2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

OAS-GPUCB: On-the-way Adaptive Sampling Using GPUCB for Bathymetry Mapping

Rajat Agrawal*, Karthik Nambiar*, Bhawana Chhaglani, Mandar Chitre and P.B. Sujit

***Abstract*— Bathymetry mapping of static water bodies like lakes is essential for sustainable ecosystem development strategies. However, bathymetry mapping using (i) traditional manual sampling approaches using single-beam echosounder (SBE) has large mapping errors and (ii) performing multi-beam survey is very expensive. Alternatively, performing lawn-mower SBE surveys is time consuming due to limited field-of-view. In order to address the above issues, in this paper, we present an on-the-way sampling approach with Gaussian Process Upper Confidence Bound (GPUCB) algorithm called as *OAS-GPUCB* that can adaptively sample the lake to minimize the bathymetry error while reducing the distance travelled to achieve a given mapping accuracy. We validate the proposed approach using simulations on actual lake bathymetry maps and also carry out real-world experiments using an Autonomous Surface Vehicle(ASV) with SBE. Further, we compare OAS-GPUCB to lawn-mower, GPUCB, and GPUCB with fixed radius approaches. The results consistently show that the proposed approach can achieve less than 10% bathymetry error while achieving distance reduction of more than 55%compared to the lawn-mower approach, and more than 90% less distance travelled compared to GPUCB and GPUCB with fixed radius approaches. The results shows the general applicability of OAS-GPUCB for bathymetry mapping of water bodies without any prior information maps.**

I. Introduction

Static water bodies like lakes form an important component in the Indian socio-economic life cycle. They provide the required water resources for agriculture and aquaculture that essentially drive the complete rural ecosystem. With a healthy water rate in the lakes, the natural ecosystem of flora and fauna flourish. These water bodies are also crucial for excess water discharge for industries and power plants. Even though these water bodies play a key role in the Indian economic and environmental system, they have not received the required attention to maintain their health, resulting in socio-economic loss, habitat loss, and loss of the water body.

In order to maintain healthy water bodies, we need lake bed (bathymetry) information. Obtaining this information using multi-beam surveys is expensive due to the cost of the vehicle, logistics for the operation and limited availability of these systems [1],[2],[3],[4]. Traditionally, manual bathymetry surveys are conducted by considering fewer samples using a hand-held GPS with a single beam echosounder (SBE)[5]. For example, Figure 1 shows the bathymetry map of Varthur Lake in Bangalore [6] where the black dots represent the sampling points. Using these sampling points with the Kriging interpolation method [7] bathymetry map is created. However, such maps have high levels of errors. One can use an autonomous surface vehicle (ASV) with SBE and GPS to obtain a complete map by performing a lawn-mower mission. However, the SBE has small sensor footprint, and hence requires very long mission time to accomplish mapping. Further, longer mission will increase the logistics and hence the mission cost. In this paper, we ask the following question, *how can we develop an ASV planner such that the distance travelled by the ASV is significantly less than the lawn-mower mission for a given percentage bathymetry error*. For example, the question for Figure 1 would be, devise a plan such that the bathymetry error is less than 10% while travelling less than distance travelled by the vehicle using lawn mower.

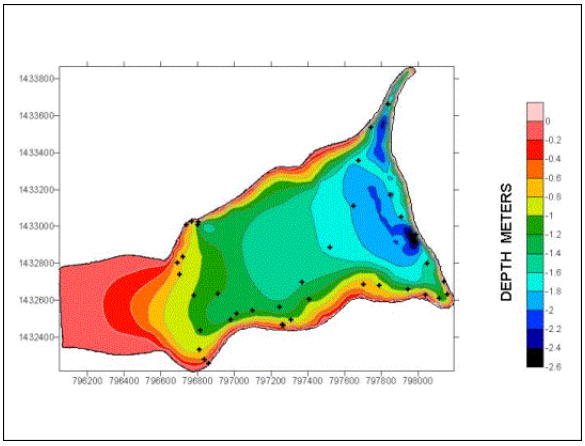


Fig. 1: Bathymetry map of Varthur lake in Bangalore [6]. • manual sampling points

One way to answer the above question is to employ adaptive sampling [8], [9]. Adaptive sampling is often considered nonparametric because it does not rely on predefined parametric models to guide the selection of sampling points. Instead, it adapts dynamically to the data collected during the sampling process, allowing for flexibility in decision-making without imposing rigid assumptions about the underlying system. As the lake bathymetry data is often unavailable, approaches based on prior information cannot be applicable[10], [11]. A common approach is to use a Gaussian process (GP) [12] as a nonparametric prior over smooth functions and also allows the decision maker to form a posterior distribution over the unknown function's values.

Rajat Agrawal, Karthik Nambiar, and P.B. Sujit are with the Department of Electrical Engineering and Computer Science, IISER Bhopal, Bhopal, India – 462038. Email: rajatagrawal1307@gmail.com {karthik23,sujit}@iiserb.ac.in.

Bhawana Chhaglani is a graduate student at University of Massachusetts Amherst, MA – 01003-9264. Email: bchhaglani@umass.edu

* Rajat Agrawal and Karthik Nambiar are equal contributors.

Mandar Chitre is with Department of Electrical and Computer Engineering, National University of Singapore, Singapore – 117583. Email: mandar@nus.edu.sg

The Upper Confidence Bound (UCB) [13], [14] algorithm, a multi-arm bandit methodology, was chosen with Gaussian processes (GPs) for adaptive sampling strategy because of its structured approach to balancing exploration and exploitation during sampling. By leveraging upper confidence bounds of expected rewards, UCB optimally guides the selection of sampling points, facilitating efficient exploration of the parameter space and enabling informed decision-making. Consequently, the GPUCB algorithm[15], [16], which iteratively selects the point with the highest upper confidence bound according to the posterior, achieves a theoretical no-regret guarantee[17], [18]. However, GPUCB relies on predefined confidence bounds and does not actively adapt to variations in depth, underwater features, or other environmental factors encountered during the survey. As a result, it may fail to optimize sampling trajectories leading to suboptimal data collection and prolonged survey times.

A. Related Work

Traditionally, bathymetric surveys adopt a pattern of perpendicular survey lines resembling a lawn mower, aligned with underwater slopes, with spacing adjusted to the desired resolution and scale. However, these may result in sparse point data, necessitating the utilization of interpolation methods as mentioned in [5], [19], [20] to fill gaps and ensure comprehensive spatial coverage of underwater terrain.

Mobile robotic systems, such as AUVs or ROVs, introduce dynamic sampling capabilities, allowing for adaptive and high-resolution mapping of underwater environments. Smith et al. [21] demonstrated the use of AUVs for adaptive sampling in oceanographic studies, employing strategies that dynamically adjust sampling path based on real-time data.

Gaussian Processes have been widely used for modelling environmental variables in spatial contexts, including bathymetry. Marchant et al. [22] applied Gaussian Process Regression (GPR) for modelling and predicting environmental fields. This is particularly useful in bathymetry for interpolating measurements taken at discrete points, providing a continuous estimate of seafloor topography. Wilson et al. [23] present an autonomous bathymetric mapping approach under depth constraints, employing real-time Gaussian Process modelling to track the intersection of a bounding polygon and depth contour, guiding path planning for systematic coverage and map construction. Also, in [24],[25] has examined selection strategies based on Bayesian optimization, where sampling relies on the Upper Confidence Bound (UCB), a linear combination of mean and uncertainty, to identify the maximum value using a Gaussian Process (GP). The adaptive sampling methods discussed above are not specifically tailored for bathymetry, potentially resulting in notable mapping inaccuracies when applied. Moreover, these techniques are not well-suited for single-beam echosounder (SBE) bathymetric surveys, as they may lack adequate coverage or resolution. Additionally, they may struggle to adapt to sparse and dynamic data, resulting in suboptimal sampling trajectories and reduced data collection efficiency.

B. Contributions and Outline

When using SBE for bathymetry mapping for the above approaches, the information is collected only at the sampling location given by the GPUCB. However, depth information can be collected for bathymetry applications as the vehicle moves. That is, the vehicle can perform on-the-way sampling. This aspect is missing in the literature, and we exploit this feature to increase mapping efficiency. The main contributions are summarized as follows.

- We introduce the On the way Adaptive Sampling GPUCB (OAS-GPUCB) algorithm to achieve accurate bathymetry estimates while minimizing travel distance within a specified mapping error threshold.
- Comprehensive evaluation showcasing the algorithm's capability to achieve less than 10% bathymetry error with less than 55% distance travelled compared to the lawn-mower approach.

This paper is structured as follows: Section II thoroughly discusses the problem statement. Section III delves into the Gaussian process and different adaptive sampling techniques, followed by Section IV, where OAS-GPUCB adaptive path planning is introduced. We showcase results from simulation and field experiments in Section V. Finally, Section VI gathers our conclusion and discusses future work.

II. Problem formulation

Consider an area ($L \times W$) to map, represented as a gird ($\mathcal{G}$) where each cell ($c_{ij}, i = 1, \ldots, n$ and $j = 1, \ldots, m$) is of size $l \times w$. Assuming that the vehicle obtains the depth information of a particular cell by visiting that cell, the vehicle needs to visit every cell in the grid for a complete mapping. The path taken by the vehicle can be represented as a set of waypoints, $\mathcal{W} = \{\omega_1, \ldots, \omega_N \ | \omega_k \in \mathcal{G}\}$ where the total distance travelled by the vehicle is $\ell = \sum_{k=1}^{N-1} ||\omega_k - \omega_{k+1}||$. Let V be the set of cells visited by the vehicle, U be the set of unvisited cells and d_{ij} be the depth information at cell c_{ij}. The mapping error is given by

$$\mathcal{E} = \frac{\sum_{i,j \in U} d_{ij}}{\sum_{i,j \in \mathcal{G}} d_{ij}}. \tag{1}$$

The objective is to determine paths for the vehicle such that for a given mapping error of γ, the path length travelled by the vehicle should be minimized,

$$\text{Objective: Minimize} \quad \ell \tag{2}$$

$$\text{Subject to:} \quad \mathcal{E} \leq \gamma. \tag{3}$$

One can use lawn-mower to visit all the cells to map a given area. However, this will not minimize the distance travelled for a given mapping error γ since it assumes that the depth information of adjacent cells are not correlated. As a certain degree of smoothness in depth is often warranted, it is safe to assume that the depth information of adjacent cells are correlated. Hence, a probabilistic framework can decrease the total distance travelled for a given mapping error.

III. Gaussian process modelling and Different Adaptive path planning techniques

A. Gaussian Process (GP)

Gaussian Process [12], [26] is a non-parametric probabilistic framework used to model complex systems and processes. It is defined as a probability distribution over possible functions, where any finite collection of function values has a joint Gaussian distribution. A Gaussian Process, denoted as $GP(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, is characterised by its mean function $\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and covariance function $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))]$.

We assume the prior distribution over f to be $GP(0, k(\mathbf{x}, \mathbf{x}'))$. Given observed data $\mathbb{D} = \{(\mathbf{x}_i, y_i) | y_i = f(\mathbf{x}_i) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma^2)\}_{i=1}^{t}$, the posterior distribution over f is a GP with mean $\mu_t(\mathbf{x})$, covariance $k_t(\mathbf{x}, \mathbf{x}')$ and variance $\sigma_t^2(\mathbf{x})$ where

$$\begin{aligned} \mu_t(\mathbf{x}) &= \mathbf{k}_t(\mathbf{x})^T (\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{y} \\ k_t(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_t(\mathbf{x})^T (\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_t(\mathbf{x}') \\ \sigma_t^2(\mathbf{x}) &= k_t(\mathbf{x}, \mathbf{x}) \end{aligned} \tag{4}$$

where $\mathbf{y} = [y_1, \ldots, y_t]^T$ is a noisy sample, $\mathbf{k}_t(\mathbf{x}) = [k_t(\mathbf{x_1}, \mathbf{x}), \ldots k_t(\mathbf{x_t}, \mathbf{x})]^T$ and $\mathbf{K}_t$ is the covariance matrix $[k(\mathbf{x}, \mathbf{x}')]_{\forall \mathbf{x}, \mathbf{x}'}$. In our problem formulation, $\mathbf{x}_i$ denotes a sampling point in the grid $\mathcal{G}$, and y_i represents the depth measurement corresponding to that point.

B. Upper Confidence Bound (UCB)

UCB [13], [14] algorithm is based on the principle of optimism in the face of uncertainty that uses uncertainty in the estimates to drive exploration. In UCB, the next sampling point x_t is determined by maximizing the acquisition function $h(\mu_t, \sigma_t)$.

$$\begin{aligned} h(\mu_t, \sigma_t) &= \mu_t + \sqrt{\beta}\sigma_t, \\ x_t &= \arg\max_{x \in \mathcal{G}} h(\mu_t, \sigma_t), \end{aligned} \tag{5}$$

where μ_t and σ_t denotes the mean and standard deviation of the distribution at time t and β denotes the exploration-exploitation trade-off parameter. Larger β produces exploratory sampling points, while smaller β favours exploitative ones.

C. Gaussian Process Upper Confidence Bound (GPUCB)

GPUCB [26] algorithm is used to optimize exploration-exploitation trade-offs in sequential decision-making tasks. It uses $GP(0, k)$ as a prior for f and at the next time step t, it uses the posterior GP distribution to compute the mean μ_t and standard deviation σ_t. Subsequently, GPUCB selects the next sampling point based on the acquisition function (Eq 5), iteratively refining the posterior GP distribution with each selection to inform subsequent decisions.

The GPUCB algorithm prescribes the process of iteratively choosing the sampling points according to the acquisition function, which dictates whether to prioritize exploration or exploitation. This leads to the generation of sampling points in a non-sequential random order, resulting in potential overlaps between previously visited paths. This inefficiency escalates the total distance travelled, undermining mapping efficacy.

Algorithm 1 GPUCB Algorithm

1: Input grid space $\mathcal{G}$; GP priors $\mu_0 = 0$, σ_0
2: Initial sampling point $\mathbf{x_0}$; $y_0 = f(\mathbf{x_0}) + \epsilon_t$,
3: $\mathcal{D}_0 \leftarrow \{\mathbf{x_0}, y_0\}$
4: **for** $t = 1, 2, \ldots$ **do**
5: Fit GP to $\mathcal{D}_{t-1}$
6: Perform Bayesian update to obtain μ_t and σ_t.
7: Choose $\mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{G}} \mu_t(\mathbf{x}) + \sqrt{\beta}\sigma_t(\mathbf{x})$
8: Observe $y_t = f(\mathbf{x_t}) + \epsilon_t$
9: $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{\mathbf{x_t}, y_t\}$
10: **end for**

D. Fixed Radius GPUCB (FR-GPUCB)

FR-GPUCB is a modified GPUCB algorithm where the sampling points generated from the acquisition function is limited within a given radius of current point. This fixed radius limitation of the acquisition function helps in reducing the overlapping of previously visited paths to some extend.

Algorithm 2 FR-GPUCB Algorithm

1: Input grid space $\mathcal{G}$; GP priors $\mu_0 = 0$, σ_0
2: Initial sampling point $\mathbf{x_0}$; $y_0 = f(\mathbf{x_0}) + \epsilon_t$
3: $\mathcal{D}_0 \leftarrow \{\mathbf{x_0}, y_0\}$; Radius r
4: **for** $t = 1, 2, \ldots$ **do**
5: Algorithm 1 lines 5 and 6
6: Find $\mathcal{G}_r = \{\mathbf{x} \in \mathcal{G} : \|\mathbf{x} - \mathbf{x_{t-1}}\| \leq r\}$
7: Choose $\mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{G}_r} \mu_t(\mathbf{x}) + \sqrt{\beta}\sigma_t(\mathbf{x})$
8: Observe $y_t = f(\mathbf{x_t}) + \epsilon_t$
9: $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{\mathbf{x_t}, y_t\}$
10: **end for**

E. On-the-way Adaptive Sampling GPUCB (OAS-GPUCB)

OAS-GPUCB takes all points lying between the current location and the sampling point generated by the acquisition function of UCB. Unlike GPUCB, where only the sampling points contribute to determining the posterior distribution, OAS-GPUCB considers all intermediate points visited en route to the sampling point. Consequently, at each iteration, OAS-GPUCB estimates the posterior distribution over f based on the collective evidence from both the sampling points and the intermediate waypoints. This inclusion of additional visited points aids in more accurately estimating the posterior distribution and alleviates path overlap by reducing uncertainty at in-between points.

We utilize the A* path planning algorithm [27] to determine the optimal route from the current location to the designated sampling point. Subsequently, the generated polyline path undergoes a pruning process, where it is replaced with a straight line at the path's inflection point. This refinement step serves to minimize travel distance even further.

Algorithm 3 OAS-GPUCB Algorithm

Input grid space $\mathcal{G}$; GP priors $\mu_0 = 0$, σ_0
Initial sampling point $\mathbf{x_0}$; $y_0 = f(\mathbf{x_0}) + \epsilon_t$
$\mathcal{D}_0 \leftarrow \{\mathbf{x_0}, y_0\}$
for $t = 1, 2, \ldots$ **do**
    Algorithm 1 lines 5 and 6
    Choose $\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{G}} \mu_t(\mathbf{x}) + \sqrt{\beta}\sigma_t(\mathbf{x})$
    $\mathcal{P} = \text{PrunedAStarPath}(\mathbf{x}_{t-1}, \mathbf{x}_t)$
    for each $\mathbf{x}_i \in \mathcal{P}$ **do**
        Observe $y_i = f(\mathbf{x_i}) + \epsilon_i$
        $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{\mathbf{x_i}, y_i\}$
    end for
end for

IV. Results

We evaluate the effectiveness of the lawn mower, GPUCB, FR-GPUCB, and OAS-GPUCB algorithms on various open-source bathymetry datasets coupled with real-world experimentation.

A. Results simulated on open-source bathymetry maps

We utilized GPUCB, FR-GPUCB (with radii set at 20%, 30%, 40%, and 50% of the map's width), and OAS-GPUCB algorithms with actual bathymetry maps of School-Section Lake[28],[29], Montana Lake[30],[31], and Waboose Lake[32]. These maps were rescaled, as shown in Figures 3a, 3b, 3c, with each pixel representing real-world dimensions obtained using ArcGIS Pro[33]. Specifically, pixel dimensions were 10.78m x 10.06m for School-Section Lake, 16.30m x 15.47m for Montana Lake, and 17.90m x 19.40m for Waboose Lake along the X and Y axes.

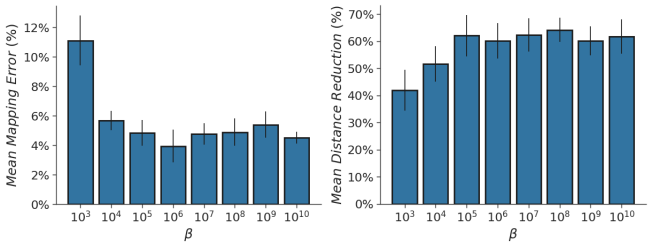
In Figure 2, the impact of β on OAS-GPUCB is depicted for all three lakes. From the simulation results, we can see that for β values 10^6 and 10^7, OAS-GPUCB have similar performance in terms of statistical significance and hence β value of 10^7 is considered in the rest of the experiments. Additionally, OAS-GPUCB achieves mapping errors of 3.53%, 4.88%, and 5.5% for Montana, Waboose and School-Section lakes, respectively, compared to the equivalent distance travelled by the lawn-mower pattern. This is significantly lower than GPUCB and FR-GPUCB algorithms.

Furthermore, OAS-GPUCB achieves a 10% mapping error with 70.62%, 60.41%, and 55.84% less distance travelled compared to the lawn-mower mission for Montana, Waboose, and School-Section lakes, respectively.

B. Experimental Results

We evaluate the complete functionality of the OAS-GPUCB through real-world demonstrations on an in-house ASV based on a single-hull kayak hull called "*Dolphin*" (Figure 4) in the Bhopal Lower Lake, India.

1) Dolphin vessel: The vessel is a monohull kayak of length 2400mm made from plastic. The vessel's width is 700mm, and the hull weighs 13 kg with a payload capacity of 75 kg. Equipped with T200 thrusters from Blue Robotics, it employs a differential drive mechanism. The vessel utilizes



(a) (b)

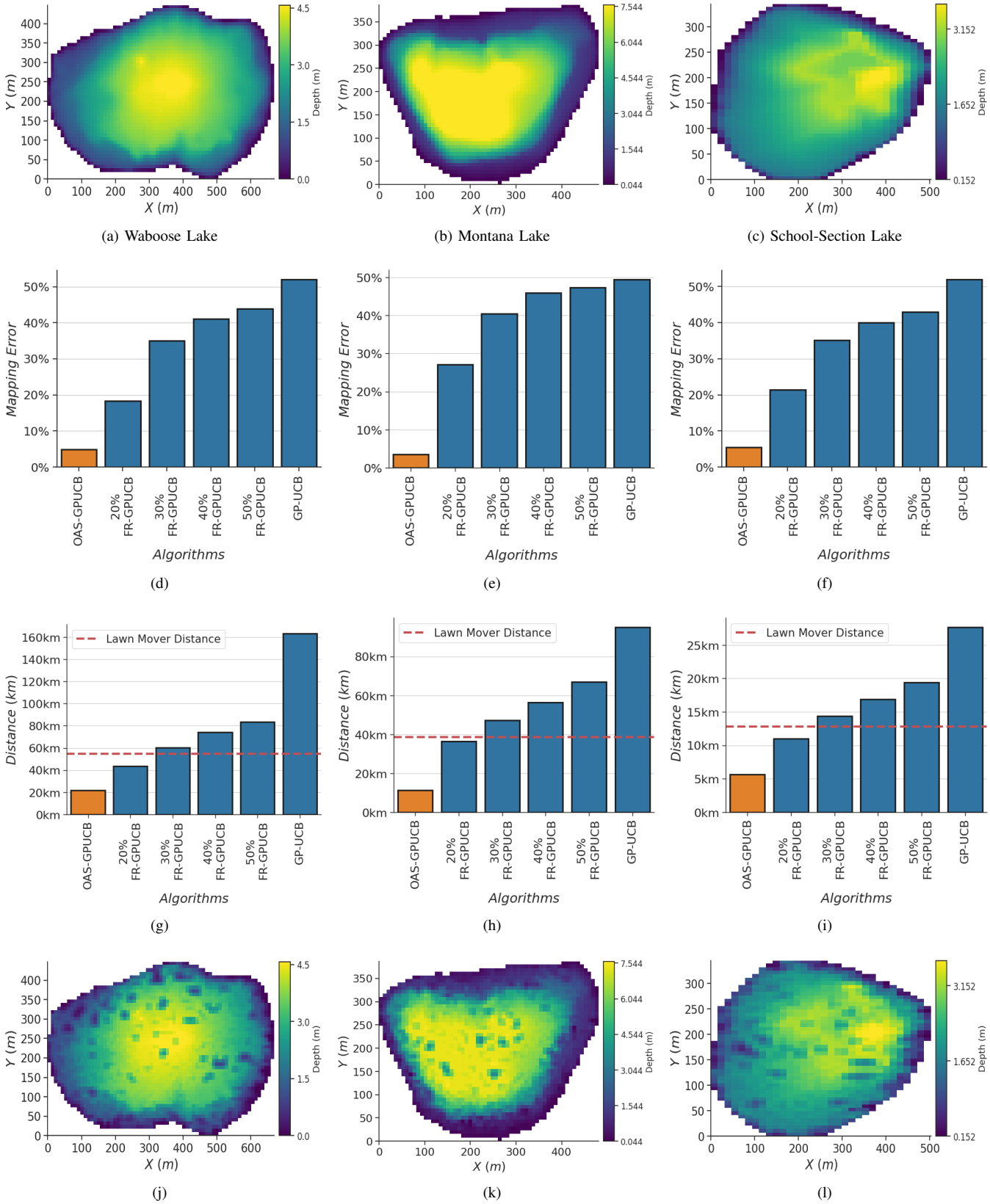
Fig. 2: Effect of β with (a) Mean Mapping Error (%) and (b) Mean Distance Reduction (%) across all three maps.

a Pixhawk 2.4.8 autopilot running ArduRover firmware and a Ping Sonar echosounder from Blue Robotics. An onboard computer – Nvidia Jetson Nano handles computation. The system employs RFD 868x radio modems for wireless communication between the base station and the Pixhawk. Additionally, Ubiquity 5 GHz Prism and Rocket with omnidirectional antenna facilitates communication between the onboard computer and the base station. The maximum speed of a Dolphin is about 2 knots.

2) Experiment: We consider a rectangular polygonal area within Bhopal Lower lake, India for experiments. This region was subdivided into cells based on length-to-width ratio, resulting in a segment measuring $51m \times 23.25m$, with cells of dimensions 29×13. Each cell was standardized to $1.761m$ in length and $1.788m$ in width. The ASV initially moved towards the grid origin, then navigated within it in accordance with the coordinates provided by the OAS-GPUCB algorithm. Ground truth data for high-resolution mapping was collected using a lawn mower with 1m lane spacing, covering 1485.49m as shown in Figure 4. The OAS-GPUCB algorithm covered a distance of 350.01 meters to reach a mapping error of 10%, representing a 76.44% reduction in distance compared to the lawn mower method. Additionally, after covering 892.95 meters, it achieved a mapping error of 1.54%, which corresponds to a 39.88% reduction in distance relative to the lawn mower approach, as shown in Figure 5.

V. Conclusions

This paper introduces a novel approach for efficiently mapping bathymetry in static water bodies using the (OAS-GPUCB) technique. The algorithm dynamically selects sampling points by integrating on-the-way sampling into the vehicle's trajectory towards waypoints, optimizing resource allocation and reducing mission time while maintaining map accuracy. Extensive simulation on open-source bathymetry lake datasets and real-world experiments on an ASV demonstrate the effectiveness of the OAS-GPUCB algorithm by comparing it with various other algorithms. Based on the current approach, we can extend the application of OAS-GPUCB to adaptive radius OAS-GPUCB, to improve its performance. Additionally, we can integrate these techniques into multi-agent systems to enable coordinated bathymetry surveys by utilizing uncorrelated information [34].



(a) Waboose Lake (b) Montana Lake (c) School-Section Lake

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 3: Comparison between the OAS-GPUCB approach and alternative algorithms on real lake bathymetry maps. Panels (a), (d), (g), (j) shows the results for Lake Waboose. Panels (b), (e), (h), (k) shows the result for Lake Montana. Panels (c), (f), (i), (l) shows the result for School-Section Lake. Panels (d), (e), (f) shows the mapping error achieved by each algorithm for traversing equivalent distance as the lawn mower pattern. Panels (g), (h), (i) shows the total distance travelled by each algorithm to achieve a 10% mapping error. Panels (j), (k), (l) shows the bathymetry map generated using OAS-GPUCB.

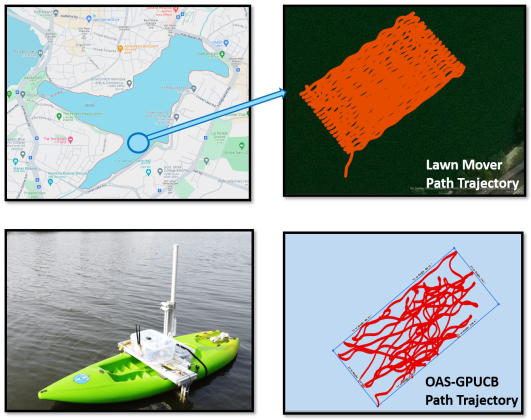


Fig. 4: Bhopal Lake real-time experimental ASV trajectories.

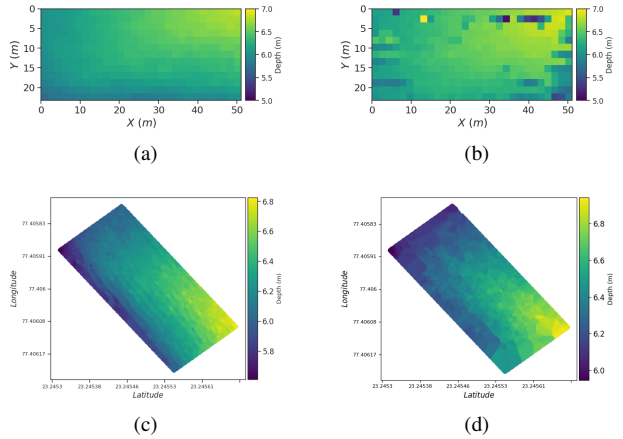


Fig. 5: Bathymetry maps generated through: (a) lawn mower mission, (b) OAS-GPUCB approach, (c) Kriging interpolation of the lawn mower mission, and (d) Kriging interpolation of the OAS-GPUCB approach.

REFERENCES

[1] C. De Moustier and H. Matsumoto, "Seafloor acoustic remote sensing with multibeam echo-sounders and bathymetric sidescan sonar systems," *Marine Geophysical Researches*, vol. 15, pp. 27–42, 1993.

[2] D. Pratomo, I. Saputro *et al.*, "Comparative analysis of singlebeam and multibeam echosounder bathymetric data," in *IOP Conference Series: Materials Science and Engineering*, vol. 1052, no. 1, 2021, p. 012015.

[3] I. O. Ferreira, L. C. d. Andrade, V. G. Teixeira, and F. C. M. Santos, "State of art of bathymetric surveys," *Boletim de Ciências Geodésicas*, vol. 28, no. 1, p. e2022002, 2022.

[4] A. Kenny, I. Cato, M. Desprez, G. Fader, R. Schüttenhelm, and J. Side, "An overview of seabed-mapping technologies in the context of marine habitat classification," *ICES Journal of Marine Science*, vol. 60, no. 2, pp. 411–418, 01 2003.

[5] A. I. EL-Hattab, "Single beam bathymetric data modelling techniques for accurate maintenance dredging," *The Egyptian Journal of Remote Sensing and Space Science*, vol. 17, no. 2, pp. 189–195, 2014.

[6] T. Ramachandra, N. Ahalya, and M. Payne, "Status of varthur lake: opportunities for restoration and sustainable management," CES, IISc, India, Tech. Rep., 2003.

[7] I. O. Ferreira, D. D. Rodrigues, G. R. d. Santos, and L. M. F. Rosa, "In bathymetric surfaces: Idw or kriging?" *Boletim de Ciências Geodésicas*, vol. 23, no. 3, p. 493–508, Jul 2017.

[8] L. V. Alvarez, H. A. Moreno, A. R. Segales, T. G. Pham, E. A. Pillar-Little, and P. B. Chilson, "Merging unmanned aerial systems (uas) imagery and echo soundings with an adaptive sampling technique for bathymetric surveys," *Remote Sensing*, vol. 10, no. 9, p. 1362, 2018.

[9] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric, "Adaptive importance sampling: The past, the present, and the future," *IEEE Signal Processing Magazine*, vol. 34, no. 4, 2017.

[10] W. S. Wijesoma, K. W. Lee, and J. Ibañez-Guzmán, "Motion constrained simultaneous localization and mapping in neighbourhood environments," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, pp. 1085–1090.

[11] M.-J. Rendas, I. Lourtie, and G. Pichot, "Adaptive sampling for sand bank mapping using an autonomous underwater vehicle equipped of an altimeter," in *ISESS*, 2003.

[12] C. Williams and C. Rasmussen, "Gaussian processes for regression," in *Advances in Neural Information Processing Systems*, D. Touretzky, M. Mozer, and M. Hasselmo, Eds., vol. 8. MIT Press, 1995.

[13] R. Sutton and A. Barto, *Reinforcement Learning, 2nd edition: An Introduction*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2018.

[14] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.

[15] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.

[16] S. R. Chowdhury and A. Gopalan, "On kernelized multi-armed bandits," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 844–853.

[17] D. Calandriello, L. Carratino, A. Lazaric, M. Valko, and L. Rosasco, "Gaussian process optimization with adaptive sketching: Scalable and no regret," in *Proceedings of the Thirty-Second Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, A. Beygelzimer and D. Hsu, Eds., vol. 99. PMLR, 25–28 Jun 2019.

[18] K. Wang, B. Wilder, S.-c. Suen, B. Dilkina, and M. Tambe, "Improving gp-ucb algorithm by harnessing decomposed feedback," in *Machine Learning and Knowledge Discovery in Databases*, P. Cellier and K. Driessens, Eds. Cham: Springer International Publishing, 2020.

[19] C. Parente and A. Vallario, "Interpolation of single beam echo sounder data for 3d bathymetric model," *International Journal of Advanced Computer Science and Applications*, 2019.

[20] S. K. Kartal, R. Hacıoğlu, K. S. Görmüş, Ş. H. Kutoğlu, and M. K. Leblebicioğlu, "Modeling and analysis of sea-surface vehicle system for underwater mapping using single-beam echosounder," *Journal of Marine Science and Engineering*, vol. 10, no. 10, p. 1349, 2022.

[21] R. N. Smith, A. Pereira, Y. Chao, P. P. Li, D. A. Caron, B. H. Jones, and G. S. Sukhatme, "Autonomous underwater vehicle trajectory design coupled with predictive ocean models: A case study," in *IEEE International Conference on Robotics and Automation*, 2010.

[22] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6136–6143.

[23] T. Wilson and S. B. Williams, "Adaptive path planning for depth-constrained bathymetric mapping with an autonomous surface vessel," *Journal of Field Robotics*, vol. 35, pp. 345 – 358, 2016.

[24] Y. T. Tan, A. Kunapareddy, and M. Kobilarov, "Gaussian process adaptive sampling using the cross-entropy method for environmental sensing and monitoring," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6220–6227.

[25] R. Marchant and F. Ramos, "Bayesian optimisation for intelligent environmental monitoring," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2242–2249.

[26] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.

[27] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[28] Wisconsin Department of Natural Resources, "School section," https://apps.dnr.wi.gov/lakes/lakepages/LakeDetail.aspx?wbic=107500.

[29] "School section bathymetry," http://www.bathybase.org/Data/1-99/6/.

[30] Wisconsin Department of Natural Resources, "Montana Lake," https://apps.dnr.wi.gov/lakes/lakepages/LakeDetail.aspx?wbic=518300.

[31] "montana lake bathymetry," http://www.bathybase.org/Data/1-99/56/.

[32] "Waboose bathymetry," http://www.bathybase.org/Data/200-299/243/.

[33] Esri, "ArcGIS Pro," Software, Year. [Online]. Available: https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview

[34] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009.