Path Planning for Robotic Demining and Development of a Test Platform

Ercan U. Acar, Yangang Zhang, Howie Choset, Mark Schervish,
Albert G. Costa, Renata Melamud, David C. Lean, Amy Graveline
Carnegie Mellon University, Pittsburgh, U.S.A.

Abstract: This paper overviews path planning methods, systematic versus probabilistic for robotic demining, and a robot development project. We outline a complete sensor-based coverage algorithm that uses an exact cellular decomposition method. To decrease coverage duration, we introduce a probabilistic method that takes advantage of *a priori* known mine patterns. Then, we give a comparison of random versus coordinated coverage. Finally, we present a summary of the development of the mobile test platform.

Keywords: Demining, Coverage.

1 Introduction

The problem of detecting mines in a surface-laid minefield using autonomous robots is of interest to civilians and military alike. The use of robots decreases the danger and the cost involved in manual mine detection. [Trevelyan, 1998] gives an overview of the world-wide demining problem and outlines the challenges. We believe that the main challenge for demining is the development of better mine detection methods. Currently, metal detectors are widely used mine detectors. However there are efforts to build detectors such as infrared imaging, TNT sniffing etc.. In this paper, we focus on path planning methods for demining, assuming that a mine detector is available to us (not necessarily perfect).

The robot can employ a coverage algorithm, a path planning technique where the robot explicitly passes over all points in the minefield at least once. We achieve complete coverage using an exact cellular decomposition and a coverage algorithm. The coverage algorithm incrementally constructs the cellular decomposition and covers the space simultaneously. Our algorithm only requires a sensor suit that can guide the robot along the boundaries of the obstacles. This feature of the algorithm leads to development of low cost robots that can be perishable.

When resources are limited, the robot's planner can use *a priori* information to opportunistically guide its search. We assume that the mines are laid out using a regular pattern characterized by six parameters. The robot covers only a small portion of the field to estimate these parameters using our probabilistic algorithm. Once the parameters are determined the robot can visit each mine location and does not need to cover the entire field.

The present state of the art for path planning for demining is to move around randomly. Even though performing random motions does not require sophisticated sensor suits, making sure that a minefield is completely covered is not feasible. In section 2.3, we compare the coverage time taken by our complete coverage algorithm to the time required by random motions in the presence of a variety of obstacle configurations.

Finally, we discuss about the choices that we made concerning the cost of the robot platform, maneuverability, positioning, and on-board computing. Our intentions are not to describe the ultimate how-to-build a bread-box sized outdoor mobile robot, but rather share with the community some of experiences and choices in developing an outdoor demining robot.

2 Path Planning for Demining

Humanitarian demining requires complete coverage of a minefield in order to locate and then recovery of all the mines. Thus the first algorithm that we consider is complete sensor-based coverage (Section 2.1). However rather than performing exhaustive coverage, if we know *a priori* that a mine pattern exists, we can direct the robot to certain

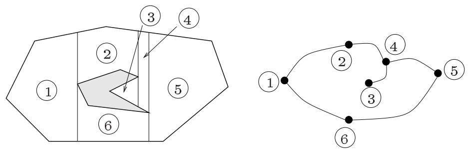


Figure 1: The boustrophedon decomposition of an environment and its adjacency graph. Nodes are the cells and edges represent adjacency information.

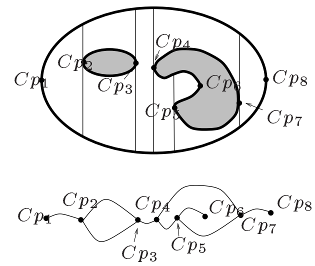


Figure 2: (a) Exact cellular decomposition of the environment in terms of critical points Cp_k. (b) Graph representation of the environment. Nodes represent critical points, edges represent cells.

locations where existence of a mine is very likely. This can be useful in military demining where time is critically limited. In Section 2.2, we give an outline of this probabilistic approach. Then we compare the performance of a robot that is executing our complete coverage algorithm with a robot that moves around randomly.

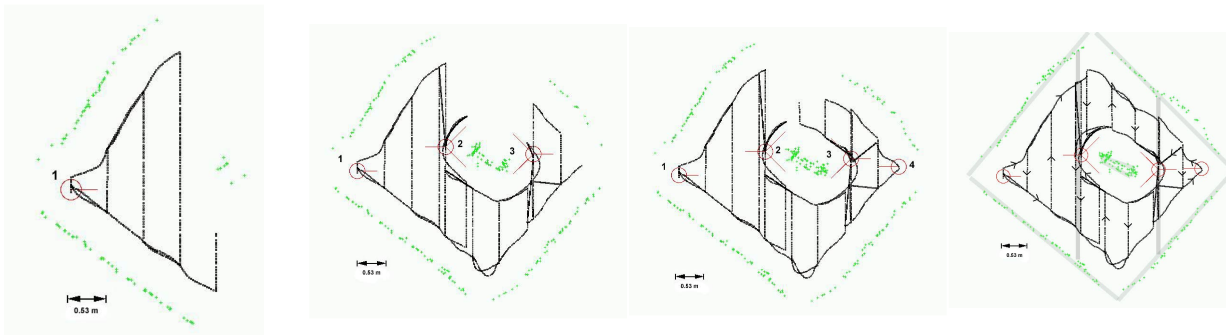
2.1 Sensor-based Coverage for Demining

There exists a variety of coverage algorithms ([Zelinsky et al., 1993],[Cao et al., 1988], [Lumelsky et al., 1990],[Hert et al., 1996]) using different approaches and making certain assumptions about the environment and the sensors. Our method for coverage is based on a geometric structure called cellular decomposition [Latombe, 1991], which is the union of non-overlapping subregions of the free space, called *cells*. An adjacency graph encodes the topology of the cells in the environment where nodes are cells and edges connect nodes of adjacent cells (Fig. 1). We define our cells such that simple back and forth motions cover each cell, and thus *complete* coverage is reduced to finding an exhaustive walk through the adjacency graph [Choset and Pignon, 1997].

Our cells are defined such that simple motions, such as back and forth motions, can simply cover the cell. We use a particular function, called a Morse Function[1], to model this type of "simple" motion. The cell boundaries are then defined by critical points[2] of these Morse functions [Choset et al., 2000]. It is worth noting that for the work in this paper, the critical points occur on the boundary of the robot's free space. If the robot knows the critical points, then it effectively knows the decomposition. When the environment is not known, neither are the critical points and thus sensor based coverage is reduced to covering the environment while determining the locations of critical points. We presented a method to sense critical points in unknown environments using range measurements in an earlier paper [Acar and Choset, 2000].

Generically, each cell is characterized by two critical points[3]. Instead of forming an adjacency graph with nodes as cells, we form a graph where nodes are critical points and edges are the cells (Fig. 2). This particular graph representation encodes all the information we need to incrementally construct the cellular decomposition. Each time the robot encounters a new critical point, a new node is created, the edge corresponding to the current cell is terminated at the new node, and depending on the type of the critical point, two more edges are instantiated or no edge is created. If the robot encounters an already discovered critical point, then the edge corresponding to the current cell is terminated at the critical point and the "dangling" edge (*i.e.* it only has one node) of the already discovered critical point is deleted. When all the nodes have edges ending with another node, coverage is completed. We presented a coverage algorithm that guarantees that the robot will encounter all the critical points while its is achieving coverage in [Acar and Choset, 2000]. An experimental result obtained running our coverage algorithm on Nomad Mobile robot [Nomad, 1996] is shown in Fig. 3.

[1]Functions with non-degenerate critical points [Milnor, 1963].
[2]At the critical points, a function takes its extrema.
[3]Handling multiple critical points defining a single cell's boundary is an implementation detail.



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

Figure 3: The coverage path followed by the robot in an unknown environment is shown by dotted black lines. The gray dots represent the back track paths. The robot incrementally constructs the graph representation by sensing the critical points $1, 2, 3, 4, 2$ (in the order of appearance) while covering the space. Since all the critical points have explored edges, the robot concludes that it has completely covered the space. For the sake of discussion, we outlined the boundaries of the obstacles and cells in (d).

2.2 Probabilistic Demining

Exhaustive coverage is the best strategy when the robot has unlimited time and a perfect mine detector. However, in many situations time or power limitations may not permit covering a target environment completely. Probabilistic planning can significantly improve the search efficiency in such demining applications. Probabilistic search in an unknown environment raises two major questions: (1) How we can efficiently construct a probabilistic map of mine locations by navigating the target area? (2) Given *a priori* information, what is the optimal search path to guide a robot to locate mines. [Gelenbe and Cao, 1998] have discussed their Simplified Infinite Horizon Optimization algorithm to answer the second question. In our research, we focus our efforts to solve the first question.

Extracting characteristics of a dispersion pattern of a minefield helps to quickly build a probability map and plan a demining path. There are two types of typical dispersion patterns: scattered and regular pattern. Scattered patterns are usually produced by sub-munitions released from a plane or projectile. Elliptical impact pattern with the higher density of impacts towards the center is also a common dispersion pattern. When mines are deployed by ground vehicles or human, it is possible that minefields follow some forms of regularity, because of the military doctrine, tactical efficiency and inherent limitations in the mine laying process [Lake and Keenan, 1995]. Typical characteristics of regularity are collinearity and equal-spacing.

As a start-up problem, we work on extracting the characteristics of a regular pattern and we will move to model the scattered pattern in our future work. We focus on a particular minefield pattern shown in Fig. 4. The key to this work is to extract the "true" pattern parameters of the spatial distribution during the process of the detection. The extracted pattern information can be used to design optimal search strategies and provide real-time decision models for spatial orientation of robots.

The following are the constraints that we consider:

- **The method should be able to deal with uncertain information.** The detector error complicates the problem. Detector produces false negative, false positive errors and inaccurate reading about the location of a detected mine. Moreover, the model of the minefield pattern is generally inaccurate. Possible reasons for deviations of the model from the real world are deployment errors, mine explosions and model simplifications. All of these random errors increase the difficulty to decode the underlying pattern.

- **The method should be computationally efficient.** It is important that the computation can be finished in real-time on the robot.

The robot first covers a small portion A of the minefield. The observed information is a set of detected mines at positions $\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_k)$, where

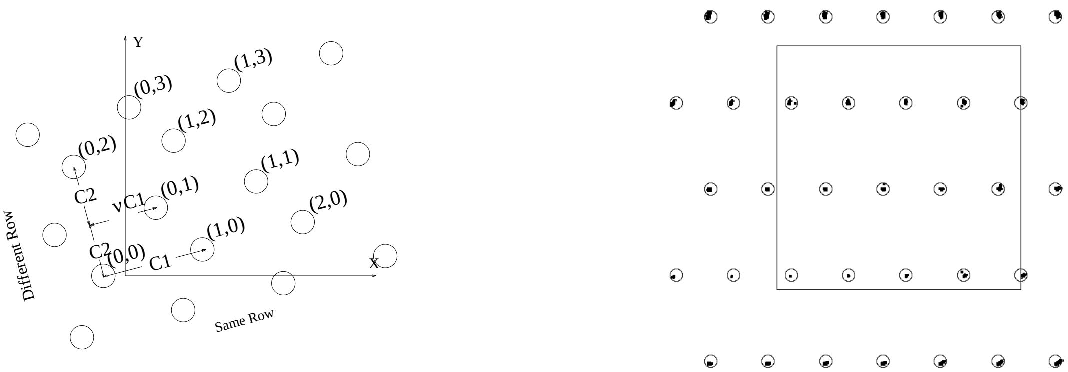


Figure 4: The grid pattern with one row shifted with respect to the neighboring row. The pattern can be characterized by six parameters, inter-row and column spacing, amount of row shift, orientation and position of the minefield with respect to a fixed-frame. Sample of possible intended minefields simulated from MCMC algorithm (11 ”found” mines). Simulated minefield matches the “true” intended minefield.

$\mathbf{y}_i \in A$, $i = 1, ..., k$. A Bayesian approach is used to solve the pattern parameter estimation problem using the information collected in the covered region. A Bayesian approach calculates the posterior distribution $f_{\mathbf{\Delta}|\mathbf{Y}}(\delta|\mathbf{y}) = \frac{f_{\mathbf{Y}|\mathbf{\Delta}}(\mathbf{y}|\delta) f_{\mathbf{\Delta}}(\delta)}{f_{\mathbf{Y}}(\mathbf{y})}$ after observing the locations of some mines (Δ is the set of parameters). Meanwhile, the observed mines $\mathbf{y}$ depend on the true minefield pattern $\mathbf{x}(\delta)$ through a known conditional probability density $f_{\mathbf{Y}|\mathbf{\Delta}}(\mathbf{y}|\mathbf{x}(\delta))$, which is also called likelihood function. $f_{\mathbf{Y}|\mathbf{\Delta}}$ can be specified based on a noise model. The posterior distribution is often impossible to compute in closed form, and even if it were possible, the density $f_{\mathbf{\Delta}|\mathbf{Y}}$ is typically impossible to recognize as anything familiar. Instead of trying to calculate the density, we will use Markov Chain Monte Carlo (MCMC) to create a sample from the posterior distribution of the parameters.

Results from simulation experiments indicate that our approach works well in terms of performance and efficiency. A set of “found” mines in a covered region is randomly generated based on the noise model from a “true” intended minefield characterized by the parameters $\mathbf{\Delta}$. In Fig. 4, we show the sample of possible intended minefield simulated from our MCMC algorithm based on the posterior distribution of $\mathbf{\Delta}$.

2.3 Random versus Coordinated Coverage

Currently, the state of the art for robotic demining is to drive the robot randomly in a minefield. To establish the trade offs between the random versus coordinated strategies, we provide the analysis of simulation experiments. In our simulations we place 100 mines uniformly on a $100 \times 100 ft^2$ area. Simulation algorithms are used to obtain the search time by both random and coordinated strategies. The mine search follows the following procedure: the robot first locates a mine, then picks the mine, delivers it to the boundary and finally drops the mine on the boundary. The width of the detector is $1ft$. The robot moves at $1ft/s$. It rotates at 0.1 revolution/s. It spends totally two minutes to pick and drop the mine, not including the travel time.

First, we compare both random and coordinated coverage methods in an environment without obstacles. Figure 5 shows the plot of the number of found mines versus total search time in the random search. Table 2.3 summarizes what we have found. With a perfect sensor (100% detection rate), in order to find 100% of targets, random search takes much longer time compared to the coordinated search. With imperfect sensor, difference in mean search time between random search and coordinated search becomes smaller when the detection rate becomes worse. However, the variability of search time for random search is much larger than the coordinated search. Therefore achieving complete coverage with random motions is not feasible.

With obstacles, coverage duration of random search could increase drastically; coverage duration for coordinated search is proportional to the size of the target area and it will not change dramatically with the presence of obstacles. Here, we present two types of obstacle configurations

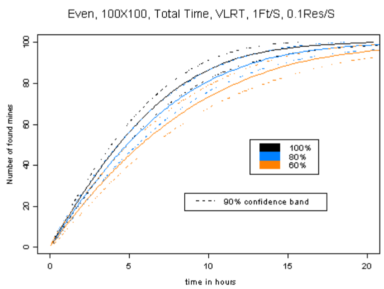


Figure 5: Plot of the number of found mines versus total search time by random search using different detection rates ($100 \times 100 ft^2$, no obstacles).

	Detection rate	60%	80%	100%
Find 100% targets	Random	-	-	20.0h
	Coordinated	-	-	7.25h
Find 80% targets	Random	11.6	9.70h	8.9h
	Coordinated	8.75h	6.38h	5.80h
	Ratio	1.3	1.5	1.53

The mean search time for random and coordinated search methods.

(Figs. 6, 8). In the first configuration, the robot bumps among obstacles; which make it difficult to move to uncovered area. Figure 7 shows the amount of found mines versus total search time. The search time is longer in this case. In the second obstacle configuration, we place all of the mines inside the area surrounded by the obstacles. In random search, once the robot moves outside the area enclosed by the obstacles, it has very small chance to move back the enclosed area, because the size of the exit is very small. In this case, the random search time increases dramatically. Figure 9 shows the amount of found mines versus total search time.

3 Development of the Test Platform

The conceptual design of the demining robot began with evaluating several existing mechanisms for outdoor navigation. We considered advantages and disadvantages of locomotion exhibited by robots such as the Nomad Antartica Meteorite Search Robot, iRobot's Real World Product Line (RWI), the Marsokhod, Swiss Federal Institute of Technology Pemex robot, and the Free University of Brussels Tridem robot. The type of locomotion used can generally be divided into four categories: skid steering robots, legged robots, differential drive robots, and articulated drive mechanisms. Nomad, Marsokhod and Tridem are examples of complex mechanisms that allow high mobility but are more difficult to control, have greater power requirements, and are more expensive to manufacture. Legged robots provide discretized movement which is helpful for reducing positioning error and perhaps in the future, greater maneuverability. However, legged robots, like the complex wheeled mechanisms, are prohibitively expensive and require more power than simple wheeled mechanisms. Less power translates to less cost, weight and size.

Skid steering serves as a simple drive mechanism that has the advantage of bulldozing through rugged terrain quite easily. The RWI outdoor ATR's have skid steering mobile platforms. Unfortunately, for our purposes, a skid steering robot has two drawbacks. First, the robot must overcome a great deal of friction in order to make a simple turn. The RWI ATR's use four one horse power motors, which again increase cost, power, size, and weight budgets of the robot for our needs. The second reason why we chose against skid steering is controllability in terms of being able to precisely cover an area. Using dead-reckoning to determine the robot's position with skid steering is impossible because the point of contact with the ground is constantly changing during rotation.

The final alternative is a simple differential drive robot that uses casters to provide support. This affords a simple and cost effective design that is easy to manufacture and implement. Unfortunately, what we gained in cost, low-weight, low power, and small size, we lost in maneuverability. The casters that supported the back-side of our robot also caused problems, especially when driving on soft damp ground. Our first robot, named Finder (Figure 10), is a differential drive robot that uses inexpensive materials and parts commonly found in catalogs. We recently manufactured a second mobile platform that has some improvements over Finder. This robot is named Slugger (See Figure 10). A unique aspect of Slugger is the mounting of its twenty-six ultrasound sensors. We designed a modular sonar track which fits

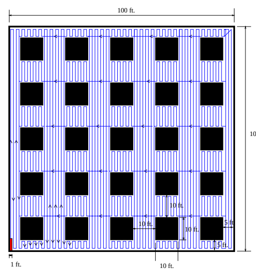


Figure 6: The search path by the coordinated search in a $100 \times 100 ft^2$ environment with 25 obstacles.

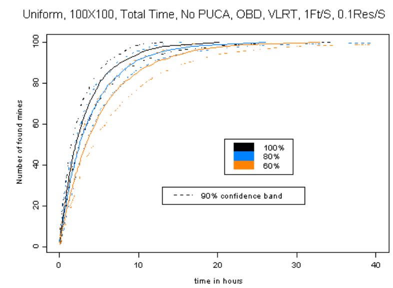


Figure 7: Plot of the number of found mines versus total search time by random search using different rate of detectors ($100 \times 100 ft^2$ environment with 25 obstacles).

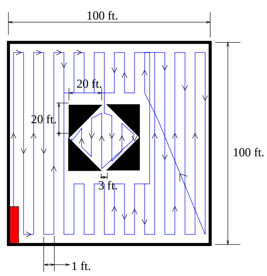


Figure 8: The search path by the coordinated search in a $100 \times 100 ft^2$ environment with 4 obstacles.

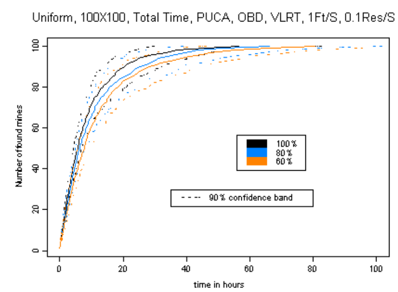


Figure 9: Plot of the number of found mines versus total search time by random search using different detection rates ($100 \times 100 ft^2$ environment with 4 obstacles).

around the chassis of the robot and is mounted to the top of the frame. Ten sensors are placed along the length of the robot. The front and back of the track each holds eight sensors. These sensors are positioned at twenty-two and a half degrees apart. We determined that this is the optimal configuration of sensors for a rectangular robot when the dimensions and the range of the sensors are considered. At completion, Slugger had dimensions of 88 cm (long) by 40cm (wide) by 40cm (high). It has two pneumatic wheels, two casters, twenty-six ultrasonic transducers, and two DC motors with built-in encoders. Three 12V, 12.0 Amp/hr batteries power the robot. See Figure 10 for the final assembly.

3.1 Computer Architecture

The obstacle sensors, motors, and localization are driven by a set of embedded computers on board Finder. A Pentium single-board computer running a custom Linux distribution provides high-level control of the robot, communicating via standard RS-232 serial lines with two Motorola 68HC16 slave microcontrollers. One microcontroller drives the sonar and buffers the distance-to-object values returned by the sonar board; the other handles low-level motor control and servoing. This other board also manages an external positioning device, described below. We chose the 68HC16's as motor controllers because we had previous experience using them (i.e., we did not have any special reason). The second 68HC16 interfaces with custom circuitry that drives 16 ultrasonic sensors multiplexed off of one TI Sonar Ranging Board. The high-level path planning resides on the embedded Pentium processor. This computer constantly receives sonar information from the sonar 68HC16 and sends "goto" commands to the motor controller 68HC16. Essentially, the 68HC16s are transparent to the programmer on the Pentium board.

Figure 10: The final assembly of the robots Finder and Slugger.

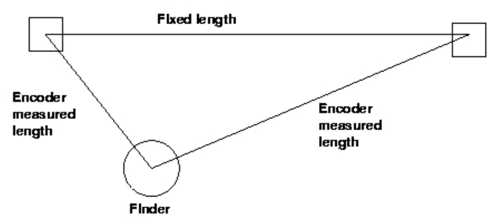


Figure 11: The linear encoder box and a top view of the geometric locations of the box and poles.

3.2 Positioning

Finder has a high precision, low-cost, and low-range external positioning device. Positioning is a perhaps the largest challenge facing mobile robotics. For the near-term, we have developed a temporary solution that involves linear encoders. Two strings wrapped around reels inside the robot are attached to two poles at given locations. Using simple trigonometry given the lengths of the strings and the known fixed distance between the two poles, the robot determines its location.

The disadvantage of this system is the error involved in the reliability of the strings wrapping around the reels. When covering large areas, error is created because the string used does not wrap exactly the same each time it is reeled in and out. However, this process can produce an accuracy of 0.3 cm in spaces up to 10 meters by 10 meters. This has been tested over a variety of surfaces, including being taken outside and tested on grass. In addition to the level of accuracy, the entire system can be implemented for less than $50, which is very inexpensive in comparison to other positioning methods such as GPS or DGPS. However, we are not suggesting that this linear encoder is the cure-all for all positioning problems. First, it assumes that all obstacles are lower than the height of the wire. Second, for distances greater than 10 meters, we experienced a stochastic process in which wire was wrapped and unwrapped on the reel allowing for inaccurate readings at 30 meters.

Instead of trying to perfect this technology, we opted to develop a vision system where the robot looks at engineered landmarks to determine its position. The benefit of this approach is that we can localize the robot at great distances, on the order of a 50 meter by 50 meter field, but the drawback is cost.

4 Conclusion

We presented two path planning approaches, sensor-based complete and probabilistic. Sensor-based coverage was based an exact cellular decomposition in terms of critical points. The robot executing the coverage algorithm incrementally constructs this cellular decomposition while it is covering the space with back and forth motions. Our algorithm that is presented in [Acar and Choset, 2000] guarantees complete coverage of unknown spaces.

For demining scenarios where time is limited and there exists *a priori* information about the minefield, we developed a probabilistic method that extracts the minefield parameters. Once these parameters were determined, the minefield layout was fixed. Then one can guide the robot opportunistically to decrease demining time.

The common path planning approach for demining is a random walk. However, we showed that a coordinated search out performs random strategies. Even though general belief is that coordinated strategies require high cost robot systems, our complete sensor-based algorithm requires a sensor suit that can guide the robot along the boundaries of the obstacles such as infrared rings, tactile sensors. Therefore, it leads to development of low cost robots.

We designed a mobile mechanism that might not be novel, but was a result of tradeoffs between cost, performance, and transportability. This had led us to the standard differential drive robot. We are currently developing smart casters and hope to use them to improve our performance. One measure of performance was the robot's ability to position itself, i.e., overcome dead-reckoning error. Ultimately, we would like to use our obstacle sensors (range sensors) to bypass this problem or in the future, some new GPS-like technology will be developed that will work in all sorts of conditions, at high resolution and at low cost, but today we are using two forms of a triangulation system. The first is the linear encoder that measures the length of fishing wire; although this may seem like a clumsy solution, it is incredibly accurate and inexpensive. Unfortunately, its range is limited, so we are moving to a visual system.

Acknowledgements. The authors gratefully acknowledge Chris Debolt and Tuan Nguyen at the Navy's EOD Technology Division, Douglas Gage formerly at SPAWAR, but now at DARPA, and Tom Swean at ONR for their suggestions and feedback towards this work. Finally, we would like to thank Alfred Rizzi at CMU and Jon Luntz at the University of Michigan for their help on this project.

References

E. Acar and H. Choset. Critical point sensing in unknown environments. In *Proc. of IEEE ICRA'00*, San Francisco, CA, 2000.

Z. L. Cao, Y. Huang, and E. Hall. Region filling operations with random obstacle avoidance for mobile robots. *Journal of Robotic systems*, pages 87–102, February 1988.

H. Choset, E. Acar, A. Rizzi, and J. Luntz. Exact cellular decompositions in terms of critical points of morse functions. In *Proc. of IEEE ICRA'00*, San Francisco, CA, 2000.

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

E. Gelenbe and Y. Cao. Autonomous search for mines. *European Journal of Operational Research*, 108:319–333, 1998.

S. Hert, S. Tiwari, and V. Lumelsky. A Terrain-Covering Algorithm for an AUV. *Autonomous Robots*, 3:91–119, 1996.

D. E. Lake and D. M. Keenan. Identifying minefields in clutter via collinearity and regularity detection. In A. C. Dubey, I. Cindrich, J. M. Ralston, and K. Rigano, editors, *Detection and Remediation Technologies for Mines and Minelike Targets*, volume 2496, pages 519–530, 1995.

J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.

V. J. Lumelsky, Snehasis Mukhopadhyay, and Kang Sun. Dynamic path planning in sensor-based terrain acquisition. *IEEE Transactions on Robotics and Automation*, 6(4):462–472, August 1990.

J. Milnor. *Morse Theory*. Princeton University Press, Princeton, New Jersey, 1963.

Nomad. *Nomad 200 User's Manual*. Nomadic Technologies, Inc., Mountain View, CA, 1996.

J. Trevelyan. Robots: A premature solution for the land mine problem. In Shirai Y. and Hirose S., editors, *Proc. of Eighth International Symposium on Robotics Research*, pages 382–390, 1998.

A. Zelinsky, R.A. Jarvis, J.C. Byrne, and S. Yuta. Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot. In *Proceedings of International Conference on Advanced Robotics*, pages pp533–538, Tokyo, Japan, November 1993.