Searching for Regions Out of Normal Conditions Using a Team of Robots

David Saldaña, Luiz Chaimowicz, and Mario F. M. Campos *

Computer Vision and Robotics Laboratory (VeRLab), Computer Science Department, Universidade Federal de Minas Gerais, MG, Brazil.
{saldana,chaimo,mario}@dcc.ufmg.br

Abstract. Searching for regions in abnormal conditions is a priority in environments susceptible to catastrophes (*e.g.* forest fires or oil spills). Those disasters usually begin with an small anomaly that may became unsustainable if it is not detected at an early stage. We propose a probabilistic technique to coordinate multiple robots in perimeter searching and tracking, which are fundamental tasks if they are to detect and follow anomalies in an environment. The proposed method is based on a particle filter technique, which uses multiple robots to fuse distributed sensor information and estimate the shape of an anomaly. Complementary sensor fusion is used to coordinate robot navigation and reduce detection time when an anomaly arises. Validation of our approach is obtained both in simulation and with real robots. Five different scenarios were designed to evaluate and compare the efficiency in both exploration and tracking tasks. The results have demonstrated that when compared to state-of-the art methods in the literature, the proposed method is able to search anomalies under uncertainty and reduce the detection time by automatically increasing the number of robots.

Keywords: Multi-robot systems, robotic sensor networks, particle filter, perimeter detection, level-curve tracking.

1 Introduction

Real-time monitoring is paramount in environments where disasters may occur at any moment and when human or animal lives are in danger. Disasters are usually initiated by anomalies which were not timely detected and possibly corrected or even reported. In most cases it would be highly desirable to not only detect, but also to identify the affected area, whose perimeter may change over time. A typical example is the monitoring of a forest, where not only the identification of the increase in temperature, possibly due to a fire is of utmost importance, but also to be able to determine the affected area in real time, which would be of great relevance to firefighters. Similarly, detecting and tracking anomalies are important tasks in several domains such as: oil spills in water bodies, radiation leaks from nuclear power plants, and algae bloom in lakes.

* The authors thanks to Professor Renato Asunção for his advises and contribution in the proposal. The authors also gratefully acknowledge the support of CAPES, CNPq and FAPEMIG.

Systems composed of cameras or multiple static sensors dispersed in the environment may be used for detection, but their capabilities are constrained when dynamic anomalies need to be tracked. Therefore, one feasible approach is to spread multiple mobile robots in the environment, which may be coordinated to navigate and to dynamically monitor changing anomalies in the environment. Complementary sensor fusion may be readily applied to estimate the sensed phenomenon based on the information acquired by the multiple robots, taking advantage of any additional information that may become available to provide a better partial observation. Such mobile robots, which are usually equipped with processors, wireless communication, and several sensors, constitute what has been known in the literature as Robotic Sensor Networks (RSN) [21,4]. Their locomotion system enables them to cover large areas and to adjust their location based on the environmental dynamics or other natural occurrences. In addition, RSNs have a great advantage over other monitoring techniques, such as wireless sensor networks or multiple static monitoring cameras, since they are able to dynamically modify their actions over time, which enables them to sense, detect, and also track anomalies in the monitored environment.

We use the term *anomaly* to designate an area in the environment where the value of a given physical variable is out of its typical range. An *anomaly* can be modeled as a gradient [8] or a (gradient-free) binary surface [12]. For instance, a gradient is generated by a physical phenomenon like temperature or light, which typically decay with distance. In this case, a robot would follow an iso-temperature or iso-illumination level-curve. In the gradient-free case, a robot would just identify if it is within or without the affected area and then proceed to identify the boundary of the sensed anomaly.

In this paper, we propose a probabilistic technique based on particle filters, to search, detect and track multiple dynamic anomalies. A group of anomalies can be seen as a multi-modal probabilistic distribution, and the mobile robots in the RSN move about in the environment while estimating this distribution by combining the information acquired from their sensors.

The rest of this paper is organized as follows: Section II describes the main techniques in the literature for perimeter detection and tracking. Section III presents the proposed model. Section IV details the implementation used to validate the proposed model and its results. Finally, Section V discusses the main findings, the conclusions and indicates directions for further investigation.

2 Related works

Bruemer et al. [4] developed one of the first studies on perimeter detection and tracking, a bio-inspired approach based on swarm intelligence to detect gradient-free chemicals. In their work, a robotic swarm detects the chemicals and communicates capturing and generating sounds. The prototype runs on low-cost robots deployed on a physical environment. With an uncoordinated method, those robots were able to detect multiple anomalies within the environment without using any tracking systems nor wireless communication.

Bachmayer and Leonard [1] proposed a bio-inspired technique for navigating on gradients with underwater robots in the ocean. Each robot behaves like a individual of a school of fish to find the most dense source of food by individually responding not only to local perception but also integrating shared information from its nearest neighbors. The authors further improved their work to enable tracking on gradients [7,8]. Their robots worked in pairs to determine their movement via an algorithm which used the measured potential between both robots. Their algorithm applies a planning technique to explore the underwater environment in three dimensions and defined virtual leaders to follow the defined path or to track the contour of the anomalies. Leadership was allowed to change depending on the position or actions while executing a specific plan [18,17].

Marthaler et al. [14,15] proposed a method based on the well-known computer vision deformable contour technique *snake* which was modified to detect gradient-free anomalies. Their method was defined as a kind of edge search algorithm by a group of autonomous mobile robots, which required local communication and interaction among them. In a subsequent work, the authors proposed an algorithm to detect and track gradient-free substances, known as UUV-Gas [12,10], where tests are performed with a single vehicle for evaluating and comparing the classical *bang-bang* algorithm performance [2]. The algorithm *Page's cumulative sum* (CUSUM) also integrates *bang-bang* algorithm and additionally includes a filter to increase the turning points by improving the accuracy of the sensing data. In 2009, J. Abhijeet [11] implemented the CUSUM algorithm on low-cost robots and subsequently Jin and Bertozzi refined this technique to create *improved CUSUM* [3].

Clark and Fierro [5,6] proposed a biologically inspired, hierarchical architecture for decentralized control and coordination that allows a robot swarm to locate and track a dynamic perimeter. One of their main features is a software architecture based on behavior-focused fault tolerant cooperative control. Their architecture defines two types of agents: Sensor agents, whose role is to sample environmental data and communicate useful information wirelessly with their robot neighbors and groupal agents in charge of receiving and processing all related information about the environment and located perimeters. The agent control is a model based on modes, which are represented by a hierarchical state machine. The hierarchical architecture that consists of four drivers that act according to the robots state. A main controller integrates three sub-controllers for: random coverage by spiral exploration; point attraction by potential field; and tracking. The strengths of that work are the following: the algorithm covers the entire process of anomaly detection (exploration, attraction and tracking), their model is tested through simulation and real prototype ground robots. Those sub-controllers internally also include collision avoidance. Their architecture is scalable and robust mainly due to the major states that are defined for the detection and tracking process, as well as different connections to change control status. Although their architecture is robust and defines some links to state control, the finite state machine may generate unnecessary state changes due to noisy sensors or small changes in the conditions in the environment.

3 A Model For Searching and Tracking

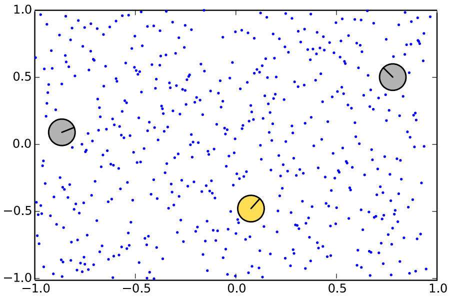
The problem of searching for anomalies with multiple robots can be stated as follows. Consider a team $\mathcal{R}$ of n robots, located in an environment defined in the Euclidean space $\mathbb{R}^2$. Let $\mathcal{W}$ be the robot's workspace in the environment. The problem is focused on detecting and tracking the perimeter that surrounds a dynamic shape $\mathcal{A}$, with $\mathcal{A} \subset \mathcal{W}$. $\mathcal{A}$ is the representation of a single or multiple anomalies. The size, form and position of each anomaly may dynamically change over time.

Each robot $r \in \mathcal{R}$ has sensing capabilities to identify the anomaly in time t. The measured value z_t represents the binary sensor's output, which determines if the area sensed by the robot is located inside the area of the anomaly location $loc(r) \subset \mathcal{A}$ for binary anomalies or if the sensed value is greater than the reference value of the level-curve. $Z_t = [z_t^{[r_1]}, z_t^{[r_2]}, ..., z_t^{[r_n]}]$ is the set of all measurements performed by each robot at time t. Each robot r can gather its global location, and the uncertainty of its measurements is known *a priori* (*e.g.* by sensor calibration). Additionally, a communication network allows any robot to send and receive messages from other robots. Every robot $r \in \mathcal{R}$ can communicate its sensed value $Z_t^{[r]}$ in time t.

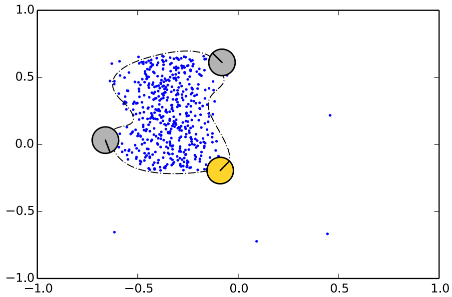
As described in the previous section, searching, attraction and tracking are behaviors performed with a high level control for this problem. In this work, robot navigation and coordination are based on the uncertainty of the anomaly; every robot tries to visit the nearest spot with highest likelihood of being an anomaly. The shape of an anomaly can be seen as a multi-modal distribution, and our method attempts to estimate this distribution using *particle filter* technique. Searching and tracking are the main robot behaviors, but attraction is implicit in the searching process in order to avoid erroneous motions due to noisy data. Our approach also supports multiple anomalies and robot coordination with concurrent exploration and tracking.

3.1 Particle Filter for Anomaly Estimation

The particle filter technique has many applications since it offers a probabilistic method that converge to multi-modal probability distributions. In robotics this technique has been thoroughly applied in localization, mapping, target tracking, and many other tasks [22]. In our specific scenario, we want to identify the anomaly shape by using multiple robots. There are three relevant reasons for using particle filter in this context. (1) If an anomaly exists in the environment, exploration with multiple robots leads the particles to converge to the shape of the anomaly, otherwise the robots will continue to explore the environment which tends to reduce the uncertainty of the occurrence of an anomaly along the traversed path. (2) Each robot has an independent representation of the updated map, based on the robot's measurements and the communicated information from other robots. The set of particles offer a representation of the areas in the map which have been recently covered, incorporating the uncertainty in the non-visited zones and reducing the belief of the sensed path in time. This updating process could be modeled as a complex geometric process, but under



(a) Spreaded particles



(b) Converged particles

Fig. 1: Particle representation for the yellow robot. The other robots of the team are represented as gray circles, blue points for particles, and the perimeter of the anomaly is a non-continuous black line.

this approach, it can be computed with simple arithmetic operations on the particle's motion. (3) The natural, randomized movement of the anomaly is incorporated in the particle's motion. Any additional information on the anomaly can be added as *a priori* data in order to improve the estimation updating of the map.

In [16], a particle filter approach was used to track a target with multiple robots. In the initial state, when robots start searching without previous knowledge about the anomaly existence, an anomaly can be seen as a target. We assume that the map of the environment has been provided *a priori* and a global localization system is available. Therefore, our goal is to estimate the probability distribution of the anomaly x_t at time t, based on the estimated belief on x_t (Eq. 1) given the robots measurements $Z_{1:t}$ and the estimated anomaly motion $u_{1:t}$ in time.

$$Bel(x_t) = P(x_t|Z_{1:t}, u_{1:t}). \tag{1}$$

Every particle can be depicted as a point on map, and it represents the possibility of having an anomaly or apart of it, thereof, in that location. The objective of this technique is to dynamically converge the particles to the anomaly's shape. In the initial state, without a previous estimation of the anomaly, the particles are spreaded along the environment (Figure 1a). The objective is concentrating the largest number of accumulated particles (blue points) within the perimeter of the anomaly, as illustrated in Figure 1b. As each robot has its own particles, particles represents the robot's belief about the existence of an anomaly in the environment.

Particle filter require two fundamental models to be defined: The motion and the updating of the particles.

Motion model Each robot r at time t has a set of particles $\mathcal{X}_t = [x_t^1, x_t^2, ..., x_t^m]$, that represent a point in the environment with the possibility of having an anomaly. Equation 2 represents a random motion model, when every particle just moves a random distance from time $t-1$ to t based on a Gaussian distribution. The particle's motion is defined by a (2×2) covariance matrix Σ that determines the spreading velocity of the anomaly. Eq. 2 describes the motion model for particle i, $i = 1, 2, ..., m$, based on the last estimation $x_{t-1}^{[i]}$ and a random motion $u_t = \mathcal{N}(0, \Sigma^2)$:

$$p(x_t^{[i]}|u_t, x_{t-1}) \sim \mathcal{N}(x_{t-1}^{[i]}, \Sigma^2). \tag{2}$$

This model is applied when there is no additional information about the anomaly behavior, then a random motion is assumed. Having information *a priory* about the anomaly may help to enhance the exploration process as long as particles simulate that behavior. For example, a fire in a forest normally has a random movement, but if the fire is pushed by the wind, and we know the wind's strength and orientation, then a better estimation about the position of the anomaly can be predicted.

Updating Model Estimation about the anomaly's distribution is iteratively updated by using each robot's sensor readings. In each iteration, every particle is re-sampled based on weights. Each particle begins with a normalized weight, and depending on its position and the robot's sensed values, that weight may be modified. Equation 3 determines how the weight of a particle i is updated based on sensor observations.

$$w_t^{[i]} = \begin{cases} a, & \text{if } z_t = 0 \\ b, & \text{if } z_t > 0 \\ 1/m, & \text{outside sensor range} \\ 0, & \text{outside the map,} \end{cases} \tag{3}$$

where the constant a is a small value that represents a low probability of the existence of an anomaly in the sensed area when no anomaly is identified by the robot's sensors ($z_t = 0$); b is a value ≥ 1 used to increase the particle's weight given that an anomaly has been detected and c is an intermediate value to represent the uncertainty of a non sensed area at time t. In the experiments we used $a = 0.1$, and $b = 1.3$.

When every particle has its own weight, the group of particles is re-sampled [9] to randomly clone particles proportionally to their updated weight. After several iterations, the result is an accumulation of particles in places with more possibility of the existence of an anomaly. When no anomaly is detected in the environment, robots will try to navigate towards the areas with particle accumulations in order to visit the most probable spots with anomalies.

3.2 Searching for Anomalies

Each robot running the navigating and searching processes must attain the following objectives: (1) maximize the number of visited particles, giving priority to the nearest one, (2) maximize the distance to other robots, (3) maximize the distances to obstacles and map borders. To fulfill these requirements, a few techniques could be applied like Partially Observable Markov Decision Processes

(POMDPs) and potential fields in a discretized map [16]. In both cases, a map discretization is required to create a grid. For large cells, the robot will move through cell centers, whereas for small cells, complex motions for non-holonomic robots are likely to be generated, especially when the best target cell is adjacent to the current robot cell, and the path's orientation is different from the current orientation of the robot.

In our approach, we assume that navigation is based on the potential field technique in a continuous space. This well known technique is based on the physical model of electrical charges, assuming that a robot is a positive electrical charge which is attracted by all the negative charges (particles generated by the defined filter in the previous section). Other robots and obstacles are also modeled as positive charges that repel the robot. Therefore, the robot's velocity and orientation are computed as the vector sum of all forces involved.

As the number of particles must be large (more than thousand) for good approximations, similar particles may be grouped into a small number of clusters, where each cluster is represented by a centroid. We have used the *k-means* clustering method to group the particles. However, one difficult issue with this method is defining k, the number of groups. In our approach, we computed $k = 3|\mathcal{R}|$ in order to give three choices for each robot.

The force generated by a centroid $c \in \mathcal{C}$ is proportional to the number of particles in the cluster and it is computed based on Coulomb's law (Eq. 4):

$$|F_c| = \alpha \frac{|q_r||q_c|}{d^2}, \tag{4}$$

where α is a constant; $|q_r|$ is the robot's charge, defined as a unitary charge $|q_r| = 1$; $|q_c| = m_c/k$ is the cluster's charge, which is proportional to the total of particles in the cluster m_c; and d is the distance between the robot r and the cluster's centroid c. Finally, the resultant force that acts on each robot F_r is computed based on the attraction force by the clusters and repealed forces F_s generated by each of the other robots, Eq. 5:

$$F_T = \sum_{c \in \mathcal{C}} F_c - \sum_{s \in \mathcal{R} - \{r\}} F_s - \sum_{o \in Obs} F_o \tag{5}$$

The force vector $F_T = \langle \rho, \phi \rangle$ may be decomposed into its magnitude ρ and orientation ϕ. The robot navigates with constant linear speed $v = K_v$ in the direction of the resultant force. Angular speed is defined by a *Proportional-Derivative (PD)* controller as described by Equation 6,

$$\omega = K_1\,\phi + K_2\,\dot{\phi}, \tag{6}$$

where K_1, and K_2 are the *PD* constants.

3.3 Tracking an Anomaly

The tracking process starts when a robot detects an anomaly and then it starts to border the whole boundary to estimate its shape. In a previous work [20], a PID (Proportional, Integrative, and Derivative) control was used for tracking gradients. It used an analog sensor such as a thermometer or light-meter. We have extended that approach for a different kind of sensor – a RGB camera –, which makes it possible to track binary anomalies running a PID controller on the angular velocity ω. Meanwhile, linear velocity v is constant.

The control of the linear speed is based on the distance to other robots in order to avoid collisions (Eq. 7) and the angular speed is computed based on the gradient of the anomaly concentration, ∇c_r in the area (Eq. 8). This gradient can be estimated by the time series of values acquired from a punctual sensor measurements or by employing multiple spatially distributed sensors [13].

$$v = v_{track} - \frac{K_{track}}{d(r, r')}, \tag{7}$$

$$\omega = K_3(\nabla c_r - \tau) + K_4\, \nabla \dot{c}_r, \tag{8}$$

where K_3 and K_4 are PD control constants, v_{track} is the maximum linear speed, K_{track} is a proportionality constant, and r' is the robot in front of robot r. In anomaly tracking, one of the most used methods is the *bang-bang* [3,11] , which can be emulated with the model of Eq. 8 and setting $K_4 = 0$. However, by sintonizing K_4, oscillations and convergence time is reduced. The linear velocity v is just proportional to the closest robot or obstacle in front of it.

3.4 Coordination

Each robot has its own particle set to update its beliefs about the world. The communication is based on broadcasted messages but their contents is very small (*e.g.* a few bytes). Every robot broadcasts its position and the sensed values. Sensor fusion is distributed and complementary, and for this purpose a robot does not depend on the others, but it can combine its current sensed data, historical measurements and all the information received from other robots. For example, if one value is missing (*e.g.* $z_t^{[r_i]}$), the algorithm can still proceed with the other measurements ($Z_t - \{z_t^{[r_i]}\}$).

4 Experiments and Results

A previous work in the literature [6] has been chosen as baseline since it is a robust distributed architecture that includes exploration, tracking, collision avoidance, and potential attraction. Additionally, our method has been tested on virtual and real robots. We implemented our method and the baseline in the Robot Operating System platform (ROS) [19].

Initially, we implemented and tested on the ROS-Gazebo 3D simulator and subsequently on physical robots in the laboratory. Each real and virtual robot is equipped with the same components: Onboard-computer for processing and communication by IEEE 802.3.11 (WIFI); one RGB camera, which is used as the anomaly sensor, and the *iCreate* base as differential mobile platform. Figure 2 shows the Gazebo environment with four robots, and those that are tracking

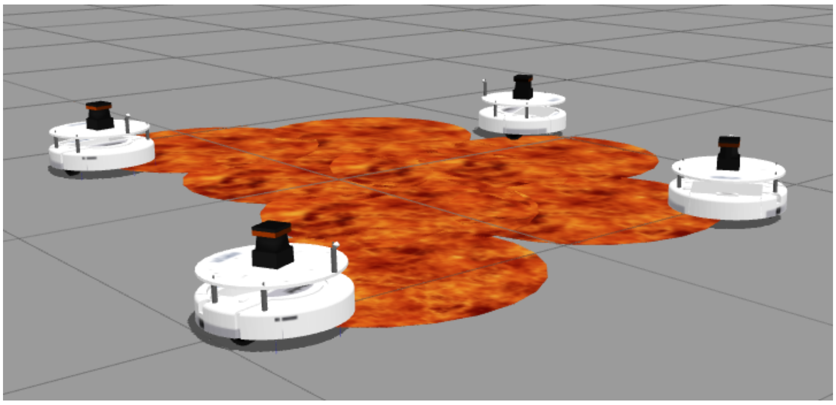


Fig. 2: Simulated environment with four robots tracking an anomaly.

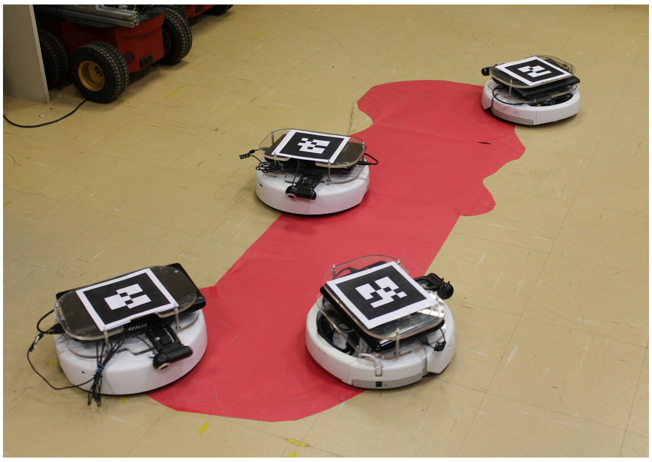
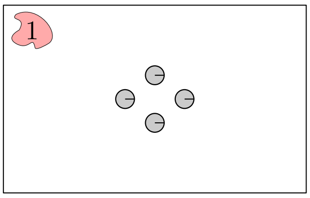


Fig. 3: Physical robots tracking an anomaly.

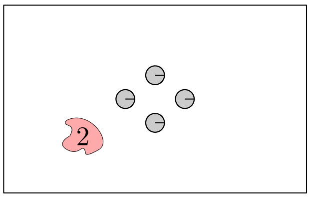
the anomaly (represented by a rug with a texture of fire). In this simulated environment, robot localization is obtained from a gazebo service.

Figure 3 shows four real robots with localization tags. For localization purposes, we use Dragonfly CCD Firewire cameras with a wide-angle lenses mounted on the ceiling. A localization computer server is connected to the cameras and runs the software *AR-Alvar-Track* to determine the localization of each robot based on the markers placed on the top of each one of them. In the accompanying video sample [`http://youtu.be/wG8WdsW_JiM`] can be seen results from simulation and real robots.

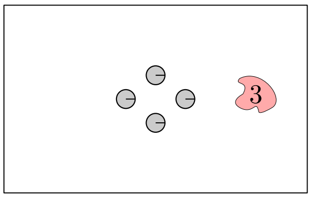
We have defined two configurations for the four robots and five different cases of anomaly positions. Figure 4a shows a common configuration where robots start near to a central area. In Figure 4e, the map is divided in four parts and each robot occupies their respective centers, which is the best configuration



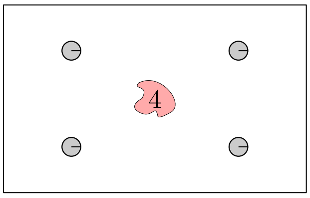
(a) Configuration 1: robots near to the center of the map.



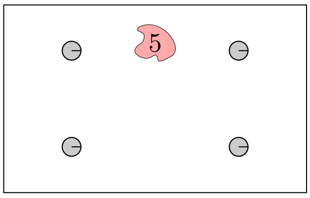
(b) Configuration 2: robots distributed along the map.



(c) Configuration 2: robots distributed along the map.



(d) Configuration 2: robots distributed along the map.



(e) Configuration 2: robots distributed along the map.

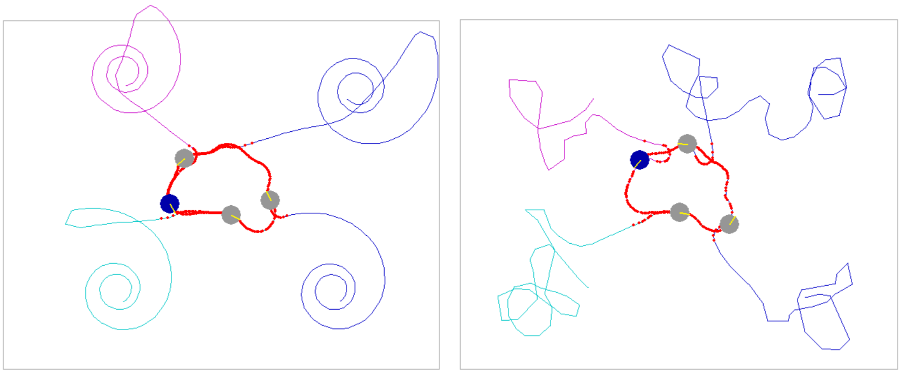
Fig. 4: Scenarios for validation and comparison

for spiral exploration (*e.g.* the exploration method of the baseline), because it reduces the redundancy of explored spaces in the map. Figures 5a and 5b show the robot paths in the detection and tracking tasks for the baseline and the proposed method, respectively. Spiral exploration is simple and fast. It can be considered as a greedy algorithm, but it is not complete, since it does not explore every spot in map, inasmuch as an anomaly may appear anywhere. For that reason, the five anomalies in Fig. 5a are located in covered places by at least one of the spiral lines. In Figure 5b it can be seen that the robot paths are more stochastic in nature than those in the baseline (Fig. 5a); however, on average, the traveled distance until boundary detection is shorter for our method (proposal=8.7m and baseline=10.7m). A more detailed comparison and analysis is presented in the next section. Both techniques were executed at least ten times for each anomaly case and two metrics were defined for comparison: detection time and tracking time.

4.1 Results for detection time

Detection time is measured from the moment the robots starts in the initial positions (Figures 4a and 4e) until one or more robots detect an anomaly. This measurement is an indication of the efficiency to explore the environment since the anomaly may be located in any of the five places represented in Fig. 5b.

Figure 6 shows a box-plot to compare the baseline and our method in the exploration process. We assume an error distributed normally and represent the confidences interval as notches. The spiral exploration method (baseline)



(a) Spiral exploration (baseline). (b) Exploration based on particle filter.

Fig. 5: Robot paths for exploration and tracking in scenario 4

finds the anomaly with standard deviation between 1.0 and 2.0 seconds. It has a small variation of 2.8% because the detection is almost always at the same point of the spiral. In contrast, our method has standard deviation between 5.5 and 13.1 seconds, since robot navigation is associated with the particles random movement. However, our method detects anomalies approximately two times faster than the baseline, on average. Case 3 and 5 are the cases, where the methods are similar based on a *t-test*. Although cases 4 and 5 are most favorable for the baseline, the spiral exploration require many iterations to arrive to anomaly 4 and the detection time increases.

4.2 Results for tracking time

Tracking time is measured since an anomaly is detected until its perimeter is completely defined by one or more robots. The perimeter identification can be reduced when some robots border the anomaly in different sections, that is why our method works better for cases 1,2, and 3. One robot detects the anomaly and the others continue exploring the environment where there is a possibility of anomaly. Robots navigate attempting to follow the uncertainty. Baseline has a different behavior, when a robot detects an anomaly all the other robots go to the identified point. This generates redundancy, since robots explore the same detected point and, additionally, it becomes a possible point of collision. On the one hand, only in cases 4 and 5, for which the redundant method works better because there is an anomaly between the robot and the detected point, the robots cover the anomaly by different sections. On the other hand, cases 1,2, and 3 are the most common, because the anomaly may appear at any time and the robots may not be in the optimal position.

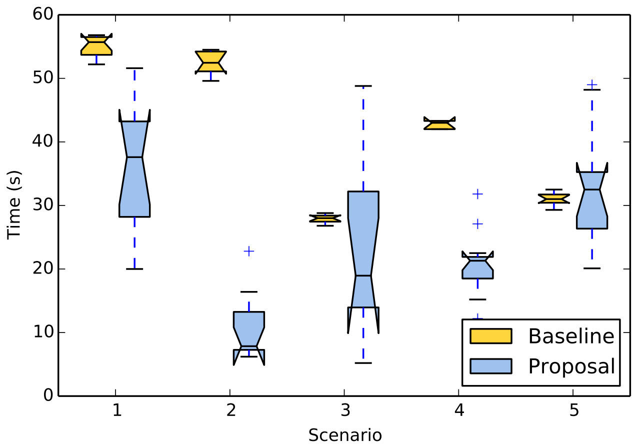


Fig. 6: Comparison for detection time in 5 anomaly cases.

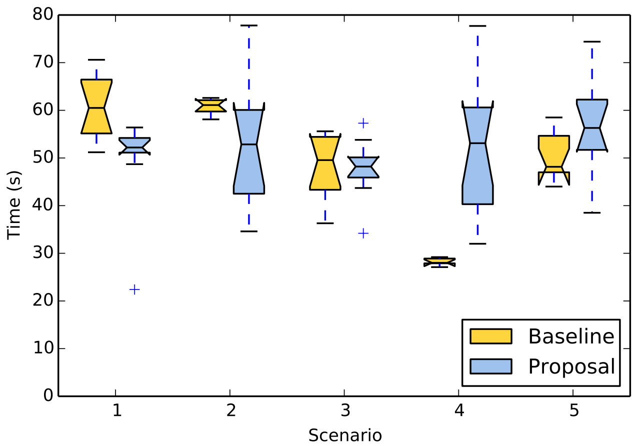


Fig. 7: Comparison for time to surround an anomaly in 5 anomaly cases.

4.3 Results for the number of robots

Now we want to analyze the behavior of the proposal when the number of robots increase. In the baseline, the spiral behavior does not improve the value of the metrics because more robots only increase the redundancy on the same visited places.

In our approach, the robots try maximize the places that were not visited before and also maximize the distance among the other robots. By this reason, when there are more robots in the exploration process, the time for detection

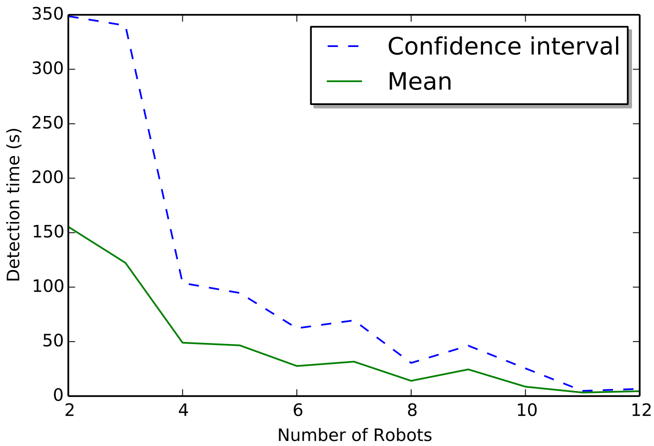


Fig. 8: Detection time by increasing the number of robots.

is reduced. Figure 8 shows the results for the scenario 1, where the experiments where replicated 85 times. We can see that the detection time reduces exponentially with the number of robots. The limit is estimated as the confidence interval in one direction with a confidence of 95%. The oscillatory behavior of the descent curves for confidence interval and mean is related to the initial configuration of the robots. When a new robot is added, it reduces the detection time if it is located near to the anomaly, in other case, the impact of the new robot influences with less impact.

Therefore, in an ambient of dimensions $80m^2$, an anomaly is detected in less than 6.6 seconds with 12 robots (confidence of 95%).

5 Conclusions and Future Work

In this article, we proposed a probabilistic distributed coordination method for multiple robots used in the task of anomaly detection and tracking. We have experimentally shown, both in simulation and with real robots, that it improves the searching and coordination processes by taking advantage of the particle filter technique. Experimental results demonstrated efficiency in exploration and tracking for most of the cases. Furthermore, it offers additional advantages such as: Support for multiple anomalies, fully environmental exploration, and predictions for dynamic anomalies.

On the one hand, in the implementation, the ROS platform offered a very useful development environment to implement programs for robots; it works well regardless if robots are simulated or real. On the other hand, in a real deployment, parameter configuration for the proposed model has shown to be critical, since a poor parametrization may generate collision among robots or even induce navigation outside the map.

As a future work, this method could be extended to three dimensions with the use of heterogeneous robots such as aerial and ground vehicles to detect falling rocks and avalanches from a mountain. In real scenarios, communication delay is a very common problem in robot message exchange. Dealing with this issue can be included in the proposed model by adding a belief function for updating the particles based on the magnitude of the delay.

References

1. Bachmayer, R., Leonard, N.: Vehicle networks for gradient descent in a sampled environment. In: Procedings of 41s IEEE Conference on Decision and Control. No. December (2002)
2. Bertozzi, A., Kemp, M.: Determining environmental boundaries: Asynchronous communication and physical scales. In: Proc. Block IslandWorkshop Cooperative Control (2004)
3. Bertozzi, A.L.: Environmental boundary tracking and estimation using multiple autonomous vehicles. 2007 46th IEEE Conference on Decision and Control pp. 4918–4923 (2007)
4. Bruemmer, D.: A robotic swarm for spill finding and perimeter formation. Spectrum: International Conference on Nuclear and Hazardous Waste Management (2002)
5. Clark, J., Fierro, R.: Cooperative hybrid control of robotic sensors for perimeter detection and tracking. Proceedings of the 2005, American Control Conference, 2005. pp. 3500–3505 (2005)
6. Clark, J., Fierro, R.: Mobile robotic sensors for perimeter detection and tracking. ISA transactions 46(1), 3–13 (Feb 2007)
7. Fiorelli, E., Bhatta, P., Leonard, N.E.: Adaptive Sampling Using Feedback Control of an Autonomous Underwater Glider Fleet. Aerospace Engineering (August) (2003)
8. Fiorelli, E., Leonard, N.E.: Exploring scalar fields using multiple sensor platforms: Tracking level curves. 2007 46th IEEE Conference on Decision and Control pp. 3579–3584 (2007)
9. Hol, J.D., Schon, T.B., Gustafsson, F.: On resampling algorithms for particle filters. In: Nonlinear Statistical Signal Processing Workshop, 2006 IEEE. pp. 79–82. IEEE (2006)
10. Hsieh, C., Marthaler, D., Nguyen, B., Tung, D., a.L. Bertozzi, Murray, R.: Experimental validation of an algorithm for cooperative boundary tracking. Proceedings of the 2005, American Control Conference, 2005. pp. 1078–1083 (2005)
11. Joshi, A., Ashley, T., Huang, Y.R., Bertozzi, A.L.: Experimental Validation of Cooperative Environmental Boundary Tracking with On-board Sensors. Control pp. 2630–2635 (2009)
12. Kemp, M., Bertozzi, A., Marthaler, D.: Multi-UUV perimeter surveillance. In: IEEE OES Workshop on Multiple AUV Operations (2004)
13. Li, S., Guo, Y., Bingham, B.: Multi-robot cooperative control for monitoring and tracking dynamic plumes. In: IEEE International Conference on Robotics and Automation. Proceedings. ICRA 2014. IEEE (2014)
14. Marthaler, D., Bertozzi, A.: Collective motion algorithms for determining environmental boundaries. In: SIAM Conference on Applications of Dynamical Systems (2003)
15. Marthaler, D., Bertozzi, A.: Tracking environmental level sets with autonomous vehicles. Journal of the Electrochemical Society 129, 2865 (2003)

16. Mottaghi, R., Vaughan, R.: An integrated particle filter and potential field method for cooperative robot target tracking. In: IEEE International Conference on Robotics and Automation. ICRA 2006. IEEE (2006)
17. Ogren, P., Fiorelli, E., Leonard, N.: Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment. IEEE Transactions on Automatic Control 49(8), 1292–1302 (Aug 2004)
18. Ogren, P., Fiorelli, E., Leonard, N.E.: Formations with a mission: Stable coordination of vehicle group maneuvers. In: Symposium on mathematical theory of networks and systems (2002)
19. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA workshop on open source software (2009)
20. Saldana, D., Ovalle, D., Montoya, A.: Improved algorithm for perimeter tracking in robotic sensor networks. In: XXXVIII Latin American Conference on Informatics (CLEI). pp. 1–7. IEEE (2012)
21. Sibley, G., Rahimi, M., Sukhatme, G.: Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks. In: IEEE International Conference on Robotics and Automation. Proceedings. ICRA 2002. vol. 2, pp. 1143–1148. IEEE (2002)
22. Thrun, S.: Probabilistic robotics. Communications of the ACM 45(3), 52–57 (2002)