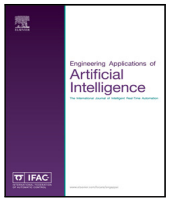
Engineering Applications of Artificial Intelligence 121 (2023) 106054

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Gaussian processes and Fast Marching Square based informative path planning☆

Javier Muñoz *, Blanca López, Fernando Quevedo, Santiago Garrido, Concepción A. Monje, Luis E. Moreno

Robotics Lab, Universidad Carlos III de Madrid, Leganés, Spain

ARTICLE INFO

Keywords:
Unmanned Aerial Vehicles
Gaussian processes
Path planning
Exploration
Informative Path Planning

ABSTRACT

The exploration of unknown environments is a challenge in robotics. The proposed method approaches this problem by combining the Fast Marching Square path planning technique with the machine learning method called Gaussian processes (GP). The Fast Marching Square method is used to determine the most unexplored areas of the environment and to plan the path of the vehicle from the current position to the selected point. The GP model is used to obtain predictions about the unexplored regions of the environment based on the collected data so far during the exploration. The use of Unmanned Aerial Vehicles (UAVs) for exploration and surveillance has increased exponentially in the recent years, due to their sensor equipment capabilities and their versatility for flying over difficult terrain. By defining the weight each method has on the selection of the next point to explore, we can focus the UAV on the points with more interesting data defined by the user (*i.e.* bodies of water), the most unexplored regions, or a combination of both. We present an study on the influence of these weights on the mean absolute error (MAE) and predictive variance obtained from the GP model and test the algorithm on a real environment obtained from a satellite image. We show that we are able to generate an accurate depiction of the environment way faster than traditional methods such as the Boustrophedon.

1. Introduction

Informative path planning (IPP) of unknown areas is the problem of choosing a path for a robot that maximizes the sensor information obtained about an underlying field of interest (Binney et al., 2013). On the other hand, exploration of an unknown environment is defined as the task of mapping an unseen environment while optimizing the path taken to avoid passing over already visited areas. The difference between IPP and basic exploration is that the IPP is usually focused on obtaining certain information about the environment and focusing on those discoveries to gain a better understanding of the whole area, while the exploration of an area frequently gives an equal value to every point of the space and targets mapping the whole region. Furthermore, IPP algorithms usually start from an outdated map on the environment and use that information to select the points to visit in order to obtain the most information while doing the smaller number of data collections possible.

Depending on the exact nature of the mission, IPP or exploration, a specific algorithm has to be designed for each task. In this paper, we introduce an algorithm capable of adapting to both IPP and exploration missions, using a mixture of Gaussian Processes (GPs) and the Fast Marching Square method (FM^2) to leverage the algorithm either towards data collection of important features of the environment or towards exploring the whole extension of the environment minimizing the distance transversed which results in a shorter mission with a smaller energy consumption. We use a GP to build a model of the environment that gets updated as the UAV moves through the area. Then, we use the predictive mean obtained from the GP model output for all points of the map and the velocity map obtained from the FM^2 method to select the next point of the environment to visit. Finally, we use the FM^2 path planner to travel to that desired point, marking the already visited points as obstacles so we obtain a safe and smooth trajectory that traverses the most unexplored areas of the environment, allowing for better data collection and thus improving the GP model. By using two different weights, we can guide the exploration towards the highest values of the GP model outputs or towards the most unexplored areas marked by the highest values in the velocity map.

To solve the IPP problem, Binney et al. (2010) use GPs to model the area of interest given a set of static waypoints to visit, and then

☆ This research was funded by the EUROPEAN COMMISSION: Innovation and Networks Executive Agency (INEA), through the European H2020 LABYRINTH project. Grant agreement H2020-MG-2019-TwoStages-861696.

* Corresponding author.
E-mail address: jamunozm@ing.uc3m.es (J. Muñoz).

https://doi.org/10.1016/j.engappai.2023.106054
Received 1 July 2022; Received in revised form 17 January 2023; Accepted 22 February 2023
Available online xxxx
0952-1976/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

use a modified recursive greedy algorithm to find the best path that limits the energy consumption of the underwater glider while reducing the uncertainty in the field being measured. Our approach has an advantage over this method in that the vehicle's movements are not restricted to a limited number of nodes on a graph, as we use the full resolution of the map to plan paths with the FM^2 method. Ma et al. (2017) use sparse online Gaussian processes (SOGP) to perform ocean monitoring tasks. The next point for the marine vehicle to visit is determined via mutual information, aiming to obtain the most information-rich data. By visiting these points and using SOGP, they are able to build a map of the environment. However, they greatly reduce the resolution of the map to improve the computational cost and travel between sampling points using straight lines. Chen and Liu (2021) present an anytime multi-objective informative planning method called Pareto Monte Carlo tree search which allows the robot to handle potentially competing objectives such as exploration versus exploitation. Our proposed approach has a major advantage over their method in that we actively avoid revisiting previously visited points in the environment by modifying the velocity matrix of the FM^2 method. Yang et al. (2013) use a Gaussian process based RRT planner for the exploration of an unknown and cluttered environment with a UAV. They take the laser scans from the sensor and build a Gaussian process obstacle probability map, and use an RRT planner with constraints and address replanning strategies based on probability of collision along the path produced by the GP map. These movement constraints can be easily achieved by modifying the velocity matrix of the FM^2 method, as previously demonstrated in our work (Garrido et al., 2022). Viseras et al. (2016) propose a GP based multi-agent exploration algorithm that leverages what information to transmit to achieve multi-agent coordination, how to implement a light-weight collision avoidance strategy and how to learn the data's model without prior information. They achieve the best results by using pre-set hyperparameters learned from previous missions, whereas we estimate the hyperparameters from scratch for each mission. Luo and Sycara (2018) use a mixture of locally learned GPs to perform adaptative sampling and online learning of the model of the environment with a team of UAVs. Jadidi et al. (2014) use GPs to handle sparse sensor measurements to build a gradient field of occupancy probability distribution, providing frontier boundaries for further exploration. They expand this proposal in Ghaffari Jadidi et al. (2018) by adding a mutual information-based greedy exploration technique that takes into account all possible future observations. Xu et al. (2011) propose a centralized navigation algorithm for mobile sensor networks to move in order to reduce prediction error variances at selected locations using spatiotemporal GPs.

More recent approaches such as Zhu et al. (2021) present a method for actively gathering information about a 3D surface using a robotic platform. The method combines a GP model of the surface with an IPP algorithm called the information gain-driven adaptive A* (IG-A*) to quickly gather accurate information about the surface while minimizing the travel time. They connect the waypoints with a polynomial trajectory, which is not as smooth and straight-forward as the FM^2 path planner. Di Caro and Yousaf (2021) propose a method for coordinating the movement of multiple robots to gather information about an unknown environment using a leader–follower architecture. The leader robot uses a GP model of the environment to plan paths that will reduce uncertainty as quickly as possible and followers robots use this information to plan their own paths while avoiding collisions. The method is able to quickly gather accurate information about the environment while minimizing the travel time. They start the missions assuming they already have previously collected some data from the environment, while our proposed method starts with no prior information about the environment.

Previous works in our group Gómez et al. (2013) propose using the velocity map of the Fast Marching Square method to calculate the most unexplored areas, and then move the robot to that point. They mark the visited points as obstacles and calculate the velocity map again to calculate the next point.

In this paper, we propose an IPP method that combines GPs and the Fast Marching Square method to focus the exploration on either the most valued areas based on the user criteria or the most unexplored areas. Other methods use metrics such as the predictive variance obtained from the GP model to guide the exploration and use straight lines or traditional path planning methods to plan the path for the vehicle, while the proposed method uses the Fast Marching Square method to determine the most unexplored areas of the map and to plan the path for the UAV, which is a smooth path that travels across unexplored regions instead of a straight line. As the UAV explores the area, a GP model of the environment is built, which gives information about the most valued areas and guides the exploration prioritizing the observation of those points.

The paper is structured as follows. Section 2 details the IPP problem statement. Section 3 showcases the results obtained when applying our proposed method to a real environment. Finally, Section 4 provides the conclusions obtained from this work.

2. Problem statement

In this section, we present the methods used to develop the informative path planning and exploration algorithm based on Gaussian Processes and the Fast Marching Square method.

2.1. Gaussian process regression

Regression is the problem of estimating a function h given a set of input vectors $x_i \in \mathbb{R}^D$ and observations $y_i = h(x_i) + \epsilon_i \in \mathbb{R}$ of the corresponding function values, where ϵ_i is a noise term. Due to finiteness of measurements y_i, and the presence of noise, the estimate function h is uncertain (Deisenroth, 2010). A Gaussian process (GP) provides the concept of a distribution over functions, which allows us to express this uncertainty in terms of probability distributions. An example of GP regression can be seen in Fig. 1.

A GP can be defined as a collection of random variables, any finite number of which have a consistent joint Gaussian distribution (Williams and Rasmussen, 2006), *i.e.* every finite linear combination of them is normally distributed. Gaussian processes are useful in statistical modeling, benefiting from properties inherited from the normal distribution. One main advantage of GPs is that they are non-parametric, which allows to determine the shape of the underlying function h from the data and higher-level assumptions. This does not imply that the model has no parameters, but that the effective number of parameters is flexible and grows with the sample size.

To estimate function h, that is, the model of the environment, we use the squared exponential kernel (Eq. (1)).

$$k(x_1, x_2) = \sigma_f \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma_l^2}\right) \tag{1}$$

where σ_f is the signal standard deviation and σ_l is the characteristic length scale.

The process of training a GP consists on optimizing the set of hyperparameters $\theta \triangleq \{\sigma_f, \sigma_l\}$ so the kernel function can describe the underlying environment as accurately as possible. A common approach to learning the hyperparameters is via the quasi-Newton optimizer, which uses a trust-region method with a dense, symmetric rank-1-based (SR1), quasi-Newton approximation to the Hessian (Dennis and Schnabel, 1996).

One major disadvantage of Gaussian processes is that they become computationally expensive as the number of data points grow. Since the computation of the inverse of an $n \times n$ matrix, where n is the number of data points, is needed to make predictions. This computation has a complexity of $\mathcal{O}(n^3)$. To tackle this problem, methods such as the Subset of Datapoints (Williams and Rasmussen, 2006) were developed. This method proposes to keep the GP predictor, but only on a subset of size m

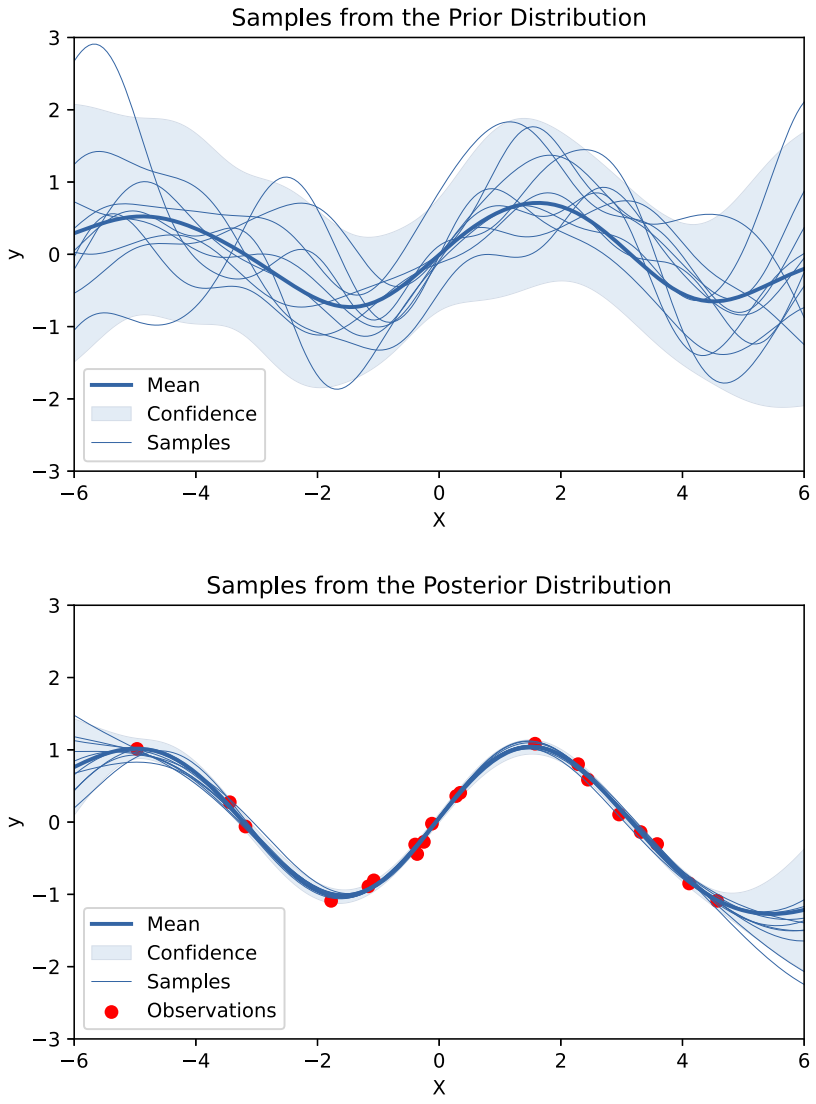


Fig. 1. The top panel shows ten samples drawn from the prior distribution. The bottom panel shows the posterior distribution after twenty datapoints have been observed. In both plots the shaded region denotes twice the standard deviation at each input value X, which represents the 95% confidence interval.

of the data. This subset is called the active set vector. Even though this implies that some data is wasted, it can make sense if the predictions obtained with m points is accurate enough for our needs. This reduces the complexity of the algorithm to $\mathcal{O}(m^3)$. There are several approaches to selecting the most valuable points for the active set (Lawrence, 2003). To choose the next point for inclusion into the active set, we will use the differential entropy score $\Delta_j \triangleq H[p(f_j)] - H[p^{new}(f_j)]$, where $H[p(f_j)]$ is the entropy of the Gaussian at site $j \in R$ and $H[p^{new}(f_j)]$ is the entropy at this site once the observation at site j has been included. In this work, the selected number of points in the active set is 128.

For applications where the data is being collected constantly, we need a method to add data points to the Gaussian process during the mission (Ranganathan et al., 2010). In this work, we build a new GP model adding the recently explored points to the data previously collected. To avoid training the GP model again, the same hyperparameters are kept for the new GP model.

2.2. Fast Marching Square method

The Fast Marching Square (FM^2) method (Gómez et al., 2013) is a variation of the original Fast Marching method (Sethian, 1999). It creates a velocity map given a binary map of occupation by propagating a wavefront that takes all the obstacles as source points. Then, another wavefront is propagated from the desired end point to the current position of the vehicle. The result is a potential map that contains no local minima, with a single global minimum at the end point. An example of velocity map can be seen in Fig. 2.

To calculate the path, gradient descent is performed on the potential map from the current position of the vehicle. To better illustrate the differences between the two methods, paths are calculated with both methods for the same environment and shown in Figs. 3 and 4. Both figures show that the path obtained with the FM^2 method is more smooth and reproducible by a robot, thanks to the computation of the velocity map.

The velocity map values range from 0 to 1, representing the maximum speed allowed for the vehicle. Obstacles will imply speeds equal to zero, whilst points in space far enough away from obstacles will allow maximum speeds. When computing a path for the vehicles to follow, the FM^2 method will get the shortest path from the initial position to the goal position that lets the vehicles navigate at the highest speeds.

As the velocity map indicates the distances from the points in the space to obstacles, it can also be seen as a measure of the unexplored points in the environment (Garrido et al., 2011). If we mark the points already visited by the robot as obstacles, we can easily obtain the next point to explore by propagating the wave again. An example of this evolution of the velocity map can be seen in Fig. 5.

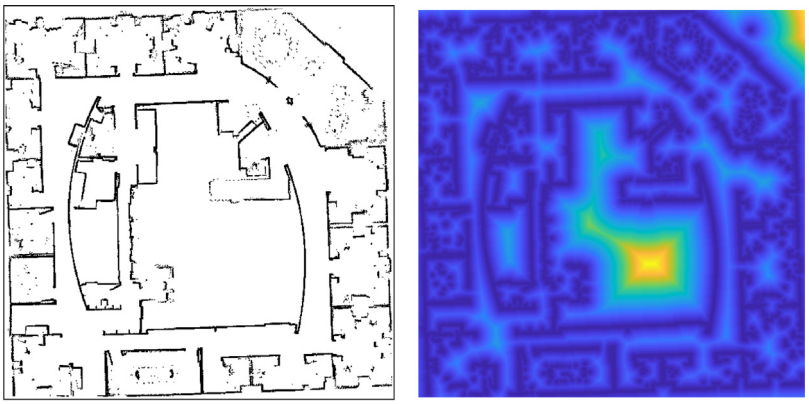


Fig. 2. Obtention of the velocity map by applying the Euclidean distance transform to the binary obstacle map.

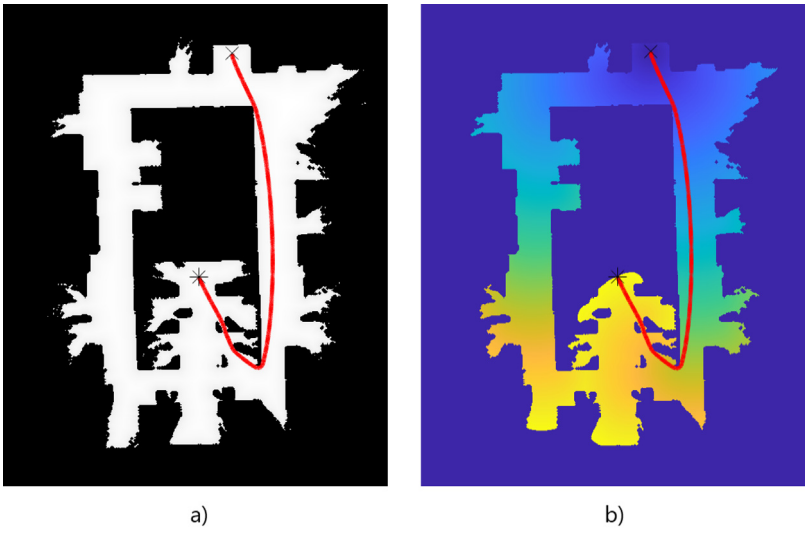


Fig. 3. **(a)** initial obstacle map, **(b)** time of arrival of the wavefront. The red line shows the path obtained with the Fast Marching Method (FMM).

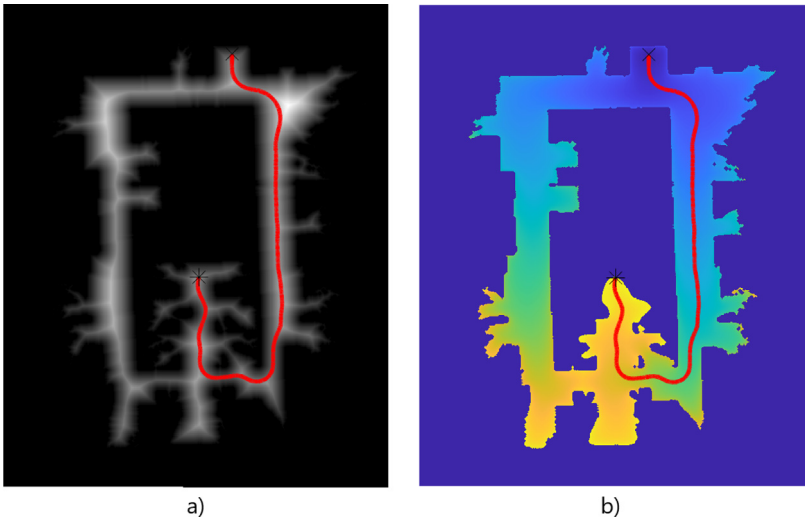
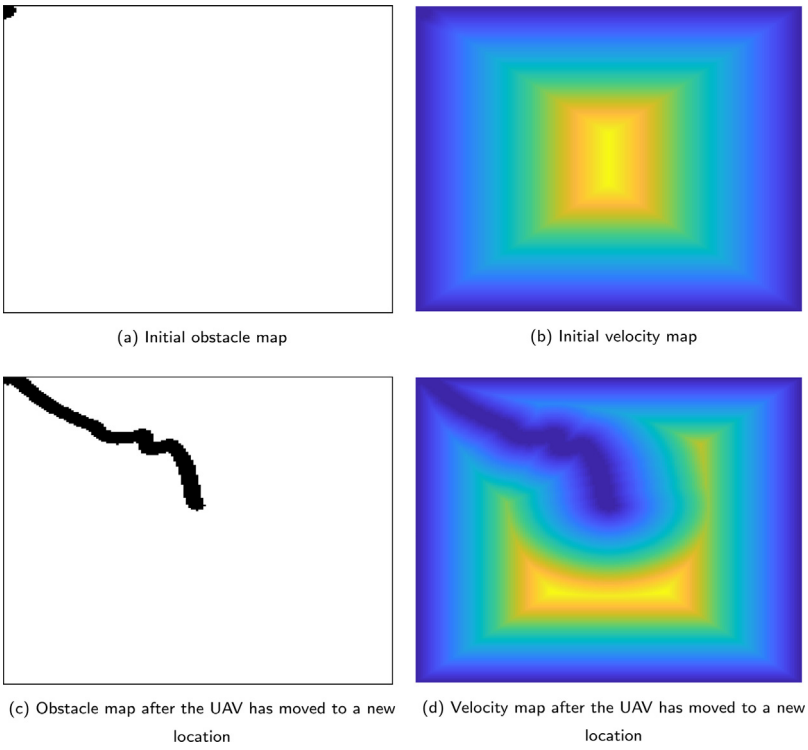


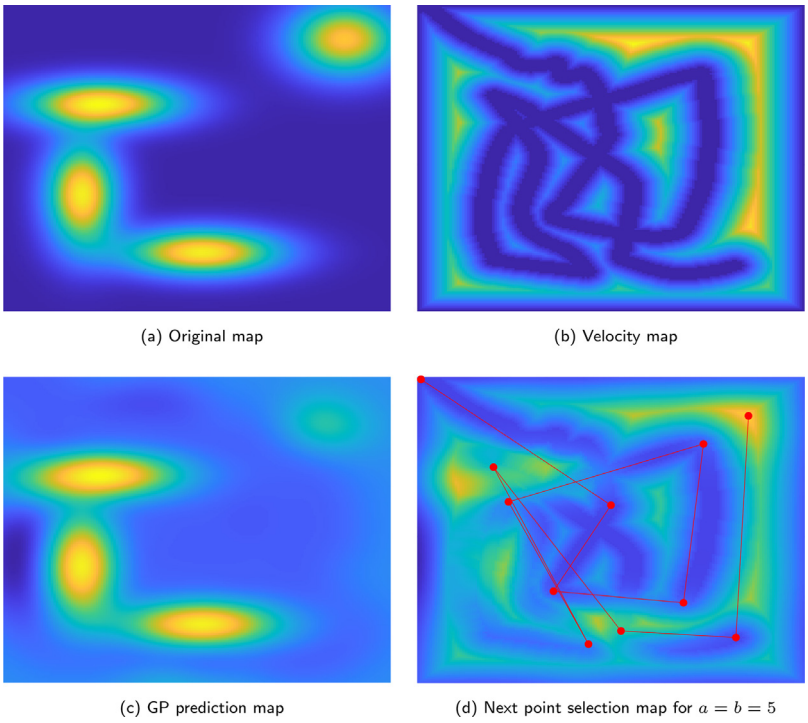
Fig. 4. **(a)** velocity map, **(b)** time of arrival of the wavefront. The red line shows the path obtained with Fast Marching Square (FM^2).



(a) Initial obstacle map (b) Initial velocity map

(c) Obstacle map after the UAV has moved to a new location (d) Velocity map after the UAV has moved to a new location

Fig. 5. Evolution of the velocity map as the UAV moves around the environment and the visited points are marked as obstacles.



(a) Original map (b) Velocity map

(c) GP prediction map (d) Next point selection map for $a = b = 5$

Fig. 6. Process to obtain the next point to explore by the UAV. The previously selected points are marked as red dots and connected by red lines.

2.3. Proposed methodology

In this paper, we propose a method that mixes both GP regression and the Fast Marching Square method. At every iteration of the algorithm, the velocity map and a GP predictive mean for all points in the map are calculated. Then, based on the weight given to each map, the next target point for the UAV is obtained (see Eq. (2)), and the path is calculated using FM^2 while considering the already visited points as obstacles. Note that if we consider the already visited points as hard constraints, the map would eventually get cut in half and no feasible path could be calculated from one side to the other. To tackle that problem, the velocity map value for already visited points is changed from 0 to 0.01 so the UAV only crosses those zones when absolutely necessary.

$$nextPoint = max(a * W + b * GP_{map}) \tag{2}$$

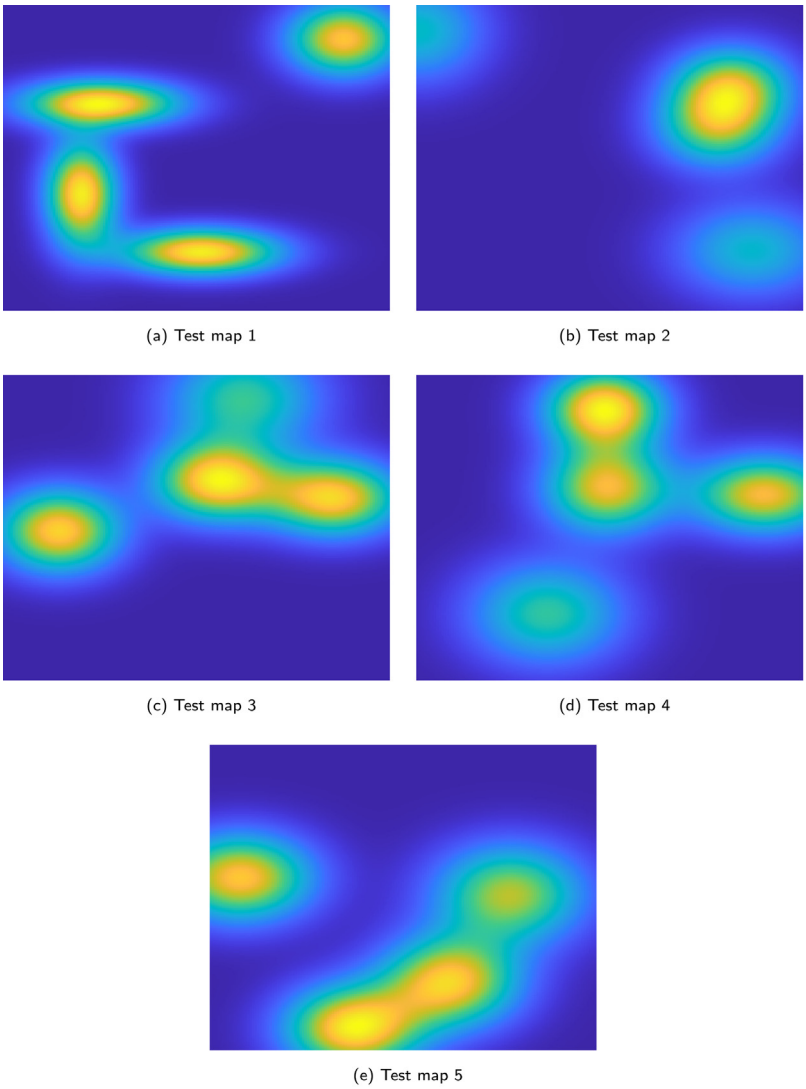


Fig. 7. Test maps used to study the effects of a and b on the MAE and predictive variance. All maps consist of 4 randomly created and located Gaussian mixtures.

where a is the weight given to the velocity map, b is the weight given to the GP prediction map, W is the velocity map given by the Euclidean distance transform of the binary map and GP_{map} is a matrix containing the GP predictive mean for every point in the map. W and GP_{map} are rescaled between 0 and 1. An example of next point selection can be seen in Fig. 6. Fig. 6(a) shows the original map used, composed of four Gaussian mixtures that represent the areas of interest. Fig. 6(b) shows the velocity map after the UAV has followed a set of paths, the already visited zones have a 0 value and are marked as dark blue, while the most unexplored areas are marked in yellow. Fig. 6(c) shows the prediction map obtained from the GP model, which has captured the high interest zones accurately. Fig. 6(d) shows the combination of the velocity map and the prediction map for $a = b = 5$. The next point selected is located in the top right corner, where the value of the combined map is the highest.

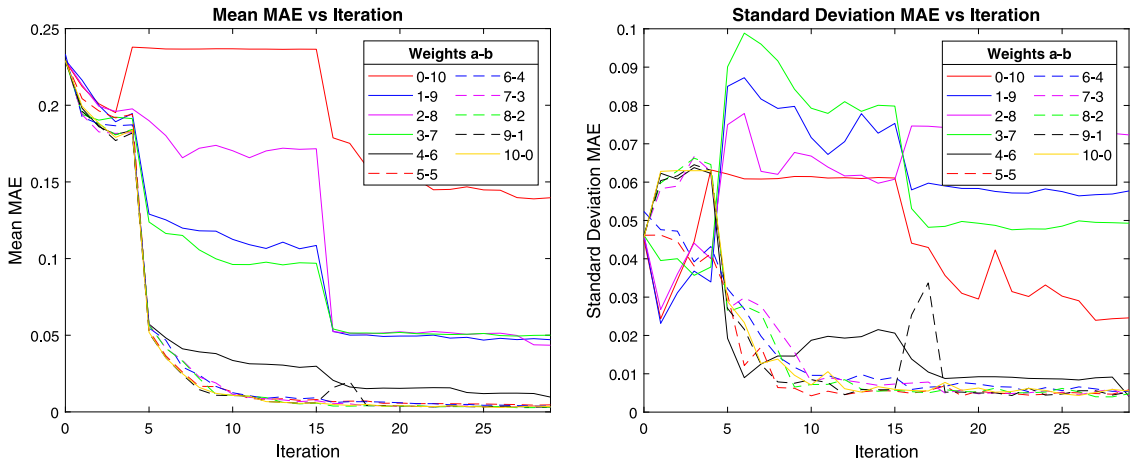
The algorithm is ran for 30 iterations to study the evolution of the obtained GP prediction map, but for real applications a condition such a percentage of the area covered can be set to stop the algorithm, and the hyperparameter optimization is done at iterations 1 and 5 to generate an initial assessment of the map to explore, and then every 15 iterations. Algorithm 1 shows the detailed steps of the proposed exploration algorithm.

Algorithm 1 Routine of the FM^2+GP exploration strategy

```
Define limits of the area to explore
Load the UAV at its initial position
Generate an initial GP model with the points around the UAV
while iter < maxIter do
    Calculate velocity map
    Calculate GP prediction for all points in the map
    Select next point to visit based on Eq. (2)
    Add points observed by the UAV's sensors to the GP model
    if iter = 5 or iter = 15 then
        Update hyperparameters of the GP model
    end if
end while
```

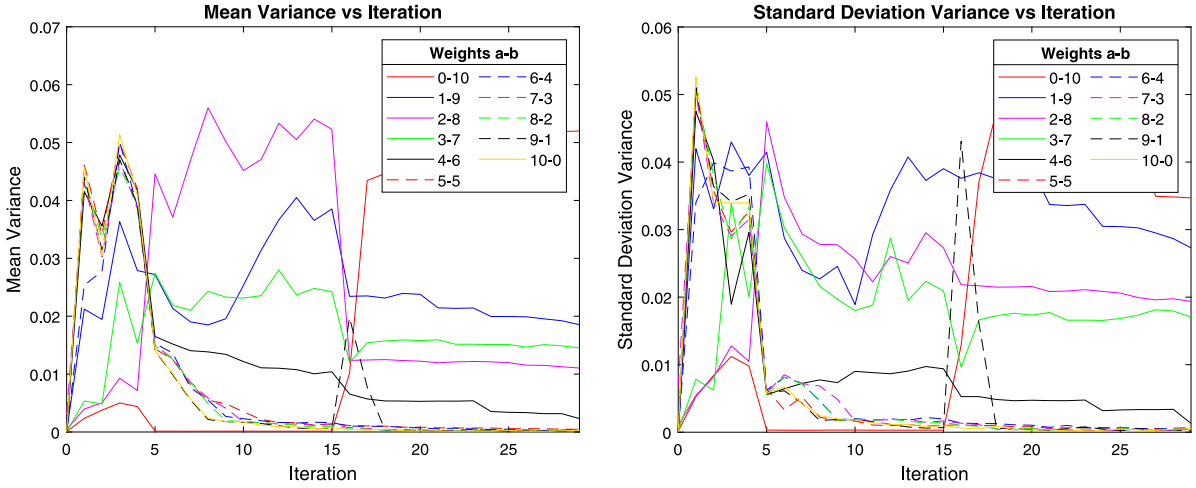
3. Results

In order to test the algorithm and study the effects of the selection of a and b in the mean absolute error (MAE) and mean variance, five artificial maps composed of four Gaussian mixtures rescaled between 0 and 1 are built (Fig. 7) and the algorithm is tested for all possible combinations of a and b. Figs. 8 and 9 show the mean and standard deviation obtained for the MAE and the predictive variance.



(a) Mean MAE (b) Standard deviation MAE

Fig. 8. Mean absolute error (MAE) calculated at each iteration of the algorithm for every combination of weights in every test map.



(a) Mean predictive variance (b) Standard deviation predictive variance

Fig. 9. Predictive variance calculated at each iteration of the algorithm for every combination of weights in every test map.

As it can be seen in Fig. 8, the MAE results improve as a improves and b decreases, and they can be clearly divided into three distinct groups: $a = 0$ and $b = 10$, $1 \leq a \leq 3$ and $7 \leq b \leq 9$ and $a \geq 4$ and $b \leq 6$. For $a = 0$ and $b = 10$, the algorithm is too greedy and focuses on the zones with the highest value, leaving the rest of the map unexplored, hence the unsatisfactory results. For $1 \leq a \leq 3$ and $7 \leq b \leq 9$, there is a balance between the greediness of the GP prediction and the need to explore unexplored areas of the velocity map. This combination of parameters can be useful when we want to focus on getting to the highest valued areas first without getting stuck like in the previous case. For $a \geq 4$ and $b \leq 6$, the velocity map leads the algorithm to explore unexplored areas, which overall gives a better coverage of the area and allows to get better predictions faster. Notice that after the hyperparameter updates in iterations 5 and 15 the MAE is reduced greatly.

Concerning the predictive variance results in Fig. 9, the distinction into three groups is not that clear, but for $a = 0$ and $b = 10$ it is still clear that the algorithm is too greedy and does not perform a thorough exploration of the environment, resulting in a high predictive variance at the end of the mission. As in Fig. 8, we can see that the predictive variance values go down after the hyperparameter updates.

3.1. Variance effect in exploration efficiency

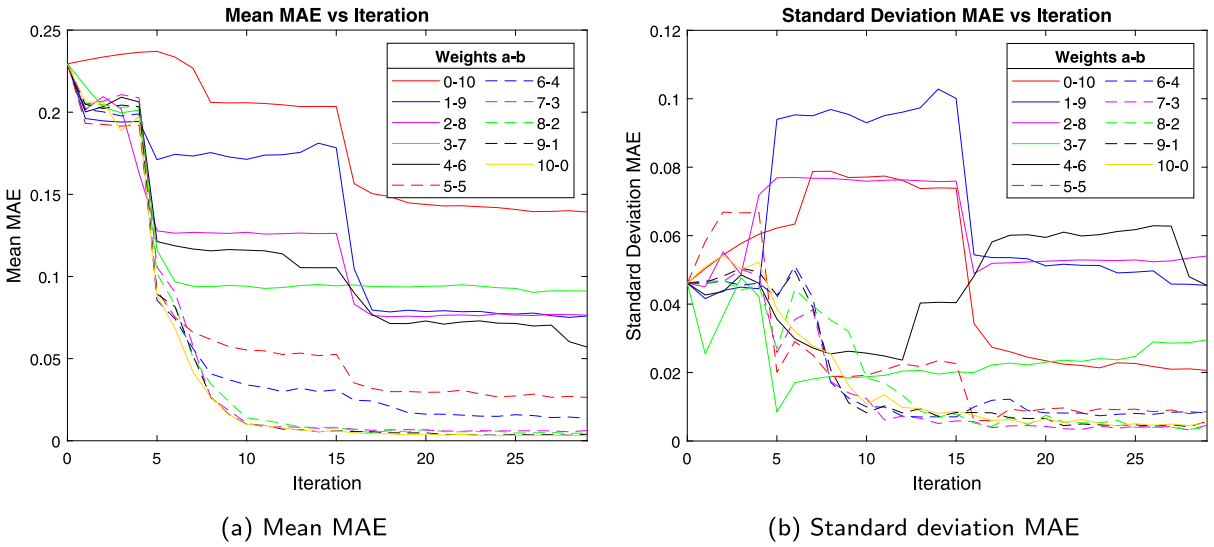
Other IPP approaches commonly use the variance as a variable to select the most unexplored areas, while in this work we use the velocity map W. To further test our algorithm, we use a measure of variance to modify the weight assigned to W, so the desire to explore unvisited zones is greater when the measure of variance is bigger. The formula used to modify the weight of W when selecting the next point for exploration can be seen in Ec. (3).

$$nextPoint = max(a * W * \sigma^2 + b * GP_{map}) \tag{3}$$

where a is the weight given to the velocity map, b is the weight given to the GP prediction map, σ^2 is a matrix containing the GP predictive variance for every point in the map, W is the velocity map obtained with the first propagation of the FM^2 method and GP_{map} is a matrix containing the GP predictive mean for every point in the map. W, σ^2 and GP_{map} are rescaled between 0 and 1.

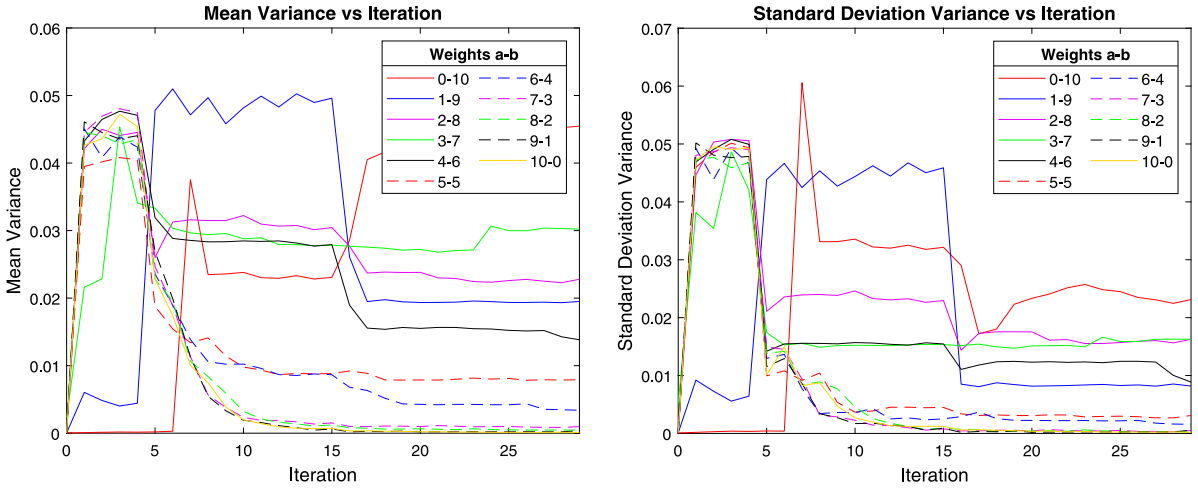
Using the same weight combinations as in the previous experiment, we obtain the following results in Figs. 10 and 11.

Figs. 10 and 11 show that the orange and blue lines, which give a greater a value and therefore give more importance to W and σ^2, greatly improve their performance in the early stages of the exploration



(a) Mean MAE (b) Standard deviation MAE

Fig. 10. Mean absolute error (MAE) calculated at each iteration of the algorithm for every combination of weights in every test map when taking into account the predictive variance for the next point selection.



(a) Mean predictive variance (b) Standard deviation predictive variance

Fig. 11. Predictive variance calculated at each iteration of the algorithm for every combination of weights in every test map when taking into account the predictive variance for the next point selection.

mission when compared to the previous experiment, which demonstrates that using the variance helps guide the exploration and is a good complement to the velocity map.

Now that we have studied the effects of the parameter selection on the MAE and the mean variance, we test the algorithm on a real environment. Fig. 12 shows a LANDSAT (USGS) image of the Gallocanta lagoon in Aragón, Spain. The image has a resolution of 30 × 30 meters per cell and has been rescaled between 0 and 1. The yellow points represent values closer to 1, in this cases bodies of water, and the blue points represent values closer to 0.

Based on the study of the parameters a and b, we select three sets of parameters to perform the exploration. The results for the MAE and the mean variance can be seen in Figs. 13 and 14.

Figs. 13 and 14 prove the assumptions we made with the test environments related to a and b. When observing the resulting final values of the missions, we notice that the minimum MAE is achieved when $a = 8$ and $b = 2$ and the minimum variance is achieved when $a = 10$ and $b = 0$. This makes sense due to the fact that we are using the variance with b in order to get more coverage of the environment. Overall, the minimum sum of the MAE and the variance is achieved with $a = 8$ and $b = 2$. To better see how the algorithm performs in real time, the GP prediction map and the variance map are plotted at iterations 5, 10, 20 and 30 for $a = 8, b = 2$ and $a = 10$, $b = 0$ in Figs 15, 16 and 17, 18, respectively.

Figs. 15 and 17 show that for $a = 8$ and $b = 2$, the algorithm is able to identify the main body of water at the center and the smaller body of water at the top at iteration 10, while for $a = 10$ and $b = 0$ the body of water at the top remains unidentified even after 30 iterations and the shape of the main body of water is better represented for $a = 8$ and $b = 2$ than for $a = 10$ and $b = 0$. For both cases, the variance is constant at all points in the map at iteration 5 except for the already visited points, which have zero variance. This is due to the fact that at iteration 5 the number of explored points is smaller or really close to the number of points considered in the active set, so all points are considered for making the prediction. The lack of understanding of the main body of water in Fig. 17 makes the need for $b > 0$ evident, as we can use the GP prediction values to guide the exploration to these more interesting zones of the environment.

Figs. 16 and 18 show that the variance map gets "flattened" as the UAV explores the area, which reduces the uncertainty about the terrain.

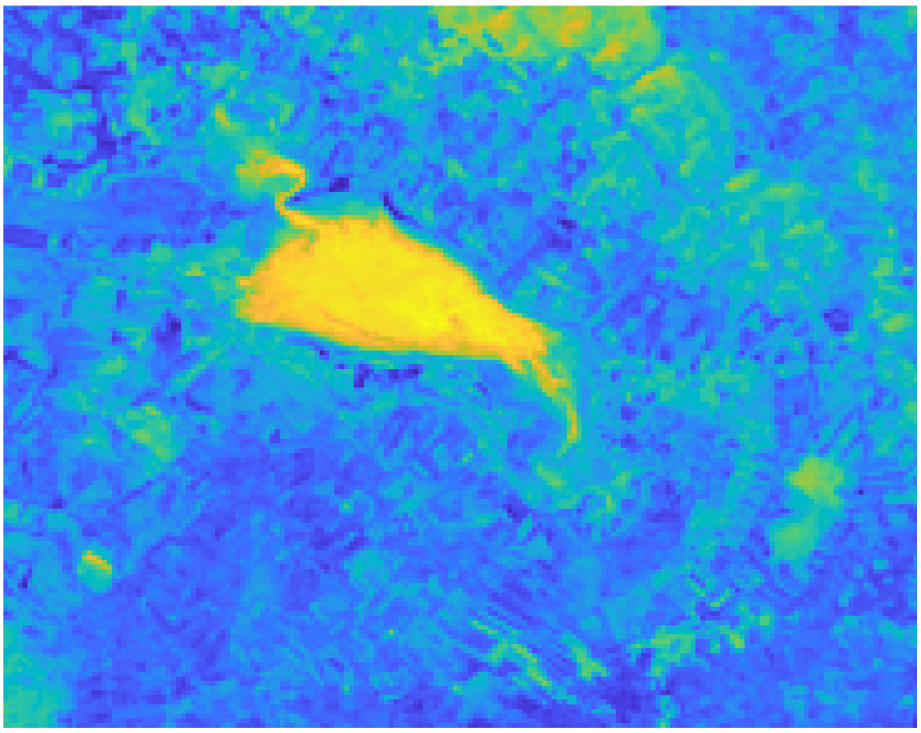


Fig. 12. Original map used for the exploration test.

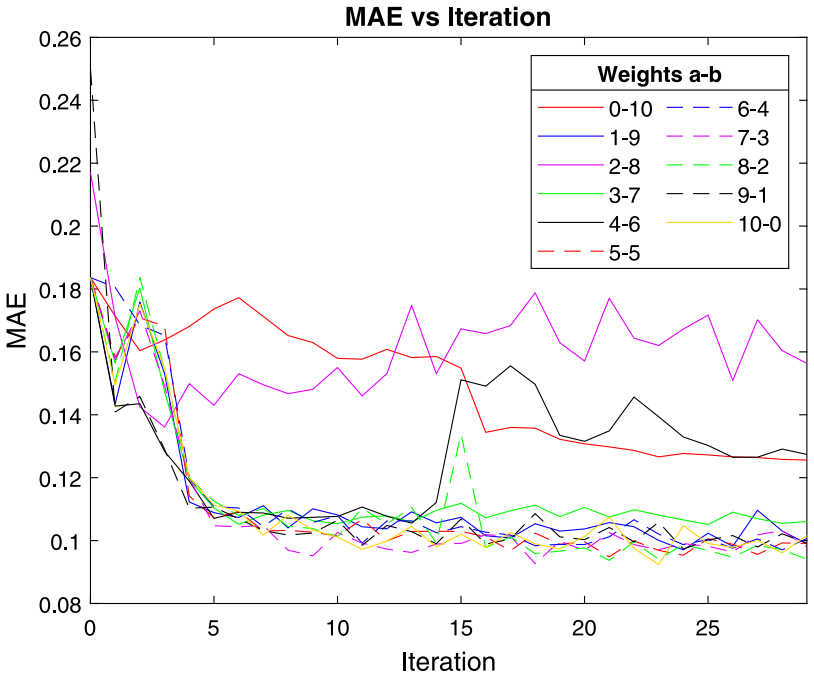


Fig. 13. Mean absolute error (MAE) calculated at each iteration of the algorithm for every combination of weights.

See that for $a = 8$ and $b = 2$ the variance converges faster at iteration 10 than for $a = 10$ and $b = 0$, and has a less bumpy appearance. The variance is always higher at the edges of the map, since those points do not give much information and are usually not considered in the active set. Notice that for $a = 10$ and $b = 0$ the model is overall uncertain about the predicted values of the map.

Finally, we test the proposed algorithm against the Boustrophedon coverage (Choset and Pignon, 1998; Muñoz et al., 2021b,a). Fig. 19 shows that the proposed method is able to generate an accurate prediction on the whole map with a path of the same length as the one calculated by the Boustrophedon method.

4. Conclusions and future work

The main objective of this work was to study and implement an informative path planning/exploration algorithm based on GP regression models and the Fast Marching Square method. The weight given to each

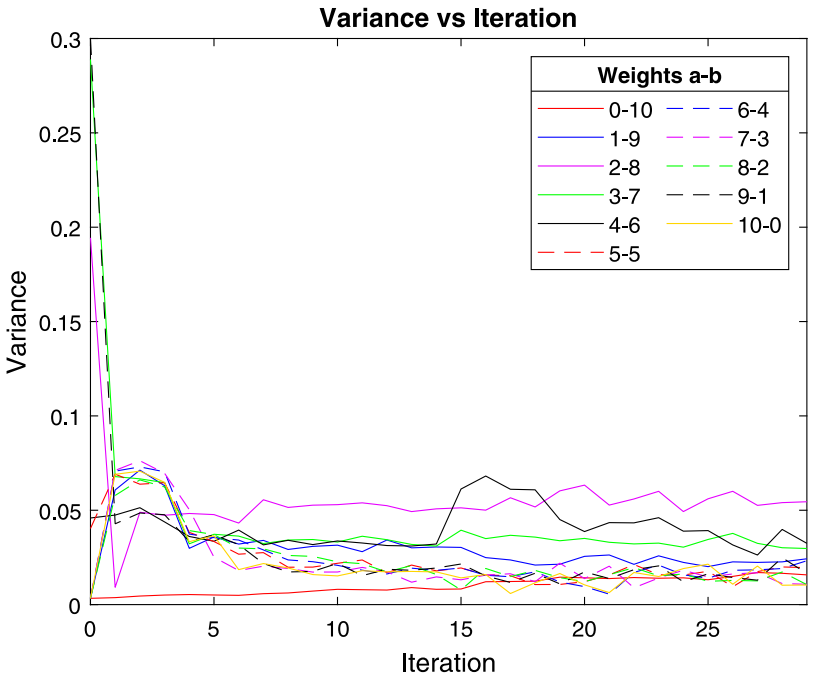


Fig. 14. Variance calculated at each iteration of the algorithm for every combination of weights.

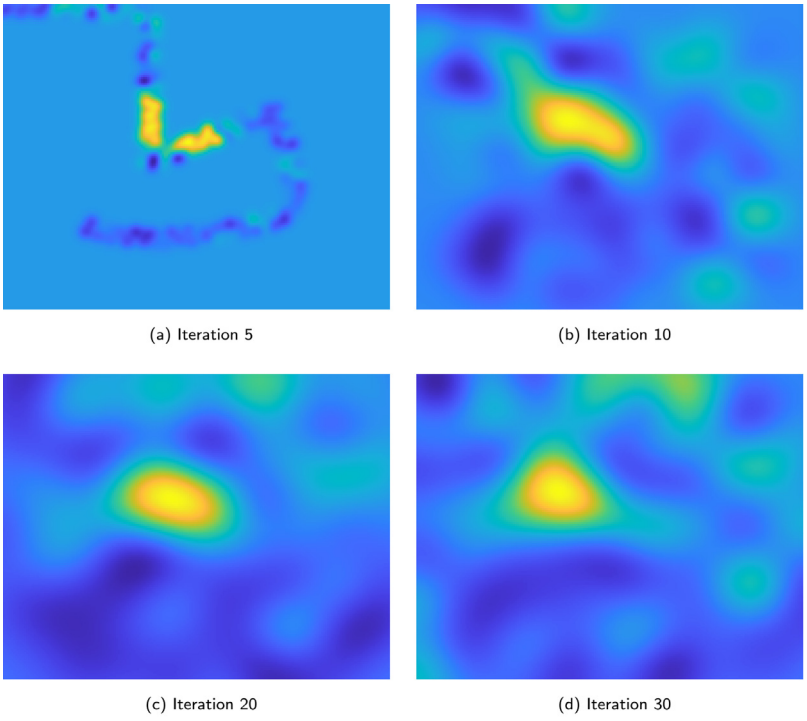


Fig. 15. Prediction map obtained from the GP model for $a = 8$ and $b = 2$.

method is selected by performing tests on five simulated environments and selecting two samples to apply to a real environment obtained from a LANDSAT image. The algorithm is simulated in this environment and measures of the mean absolute error and the variance are given. The results show that the algorithm is able to rapidly explore and generate an accurate representation of the environment by using the GP to make predictions on all points of the environment. Results improve for greater weights of the velocity map W when a measure of the variance is added to increase the need to visit unexplored regions. However, decreasing b to $b = 0$ is not efficient as we are neglecting the importance of the GP predictions when selecting the next point to explore. The proposed method is compared against the traditional Boustrophedon coverage method, and the results show that for a path of the same length, our algorithm yields better results and an accurate depiction of the area to explore.

Some future works for this approach are to use optimization methods such as Differential Evolution or Particle Swarm Optimization to select a and b or more complex methods such as Bayesian optimization in order to reduce the number of tests needed to obtain the optimal

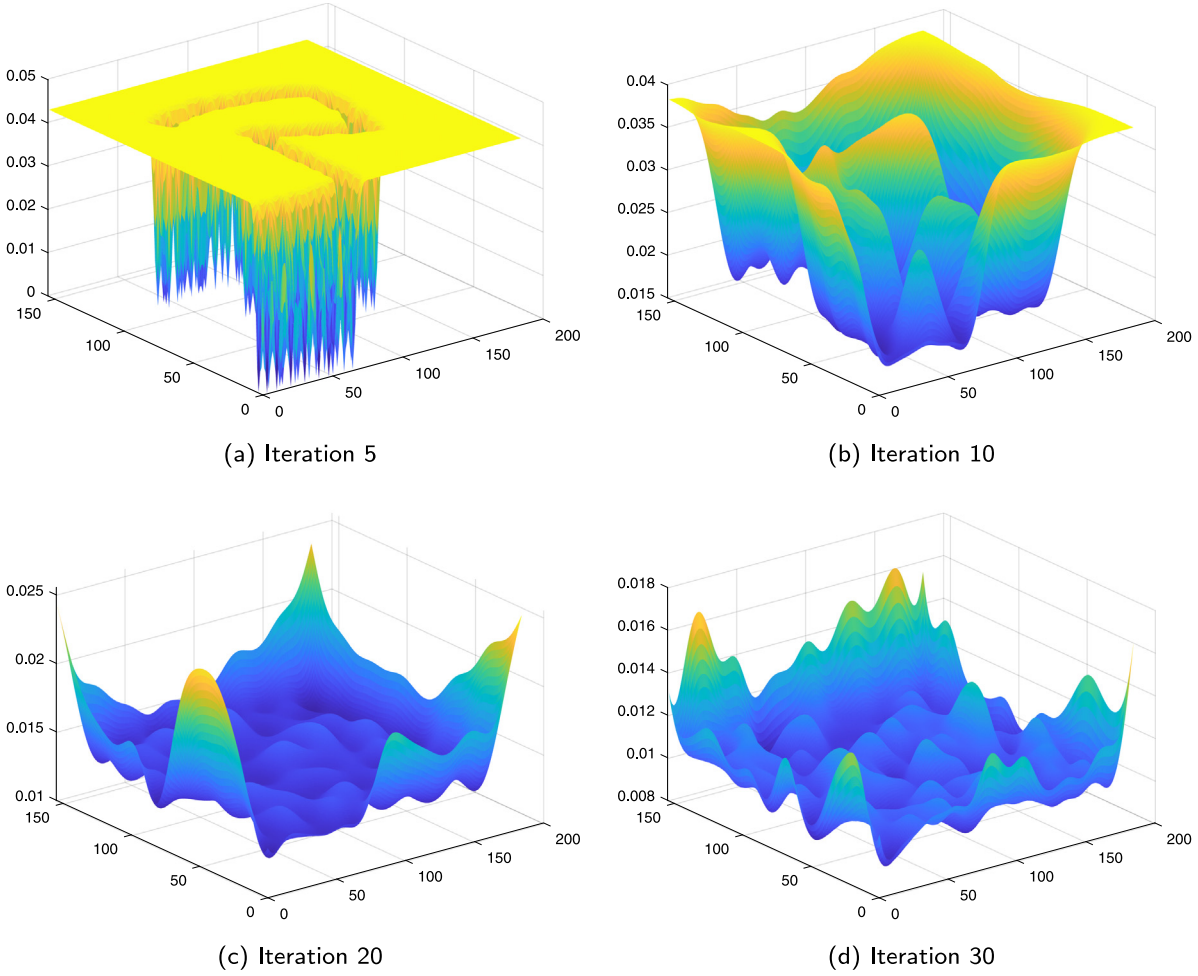


Fig. 16. Variance map obtained from the GP model for $a = 8$ and $b = 2$.

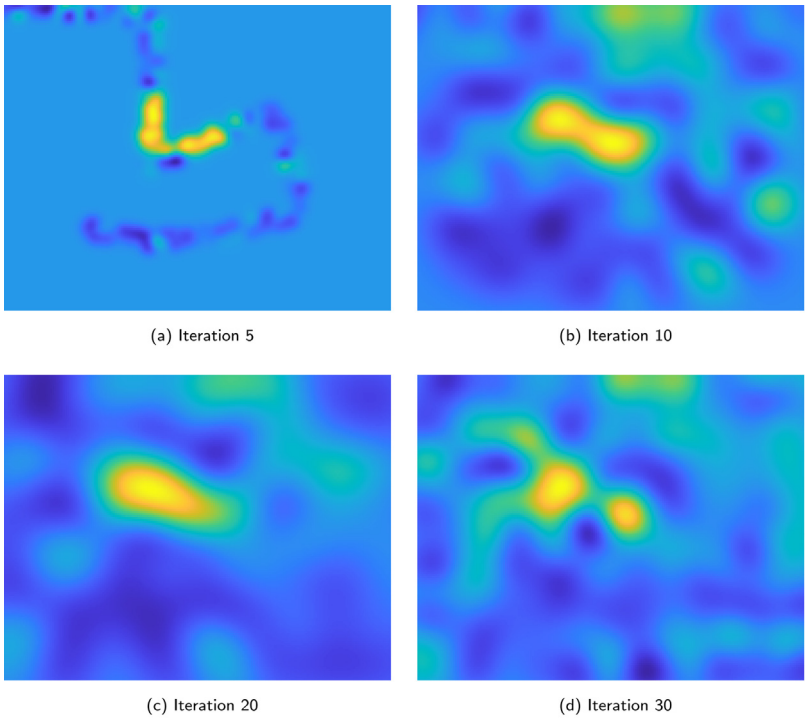


Fig. 17. Prediction map obtained from the GP model for $a = 10$ and $b = 0$.

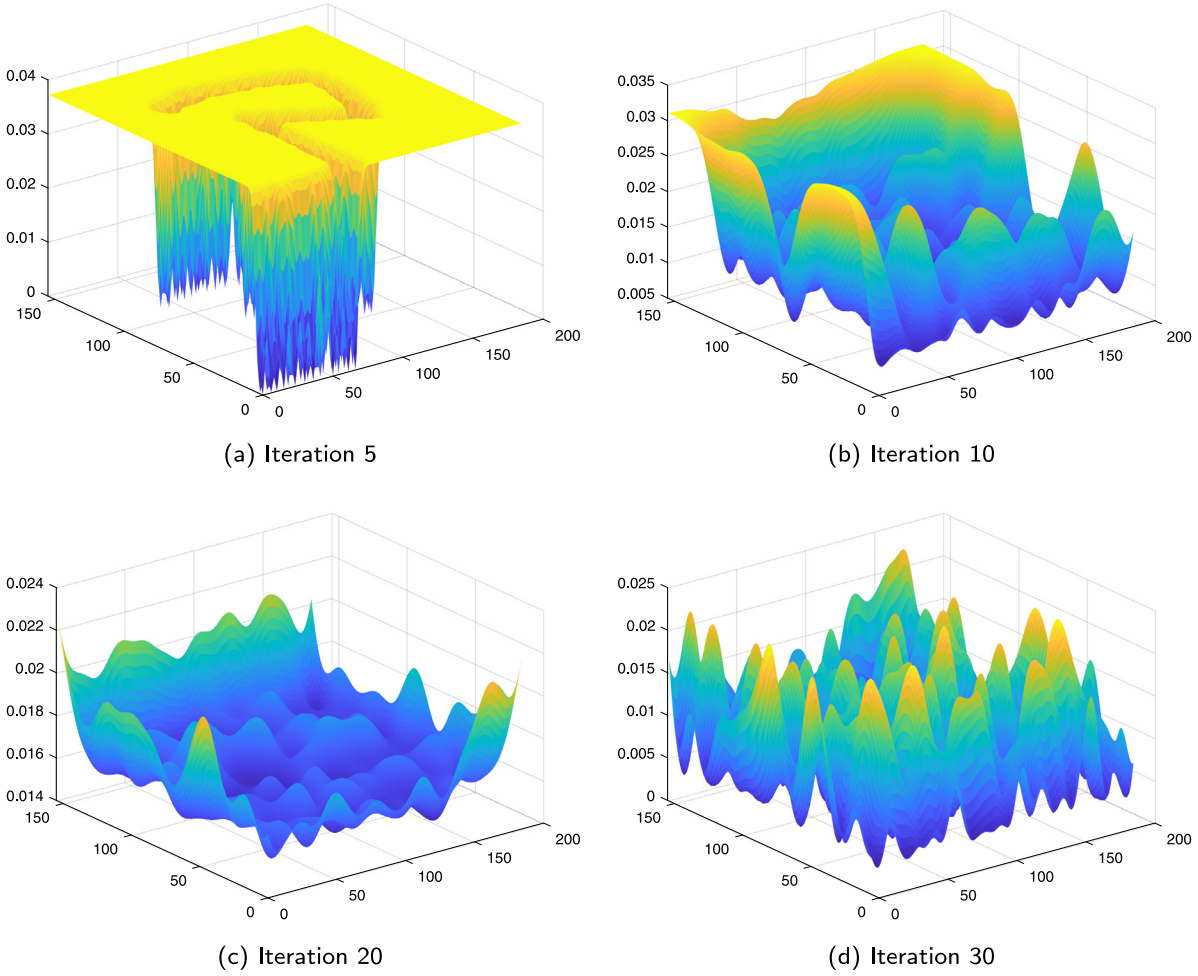


Fig. 18. Variance map obtained from the GP model for $a = 10$ and $b = 0$.

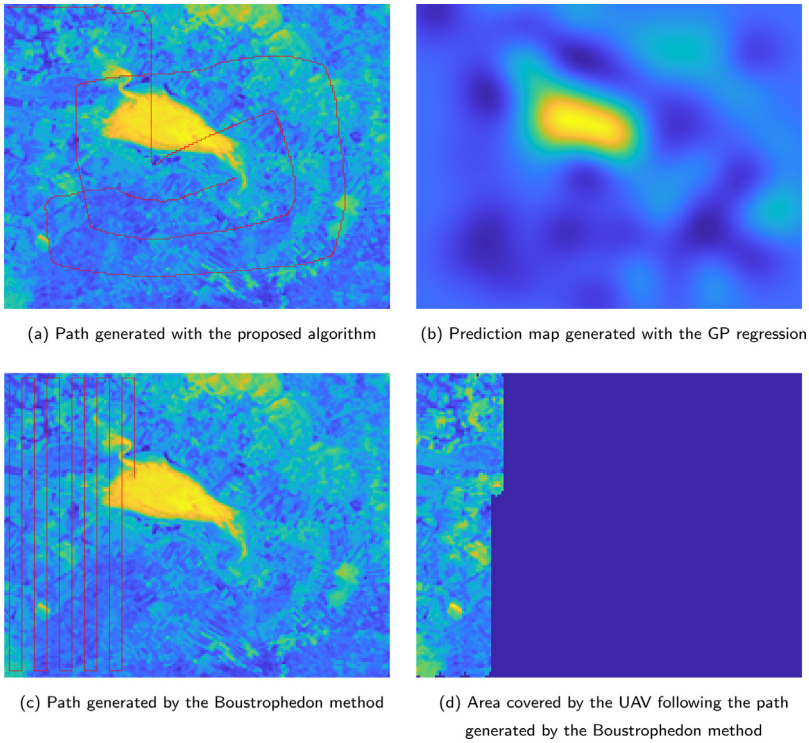


Fig. 19. Comparison between our proposed algorithm with $a = 8$ and $b = 2$ and the classic Boustrophedon method for a path length of 1622 points.

parameters. These methods can also be used to change the frequency of the hyperparameter optimizations.

CRediT authorship contribution statement

Javier Muñoz: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing. **Blanca López:** Visualization, Validation, Writing – review & editing. **Fernando Quevedo:** Data curation, Resources. **Santiago Garrido:** Formal analysis. **Concepción A. Monje:** Writing – review & editing, Project administration. **Luis E. Moreno:** Funding acquisition, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Javier Muñoz reports financial support was provided by EUROPEAN COMMISSION Innovation and Networks Executive Agency (INEA).

Data availability

No data was used for the research described in the article.

References

Binney, J., Krause, A., Sukhatme, G.S., 2010. Informative path planning for an autonomous underwater vehicle. In: 2010 IEEE International Conference on Robotics and Automation. pp. 4791–4796. http://dx.doi.org/10.1109/ROBOT.2010.5509714.

Binney, J., Krause, A., Sukhatme, G.S., 2013. Optimizing waypoints for monitoring spatiotemporal phenomena. Int. J. Robot. Res. 32 (8), 873–888.

Chen, W., Liu, L., 2021. Pareto Monte Carlo tree search for multi-objective informative planning. arXiv preprint arXiv:2111.01825.

Choset, H., Pignon, P., 1998. Coverage path planning: The boustrophedon cellular decomposition. In: Field and Service Robotics. Springer, pp. 203–209.

Deisenroth, M.P., 2010. Efficient Reinforcement Learning using Gaussian Processes. Vol. 9. KIT Scientific Publishing.

Dennis, Jr., J.E., Schnabel, R.B., 1996. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. SIAM.

Di Caro, G.A., Yousaf, A.W.Z., 2021. Multi-robot informative path planning using a leader-follower architecture. In: 2021 IEEE International Conference on Robotics and Automation. ICRA, pp. 10045–10051. http://dx.doi.org/10.1109/ICRA48506.2021.9561955.

Garrido, S., Moreno, L., Blanco, D., 2011. SLAM and Exploration using Differential Evolution and Fast Marching. pp. 81–98. http://dx.doi.org/10.5772/17376.

Garrido, S., Muñoz, J., López, B., Quevedo, F., Monje, C.A., Moreno, L., 2022. FM 2 path planner for UAV applications with curvature constraints: A comparative analysis with other planning approaches. Sensors 22 (9), 3174.

Ghaffari Jadidi, M., Valls Miro, J., Dissanayake, G., 2018. Gaussian processes autonomous mapping and exploration for range-sensing mobile robots. Auton. Robots 42 (2), 273–290.

Gómez, J.V., Lumbier, A., Garrido, S., Moreno, L., 2013. Planning robot formations with fast marching square including uncertainty conditions. Robot. Auton. Syst. 61 (2), 137–152.

Jadidi, M.G., Miró, J.V., Valencia, R., Andrade-Cetto, J., 2014. Exploration on continuous Gaussian process frontier maps. In: 2014 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 6077–6082.

Lawrence, N., 2003. Gaussian process latent variable models for visualisation of high dimensional data. Adv. Neural Inf. Process. Syst. 16.

Luo, W., Sycara, K., 2018. Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes. In: 2018 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 6359–6364.

Ma, K.-C., Liu, L., Sukhatme, G.S., 2017. Informative planning and online learning with sparse gaussian processes. In: 2017 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 4292–4298.

Muñoz, J., López, B., Quevedo, F., Monje, C.A., Garrido, S., Moreno, L.E., 2021a. Coverage strategy for target location in marine environments using fixed-wing UAVs. Drones 5 (4), 120.

Muñoz, J., López, B., Quevedo, F., Monje, C.A., Garrido, S., Moreno, L.E., 2021b. Multi UAV coverage path planning in urban environments. Sensors 21 (21), 7365.

Ranganathan, A., Yang, M.-H., Ho, J., 2010. Online sparse Gaussian process regression and its applications. IEEE Trans. Image Process. 20 (2), 391–404.

Sethian, J.A., 1999. Fast marching methods. SIAM Rev. 41 (2), 199–235.

USGS, United States Geological Survey Real-time data information. USGS.Gov | Science for A Changing World. URL https://www.usgs.gov/.

Viseras, A., Wiedemann, T., Manss, C., Magel, L., Mueller, J., Shutin, D., Merino, L., 2016. Decentralized multi-agent exploration with online-learning of gaussian processes. In: 2016 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 4222–4229.

Williams, C.K., Rasmussen, C.E., 2006. Gaussian Processes for Machine Learning. Vol. 2. No. 3. MIT Press, Cambridge, MA.

Xu, Y., Choi, J., Oh, S., 2011. Mobile sensor network navigation using gaussian processes with truncated observations. IEEE Trans. Robot. 27 (6), 1118–1131.

Yang, K., Keat Gan, S., Sukkarieh, S., 2013. A Gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with a UAV. Adv. Robot. 27 (6), 431–443.

Zhu, H., Chung, J.J., Lawrance, N.R., Siegwart, R., Alonso-Mora, J., 2021. Online informative path planning for active information gathering of a 3D surface. In: 2021 IEEE International Conference on Robotics and Automation. ICRA, pp. 1488–1494. http://dx.doi.org/10.1109/ICRA48506.2021.9561963.