Confidence-guided path planning for mobile sensors

Damla Turgut*, O. Patrick Kreidl†, Ayan Dutta† and Ladislau Bölöni*
*University of Central Florida, Orlando, FL, USA
† University of North Florida, Jacksonville, FL, USA

***Abstract*—This paper introduces Confidence Guided Path-planning (CGP), an algorithm for planning the path of mobile sensor nodes with the goal to increase confidence in the accuracy of the estimated model at any time point in the data collection process. The approach employs a local estimator based on a Gaussian process regressor and takes advantage of the uncertainty estimation to guide the sensor to areas of lower confidence. In an experimental study comparing CGP with systematic lawnmower-type exploration and random waypoint movement, we found that CGP achieves better scores than both during most of the exploration process, being outperformed only by a fully completed systematic exploration. We also found that, as an emergent property of pursuing higher confidence, CGP achieves good coverage of the area of interest. The proposed algorithm has wide applications in precision agriculture, wildlife tracking, and road monitoring, where exhaustive coverage is not feasible.**

***Index Terms*—mobile sensor, path planning, algorithm, Gaussian Process**

I. Introduction

Recent advancements in mobile sensing technology, including the use of drones, have enabled the acquisition of more extensive and precise data for various applications, such as precision agriculture, road monitoring, and wildlife tracking [1], [2]. However, covering the entire area of interest is often infeasible. Instead, selective sampling of observations and the use of estimation techniques are required to model the environment. Traditional approaches to path planning for mobile sensors have focused either on systematic techniques that optimize coverage of a geographic area or random waypoint-based techniques that approximate the random sampling of the system generating the observations.

The ultimate aim of mobile sensing is to comprehensively understand the environment by creating a model that characterizes it accurately. The model we seek can often be implemented as a scalar field that provides a numerical value for every point in the geographical area. This model must be constructed from the observations provided by the sensors. Both the path planner and the estimator contribute to the quality of the model. The path planner dictates which observations are made and when during the exploration process. The estimator can enhance its accuracy by deploying sophisticated probabilistic modeling techniques that consider the dynamics of the underlying phenomena. A frequently used, high-quality, albeit computationally expensive algorithm is based on Gaussian process regression.

Path-planning techniques, often designed independently of the estimator, are frequently based on a surrogate objective that doesn't rely on the estimator or the collected data. For numerous algorithms, this surrogate metric is *coverage* [3]. Algorithms systematically covering the area of interest, such as those following a lawnmower pattern, benefit from predictability. Once the trajectory is complete, a high-quality result can be expected. However, a disadvantage of such systematic exploration is that certain areas may remain completely unexplored until the exploration process concludes. In many applications, it is advantageous to initially acquire a rough model of the entire area of interest and iteratively, refine it. Therefore, comparatively simple algorithms like random waypoints can often outperform more systematic strategies in the early stages of exploration. However, these techniques do not interact with the state estimator, nor do they take into consideration the observations that have been made to date by the sensor. One advantage of this approach is that, for both methods, it's possible to pre-calculate the path of the mobile sensor.

The work in this paper starts with the conjecture that a better accuracy can be obtained by a path policy that takes into consideration, in real-time, the observations that have been made to date, as well as an understanding of the needs of the estimator used.

To achieve this, we propose an algorithm called Confidence Guided Path-planning (CGP). CGP uses a local model of the environment and constantly reevaluates the model based on up-to-date observations collected by the robot sensor. The local model uses a Gaussian process-based estimator, which provides not only an estimate of the measured values at each point in the environment but also the associated confidence value. Based on these confidence values, the mobile sensor chooses its next destination such that it will investigate areas where it has the least confidence. CGP employs several additional techniques to enhance the efficiency of the sensing process. We demonstrate that compared to traditional techniques such as systematic or random waypoint-based exploration, CGP achieves a significantly better score earlier in the exploration process, while leaving fewer "blind spots" than random waypoint.

The main contributions of this paper can be summarized as follows:

- We propose and describe Confidence Guided Path-planning (CGP), an online, real-time algorithm for path planning that uses a local estimator to find paths that improve the confidence of the model.
- We investigate the emerging properties of the algorithm, such as the shape of the emergent path, its ability to cover

This work is supported in part by NSF CPS Grants #1932300 and #1931767.

the area, and its avoidance of self-intersecting paths.

- We compare CGP with two important baseline algorithms in mobile sensor path planning (systematic exploration and random waypoint), and show that it outperforms both of them in achieving a better accuracy score earlier in the process.

II. Related Work

The use of mobile sensors to explore an environment has been extensively investigated in multiple fields, with a variety of instances of mobile sensors, such as UAVs [4], UGVs, other terrestrial robots, drone ships [1], and autonomous underwater vehicles [5], [6], [7]. We want to emphasize that sensing is not the only task a robot might execute in a sensor network [8] - other alternatives might include data mulling, recharging nodes, and tasks that the robots might perform in addition to or instead of the sensing task.

In cases where the goal of the mobile sensor is to gather information about the environment, the natural optimization criteria are the accuracy of the model or the model's suitability for certain tasks. In practice, however, many systems use some type of surrogate optimization criteria, such as the coverage of observations or the amount of non-duplicated data collected. The coverage problem [9], sometimes referred to as the orienteering problem [10], is closely related to the traveling salesman problem and, as such, is NP-complete. In certain applications, coverage is reformulated as a graph traversal problem. In other circumstances, the path is designed to account for the various shapes and topologies of the geographical area under consideration, providing variations of the basic lawnmower pattern [3].

Chen et al. [11] propose an approximation algorithm inspired by density-based clustering methods for the coverage path planning of a bounded number of regions. This algorithm is then compared to the optimal solution obtained by formulating an integer linear programming problem. Special consideration must be given to the case of multiple explorer robots. Hu et al.[2] suggest an approach in which multiple UAVs use a Voronoi partition of the area to divide the regions to be covered, with strategies for covering individual areas then learned using deep reinforcement learning.

Other researchers, similarly to our approach, have considered the fact that observations made by the sensor will be processed by an estimator. The use of a Gaussian process as an estimator was proposed by Guestrin, Krause, and Singh [12]. With this model, the selection of observation points can be considered as the problem to maximize mutual information (a problem that is NP-complete by itself, even without considering the search for an efficient path through these points). A recursive greedy algorithm for a graph theoretical formulation was proposed by Chekuri and Pál [13].

III. Confidence guided path planning

A. Design principles

The algorithm that we are proposing is based on three principles: a) online decision-making, b) observation dependence, and c) computation budgeting.

First, *online decision making* means that the mobile sensor makes decisions during run-time, as opposed to *offline algorithms* that pre-plan the trajectory at the beginning of the scenario. Generally, coverage-optimization algorithms for a known environment can be executed offline. Interestingly, algorithms such as random waypoints are often described as though, upon reaching a waypoint, the sensor randomly generates the next one. This mode of description obscures the fact that the path can be generated ahead of time, as it does not depend at all on the state of the mobile sensor. Thus, the random waypoint path can be conveniently pre-generated ahead of time and followed during the execution time. In the case of true online algorithms, the decision process depends on real-time information that is not available offline.

Second, *observation dependence* means that the path of the mobile sensor depends on the observations received. Thus, the sensor node adapts the next waypoint based on the information it has sensed up to the current point. Informally, the sensor node aims to maximize the value of the information to be sensed in the future based on the information collected in the past. In general, such an optimization can only be performed probabilistically, as the sensor does not have access to its future observations. The particular type of observation dependence we propose for this algorithm is based on the calculation of a *confidence map*. This map calculates the uncertainty in the predicted environment values given the current set of observations and schedules the future movement of the sensor aiming to reduce this uncertainty.

Third, the algorithm utilizes *computational budgeting*. We design the proposed algorithm to be a practically deployable system, capable of working on a large scale. As we define the objectives of the algorithm as abstract optimization problems, we need to be aware that the decisions depend on computationally expensive sub-problems and naïve solutions are unfeasible. For instance, a naïve solution could involve calculating the confidence at each step on a specific grid $g \times g$ and finding the path that optimally visits the points with the lowest confidence. However, such an algorithm would be impractical. Just the calculation of the confidence values using a Gaussian Process has a computational complexity of $O(o^3 + g^2)$, with o being the number of observations that increases at every timestep. Meanwhile, the path visiting the low confidence points is NP-complete with regard to the number of points planned to be visited. Even if we could somehow execute these computations, we would need to recalculate them when the next observation comes in. This implies a significant amount of wasted computation, which, depending on the setting, might need to be executed onboard a mobile sensor with limited computational and energy resources. In conclusion, the proposed algorithm must carefully budget what computation it performs and at what moment in time, to achieve reasonable performance.

B. Gaussian process regression for a scalar field

A Gaussian process (GP) is a statistical model of a series of random variables, with the assumption that any subset of them follows a multivariate normal distribution. A common application of a GP is for Gaussian process regression in a scenario in which the random variables are correlated by their spatial distribution in a 2D space. Some of the values (termed *observations*) are known, while the other values need to be estimated using the inference process of the GP (points we will refer to as *queries*). This technique is closely related to the technique called *kriging* in geostatistics, where it has been used to identify the location of mineral resources from samples.

In scenarios where a mobile sensor collects information in an environment, this approach is typically applied as follows: the points where the sensor collected data are the observations, while all the other points on the observation grid are queries. In such scenarios, we can usually assume that there is a correlation between nearby observations. This observation is captured by the use of a kernel function $K(\mathbf{x}, \mathbf{x}')$ that captures the covariance between two arbitrary points in the scalar field. The results of the query are expressed in the form of a normally distributed random variable, whose mean represents the predicted value, while the standard deviation is associated with the degree of uncertainty in the estimate, or conversely, with the confidence of the prediction.

C. Evaluating the confidence

We are considering an area A that includes a collection of points of interest $p_i = (x_i, y_i)$. For ease of exposition, we will assume for the rest of this paper that this area is a rectangular region of size $h \times w$ with the points of interest laid out as a grid. However, the algorithm does not require such a regular layout.

At the time t, the mobile sensor makes an observation at its current location $p = (x, y)$, which we will assume is a real number $o = E(x, y, t) \in \mathbb{R}$. This notation assumes that the observation reads a value from an environment represented by the tensor E. Over the course of its trajectory, the mobile sensor will collect a series of observations at various locations and times. In most practical circumstances, the sensor will not be able to collect data from all points of interest. The observations are used by an estimator $\theta(O) \rightarrow I$ to create an *information model* I that has the same structure as the environment E. Let's consider that we are at the time point t_{now}, with the set of observations $O = o_1, \ldots, o_{mt_{now}}$. The calculation of the information model $I = I(O)$ allows us to read out an estimated, probabilistic value for any metric and any location of interest at the current time $I(x_i, y_i, t_{now}, m) \approx E(x_i, y_i, t_{now}, m)$.

One question this paper addresses is: how confident are we in the estimate for a given point of interest? If we have a recent measurement at exactly that point, our confidence is limited only by the quality of the instruments. However, if we are inferring this value at a location where we don't have a measurement, our confidence depends on how far away these measurements are, how well they agree with each other, and prior knowledge about the variance of the scalar field. Gaussian process regression provides a rigorous way to simultaneously provide such an estimate along with a confidence value, based on a reasonable assumption that the underlying model follows a normal distribution with dependencies between nearby models modeled by a kernel function.

For the implementation of the CGP algorithm, we need to differentiate between the system estimator, which creates an estimate for the beneficiary of the sensor network, and the local, online estimator that is run by the node during execution and is used for path planning. Generally, we assume that the system estimator has adequate computing resources and time to complete complex calculations. The online estimator, however, needs to operate under assumptions of scarcity. Furthermore, the system estimator is only invoked when the beneficiary intends to query the estimate, which typically happens once at the end of the data collection session. In contrast, the online estimator is invoked every time the mobile sensor needs to make a movement decision.

D. Movement decisions in CGP

To reduce the computational cost of the CGP path planner to an acceptable level, CGP makes two design decisions:

- **Commit to a waypoint**. At the first invocation, CGP chooses and commits to a waypoint. Until that waypoint is reached, the path planner reverts to a simple waypoint-following algorithm, without recalculating the confidence estimate at every moment. This approach not only reduces the cost of computation but also improves the stability of the algorithm and the smoothness of the trajectories.
- **Neighborhood model**. When a decision about the next waypoint needs to be made, CGP calculates an online model using the local estimator. The model is only sampled for a set of candidate waypoints that are generated from a specific neighborhood *span* of the current position. The next waypoint is chosen as the point with the lowest uncertainty (highest confidence), with ties broken randomly.

The overall structure of the CGP path planner is described in Algorithm 1. A distinguishing feature of this algorithm is that, as opposed to coverage planning algorithms that are explicitly taking into consideration the geometry of the area of interest, this algorithm proceeds in an essentially greedy fashion toward the areas with maximum uncertainty in the immediate neighborhood weighted by the importance of the area. Yet, as we shall see in the experimental results, such as Figure 2, coverage is an emergent property of the presented algorithm.

IV. Experiments

A. Experimental framework

To investigate the properties of the CGP algorithm, we compared it in a series of experiments using the Waterberry

Algorithm 1: Confidence Guided Path Planning

Input : t, x, y, o, $span$
Output: action
$O \leftarrow O \bigcup o$
if *x, y != next_waypoint* **then**
| **return** move towards next waypoint
end
I $\leftarrow$ calculate online model I(O)
feasible_wps $\leftarrow$ generate_waypoints(x,y,$span$)
min_confidence $\leftarrow$ lowest $I[wp]$ value from feasible_wps times importance value of location
min_confidence_wps $\leftarrow$ feasible_wps with min_confidence values
next_waypoint $\leftarrow$ randomly select a waypoint from min_confidence_wps
return move towards next_waypoint

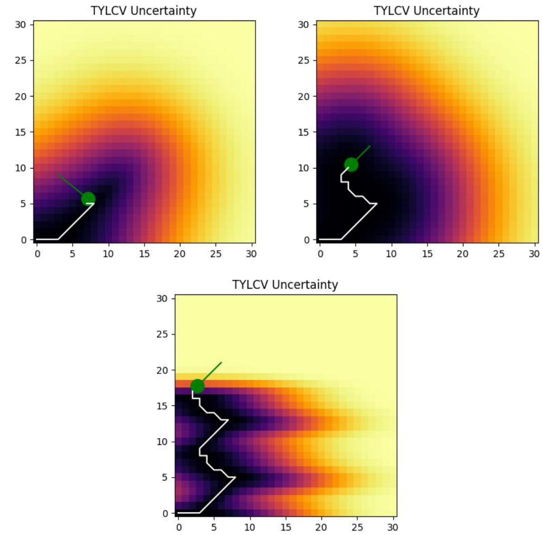

Fig. 1. The emergent behavior of the GCP path planner with $span = 5$ plotted at $t = 12$, $t = 20$ and $t = 32$. The background shows the uncertainty level in the observation, as returned by the local GP estimator. The white path shows the trajectory of the robot, while the green line the planned future trajectory.

Farms (WBF) benchmark framework [14]. This framework provides a way to compare the performance of mobile nodes sensing an area modeled on a precision agriculture scenario with a geographically complex area, partially planted with tomatoes and partially with strawberries. For the purposes of this paper, we used the MiniBerry-30 benchmark, which considers a rectangular area of 30 x 30m, half of it planted with strawberries and the other half with tomatoes, with a time budget of 400s. We assumed that the mobile node starts at the location (0,0), and moves at a velocity of 1 m/s. We only considered the observations concerning outbreaks of the tomato yellow leaf curl virus (TYLCV), which only affects tomato plants. The WBF framework provides observations to the mobile robot, simulates the commands provided by the path planner, and provides a collection of sample estimators.

As the scores measured by WBF are calculated based on the accuracy of the created model, the benchmark always compares *pairs* of path planners and estimators. For the purposes of this paper, we used the same estimator for all the path planners compared, thus any difference in performance is due to the path planner.

B. Baseline path planners

To evaluate the performance of the CGP algorithm, we compare it to two baselines that represent important classes of mobile sensor movement algorithms.

The **fixed budget lawnmower (FBLM)** plans a uniform, lawnmower coverage of the area choosing the densest possible pattern that will still fit in the time budget. Note that for this simple shape of the area of interest, this is an optimal coverage path. FBLM is an offline path planning technique, which requires initial knowledge of the shape of the area and the time budget.

The **random waypoint algorithm (RW)** plans a path where the mobile sensor moves to successive waypoints, chosen in a uniform random manner from the area of interest. In this implementation, this is also an offline algorithm, as the choice of the waypoints can be done ahead of time, leading to a random, but pre-calculated path. The planning needs information about the shape of the area, but not about the time budget.

Against these two baselines, we compared two parametrizations of CGP, with $span = 5$ and $span = 10$ respectively, denoted as GCP-5 and CGP-10.

C. Experimental results

In the first experiment, we investigated the shape of the trajectory generated by CGP. This shape is not intuitively obvious. For FBLM, the lawnmower pattern is explicitly programmed into the algorithm, similarly, RW creates straight-line traversals between random points. However, for CGP the shape of the path is *emergent* as the algorithm itself does not make any geometry-based decisions. Figure 1 shows the behavior of CGP-5 starting from a completely unknown environment for the first 32 timesteps, plotted at $t = 12$, $t = 20$, and $t = 32$. We notice that the emergent behavior of the algorithm leads to a zig-zagging coverage shape. It is easy to see that the confidence guidance pushes the robot to explore new areas. Another observation is that while there is nothing in the algorithm that explicitly prevents the robot from crossing its own path, this will rarely happen, as in general, the robot will aim towards areas that had not been visited before. Another observation is that with $span = 5$, the farthest the algorithm plans ahead is 5 steps, which is also the length of the longest straight line sequences in the trajectory.

Figure 2 shows the results of the experiments comparing the models built by four path-planning algorithms: FBLM, RW, and two variations of CGP parameterized differently, with a $span$ variable set to 5 and 10 meters respectively. The diagrams in

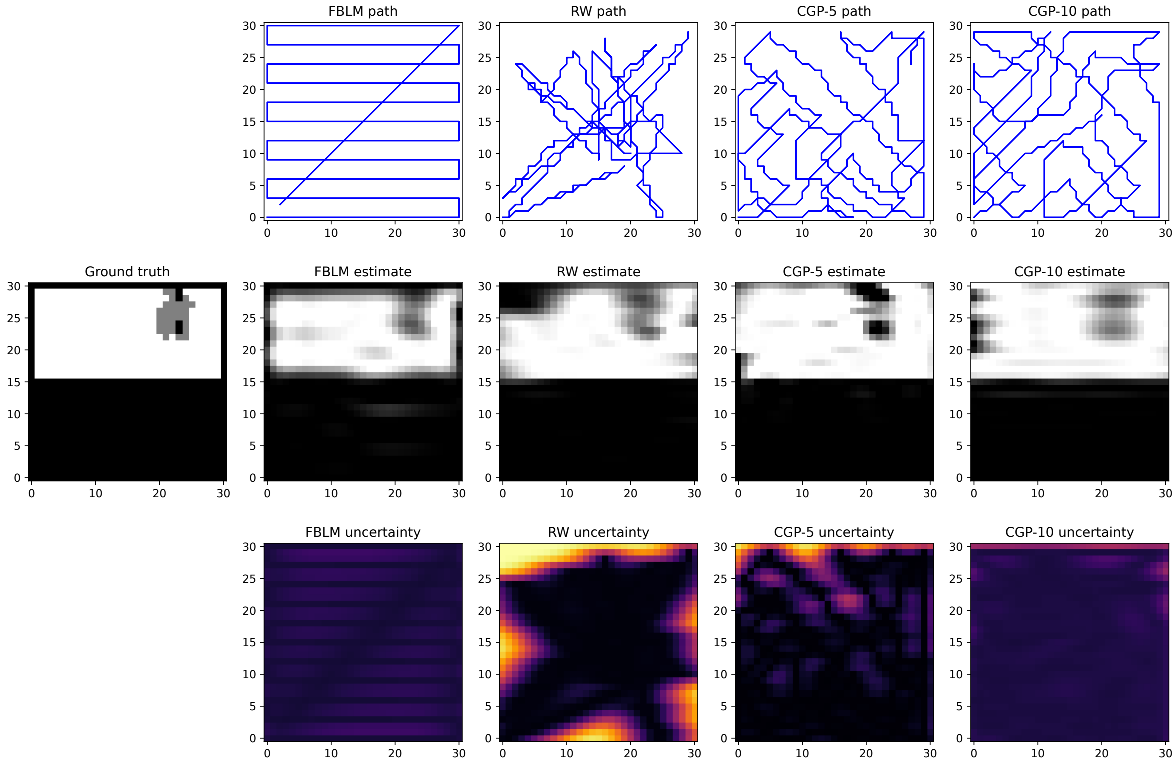

Fig. 2. The exploration behavior of the four algorithms (FBLM, RW, CGP-5 and CGP-10). Top row: the path of the robot over the 400 timestep exploration. Middle row: the ground truth of the experiment (a TYLCV outbreak in the area planted with tomatoes), and the final estimates of the system GP estimator for each of the path planners. Bottom row: the uncertainty values at the end of the experiment for the different path planners.

the figures show the path taken by the mobile sensor, and for each of the three metrics, the ground truth, the model estimate, and the uncertainty map. Figure 3 compares the evolution of the score achieved by the mobile robots over time.

There are several interesting observations that can be made based on these results. First, the path of the mobile sensor under the FBLM, as expected, performs a uniform coverage. A disadvantage of this path planner is that the systematic lawnmower pattern creates a highly unbalanced coverage during the exploration - for instance, halfway through the exploration, the model covered all the strawberries, but none of the tomato planted area.

The RW model has the advantage that by randomly sampling from the entire area, the waypoints are uniformly distributed across any location, which means that the neighborhood of any given area has a strong likelihood to be visited early in the exploration. Unfortunately, the RW model also has disadvantages. Despite the waypoints being sampled uniformly from the area of interest, the mobile sensor spends more time in the interior of the area of interest, a well know problem of random waypoint algorithms [15]. With the budget of 400 timesteps, relatively large spans in the corner of the area of interest are not covered. Furthermore, due to the random sampling, the path of the mobile sensor self-intersects a significant number of times, leading to unproductive revisits of the same location.

Interestingly, the CGP model performs a relatively consistent, almost uniform coverage of the area, without being explicitly programmed for it. The density of the coverage is somewhat more uniform in the case of the $span = 10$ variant, which is considering a larger vicinity for the optimization. While the path occasionally intersects, it does so much more rarely compared to RW. At the same time, the path taken by the CGP model also has a significant random component of it, due to the random breaking of ties in Algorithm 1.

Figure 3 compares the four algorithms with respect to the score obtained as the exploration progresses. To obtain this graph, we run the system estimator (GP-based) at every timestep to obtain a $score$, which measures the average mean error of the system estimator compared to the ground truth of the TYLCV outbreak. To achieve a “higher the better” score metric, we are taking the negative of this mean error. Note that in an actual deployment, the system estimator would be only invoked once, at the end of the data collection.

The first observation we can make based on Figure 3 is that, as a matter of long-term trend, the score improves for all path planners as the exploration proceeds, and new observations are added to the pool. The improvement of the score, however, is not necessarily monotonic. For instance, the discovery of a patch of infected tomatoes might make the estimator assume the existence of an infected area that might be much larger than

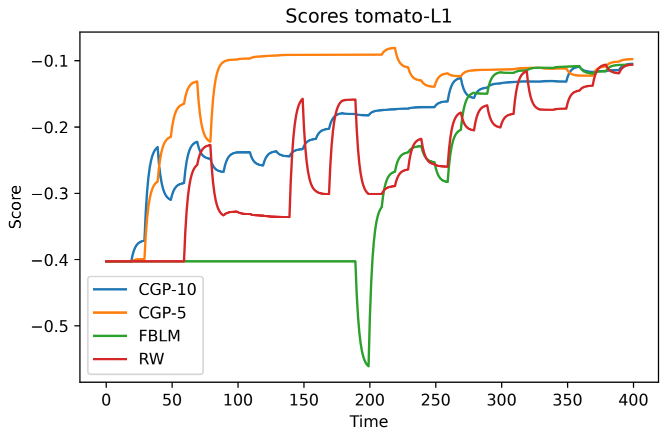

Fig. 3. The evolution of L1-score (negative mean absolute error) for the four algorithm (FBLM, RW, CGP-5 and CGP-10) over the course of the experiments.

the real one. Thus, even a correct observation might lower the quality of the estimate. Overall, however, adding observations improves the estimates.

Of particular interest is that score at the end of the allocated exploration budget. Interestingly, all four techniques obtain an essentially indistinguishable score at the end of the exploration. One would have expected that the RW model with its inconsistent coverage and wasted exploration time by revisiting previously explored areas to be significantly lower than the FBLM model with its highly regular and thorough coverage pattern. However, the difference is minimal.

A second consideration applies to the manner in which the score evolves over time. The concept of an exploration budget, expressed in the form of a static time value, is often an optimistic assumption. In many scenarios, it frequently happens that the full exploration cannot be completed due to environmental hazards, or the model needs to be queried before the exploration budget has been fully spent. For these reasons, we prefer to have *anytime* systems, which not only reach a good score at the end of the exploration but also provide a good estimate if queried earlier.

If we examine the evolution of the score over time, the behavior of the path-planning models is significantly different. The FBLM model improves very slowly because, for instance, halfway through the time budget, it only has information from one-half of the area. In contrast, the RW model exhibits an initially strong value, as the waypoints are randomly spread across the entire area. However, this initial high performance slows down as the random exploration might not fill in some under-explored areas. Conversely, the two variations of the CGP model both perform significantly better than all the other models in the early part of the exploration and retain their advantage until almost the very end of the exploration.

V. Conclusions

In this paper, we described an algorithm for mobile sensors that adapts its path depending on the information captured by the sensors. The system builds a local model of the environment using a Gaussian Process regression-based estimator and uses the level of confidence of the estimator to guide the movement of the mobile sensor, moving toward the areas where it has the least confidence values while breaking ties randomly. We show that this, a comparatively simple approach, leads to an efficient coverage of the area without explicit geometric reasoning. Compared to the standard coverage and random waypoint models, we show that the proposed approach yields significantly higher accuracy in the early part of the exploration, providing a better anytime performance.

References

[1] 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 *Proc. of 2014 IEEE Int. Conf. on Robotics and Automation (ICRA-2014)*, 2014, pp. 2658–2664.

[2] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 413–14 423, 2020.

[3] T. M. Cabreira, L. B. Brisolara, and F. J. Paulo R, "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, no. 1, p. 4, 2019.

[4] J. Xu, G. Solmaz, R. Rahmatizadeh, D. Turgut, and L. Bölöni, "Internet of things applications: Animal monitoring with unmanned aerial vehicle," *arXiv preprint arXiv:1610.05287*, 2016.

[5] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA-2010)*, 2010, pp. 4791–4796.

[6] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, 2016.

[7] F. A. Khan, S. A. Khan, D. Turgut, and L. Bölöni, "Greedy path planning for maximizing value of information in underwater sensor networks," in *Proc. the 10th IEEE International Workshop on Performance and Management of Wireless and Mobile Networks (P2MNET-2014)*, September 2014.

[8] H. Huang, A. V. Savkin, M. Ding, and C. Huang, "Mobile robots in wireless sensor networks: A survey on tasks," *Computer Networks*, vol. 148, pp. 1–19, 2019.

[9] 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.

[10] C. Chekuri, N. Korula, and M. Pál, "Improved algorithms for orienteering and related problems," *ACM Transactions on Algorithms*, vol. 8, no. 3, pp. 1–27, 2012.

[11] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, "A clustering-based coverage path planning method for autonomous heterogeneous uavs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 546–25 556, 2021.

[12] C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in Gaussian processes," in *Proc. of the 22nd Intl. Conf. on Machine Learning (ICML-2005)*, 2005, pp. 265–272.

[13] C. Chekuri and M. Pál, "A recursive greedy algorithm for walks in directed graphs," in *Proc. of IEEE Symposium on Foundations of Computer Science (FOCS-2005)*, 2005, pp. 245–253.

[14] S. Matloob, P. P. Datta, O. P. Kreidl, A. Dutta, S. Roy, and L. Bölöni, "Waterberry farms: A novel benchmark for informative path planning," *arXiv preprint arXiv:2305.06243*, 2023.

[15] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.