A Distributed Multi-Robot Approach for the Detection and Tracking of Multiple Dynamic Anomalies

David Saldaña[1], Renato Assunção[1], and Mario F. M. Campos[1]

***Abstract*— In many cases, large area disasters could be possibly be prevented if the incipient small-scale anomalies are detected in their early stages. A way to accomplish this would be to have multiple sensors deployed in disaster prone areas to detect anomalies. However, compared to static sensor networks, robotic sensor networks offer advantages such as active sensing, large area coverage and anomaly tracking. This paper addresses the problem of coordinating and controlling multiple robots for the detection of multiple dynamic anomalies in the environment. The main contribution of the work is a combined approach for the effective exploration under uncertainty, the anomaly tracking, and the autonomous on-line allocation of agents. Robots explore the work area maintaining the history of the sensed areas to reduce redundancy and to allow for full-map coverage. When an anomaly is detected, a robot autonomously determines how to either track the anomaly or to continue the exploration of the environment, depending on the size of the anomaly, which is estimated by the length of the perimeter of the enclosing polygon. We show results of our methodology both in simulation and with actual robots which have demonstrated that robots can autonomously and distributively be allocated to track or to explore depending on the behavior of the detected anomalies.**

***Index Terms*— Multi-robot systems, robotic sensor networks, particle filter, perimeter detection, tracking.**

I. Introduction

Large area disasters are usually triggered by small-scale anomalies in small areas which could possibly be detected in their early stages. The past decade has seen effective proposals to approach this problem with the deployment of multiple mobile sensors for monitoring disaster prone areas. Compared to static sensor networks, robotic sensor networks offer advantages such as active sensing, large area coverage and anomaly tracking. This paper addresses the problem of coordinating and controlling multiple robots for the detection of anomalies in the environment with accurate estimations of their localization and area. The main contribution of the work is a combined approach to effectively detect and track dynamic anomalies with a team of robots.

In this paper, 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. Existing approaches for detecting and tracking anomalies can be broadly divided into three classes: searching, anomaly tracking, and integrated techniques which combine the two first ones. In searching, mobile robots equipped with sensors are scattered in the environment and are tasked to find anomalies. In a related work, Bruemer et al. [1] propose a bio-inspired algorithm to detect gradient-free chemical contamination area. It uses social potential fields to cover the full area to search and find anomalies, which in their work are assumed to be static. An anomaly can be seen as a punctual target at the first stage, in [2], and [3] are used probabilistic methods to achieve this task using multi-robot systems. Marthaler et al. [4] propose a method based on deformable contours (e.g. snake) which was modified to position multiple robots around a gradient-free anomaly. In [5] the authors propose an algorithm to distribute multiple sensors around a static region in order to reduce latency, and to increase the precision of the perimeter estimation. A cooperative algorithm based on statistical estimation to detect areas with rapid changes is proposed in [6].

In the second category, the main goal is to track anomalies, and most approaches are control-based. In [7], a bio-inspired control model is proposed to navigate (climb and descent) in gradients with a team of robots. A multiple underwater vehicle control model is developed in [8], where cooperative gliders are coordinated with adaptive sampling strategies, allowing teams of three robots to navigate along gradients with different formations. An experimental validation of the Page's cumulative sum algorithm (CUSUM) [9] is described in [10]. This work experimentally compares latency and precision for tracking anomalies with the classic *bang-bang* type steering controller [11] with the CUSUM algorithm. Susca et al. [12] propose a way to approximate a convex static contour with a polygon in order to track such contour with a robotic sensor network. Also, gradient descent laws based on performance metrics such as the area of the inner, outer, and "outer minus inner" approximating polygons. A recent work [13] proposes a control method to track dynamic plumes (pollutants released at a point source) based on the advection-diffusion model. The robot control they apply is analytically constructed with probable convergence for chemicals or liquid substances poured in a marine environment. In [14], a control model is proposed to track multiple gradients with UAVs, where robots starts following the gradient and then track level curves.

In the third category are the methods which integrate both detection and tracking. The objective is to develop systems that improve the synergy between the two aforementioned processes. On such a work is [15], where the authors model a finite state machine to switch among behaviors such as: spiral exploration, obstacle avoidance, potential attraction, and tracking. However, spiral exploration does not cover the full map and has high redundancy in explored zones [16]. In [17] the authors extend the robot formation based on the *snake* algorithm [4] to track an anomaly. [18] deal with the

[1]D. Saldaña, R. Assunção and M.F.M. Campos are with the Computer Vision and Robotics Laboratory (VeRLab), Computer Science Department, Universidade Federal de Minas Gerais, MG, Brazil. E-mails: {saldana, assuncao, mario}@dcc.ufmg.br

*The authors also gratefully acknowledge the support of CAPES, CNPq and FAPEMIG.

problem of forest fire monitoring with multiple *Unmanned Aerial Vehicles*, in which onboard cameras detect the fire and image based techniques are used to detect the perimeter of the affected zone.

Our work falls in the third category, since we propose a probabilistic approach that integrates full-map coverage and tracking of multiple dynamic anomalies with a team of robots. In the searching phase, robots are guided toward the spots with high probability of the existence of an anomaly (Section III). When an anomaly is detected, the robots autonomously determine how to either track the anomaly or to continue exploration, depending on the size of the anomaly, which is estimated by the length of the perimeter of the enclosing polygon (Section IV).

The main contributions of our approach are: (1) an algorithm for the coordination of a team of robots that are tasked to find and track multiple anomalies. This is an important feature for real world applications, since in some contexts, such as fire in a forest or algae bloom in a lake, more than one anomaly may appear/disappear along the time. Among previous works, we have found that [1] approaches the problem of searching for multiple anomalies in an uncoordinated way, where an anomaly is detected but not tracked. Our approach allows for autonomous on-line allocation and deallocation of robots during the tracking process, depending on the size of the anomalies, which we assume, may dynamically vary. In our approach, a growing anomaly demands closer attention and demands that more robots be allocated throughout mission execution. Meanwhile, if an anomaly is shrinking, robots formerly allocated to that anomaly may be released to explore other areas with higher priority; (2) Efficient exploration. The exploration under uncertainty maintains the history of the sensed areas to reduce redundancy and to allow for full-map coverage.

II. Problem Statement

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 workspace for all robots in the environment. Within this workspace anomalies, $\mathcal{A}_i \subseteq \mathcal{W}, \forall i \in \{1, 2, ..., m\}$, may dynamically appear, which are are modeled as compact areas with nonempty interior, and it is assumed that there are no intersections among them, that is, $\bigcap_i \mathcal{A}_i = \emptyset$.

Let P_i be a polygon defined the line segments that connect a sequence of points $\{p_1, p_2, ..., p_l\}$, with $l \geq 3$, and which completely enclose anomaly $\mathcal{A}_i$ respectively. As the anomalies are dynamic (size, shape, and position can change over time), we denote the anomaly i in time t as $\mathcal{A}_i^t$. We assume that the anomalies have a slowly-moving behavior for boundaries, which is less than robots' velocity.

Problem 1 (Full-map searching for anomaly detection): How to coordinate a team $\mathcal{R}$ of robots to persistently patrol the full-map $\mathcal{W}$ in order to detect, and enclose with a Polygon P_i, each anomaly $\mathcal{A}_i$.

Problem 2 (Tracking of multiple dynamic anomalies): How to allocate robots (possible all n) to track multiple dynamic anomalies, depending on each anomaly size (e.g perimeter of the enclosing polygon). In such a way larger anomalies require more robots for tracking.

A. Available Information

Each robot $r_j \in \mathcal{R}, \forall j \in \{1, 2, ..., n\}$ has sensing capabilities to acquire local information, including its pose, and the concentration levels of the anomaly in the environment. The anomaly concentration sensed by r_j at sampling point $p \in \mathcal{W}$ is defined as $c = c(t, r_j, p)$, and its gradient is denoted as $\nabla c = \nabla c(t, r_j, p)$. This gradient can be estimated by the time series of values acquired from a punctual sensor measurements or by employing multiple spatially distributed sensors (e.g. from mobile robots [13]). The area occupied by the anomalies obey $c > \tau$, when $c \in \cup_i \mathcal{A}_i$, where τ is a threshold.

In addition, we assume that each robot can communicate with all other robots of the team by broadcasting messages. Therefore, at time t, each robot can receive the samples of all other robots or polygons, if detected. Let $\widetilde{P}_i \approx P_i$ be an approximate of the exact enclosing polygon P_i, which is estimated by each robot and broadcasted through the communication network.

B. Robot Model

Our model assumes that the robots are non-holonomic, and their pose is represented by the vector $(x, y, \theta)^T$. We use the *velocity control model*, which assumes that a robot may be controlled by issuing both the rotational and the translational velocity $u_t = (v_t, \omega_t)^T$.

III. Searching for anomalies

In this section, a method to solve Problem 1 is proposed. We treat the anomalies in the environment as a multi-modal distribution that can change dynamically at each time step. Our method tries to incrementally improve the estimation of the distribution by merging robot measurements and simulating the random behavior of the anomaly. Then we try to approximate the anomaly shape by a polygon.

A modification of the particle filter technique [2] is developed since it offers a probabilistic method that converges to multi-modal probability distributions. Robot navigation and coordination are based on the uncertainty about anomalies, with every robot trying to visit the nearest spot with highest likelihood of being an anomaly.

A particle can be considered as a point on a map, and it represents the probability of having an anomaly in that location. Each robot has its own set of particles $\mathcal{X}_t = \{x_t^{[1]}, x_t^{[2]}, ..., x_t^{[N]}\}$, which represents its belief over the possible anomalies in the environment. An accumulation of particles in a region may indicate a high probability of having an anomaly there. As a consequence, the robot will try to pursue it in order to reduce the uncertainty at that specific location.

At the initial state, robots start searching the area without previous knowledge of the existence or position of any anomaly. Therefore, the N particles are spread based on a uniform distribution within $\mathcal{W}$. The particle filter is an iterative process that requires two fundamental models to be defined: the motion and the updating of the particles.

1) Motion model: A random motion model displaces a particle in 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 in order to capture the spreading velocity of the anomaly. Equation 1 describes the motion model for particle k, $k = 1, 2, ..., N$ based on the last estimation, $x_{t-1}^{[k]}$, and a random motion with normal distribution $\mathcal{N}(0, \Sigma)$:

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

This model is used when there no additional information about the behavior of the anomaly is available. In this case, a random motion is assumed. Although in this paper no *a priori* information about the dynamic behavior of the anomaly is used, this information may help to enhance the exploration process as long as the particles simulate that behavior. For example, a fire in a forest evolves in an apparently random fashion, but if the wind is blowing, the fire will tend to spread in the direction of the wind, and if the wind's strength and orientation is known, a better estimation of the evolving anomaly may be obtained.

2) Updating Model: The estimation of the anomaly's distribution is updated using the robot's sensor readings. At each iteration, every particle is re-sampled based on weights. Equation 2 determines how the weight of a particle k is updated based on sensor readings.

$$w_t^{[k]} = \begin{cases} a, & \text{if } c < \tau \\ b, & \text{if } c \geq \tau \ \vee \ x_t^{[k]} \in \cup_i P_i \\ 0, & \text{if } x_t^{[k]} \notin \mathcal{W} \\ 1/m, & \text{outside sensor range,} \end{cases} \tag{2}$$

where the constant a is a small value that represents a low probability of the existence of an anomaly at a given point when no anomaly is sampled ($c = 0$); b is a value ≥ 1 used to increase the particle's weight given that an anomaly is sampled or particle is inside an *available-polygon* $x_t^{[k]} \in \cup_i P_i$; value 0 is set if the particle is outside the map $x_t^{[k]} \notin \mathcal{W}$; and for a point not sampled at time t, a normalized value $1/N$ is set. (e.g. in the experiments we used $a = 0.1$, and $b = 1.3$).

When each particle has its own weight, the group of particles is re-sampled to randomly clone particles proportionally to their updated weight [19]. After several iterations, the result is an accumulation of particles in places with higher possibility of the existence of an anomaly. When no anomaly is detected in the environment, the robots will try to navigate towards the areas with particle accumulations in order to visit the spots which most probably hash anomalies.

We define two working modes: The exploration and tracking modes. When a robot does not detect an anomaly locally, it starts to explore the environment by pursuing particles in order to visit places with probable anomalies.

A. Exploration

The exploration process must attain the following objectives: (1) maximize the number of visited particles, giving priority to the nearest one, (2) maximize the distance to the other robots, (3) maximize the distances to the obstacles and to the map borders. Therefore, we use potential fields to navigate in the continuous space [20]. 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. 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.

The forces acting on robot r_j are generated by particles $x \in \mathcal{X}$, other robots $r_k \in \mathcal{R} - \{r_j\}, \forall k \neq j \in \{, 1, 2, ..., n\}$, and obstacles $o \in Obs$ (map borders are also introduced and are model as obstacles). Eq. 3 computes the resulting force magnitude for any entity $z = \{x, r_k, o\}$ with constants of proportionality β_x, $\beta_{r'}$, and β_o respectively. The orientation of vector F_z is computed as the angle between the location of the entity z and the robot r_k.

$$|F_z| = \frac{\beta_z}{d(r_j, x)^2}. \tag{3}$$

Eq. 4 describes the resultant force vector F_T. It sums the attraction forces generated by all particles and subtracts the repulsion forces due to to obstacles and other robots.

$$F_T = \sum_{x \in \mathcal{X}} F_x - \sum_{r_k \in \mathcal{R} - \{r_j\}} F_{r_k} - \sum_{o \in Obs} F_o. \tag{4}$$

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 5,

$$\omega \ = \ K_1 \, \phi + K_2 \, \dot{\phi}, \tag{5}$$

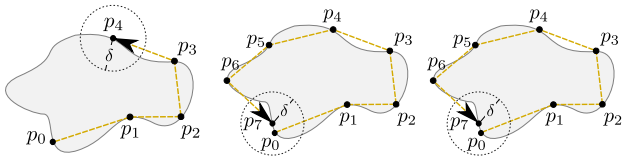
where K_1, and K_2 are the *PD* constants.

B. Tracking an anomaly

When a robot r detects an anomaly $c_r > \tau$, it enters in its tracking mode to enclose the anomaly. The control of the linear speed is based on the distance to other robots in order to avoid collisions, and the angular speed is computed based on the gradient of the anomaly concentration, ∇c_r in the area (Eq. 6).

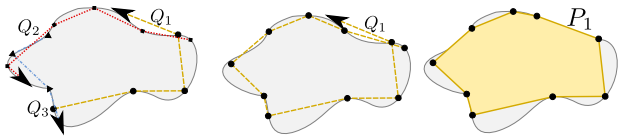
$$\begin{bmatrix} v_t \\ \omega_t \end{bmatrix} = \begin{bmatrix} v_{track} - \frac{K_{track}}{d(r,r')} \\ K_3(\nabla c_r - \tau) + K_4 \, \nabla \dot{c}_r \end{bmatrix}, \tag{6}$$

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* [9], [10], which can be emulated with the model of Eq. 6 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.



(a) Open polyline (b) Closed polyline (c) Resultant polygon

Fig. 1: Processing robot path to identify a polygon.



(a) Three polylines (b) Fused polyline (c) Resultant polygon

Fig. 2: Fusioning polylines Q_1, Q_2, and Q_3.

C. Estimating the enclosing polygons

Our method tries to approximate the dynamic anomaly $\mathcal{A}_i^t$ by a polygon P_i^t. When an anomaly is detected by a robot r, ($c > \tau$), it enters in the tracking mode and creates a polyline Q. For each new sample c, the polyline will add the location of the sensed value to the polyline Q^{t-1},

$$Q^t = Q^{t-1} \cup \{loc(c)\}. \tag{7}$$

At each time step, we check if the polyline closes, that is, if a polygon has been formed formed by that polyline. The verification if Q^t generates a polygon is accomplished by comparing the distance of the last sampled point p_n and all the other lines. The resulting sequence to generate a polygon is restricted to be ordered in counter-clockwise manner. If the last point p_l is close ($< \delta$) to line (p_j, p_{j+1}), Q^t is updated to $Q^t = \{p_{j+1}, p_{j+2} \dots p_l\}$. For instance, Fig. 1a shows an open polygon (the arrowhead represents the robot pose, and points are samples where the anomaly has been detected), where the point p_4 is very far from the other edges. The robot proceeds on tracking and sampling until the polyline closes. Fig. 1b depicts the case where the first point p_0 is close to the edge (p_0, p_1), then, point p_0 is deleted and the edge (p_7, p_1) is added in order to close the polygon.

The parameter δ should be defined carefully because a can close the polygon too early and consequently losing precision. In contrast, a short distance could not identify the polygon when the anomaly is shrinking or expanding.

IV. Multiple Anomaly Support

Most of the state of the art techniques are focused on tracking a single anomaly. The existence of multiple regions of interest involve coordination issues, task allocation and complementary sensor fusion. For instance, allocating many robots to enclose a perimeter is a problem for most of the approaches, because this creates redundancy in the sensed area and reduces the perception for new anomalies.

A. Polygon estimation

Each robot r continuously communicates the value of a sensed point or a polygon, if the last was obtained. If the robot has a polygon, then it broadcasts the simplified polygon data and the time of its detection. This shared information is fused with local information as follows:

1) Fusing polylines: Robot r locally can estimate the polylines of the other robots by receiving the punctual samples and applying Equation 7 for each one independently. Those polylines can be merged to estimate a polygon. To merge the polyline Q_r and the polyline of other robot $Q_{r'}$, we check if one of the edges in Q_r intersects with one of $Q_{r'}$ edges by cross-validation. If an edge $(p_i, p_{i+1}) \in Q_r$ intersects with an edge $(p'_{j+1}, p'_j) \in Q_{r'}$, at the intersecting point p_{int}, then the merge polyline, which is computed by

$$Q_r = \{p'_0, p'_1, \dots, p'_j\} \cup \{p_{int}\} \cup \{p_{i+1}, \dots, p_l\}. \tag{8}$$

The new merged polyline is replaced by the current polyline only if the length of the merged polyline is larger than the current one. Then the new polyline is checked to identify if it closes. This process is repeated for each polyline. Figure 2a illustrates an example with three polylines from three different robots, which are merged by applying Equation 8 twice to get a larger polyline Q_1 (Fig. 1b), which finally obtains polygon P_1 of Figure 2c.

2) Using shared polygons: Robot r compares its samples with the measurements sent by the other robots in order to take advantage of a previously detected polygon. Polygons shared in the network have two characteristics: (1) They are simplified to drastically reduce the number of points to be shared by applying the Ramer-Douglas-Peucker algorithm [21], which deletes redundant points based on a pre-defined distance threshold, and (2) each polygon includes a tag *first-detection-time* so it may have priority in robot allocation.

The metric to determine the required number of robots η_i to track an anomaly $\mathcal{A}_i$ is defined by

$$\eta_i = \lceil \gamma \sum_{i=1}^{k} d(p_i, p_{i-1}) \rceil, \tag{9}$$

where γ is a constant that determines the number of robots required per unit of length. Therefore, we tag the polygon as *available-polygon* if the number of robots tracking the anomaly is less than η_i, and *full-polygon* in otherwise.

If the distance from an *available-polygon* (shared by other robot) is less than a threshold, then we merge the point to polygon, and use the previously shown approach. Let $P_{r'}$ be the polygon received by robot r', and (p'_j, p'_{j+1}) be the edge nearer to the robot r. The resulting polygon is computed as

$$P = \{p'_{j+1}, ..., p'_{l'}\} \cup \{p'_0, ..., p'_j\} \cup \{p_0\}. \tag{10}$$

B. Multi-Robot coordination

We use the particles to represent and to update the world model. In order to determine if a polygon is open or closed for a robot, we use Equation 9 and attract η_i robots by increasing the particles according to Equation 2. However, if the anomaly increases in size, then η_i value will be augmented, and the tag will change to *available-polygon* to attract other robots. The case when the anomaly reduces its size is more complex. For a shrinking anomaly Eq. 9 gives a smaller number which will cause some of the robots to

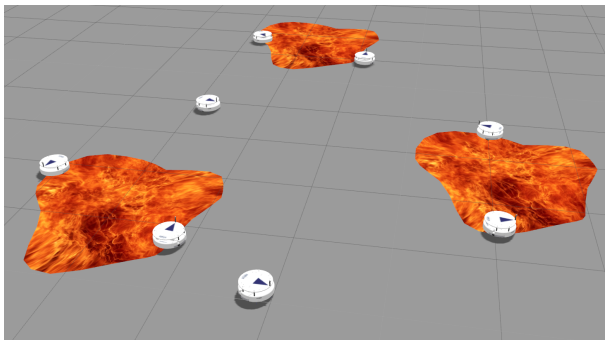
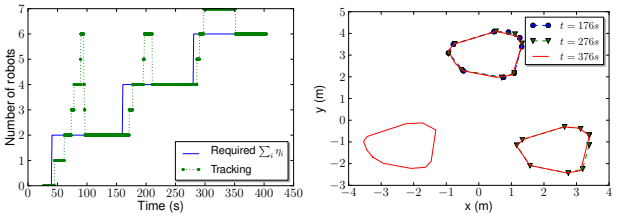


Fig. 3: Simulation with six robots tracking an anomaly and two robots exploring the environment.



(a) Number of robots in the tracking task

(b) Detected polygons

Fig. 4: Allocation of robots to multiple static anomalies

stop tracking and start exploration. Choosing which robots will stop and which will continue to track the anomaly is determined by comparing the *first-detection-time* of the polygon, when recent robots left, and they put tag as *full-polygon* and spread the particles randomly across the map. To avoid robots to be attracted by *full-polygons*, Equation 2 is extended with the clause $w_t^{[i]} = 0$ if the particle $x_t^{[i]}$ is within a *full-polygon*.

V. Experiments and Results

Three validation scenarios have been used with three types of anomalies: Static, expanding and shrinking. We analyze the accuracy of estimation of the detected area and of robot allocation along the time for each scenario. The area of the anomaly is denoted by $area(P_1^t)$ and the estimation by $area(\widetilde{P}_1^t)$. As the approximate polygon also covers areas without anomaly, $area(\widetilde{P}_i^t - P_i^t)$, and uncovered areas, $area(P_i^t - \widetilde{P}_i^t)$, we have defined the estimation error ϵ_t as denoted in Equation 11.

$$\epsilon_t = area(P_i^t - \widetilde{P}_i^t) + area(\widetilde{P}_i^t - P_i^t). \tag{11}$$

We initially conducted simulated experiments on the ROS-Gazebo 3D simulator, and then with real robots. Real and virtual robots are assumed to be the *iCreate* base, differential mobile platform equipped with: Onboard-computer, communication module IEEE 802.3.11 (WIFI), and an RGB camera. Figure 3 shows the Gazebo environment with eight robots, and those that are tracking the anomaly (represented by a rug with a texture of fire). Our software for real and virtual robots is open source and is available[1].

A. Simulations

An important factor that needs to be taken into account is the number of points that conforms to the detected polygon and that will be broadcasted. The error ϵ_t increases linearly with the applied threshold in the *Ramer-Douglas-Peucker* algorithm, however, the number of required points decreases exponentially. We used a non-convex static anomaly with $area = 3.49m^2$ and $perimeter = 7.15m$, which was sampled with 186 points by a single robot. Using a $threshold = 0.1m$, the number of points are drastically reduced to 10 points (e.g. 95% of the original number of points) with error $\bar{\epsilon}_t \approx 0.23m^2$.

The simulated environment is defined by a map with dimensions of 5×4 meters. In the initial setup, eight robots are distributed over the map. We analyze the following three scenarios.

1) Static anomalies: In this scenario, three static anomalies turn up in different places at different times $[40s, 160s, 280s]$. Fig. 4a illustrates the robot allocation for tracking. When the first anomaly appears, the number of robots increases progressively until the polygon is closed, as described in section IV, where the required number of robots cover the anomalies is computed as $\sum_i \eta_i$ (Eq. 9), and the remaining robots proceed to continue the exploration. This can be seen in the figure as the three initial peaks in the tracking robots, which are reduced to one after the anomaly is detected. Since the anomalies are static, the error ϵ_t is proportional to the total area generated by the anomalies. Figure 4b shows the shared polygons at different times.

2) Expanding: Two anomalies show up at different locations of the map. They start to expand in different directions and with different velocities. Fig. 5a shows that while an anomaly is growing, the area is sub-estimated and the error ϵ_t is proportional to the velocity of expansion. Fig. 5b illustrates the heterogeneous expansion along the time.

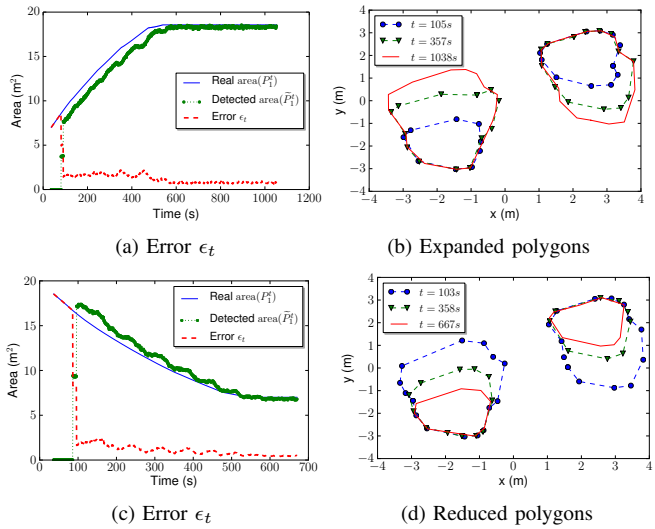
3) Shrinking: In this scenario, the same two anomalies shrink until they stabilize. As the anomaly is tracked in counter-clockwise manner, the controller generates movements to the right-side but the anomaly is moving to the opposite direction. This continuous rectification in orientation generates an oscillatory movement that is reflected in the error ϵ_t, as it can be seen in Figure 5c. Similarly to the expanding scenario, the ϵ_t is affected by the velocity of the anomaly behavior and by the rectification of the tracking controller.

B. Experiments with real robots

Our testbed is composed by a map of 3×1.8 meters and three robots (Figure 6). Localization is acquired by processing images of an overhead camera with a wide-angle lenses. Dynamic anomalies are computer generated and projected on the floor.

On the one hand, reduced spaces limit the deployment and movement of a larger number of robots in the environment. On the other hand, the time for finding all anomalies and

[1]The source code is available at `https://github.com/dsaldana/roomba_sensor_network`



(a) Error ϵ_t (b) Expanded polygons

(c) Error ϵ_t (d) Reduced polygons

Fig. 5: Two dynamic anomalies reducing their area

Fig. 6: Actual robots tracking anomalies.

generating enclosing polygons is reduced. On these experiments, we found that the detecting process speeds up in comparison with simulations because of the small space and size of the anomalies, but the precision of our sensor suit limits the quality of the estimated polygons.

VI. Conclusions and Future Work

In this paper we proposed a probabilistic distributed coordination method to detect and track multiple dynamic anomalies using a robotic sensor network. Two problems were tackled, namely, full map coverage to find anomalies and multi-robot coordination to track multiple dynamic anomalies. We have experimentally shown, both in simulation and with real robots, that our approach can simultaneously detect regions out of normal conditions.

Experimental results demonstrated that the proposed method works in different scenarios, where autonomous robot allocation is effectively executed based on the regions which require more attention.

An important issue in scalability is the message broadcasting. We have shown experimentally using eight robots, but for communication may become burdensome for larger number of robots.

As a future research, the estimation of the anomaly can be improved by merging detected polygons based on the sampled time, since recent measurements give better estimations. Another aspect concerns complex shapes, for example, regions with interior holes, which we understand it might be tackled with the use of different types or robots (e.g. aerial and ground robots).

References

[1] D. Bruemmer, "A robotic swarm for spill finding and perimeter formation," *Spectrum: International Conference on Nuclear and Hazardous Waste Management*, 2002.

[2] R. Mottaghi and R. Vaughan, "An integrated particle filter and potential field method for cooperative robot target tracking," in *IEEE International Conference on Robotics and Automation. ICRA 2006.*

[3] J. Tisdale, A. Ryan, Z. Kim, D. Tornqvist, and J. K. Hedrick, "A multiple uav system for vision-based search and localization," in *American Control Conference, 2008*. IEEE, 2008, pp. 1985–1990.

[4] D. Marthaler and A. Bertozzi, "Collective motion algorithms for determining environmental boundaries," in *SIAM Conference on Applications of Dynamical Systems*, 2003.

[5] S. Srinivasan, K. Ramamritham, and P. Kulkarni, "ACE in the Hole: Adaptive Contour Estimation Using Collaborating Mobile Sensors," *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pp. 147–158, Apr. 2008.

[6] J. Cortés, "Cooperative detection of areas of rapid change in spatial fields," *Automatica*, vol. 48, no. 4, pp. 673 – 681, 2012.

[7] R. Bachmayer and N. Leonard, "Vehicle networks for gradient descent in a sampled environment," in *Procedings of 41s IEEE Conference on Decision and Control*, no. December, 2002.

[8] E. Fiorelli and N. E. Leonard, "Exploring scalar fields using multiple sensor platforms: Tracking level curves," *2007 46th IEEE Conference on Decision and Control*, pp. 3579–3584, 2007.

[9] A. L. Bertozzi, "Environmental boundary tracking and estimation using multiple autonomous vehicles," *2007 46th IEEE Conference on Decision and Control*, pp. 4918–4923, 2007.

[10] A. Joshi, T. Ashley, Y. R. Huang, and A. L. Bertozzi, "Experimental Validation of Cooperative Environmental Boundary Tracking with Onboard Sensors," *Control*, pp. 2630–2635, 2009.

[11] M. Kemp, A. Bertozzi, and D. Marthaler, "Multi-UUV perimeter surveillance," in *IEEE OES Workshop on Multiple AUV Operations*, 2004.

[12] S. Susca, F. Bullo, and S. Martinez, "Gradient algorithms for polygonal approximation of convex contours," *Automatica*, vol. 45, no. 2, pp. 510–516, Feb. 2009.

[13] S. Li, Y. Guo, and B. Bingham, "Multi-robot cooperative control for monitoring and tracking dynamic plumes," in *IEEE International Conference on Robotics and Automation. Proceedings. ICRA 2014.* IEEE, 2014.

[14] J. Euler, A. Horn, D. Haumann, J. Adamy, and O. Stryk, "Cooperative n-boundary tracking in large scale environments," in *Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 9th International Conference on*. IEEE, 2012, pp. 1–6.

[15] J. Clark and R. Fierro, "Mobile robotic sensors for perimeter detection and tracking." *ISA transactions*, vol. 46, no. 1, pp. 3–13, Feb. 2007.

[16] D. Saldaña, L. Chaimowicz, and M. M. Campos, "Searching and tracking anomalies with multiple robots: A probabilistic approach," in *LARS'2014 - 11th Latin American Robotics Symposium*, 2014.

[17] D. Marthaler and A. Bertozzi, "Tracking environmental level sets with autonomous vehicles," *Journal of the Electrochemical Society*, vol. 129, p. 2865, 2003.

[18] D. Casbeer, D. Kingston, R. Beard, and T. McLain, "Cooperative forest fire surveillance using a team of small unmanned air vehicles," *International Journal of Systems Science*, vol. 37, no. 6, pp. 351–360, 2006.

[19] J. D. Hol, T. B. Schon, and F. Gustafsson, "On resampling algorithms for particle filters," in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*. IEEE, 2006, pp. 79–82.

[20] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1991, pp. 1398–1404.

[21] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.