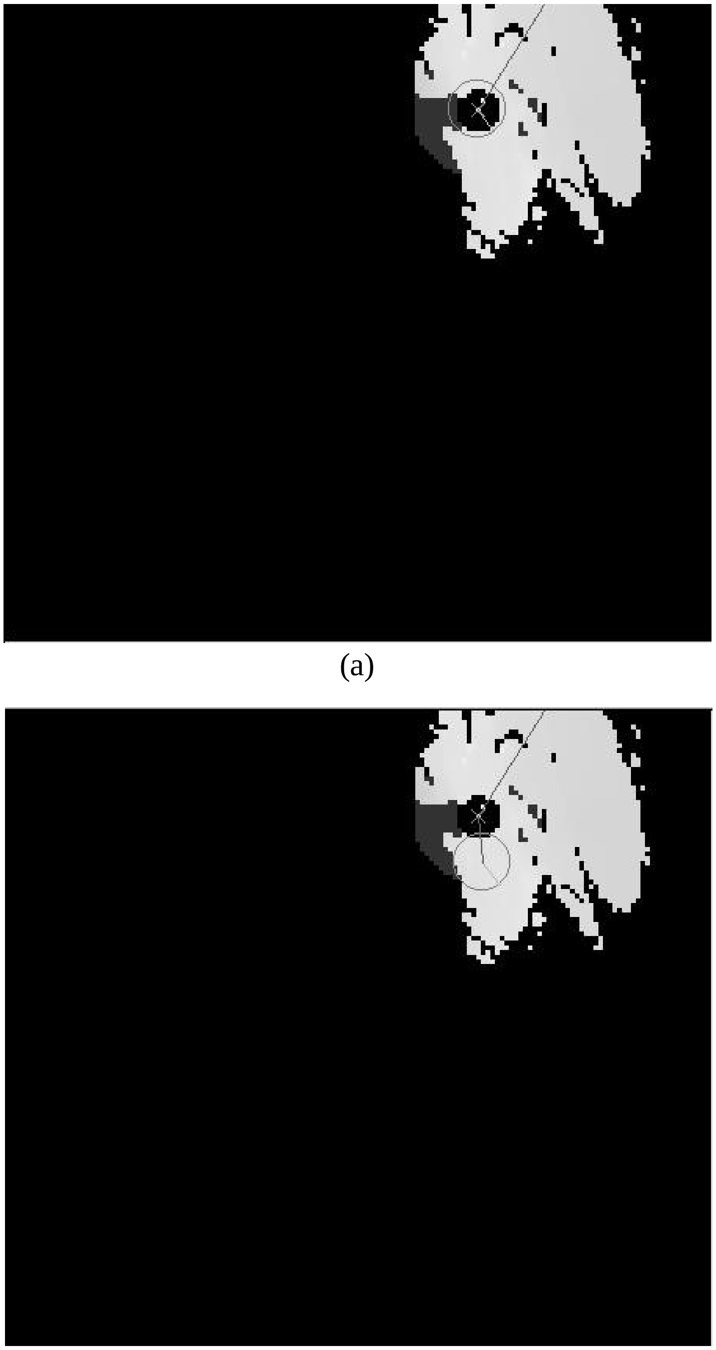
With the data from the new scan in its map, the exploration planner computes the information gains for the frontier, viewing cliff faces and go down information metrics. Next the planner uses the greedy planning strategy to find the path to the next scan point which maximizes the utility as computed by equation (3.2). The position of the next scan point is then drawn on the map and reported to the human experimenter.

With the position of the next scan point ready, the cart is manually moved to a spot close to the desired position. An autonomous robot, especially a large non-holonomic robot like Nomad, is unlikely to arrive exactly at a target waypoint. Therefore, the cart was maneuvered to the general region within 1 or 2 meters of the desired scan point. Once in place, a new scan was commanded and the process repeated.

The Z+F laser can scan ±30 degrees in elevation. Since the scanner is mounted approximately 1.5m above the ground, each scan has a hole of no data around the cart. After the first scan, all possible paths must pass through this unknown region so the planner is unable to choose any paths with positive utility. To overcome this problem, the cart is moved outside the no data hole and another scan is taken. Thus the exploration planning does not start until after the second scan. This process is illustrated in Figure 5-3.

5.3 Results

The results of the cliff exploration are shown in Figures 5-4 through 5-7. In the figures of the exploration planner’s map, higher elevations are whiter with black being unknown terrain. In the information gain maps, a higher information gain is whiter. The +’s on the maps are the desired scan locations and the x’s are the actual scan locations. The line connecting the x’s is a straight line approximation of the path driven by the cart. It closely follows the actual cart route except at the turn onto the downward path where the straight line approximation cuts the corner.



(a)

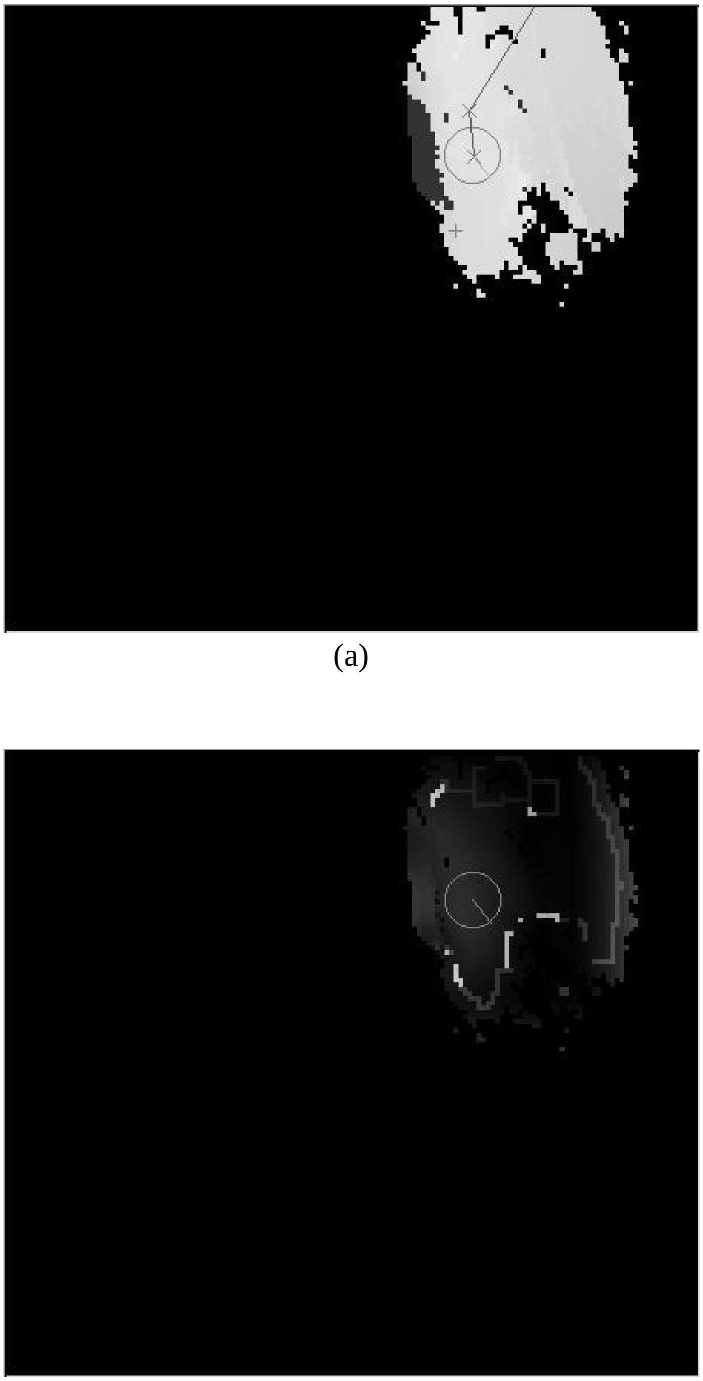
(b)

Figure 5-3: First laser scan. Notice the hole in the center where no data is collected from the sensor. With no data around the robot the planner cannot find any traversable paths. The sensor is manually moved outside the hole (as shown in figure b) and a second scan is taken. The circle in the figures is the robot.

Figure 5-4(a) shows the height attribute of the exploration planner's map after the second laser scan. As described above, two scans were needed before the planning could start. In the map, the cliff is along the left edge of the white region. The cart is on a rise which runs along the cliff edge so the terrain on the right side of the map is a little lower than that in the center. Figure 5-4(a) also shows the results of the first planning step. The planner wishes to travel south, following the edge of the cliff. The computed position of the next scan is indicated with a + sign. Figure 5-4(b) shows the information gains for this same map. Notice the high information gain near the computed destination. The information gain is high here due to high information gains in both the frontier and view cliff faces information metrics. There is also a region of high information gain on the frontier along the cliff edge to the north of the cart. This is another possible direction to search for the path to the bottom.

Figure 5-5(a) shows the height map a few scans later. The cliff is still visible along the left side of the map and is starting to curve to the east. This is the edge of the path to the bottom, part of which can be seen as the isolated white blobs to the bottom of Figure 5-5(a). The planner decides to continue following the cliff until the scan shown in Figure 5-5(b) where the head of the path down to the cliff base becomes fully visible. At this point, the go down information metric adds its weight to the frontier and viewing cliff faces information metrics to attract the cart down this path. The computed scan point in Figure5-5(b) is several meters below the previous scans and starts the cart on the way to the cliff bottom.

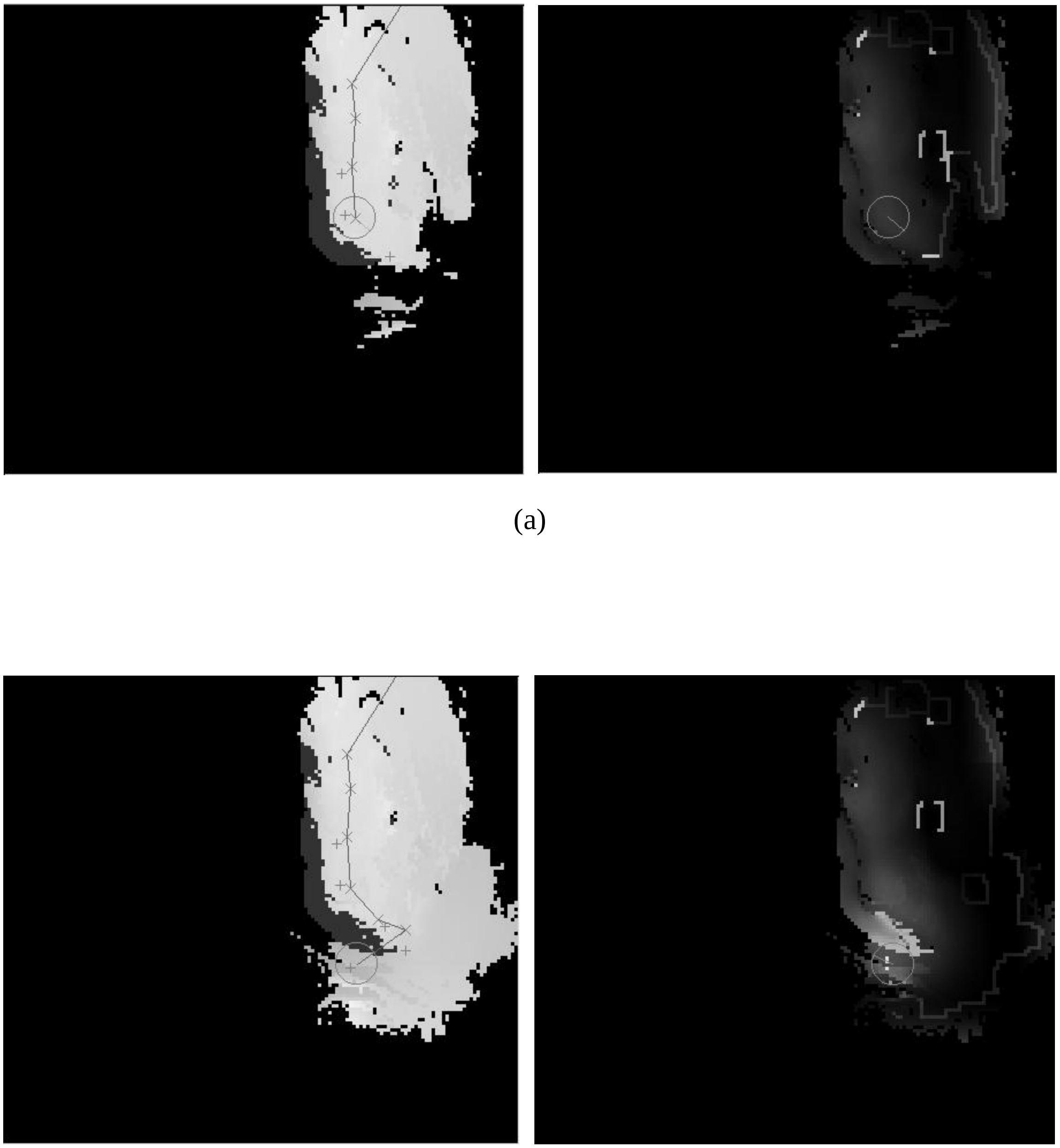
The exploration planner continued to guide the cart down to the cliff base and then along the face as shown in Figure 5-6. This figure shows the complete exploration path and the map at the end of exploration. It shows that the multiple information metric planner is capable of guiding a robot to the face of a cliff. The technique is robust enough to handle sensor and position noise and can even cope with errors in sensing locations. The exploration path is superimposed onto a picture of the cliff in Figure5-7.



(a)

(b)

Figure 5-4: Result of first planning step. (a) Map after the second scan. The + indicates where planner wants to take next scan. (b) Information gains. Notice high information gain at edge of cliff and unknown.



(a)

(b)

Figure 5-5: Finding the path to the cliff base. Figures show the height attribute on the left side and the total information gain on the right side. (a) Sensor is detecting parts of a possible path to the base. (b) A couple of scans later, the head of the path is more fully seen and the planner decides to investigate further.

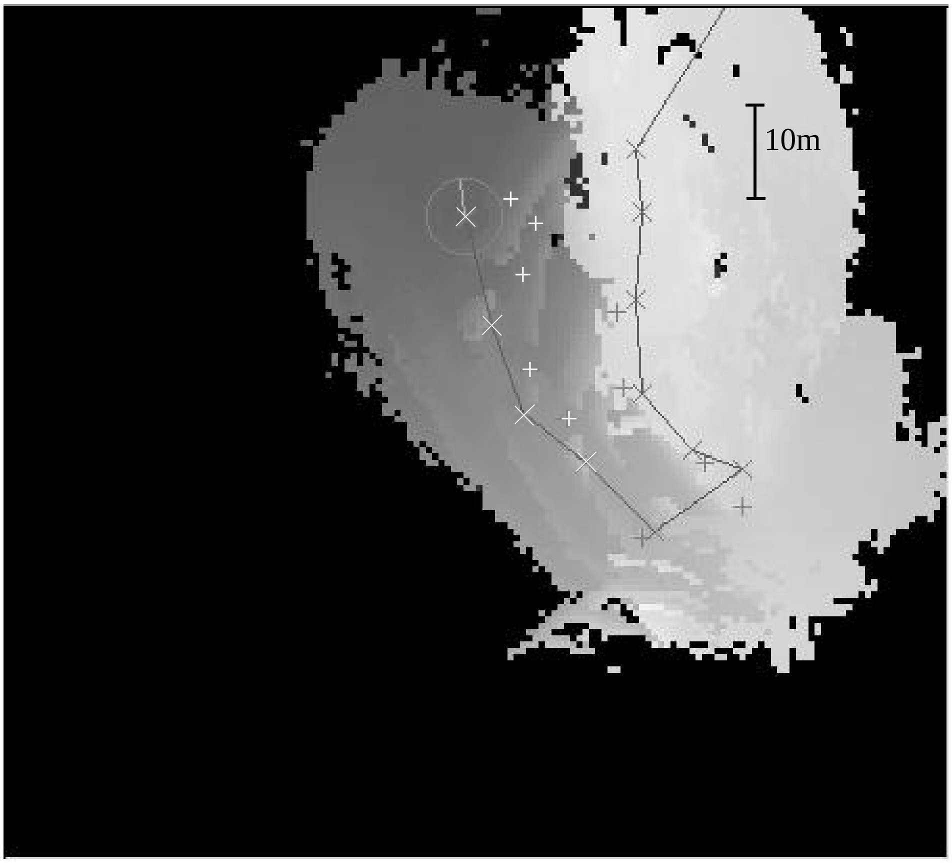


Figure 5-6: Final exploration route. Higher elevations are whiter in color. Black is unknown terrain. The +'s indicate scan position requested by the planner. X's are the actual scan locations. The line is a straight line approximation of the path between scan points. It closely approximates the actual path of the sensor except around the turning point where it cuts off the corner.

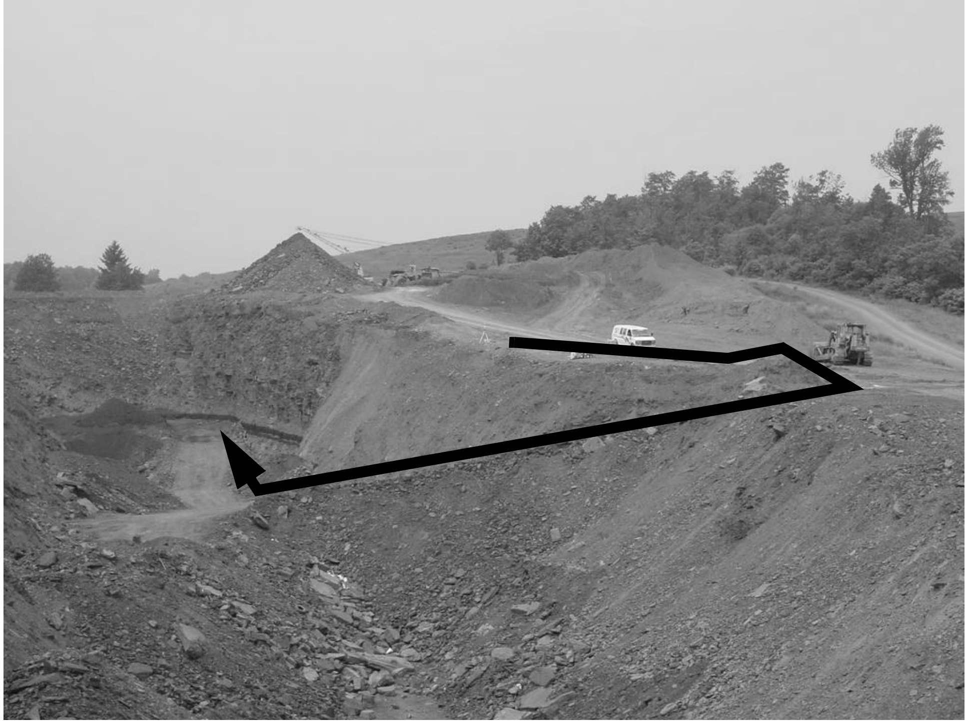


Figure 5-7: Final Exploration Path. The general path taken by the robot is superimposed on the image of the cliff.

5.4 Conclusions

The results demonstrate that the multiple information metrics exploration planner is capable of successfully completing the finding and viewing cliff faces exploration problem in the real world with real sensor and localization noise. Further, since the same exploration planning software was used in both the simulation and field cliff demonstrations, these results lend validity to the other simulation results.

In the field tests the cart was not placed at the exact spot requested by the exploration planner. Instead the cart was moved to a spot close to the planned sensor location and the sensor scan was taken. This experimental procedure was adopted based on experience with actual field robots that combined global planners with local navigation [37]. In these systems, the robot rarely achieved the exact waypoint specified by the global planner due to the competing directions from the local planner as well as the vehicle kinematics which made some motions difficult. The fact that the multiple information metrics exploration planner was able to handle these errors in waypoint tracking and still successfully explore the cliff face is a sign of robustness in the planner and indicates that it might work well in other explorations on real robots.

However, it is possible that these sensor scan positional errors introduced an experimental bias in the results and that if the cart had been placed on the desired scan points the exploration would have failed. This seems unlikely for several reasons. First, the positional errors are on the order of two meters in most cases which for a sensor with a radius of 20 meters means that what is seen from the actual and desired scan points is mostly the same. Further, given the actual and desired positions in the experiment, in no case is what would be seen from the actual scan location significantly different than what is seen from the desired location. Finally, at no point in the exploration does it appear that the desired sensor location is diverting the robot away from the cliff and the actual location has been chosen to pull the robot back into the cliff. In fact it is quite the opposite with the actual scan location in many cases further from the cliff than the desired.

CHAPTER 6 Conclusions

This thesis presents a novel methodology for solving complex exploration tasks that combines multiple information metrics to guide a robot explorer in collecting the most information about the environment. The information metrics used to explore, and ultimately to determine what was important in the environment, depend on the exploration task to be solved. The methodology also explicitly considers the cost of collecting the information and plans paths which maximize the utility, or the information gain minus the collection costs, of the exploring robot.

The thesis details several possible attributes for the robot's map or knowledge database including traversability, reachability and probability of being part of a cliff. Several information metrics are also developed to illustrate how information metrics are defined and are used in later exploration examples. The information metrics defined are: frontier, increase map certainty weighted by traversability, determine reachability, view cliff faces and seek lower elevations.

To demonstrate the feasibility of the multiple information metrics exploration planner and how it can be applied to different exploration problems, two exploration tasks are completed in simulation. The first, Create Traversability Maps, creates maps of an unknown region which tell the robot how safe it is to drive in each area. Special attention is given to making the maps accurate in and around unsafe areas, and the ability of the robot to drive from a given start cell to another cell, the cell's reachability, is also determined in the map.

The Create Traversability Maps problem builds a map with attributes: height, traversability and reachability. It also uses the information metrics: frontier, increase map certainty weighted by traversability and determine reachability which are developed in this thesis. The effects of weighting the three information metrics differently is illustrated and completed traversability maps are presented.

The second exploration task performed in simulation is the Find and View Cliff Faces task. In this problem the robot is asked to find a cliff in the environment and view the face of the cliff. The robot starts at the top of the cliff, where the face is not visible, and has to find its way down to the cliff base and then travel along the cliff face. The Find and View Cliff Faces problem builds a map with attributes: height, traversability and probability of being a cliff which are similar to those used in the create traversability maps problem. The information metrics used are: frontier, view cliff faces and seek lower terrain. This example shows that by using different information metrics and map attributes the multiple information metrics exploration methodology can be applied to different exploration problems.

To demonstrate its ability to handle real world data and noise the multiple information metrics exploration planner is tested in a real environment. The planner is connected to the Z+F laser sensor and asked to perform the Find and View Cliff Faces exploration problem on a real cliff. Using the same exploration planning code used in the simulated cliff exploration and just changing the laser driver and vehicle controller modules, the multiple information metrics exploration planner successfully explores a real cliff.

6.1 Contributions

This thesis has made several contributions to the field of autonomous exploration.

- Presentation of a general methodology for performing autonomous exploration. This methodology allows the exploring robot to consider more than one thing while exploring and explicitly considers the cost of gathering information while exploring. Further, the criteria of exploration are easily changed in the presented framework. This allows the method to be easily adapted and applied to many diverse exploration problems. The methodology was developed for surface exploration of rugged, outdoor terrains and the examples provided performed exploration in this type of environment. However, the multiple information metrics exploration methodology is applicable to a much broader set of environments and exploration tasks than this thesis considered. Provided that the environment can be adequately represented in a grid based map, the costs of exploring can be quantified and the exploration goals can be expressed as the weighted sum of information metrics, the multiple information metrics exploration methodology should be applicable. Indeed, the methodology was successfully applied to an indoor world to perform the create traversability maps exploration problem [38].

- Application of information and decision theories to the problem of exploration. By expressing the information metrics in terms of the expected information gain, a common unit, the bit, is applied to the very different criteria used in the exploration. By expressing all of the costs in units of seconds, the expression for robot utility only contains two different units — bits and seconds. These units are compared with the *value of information* parameter which has the physical interpretation of how many seconds the robot is willing to spend to gain 1 bit of information. Thus the approach used in this thesis has significantly reduced the number of hand tuned constants needed to convert disparate units.

- Implementation of the multiple information metrics exploration planner to multiple exploration tasks. The implementations in this thesis not only demonstrate the validity of the approach but they also solve important exploration problems. The creation of a traversability map of an unknown area is a classic problem in robotics, particularly in field robotics where traversability is not just a binary traversable or not quantity. The introduction of the concept of having greater certainty near poor terrain is an interesting twist which provides maps that are more useful than conventional traversability maps. The second example, find and view cliff faces, has great potential in the search for water and life on Mars.

- Demonstration of the exploration of a real cliff. The find and view cliff faces example was implemented using a real laser sensor on a real cliff. This demonstration in the field proved that the multiple information metrics exploration planner can function on real, noisy data.

6.2 Future Work

This thesis developed a novel method for exploring rugged, outdoor surfaces and applied it to two important exploration problems in simulation. The method was also validated in a field experiment which explored a cliff finding and recording the face of the cliff. However, there are still issues to be resolved and further research to be done to expand the multiple information metrics exploration planner.

6.2.1 Considering Robot Heading and Limited Field of View Sensors

The current implementation of the methodology only considers sensors with 360° fields of view. This limitation was imposed to remove the need, when calculating the expected information gains and planning exploration paths, to consider robot heading and thus significantly reduce the computation required. However, not all robot sensors have a 360° field of view and the general methodology presented in chapter 3 does not require this restriction.

The easiest way to extend the current implementation to handle limited field of view sensors is to increase the dimensions of the robot's map to include heading. Currently a 2D grid map is used to map the robot's x and y position to a unique cell. By using a 3D map the robot's x, y and heading can be mapped to a unique cell. Given the sensor field of view, the expected information gains for each possible heading at an x, y location can be calculated. Then the path planner would generate a path that not only specified the xy locations but also the headings in those locations to maximize utility to the robot.

Extending the dimension of the map greatly increases the amount of memory required. However, each map cell has two vectors, the attribute vector and the expected information gain vector. In general, the attributes vector contains items such as height, reachability and probability of being a cliff, which are independent of robot heading. Therefore instead of simply adding an entire new dimension to the map it is possible to only extend the expected information gain vector in this new dimension. This will reduce the memory required without loosing any functionality.

Another possibility for handling limited field of view sensors was proposed by Younes [77]. In this method the expected information gains and the exploration path are computed assuming a 360° field of view sensor. Next, in each cell requiring a sensor reading, the expected information gains are recomputed for all possible headings. The heading with the largest expected information gain is the one used for the sensor reading. This approach greatly reduces the amount of computation and memory required but the cost is a loss in plan quality. Some cells with a low expected information gain for a 360° field of view may actually have very high expected information gains for particular headings.

6.2.2 Considering Multiple Sensors

Most robot explorers have more than one sensor. For example, in Antarctica, Nomad had a laser scanner, high resolution color camera and a spectrometer [69]. Each sensor collects different types of information, has a different field of view and has a different cost to use it. An exploring robot must therefore choose which sensor to use and when to use it when exploring a region.

The multiple information metrics exploration planner can be easily extended to consider multiple sensors. Instead of one expected information gain vector, the map now needs to have one expected information gain per sensor. Thus each sensor has its own set of information metrics which are calculated based in the field of view, range and other characteristics of that sensor. At each cell the planner will then know how much information it can gain from taking a sensor reading with a particular sensor. The different sensor deployment costs can also be taken into account in the cost part of the utility equation.

One difficulty in this approach could be comparing the value of information from each sensor. Ideally, the information metrics will be designed in such a way that one bit of information from sensor A is equivalent to one bit of information from sensor B. However, in the real world, this may not always be possible. In this case, each sensor can have its own value of information parameter, converting its expected information gains into units of cost.

6.2.3 Representing Information Gain of Continuous Valued Variables

In implementing the methodology outlined in this thesis it is necessary to compute the expected information gains for the various information metrics used in an exploration problem. From information theory the information gain can be found by looking at the change in entropy (see equation (2.1)).

For information metrics which depend on binary or discrete valued variables computing the entropy is quite simple. In fact this was used for the information metric determine reachability which used the binary valued variable reachability. For continuously valued variables, such as height, the computation of entropy is more difficult. Equation (2.2) shows that to calculate the entropy of a continuous value variable the probability density function must be known over the entire range of the variable. Further, for some probability density functions the entropy could be negative or infinite causing great difficulties in the interpretation of continuous variable entropy.

In this thesis the probability density functions of continuous variables are not computed and thus the information metrics based on these continuous variables, such as frontier, are not true information gains from an information theoretic point of view. Instead of the probability density function a value called certainty is used to indicate how accurate or reliable the value assigned to that variable is. For example a cell might have a height of

20m with a certainty of 70%. This certainty is not a probability but is a confidence measure and is being used much like an entropy score in this thesis (in fact it is being used as one minus the entropy for information calculations). Using this confidence as an entropy is an abuse of notation but it does work fairly well in practice (see the examples in chapters 4 and 5 as well as [37], [38] and [39]).

Since the real formula for entropy of a continuous variable is known, why not use this when computing the expected information gains like the frontier information metric? One reason is that continuous entropies may be negative or infinite. Fortunately these degenerate cases usually only occur when some value of the probability density function is greater than one [49] and with physical variables such as height this is unlikely to happen if the probability density function computed reflects reality. This brings up another reason continuous entropies are difficult to use in practice — computing the true probability density function, given the sensor measurements, is in general difficult.

Consider the laser scanner sensor used in this thesis and the continuous variable height. Before the first laser scan is taken, the prior probability density function must be chosen for the height variables in each map cell. This might be a flat line, indicating equal probabilities for every possible height, or it might be generated from an *a priori* model of the environment. With each laser scan, two types of information can be gained to alter the probability density functions in each map cell. First, lasers will pass through some cells without hitting anything. Given the height of the laser passing through the cell the probability that the cell height is less than the laser height increases. The second case is if a laser hits something in a cell. In this case the probability that the height of the cell is greater than the laser height increases. If some of the lasers in a scan hit a cell and some pass through the same cell, a fairly accurate picture of the true cell height emerges. The effects of a laser scan on the height probability density function for a particular cell are shown in Figure 6-1.

The probability density functions in Figur e6-1 are just sketches, exactly what slope to use in figures (b) and (c) around the measurement points is not entirely obvious. The images

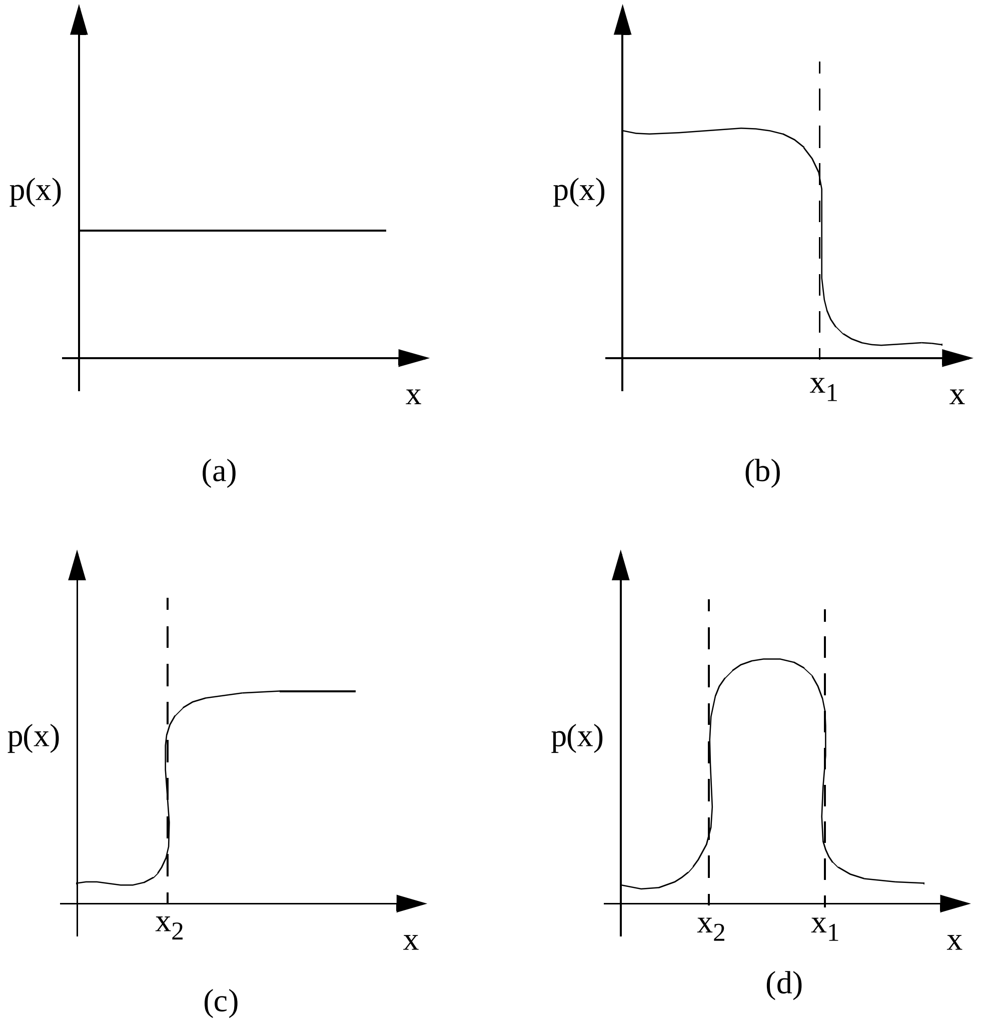


Figure 6-1: Updating probability density function. (a) *A priori* probability density function for height variable in cell m. (b) Updated probability density function based on a laser reading at height x_1 passing through cell m without hitting anything. This indicates that the height of cell m is less than x_1. (c) Updated probability density function (from *a priori* density) based on a laser reading at height x_2 hitting something in the same cell, m. This indicates that the height of cell m is greater than x_2. (d) Combining information from probability density functions (b) and (c) for cell m. Based on these two laser readings in the cell, the probability of the height being between x_2 and x_1 is much greater than for other values.

also don’t account for noise, the possibility of overhanging terrain (such as caves) or the possibility that particulate matter, such as dust or snow, in the air will generate a return signal [37]. However, the figure does illustrate how one could generate a probability density function for height and then update it using sensor information.

The problem with using a probability density function in practice is in storing and representing it. It is unlikely that a closed form representation for the functions illustrated in Figure 6-1 can be found and even if it could it is unlikely that the representation could be maintained and updated as more data is added with each scan. One alternative is to discretize the distribution. This is equivalent to discretizing the variable and using the discrete entropy equation. This does reduce accuracy but more importantly it greatly increases the storage space required because this discretized distribution must be recorded in all map cells.

While using the complete probability density function for continuos valued variables would make the methodology presented in this thesis more theoretically sound from an information theory point of view, it is unclear if this would significantly improve the performance of the exploration path planning. Uncertainties in determining probability priors and loss of accuracy in discretizing the density function mean that at best an approximation of the true probability density function is used. Combined with the cost of storing a density function in every map cell, for every continuous variable, may make the use of complete probability density functions impractical.

6.2.4 Planning Beyond the First Sensor Reading

This thesis implemented a greedy search technique to plan exploration paths up to the next sensor reading. Using the greedy planner was justified because one property of the exploration was to collect information as quickly as possible. However, the greedy planner’s short term planning horizon does produce paths which are less efficient over the entire exploration than might be possible with a planner that planned sequences of sensing actions.

The major impediment to developing a planner which plans multiple sensor readings is that the expected information gains in neighboring areas are dependent on each other. Plans are path dependent — planning the next step not only depends on where you are but how you got there. The reason the information gains are dependent is that they are computed from the current map. Once a sensor reading is taken, the map changes and so do the information gains. Therefore not only does a long range planner need to plan path dependent paths, which is a very large search space, it must also be able to reasonably predict what each planned sensor reading will sense. This need to predict what will be seen, so that the new information gains can be calculated, makes planning paths past the first sensor reading almost impossible.

One way to plan past the first sensor reading is to group the information gains into large regions and then plan a route through these regions. By assuming that the information gain in each region is independent from those in other regions an optimal path through the regions can be found by solving the prize collecting travelling salesman problem — each region is a city and the aggregate information gain in the region is the prize. The assumption of regional independence holds provided that a sensor reading in one region cannot see any cell changed by a sensor reading in another region. This assumption should hold in the centers of large regions but becomes less valid as one gets the edges of neighboring regions. While this is still an NP-complete planning problem, if the number of regions is kept small, the problem may be tractable.

What to do inside a region and when to continue on to the next region in this plan are two questions that need to be answered. Since the problem of information gain dependence occurs inside the region it makes sense to use the greedy planner once inside a region. One way to determine when to continue on to the next region is to use a threshold on the amount of information collected, once above that move on to the next region.

A better way to decide when to move on is to look at the rate of information gain. As the robot explores an area it will generally gain a lot of information in the early part since it knows little of the area. As the robot learns more about the region there is less to learn and so the rate of information gain decreases. Once the rate drops below a certain threshold the robot continues on to the next region of interest. This concept can be formalized using an area of biology called Foraging Theory [62]. Foraging theory has developed models which describe how animals forage for food amongst multiple patches of food. Most importantly, a model has been derived which, given a set of food patches (imagine berry patches), the expected gain from each patch and the distance between patches, shows how much time to spend in each patch collecting food. This theory has been applied not only in biology to predict animal movements but also to web browsers to predict document viewing by humans [47]. Foraging theory could be combined with the region planner to generate a plan that specifies which regions to visit (like the patches of food) and how long to spend in each patch. This explicitly takes into account the reduction in rate of information in a region and gives a rule for passing on to the next region in the plan.

The concept of a regional plan sidesteps the issue of information gain dependence and alleviates the need to predict the results of a sensor scan but still allows the robot to make some long term plans that guide its exploration. This type of planner might work well in cases where the robot has some *a priori* information, perhaps from satellite imagery, about the environment to explore. If the robot starts with a blank map, as done in this thesis, then it tends to have one region of well known data surrounded by unknown areas. In this case the robot usually has to make decisions about whether to examine the edge of knowledge close to its position or to travel to a further edge.

Another approach to planning beyond the first sensor reading is to predict what a sensor scan will see. In this case, every time the planner called for a scan, it would predict what the scan sees to create an estimate map, $\boldsymbol{M'}(t)$ and then continue to plan based on this estimated map. The estimate could be created using a statistical model of the environment being explored. This still requires planning paths which are path dependent so the planning search space is large and it remains an open question as to how well the planner can predict sensor scans. Further, as one plans more and more scans the quality of the map estimate will degrade.

References

[1] Acar, E.U., H. Choset, “Critical Point Sensing in Unknown Environments”, *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA., 2000.

[2] Arkin, R., “Motor Schema-Based Mobile Robot Navigation”, *International Journal of Robotics Research*, vol 8(4), August 1989, pp. 92-112.

[3] Argamon-Engelson, S., S. Kraus, S. Sina, “Utility-based On-line Exploration for Repeated Navigation in an Embedded Graph”, *Artificial Intelligence*, vol. 101, no. 1-2, pp. 267-84, May 1998.

[4] Awerbuch, B., Y. Azar, A. Blum, S. Vempala, “New Approximation Guarantees for Minimum-Weight k-Trees and Prize-Collecting Salesman”, *SIAM Journal of Computing*, 28(1), 1999.

[5] Baeza-Yates, R.A., J.C. Culberson, G.J.E. Rawlins, “Searching in the Plane”, *Information and Computation*, no. 106, pp. 234-252, 1993.

[6] Bresina, J.L., R. Washington, “Expected Utility Distributions for Flexible, Contingent Execution”, *Proceedings of the AAAI-2000 Workshop: Representation Issues for Real-World Planning Systems*, Austin, TX., 2000.

[7] Brooks, R., “A Robust Layered Control System for a Mobile Robot”, *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, April 1986, pp. 14-23.

[8] Brumitt, B.L., A. Stentz, “GRAMMPS: A Generalized Mission Planner for Multiple Mobile Robots in Unstructured Environments”, *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.

[9] Cao, Z.L., Y. Huang, E.L. Hall, “Region Filling Operations with Random Obstacle Avoidance for Mobile Robots”, *Journal of Robotic Systems*, 5 (2), pp. 87-102, 1988.

[10] Choset, H., P. Pignon, “Coverage Path Planning: The Boustrophedon Cellular Decomposition”, *International Conference on Field and Service Robotics*, Canberra, Australia, 1997.

[11] Connel, J., S. Mahadevan, “Rapid Task Learning for Real Robots”, *Robot Learning*, J. Connel and S. Mahadevan, Eds., Kluwer Academic Publishers, Norwell, MA., pp. 105-139, 1993.

[12] Deans, M., S. Moorehead, B. Shamah, K. Shillcutt, W. Whittaker, “A Concept for Robotic Lunar South Pole Exploration”, *Space ‘98*, 1998.

[13] Devy, M., R. Chatilla, P. Fillatreau, S. Lacroix, F. Nashashibi, “On Autonomous Navigation in a Natural Environment”, *Robotics and Autonomous Systems*, vol. 16, pp. 5-16, 1995.

[14] Duckett, T., U. Nehmzow, “Exploration of Unknown Environments Using a Compass, Topological Map and Neural Network”, *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1999.

[15] Elfes, A., “Robot Navigation: Integrating Perception, Environmental Constraints and Task Execution Within a Probabilistic Framework:, *Reasoning with Uncertainty in Robotics*, L. Dorst, M. van Lambalgen and F. Voorbraak (eds.), Amsterdam, 1995.

[16] Elfes, A., “Dynamic Control of Robot Perception Using Stochastic Models”, *Information Processing in Mobile Robots,* G. Schmidt (ed.), Springer Verlag, 1991.

[17] Estlin, T., A. Gray, T. Mann, G. Rabideau, R. Castano, S. Chien, E. Mjolsness, “An Integrated System for Multi-Rover Scientific Exploration”, *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, 1999.

[18] Estlin, T., R. Volpe, I. Nesnas, D. Mutz, F. Fisher, B. Engelhardt, S. Chien, “Decision-Making in a Robotic Architecture for Autonomy”, *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Montreal, Canada, 2001.

[19] Foley, J., A. vanDam, S. Feiner, J. Hughes, *Computer Graphics Principles and Practice*, Addison-Wesley, 1990.

[20] Gage, D.W., “Randomized Search Strategies with Imperfect Sensors”, *Proceedings of SPIE Mobile Robots VIII*, vol. 2058, pp. 270-279, Sept. 1993.

[21] Gonzalez-Banos, H., J.C. Latombe, “Planning Robot Motions for Range-Image Acquisition and Automatic 3D Model Construction”, *AAAI Fall Symposium Series 1998 Integrated Planning for Autonomous Agent Architectures*, 1998.

[22] Hebert, H., A. Stentz, and C. Thorpe, “Mobility Planning for Autonomous Navigation of Multiple Robots in Unstructured Environments”, *Proceedings of ISIC/CIRA/ISAS Joint Conference on the Science and Technology of Intelligent Systems*, September, 1998, pp. 652 - 657.

[23] Hert, S., S. Tiwari, V. Lumelsky, “A Terrain-Covering Algorithm for an AUV”, *Autonomous Robots*, vol. 3, pp. 91-119, 1996.

[24] Hyati, S., R. Arvidson, “Long Range Science Rover (Rocky 7) Mojave Desert Field Tests”, *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Noordwijk, Netherlands, 1999.

[25] Kamon, I., E. Rivlin, E. Rimon, “A New Range-Sensor-Based Globally Convergent Navigation Algorithm for Mobile Robots”, *Technical Report, CIS-Center of Intelligent Systems 9517*, Computer Science Department, Technion, Israel, 1995.

[26] Kelly, A., *An Intelligent Predictive Control Approach to the High-Speed Cross-Country Autonomous Navigation Problem*, CMU-RI-TR-95-33, Ph.D. thesis, Carnegie Mellon University, July 2000.

[27] Koenig, S., C. Tovey, W. Halliburton, “Greedy Mapping of Terrain”, *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, South Korea, May 2001.

[28] Kuipers, B., Y-T. Byun, “A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations”, *Journal of Robotics and Autonomous Systems*, 8:47-63, 1991.

[29] Langer, D., M. Mettenleiter, F. Hartl, C. Frohlich, “Imaging Ladar for 3-D Surveying and CAD Modeling of Real World Environments”, *International Journal of Robotics Research*, vol. 19, no. 11, pp. 1075-1088, November 2000.

[30] Latombe, J.C., *Robot Motion Planning*, Kluwer Academic Publishers, 1991.

[31] Laubach, S.L., J. Burdick, L. Matthies, “An Autonomous Path Planner Implemented on the Rocky 7 Prototype Microrover”, *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.

[32] Laubach, S.L., C.F. Olsen, J.W. Burdick, S. Hayati, “Long Range Navigation for Mars Rovers Using Sensor-Based Path Planning and Visual Localisation”, *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Noordwijk, Netherlands, 1999.

[33] Malin, M.C., K.S. Edgett, “Evidence for Recent Groundwater Seepage and Surface Runoff on Mars”, *Science*, June 2000.

[34] Mataric, M., “Integration of Representation into Goal-driven Behavior-based Robots”, *IEEE Transactions on Robotics and Automation*, 8(3):304-312, June 1992.

[35] Maver, J., R. Bajcsy, “Occlusions as Guide for Planning the Next View”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 5, May 1993.

[36] Mishkin, A.H., J.C. Morrison, T.T. Nguyen, H.W. Stone, B.K. Cooper, B.H. Wilcox, “Experiences with Operations and Autonomy of the Mars Pathfinder Microrover”, *Proceedings of the IEEE Aerospace Conference*, 1998.

[37] Moorehead, S.J., R. Simmons, D. Apostolopoulos, W. Whittaker, “Autonomous Navigation Field Results of a Planetary Analog Robot in Antarctica”, *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Noordwijk, Netherlands, 1999.

[38] Moorehead, S.J., R. Simmons, W.L. Whittaker, “Autonomous Exploration Using Multiple Sources of Information”, *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, South Korea, May 2001.

[39] Moorehead, S.J., R. Simmons, W.L. Whittaker, “A Multiple Information Source Planner for Autonomous Planetary Exploration”, *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Montreal, Canada, 2001.

[40] Nagatani, K., H. Choset, “Toward Robust Sensor Based Exploration by Constructing Reduced Generalized Voronoi Graph”, *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol 3, pp. 1687-1692, 1999.

[41] Nozette, S., C.L. Lichtenberg, P. Spudis, R. Bonner, W. Ort, E. Malaret, M. Robinson, E.M. Shoemaker, "The Clementine Bistatic Radar Experiment", *Science*, vol. 274, pp. 1495-1498, 1996.

[42] Okamura, A.M., M.R. Cutkosky, "Feature-Guided Exploration with a Robotic Finger", *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, South Korea, May 2001.

[43] O'Rourke, J., *Art Gallery Theorems and Algorithms*, Oxford University Press, New York, 1987.

[44] Pedersen, L., "Autonomous Characterization of Unknown Environments", *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, South Korea, 2001.

[45] Pedersen, L., M. Wagner, D. Apostolopoulos, W.L. Whittaker, "Autonomous Robotic Meteorite Identification in Antarctica", *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, South Korea, 2001.

[46] Peters, S.F., "A Science-Based Executive for Autonomous Planetary Vehicles", *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Montreal, Canada, 2001.

[47] Pirolli, P., S.K. Card, "Information Foraging Models of Browsers for Very Large Document Spaces", *Proceedings of the Working Conference on Advanced Visual Interfaces*, L'Aquila, Italy, 1998.

[48] Raiffa, H., *Decision Analysis*, McGraw-Hill, New York, 1997.

[49] Reza, F.M., *An Introduction to Information Theory,* Dover Publications, 1994.

[50] Rosenblatt, J.K., "The Distributed Architecture for Mobile Navigation", *Journal of Experimental and Theoretical Artificial Intelligence,* vol. 9, no. 2/3, April-Sept 1997, pp. 339-360.

[51] Rosenblatt, J.K., "Utility Fusion: Map-Based Planning in a Behavior-Based System", *Field and Service Robotics*, Springer-Verlag, 1998.

[52] Roy, N., W. Burgard, D. Fox, S. Thrun, "Coastal Navigation - Mobile Robot Navigation with Uncertainty in Dynamic Environments", *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, MI., 1999.

[53] Seraji, H., “Fuzzy Traversability Index: A New Concept for Terrain-Based Navigation”, *Journal of Robotic Systems*, 17(2), pp. 75-91, 2000.

[54] Shannon, C.E., “A Mathematical Theory of Communication”, *Bell System Technical Journal*, vol 27, pp. 379-423, 623-656, 1948.

[55] Shillcutt, K., D. Apostolopoulos, W. Whittaker, “Patterned Search Planning and Testing for the Robotic Antarctic Meteorite Search”, *Meeting on Robotics and Remote Systems for the Nuclear Industry*, Pittsburgh, PA., 1999.

[56] Simmons, R., “Structured Control for Autonomous Robots”, *IEEE Transactions on Robotics and Automation*, 10:1, pp. 34-43, February 1994.

[57] Simmons, R., E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, L. Katragadda, S. Koenig, G. Krishnaswamy, Y. Shinoda, W.L. Whittaker, and P. Klarer, “Experience with Rover Navigation for Lunar-Like Terrains”, *Proceedings of the 1995 Conference on Intelligent Robots and Systems (IROS '95)*, 1995, pp. 441 - 446.

[58] Simmons, R., D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, H. Younes, “Coordination for Multi-Robot Exploration and Mapping”, *Proceedings National Conference on Artificial Intelligence*, Austin TX., August 2000.

[59] Singh, S., R. Simmons, M.F. Smith, III, A. Stentz, V. Verma, A. Yahja, K. Schwehr, “Recent Progress in Local and Global Traversability for Planetary Rovers”, *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA., April, 2000.

[60] Stentz, A., “Optimal and Efficient Path Planning for Unknown and Dynamic Environments”, *Carnegie Mellon Robotics Institute Technical Report CMU-RI-TR-93-20*, 1993.

[61] Stentz, A., “The D* Algorithm for Real-Time Planning of Optimal Traverses”, *Carnegie Mellon Robotics Institute Technical Report CMU-RI-TR-93-20*, 1993.

[62] Stephens, D.W., J.R. Krebs, *Foraging Theory*, Princeton University Press, Princeton, N.J., 1986.

[63] Swain, M.J., M.A. Stricker (editors), “Promising Directions in Active Vision”, *International Journal of Computer Vision*, vol. 11, no. 2, pp. 109-126, 1993.

[64] Thrun, S., A. Bucken, “Integrating Grid-based and Topological Maps for Mobile Robot Navigation”, *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pp. 944-950, Portland, OR., August 1996.

[65] Thrun, S., D. Fox, W. Burgard, “Probabilistic Mapping of an Environment by a Mobile Robot”, *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998.

[66] Thrun, S., W. Burgard, D. Fox, “A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping”, *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA., 2000.

[67] Volpe, R., I. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das, “The CLARAty Architecture for Robotic Autonomy”, *Proceedings of the 2001 IEEE Aerospace Conference*, Big Sky, Montana, March 10-17, 2001.

[68] Volpe, R., “Navigation Results from Desert Field Tests of the Rocky 7 Mars Rover Prototype”, *International Journal of Robotics Research, Special Issue on Field and Service Robots*, 18(7), July 1999.

[69] Wagner, M.D., D. Apostolopoulos, K. Shillcutt, B. Shamah, R. Simmons, and W.L. Whittaker, “The Science Autonomy System of the Nomad Robot”, *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, May, 2001.

[70] Washington, R., K. Golden, J. Bresina, “Plan Execution, Monitoring and Adaptation for Planetary Rovers”, *Proceedings of the IJCAI-99 Workshop, Scheduling & Planning meet Real-time Monitoring in a Dynamic & Uncertain World*, Stockholm, Sweden, 1999.

[71] Wettergreen, D., D. Bapna, M. Maimone, H. Thomas, “Developing Nomad for Robotic Exploration of the Atacama Desert”, *Robotics and Autonomous Systems*, Vol. 26, No. 2-3, February, 1999, pp. 127-148.

[72] Whaite, P., F.P. Ferrie, “Autonomous Exploration: Driven by Uncertainty”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, March 1997.

[73] Whittaker, W., S.J. Moorehead, “Field Robots”, *International Conference on Engineering and Technical Sciences*, 2000.

[74] Yahja, A., A. Stentz, S. Singh, B. Brumitt, “Framed-Quadtree Path Planning for Mobile Robots Operating in Sparse Environments”, *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.

[75] Yamauchi, B., “A Frontier-Based Approach for Autonomous Exploration”, *Proceedings of CIRA ‘97*, Monterrey, CA., July 1997.

[76] Yamauchi, B., A. Schultz andW. Adams, “Mobile Robot Exploration and Map-Building with Continuous Localization”, *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.

[77] Younes, H., Carnegie Mellon University, personal communications, 2001.

[78] Zelinsky, A., R.A. Jarvis, J.C. Byrne, S. Yuta, “Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot”, *ICAR ‘93*, pp. 533-538, 1993.

APPENDIX A

Additional Simulation Results

This appendix contains a record of all the simulations run for the Create Traversability Map exploration initially described in section 4.2. As mentioned there, the Create Traversability Map exploration was run with four different information metric weights which are summarized in Table A-1. For each information metric weighting the exploration was performed from five different starting points which are indicated in Figure A-1. Each information metric weighting was given a different run number (as indicated in Tab leA-1) and each starting location was given a unique letter as shown in Figure A-1. In this way each simulation run can be uniquely identified with a run number and letter. For example, the run with equal weights to the three information metrics and starting in the upper right hand corner of the map is called Run 1c.

Table A-1: Information Metric Weights for Three Runs

Information Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**
Frontier	0.33	1.0	0.0	0.85
Increase Map Certainty weighted by Traversability	0.33	0.0	1.0	0.15
Reachability	0.33	0.0	0.0	0.00

The quantitative data presented in the next sections include the map information at the end of the explorations for the categories of height information, height information weighted by traversability and reachability and were computed using equations (4.1) through (4.3). Further the total path lengths of the explorations and the number of sensor scans taken are

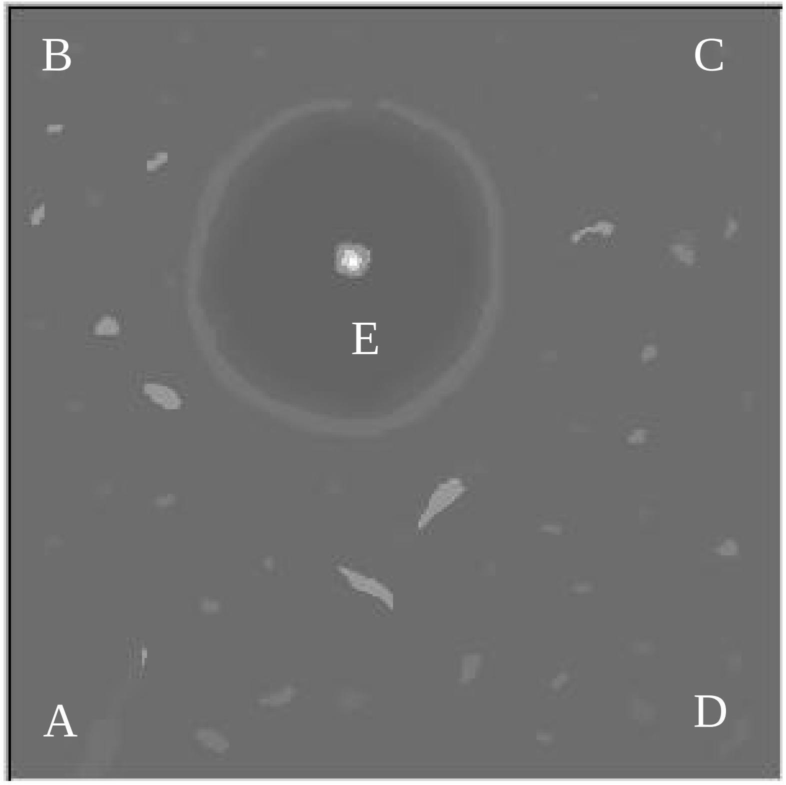


Figure A-1: Starting positions of Create Traversability Map exploration. Map shows the crater world environment and the letters indicate the five starting positions used for the simulation runs.

presented. The normalized path length is the total exploration path length normalized by the length of the environment, in this case 300 m, as is presented to give an idea of how many times the size of the environment the robot had to drive in order to explore the region. Finally the average distance between sensor scans is presented to indicated how densely the robot sampled the environment.

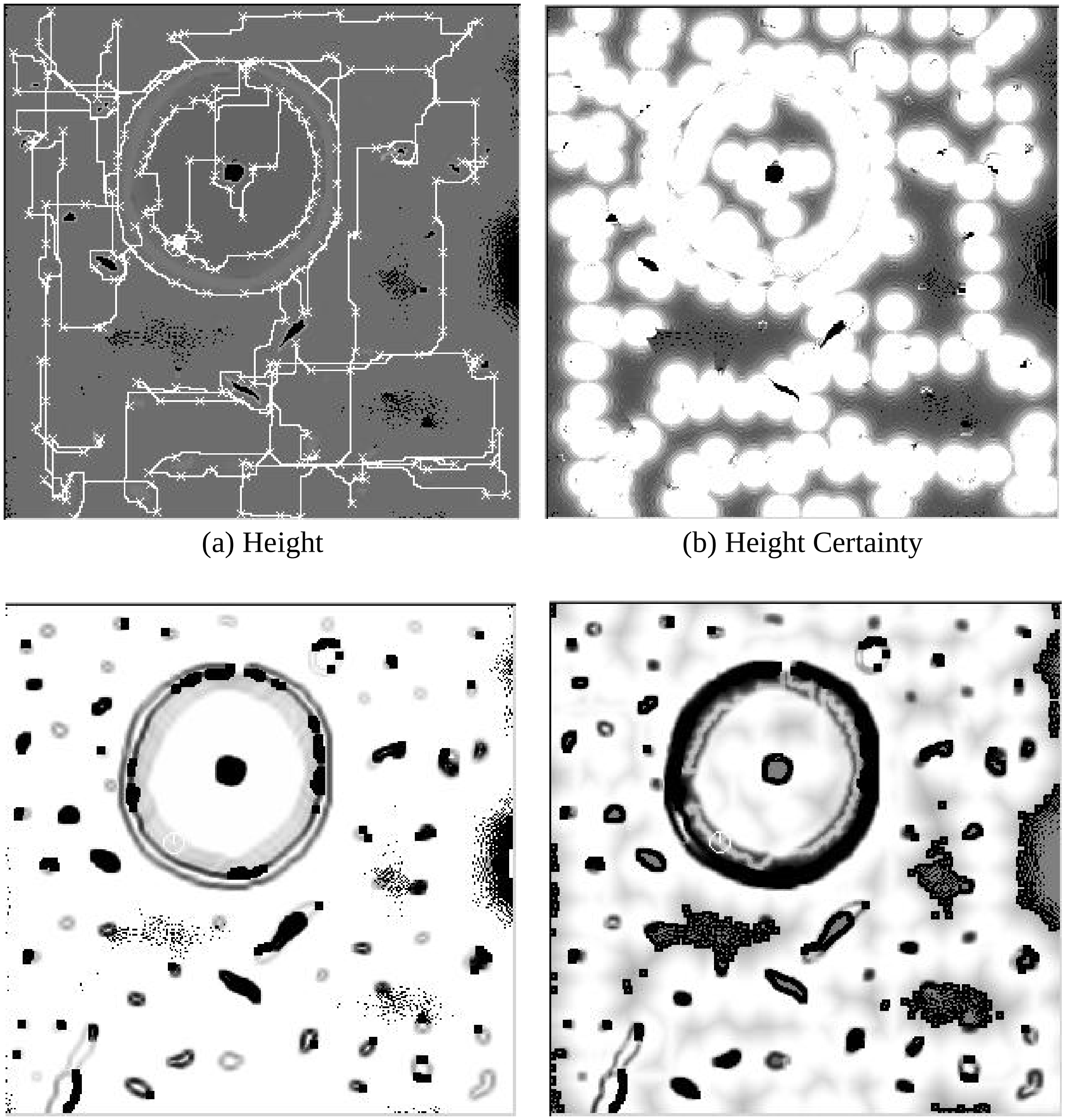
A.1 Run 1

Table A-2: Run 1 Create Traversability Map Exploration Gains

Final Map Information	**A (kbits)**	**B (kbits)**	**C (kbits)**	**D (kbits)**	**E (kbits)**
Height Information	71.1	75.4	74.2	79.8	71.5
Height Info weighted by Traversability	10.5	11.4	11.1	11.3	11.4
Reachability Information	64.8	67.4	67.5	71.4	65.4

Table A-3: Run 1 Create Traversability Map Exploration Costs

Exploration Costs	**A**	**B**	**C**	**D**	**E**
Path Length (meters)	3408	4199	3984	4878	3559
Normalized Path Length (path length / site side length)	11.4	14.0	13.3	16.3	11.9
Number of sensor readings	233	254	243	314	285
Avg. Distance between Sensor Readings (m/scan)	14.6	16.5	16.4	15.5	12.5



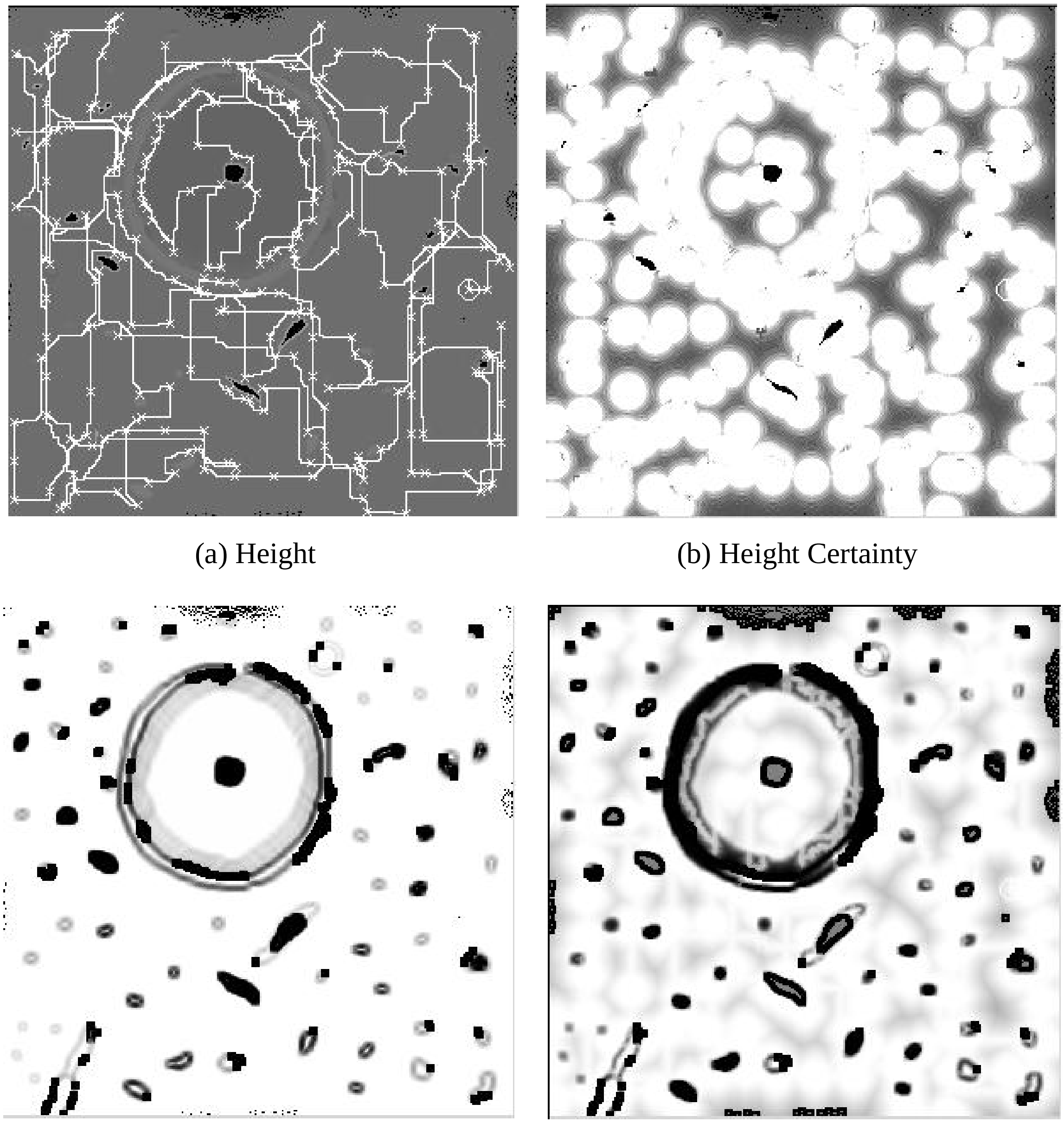
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-2:Run 1a Path Results.



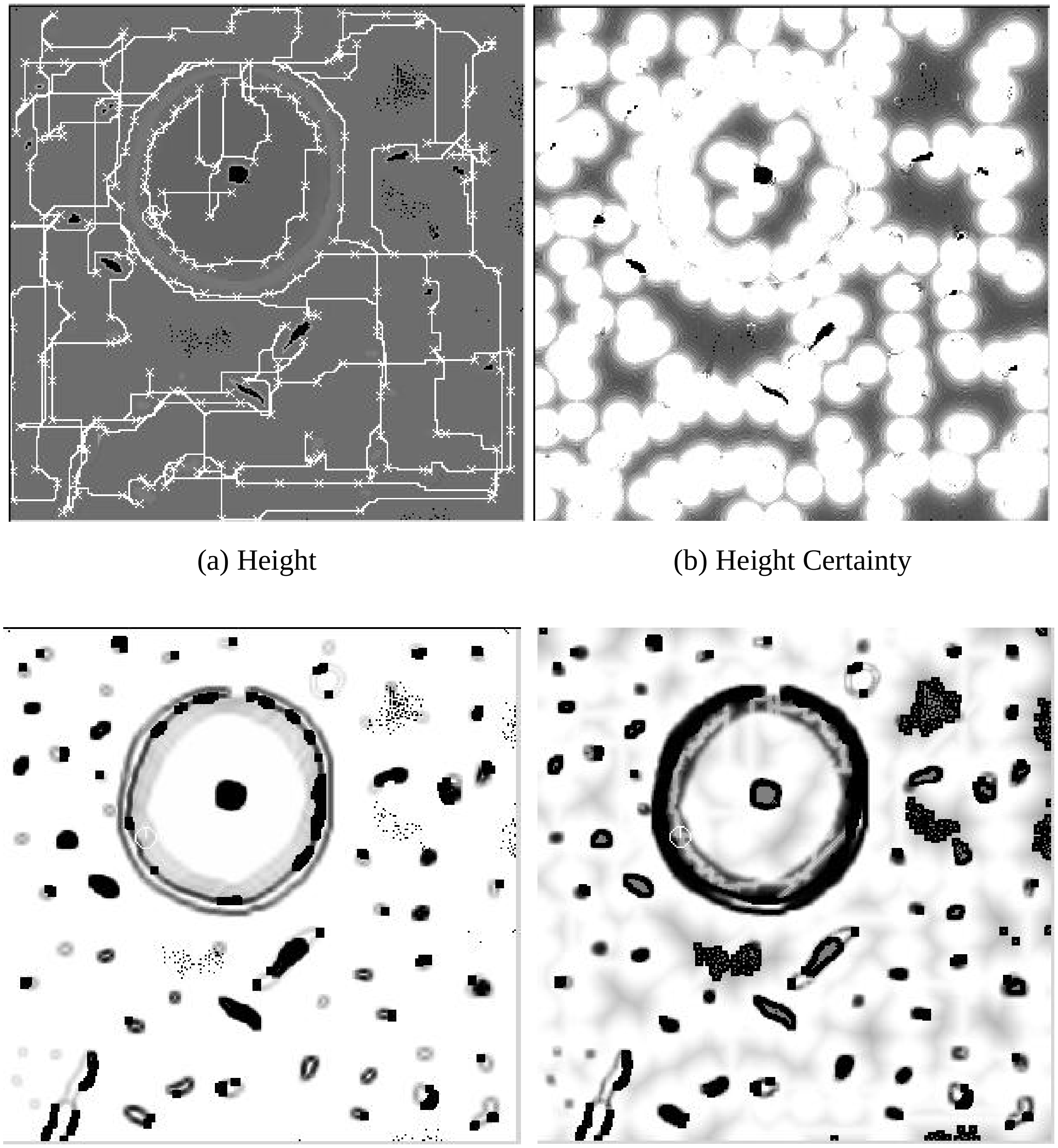
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-3:Run 1b Path Results.



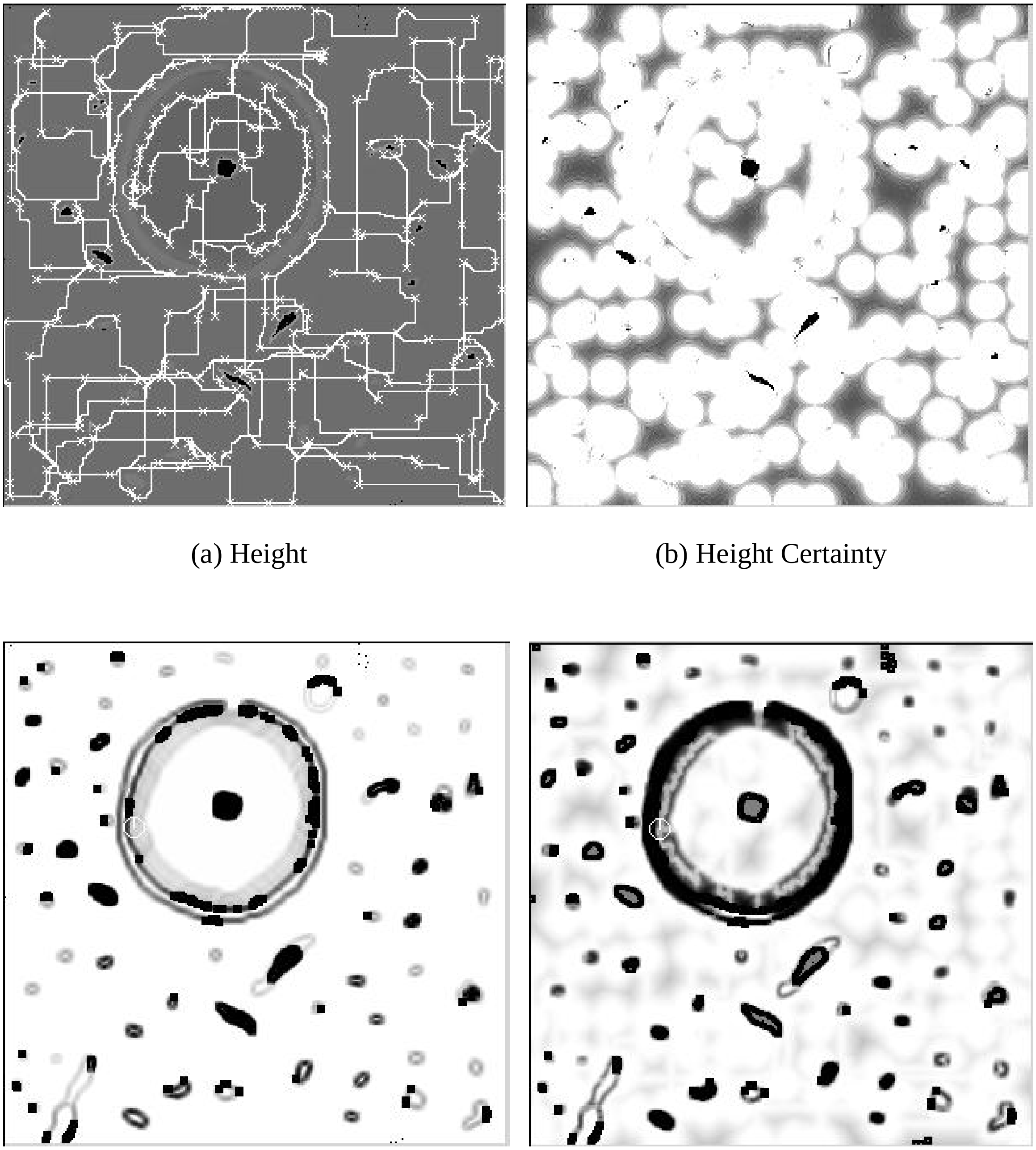
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-4:Run 1c Path Results.



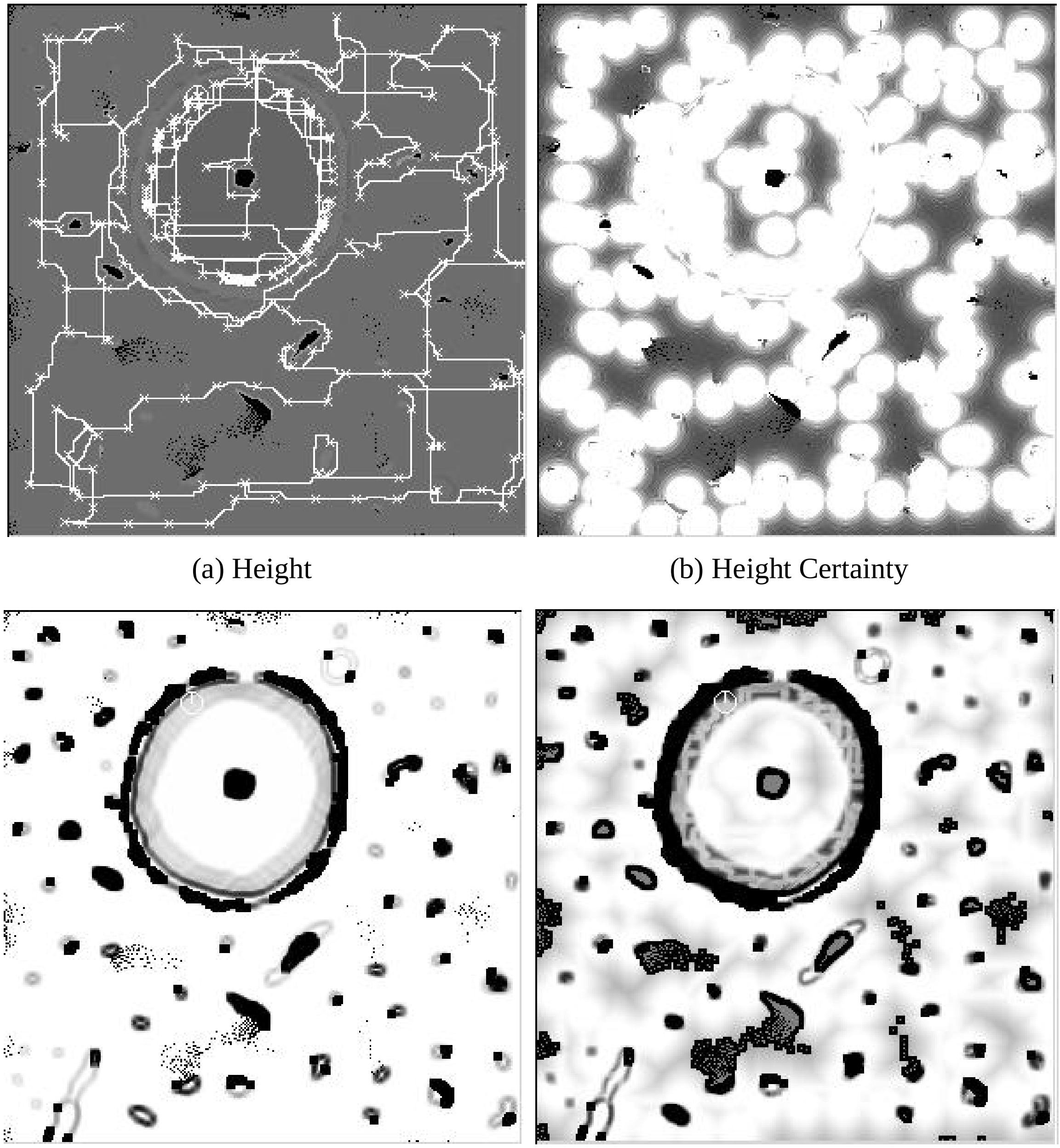
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-5:Run 1d Path Results.



(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-6:Run 1e Path Results.

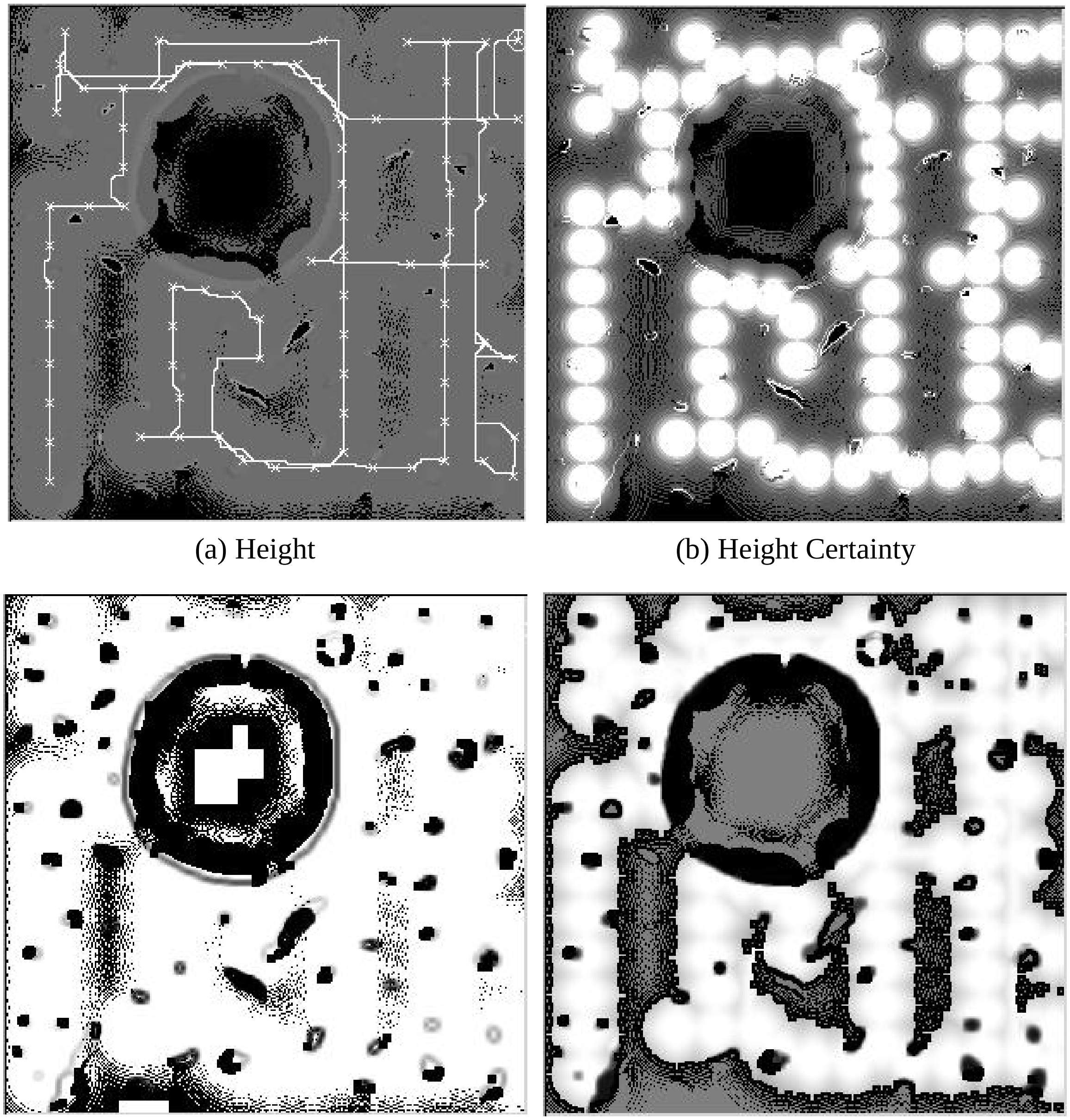
A.2 Run 2

Table A-4: Run 2 Create Traversability Map Exploration Gains

Final Map Information	**A (kbits)**	**B (kbits)**	**C (kbits)**	**D (kbits)**	**E (kbits)**
Height Information	52.8	59.5	63.3	58.6	58.0
Height Info weighted by Traversability	7.3	7.7	9.6	7.8	8.1
Reachability Information	61.2	66.0	66.5	66.1	64.4

Table A-5: Run 2 Create Traversability Map Exploration Costs

Exploration Costs	**A**	**B**	**C**	**D**	**E**
Path Length (meters)	1519	1997	2622	1915	2367
Normalized Path Length (path length / site side length)	5.1	6.7	8.7	6.4	7.9
Number of sensor readings	78	89	103	89	92
Avg. Distance between Sensor Readings (m/scan)	19.5	22.4	25.5	21.5	25.7



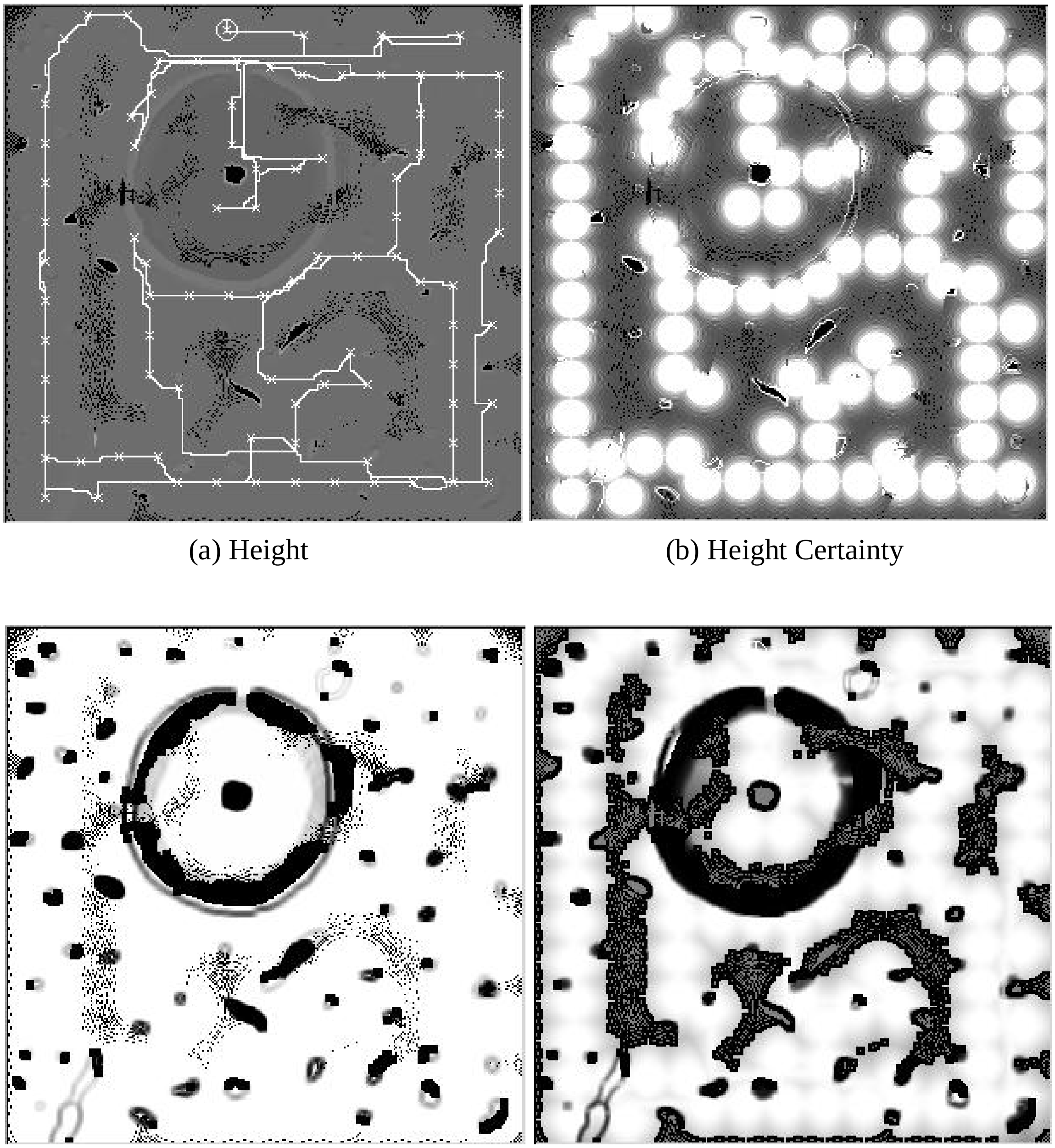
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-7:Run 2a Path Results.



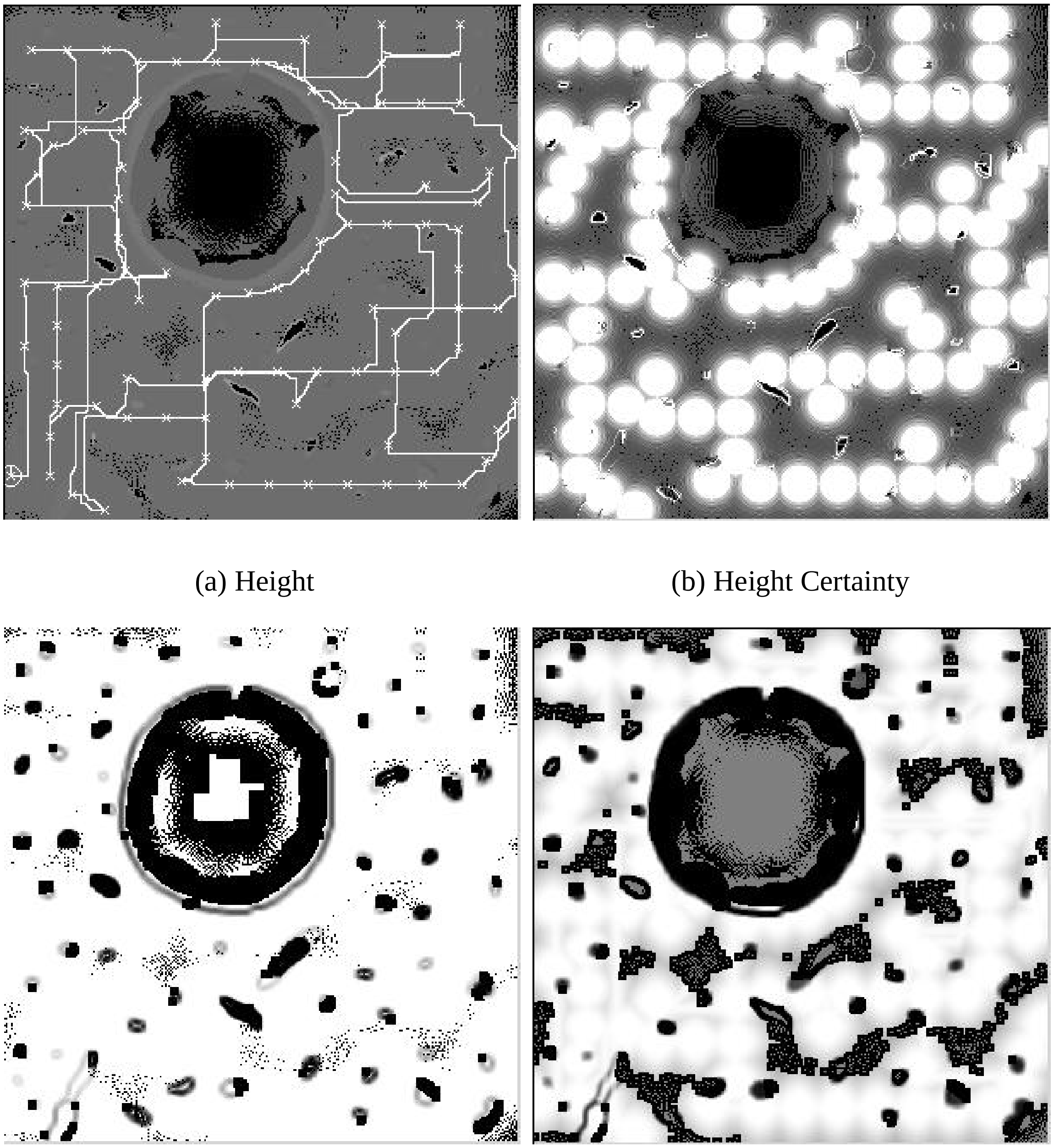
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-8:Run 2b Path Results.



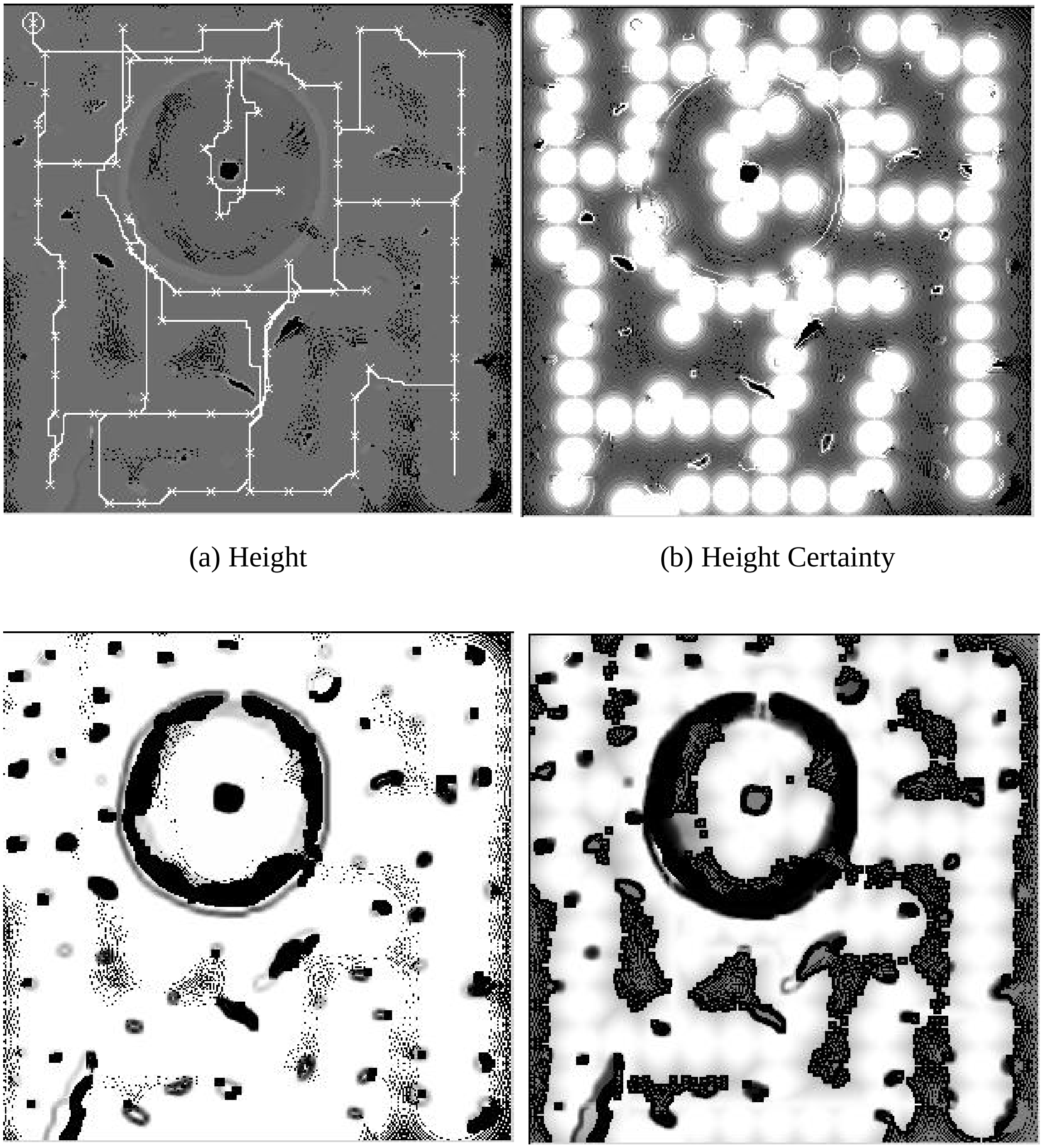
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-9:Run 2c Path Results.



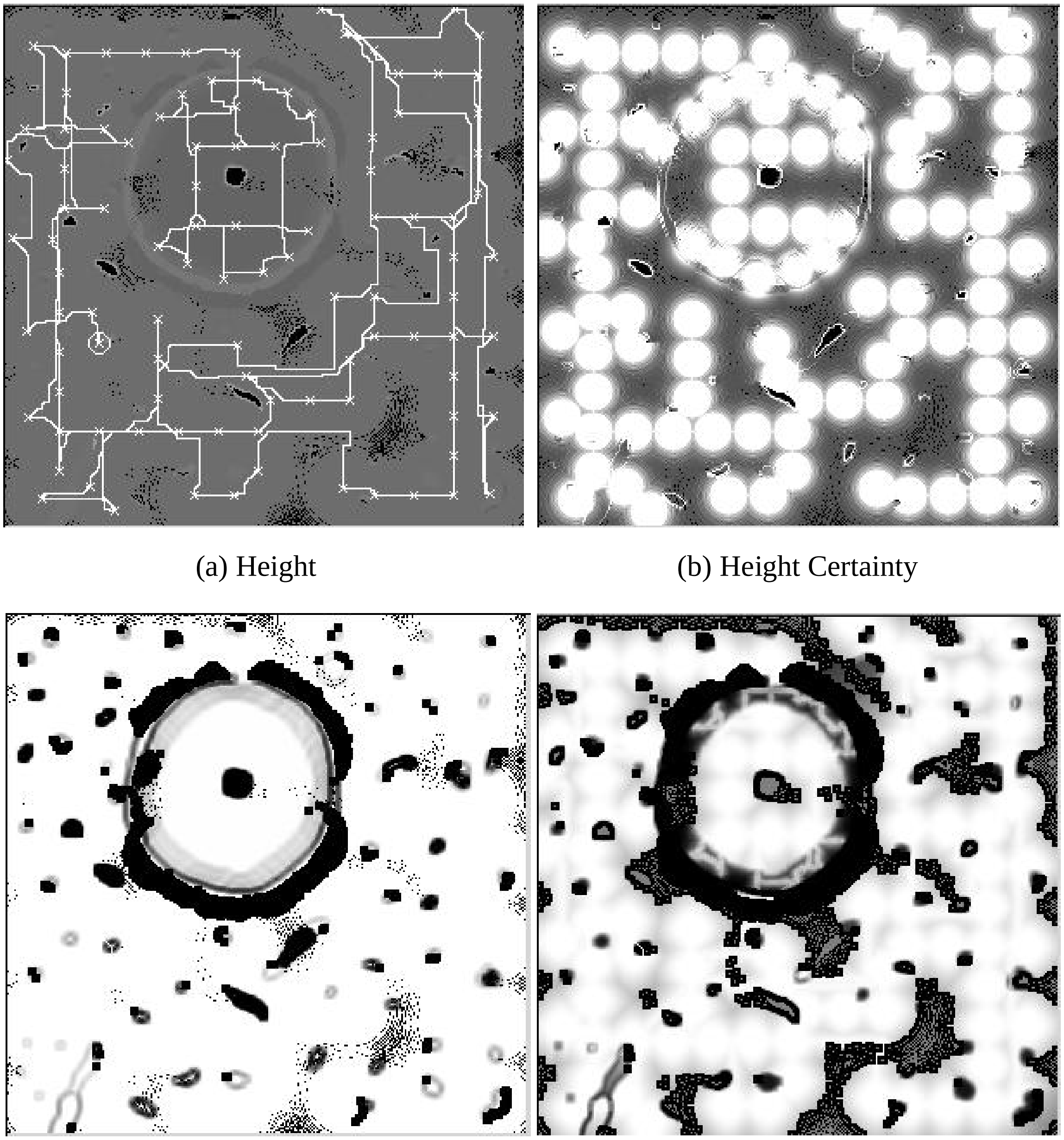
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-10:Run 2d Path Results.



(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-11:Run 2e Path Results.

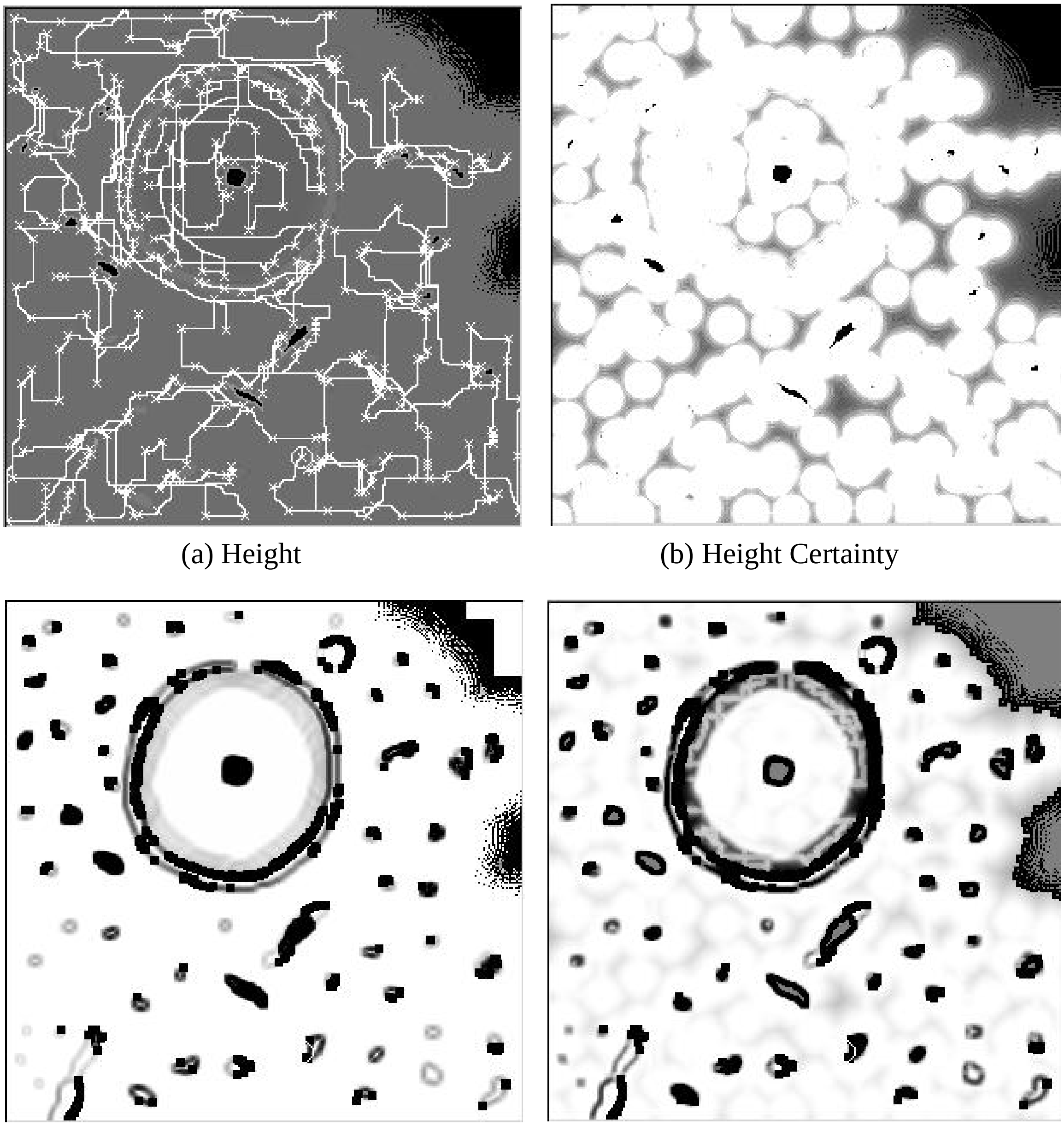
A.3 Run 3

Table A-6: Run 3 Create Traversability Map Exploration Gains

Final Map Information	A (kbits)	B (kbits)	C (kbits)	D (kbits)	E (kbits)
Height Information	79.1	86.7	87.6	75.8	80.9
Height Info weighted by Traversability	12.8	13.4	13.7	12.5	12.7
Reachability Information	71.9	78.2	79.8	69.2	73.3

Table A-7: Run 3 Create Traversability Map Exploration Costs

Exploration Costs	A	B	C	D	E
Path Length (meters)	5791	7100	6997	5527	6099
Normalized Path Length (path length / site side length)	19.3	23.7	23.3	18.4	20.3
Number of sensor readings	415	506	531	407	452
Avg. Distance between Sensor Readings (m/scan)	14.0	14.0	13.2	13.6	13.5



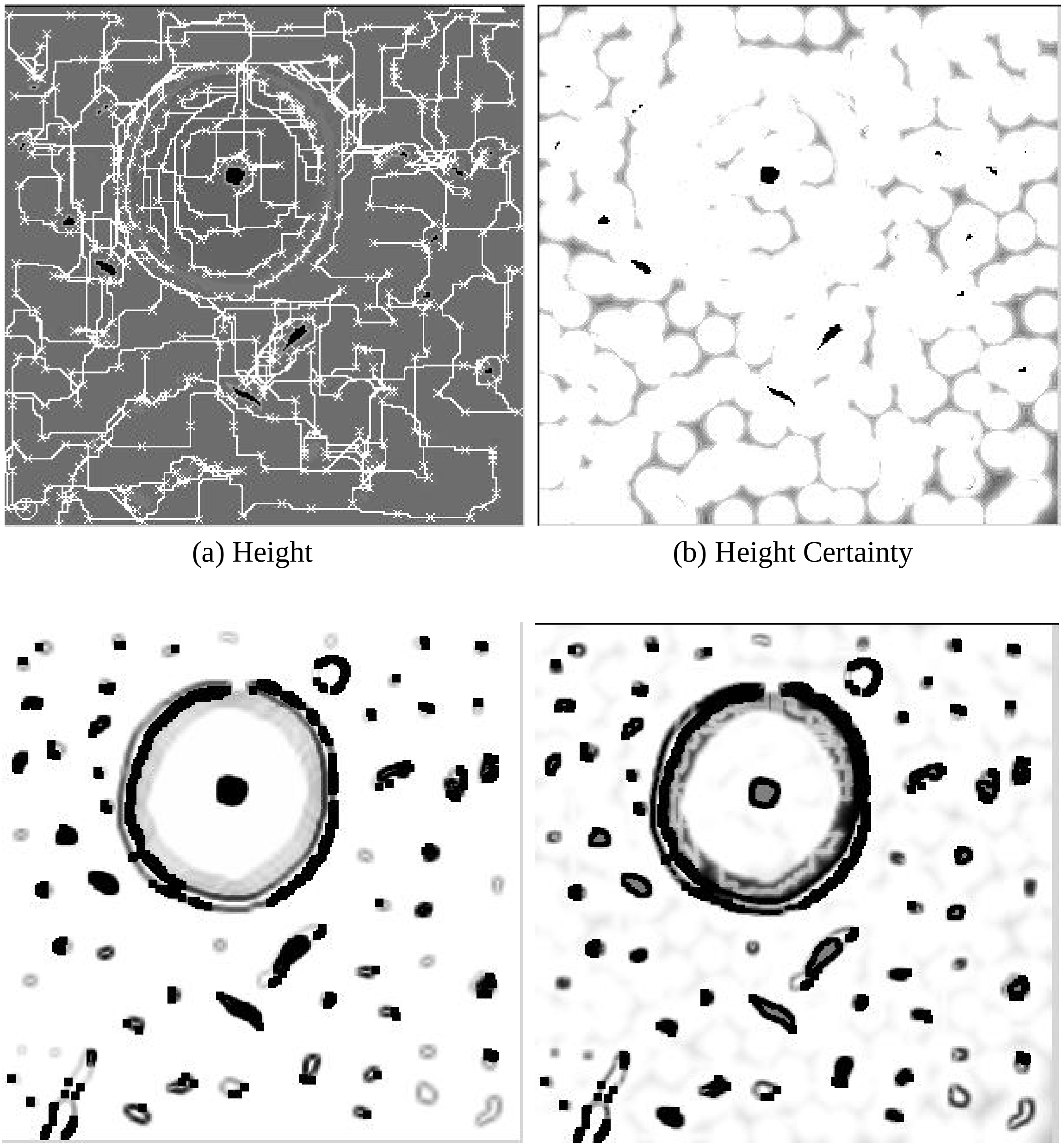
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-12:Run 3a Path Results.



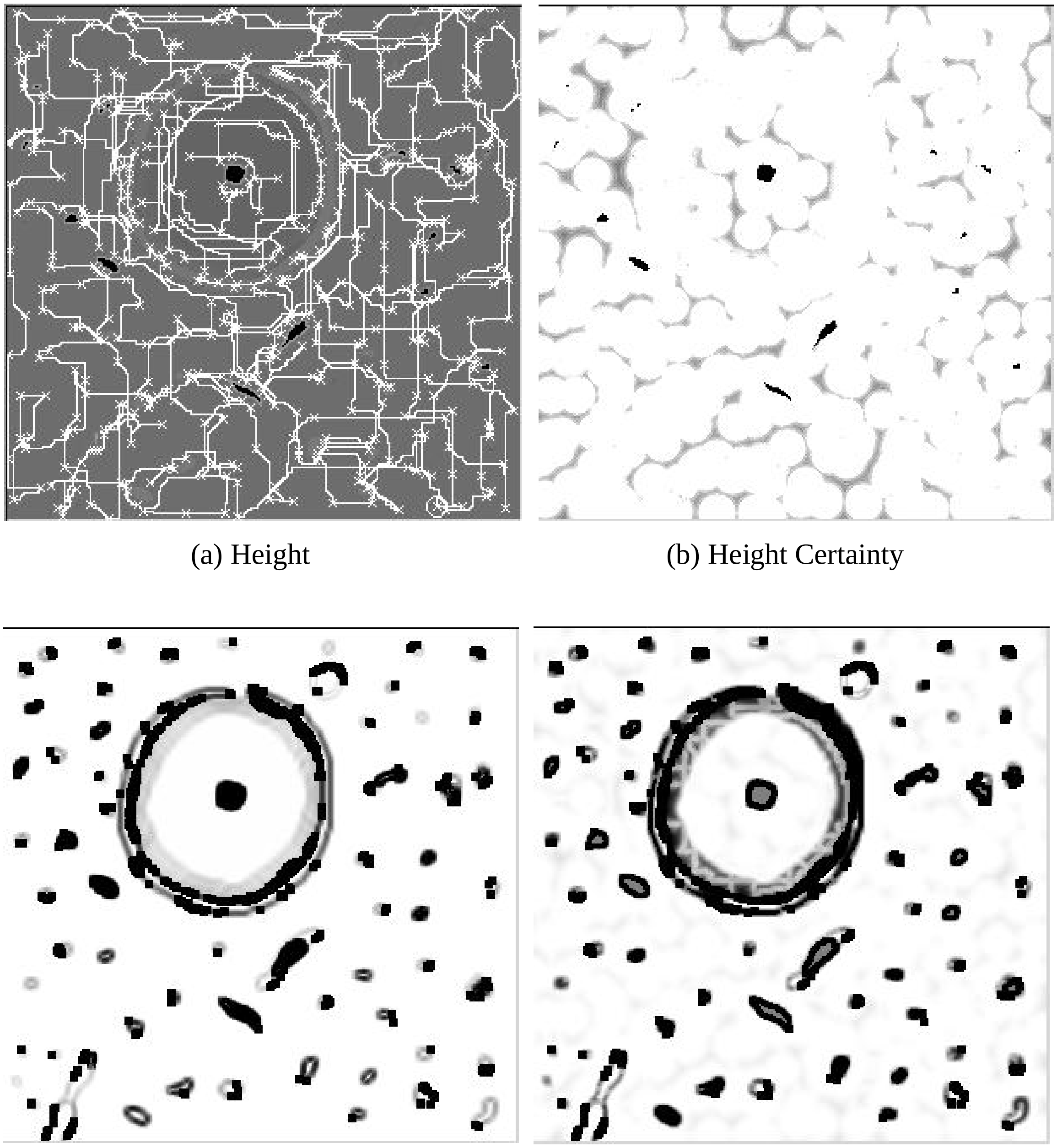
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-13:Run 3b Path Results.



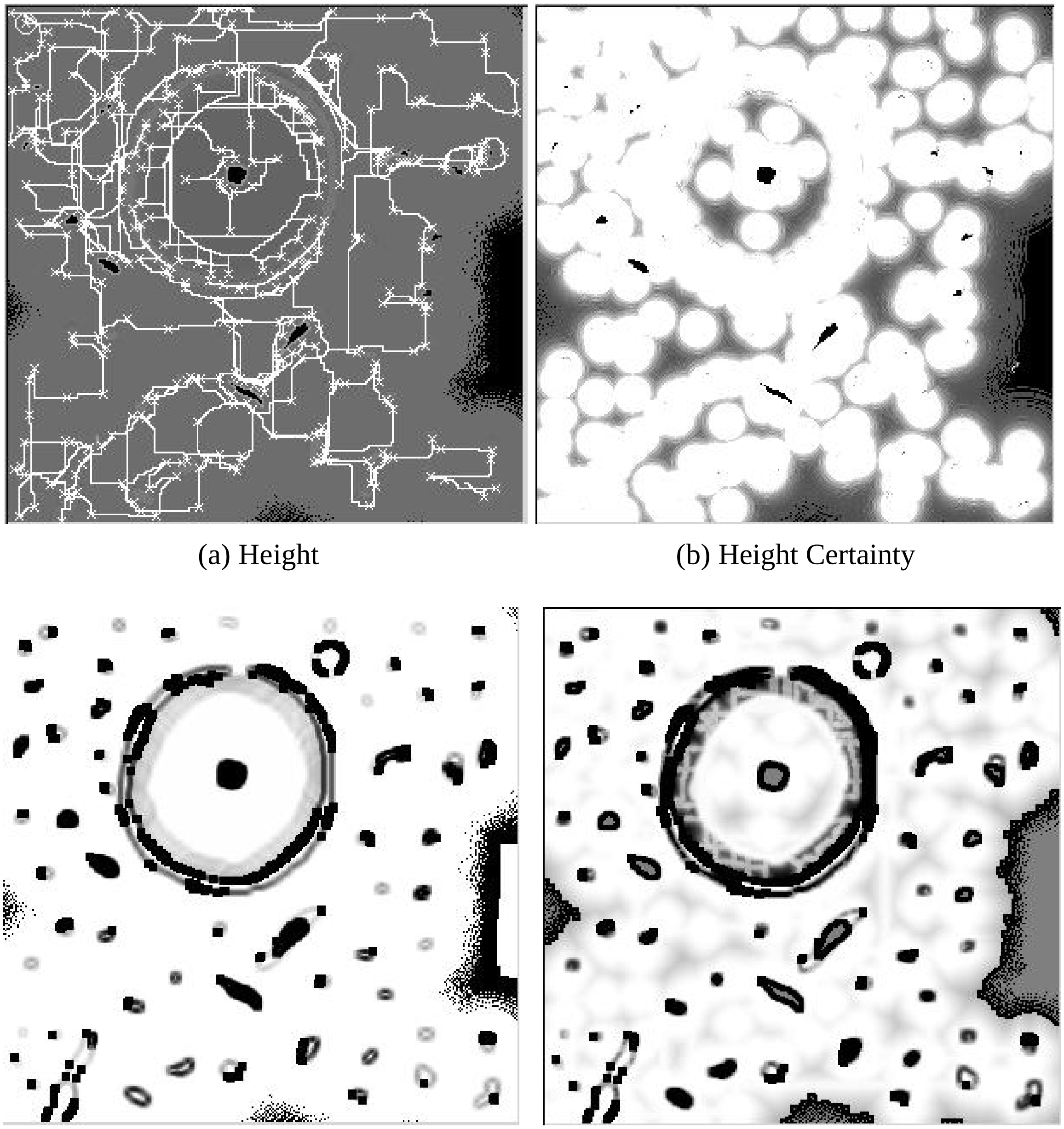
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-14:Run 3c Path Results.



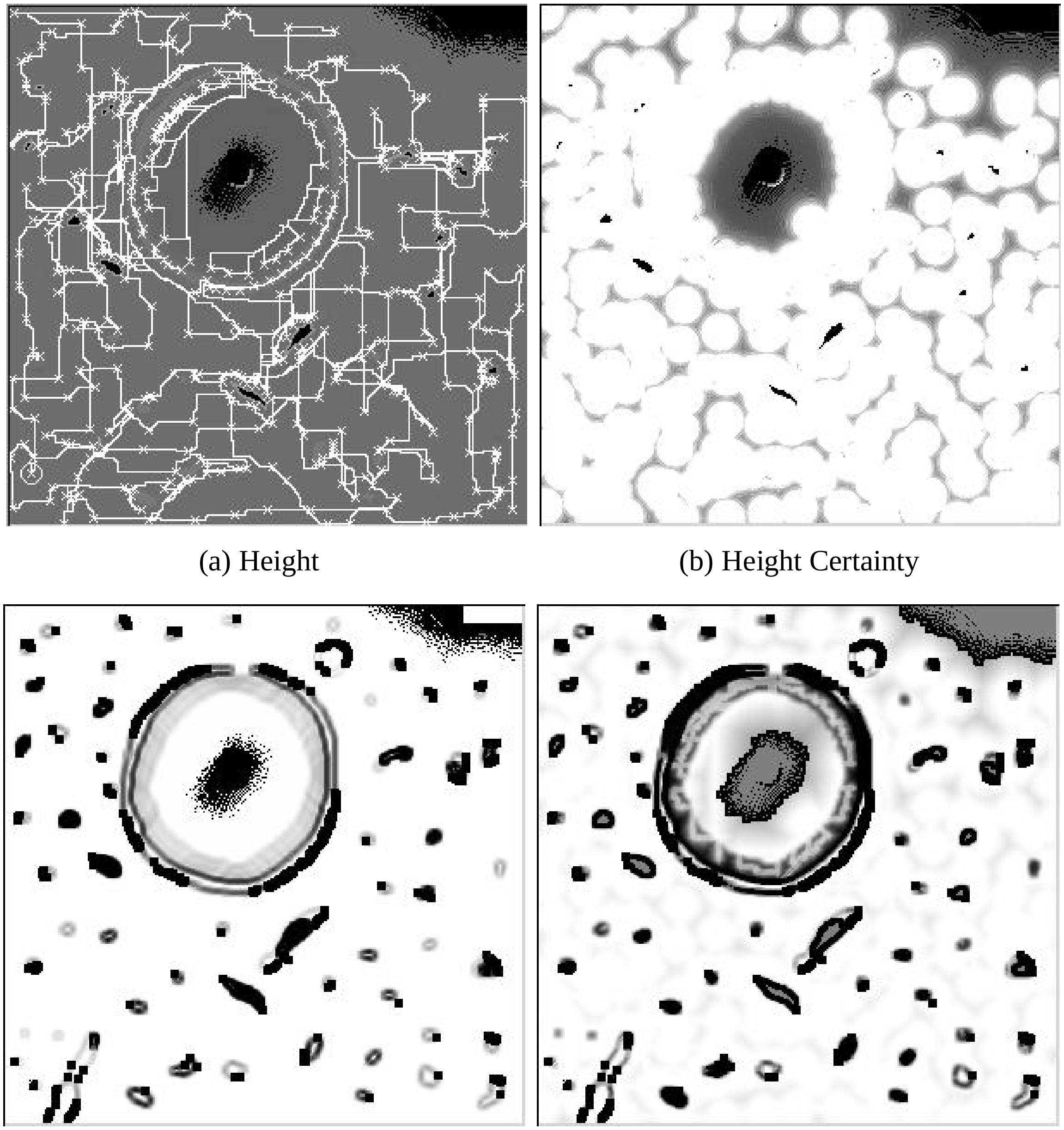
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-15:Run 3d Path Results.



(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-16:Run 3e Path Results.

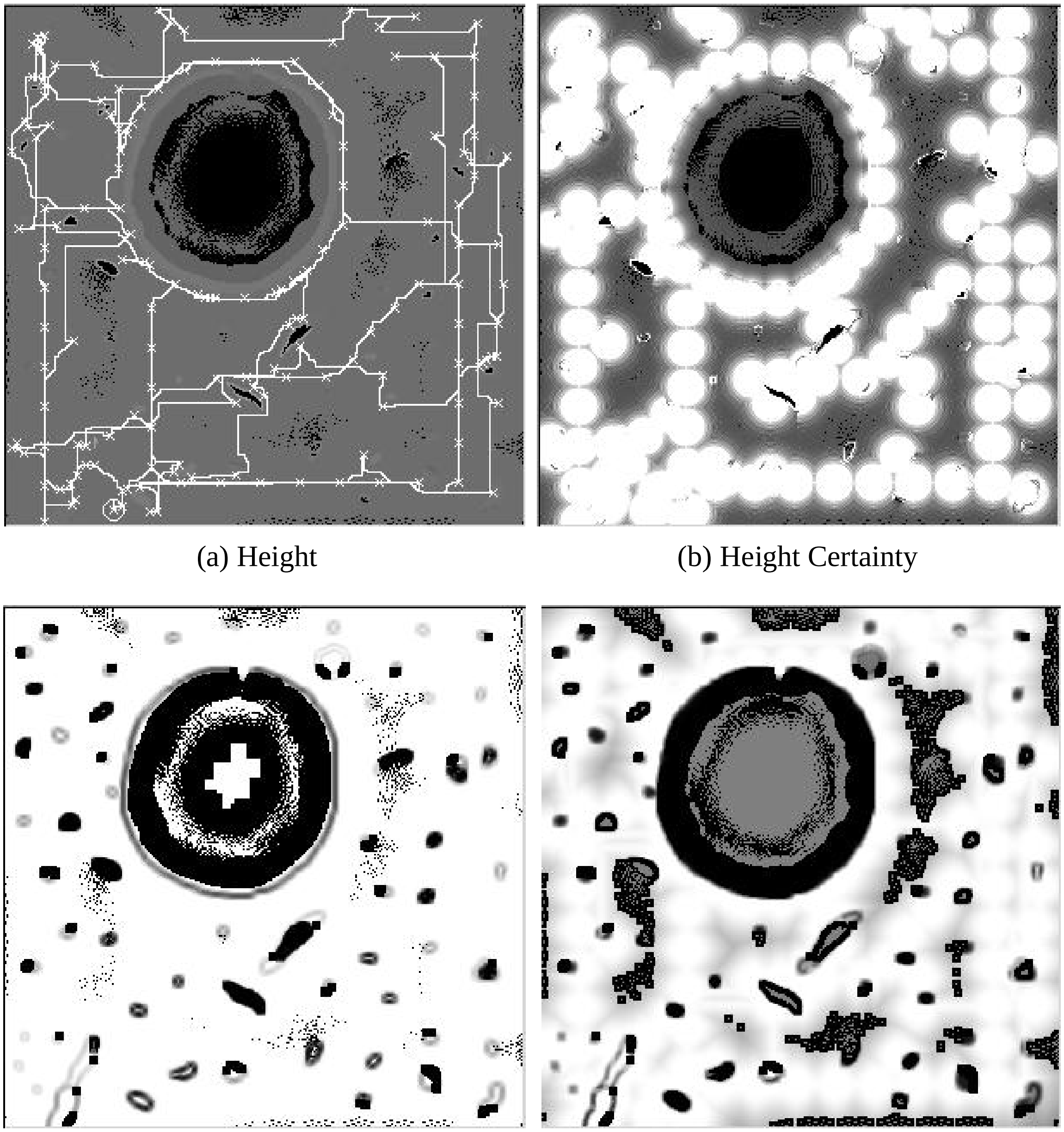
A.4 Run 4

Table A-8: Run 4 Create Traversability Map Exploration Gains

Final Map Information	A (kbits)	B (kbits)	C (kbits)	D (kbits)	E (kbits)
Height Information	62.2	65.8	68.2	72.8	67.3
Height Info weighted by Traversability	9.1	9.5	9.7	10.2	10.3
Reachability Information	63.9	64.5	64.8	67.3	65.2

Table A-9: Run 4 Create Traversability Map Exploration Costs

Exploration Costs	A	B	C	D	E
Path Length (meters)	2708	2799	3247	3965	3012
Normalized Path Length (path length / site side length)	9.0	9.3	10.8	13.2	10.0
Number of sensor readings	158	154	173	219	171
Avg. Distance between Sensor Readings (m/scan)	17.1	18.2	18.8	18.1	17.6



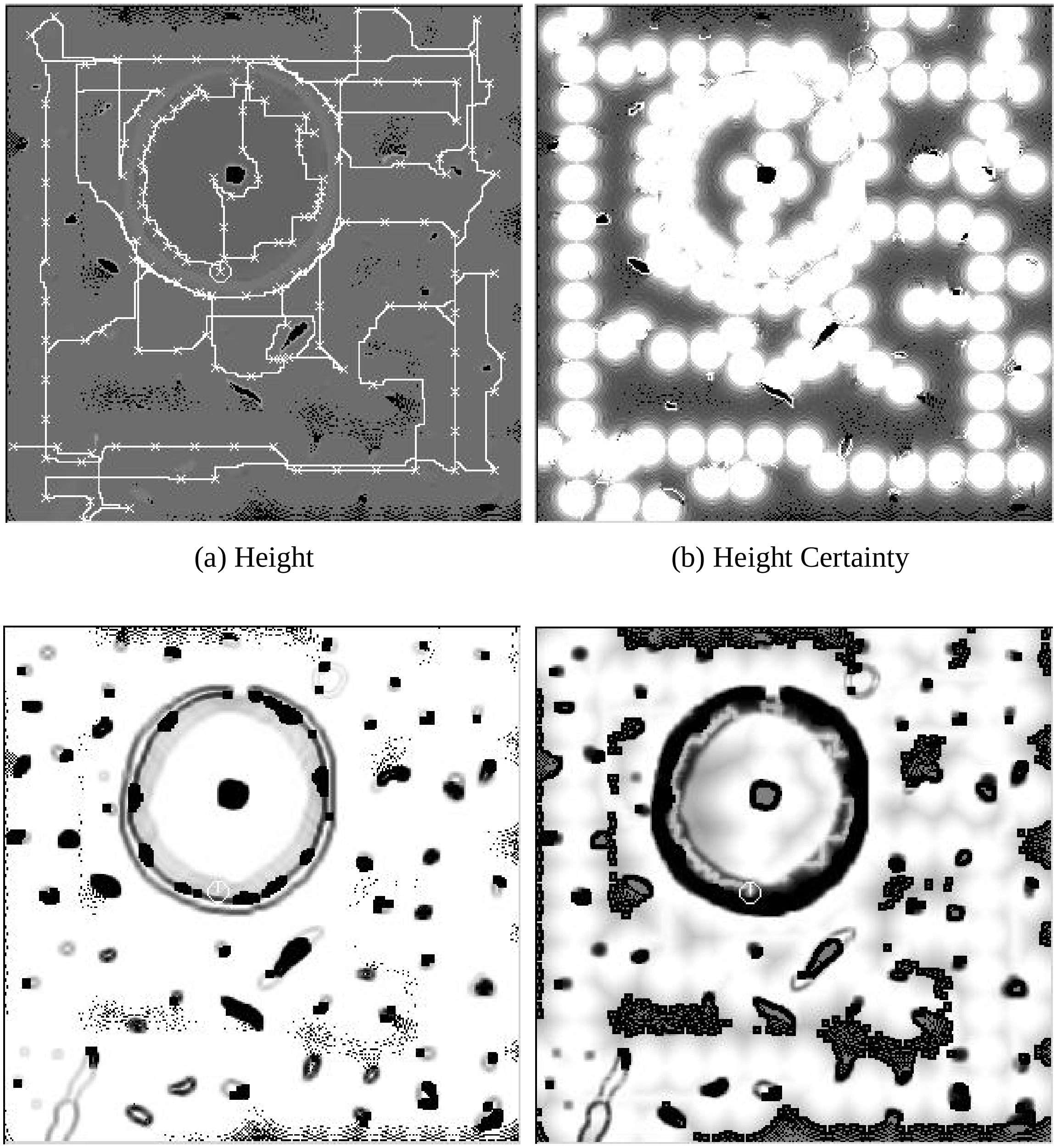
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-17:Run 4a Path Results.



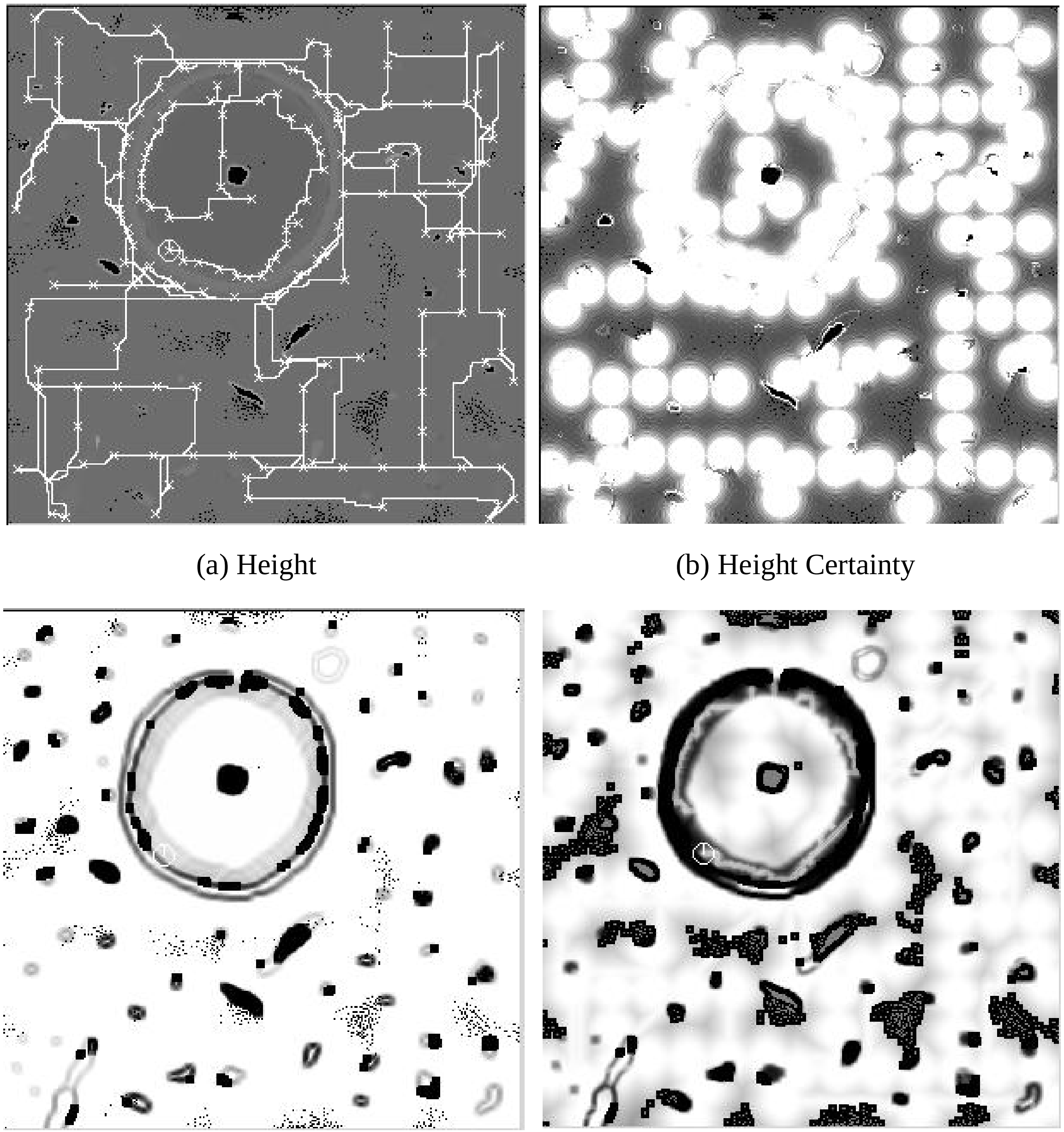
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-18:Run 4b Path Results.



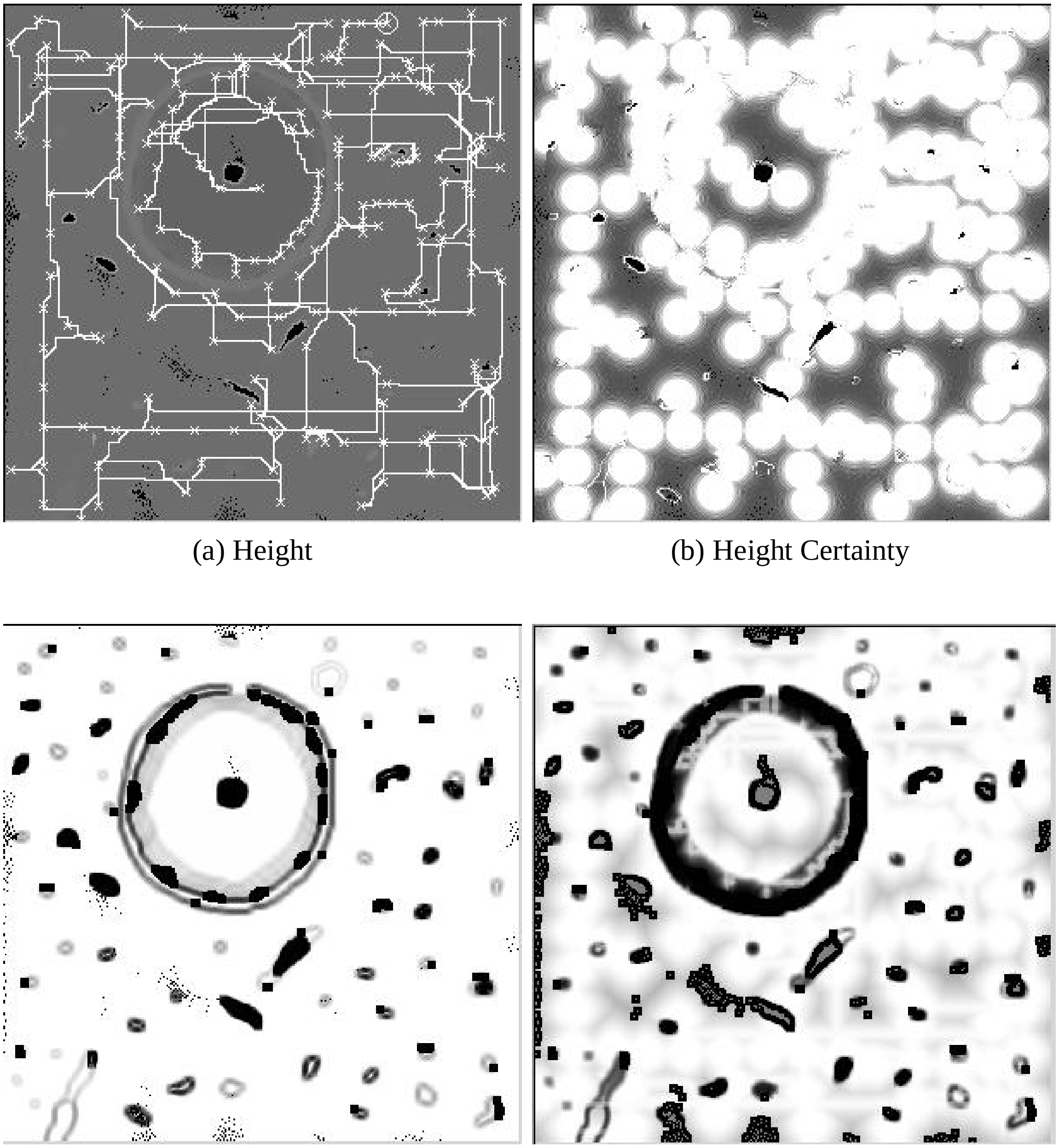
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-19:Run 4c Path Results.



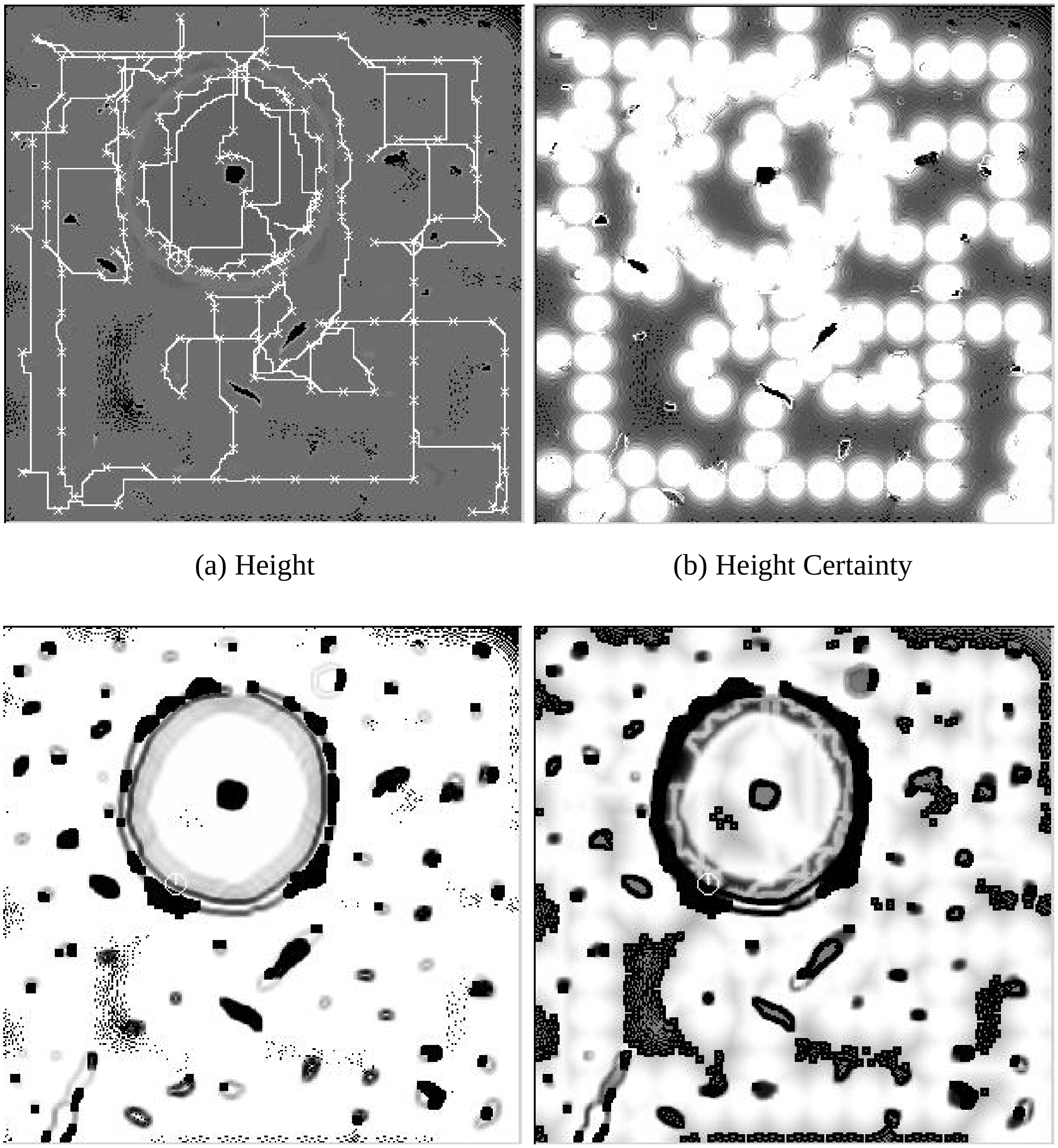
(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-20:Run 4d Path Results.



(a) Height

(b) Height Certainty

(c) Traversability

(d) Reachability

Figure A-21:Run 4e Path Results.