Experimental Analysis of Receding Horizon Planning Algorithms for Marine Monitoring

Soo-Hyun Yoo, Andrew Stuntz, Yawei Zhang, Robert Rothschild, Geoffrey A. Hollinger and Ryan N. Smith

Abstract Autonomous surface vehicles (ASVs) are becoming more widely used in environmental monitoring applications. Due to the limited duration of these vehicles, algorithms need to be developed to save energy and maximize monitoring efficiency. This paper compares receding horizon path planning models for their effectiveness at collecting usable data in an aquatic environment. An adaptive receding horizon approach is used to plan ASV paths to collect data. A problem that often troubles conventional receding horizon algorithms is the path planner becoming trapped at local optima. Our proposed Jumping Horizon (J-Horizon) algorithm planner improves on the conventional receding horizon algorithm by jumping out of local optima. We demonstrate that the J-Horizon algorithm collects data more efficiently than commonly used lawnmower patterns, and we provide a proof-of-concept field implementation on an ASV with a temperature monitoring task in a lake.

S.-H. Yoo (✉) · Y. Zhang · G.A. Hollinger
Oregon State University, Corvallis, OR, USA
e-mail: yoos@onid.oregonstate.edu

Y. Zhang
e-mail: zhanyawe@onid.oregonstate.edu

G.A. Hollinger
e-mail: geoff.hollinger@oregonstate.edu

A. Stuntz · R. Rothschild · R.N. Smith
Fort Lewis College, Durango, CO, USA
e-mail: abstuntz@fortlewis.edu

R. Rothschild
e-mail: rarothchild@fortlewis.edu

R.N. Smith
e-mail: rnsmith@fortlewis.edu

© Springer International Publishing Switzerland 2016

D.S. Wettergreen and T.D. Barfoot (eds.), *Field and Service Robotics*,
Springer Tracts in Advanced Robotics 113, DOI 10.1007/978-3-319-27702-8_3

1 Introduction

Autonomous surface vehicles (ASVs) are becoming more commonly used to collect data in oceans and inland waterways using instruments such as: acoustic doppler current profilers (ADCPs); conductivity, temperature, and depth sensors (CTDs); and sidescanning sonars. These autonomous vehicles allow data collection in tight places, such as in and around glaciers or ice, as well as in close proximity to land (e.g., around river deltas) [2, 5].

Commercially-available ASVs, such as the Platypus Lutra (Fig. 1b) and Ocean-Sever Q-Boat, typically execute a simple lawnmower path to cover the area to be explored (Fig. 2). Such a path can provide high data yield, but at the expense of substantial fuel and time costs [11].

Previous work has shown that sampling in a spiral pattern is slightly more energy-efficient than doing so in a lawnmower pattern [9], but only by a margin of less than 5 %. This margin will be shown to be negligible compared to that demonstrated by J-Horizon over lawnmower, so for simplicity, the lawnmower pattern will be simulated and considered as the baseline.

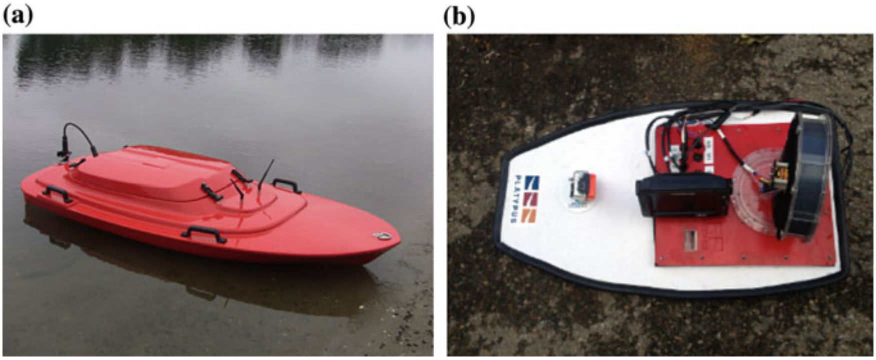
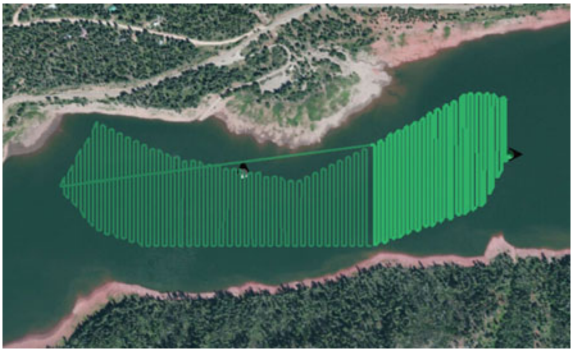


Fig. 1 Two commercially-available autonomous surface vehicles for aquatic sampling. **a** Q-Boat 1800P with an integrated ADCP. **b** Platypus Lutra with a dissolved oxygen and pH sensor

Fig. 2 The proprietary area search algorithm from Platypus generates a dense lawnmower pattern that is highly energy-inefficient



Here, we propose a receding horizon path-planning algorithm that, given an information or uncertainty map, generates a sampling path to maximize the information gathered or reduce the uncertainty. We compare this algorithm against the simple lawnmower path planner for a given transport budget and examine the effects of various algorithm parameters on the quality of the generated path. Furthermore, we propose a Jumping Horizon (J-Horizon) algorithm that improves on the conventional receding horizon algorithm by varying the look ahead step size if desired threshold values cannot be found within the current horizon. This allows the planner to "jump" out of local optima if higher peaks can be found elsewhere on the map. Finally, we validate our simulated results during a field trial using an ASV. An initial data set is collected to provide a base scalar field. The J-Horizon algorithm is then run over this scalar field produced, and a qualitative analysis is given. The J-Horizon planner is able to produce paths superior to the simple lawnmower pattern in simulation, and experimentally the J-Horizon path is shown to cover more area and generate a more representative scalar field.

2 Related Work

Past work has shown that a receding horizon path planner is effective at optimizing paths in "no-fly" zone environments with hard constraints [10], where the agent is prohibited from entering certain areas bounded by walls. This is a useful constraint for aerial and land vehicles that must navigate cluttered environments. However, these constraints do not apply to an ASV that must cover a large body of water such as a lake or the open ocean.

In previous work, AUVs have played a similar role as the ASV in our project. Binney et al. [1] describe an offline path planner for an uncertainty area. Hollinger and Singh [7] describe an approach for multiple agents searching for a target in a known environment. Hitz et al. [6] discuss a path planner that can choose an efficient path for measurement of fluorescent bacteria in the ocean using an ASV. To reduce computational complexity, all of these authors employ a receding horizon path planner. Besides implementation on ASVs, receding horizon algorithms are widely used in other robotics scenarios. Tisdale et al. [12] describe a receding horizon path planner for multiple unmanned aerial vehicles to search for a stationary object. For currently implemented receding horizon planners, no research exists that examines the effect of the horizon length, or the possibility of modifying this horizon based on the remaining information.

Frolov et al. [3] compares lawnmower paths to other planning algorithms using fleets of research vehicles. They come to the conclusion that lawnmower paths are only marginally worse than adaptive algorithms. They also conclude that graph-based search algorithms are actually worse than lawnmower patterns, thus cannot maximize their performance, because they are unable to adapt to prior uncertainty. Our J-Horizon algorithm adapts to the environment and removes these limitations to provide improved performance over other adaptive algorithms.

Gotovos et al. [4] propose a Level Set Estimation (LSE) algorithm that uses Gaussian Processes to estimate level sets of measured quantities and generate sampling points that reduce uncertainty around a certain threshold level. In a different context, [8] describe an incremental sampling-based motion planning algorithm. Instead of reducing the uncertainty, they try to optimize the information gathering, depreciating the information value of sampled points.

A key limitation of existing research in receding horizon planning is that none of the aforementioned works discusses the role of the parameters in the receding horizon algorithm, e.g., horizon length or adaptation based on gathered or remaining information. In addition, prior research has not focused on a single ASV performing data collection over large areas. In this paper, we address this gap in research in the aforementioned papers through the presentation of the J-Horizon algorithm. We present the application of our proposed method over different scalar fields both in simulation and through field experiments. The algorithm's performance is experimentally demonstrated to outperform existing lawnmower and traditional receding horizon methods.

3 Problem Setup

Due to the wide variety of data that is sampled, it is challenging to model the data collection in a general way. The areas of interest being surveyed by ASVs are often dynamic environments, and the data collected is often a reflection of changes in the environment. Data collection around glaciers, in river deltas, or in relatively shallow water are environments that are changing quickly. The data collected is often collected to provide a *snapshot* of the processes that are evolving in the general area, and plan for future targeted sampling. Prior surveys can provide a heuristic upon which to formulate plans for future surveys, and multiple surveys can be combined to provide a time-series evolution of the region of interest. Here, we exploit the existence of a partially known underlying field and present a method for improved sampling based on time and energy optimization while gathering data of maximal reward.

3.1 Objective Function

In this paper, the J-Horizon planner addresses the following maximization problem:

$$p^* = \arg\max_{p \in \psi} \; R(p) \quad \text{s.t.} \quad c(p) \leq B,$$

where ψ is the space of possible trajectories for the ASV, B is the initial budget (e.g., time, fuel), and R is a reward function that represents the information gathered or uncertainty reduced along the trajectory p.

Furthermore, we depreciate the value of $R(p)$ each time we sample p. That is, for intersecting partial trajectories p_a and p_b (i.e., $p_{a\cap b} \neq \emptyset$),

$$R(p_{a\cup b}) + R(p_{a\cap b}) \leq R(p_a) + R(p_b),$$

where $p_{a\cup b}$ and $p_{a\cap b}$ are the union and intersection of p_a and p_b, respectively. This makes the objective function submodular.

3.2 Experimental Setup

We first present a simulation setup that uses computer-generated scalar fields to compare the performance of J-Horizon, a conventional receding horizon, and the lawnmower planning algorithms. We then present a real-world dataset acquired from Lake Haviland outside of Durango, CO to generate a path maximizing gathered information for a given transport budget.

3.2.1 Simulation

The J-Horizon algorithm is most effective when there is a prior dataset that can be used to generate an information map. The reward function is then specified by the maximum amount of new information that could be gathered at a map location. Furthermore, the algorithm improves upon the conventional receding horizon algorithm by seeking out areas of high reward when the local map area has been exhausted of new information, resulting in its “jumping” behavior.

For our simulated testing, a MatLab script was used to randomly generate 2960 different scalar fields with varying numbers and distributions of high-reward peaks. Between 5 and 50 such peaks were randomly generated on each map with a reward value that decays as a function of distance from the peak center. One such field is visualized in Figs. 4 and 5 as contour maps.

The total reward accumulated along a path generated by J-Horizon for a given fuel budget was averaged for these scalar fields. This performance was compared with that of a lawnmower exploration pattern on the same datasets and fuel budget.

3.2.2 Hardware

A Platypus Lutra ASV was used to take physical samples from Lake Haviland. This ASV is fan-powered, maneuverable and capable of sampling data in lakes or other small bodies of water. This small ASV is an ideal platform upon which to implement our algorithm, based on the limited deployment duration and sensing capabilities for relatively large bodies of water. The ASV is capable of simultaneously sampling temperature, conductivity, pH, and Dissolved Oxygen. Additionally, it measures depth and has a side-scan sonar. The latter sensors allow for bottom mapping of the lake.

The Platypus Lutra ASV has non-holonomic constraints that limits its ability to execute some of the sharper turns produced by the J-Horizon algorithm. Thus, due to hardware limitations, it is necessary to modify the path produced by J-Horizon. These modifications allow the ASV to follow the planned path. Due to the limited locomotion of the Platypus Lutra as well as a need to simplify data collection, some assumptions have to be made:

1. The ASV is limited in its motion and has non-holonomic turning constraints.
2. That sampled scalar fields were not dramatically changing over time.
3. Distance traveled equates to using a linear and constant amount of energy.
4. Additional data sampling points at a given location correlates to better quality data.

4 Algorithm Design

We seek to maximize the reward function for a given transport budget. In reality, this budget is a combination of fuel expenditure, time, and distance, each of which are specific to the vehicle and data collection scheme in use. For simplicity, we assume these factors are linearly related and that acceleration (e.g., due to turning, data collection) has zero cost. In addition, we enhance the conventional receding horizon algorithm by increasing the look-ahead step size if none of the predicted future states satisfy a reward threshold, allowing the planner to “jump” out of low-information areas. This makes J-Horizon especially effective when the input scalar field has high local variability.

The sequence of potential future steps, as well as the final generated path, are stored in a tree wherein each node stores the state of the ASV, which consists of the cumulative reward value of the path, remaining budget, and the location of the ASV. Each look-ahead step recursively generates a number of possible future states. Of these, the best branch is chosen, and the rest are pruned. The sequence of nodes remaining after the remaining budget reaches zero is considered the optimal path. The lawnmower and J-Horizon algorithms share the same functions to calculate the information available at a map location and to depreciate the available information after sampling that location.

The following sections describe the J-Horizon implementation shown in Algorithms 1, 2, and 3 by their respective line numbers.

4.1 Algorithm 1—Main

Path planning begins with the specified transport budget B and loops over the following four steps until either the budget is expended or the planner covers the prescribed area.

6: From the current state σ, take L look-ahead steps with LOOK-AHEAD. This updates the path tree with possible future states L levels below the current node.

7: Find the location of the "best" adjacent node that will achieve the highest reward at the end of L steps through that node.

8: Prune the path tree of all descendants under the current node.

9: Add sample point nodes between current and best locations and update the current node to the latest node.

4.2 Algorithm 2—Look-ahead

Given an initial state σ and maximum recursion depth d, we recursively generate and add possible future states to the path tree. Each step is taken with a new, temporary copy of all data. During each call, it performs the following:

1: Generate set of future states S_f from σ.

3: Remove a fraction $R \sim U([0, 1))$ of the states (but not all) in S_f.

7: Recurse on each descendant node.

4.3 Algorithm 3—J-Horizon

Given a state σ and an information threshold t, probe outwards from the given location and update the map:

2: Start with a sample interval of D.

3: Calculate number of future states to generate b per some factor F.

4: While S_f is empty, perform the following:

> 5: Generate b equally spaced points around a circle of radius D around σ.
>
> 6: For each such point, if the quality of the map at that point exceeds t, then add the point to S_f.
>
> 8–9: Increment δ by D and update b.

10: Update the information map.

Algorithm 1: MAIN Main function to run

```
$loc \leftarrow (0, 0)$
$reward \leftarrow 0$
$budget \leftarrow B$
$\sigma \leftarrow \{loc, reward, budget\}$
while $\sigma.budget > 0$ do
    look-ahead($\sigma$, $L$)
    $l_b \leftarrow$ FindBest($\sigma$)
    Prune($\sigma$)
    $\sigma \leftarrow$ AddSamples($\sigma.loc$, $l_b$)
return
```

Algorithm 2: LOOK- AHEAD Recursively look several steps ahead

```
Input: State $\sigma$, recursion depth $d$
Output: Number of future states from location
$S_f$ = JHorizon($\sigma$, $T$)
for $i \leftarrow 1$ to $\lfloor R \cdot |S_f| \rfloor$ do
    RemoveRandom($S_f$)
$n \leftarrow 0$
foreach $\{\sigma_f \in S_f\}$ do
    if $R > 1$ then
        $n \leftarrow n + 1 +$ look-ahead($\sigma_f$, $d - 1$)
return $n$
```

Algorithm 3: JHORIZON Generate frontier locations

```
Input: State $\sigma$, threshold $t$
Output: Set of possible future states to explore
$S_f \leftarrow \emptyset$
$\delta \leftarrow D$
$b \leftarrow F\sqrt{\delta}$
while $|S_f| = 0$ do
    for $\sigma_m \in$ GenerateNew($\sigma$, $\delta$, $b$) do
        if GetInfo($\sigma_m$) $> t$ then
            $S_f \leftarrow S_f \cup \{\sigma_m\}$
    $\delta \leftarrow \delta + D$
    $b \leftarrow F\sqrt{\delta}$
Depreciate($\sigma$)
return $S_f$
```

5 Results

In this section, we present the results of application of the J-Horizon algorithm on simulated data, as well as data collected during a field trial.

5.1 Simulation Results

We validate the J-Horizon planner using 2960 simulated scalar fields. We compare the quality of the paths generated by the J-Horizon planner to that of a simple lawnmower pattern using the sum of the data collected over the path given the same transport budget as the metric. For the purpose of comparison, the information gathered was arbitrarily selected to correspond to the deviation from a particular target value of the quantity of interest (Fig. 3). By identifying pertinent or interesting data, the algorithm is able to successfully maximize data collection for a given deployment region. Such quantities are representative of the uncertainty in temperature or another physical parameter that can be approximated by a scalar field.

The simulated vector field seen in Fig. 3 is representative of many types of scalar data. One of the benefits of the J-Horizon algorithm is that it can plan across any type of scalar field, e.g., temperature, humidity, pressure, salinity or dissolved oxygen. The point being that the user may specify the data being looked for and J-Horizon will attempt to maximize the data collection. Figure 3 is an example of the J-Horizon planning over a simulated scalar field.

Figures 4, 5a, and 5b show a typical lawnmower, receding horizon, and J-Horizon path, respectively, planned over a simulated scalar field. The same transport budget was used for all three paths, yet the quality of the paths were 132, 189, and 420, respectively. The J-Horizon planner outperforms lawnmower by a factor of 3.18. Lawnmower required 3.51 times the transport budget to achieve the same reward on

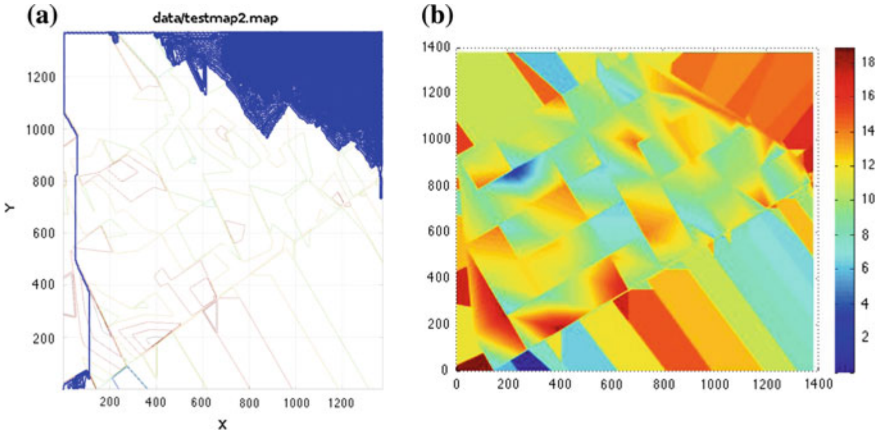


Fig. 3 J-Horizon path planned over simulated scalar field. **a** Dense path planned over reward field. **b** Field reward level map. *Red* indicates high reward

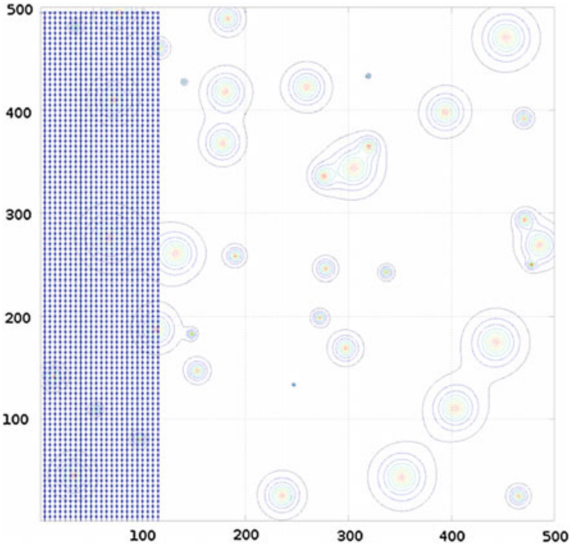


Fig. 4 Lawnmower path on generated scalar field. Reward of 132

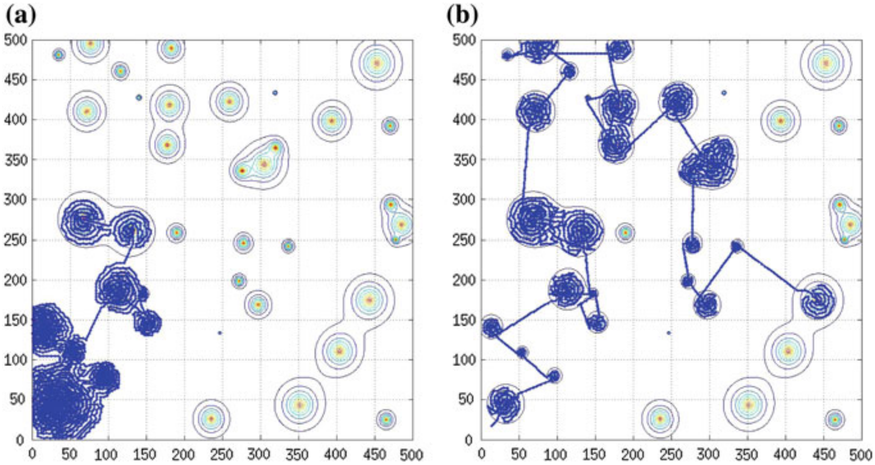


Fig. 5 Effect of reward threshold on J-Horizon jumping behavior. **a** Path generated by receding horizon planner. Note the path lingers in the high-reward area at the *lower left* for a long time before moving on to more worthwhile areas. Reward of 189. **b** Path generated by J-Horizon planner with threshold of 0.1. Once the peak at the *lower-left* has been exhausted of potential reward, the planner quickly moves on to other points of interest. Reward of 420

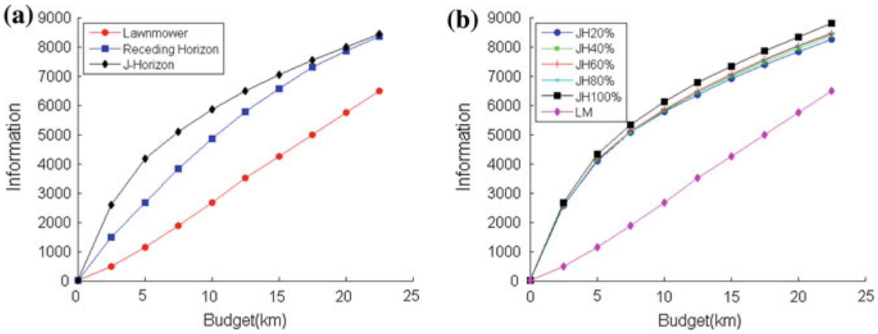


Fig. 6 Performance comparison between lawnmower, receding horizon, and J-Horizon algorithms. **a** Information gathered with increasing budget. **b** The information gathering ability of the J-Horizon planner against a lawnmower pattern as we decrease the fraction of generated future states

the same map and still fails to bring the maximum uncertainty below the threshold of 0.1.

Figure 6 demonstrates the general behavior of the J-Horizon planner, which outperforms the lawnmower planner by a factor of 4 for the first 5 Km of the 25 Km total budget and slows down as it explores the remaining, lower-reward regions of the scalar field.

Figure 6a compares the information gathering ability of the three algorithms with increasing budget. J-Horizon gathers information most rapidly. When the budget is large enough, the information gathered by the receding horizon planner is nearly equivalent to that of the J-Horizon planner.

Figure 6b shows the information gathering ability of the J-Horizon planner against a lawnmower pattern as we decrease the fraction of generated future states. For example, JH80 % indicates the algorithm generates 80 % of the usual number of future states. Even at JH20 %, J-Horizon significantly outperforms a lawnmower path. This suggests it is possible to drastically reduce computational complexity with only a minor performance penalty.

One of the most advantageous qualities of this planner is that it is not limited to any particular search space. It is capable of planning paths over anything that can be estimated by a scalar field.

5.2 Experimental Results

Here, we present results from field trials for the implementation of the J-Horizon planner over a scalar field of surface temperature in a small lake in Colorado.[1] Specif-

[1] The specific location of the field trials is Lake Haviland, outside of Durango, CO, located at 37°31′55″N 107°48′27″W.

ically, the goal presented to the ASV is to focus sampling at low-temperature regions, which correspond to high information reward in this case. We use the Platypus Lutra ASV, as shown in Fig. 1b, to conduct an initial survey and then use the J-Horizon planner to compute a new path with the objective to minimize the uncertainty and maximize information gain on the underlying scalar field. The initial path for representative data collection is presented in Fig. 7a, with the scalar field generated from these data and the path planned by the J-Horizon planner shown in Fig. 7b. The ASV executed the first section of the path prescribed by the J-Horizon planner, and results are shown in Fig. 8. At the terminus of this executed section, J-Horizon prescribed a *jump* to a new region for further sampling. This second section was not executed for the proof-of-concept field trial.

As seen in Fig. 7b, the J-Horizon gathers data in areas of low data yield from the initial data collection. For instance, the lower left hand corner of Fig. 7b is an area that was not covered in the initial survey and requires more data collection to accurately represent the underlying field. This is the area of focus for our execution of the J-Horizon planner path, as more than half of entire length of the planned path lies within this region. The portion of the planned path that was executed is shown in Fig. 8 by the red path.

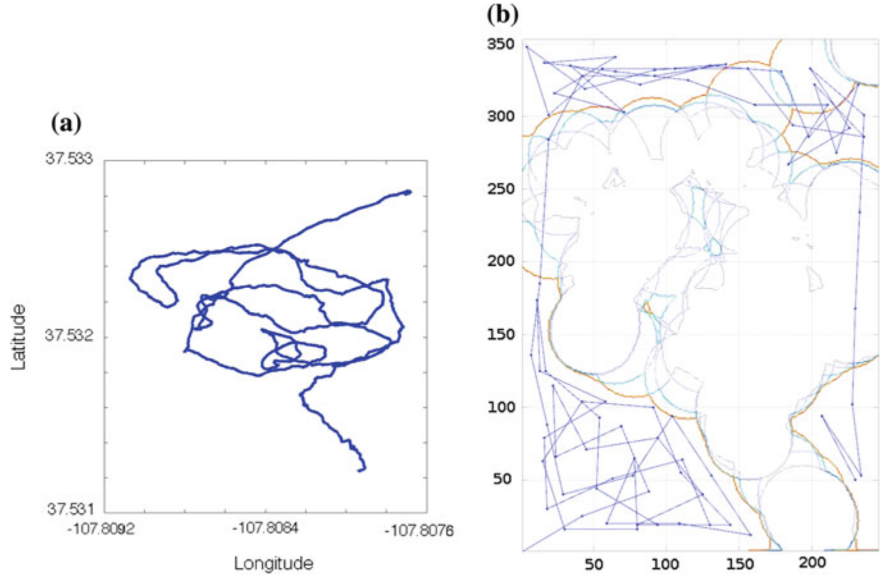


Fig. 7 Paths executed by the ASV to test and demonstrate the J-Horizon planner. **a** The ASV's initial path on Lake Haviland outside Durango, CO. **b** J-Horizon path generated on ASV path shown in (**a**)

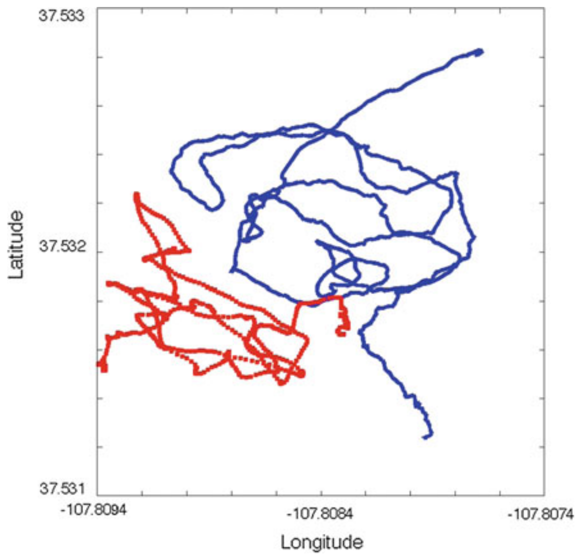


Fig. 8 The initial, data collection path (*upper*, *blue*) and the J-Horizon path (*lower left*, *red*) executed by the ASV on Lake Haviland

6 Conclusion

Improved algorithm design for autonomous vehicles operating on water has a promising future in robotics. Collecting higher quality data that can be better utilized by scientists, as well as reducing costs of the data collection, is a key goal in making autonomous monitoring a reality. In this paper, we presented a receding horizon algorithm that attempts to find an optimized path to perform costly, and sometimes difficult or dangerous, data collection in oceans or other large bodies of water. In conclusion, we have presented a novel approach to better collect data over scalar fields. Simulation results show a 14.53 % gain in reward of information collection compared to a lawnmower pattern while in simulation. The J-Horizon planner shows a 23.85 % increase in information gathered in a simple experimental trial compared to a lawnmower pattern. Both of these results show quality gains compared to a lawnmower path of equivalent length. These optimized sampling paths allow scientists to more easily collect pertinent data in the field.

7 Future Work

Extended hardware trials would verify that the simplifying assumptions made in the algorithm design are realistic. Additional performance improvements could be made by running the planning algorithm on the ASV to update the error of the scalar field and re-plan in realtime. This would serve to allow the ASV to run autonomously in highly dynamic environments for longer periods of time without having to transmit data to the shore for processing.

The obvious extension of this work is the application to Autonomous Underwater Vehicles, and sampling in three dimensions. After further testing and validation on 2-D scalar fields, we are planning to investigate problems that exist for both underwater and aerial applications.

Finally, we are investigating an extension to the J-Horizon planner that includes applications for frontier searching, enabling a robotic platform to explore areas with unknown data quality. Such an algorithm will aim to balance explore vs. exploit in missions, searching new areas while also collecting data in areas that are deemed interesting or have low data density.

Acknowledgments The authors would like to thank Jonathan Nash at Oregon State University for his insightful comments and insight into the algorithmic development. A. Stuntz was supported in part by ONR grant N00014-14-1-0490. R.N. Smith was supported in part by NSF Grant DUE-1068341 and a gift from the Fort Lewis Foundation. S. Yoo, Y. Zhang, and G. Hollinger were supported in part by ONR grant N00014-14-1-0509.

References

1. Binney, J., Krause, A., Sukhatme, G.: Informative path planning for an autonomous underwater vehicle. In: IEEE International Conference on Robotics and Automation(ICRA), pp. 4791–4796. Anchorage, Alaska (2010)
2. Curcio, J., Leonard, J., Patrikalakis, A.: Scout—a low cost autonomous surface platform for research in cooperative autonomy. Oceans, pp 725–729 (2005)
3. Frolov, S., Garau, B., Bellingham, J.: Can we do better than the grid survey: optimal synoptic surveys in presence of variable uncertainty and decorrelation scales. J. Geophys. Res. Oceans **119**(8), 5071–5090 (2014). doi:10.1002/2013JC009521
4. Gotovos, A., Casati, N., Hitz, G., Krause, A.: Active learning for level set estimation. In: International Joint Conference on Artificial Intelligence, Beijing, China (2013)
5. Grasmueck, M., Eberli, G.P., Viggiano, D.A., Correa, T., Rathwell, G., Luo, J.: Autonomous underwater vehicle (auv) mapping reveals coral mound distribution, morphology, and oceanography in deep water of the straits of florida. Geophys. Res. Lett. **33**(23) (2006)
6. Hitz, G., Gotovos, A., Pomerleau, F., Garneau, M.E., Pradalier, C., Krause, A., Siegwart, R.: Fully autonomous focused exploration for robotic environmental monitoring. In: IEEE International Conference on Robotics and Automation (ICRA), pp 2658–2664 (2014). doi:10.1109/ICRA.2014.6907240
7. Hollinger, G., Singh, S.: Proofs and experiments in scalable, near-optimal search by multiple robots. In: Robotics: Science and Systems, June 2008
8. Hollinger, G.A., Sukhatme, G.: Sampling-based motion planning for robotic information gathering. In: Robotics: Science and Systems (2013)
9. Mora, A., Ho, C., Saripalli, S.: Analysis of adaptive sampling techniques for underwater vehicles. Auton. Robots **35**(2–3), 111–122 (2013)
10. Schouwenaars, T., How, J., Feron, E.: Receding horizon path planning with implicit safety guarantees. In: Proceedings of the American Control Conference, vol 6, pp 5576–5581. IEEE (2004)
11. Stoker, C., Barch, D., Farmer, J., Flagg, M., Healy, T., Tengdin, T., Thomas, H., Schwer, K., Stakes, D.: Exploration of mono lake with an rov: a prototype experiment for the maps auv program. Autonomous Underwater Vehicle Technology AUV'96, 33–40 (1996)
12. Tisdale, J., Kim, Z., Hedrick, J.K.: Autonomous path planning and estimation using uavs. IEEE Robot. Autom. Mag. **16**(2), 35–42 (2009)