Data-driven Comparison of Spatio-temporal Monitoring Techniques

Jeffrey A. Caley and Geoffrey A. Hollinger

***Abstract*— Monitoring marine ecosystems is challenging due to the dynamic and unpredictable nature of environmental phenomena. In this work we survey a series of techniques used in information gathering that can be used to increase experts' understanding of marine ecosystems through dynamic monitoring. To achieve this, an underwater glider simulator is constructed, and four different path planning algorithms are investigated: Boustrophendon paths, a gradient based approach, a Level-Sets method, and Sequential Bayesian Optimization. Each planner attempts to maximize the time the glider spends in an area where ocean variables are above a threshold value of interest. To emulate marine ecosystem sensor data, ocean temperatures are used. The planners are simulated 50 times each at random starting times and locations. After validation through simulation, we show that informed decision making improves performance, but more accurate prediction of ocean conditions would be necessary to benefit from long horizon lookahead planning.**

I. Introduction

Marine ecosystems are complex. To gain insight into these ecosystems, underwater gliders are employed to perform long duration monitoring missions of physical and biological phenomenon. One example is observing animal aggregations, which play an important role in marine ecosystems [1] and are not fully understood. To increase understanding, there is a need to make underwater gliders adapt to sensor readings in real time to maximize data collection on these aggregations. This is an active sensing problem, where not all data are treated equally. Paths must be planned to maximize the information gained for specific areas of interest.

Consider the case of using an underwater glider (Fig. 1) to find and detail biological hotspots created by episodic upwelling. The underwater glider needs to find a hotspot of sea-life and then constantly monitor it as it evolves over time. To achieve this, algorithms are needed that can operate in a spatially and temporally dynamic environment.

To examine this problem, we have constructed a glider simulator to aid in the design, testing and verification of algorithms for tracking physical and biological phenomenon. Based on historic ocean data taken from the coast of California, our simulator models the motion and sensor readings of a underwater glider as it travels through the water column. The simulator is built such that 20 day missions can be simulated in seconds, resulting in a sandbox for experimentation.

Using our simulator, a series of techniques used for information gathering are surveyed: Boustrophedon paths [2], a gradient based approach, Sequential Bayesian Optimization [3] and a Level Set Estimation approach [4]. A data driven comparison of these planning techniques is performed using real ocean data. To our knowledge, these methods have not been compared on a common set of real ocean data. Our main contributions are the data driven comparison of the four methods and a 3D glider simulator with Robot Operating System (ROS) [5] integration for future glider path planning algorithm development.

This work has been funded in part by the Office of Naval Research grant N00014-14-1-0509 and support from the W. M. Keck Foundation.

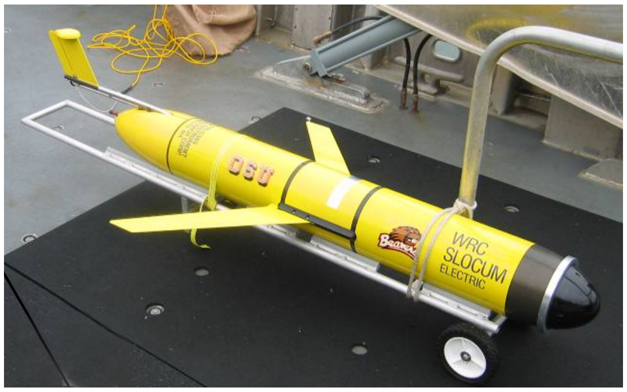
Fig. 1: A Slocum glider, used for long endurance environmental monitoring missions

II. Background And Related Work

A. Underwater Glider

Underwater gliders are a class of low power autonomous underwater vehicles (AUV). Gliders are designed to move through the water column using small changes in buoyancy [6]. Although not as fast as traditional propeller driven AUVs, they have significantly increased range and mission duration, allowing them to spend months in the ocean traveling thousands of kilometers. When equipped with sensors, gliders can be used to make measurements of temperature, conductivity, current velocity and other vital ocean phenomena. The simulator detailed in this paper models the macro-scale movement of the these gliders in the ocean.

B. Gaussian Process

In this work, we use a Gaussian Process to model the ocean environment. A Gaussian Process (GP) is defined as a collection of random variables with a joint Gaussian distribution [7]. A Gaussian prior is placed over a function and can be completely defined by its mean function $\mu(x)$ and its covariance function $k(x, x')$. A noisy function f can be represented as $f(x) \sim GP(\mu(x), k(x, x'))$. Given a set of training inputs N and corresponding outputs y, a predictive distribution of f at an unknown query location can be computed. The needed parameters for the mean and covariance functions can be estimated by maximizing the marginal likelihood of the data. These predictions have been shown to be highly effective in modeling environmental phenomena [8]. We examine their use in modeling oceans processes.

C. Informative Planning Problems

Previous work on informative path planning has borrowed from the ideas of adaptive sampling and path planning problems. This has been done through the use of heuristic search [9], random graph search [3], or sampling based approaches [10] [11]. The goal of these is normally to find either a feasible path or an optimal path length.

Adaptive sampling is the problem of deciding where to take measurements to best monitor some variable of interest. This is a well explored area and it has been shown that the greedy algorithm performs well in submodular cases [12]. Informative path planning looks to combine both path planning and the sensor placement problem into a single problem. Yilmaz [13] approached the problem using mixed integer programming, but this requires a linear objective functions. A recursive-greedy algorithm for submodular orienteering is presented in [14], but this is more concerned with reaching a destination as oppose to the path to get there. A branch and bound technique in [15] solves for the optimal path and show significant speedup, but requires a complete map of the world.

III. Problem Formulation

In this work, we model and control an underwater glider as it explores an unknown environment performing an information gathering mission. Starting from its deployment location, the glider must explore the environment to observe a specific environmental variable, such as sea temperature, salinity or chlorophyll, above a certain threshold value. These variables are hypothesized to correlate with aggregations of marine life [8]. The goal of the glider is to maximize a utility function R, based on the time the glider spends observing data above a threshold value on a long duration mission. To measure this, the world is discretized into k grid squares that exist for the length of the mission t_{total}. The glider receives a score for observing above threshold data for all unique squares transversed in its most recent trajectory. Making a second observation of the same square at any time t results in no additional score unless the square's value has changed (from above to below the threshold) between observations.

$$R = \sum_{t=0}^{t_{total}} \sum_{d=0}^{k} Observed(d_t), \tag{1}$$

where $Observed(d_t) = 1$ if d_t is above the threshold and observed for the first time. $Observed(d_t) = 0$ otherwise.

The simulated world consists of a 3D continuous map with historic data for ocean current, temperature and salinity imported from the Jet Propulsion Laboratory's website [16]. The glider is provided with previous day's data to generate an initial estimate of potential areas of interest.

To navigate the simulated world, a planner selects from the four cardinal directions and places a waypoint for the glider to travel to in the chosen direction (Fig 2). Waypoints are placed at a distance such that it takes 12 hours of travel time to move between them. The planner selects the waypoint that will maximze uncertainty reduction of it's enviromental model and/or maximize Eq. 1. As the glider travels along its trajectory, the glider acquires data samples it can use for future planning. After each waypoint is reached, the planner uses its newly learned information to select the next waypoint. This will continue for the entire duration of the mission.

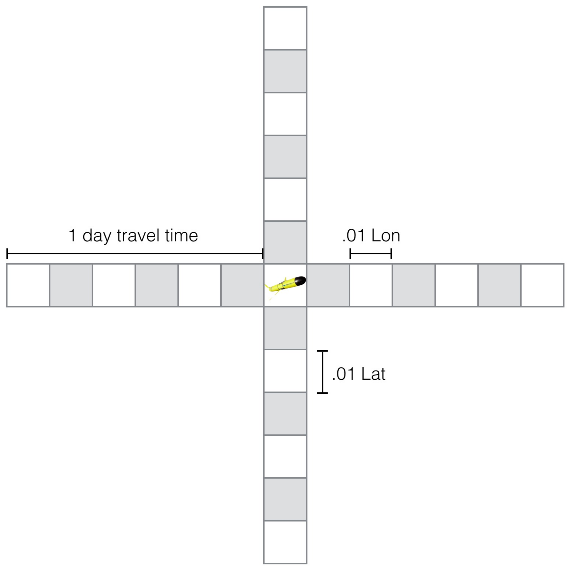

Fig. 2: Motion primitives for an ocean glider. Each Square represents .01 degree of latitude or longitude

IV. Simulator

To facilitate the investigation of the different informative path planning algorithms, a glider simulator (gliderSim) is constructed. GliderSim is built in Unity, a cross-platform game engine. It interfaces with the planners via ROS through the use of ROSBridge [17]. GliderSim simulates the motion of the glider through the water column and returns sensor data to ROS as it travels from waypoint to waypoint. All planning algorithms were implemented in Python on Ubuntu Linux. The Gaussian Process implementation is provided by Scikit-learn [18].

GliderSim simulates the movement of an ocean glider as it travels through the water column towards a GPS waypoint. This simulation is done in continuous space and time. At the beginning of any simulation, a time of day is specified, and the gliders initial location and GPS waypoint is selected. A deltaTime variable is used to select the step size for each screen redraw. For this work, a deltaTime of 1000 seconds was used. The gliders speed is set to $0.15\ m/s$ with an angle of attack at 26.5^o. This is consistent with the movement of a Slocum Glider, a glider commonly used for environmental monitoring [6]. All data measurements are taken at 30m in depth. Due to the nature of the underlying ocean data, this is a reasonable simplification.

GliderSim uses depth, temperature and ocean currents data from The Southern California Bight (SCB) ocean forecasting

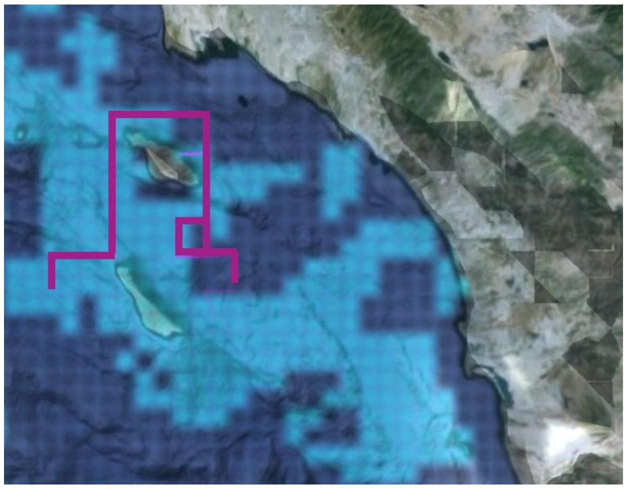
Fig. 3: An example path the glider would take during an information gathering mission.

system [16]. This data covers an area of $32.5^o - 34.7^o$ latitude and $238.8^o - 243.0^o$ longitude from March 01, 2013 to March 31, 2013. The time data is discretized to three times per day: 03:00:00, 09:00:00 and 15:00:00. The data resolution is 0.01^o in both directions. This natural discretization was used to define the grid squares used to calculate the movement, utility and score of each glider simulation. Because the glider simulator operates in continuous space and time, both location and time were rounded to the nearest point existing in the data set when requests for data were made by the planners. All simulations are run on a single laptop with a 2.5 GHz Intel Core i7 processor with 16 GB of RAM.

V. Methods

Four motion planning algorithms for environmental monitoring have been selected for comparison in this work. Each method has been adapted to fit the problem formation.

A. Boustrophedon Path

A Boustrophedon Path moves the glider back and forth across the region of interest in parallel, evenly spaced trajectories [2]. This is a common movement pattern used by glider operators for environmental monitoring and will serve as a baseline for our results [19].

B. Gradient based approach

The gradient based approach is a greedy algorithm that uses known information about the environment to exploit it. It starts by constructing a GP model M of the world using three days of previous environmental data D. Previous environmental data is available through in-situ sensors or satellites, depending on the environmental variable. Providing M with a latitude (lat), longitude (lon) and time returns an estimate of the environmental variables at that location and time. After constructing M, the gradient based approach looks at the four possible paths p the glider might traverse and calculates which discretized squares s the glider will path through as it travels p (Fig 2). For each s, it uses M, given lat, lon and estimated time of arriving at s, to estimate the value of the environmental variable of interest at location s. By summing each s for each location in p, the planner can get an estimate of the value of traveling along each p. By selecting the p with the largest summed path, the gradient based approach will travel greedily along the path likely to return the largest short term gain.

1) Expectation:

An alternative to the simple summing of estimated values seen in the gradient based approach is to incorporate both the mean $u(x)$ and variance $\sigma(x)$ provided by the GP to calculate the likelihood of any particular location s being above the threshold value h. The expectation uses

$$P_t(s) = \frac{u_t(s) + \sigma_t(s) - h}{[h - u_t(s) - \sigma_t(s)] + [u_t(s) + \sigma_t(s) - h]} \quad (2)$$

to calculate the percent chance that location s will be above or below h. These percentages can then be summed, and the largest total path percentage can be used to determine the p which will be traversed next.

C. Level Set Method

The Level-Sets Method adapts an approach found in [4] to estimate the exploratory value of each p. This technique is designed to maximize the exploration of the glider, navigating solely for the purpose of building a better model M by traveling to areas of high uncertainty. Based on the Level Set Estimation Algorithm (LSE) [4], this Level-Sets method uses a GP's inferred mean $u(x)$ and variance $\sigma(x)$ to construct a confidence interval $Q_t(s)$ for any point $x \in D$.

$$Q_t(s) = [u_t(s) \pm \beta^{1/2}\sigma_t(s)] \quad (3)$$

This captures the uncertainty of our GP estimate given the previously provided data. β is a scaling factor, and is set to 1 in this work. This confidence interval can then be used to determine if a point can be classified as above or below the threshold value of interest or if more information is needed to make that determination.

The Level-Sets Estimate planner looks to take the path that will decrease the uncertainty of M the most. To accomplish this, the confidence interval is calculated for each $s \in p$. s that are classifiable are given a value of 0. s determined to need more information are examined further. For each unclassified point s_{uc} the ambiguity is calculated as:

$$a_t(x) = min\{max(Q_t(x)) - h, h - min(Q_t(x))\} \quad (4)$$

Ambiguity qualifies the uncertainty about the s_{uc} classification. The higher the ambiguity, the larger classification uncertainty; thus we would expect to gain more information by traveling there. The sum of ambiguities for each $s \in p$ is calculated, and the p with the largest ambiguity sum is traversed next.

D. Sequential Bayesian Optimization

Bayesian Optimization (BO) is a global optimization technique useful in finding the maximum of partially observable objective functions that are difficult to solve efficiently. Marchant et al. [3] introduce Sequential Bayesian Optimization (SBO), an extension of BO for sequential decision making that can formulate the problem as a Partially Observable

Markov Decision Process (POMDP). To solve the POMDP efficiently, a Monte-Carlo tree search is employed to provide an estimated near-optimal solution. Just like the gradient based approach, a GP will be used to model the data in the environment, with an SBO solving the planning problem. SBO builds on the previous techniques by performing a multi-step look ahead of potential paths.

We adapt SBO for this problem by implementing a Monte-Carlo tree search that uses our gradient based approach as an evaluation metric for each look ahead step. The tree is built incrementally, starting from an initial node n_0. n_0 represents where the glider is currently and its belief representation for f. Nodes are expanded by simulating the outcome of traveling along all possible next paths p. The maximum likelihood observation is used to update the GP for each of the leaf nodes. The leaves are updated with their new current location and their belief representation for f given the newly acquired observations. Selecting a leaf node for expansion is done using a softmax citesoftmax proportionally weighted selection.

Selection and expansion of new leafs is done until a maximum depth is reached. The utility acquired at each step is then run back up the tree, updating the scores of each leaf node evaluated on the way down. If at any point the score being updated is less then the parent node's current score, the update is stopped. This means the current path is less desirable then a previously evaluated path on the same branch. When all scores are updated, the whole process is repeated until the maximum iterations is reached. At this point, the best action is determined by evaluating the leaf nodes under n_0 and selecting the leaf with the highest score. Algorithm 1 shows the full procedure.

Algorithm 1 Monte Carlo Tree Search for SBO

```
1:  procedure MCTS($b(f), p, depth_{max}$)    ▷ Inputs: belief $b(f)$, path p, max depth $depth_{max}$
2:      $n_o \leftarrow NewNode(b(f), p, reward_{min})$
3:      $i \leftarrow 0$
4:      while $i < MaxIterations$ do
5:          $TreeDescent(n_o, depth_{max})$
6:  procedure TREEDESCENT($n_{parent}, depth_{max}$)
7:      $i \leftarrow 0$
8:      while $i < depth_{max}$ do
9:          $children \leftarrow GenerateChildren(n_{parent})$
10:         $n_{parent} \leftarrow selectChild(children)$
11:     $i \leftarrow 0$
12:     while $i < depth_{max}$ do
13:         $n_{parent}.Score \leftarrow n_{parent}.Utility$
14:         if $n_{parent}.Parent.Score < n_{parent}.parent.Utility + n_{parent}.Score$ then
15:             $n_{parent}.Parent.Score \leftarrow n_{parent}.parent.Utility + n_{parent}.Score$
16:         else
17:             Break
18: procedure GENERATECHILDREN($n$))
19:     if $n.Children = empty$ then
20:         Create nodes for next possible waypoints
21:         Calculate Utility for each Node
22:     else
23:         Break
24: procedure SELECTCHILD($children$))
25:     if a Children.score $= 0$ then
26:         Randomly select Child with score $= 0$
27:         Return Child
28:     else
29:         Use Softmax to select a proportionally weighted random child
30:         Return Child
```

VI. Simulations And Experiments

We now discuss the simulations and experiments run to compare the proposed techniques. For these experiments, ocean temperature is the variable of interest with a threshold value of 13.2^o. The value 13.2^o was selected because it generated a compelling environment for the glider to navigate. Trials with ocean currents affecting and not affecting the glider as it moves through the water were conducted.

A. Building the GP model

To construct the Gaussian process model, the SCB data was downsampled to a resolution of 0.2^o. The downsampling was done to reduce the data set down to a manageable size for GP construction. At each waypoint, a GP model is constructed using downsampled SCB data for the previous 3 days plus every grid square the glider has traversed to that point. The hyperparameters are learned the first time a GP model is constructed, and then the same hyperparameters are used for all following GP models.

B. Running a Simulation

Starting at a random location between $32.80^o - 33.45^o$ Latitude and $241.32^o - 242.18^o$ Longitude and between March 4-9, 2013, the glider traverses the ocean on a 20 day data collection mission. The glider operates by choosing a GPS waypoint to navigate to and then travels in a straight line (using dead reckoning) until it reaches it. Once a waypoint is reached, the glider simulator pauses simulation time to allow the planner time to calculate the next waypoint. The simulator operates at a rate of roughly 12 hours of simulated time per 10 seconds of actual time, so pausing the simulator to allow calculation time is important to keep a realistic simulation of operation in the real world. Additionally, we note that these algorithms are computationally efficient enough to be run in real time onboard a glider. Once the next waypoint has been selected by the planner, the simulator unpauses and simulates travel to the next waypoint.

C. Experiments

Each of the five planners described above was run a total of 50 times at random times and starting locations. The same random starting time and location was used for each planner. The scores for each simulation were recorded and averaged to produce an mean performance for each planning

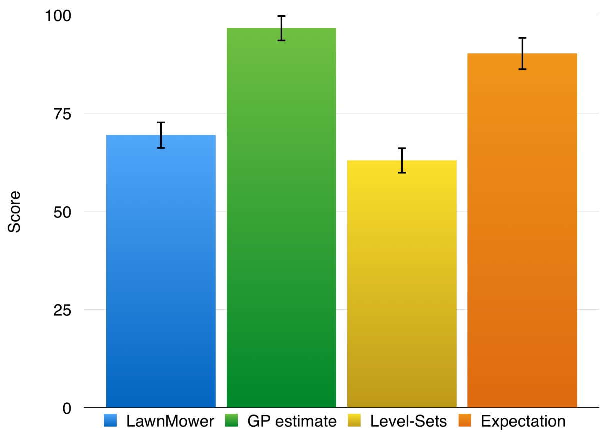

Fig. 4: The average score for a 20 day exploration mission given no ocean currents affected the motion of the glider. Score is defined as the number of data observations above the threshold value. GP estimation outperforms other techniques. Error in SEM.

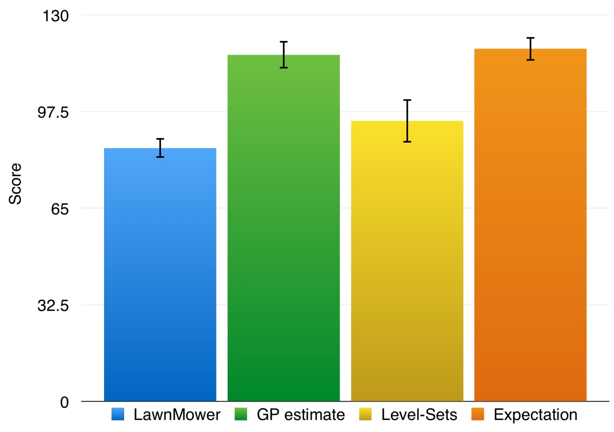

Fig. 5: The average score for a 20 day exploration mission given ocean currents affected the motion of the glider. Score is defined as the number of data observations above the threshold value.

technique.

1) SBO: Sequential Bayesian Optimization is a planning technique that is an extension of the others. The other four techniques only use a single-step lookahead while SBO uses a multi-step look ahead utilizing one of the original four's utility function for path evaluations. In this way, when SBO's lookahead depth is set to one, it performs exactly the same as its as its extended planner. Because of this, we evaluate SBO separate from the other planning techniques.

D. Results

Fig. 4 shows the results of the four planners when ocean currents are not considered. The Boustrophedon path sets a baseline result with an average score of 69 (meaning the glider passed through 69 grid squares that were above the 13.2^o during its 20 day mission). The Level-Sets method performed slightly worse than the lawnMower pattern with a score of 63. This is somewhat expected because of the nature of the Level-Sets method. Level-Sets has no interest in maximizing the glider's time spent in areas of high temperature; it

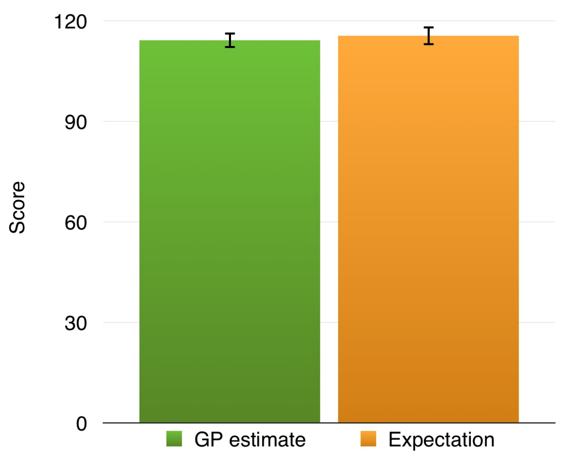

Fig. 6: The average score for a 20 day exploration mission given no ocean currents affect the motion of the glider. These results use actual data values instead of GP estimates for path utility calculation. Score is defined as the number of data observations above the threshold value.

is simply interested in maximizing its internal model of the world. This strict exploration based approach is especially difficult due to the fact the world is constantly changing, meaning the glider's model of the world is constantly being put out of date.

There is a great deal of variance in score between individual simulated trials of each planner. This is due to the random starting location and time. Some starting locations are naturally located in high temperature areas, meaning a high score will be achieved no matter the planner. Some planners will be perform better, but all planners by default will navigate to warm water. Other starting locations are in extremely cold locations in the data and no matter the decision making, wouldn't achieve a large score.

Using the GP to estimate temperatures and traveling to the hottest area performed best, with an average score of 96.5. This makes sense as ocean temperatures are relatively smooth gradients and traveling up these gradients should result in the glider navigating to warmer waters. The expectation, which uses the GP estimates mean and variance to perform a hybrid exploration and exploitation of the region performed nearly as well with a score of 90.2. Fig 6 shows the result of performing the same experiments with the true temperature values replacing the estimates of the GP. These scores represent the best score possible given a perfect GP estimate. With scores of 115 and 116, we can see that the GP isn't a perfect model of ocean temperatures, but does provide enough of an estimate to out perform a non-adaptive technique (Boustrophedon path).

Fig. 5 shows the results for the planners while ocean currents are affecting their paths. Here we see a similar story, with the Boustrophedon path and Level-Sets method performing comparably while the expectation and gradient based approach perform experimentally better. The standout difference between the two is the nearly 20 percent increase in all scores. This difference can be attributed to the increased

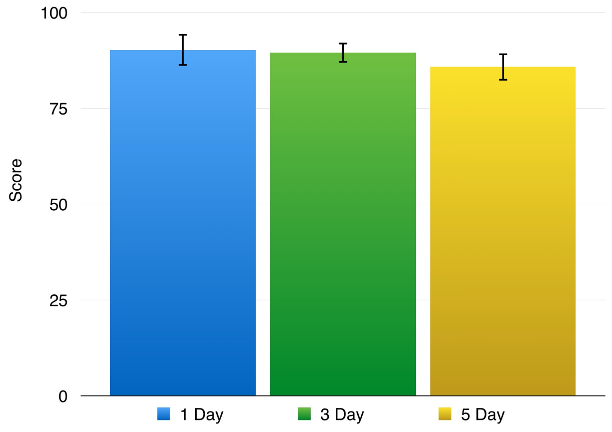

Fig. 7: The average score for a 20 day exploration mission using the SBO method. Trials with 1, 3 and 5 day lookahead. Planner performance decreased with increased lookahead distance. Score is defined as the number of data observations above the threshold value.

distance traveled by the glider due to the currents. Because the glider is able to cover more ground while on its 20 day mission, the scores are naturally larger than trials without ocean currents.

1) SBO: For Sequential Bayesian Optimization, three different tree depths were tried; 1, 3 and 5 depth. Our SBO planner used the expectation as its evaluation metric and was run over the same 50 location and times as other planners. The results, seen in Fig. 7, show that the average score decreases as the SBO explores farther out into the future while planning each move. These results seem to suggest that the GP isn't providing a good model for future data. The farther out the GP attempts to estimate, the further its estimates get from the true value. Because all future paths down the MCTS tree are weighted the same, a place that seems promising in the distant future can dramatically influence the next decision.

VII. Conclusions And Future Work

In this paper, we performed a survey of techniques used for information gathering. We have shown empirically that using a GP to model ocean data for planning future paths outperforms current methods. These techniques were validated in simulation using real ocean data. These improvements in planning can increase the efficiency and robustness of future glider missions as they study the dynamics of the oceans.

The results in this paper open up a number of interesting areas for future work. This work does not attempt to address the problem of ocean currents affecting the path of the glider, or the planning problems that occur when ocean currents are introduced. Due to the slow nature of ocean gliders, ocean currents can dramatically affect the paths of gliders. Incorporating these currents into the planning process is essential to make these planners practical for use in actual ocean environments.

While using the GP to model ocean data showed improvements on current methods, its ability to forecast ocean changes into the future showed limitations. Future work will also include looking at other modeling techniques, such as deep learning networks, to replace the GP model.

Acknowledgements

We thank Jack Barth and Kelly Benoit-Bird at Oregon State University for their insightful comments.

References

[1] C. Breder, "Equations descriptive of fish schools and other animal aggregations", *Ecology*, vol. 35, pp. 361–370, 3 1954.

[2] H. Choset, "Coverage of known spaces: the boustrophedon cellular decomposition", *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.

[3] R. Marchant *et al.*, "Sequential bayesian optimisation for spatial-temporal monitoring", in *The Thirtieth Conference on Uncertainty in Artificial Intelligence*, AUAI Press, 2014, pp. 553–562.

[4] A. Gotovos *et al.*, "Active learning for level set estimation", in *The Twenty-Third international joint conference on Artificial Intelligence*, AAAI Press, 2013, pp. 1344–1350.

[5] M. Quigley *et al.*, "ROS: an open-source robot operating system", in *The Open-Source Software Workshop, International Conference on Robotics and Automation*, 2009.

[6] R. Bachmayer, N. E. Leonard, J. G. Graver, E. Fiorelli, P. Bhatta, and D. Paley, "Underwater gliders: recent developments and future applications", in *Proc. International Symposium on Underwater Technology*, 2004, pp. 195–200.

[7] C. Rasmussen and C Williams, *Gaussian Processes for machine Learning*. MIT Press, 2006.

[8] J. Das *et al.*, "Hierarchical probabilistic regression for auv-based adaptive sampling of marine phenomena", in *IEEE International Conference on Robotics and Automation*, IEEE, 2013, pp. 5571–5578.

[9] M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[10] J. J. Kuffner and S. M. LaValle, "Rrt-connect: an efficient approach to single-query path planning", in *IEEE International Conference on Robotics and Automation*, IEEE, vol. 2, 2000, pp. 995–1001.

[11] G. A. Hollinger and G. Sukhatme, "Sampling-based motion planning for robotic information gathering.", in *Robotics: Science and Systems*, 2013.

[12] C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in gaussian processes", in *The 22nd international conference on Machine learning*, ACM, 2005, pp. 265–272.

[13] N. K. Yilmaz *et al.*, "Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming", *IEEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 522–537, 2008.

[14] C. Chekuri and M. Pal, “A recursive greedy algorithm for walks in directed graphs”, in *IEEE Symposium on Foundations of Computer Science*, IEEE, 2005, pp. 245–253.

[15] J. Binney and G. S. Sukhatme, “Branch and bound for informative path planning.”, in *International Conference on Robotics and Automation*, Citeseer, 2012, pp. 2147–2154.

[16] Z. Li. (2015). JPL OurOcean portal, [Online]. Available: `http://ourocean.jpl.nasa.gov/` (visited on 05/19/2015).

[17] S. Livingston. (2015). Rosbridge suite, [Online]. Available: `http://wiki.ros.org/rosbridge_suite` (visited on 06/07/2015).

[18] F. Pedregosa *et al.*, “Scikit-learn: machine learning in Python”, *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[19] J. Binney *et al.*, “Informative path planning for an autonomous underwater vehicle”, in *The IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 4791–4796.