Planning Efficient Paths through Dynamic Flow Fields in Real World Domains

Christopher Tomaszewski
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
christomaszewski@cmu.edu

Abhinav Valada
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
avalada@cmu.edu

Paul Scerri
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
pscerri@cs.cmu.edu

***Abstract*—This research addresses the problem of planning efficient paths for agents through flow fields in small real-world domains where vehicle dynamics and environmental uncertainty can significantly affect the optimality of a path. In particular, we consider the task of planning routes for small autonomous airboats deployed in various river domains so as to best take advantage of water currents to save energy and time. Existing planning techniques for flow fields were implemented on our airboat platform and evaluated on these domains with current models developed from the data gathered using a Nortek AD2CP-Glider acoustic doppler current profiler. The real-world performance of these algorithms were compared to theoretical estimates and several modifications are suggested to improve their performance in specific domains.**

I. Introduction

The problem of planning efficient routes for agents is of great importance across domains in mobile robotics, where energy is at a premium. In domains involving movement through fluids which themselves may be moving, such as flight through various wind patterns or navigation through river currents, it is understandably desirable for algorithms to harness the available kinetic energy in the environment to decrease a robots overall energy consumption. Successful implementation of such techniques in turn facilitates new robotic applications involving extended deployment range and duration (e.g. autonomous environmental monitoring of long stretches of rivers and coastal surveillance).

A. Prior Work

This particular domain has motivated much research in the past and many planning algorithms have been adapted or developed to solve this problem. In their work, Garau, Alvarez, and Oliver assume constant thrust navigation and use the A* algorithm to compute the optimal paths for a set of different current and eddy distributions [1]. Warren has proposed a method based on artificial potential fields, which is less susceptible to local minima but requires most of the global workspace to be known and does not account for variabilities in the flow [2]. A method using the forward evolution of level sets that can produce time optimal paths assuming a constant vehicle velocity through time dependent flow fields was developed by Lolla et al [3]. In their work on autonomous underwater gliders, Davis, Naomi, and Fratantoni develop a variational calculus approach for path planning through time invariant velocity fields of comparable magnitude to the operating velocities of long range gliders [4].

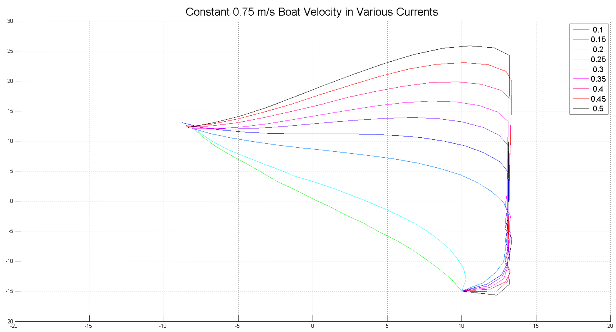
B. Motivation

Many of the proposed algorithms make simplifying assumptions regarding the vehicle dynamics and oftentimes operate with at least some model of the environment. Although this makes for efficient performance in simulation, in reality measuring the rate of a moving fluid while moving through that fluid is difficult and developing a detailed environmental model beforehand is impractical and not robust to unexpected events. Further difficulty comes from the inherent uncertainty of state estimation, as even small differences in a robots perceived state can yield very different optimal paths to the goal state, as evident in Figure 1 below. The plot on the left shows time-optimal paths for a boat travelling at a fixed velocity upstream in a river against parabolic current distribution of varying intensities. The plot on the right simillarly shows time-optimal paths for a boat travelling up river, however, this time the current distribution is fixed while the vehicle velocity is varied. These results help illustrate just how sensitive optimal paths can be to variations or uncertainty in the environment, and they offer compelling motivation for analysis of the path planning algorithms o the theoretical realm and in the field.

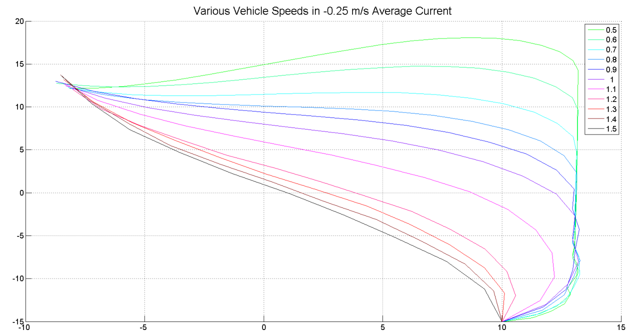
This paper describes the ongoing work to adapt and deploy existing planning techniques on a low-cost autonomous airboat platform for the purpose of experimentally verifying their theoretical performance and improving them. In order to limit the scope of the experiments to be done, this research emphasizes the analysis of path planning techniques for use in different river domains. The relative size of the domains under scrutiny with respect to these vehicles is such that the dynamics of the agents have significant bearing on their optimal paths and can no longer be assumed negligible when path planning. The following section provides a more detailed discussion of the airboat platform, our river domain simulation representation, and level set evolution techniques for path planning.

II. Methodology

The planning algorithms under research were adapted for and implemented on the Cooperative Robotic Watercraft (CRW) platform developed at the Robotics Institute of Carnegie Mellon University. In order to provide feedback on the current distribution in the river, the CRW platform is equipped with the Nortek AD2CP-Glider acoustic doppler current profiler.



(a) Fixed Velocity, Varying Currents



(b) Fixed Currents, Varying Velocities

Fig. 1: Optimal Paths Generated for Various Current Distributions and Vehicle Velocities

A. Testing Platform

The testing platform chosen for this research is the Lutra 1.0 series of the Cooperative Robotic Watercraft system developed at the Robotics Institute of Carnegie Mellon University. This low-cost platform with limited thrust and a slim onboard energy storage budget exemplifies systems which could benefit greatly from the use of intelligent path planning algorithms to extend their operating capabilities. The hull of the platform, shown in figure 2 is built using vaccuum-formed ABS, filled with expanding polyurethane foam. The shallow hull design enables the airboat to cruise even on very shallow, debris-filled waters. The high maneuverability of the CRW system can be attributed to the unique propulsion system. The propulsion system is composed of a ducted propeller assembly built using high impact acrylic and pvc. The propeller attached to a brushless motor and the entire propulsion assembly is actuated by a waterproof servo motor. The hull of the airboat has a built in electronics compartment which is then sealed from the top using a deck plate-gasket assembly. The deck plate is equipped with a phone mount which houses an Andriod smartphone and a custom made embedded computing board. The entire system is powered by two 8000mA lithium polymer batteries wired in parallel. All the algorithms are built into a custom app which runs on the Android smartphone, which also provides integrated intertial sensors and GPS for navigation.

A Nortek AD2CP current profiler is interfaced to the CRW system though the sensor ports on the computing board. The profiler has five 1 MHz transducers, a pressure sensor and a temperature sensor. The profiler is itself powered using an internal 7500mAh alkaline battery pack. The Nortek AD2CP can be used for obtaining absolute velocity profiles of the water. The velocity measurements provided by the profiler can be further processed to obtain depth averaged velocity if required [5].

These characteristics of the CRW system make it an ideal candidate for the deployment of the efficient algorithms this research seeks to produce; as a low-cost autonomous monitoring system, the CRW platform carries a limited energy supply, making it desirable to optimize navigation and sampling procedures as to maximize the information that can be gathered before recharging is necessary. A more detailed treatment of the platform can be found in our previous publications [6] and [7].

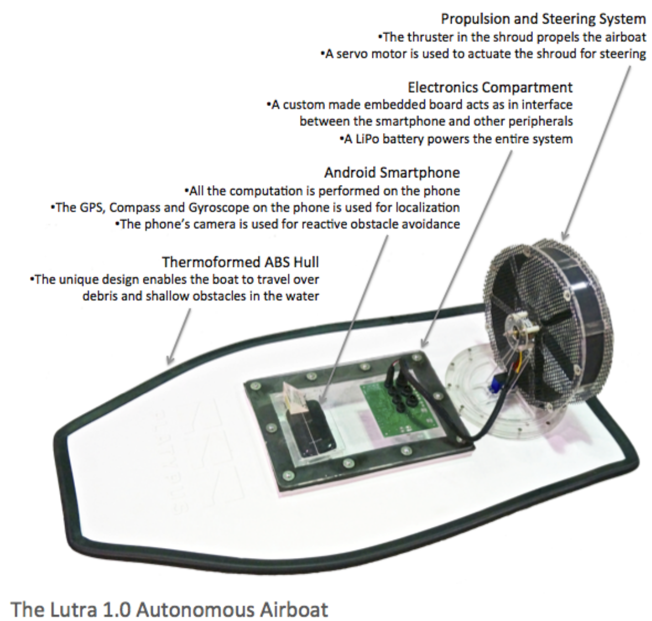


Fig. 2: A Complete Lutra 1.0 Series Autonomous Airboat

In order to take into account the vehicle dynamics, it was necessary to develop a rough dynamic model for the system. The airboats were measured under controlled conditions in the lab and deployed at a nearby lake in order to characterize the system parameters. The resulting dynamic model is of the form show below:

$$m\ddot{\vec{x}} = -\vec{b}\dot{\vec{x}}^2 + \vec{F}$$
$$J\ddot{\theta} = -b_\theta\dot{\theta}^2 + ||F||L\sin\phi$$

where m is the mass of the vehicle, $\vec{b}$ is the vector of linear drag coefficients, $\vec{F}$ is the force vector applied by the propulsion assembly, J is the moment of inertia of the vehicle, b_θ is the torsional drag coefficient, L is the moment arm of the vehicle, and ϕ is the angle of the propulsion assembly.

B. Planning Algorithms

The level set method described by Lolla et al. in their paper is a two phase algorithm where the first part involves propagating an interface (denoted as the zero level set of the signed distance function ϕ) forward in time, and the second part entails tracking a particle back from the goal state along the normal vectors to the intermediate interfaces [3]. The interface being tracked actually represents the farthest set of points reachable by the vehicle at any point in time, under some fairly strict assumptions. The process modeling the evolution of this interface is constructed as a Hamilton Jacobi partial differential equation, which is then solved discretely over a grid. The general form of this PDE is given below.

$$\begin{aligned} 0 = &\frac{\delta\phi(\vec{x},t)}{\delta t} && (1)\\ &+ F|\nabla\phi(\vec{x},t)| && (2)\\ &+ V(\vec{x},t)\cdot\nabla\phi(\vec{x},t) && (3)\\ &- b(\vec{x},t)\kappa(\vec{x},t)|\nabla\phi(\vec{x},t)| && (4)\\ &+ \lambda(\vec{x},t)\phi(\vec{x},t) && (5)\\ &+ H(\vec{x},t,\phi(\vec{x},t),\nabla\phi(\vec{x},t)) && (6) \end{aligned}$$

Lines 1 and 6 make up the general form of the hamilton jacobian partial differential equation, however, these general form equations are typically more difficult to solve. Therefore, there are a few terms that are sometimes added, which are specifically constrained and can be more easily solved. In their algorithm, Lolla et al use only the terms on lines 1, 2, and 3 to represent the farthest points reachable by some vehicle in a given amount of time. Line 2 represents motion normal to the level set at a constant speed F, while line 3 represents motion due to the external velocity field. In using this representation, Lolla et al make the assumption that the vehicle is always traveling at speed F normal to the level sets. Furthermore, their method relies on having exact knowledge of the velocity field V and how it will change over time. Clearly these are not the best assumptions to make for real world implementation, which further motivates our objective of producing an algorithm that calculates paths more true to the dynamics of the vehicle.

After the interface has evolved far enough that the goal state is within the signed distance function, the forward evolution terminates and a particle is traced back along the normals to the intermediate instances of the tracked interface. Lolla et al prove in their work that if a vehicle always travels at speed F in a normal direction to the set of its currently reachable states, and the external velocity field $V(\vec{x},t)$ is known perfectly for all times, then the path computed by this algorithm is time-optimal.

In our level set experiments, we made use of Ian Mitchell's Matlab Level Sets Toolbox, which implements several partial differential equation solvers [8]. For consistency we chose to use a single domain for all of our simulations - a river 40 meters across, with a current distribution modelled after fully developed pipe flow. Figure 3 illustrates our simulation domain.

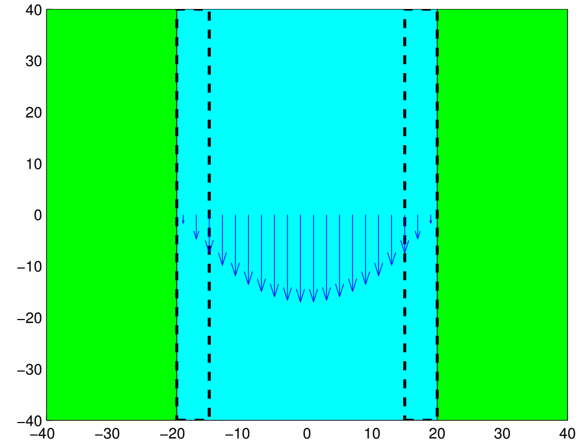


Fig. 3: Simulated River Domain

The green borders on either side of the domain represent the banks of the river and are treated as forbidden regions during planning. Two additional forbidden regions, denoted with dashed black lines in the domain diagram, were added running five meters out from each bank as a precaution to keep boats from getting stuck in shallow water when the algorithms are deployed on boats in the field. Initially the method of forbidden regions described by Lolla et al in their work was implemented, where the external velocity field is zeroed out and the normal velocity of the vehicle is held at zero within these regions. Unfortunately when such a discontinuous change in velocities was introduced, certain scenarios would experience numerical instabilities while solving the partial differential equations; after the level set evolution reached the discontinuity, the interface would sometimes develop unstable distortions that would magnify during propagation and produce inaccurate paths.

In order to remedy this issue, the external velocity field was treated as a cost function and its lower dimensionality was exploited to eliminate the discontinuities while retaining the guarantee that the optimal path will never lie within the forbidden regions. Since our domain description assumes the river has the same parabolic current distribution along its course, we can define the cost function to be proportional to the velocity of the river. When the goal state is upstream from the vehicle, the cost function is postive because the agent must fight against it and when the goal is downstream the cost function is negative because the boat is moved towards the goal by the current. Using this definition, we can exploit the lower dimensionality of the cost function (i.e. the fact that for a given X coordinate, the cost function does not vary as you move up or down stream) and apply the theorems described by Vernaza and Lee [9]. They prove that when a cost function does not vary in some dimension, then the optimal path to the goal will never move away from the goal in that dimension. Therefore, it must be that in our particular domain, the boat will never head downstream when the goal is upstream and vice versa. If we consider the forbidden regions again, and simply clamp the value of the external velocity field in these regions to the value at the boundary of the region, our cost function no longer varies in the X direction in these regions and therefore the optimal path will never head away from the

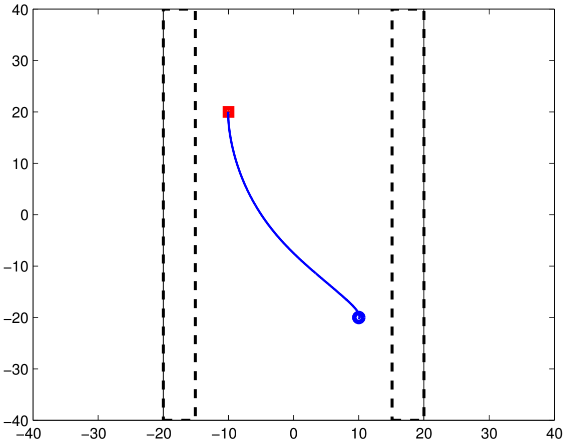


Fig. 4: Path of Boat Aiming Straight at the Goal

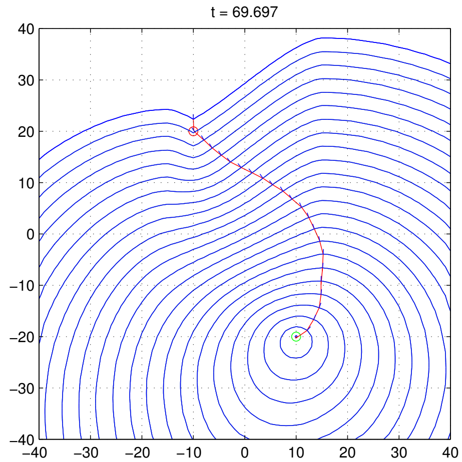


Fig. 5: Path Planning Process using Level Set Evolution

goal along the X axis in these regions as well. If we define the goal state as always lying outside of a forbidden region, this guarantees that the optimal path will never cause the boat to head into the forbidden regions, which we defined on the banks of the river. With just this small modification to the external velocity field, the discontinuities are dealt with and the level sets can be evolved to generate smooth and optimal paths.

III. Results

Using the simulated river domain described in the previous section, we were able to examine a variety of scenarios and test the efficiency of several path planning and navigational strategies. In the following subsections, we present several solutions to one planning scenario and evaluate their performance. The scenario shown involves planning a path for a vehicle initially at the point $(10, -20)$ to the goal position, $(-10, 20)$. The maximum thrust of the vehicle is limited to 3 Newtons and the average velocity of the river current is 0.25 m/s downstream (i.e. along the negative Y axis).

A. Point and Shoot Heading Controller

For a simple baseline with which to compare the performance of the level set planner previously described, we implemented and evaluated a Proportional-Derivative Heading Controller that would consistently attempt to aim the front of the vehicle towards the goal. In our implementation we allow for control of the propulsion assembly angle, but fix the thrust at 3 Newtons and use an ordinary differential equation solver to compute the motion of the craft. After starting at rest and facing upstream, the controller guided the boat straight across the river, with no regard for the fast flowing currents, as shown in figure 4.

B. Heading Controller Tracking a Precomputed Path

In order to use the level set planner to compute a time optimal path for this scenario, we made the assumption that the boat would be able to maintain a constant velocity while

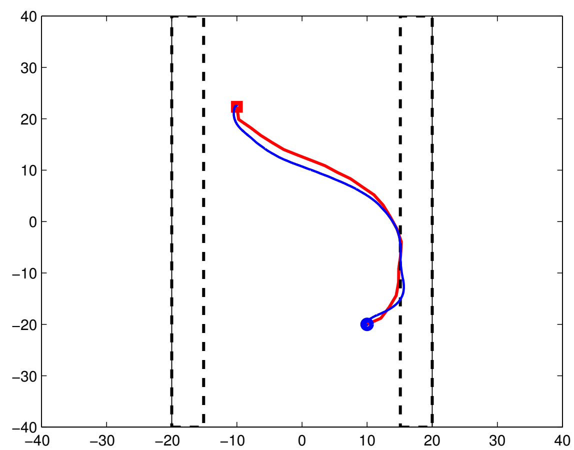


Fig. 6: Trajectory of Simulated Vehicle Tracking Path

moving. In order to compare energy usage estimates with those of other methods we chose to use a constant velocity corresponding to slightly below the terminal velocity for our platform when 3 Newtons of force are applied. Figure 5 shows the path computed by the level set planner for this scenario and under these assumptions. Next we developed another controller to attempt to guide a boat along the planned path to the goal at a constant 3 Newtons of thrust. Figure 6 illustrates the performance of this controller; the red path is the orignally planned path while the blue path is the path actually travelled by the simulated vessel.

C. Heading Controller Following a Dynamic Path

Another path planning solution we developed and tested begins like the previously described method that precomputes an optimal path with level sets and makes use of a heading controller to track this trajectory. After a specified time period (here we use 5 seconds), however, this modified planner will recompute the optimal plan based on the updated state of the

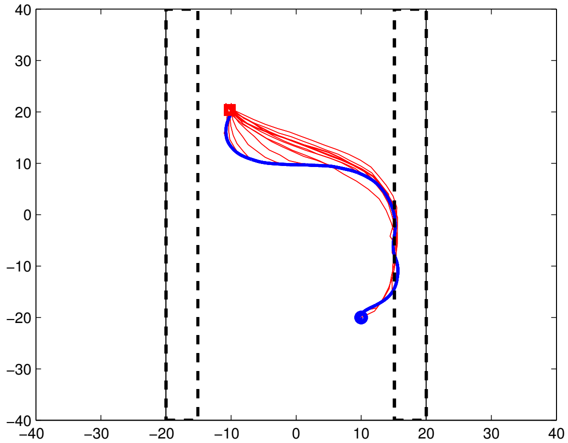


Fig. 7: Trajectory of Simulated Agent with Intermediate Paths

vehicle and the freshest environmental data available. When the controller recieves an updated trajectory to track, it throws away its outdated plan and continues tracking towards the goal. Unlike the previous method of precomputing a path and following it, this path planning technique can adapt to sudden current variations, environmental uncertainty, and even failures of a car to track the planned path perfectly. Figure 7 shows the path of the autonomous agent in blue, and all the intermediate paths computed during operation in red.

IV. Conclusion

For each of the solutions described to the scenario in the previous section, we collected some performance metrics and provide them for comparison in table I. Each row in this table corresponds to a different solution and includes the time the path took to traverse, the distance travelled along the route as seen by a stationary observer on shore, and an estimate of the energy consumed while driving the path. After examining the results, there is a clear advantage going with the path precomputed by the standard level set planner, as long as the platform for which we are planning can manage to successfully track the precomputed route. From the table we see that the attempt to track the single precomputed path ended up using more time and more energy than any other solution. The iteratively replanning technique will adjust appropriately after the next timestep if it detects that the craft is failing to track the trajectory well or maintain the desired speed. In fact, the replanning method even performs well when compared to the simple point and shoot heading controller, especially considering the additional complexity and turns on the dynamic replanning route. Additional scenarios will be examined to further investigate these findings and improve upon these techniques.

A. Future Work

Due to unforeseen manufacturing delays, the Nortek AD2CP acoustic doppler unit was not ready to have data from field tests included with this paper. Currently field experiments are scheduled to begin in mid August when we will attempt to

TABLE I: Performance Metrics

	Time (s)	Distance (m)	Energy (J)
Point and Shoot Path	84.9757	48.3823	9432.3027
Precomputed Path	69.697	56.1515	7736.367
Tracking of Precomputed Path	89.0769	58.5308	9887.5359
Tracking of Dynamically Replanned Paths	85	61.5848	9435

validate the results and models obtained thusfar in simulation. Data from these field tests and ongoing simulation work will be presented at the conference.

Further in the future, it would be interesting to examine how changing the speed of the agent can affect the optimality of the route, and investigate how to plan where to slow down or speed up in order to get the best time or lowest energy consumption overall.

Acknowledgment

The authors would like to thank Nortek for their generous support in the form of an equipment loan and grant. The AD2CP acoustic doppler unit they provided continues to play an integral part to this research, allowing us to investigate the performance of the algorithms we are studying in real world domains.

References

[1] B. Garau, A. Alvarez, and G. Oliver, "Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an a* approach," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 194–198.

[2] C. Warren, "A technique for autonomous underwater vehicle route planning," *Oceanic Engineering, IEEE Journal of*, vol. 15, no. 3, pp. 199–204, 1990.

[3] T. Lolla, M. P. Ueckermann, K. Yigit, J. Haley, P.J., and P. F. J. Lermusiaux, "Path planning in time dependent flow fields using level set methods," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 166–173.

[4] R. E. Davis, N. E. Leonard, and D. M. Fratantoni, "Routing strategies for underwater gliders," *Deep-Sea Research Part Ii-Topical Studies in Oceanography*, vol. 56, no. 3-5, pp. 173–187, Feb 2009, n/a. [Online]. Available: ¡Go to ISI¿://WOS:000266543600009

[5] P. Rusello, C. Yahnker, and M. Morris, "Improving depth averaged velocity measurements from seaglider with an advanced acoustic current profiler, the nortek ad2cp-glider," in *Oceans, 2012*, 2012, pp. 1–8.

[6] P. Scerri, P. Velagapudi, B. Kannan, A. Valada, C. Tomaszewski, J. M. Dolan, A. Scerri, K. S. Shankar, L. L. Bill-Clark, and G. A. Kantor, "Real-world testing of a multi-robot team," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, P. Conitzer, Winikoff and van der Hoek, Eds. www.ifaamas.org: International Foundation for Autonomous Agents and Multiagent Systems, June 2012.

[7] A. Valada, P. Velagapudi, B. Kannan, C. Tomaszewski, G. A. Kantor, and P. Scerri, "Development of a low cost multi-robot autonomous marine surface platform," in *The 8th International Conference on Field and Service Robotics (FSR 2012)*, July 2012.

[8] I. Mitchell, "The flexible, extensible and efficient toolbox of level set methods," *Journal of Scientific Computing*, vol. 35, no. 2-3, pp. 300–329, 2008. [Online]. Available: http://dx.doi.org/10.1007/s10915-007-9174-4

[9] P. Vernaza and D. D. Lee, "Learning and exploiting low-dimensional structure for efficient holonomic motion planning in high-dimensional spaces," *The International Journal of Robotics Research*, vol. 31, no. 14, pp. 1739–1760, 2012. [Online]. Available: http://ijr.sagepub.com/content/31/14/1739.abstract