Efficient Terrain Driven Coral Coverage Using Gaussian Processes for Mosaic Synthesis

Sandeep Manjanna, Nikhil Kakodkar, Malika Meghjani, and Gregory Dudek
School of Computer Science,
McGill University
email:{msandeep, nikhil, malika, dudek}@cim.mcgill.ca

***Abstract*—In this paper we present an efficient method for visual mapping of open water environments using exploration and reward identification followed by selective visual coverage. In particular, we consider the problem of visual mapping a shallow water coral reef to provide an environmental assay. Our approach has two stages based on two classes of sensors: bathymetric mapping and visual mapping. We use a robotic boat to collect bathymetric data using a sonar sensor for the first stage and video data using a visual sensor for the second stage. Since underwater environments have varying visibility, we use the sonar map to select regions of potential value, and efficiently construct the bathymetric map from sparse data using a Gaussian Process model. In the second stage, we collect visual data only where there is good potential pay-off, and we use a reward-driven finite-horizon model akin to a Markov Decision Process to extract the maximum amount of valuable data in the least amount of time. We show that a very small number of sonar readings suffice on a typical fringing reef. We validate and demonstrate our surveying technique using real robot in the presence of real world conditions such as wind and current. We also show that our proposed approach is suitable for visual surveying by presenting a visual collage of the reef.**

***Keywords*-Gaussian Processes; Selective Coverage; Coral Mosaic;**

I. Introduction

We present an efficient technique to build a depth map of the sea-bed with sparse measurements from a depth sensor and selectively cover the hot-spots, defined by a reward function depending on the depth, to obtain a detailed visual survey of the coral reef. Taking a general perspective, our work looks at using a reward function over the plane to develop an approach for sequential data collection, potentially on a recurring basis and in the presence of disturbances (such as wind). We further assume that a (different) sensor can be used to measure this reward function and this process of reward estimation also needs to be conducted efficiently. This emphasis on efficient data collection is motivated by our interest in field robotics, and especially marine robotics, where large areas need to be covered and the efficient coverage in the face of limited power, physical demands on human supervisors, and external influences is significant.

Environment monitoring is an activity of enormous and growing importance. In the marine environment, this is motivated by a desire to observe factors such as the impact of sea level change or increase in temperature, acidification, and changes in fauna (such as depletion of coral). Modeling these and related phenomena calls for enormous amounts of data, sometimes sampled over a long duration. Collecting such data can be tedious for a human operator, given the various resource constraints and challenges in terms of consistency and repeatability.

Figure 1: The Autonomous Surface Vehicle (ASV) used in field experiments

As per the application context and experimental validation, we consider visual mapping of shallow water coral reefs. Shallow water marine environments including wetlands and estuaries have exceptional environmental, scientific and economic value and coral reefs are the most important of all. Despite their immense importance, coral reefs in particular are dying worldwide and a first step in estimating the impact of any remediation effort is to evaluate the reef itself. Conducting reef surveys implies taking repeated measurements under varying weather conditions and in a consistent manner. Using a human diver for such tasks is both expensive and potentially dangerous. While underwater observations remain indispensable, valuable data that is economical, repeatable, scientifically objective and effective data can often be collected from the surface. Using an autonomous vehicle for collecting such visual data from coral reefs is efficient and relatively risk-free, although the limitations on available power make efficient coverage particularly important (notably using an electric vehicle, instead of a gas-powered one, facilitates access to protected marine environments).

We seek to efficiently build a bathymetric model (depth map of the sea floor) using data collected from our robot with a relatively short trajectory and a limited number of samples. We use a Gaussian process (GP) to infer a

complete terrain map from these samples. We then seek to efficiently cover the most visually valuable regions of this map using a data-driven coverage algorithm. In this case, we exploit the idea that the shallowest water allows for the best measurements of the underlying reef, and is thus the most valuable. Other measurable metrics can, in principle, be used for the reward function including visual diversity and environmental conditions. The sea floor is, of course, a complex surface especially surrounding the fringing reefs which have complex 3D structures. To build these models we exploit the power of Gaussian process regression (also known as Kriging [1]) to represent these data-driven functions, interpolate between measurements, and also provide a covariance estimate at each point that can be used to drive further data acquisition.

One of the essential features of a good coverage algorithm is to visit the hot-spots in the increasing order of their value. This becomes significant when the task is to cover a given region in limited time. This behavior of selecting the salient regions to examine first is the key feature of our selective coverage algorithm. We present a value-iteration based algorithm which covers the entire region of interest, but in a prioritized fashion. We assume an underlying distribution for the phenomenon that needs to be modeled and build an off-line trajectory to cover the high probability regions along with a concern to reduce the travel time and energy consumption. Some of the potential applications would be - collecting samples from the ocean based on the surface temperature data from static sensor nodes or temperature maps from satellites, sampling visual data of coral reefs to monitor their health and growth, and sampling atmospheric gases based on the satellite maps. Off-line planning is essential when the mission is time and precision critical. We validate and demonstrate the results on a robot boat (Fig. 1) in the presence of wind and current.

The major contributions of this work are: an intelligent coverage algorithm which generates efficient paths that cover high rewarding regions; an analysis of the effect of coarse data on the efficiency of Gaussian Processes; finally, application of these techniques to achieve a critical task of monitoring the coral reefs.

II. Related Work

Environment monitoring has been of great concern for scientists all over the world due to several factors like growing sea-levels, increase in the global temperatures, increased sea surface temperatures, and depletion of corals. Scientists are interested to understand the phenomenons of nature, but this needs enormous amount of data sampled over a long duration of time. These environments are not very safe for humans to sample regularly, thus making it a perfect application for using autonomous agents. Collecting wide-field data, especially in a sea, is often achieved using sessile sensor nodes [2] [3]. These have an advantage in terms of consistency, but need to be replaced, often provide limited measurement density, and are rarely suitable for visual data.

Another approach to collect a high resolution data for such applications is to use an active agent to cover the region of interest. Many methods for coverage consider uniformly interesting regions and aim at achieving a complete coverage of the region [4]. An application like search for lost targets, promote a complete coverage [5]. If the problem is to explore and map an unknown environment, it requires sampling the entire terrain [6] [7]. With limited resources (e.g., battery life, and time of operation), a complete coverage is impractical if the environment to be covered is large with very few hot-spots. Hollinger *et al.* [8] consider a similar problem of visiting sensor nodes deployed in the ocean using an autonomous agent as a variant of the Traveling Salesperson Problem.

Recently there is a growing interest in non-uniform coverage. Seyed *et al.* propose a coverage strategy based on space-filling curves that explore the region non-uniformly [9]. They propose a coverage tree with Hilbert-based ordering of nodes. Another interesting approach for coverage based on the model of curiosity is proposed by Girdhar *et al.* in [10]. Their information theoretic path planning technique provides paths passing through regions with higher surprise factor such that they can be used for distinguishing various terrains. However, the paths provided by such a method are highly biased by temporary variations which is good for anomaly detection but might not be suitable for persistent monitoring. Our goal is to efficiently collect coral reef data and generate adaptive off-line trajectories for repeated monitoring. We propose a selective coverage algorithm to efficiently plan a path based on the partial knowledge about the environment.

We propose a realistic application for our algorithm in terrain based coral monitoring. In this application, we need an underlying terrain map as a reward distribution. Several methods for modeling and representing terrains [11] exist in literature. But most of these methods do not handle spatially correlated data and do not have a convenient form to represent the uncertainty associated with the model. Gaussian Processes (GP) are very efficient in handling these cases and GP modeling is helpful in cases where parametric representation of the underlying phenomenon is not easily determined. It also effectively incorporates the uncertainty associated with the computed model. In the robotics context, several authors including Kemppainen examined subsampling using GPs and considered optimal sampling strategies in terrestrial environments [12]. GP's have also proved effective in modeling the spatial distribution terrain height or other signals such as radio signal strength [13], [14]. More broadly GP regression has also proven effective in a number of other robotics contexts such as modeling the dynamics of complex systems for controller design or learning [15]. We leverage GP's above mentioned terrain modeling capabilities to build the depth map of the sea floor and follow the depth

map to efficiently cover the corals in the region. We also present an evaluation of GP models to build efficient terrain models with minimal sensor measurements.

III. Gaussian Processes for Depth Map

In this section, we review the use of Gaussian Process (GP) models for regression, as presented in [16]. GP formulation uses a supervised learning setup in which it considers a training set $S = \{(x_i, y_i) \mid i = 1, ..., N\}$, where $x_i \in \Re^D$ denotes a sample from a D-dimensional input space and $y_i \in \Re$ denotes the sensor output at this sample location. The sensor measurements $y \in \{y_i\}_1^N$ are modeled as noisy observations of the underlying function $f(x)$ given by $y = f(x) + \epsilon$, where ϵ is assumed to be an independent and identically distributed Gaussian noise with zero mean and variance σ_n^2.

GP models the function $f(x)$ as a multivariate Gaussian distribution, completely defined by the mean function $m(x)$ and the covariance function $k(x, x')$, also known as the kernel, i.e. $f(x) \sim \mathcal{N}(m(x), k(x, x'))$. The training set is used to learn this model by tuning the mean and the covariance functions. Using the learned model, predictions on new inputs $X_* = \{x_{*1}, ...x_{*m}\}$ is described by $\mathcal{N}(\mu_*, \Sigma_*^2)$, where

$$\mu_* = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} y \tag{1}$$

$$\Sigma_* = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \tag{2}$$

Here K represents the covariance matrix calculated between the corresponding inputs as defined by the covariance function. Intuitively, covariance function describes the relationship between two outputs as a function over their inputs (location in our case).

IV. Efficiency of Gaussian Processes with coarse data

Given a region of interest, we seek to survey the corals occurring within the area using the Autonomous Surface Vehicle (ASV). First stage in our approach involves conducting a quick bathymetric survey and generating a depth map by employing the GP regression. Since we are limited by factors such as robot battery, our goal is to conduct this survey with relatively shorter trajectory and fewer sample points than it would take to do a complete dense coverage. To measure the reliability of a depth map generated from fewer number of samples, we first conducted a dense survey of the region shown in Fig. 2. This lawnmower was completed in two charge cycles of the vehicle, which can be seen in Fig. 2 where the robot was brought back to the shore mid-mission to replace the batteries and the mission was resumed from where it had left off.

The sensor we use provides only one point-measurement for every sampling. From the coverage in Fig. 2, we collected a total of 2962 sample points using a sonar sensor

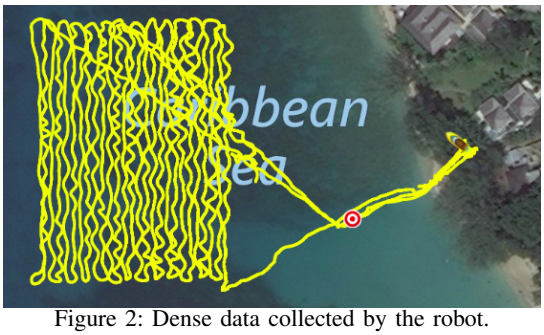


Figure 2: Dense data collected by the robot.

running at $1Hz$. Fig. 3 illustrates the performance of the GP regression technique computed as root mean squared error versus the number of training samples. For this evaluation, we randomly divide the 2962 sample points into eight sets. Hence at a time there are 2562 training points and 400 test points. We then perform 8-fold cross-validation using these sets. During each fold of the cross-validation we carry out the GP formulation multiple times, each time randomly choosing n training points from the training set, n starting from 100 with increments of 300 points. For each n we repeat the procedure 50 times, each time re-selecting n samples from the training set. Finally average the mean and standard deviation for every trial with n sample points over all the 8-folds. The results indicate that a sparse sampling in the range of $(1200, 2000)$ samples would still model the terrain well enough for our application domain. Hence, for field experiments we used sparse dataset with 1340 sample points to generate the depth map.

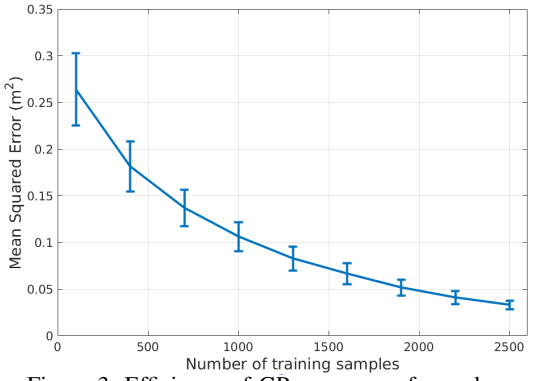


Figure 3: Efficiency of GPs over no. of samples.

V. Selective Coverage

In the previous sections we addressed the problem on mapping sea floor with minimum number of sensor readings. We use a depth map to achieve the coverage of the coral-heads in the selected region. We make an assumption that in a given area, the coral-heads are at shallower depths than the plain sand. We make this assumption based on our observations during field trials and inputs from the experts in marine robotics. This assumption is true when we are away from the shore line as the sea bed tends to be smooth with corals rising above the sand surface. Based on this

assumption, covering the shallower regions would provide a good coverage of the corals in the selected region.

We use the modeled depth map as an input distribution to cover the shallower regions with a precedence, thus making sure the corals are fully covered before the robot runs out of battery. One of the salient features of this algorithm is the use of predictions for environment dynamics, such as wind direction, wind speed, and wave currents, to generate an efficient off-line trajectory. Thus, the proposed approach adapts the trajectories according to the wind prediction inputs. Once we have the depth map of a certain region which we are interested in monitoring, the selective coverage algorithm can be used to generate coverage trajectories for regular monitoring of the coral reef.

A. Algorithm

The region of interest is discretized into grid cells and each grid-cell is assigned a utility value equivalent to the integral of underlying probability distribution over that cell. In the coral coverage task, the utility value of a grid-cell is summed over the depth distribution provided by the Gaussian Processes. We assume a finite set of states S represented by these grid-cells and a finite set of actions A the agent can take at each state (grid-cell). We allow four actions in every state (grid-cell) - East, North, West, and South. We use a Global Positioning System (GPS) to localize the agent and achieve a guaranteed state transition with the help of a precisely tuned controller. Hence, we assume certainty in the transition between two states, i.e., every action a on an agent in state s will cause the agent to reach state s' with a probability of 1 ($P(s'|s,a) = 1$). As mentioned earlier, the agent collects a reward $R(s,a)$ for visiting a state $s \in S$ and taking an action a in state s.

We use the *Value Iteration* algorithm to compute the best action to be taken at a given state. Value iteration is a method of computing an optimal MDP policy. It computes the optimal value of a state $V^*(s)$, i.e., the expected discounted sum of rewards that the agent will achieve if it starts at that state and executes the optimal policy $\pi^*(s)$. The optimal value function $V^*(s)$, $\forall s \in S$, is defined by the following Bellman equation [17],

$$V^*(s) = \max_a \left(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s') \right), \quad (3)$$

where γ is a discount factor. Thus according to Eq. 3, the value of a state s is the sum of instantaneous reward and the expected discounted value of the next state, when the best available action is used. Optimal policy defines an action for every state that achieves the optimal value. Given the optimal value function for all states, optimal policy is defined by,

$$\pi^*(s) = arg\max_a \left(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s') \right) \quad (4)$$

In our approach, the rewards are defined by the summation of depth predictions under that state. We clear the underlying rewards as and when the corresponding state is visited. Thus the reward function is changing over time as the agent clears the rewards. But this is against the Markovian assumption to keep a track of visited states. Hence, in our formulation we use an one step MDP approach, where we model every state transition of the agent as MDP in a new world and compute the value function over the updated rewards of the world. Thus the convergence of the value iteration technique still holds good for every state transition. The overview of our approach is presented in the Algorithm 1.

Algorithm 1 Selective Coverage Algorithm

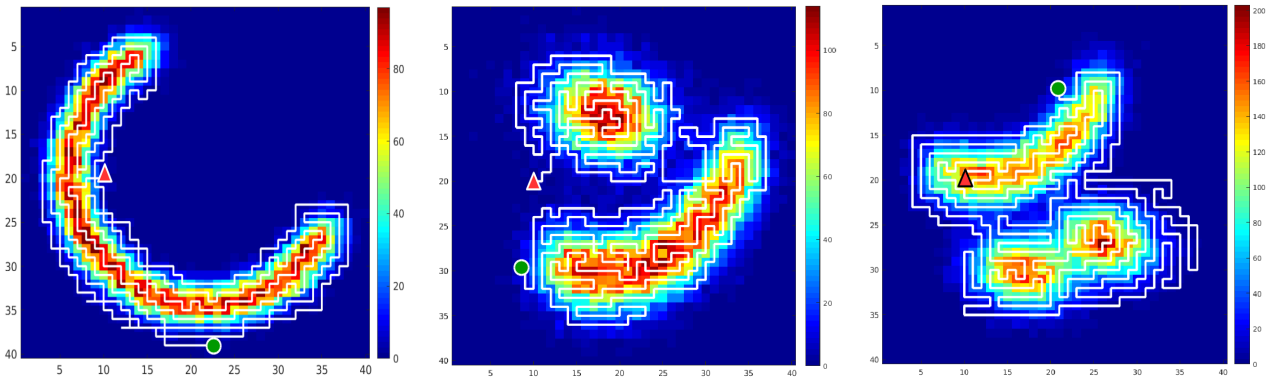
Input: Set of states S
Set of actions A
State transition probability $P(s'|s,a)$
$\forall (s,s') \in S$ and $\forall a \in A$
Reward $R(s,a)$ for each state $s \in S$
Discount factor γ
Starting state s_1
Reward threshold R_{limit}
Convergence threshold ϵ

Output: Path $\vec{W} = (s_1, s_2, ...s_n)$, a sequence of states.

$\forall s \in S$,
  Initialize $V^*(s)$, $\pi^*(s)$, and current state $s_{cur} = s_1$
Repeat
  $\vec{W} =$ **Append**$(\vec{W}, s_{cur})$
  $\forall s \in S$,
    ($V^*(s)$, $\pi^*(s)$) = **ValueIteration**$(S, A, P, R, \gamma, \epsilon)$
  Current Action, $a_{cur} = \pi^*(s_{cur})$
  ApplyAction a_{cur} on s_{cur} to obtain s_{next}
  $R(s_{cur}) = 0$, Clearing the reward at s_{cur}
  $s_{cur} = s_{next}$
until ($\sum_{s \in S} R(s) < R_{limit}$) **or** the region is fully covered.
Return $\vec{W}$

B. Trajectory adaptation

The proposed algorithm has the capability to adapt its output trajectory in accordance with predictions about the changes in the operating environments. In particular for coral monitoring, we use the predictions for wind speed and wind direction, provided by local weather station, to generate efficient coverage trajectories. The proposed approach is very useful to generate trajectories for re-sampling the given region of interest over a period of time. We incorporate the effect of wind on the agent's trajectory by subtracting a cost term into the reward function $R(s,a)$. Based on our observations in the field, we made an intuitive assumption that the horizontal component of wind (h_{wind}) affects the vertical trajectories of the agent and the vertical component



(a) C-shaped Distribution (b) Disjoint Distribution (c) Multiple-island Distribution

Figure 4: Trajectories generated by Selective Coverage algorithm over complex underlying reward distributions. The red triangle indicates the start and green circle indicates the end of the trajectory. The color-bars indicate the rewards.

of wind (v_{wind}) affects the horizontal trajectories. Hence, we subtract the weighted h_{wind} from rewards $R(s,a)$ when the action $a = \{North, South\}$. Similarly, the weighted v_{wind} is deducted from rewards $R(s,a)$ when the action $a = \{East, West\}$. The updated rewards are indicated in Eq. 5 and Eq. 6.

$$\forall a = \{North, South\} \text{ and } \forall s \in S,$$
$$R(s,a) = R(s,a) - (\epsilon * w_{speed} * sin(w_\theta)), \quad (5)$$

$$\forall a = \{East, West\} \text{ and } \forall s \in S,$$
$$R(s,a) = R(s,a) - (\epsilon * w_{speed} * cos(w_\theta)), \quad (6)$$

where ϵ is the convergence factor for the value iteration, w_{speed} is the speed of wind in ms^{-1}, and w_θ is the wind direction in radians.

C. Simulation Results

In this section, we illustrate the output of our selective coverage technique on different underlying probability distributions. We illustrate the capability of the algorithm to generate efficient trajectories with a diverse set of underlying distributions. Fig. 4(a) shows the trajectories generated on three different kinds of distributions. As it can be seen, the trajectory covers the highest rewarding states followed by lower rewarding sates. But, the trajectories generated are not greedy, i.e., as it is seen in Fig. 4(b) and Fig. 4(c), the algorithm efficiently trades-off between the rewards and the distance of the states. A near-by state with slightly lesser reward is preferred over the farther state with slightly higher rewards.

VI. Field Experiments and Results

Our motivation in this paper is to provide efficient trajectories for sampling and re-sampling the regions of interest based on depth and we examined the performance of the method in the open sea. To perform this task autonomously we provide the autonomous boat with a time and power efficient trajectory based on the depth map generated in a first phase under the supervision of human monitors on shore. We assess the performance of the coverage path generated by our algorithm with respect to three metrics - path length, time taken to execute the trajectory on the real robot, and power consumption by the robot motor.

A. Setup

We conducted field experiments on the North Bellairs reef in the Caribbean Sea off the shore of Barbados. An Autonomous Surface Vehicle (ASV), shown in Fig. 1, was used to collect the depth measurement data and execute the generated coverage trajectories. The maximum operating speed of the ASV is $1.3ms^{-1}$ and the battery life is 1.5 hours. The ASV is equipped with navigational sensor suit consisting of a GPS and an inertial measurement unit (IMU). The GPS operates at a frequency of $5Hz$. The ASV is also equipped with a sensor payload consisting of a single beam sonar and a water-sealed camera unit. The sonar has a maximum depth range of $100m$ and operates at $1Hz$. The camera unit records videos at $30fps$.

The robot was operated from the shore via a WiFi link. All the communication was relayed via a custom GUI shown in Fig. 5, which was first developed for a search application [18]. We use the GUI to design and relay custom trajectory patterns and view important mission parameters. Fig. 5 illustrates how we can use the GUI to select user defined way-points and display the ASV trajectory, in real time, as it makes its progress. The GUI also displays the 'Home' and the 'Current Target' coordinates. We evaluated the proposed coverage approach over a region of size 110 m x 75 m indicated by the rectangular box in Fig. 5.

B. Depth Map and Coverage Trajectories

We used the sparse sonar data from the robot boat to generate a depth map of the sea floor using the GP models as described in Section III. A sparse sampling trajectory (Fig. 6) with 1340 sample points was used to generate the depth map with a mean squared error of approximately $0.075m^2$ (from Fig. 3). The average sample density for sparse sampling is $0.16samples/m^2$ and that of dense

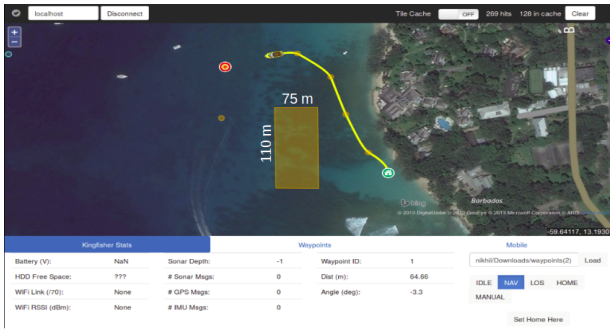


Figure 5: Graphical User Interface: Green marker indicates the the 'Home' location while the red marker shows the 'Current Target Way-point'. Each of the yellow circles shows the way-points selected for the ASV to traverse to. The GUI also draws the trail of the robot shown in yellow.

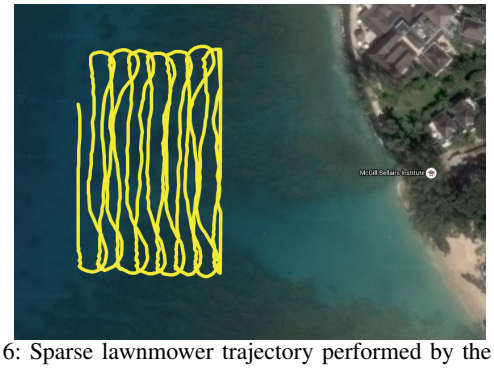


Figure 6: Sparse lawnmower trajectory performed by the boat to collect the bathymetric data to create the depth map.

sampling is $0.36 samples/m^2$. The depth map is used as an input to the selective coverage algorithm and the output trajectory overlaid on the depth map is illustrated in Fig. 7. The color-bar in Fig. 7 represents the depth measurements with deep red being the shallowest region and deep blue being the deepest. The selective coverage algorithm parameters: discount factor (γ), reward threshold (R_{limit}), and convergence threshold (ϵ) are set to 0.5, 0.001, and 0.000001 respectively to obtain the off-line coverage trajectory. The coverage region displays shallow islands and flat sandy surfaces. Our algorithm for selective coverage generates a trajectory such that the high-rewarding shallow islands are covered with a priority.

We used the generated off-line trajectory to guide the autonomous survey of the corals by the robot boat. Fig. 8 presents the actual GPS track of the robot boat performing the input trajectory from Fig. 7. The actual path followed by the boat has small discrepancies compared to the input trajectory. This is due to the non-holonomic nature of the boat and the controller parameters. Fig. 8 shows the complete GPS track of the robot in field experiment including the deployment phase on the shore.

C. Mosaic

We collected the visual data from the surface of the ocean with the robot boat performing the coverage pattern. We

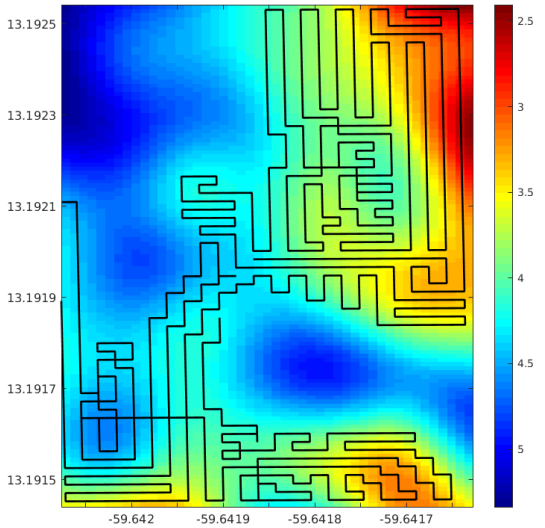


Figure 7: Selective Coverage trajectory (in black) overlaid on the depth map. Color-bar represents the depth.

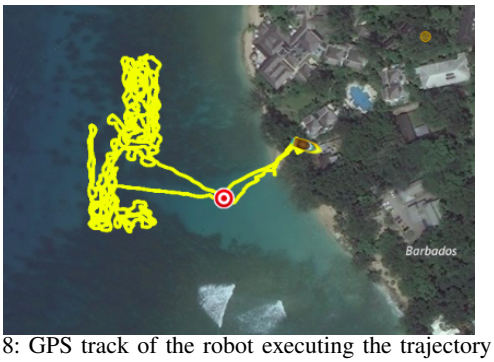


Figure 8: GPS track of the robot executing the trajectory from selective coverage. The figure also shows the deployment phase on the shore.

used the visual data from a transect of the boat trajectory to generate the mosaic presented in Fig. 9. Fig. 10 presents a collage of salient Geo-referenced visual frames extracted from the data collected to provide an overview. The trajectory is overlaid on the collage to indicate the mapping between the covered region and the Geo-referenced visual frames.

D. Performance Analysis

We compare the performance metrics (path length, trajectory execution time, and power consumption by the boat motor) of the selective coverage trajectory (Fig. 8) with the measurements from the dense lawnmower coverage (Fig. 2) to get a sense of ability to complete the task with limited battery on the robot boat.

Coverage Trajectory	**Path length** (m)	**Time** (s)	**Power Consumption** (Wh)
Selective Coverage	1612.78	2423	33.8472
Lawnmower	3119.4	4363	67.9356

Table I: Performance measurements for trajectories from selective coverage and lawnmower.

The Table I provides a comparison of the performance of the two coverage patterns. It illustrates the superior performance of value based mechanism described in this paper and allows sampling to occur over wider regions than otherwise possible with a limited power budget. The importance of this budget is exemplified by the fact that the robot boat ran out of battery before completing the coverage of the selected region using the dense lawnmower pattern (Fig. 2), as mentioned in Sec. IV. Hence, two iterations (i.e. with a fully charged battery) were needed to complete a single run. The comparison of the trajectory provided by our algorithm to the worst case lawnmower coverage shows that we can achieve the task of coral monitoring efficiently under the real world constraints, such as limited battery and execution time.

VII. Conclusion

In this paper we described an approach to surface-based reef monitoring using a 2-phase process of bathymetric mapping followed by visual data collection. In practice we expect the latter of stage to be conducted repeatedly. The first phase employs a Gaussian process and we show that this allows an acceptable depth map to be computed using fewer samples than would be need for naive methods. In the second phase, we exploit this bathymetric data along with a value based sampling methodology and show that, it allows a complete map to be computed much more efficiently than traditional methods such as repeated linear transects. We validated our approach in the open sea and produced a visual coverage map that illustrated the effectiveness of the proposed method.

For future work, we plan to explore the impact of our wind model and how it is related to the efficiency of the coverage pattern. This appears to be an useful aspect of our algorithm and is motivated by a real need, but a rigorous validation is both time consuming, challenging to execute and outside the scope of this paper.

VIII. Acknowledgement

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) through the NSERC Canadian Field Robotics Network (NCFRN).

References

[1] G. Matheron, "Principles of geostatistics," *Economic geology*, vol. 58, no. 8, pp. 1246–1266, 1963.

[2] R. N. Smith, Y. Chao, P. P. Li, D. A. Caron, B. H. Jones, and G. S. Sukhatme, "Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model," *The International Journal of Robotics Research*, p. 0278364910377243, 2010.

[3] S. L. Nooner and W. W. Chadwick, "Volcanic inflation measured in the caldera of axial seamount: Implications for magma supply and future eruptions," *Geochemistry, Geophysics, Geosystems*, vol. 10, no. 2, 2009.

[4] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Efficient complete coverage of a known arbitrary environment with applications to aerial operations," *Autonomous Robots*, vol. 36, no. 4, pp. 365–381, 2014.

[5] M. Meghjani, F. Shkurti, J. C. G. Higuera, A. Kalmbach, D. Whitney, and G. Dudek, "Asymmetric rendezvous search at sea," in *Computer and Robot Vision (CRV), 2014 Canadian Conference on*. IEEE, 2014, pp. 175–180.

[6] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 376–386, 2005.

[7] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.

[8] G. A. Hollinger, S. Choudhary, P. Qarabaqi, C. Murphy, U. Mitra, G. S. Sukhatme, M. Stojanovic, H. Singh, and F. Hover, "Underwater data collection using robotic sensor networks," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 5, pp. 899–911, 2012.

[9] S. A. Sadat, J. Wawerla, and R. Vaughan, "Fractal trajectories for online non-uniform aerial coverage," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2971–2976.

[10] Y. Girdhar, D. Whitney, and G. Dudek, "Curiosity based exploration for learning terrain models," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 578–584.

[11] M. Hugentobler, "Terrain modelling with triangle based free-form surfaces."

[12] A. KEMPPAINEN, T. MÄKELÄ, J. HAVERINEN, and J. RÖNING, "An experimental environment for optimal spatial sampling in a multi-robot system."

[13] B. Ferris, D. HŁhnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *In Proc. of Robotics Science and Systems*, 2006.

[14] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian process modeling of large-scale terrain," *Journal of Field Robotics*, vol. 26, no. 10, pp. 812–840, 2009. [Online]. Available: http://dx.doi.org/10.1002/rob.20309

[15] D. Nguyen-Tuong and J. Peters, "Local gaussian process regression for real-time model-based robot control," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept 2008, pp. 380–385.

[16] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.

[17] R. Bellman, "Dynamic programming princeton university press," *Princeton, NJ*, 1957.

[18] M. Meghjani and G. Dudek, "Search for a rendezvous with lost target at sea," in *ICRA Workshop on Persistent Autonomy for Aquatic Robotics*, 2015.

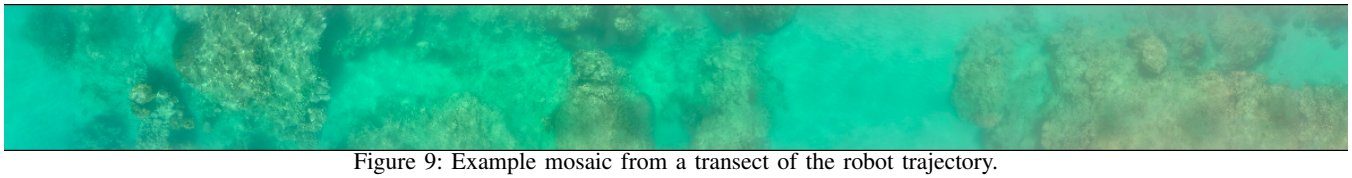


Figure 9: Example mosaic from a transect of the robot trajectory.

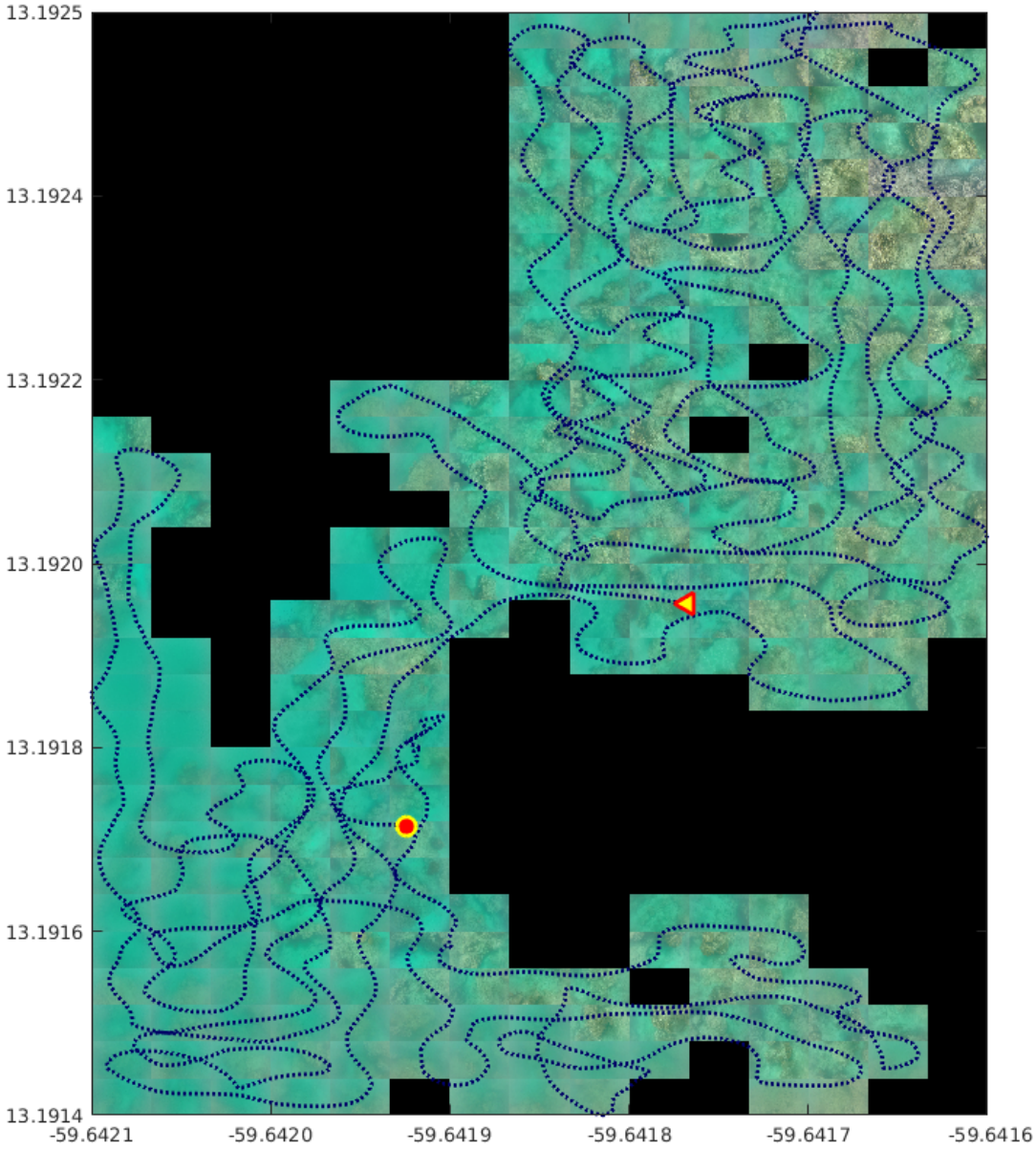


Figure 10: Selected images from the coverage. The dotted line on the image shows the coverage trajectory of the robot. Triangle and the circle indicate the start and end points of the path respectively.