

Multi-Robot Informative and Adaptive Planning for Persistent Environmental Monitoring

Kai-Chieh Ma, Zhibei Ma, Lantao Liu, Gaurav S. Sukhatme

Abstract To gain a better understanding of environmental processes, we are interested in the problem of deploying multi-robot systems for efficient collection of environmental data. For long-term autonomy, enabling persistent monitoring, it is important to consider the spatio-temporal variations of environmental phenomena. We develop a multi-robot persistent path planning method that reduces uncertainty in the environmental model. Our framework contains two components: the first component computes potential observation points that minimize model prediction uncertainty, and the second component uses this for online planning of multi-robot paths, while also taking into account the efficiency of information collection. We validated our method via simulations, and the results show that it produces multi-robot routing paths that are conflict-free, informative, and adaptive to the environmental dynamics.

1 Introduction

We are interested in the problem of deploying multiple robots for efficient collection of environmental data, to gain a greater understanding of environmental processes. In particular, we are interested in reconstruction of physical, chemical or biological scalar fields. One example is the use of autonomous underwater vehicles (AUVs) for ocean monitoring, to map physical or biological properties of the ocean, such as temperature, salinity, and chlorophyll contents. Environmental monitoring is inherently a continuous and persistent task, because many of the underlying environmental processes vary both spatially and temporally. Therefore, in order to obtain a good estimate of the state of the environment at any time, robots need to perform persistent monitoring [15, 16].

All authors are with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089. E-mail: {kaichiem, zhibeima, lantao.liu, gaurav}@usc.edu

One aspect that sets apart persistent monitoring from conventional path planning methods, is that travel costs (e.g. travel time and distance) are not the only concern, because the robots are performing the task in a continuous, lasting manner. Instead, the objectives of a planning framework for multi-robot, long-term autonomy missions, are:

- **Maximization of information gain:** At any time, the robots’ observations along their paths can not cover the entire environmental space. We will need to model and predict continuous environmental phenomena with these limited observations, which inevitably causes uncertainty. Any planning approach should thus minimize model uncertainty, or equivalently, maximize information gain.
- **Multi-robot coordination:** Any paths planned for all robots should resolve potential conflicts. For example, two paths should avoid cross or transit the same location. Furthermore, each robot’s path should collaboratively optimize for the global objective, namely the collective informativeness of the model.
- **Adaptive and online routing:** The robots should be capable of adapting to the collected data. Given the spatio-temporal variability of the environmental fields, it is crucial that the paths are adapted as the robots progress. This requires online routing of the vehicles; dynamic goals and re-planning of paths.

We use Gaussian Process (GP) regression to model the phenomenon of interest [17]. To characterize the amount of information collected, we utilize the mutual information between visited locations and the remainder of the space [19]. This allows us to obtain a set of “most informative” future observation points. However, these observation points do not yet form a path (or multiple paths), because no routing information is provided. Many traditional path planning methods require all routing goals to be determined in advance. However, such goals are unrealistic for long-term autonomy path planning, because vehicles need to continuously visit infinite number of goals. Therefore, we extend an existing matching graph-based routing method [12], such that the routing destinations can be dynamically determined, and conflict-free paths can be adaptively computed, while taking into account the information gain.

2 Related Works

Planning methodologies designed for the spatio-temporal environmental monitoring are often called *informative path planning*, because the objective is to maximize the collected information (informativeness) [1]. Representative informative path planning approaches include approaches based on recursive-greedy path planning using mutual information on top of Gaussian Process regression [2, 15, 19], where the informativeness is generalized as submodular functions built on which a sequential-allocation mechanism is designed in order to obtain subsequent waypoints. Recently, a differential entropy based method was proposed, in which a batch of waypoints can be obtained through solving a dynamic program [3, 13]. However,

the framework is formulated with an assumption that the underlying map is transected (sliced) column-wise, so that each algorithmic iteration computes waypoints within a separate column and the navigation paths are obtained by connecting those waypoints among the pairwise adjacent columns. In recent works, we have extended such a framework by allowing the path to be searched and computed across the entire space at any stage [14]. In this work, we further extend our approach to persistent monitoring tasks for multi-robot systems.

Recent works that investigate informative path planning approaches for persistent ocean monitoring include [9, 16]. In [16], an active sensing based method was proposed, which uses a criterion that trades off between gathering the most informative observations for estimating unknown local regions, and predicting the phenomenon given the current estimates of those regions. To capture and adapt to the environment’s model dynamics, we plan paths within short time horizons. In [9], paths were also planned over short time horizons, using receding horizon planning. However, they used a different metric, and they did not consider multi-robot coordination.

Other related works include the *Orienteering Problem* (OP). The OP is a routing problem in which the goal is to determine a subset of nodes to visit, and in which order, such that the total collected score is maximized, and the given time budget is not exceeded [7, 8]. Heuristics have been designed to approximate this NP-hard problem in an efficient way [5, 6, 11]. However, one drawback of approaching this problem as an OP lies in that the time limit or cost can be hard to determine for long-term autonomy scenarios. In this paper, we extend an efficient matching graph based planning method by strategically integrating metrics of information gain and travel cost. We compare our method to a popular heuristic for the OP, and our results show that our method performs better for persistent multi-robot environmental monitoring.

3 Informative and Adaptive Planning Framework

Environmental phenomena vary not only spatially but also temporally. We regard the temporal process as a sequence of short horizons of equal length, and assume that within each short horizon the latent environmental phenomena are time-invariant. This allows us to eliminate the temporal parameter and focus on constructing the environment’s spatial properties, using existing methods from spatial statistics, such as Gaussian Processes (GPs). In this section, we explain how a set of potential informative observation points can be obtained from the GP. Following that, we construct routes over observation points using conflict-free paths. The observations along the paths are then used as a prior for generating a new set of potential observation points for the next time horizon. Note that these priors are time-varying, which means that the entropy (model uncertainty) of earlier observed points grows again gradually after the last observation. Therefore we need to be able to update the routing solution for each new horizon. Our observation point selection procedures thus repeat, and routes are updated, such that we carry out environmental monitoring persistently.

3.1 Gaussian Process Regression and Information Gain

We model the environment using Gaussian Process (GP) regression [17], similar to previous works [13, 19]. A GP’s behavior is specified by its prior covariance function (also known as *kernel*), which describes the relation between two independent data points. The GP is further defined by its hyperparameters, which can be estimated using training data, typically through maximum likelihood estimation [17]. In our implementation, we use the *squared exponential automatic relevance determination* kernel function. The mean and variance of each sample location can be predicted via the GP. The variance represents the uncertainty of the predicted data value, which can be used to find future observation points.

To assess prediction uncertainty, we use mutual information as a metric. In information theory, the mutual information is used to describe the mutual dependence between two variables. It is derived from the concept of entropy which is defined to quantify the uncertainty of random variables. For two arbitrary vectors of sampling points A , B , the mutual information between A and B can be expressed in terms of (conditional) entropy

$$I(Z_A; Z_B) = I(Z_B; Z_A) = H(Z_A) - H(Z_A|Z_B) \quad (1)$$

where Z represent random variables, $H(Z_A)$ is the entropy of Z_A , and the conditional entropy $H(Z_A|Z_B)$ can be calculated via

$$H(Z_A|Z_B) = \frac{1}{2} \log \left((2\pi e)^k |\Sigma_{A|B}| \right). \quad (2)$$

The conditional covariance matrix $\Sigma_{A|B}$ can be calculated from the GP’s posterior covariance matrix.

3.2 Generating Informative Observation Points

Let W denote the sampling set of the grid map, and let n be the desired number of observation points. The objective is to find a subset of sampling points, $P \subset W$ with a size $|P| = n$, which gives us the most information for our model. This is equivalent to the problem of finding observation points that maximize the mutual information between observed and unobserved locations of the map. The optimal subset of sampling points, P^* , with maximal mutual information is

$$P^* = \arg \max_{P \in \mathcal{X}} I(Z_P; Z_{W \setminus P}) \quad (3)$$

where \mathcal{X} represents all possible combinatorial sets, each of which is of size n . P^* can be computed efficiently using a *dynamic programming* approach [14].

The dynamic programming approach is as follows: Formally, let $w_i \in W$ denote an arbitrary sampling point at stage i and $w_{a:b}$ represent a sequence of sam-

pling points from stage a to stage b . Following Equation (9), the mutual information between P and the unobserved part at the final stage n can then be written as $I(Z_{w_{1:n}}; Z_{W \setminus \{w_{1:n}\}})$. This mutual information can be expanded using the chain rule:

$$I(Z_{w_{1:n}}; Z_{W \setminus \{w_{1:n}\}}) = I(Z_{w_1}; Z_{W \setminus \{w_{1:n}\}}) + \sum_{i=2}^n I(Z_{w_i}; Z_{W \setminus \{w_{1:n}\}} | Z_{w_{1:i-1}}). \quad (4)$$

One can utilize this form of mutual information to calculate w_i step by step. However, at every stage i before the final stage, the entire unobserved set $W \setminus \{w_{1:n}\}$ is not known in advance, therefore we make an approximation:

$$I(Z_{w_{1:n}}; Z_{W \setminus \{w_{1:n}\}}) \approx I(Z_{w_1}; Z_{W \setminus \{w_1\}}) + \sum_{i=2}^n I(Z_{w_i}; Z_{W \setminus \{w_1, \dots, w_i\}} | Z_{w_{1:i-1}}), \quad (5)$$

which can be formulated in a recursive form, i.e. for stages $i = 2, \dots, n$, the value $V_i(w_i)$ of w_i is:

$$V_i(w_i) = \max_{w_i \in W \setminus \{w_1, \dots, w_{i-1}\}} I(Z_{w_i}; Z_{W \setminus \{w_1, \dots, w_i\}} | Z_{w_{1:i-1}}) + V_{i-1}(w_{i-1}), \quad (6)$$

with the base case for this recursion: $V_1(w_1) = I(Z_{w_1}; Z_{W \setminus \{w_1\}})$. Note that the optimal waypoint in the last stage n is

$$w_n^* = \arg \max_{w_n \in W} V_n(w_n). \quad (7)$$

With the optimal solution in the last stage, w_n^* , we can backtrack all optimal sampling points (optimal with respect to the approximation made in Equation (9)) until the first stage w_1^* , and get the whole set of observation points $w^* = \{w_1^*, w_2^*, \dots, w_n^*\}$.

3.3 Planning Multi-Robot Paths among Observation Points

Given the most informative observation points, we can then plan the paths for each robot. Our path planning framework differs from traditional path planning methods in three ways: First, we need to plan paths for multiple robots, where each path starts from the robot's current location and ends at a unique destination, and these paths should not interfere (e.g. no intersection). Second, the metric for path quality is not only the travel distance/time (a minimization problem), but combined with information gain (a maximization problem). In our work, we evaluate the path quality via the *information gain in unit time*, i.e. the relative information gain given the time needed to collect such information. Third, the planning needs to adapt to the spatio-temporal dynamics. Observation points with time-varying priors are generated online, and the routing of paths needs to be able to adapt to such variations. We address these problems as follows.

Multi-Robot Conflict-free Path Planning: We use a graph $G = (V, E)$ to describe the possible paths between observation points V . Each point $v_i \in V$ is weighted by its information gain $\tau(v_i)$. Each edge $e_{ij} = (v_i, v_j) \in E$ is weighted by $w(v_i, v_j)$, the travel time between the two ending vertices. Motivated by the embedding of both vertex weight and edge weight as well as the capacity for describing multi-agent assignment, we opt to extend our routing method based on bipartite graphs (also called matching graphs) [12], to plan the multi-robot informative and conflict-free paths. In essence, the bipartite graph $\tilde{G} = (V, V', \tilde{E})$ is an augmented version of the standard graph $G = (V, E)$, if we regard it in the way that each vertex weight in G is uniquely transformed to some edge weight in \tilde{G} (such that all vertex weights are eliminated). Such a bipartite graph can well represent the matching (assignment) problem, and the optimal matching solution to it can be converted and interpreted as a routing path on the standard graph. We briefly describe the idea as follows, more details can be found in [12].

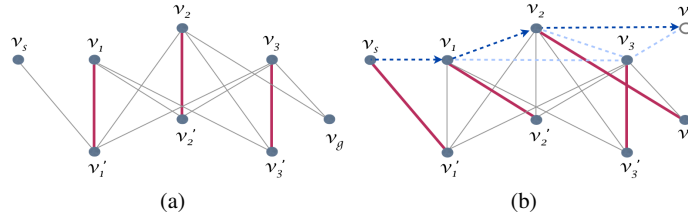


Fig. 1 Bipartite graph in the form of a 3D mesh, where $V = \{v_1, v_2, v_3\}$, $V' = \{v'_1, v'_2, v'_3\}$. The starting nodes (i.e. robots' current locations) are put in a set $V_s = \{v_s\}$; similarly, the goal nodes for each robot are in set $V_g = \{v_g\}$. In this example, we have only one start node and one goal node for each robot, and these are mutually exclusive. (a) Matched edges are in red bold, others are unmatched edges; (b) Optimal matching solution after running the Hungarian Method. The projected routing path is $v_s \rightarrow v_1 \rightarrow v_2 \rightarrow v_g$, the vertices of which are only in routing graph G . The path is illustrated by dashed arrows in the top layer.

A bipartite graph \tilde{G} has two sets of nodes, V and V' , where V' is simply a copy of $V \in G$ such that $|V| = |V'|$, and an edge $\tilde{e}_{ij} = (v_i, v'_j) \in \tilde{E}$ connects the vertices $v_i \in V$ and $v'_j \in V'$ if there is an edge $e_{ij} = (v_i, v_j) \in E \in G$. Edge $\tilde{e}_{ij} = (v_i, v'_j)$ is weighted the same as the counterpart edge $e_{ij} = (v_i, v_j)$, i.e., $\tilde{w}(v_i, v'_j) = \tilde{w}(v'_i, v_j) = w(v_i, v_j)$.

Figure 1 shows an example of bipartite graph. If we insert some starting vertices V_s and some goal/ending vertices V_g , a new matching problem is formed and we can employ the Hungarian Method [10] (with time complexity $O(n^3)$) to solve it. The output is a mapping that matches each $v_i \in V \cup V_s$ to a unique $v'_j \in V' \cup V_g$. The matched pair (v_i, v'_j) form a matched edge in \tilde{G} . Matched edges in Figure 1 are colored in red. To retrieve a routing path, all vertices on the matched edges except those in the set V' , form the path waypoints. For instance, in Figure 1(b), the path starting from v_s and ending at v_g is: $v_s \rightarrow v_1 \rightarrow v_2 \rightarrow v_g$. Note that, from multiple starting vertices V_s to multiple pre-specified goal nodes V_g , multiple paths can be obtained. Because each vertex can not simultaneously be on more than one matched

edge, the retrieved routing paths are conflict-free with no shared vertices. The paths do not cross or overlap due to the matching optimization mechanism [20].

Incorporating Informativeness: A useful property of the bipartite graph method is that paths can be tuned. This can be achieved by manipulating the weights $\tilde{w}(v_i, v'_i)$ of the corresponding *vertical* edges in Figure 1. The path tuning feature allows us to incorporate the information gain metric. Specifically, we set each weight:

$$\begin{aligned}\tilde{w}(v_i, v'_i) &= \lambda_i \tilde{w}_0(v_i, v'_i) \\ \lambda_i &= f(\tau(v_i, t)),\end{aligned}\tag{8}$$

where $\tilde{w}_0(v_i, v'_i)$ is initialized to be the minimum weight among all the outgoing edges, and $0 \leq \lambda_i \leq 1$ is a parameter for scaling the importance of the information gain versus the travel cost. λ_i is a function of $\tau(v_i, t)$, which is the information gain for vertex $v_i \in G$ at time t . Function f is empirically pre-defined to express how the raw information gain should be transformed to reflect the importance. For example, f can be a linearly increasing function. Intuitively, as λ increases, the paths become more winding and include more nearby informative observation points.

Adaptive Routing for Spatio-Temporal Dynamics: We want our path planning approach to be able to adapt to the spatio-temporal dynamics, and to handle online routing. Pre-defining routing goals for all future horizons is impractical, because the persistent monitoring task can be infinitely long. Instead, we want the planner to determine the goals online. We achieve this by further extending the above routing mechanism to address the online goal selection and path optimization problem. Specifically, we start by setting V as an empty set, V_s as the current locations of all robots, and V_g as all potential observation points, within the current time horizon. Then we solve the matching problem, which matches V_s to a set, say, $V'_g \subset V_g$. Note that this step is optimal only with respect to the one step planning horizon since it does not account for the future observation points. Therefore, we manipulate the sets by letting $V = V \cup V'_g$, $V_g = V_g \setminus V'_g$ and solve the new matching again. By repeating this process, vertices are incrementally moved from V_g to V , and the sequentially obtained vertices in V'_g form the routing paths, with the last waypoint at the end of each path as its routing destination. Algorithm 1 shows this incremental adaptive planning in pseudo-code.

4 Experimental Results

Experimental Set-Up: We validate our method through simulations, using the scenario of ocean environmental monitoring. The simulation environment is constructed as a two-dimensional ocean surface which is tessellated into a grid map. We use salinity and ocean currents data, observed in the Southern California Bight region, obtained via ROMS [18]. Figure 2(a) and 2(b) show visualizations of these data. The grid map resolution, as well as the hyperparameters of GP, are manually

Algorithm 1: Incremental Adaptive Path Planner

-
- 1 Given the starting locations of k robots $s = \{s_1, \dots, s_k\}$, the sampling waypoints $w = \{w_1, \dots, w_n\}$, and the planning horizon, h .
 - 2 Initialize, $V = \emptyset, V_s = s, V_g = w, V'_g = \emptyset$
 - 3 **for** $t = 1$ **to** h **do**
 - 4 $V = V \cup V'_g, V_g = V_g \setminus V'_g$
 - 5 Build a bipartite graph $\tilde{G} = (V, V', E)$, where $\tilde{e}_{ij} = (v_i, v'_j) \in E, \tilde{w}(v_i, v'_j) = d(v_i, v'_j)$, and the Euclidean distance $d(v_i, v'_j)$ represents travel cost.
 - 6 Parameterize V : $\tilde{w}(v_i, v'_j) = \lambda_i \min_{v'_j} \tilde{w}(v_i, v'_j)$
 - 7 Insert V_s, V_g to the above graph as starting/ending vertices, then solve the matching problem using the Hungarian Method.
 - 8 Let V'_g be the resultant matchings.
 - 9 Transform V'_g to k routing paths as described in Figure 1(b)
-

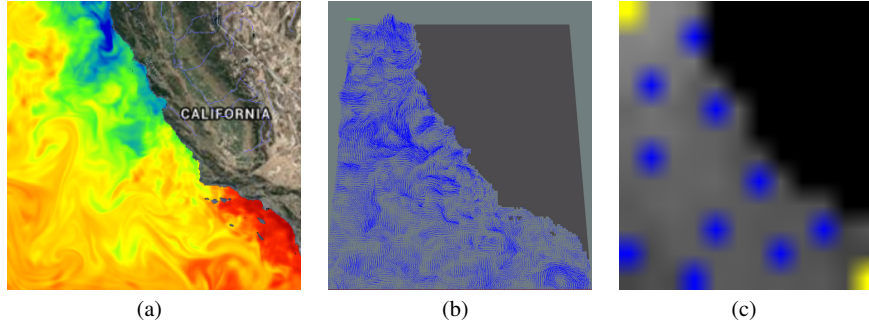


Fig. 2 (a) Ocean temperature field near southern California generated by ROMS. (b) Ocean currents predicted by ROMS. (c) Potential observation points (blue) with priors in the corners (yellow).

tuned and pre-set, such that approximately 30 observation points from the entire space can be generated and they can well cover the space. The robot used in simulation is an underwater AUV (marine glider).

We use a hierarchical model for motion planning, with two levels. At the higher level, the robots follow the planned paths presented in Sect. 3. At the lower level, the robots follow disturbance-aware motion policies, built on Markov Decision Process (MDP) (formulation details of the low level planner can be found in [14]). These motion policies let us integrate external disturbances (such as ocean currents) into the stochastic transition model. Therefore we take into account the robots' motion uncertainty caused by the ocean currents. By setting succeeding path waypoints as the short-horizon goal states, the low-level motion planner generates policies for local guidance.

Figure 2(c) demonstrates a set of 10 observation points that maximize mutual information in the map. In the figure, the black region represents land and the gray area represents the ocean. The yellow dots in the corners represent prior observation

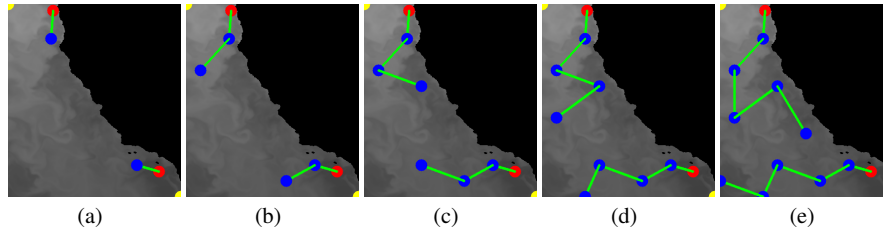


Fig. 3 Intermediate results of our multi-robot routing process. The routing points are incrementally and adaptively chosen during the process, as illustrated from (d) and (e).

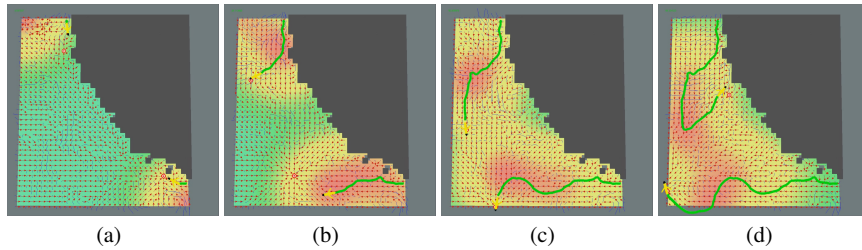


Fig. 4 Demonstration of environmental monitoring with 2 AUVs. The regions with warmer colors indicate less uncertainty (high confidence), whereas regions with colder colors indicate high uncertainty (low confidence). For the purpose of clarity, only one robot’s MDP policy map (small arrows) is shown.

points, and the blue blobs are the resultant observation points. With the observation points obtained, we run the path planner described in Algorithm 1 to generate the informative and adaptive paths for the multi-robot system. As shown in Figure 3, the routing paths are incrementally and adaptively augmented in each time step. Figure 3(d) and 3(e) show that the paths are adapted to avoid conflicts.

Results: Figure 4(a) through 4(d) shows the simulation results for two robots. Each robot follows its local decision-policy computed from an MDP model, combining both the ocean current disturbances and the reward information for the next waypoint. The colormap in these figures denotes the significance of uncertainty (red = low uncertainty, green = high uncertainty), from which we can see that the proposed method produces informative paths that explore and cover the regions with high uncertainty. As noted previously, the information gain is time-varying, i.e. the uncertainty of an observed point starts increasing again after a robot finishes its observation at this location and moves to somewhere else. Therefore, we incorporate the fact that the uncertainty of predictions increases as time elapses. Figure 4(c) and 4(d) show us that the earlier explored regions become uncertain again as time elapses, and that the robots always explore the most uncertain parts of the environment, as the environment changes.

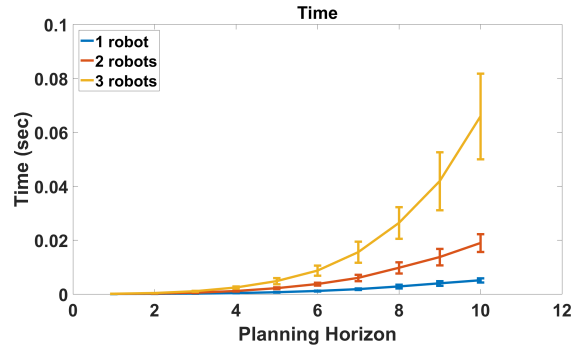


Fig. 5 Average computation time for different numbers of deployed robots and planning horizons.

Computational performance: We also evaluated the computational performance of our approach. Figure 5 shows the computation times given different numbers of deployed robots and planning horizons, as run on a computer with an 8-core 2.6GHz CPU and 12GB DDR3 memory. All statistics are mean values of 20 trials for each setting. In every simulation, 30 observation points are generated, and the prior data are randomly selected points. We can see that the computation time generally grows polynomially with the planning horizon increases, for each fixed number of robots. This can be justified by inspecting Algorithm 1. We can see that the bottleneck step is the Hungarian Method, whose time complexity is $O(n^3)$ and therefore the overall complexity is polynomial. The growth in computation time is mostly due to the generation of observation points. Table 1 shows the comparison of computation time between the generation of observation points, and the multi-robot path planning. Three robots are deployed and the planning horizon for path planning is set to cover as many observations points as possible. The informative observation point generation part is more costly due to the large search space; the observation point algorithm needs to evaluate all grid points in the grid map, whereas the path planning method only needs to compute routing solutions from the subset of obtained observation points.

	Stages = 5	10	15	20	25	30
Observation points	5.5752	12.8784	20.7266	28.6449	36.9960	47.2544
Path planning	0.0001	0.0007	0.0038	0.0070	0.0216	0.0659

Table 1 Computation time (sec) for two components of our informative path planning approach: finding informative observation points, and calculating a path between these.

To assess the quality of planned paths, we compare our method with an algorithm that solves the Orienteering Problem (OP). The OP solver aims at maximizing the collected score along the paths within some given time limits, thus it also considers

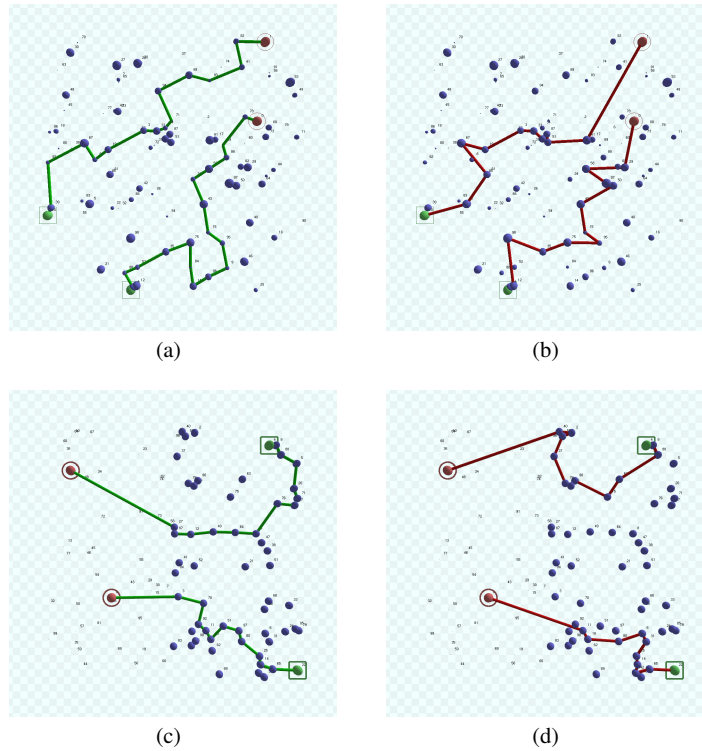


Fig. 6 Two different scenarios used to compare results between our method (green (a), (c)) and the OP algorithm (red, (b), (d)). Two robots are deployed. The circled nodes indicate their starting locations, and the squared nodes the ending locations. The size of each node reflects their significance of score. (a) and (b) compare performance on artificially created observation points, (c) and (d) compare performance on a skewed score distribution.

two competing metrics (score collection and travel time). We implemented a well-known heuristic called the *centre-of-gravity heuristic* [7], which combines other local refining heuristics such as the well-known 2-Opt heuristic [4]. One drawback of such OP solution lies in that both the ending vertices and the time limits must be specified. In contrast, our method plans the paths with their goals adaptively, in order to achieve better scores. To address the goal specification requirement for the OP, we first run our method and obtain the goals, then we assign these obtained goals to the centre-of-gravity OP. The time limits fed to the OP are the recorded time of each path computed from our method.

Figure 6 provides two sets of results for our proposed planning method (green paths) and the orienteering algorithm (red paths). The score for each vertex is $score_i \in [0, 100]$ and λ_i is set to be

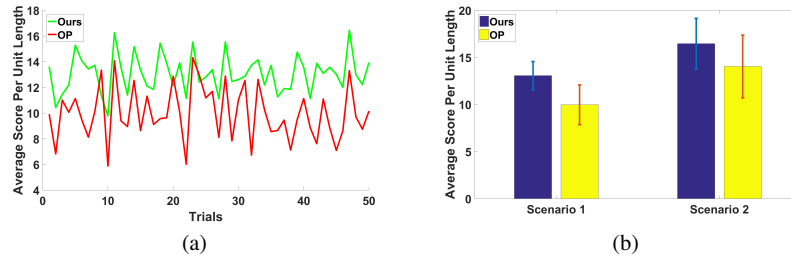


Fig. 7 The scoring performance comparison between our method and the OP solver: (a) average scores for 50 trials with uniformly distributed scores, (b) average scores for the scenarios shown in Figure 6.

$$\lambda_i = \begin{cases} score_i/100, & \text{if } score_i > 20 \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

Specifically, Figure 6(a) and 6(b) are planned paths from the two methods on the same set of artificially created observation points. The physical size of a node in the environment represents the significance of information gain (or score). We can observe that the paths produced from our method transit many high-score waypoints, whereas the *centre-of-gravity heuristic* transit fewer. Then, we manipulated the scores so that the score distribution is imbalanced, see Figure 6(c) and 6(d). We can see that our method can skip those low-score regions and transit only those high score nodes. Similar behavior can also be observed from the orienteering algorithm.

Next we compare these two approaches on their scoring performance. Figure 7 shows the detailed numerical results for 50 trials with randomly generated locations and scores. The y-axis is the average score collected, corrected for the path length. Figure 7(b) shows the average statistics of the two scenarios depicted in Figure 6, from which we can also conclude that our method is superior in terms of scoring performance.

5 Conclusions

In this paper, we presented an informative path planning approach for persistent multi-robot environmental monitoring. Taking into account the spatio-temporal variations of ocean phenomena, we first developed an information-driven component that computes the observation points, by minimizing the environmental model's prediction uncertainty. Multi-robot paths are then obtained by extending a matching graph based routing method, which allows the vehicles to transit the obtained informative observation points in an efficient manner. We validated our method through simulations with real ocean data. The results show that our method generates informative paths, which are conflict-free for multiple robots, and adaptive to the dy-

namics of the environment. Our approach is polynomial in the planning horizon, and linear in the number of robots. Furthermore, we have shown that our approach outperforms a well-known orienteering problem solver. Thereby we have developed an approach well suited for persistent monitoring with a multi-robot system.

6 Acknowledgments

The authors would like to thank Stephanie Kemna and Hordur Heidarsson for their valuable inputs on this paper.

References

1. J. Binney, A. Krause, and G. S. Sukhatme. Informative path planning for an autonomous underwater vehicle. In *International Conference on Robotics and Automation*, pages 4791–4796, 2010.
2. J. Binney, A. Krause, and G. S. Sukhatme. Optimizing waypoints for monitoring spatiotemporal phenomena. *International Journal on Robotics Research (IJRR)*, 32(8):873–888, 2013.
3. N. Cao, K. H. Low, and J. M. Dolan. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, pages 7–14, 2013.
4. A. Croes. A method for solving traveling salesman problems. *Operations Research*, 5:791–812, 1958.
5. F. V. Fomin and A. Lilngas. Approximation algorithms for time-dependent orienteering. *Inf. Process. Lett.*, 83(2):57–62, July 2002.
6. M. Gendreau, G. Laporte, and F. Semet. A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, 106(2-3):539–545, 1998.
7. B. L. Golden, L. Levy, and R. Vohra. The orienteering problem. *Naval Research Logistics (NRL)*, 34(3):307–318, 1987.
8. A. Gunawan, H. C. Lau, and P. Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315 – 332, 2016.
9. G. Hitz, A. Gotovos, M.-É. Garneau, C. Pradalier, A. Krause, R. Y. Siegwart, et al. Fully autonomous focused exploration for robotic environmental monitoring. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2658–2664. IEEE, 2014.
10. H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistic Quarterly* 2:83–97, 2:83–97, 1955.
11. G. Laporte and S. Martello. The selective travelling salesman problem. *Discrete Applied Mathematics*, 26(2):193 – 207, 1990.
12. L. Liu and D. A. Shell. Physically Routing Robots in a Multi-robot Network: Flexibility through a Three Dimensional Matching Graph. *ijrr*, 32(12):1475–1494, 2013.
13. K. H. Low, J. M. Dolan, and P. Khosla. Active markov information-theoretic path planning for robotic environmental sensing. In *Proceedings of the 10th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS-11)*, pages 753–760, May 2011.
14. K.-C. Ma, L. Liu, and G. S. Sukhatme. An information-driven and disturbance-aware planning method for long-term ocean monitoring. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

15. A. Meliou, A. Krause, C. Guestrin, and J. M. Hellerstein. Nonmyopic informative path planning in spatio-temporal models. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, pages 602–607, 2007.
16. R. Ouyang, K. H. Low, J. Chen, and P. Jaillet. Multi-robot active sensing of non-stationary gaussian process-based environmental phenomena. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, pages 573–580, 2014.
17. C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
18. A. F. Shchepetkin and J. C. McWilliams. The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling*, 9(4):347–404, 2005.
19. A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin. Efficient planning of informative paths for multiple robots. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07*, pages 2204–2211, 2007.
20. M. Turpin, N. Michael, and V. Kumar. Capt: Concurrent assignment and planning of trajectories for multiple robots. *International Journal of Robotics Research*, 33(1):98–112, 2014.