Impressionist Hole Detection and Healing Using Swarms of Agents with Quantized Perception

Giada Simionato
Dept. of Information Engineering
University of Florence, University of Pisa
Florence, Pisa, Italy
giada.simionato@phd.unipi.it

Marco Parola
Dept. of Information Engineering
University of Pisa
Pisa, Italy
marco.parola@ing.unipi.it

Mario G.C.A. Cimino
Dept. of Information Engineering
University of Pisa
Pisa, Italy
mario.cimino@ing.unipi.it

***Abstract*—Coverage holes are a key problem in wireless sensor networks. Methods that use relative localization techniques to restore the service, or heal the holes, rely on accurate range and bearing measurements. However, high-precision range and bearing sensors are too heavy, expensive, and range-limited for the agents tasked with healing. To overcome these limitations, we propose a novel impressionist algorithm, inspired by a recent swarm-based approach, that works with extremely coarse range and bearing information and at low perception frequency, to detect and heal the holes. In the proposed approach, a swarm of agents uses quantized information to navigate a potential field, generated by network nodes, to reach the nearest hole. The swarm adopts a greedy deployment behavior, preventing concurrent placement in close-by locations. After deployment, agents use their coarse perception to update the potential field, leading the rest of the swarm to unhealed area. Simulation results demonstrate that our algorithm achieves similar or better coverage compared to the state-of-the-art and to a benchmark based on random walk. This is achieved using just three bearing quantization levels and four times lower perception frequency. Overall, our impressionist approach shows faster healing, albeit at the expense of employing slightly more agents.**

***Index Terms*—swarm robotics, limited perception, hole healing**

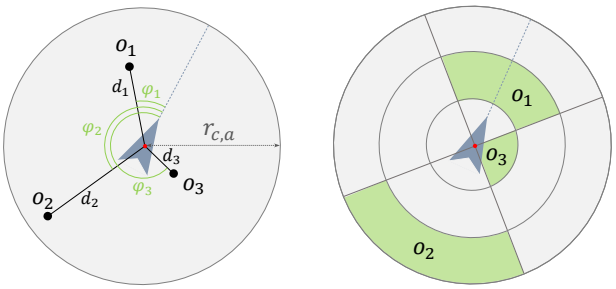
I. Introduction

In Wireless Sensor Networks (WSNs), restoring the coverage of a Region of Interest (ROI) in case of failure of one or multiple nodes is a key problem for a wide range of applications [1]–[4]. This is especially critical in case of time-sensitive, emergency situations, in which holes in the coverage must be detected and covered (or *healed*) quickly and with minimal deployment cost [5]. Most of the existing solutions to the hole detection and healing problem rely on the relocation of a subset of the network nodes to the area left uncovered [6]–[8]. However, this requires assuming that the nodes are *mobile* and *densely deployed*, which is unrealistic and impractical in case nodes include expensive or complex hardware [9]–[11].

An recent alternative solution in [12] shows that a swarm of resource-constrained agents can lead to a quick recovery of the coverage, without a priori knowledge of the environment. The agents can perceive the environment by the only means of a Range and Bearing (RaB) sensor, that provides relative distance and angle to targets within a certain range. However, RaB devices providing precise measurements are too heavy and expensive to be mounted on small agents (e.g., drones) [13]. The technology behind *portable* RaB sensors is still in its early stages: Current prototypes can not provide reliable measurements in challenging scenarios, especially at long ranges [14], [15]. Despite the algorithm in [12] demonstrated good robustness to noisy perception, it ultimately needs reliable measurements. This limitation is shared by all available solutions that use RaB sensors, or some form of relative localization, to orient in the environment relying on the relative position of surrounding nodes or agents.

In this work, we address such perception problem by proposing a novel swarm intelligence algorithm, inspired by the work in [12], to detect and heal holes in the coverage using extremely coarse RaB information. Our approach belongs to the class of *impressionist* algorithms, which work with *quantized* information obtained at low perception frequency [16]. By adopting the impressionist paradigm, our aim is to lessen the perception and computation burdens on the agents, while providing extra robustness. Instead of having access to precise RaB measurements as in [12] (Fig. 1a), our agents only have a rough indication about the distance interval and the circular sector where other nodes and agents are located (Fig. 1b). The swarm uses the quantized information to navigate a potential field generated by the nodes, following the logic introduced in [12], until it reaches the closest hole. The agents rely on a greedy deployment behavior, in which they place themselves in the first uncovered location found. A commitment strategy is adopted to prevent agents to concurrently deploy in close-by locations. Upon deployment, the agents use their coarse perception to update the potential field, driving the rest of the swarm to the part of the hole not already healed.

We compared the performance of our algorithm against: (i) [12], which has been shown to outperform state-of-the-art solutions, and (ii) a benchmark based on random walk, a behavior widely adopted as a baseline for swarm navigation [17], [18]. Moreover, we analyzed the impact of the quantization factor and the perception frequency, to evaluate the robustness of our solution. Experimental results indicate that it reaches comparable or higher coverage with respect to (i) and (ii), while using only three quantization levels for the bearing information, five levels for the distance, and one fourth of



(a) (b)

Fig. 1: RaB information provided to the agent in traditional relative positioning (a) and in our impressionist algorithm (b). Here, targets are perceived as being *"on the right"*, *"on the left"*, *"in front"* or *"behind"*, and *"close"*, *"in the middle"* or *"far"*.

the perception frequency. Notably, our approach demonstrates significantly faster healing, at the cost of a marginal increase in the number of agents employed. This makes it highly suitable for time-sensitive scenarios.

II. Related work

Hole Detection and Healing: Several studies addressed the problem of hole detection and healing in WSNs [1]–[6]. Numerous works rely on Voronoi diagram or Delauney triangulation as their main component in the recovery of coverage holes [3], [6], [7], requiring, however, dense networks [3], or centralized computations [6], [7]. Alternatively, geometry-based approaches are proposed in [1], [2], [8], leveraging information about intersection points among neighbors to select, according to geometric criteria, which and where nodes should move to restore the coverage. These solutions give priority to minimize the energy consumption of the nodes over maximizing the coverage: a secondary objective in temporary solutions [1], [8]. A different perspective is presented in [19]–[21], in which the problem is modeled as an optimization task, where enhanced versions of particle swarm optimization [19], [20], or the artificial fish swarm algorithm [21] are used to compute the position of patching nodes that maximizes the coverage. However, they necessitate global knowledge of the network, that is an unrealistic requirement for large WSNs or in highly dynamic environments. To account for this problem, methods in [9]–[11] rely only on local information, combining the advantages of geometry-based solutions with those using virtual forces, such as [22]. These *hybrid* approaches need GPS localization, which may be unavailable in mission-critical scenario. The approach presented in [12], to which our algorithm is inspired, uses RaB localization and local knowledge to control the swarm. The swarm is activated by the exposure to a potential field, locally generated by nodes upon detection of a hole, and pointing to its border. The agents follow this field using their RaB information until they reach the hole and then deploy in locally optimal positions computed using geometric criteria. Upon deployment, the agents update the potential field to reflect the changes in the hole border, thus recruiting other agents, until coverage is restored.

Sparse and Imperfect Perception: Robot swarms usually rely on external technologies, such as global positioning systems or base stations, to gather accurate data and process it with high frequency [23]–[25]. However, these technologies are not always available, especially in emergency scenarios. There exists a limited literature on the design of swarm-based algorithms that rely on low-frequency, quantized information, e.g., to search for heat sources [26], reach consensus on environmental features [27], or trail following [28], but none to address hole detection and healing in WSNs. Designing algorithms resilient to sparse, discretized updates is not only important, but can be also beneficial for swarm performance [28], [29]. To this end, [16] introduced the class of impressionist algorithms, capable of working with minimal information where traditional algorithms are prone to fail. The authors validated their findings bringing alignment, dispersion and milling behaviors as case study.

III. Methodology

In this work, we consider a 2D obstacle-free ROI in which a network of static nodes is randomly deployed. The failure of some of these nodes generates holes in the coverage. To heal the holes, we employ a swarm of mobile agents embodying robots. Both nodes and agents are treated as point-like objects with heading. Their position corresponds to that of the center of mass of the device they represent. They both have communication and sensing (i.e., coverage) capabilities, modeled as Boolean disks: We assume reliable functioning within a disk of radius r_c or r_s, respectively, and no service outside [30]. As in [12], we account for different sensing ranges, with that of the nodes being much greater than that of the agents, i.e., $r_{s,n} \gg r_{s,a}$. Moreover, we impose the communication radius to be at least twice the longest sensing radius to guarantee connectivity [31].

Agents' movement is governed by non-holonomic dynamics, with cruise driving $\bar{v}$ and steering $\bar{\omega}$ speed as control inputs. Agents, marked with unique IDs, select their behavior based on coarse RaB perceptions, provided by the sensor system every T_p seconds. The perception area, set equal to the communication disk, is divided into K adjacent circular sectors, with $2\pi/K$ central angle, and L concentric circular crowns, representing distance intervals. The circular sectors align with the agent's heading through the bisector of the first sector. For all nodes and agents within $r_{c,a}$, the quantized range and bearing information indicates the specific circular sector and distance interval in which they are located. Fig. 1b shows an example of perception with $K = 4$ and $L = 3$. This could be implemented with a minimal number of adjacent, low-precision *range-only* sensors, obtaining a coarse bearing information by knowing which sensor detected the target.

Since RaB sensors limitations mainly affect small agents, we used the same perception setting of [12] for the nodes. That is, agent o_i is provided with the relative distance d_{ij} and angle φ_{ij} to each node and agent o_j within its communication range,

as shown in Fig. 1a. The proposed impressionist algorithm comprises two stages: *hole detection* and *hole healing*.

A. Hole Detection

Hole formation is detected by network nodes using the logic in [12]. They use their perception to periodically compute the *Angular Coverage Ratio* (ACR) $c \in [0, 1]$, a coefficient that measures how much the node is surrounded by neighboring nodes, or later, deployed agents. To do so, a node o_i filters its perception, including only nodes and deployed agents o_j whose sensing disk overlaps with its own ($o_j \in \mathcal{N}_i$). For each neighbor, it computes θ_j, the semi-central angle of the circular sector delimited by the two intersection points between the borders of their sensing disks, as in Eq. (1a), where r_{s,o_i} is the sensing radius of o_i. Then, it obtains angular intervals by centering θ_j in the corresponding bearing measurement, and it merges them together following Eq. (1b). If c_i, obtained using Eq. (1c), is lower than a threshold Th_c, the node is not completely surrounded, and is on the border of a hole ($o_i \in \mathcal{B}$)

$$\theta_j = \arccos\left(\frac{r_{s,o_i}^2 + d_{ij}^2 - r_{s,o_j}^2}{2 r_{s,o_i} d_{ij}}\right), \tag{1a}$$

$$\Theta_i = \bigcup_{o_j \in \mathcal{N}_i} [\varphi_{ij} - \theta_j, \varphi_{ij} + \theta_j], \tag{1b}$$

$$c_i = |\Theta_i| / 2\pi. \tag{1c}$$

Each node o_i uses the ACR value to maintain its one-hop distance to the closest hole, the *level* ℓ_i, computed as follows

$$\ell_i = \begin{cases} 0, & \text{if } o_i \in \mathcal{B} \\ \min\limits_{o_j \in \mathcal{N}_i} \ell_j + 1, & \text{otherwise.} \end{cases} \tag{2}$$

This information generates a potential field of increasing integer values pointing toward holes border (having minimum level), that can be perceived and followed by the swarm.

B. Hole Healing

As in [12], the agents are initially resting in randomly scattered Release Points (RPs) in *inactive* state. They passively listen communications among nearby nodes, waiting to hear about the formation of a hole to activate. Upon activation, the agents use their quantized information to drive toward the closest hole, following the potential field created by nodes levels as in Eq. (2). In [12], the agents move toward the direction of the closest node with minimum level among those perceived. With the nodes at the border having absolute minimum level (i.e., 0), the agents will eventually reach a hole. However, with quantized information, more than one node with minimum level can be perceived in the same distance interval, therefore indistinguishable from a distance perspective. Moreover, these closest nodes may belong to different circular sectors, introducing ambiguity in the direction. Selecting the bearing sector that requires less rotation for the agent reduces this ambiguity. However, with this type of navigation logic, the agent might approach the hole border tangentially to the zero-level curve, ending up trapped by a zero-level node and remaining outside the hole (Fig. 2 - Closest Min. Follower). For this reason, as navigation logic, we propose to average the levels perceived by the agent in each sector, imposing forward motion as soon as the agent faces the sector with minimum average and selecting the one that requires less rotation as tie-breaking rule. In this way, the agent is aware of the shape of the level curves, being able to cross the zero-level one, and remaining inside the hole (Fig. 2 - Min. Avg. Sector-wise Follower). We stress that no distance measurements are involved in this navigation process.

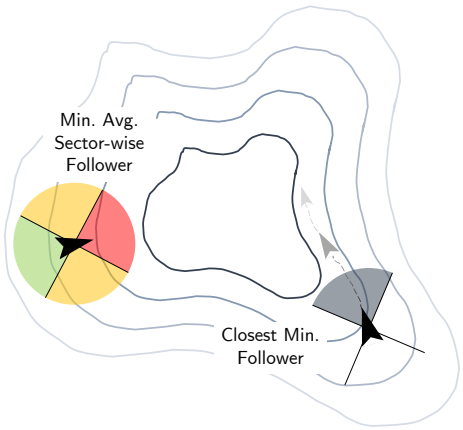


Fig. 2: Graphical representation of the difference between the navigation logic in [12] with quantized information (Closest Min. Follower) and the impressionist logic (Min. Avg. Sector-wise Follower) with $K = 4$.

While navigating, if an agent perceives at least a node, or deployed agent, of level 0 (on the border), it starts evaluating its current position for deployment. If the position of agent a_i does not lie within the sensing disk of any perceived nodes or deployed agent o_j, i.e., $d_{ij} > r_{s,o_j} \forall o_j \in \mathcal{P}_i$, the position is *free* and the agent can deploy there. To evaluate the freedom of a position, the agent needs only tertiary distance information: if a measurement is less than $r_{s,a}$, to check if it is within the sensing disk of an agent, or $r_{s,n}$ for a node. The distance is therefore quantized into $L = 3$ intervals.

Before deploying, an agent that has found a free position, changes its state to *committed*. This behavior was introduced to avoid concurrent placement in the same area, thus wasting resources. A committed agent collects the IDs of all committed agents within its sensing radius. If it has the lowest ID [32], it deploys in the network, changing its state to *deployed*. Otherwise, it switches back to the active state and continues to follow the potential field. This behavior does not require additional quantization of the distance.

Since the first free position is selected for deployment, the healing starts from the border of the hole, changing its shape. Upon deployment, the agents start acting as nodes, calculating their ACR and their level, to attract other agents. The ACR, as computed in Eq. (1), strongly relies on good RaB information. We propose four solutions, of increasing complexity, to compute the ACR using quantized information:

1) ACR-1: All deployed agents believe to be on the border of a hole, taking level zero. This solution does not require any computation nor information from the environment.

2) ACR-2: This heuristic groups perceived nodes and deployed agents by sector. If a sector has a node, it gets weight $w_n \in [0, 1]$, while, if it has an agent, it gets weight $w_a \in [0, 1]$. If the sum of weights is less than K/π, the agent may be at the border of a hole. The level is calculated using Eq. (2), except the minimum level is determined among all nodes and deployed agents perceived, not just adjacent ones, to prevent extra quantization of distance measurements.

3) ACR-3: It uses quantized RaB information from neighbors. Adjacency is established based on whether the distance is within twice the agents' sensing radius for agent-agent adjacency or within $r_{s,a} + r_{s,n}$ for agent-node adjacency. When combined with position check intervals, it brings the distance quantization levels to five. We assign each distance measurement as the mean point of its corresponding interval, and use it to compute the discretized $\tilde{\theta}_j = 2\theta_j$ as in Eq. (1a). These quantities are summed sector-wise to check if the sector is sufficiently covered. Summing angle intervals corresponds to considering the best case, that is no overlapping. For a sector k_i with $i = 1, \ldots, K$, and o_j a neighbor, if $\sum_{o_j \in k_i} \tilde{\theta}_j \geq \frac{2\pi}{K} Th_c$ the agent considers that sector covered. Otherwise, the agent checks whether the two adjacent sectors may offer additional coverage. To do so, it checks whether $\sum_{o_j \in \{k_{i-1}, k_i, k_{i+1}\}} \tilde{\theta}_j \geq \frac{6\pi}{K} Th_c$. If it is the case, the sector is considered covered, otherwise is not. An agent is on the border of a hole if at least one sector is not covered. The level is computed as in Eq. (2).

4) ACR-4: It requires the sensor system to be placed on a base, rotating at a fixed speed ω_s for the sector central angle. During rotation, it collects a number of samples of distance measurements. By comparing the samples, it is possible to obtain a bearing resolution f_{ov}, much lower than the sector angle. For each perceived neighbor, the agents consider the bisector of their new sectors of width f_{ov} as bearing, and perform the same distance quantization of ACR-3. The quantized ACR and level are then computed according to Eq. (1) and Eq. (2). More accurate bearing measurements come at the cost of delays in the computation of ACR (time required to perform rotation) and energy consumption due to sensors movement.

Fig. 3 showcases the complexity of computing the ACR with quantized information: Opposite situations may yield equal perception. Quantized ACRs approximate those obtained with exact measurements, occasionally leading to wrong levels.

C. Frontier and Loop Avoidance

The impressionist navigation logic may drive the agents outside the ROI if the RPs are too close to its frontier, since a sector without readings has a lower average than a sector with readings. To prevent this, the agents are instructed to move away from the nodes at frontier by keeping them in the rear sectors. A node communicates that it is at the frontier when its ACR is lower than Th_c right after network displacement (i.e., prior to holes formation).

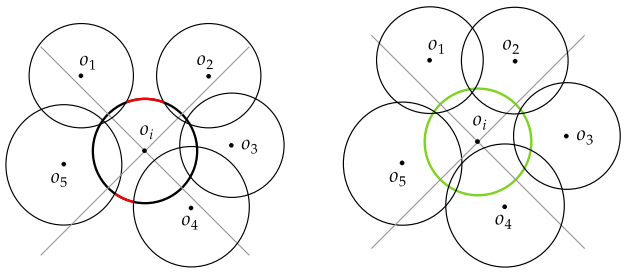


Fig. 3: Graphical example of two opposite situations generating equal perception, with $K = 4$ and the agent o_i being (a) and not being (b) on the border of holes.

The impressionist navigation logic may generate loops in the agents' behavior. We call *short-term* loops those caused by two consecutive perceptions demanding opposite actions, thus stalling the agents. To break these loops, the agents are subjected to a random impulse that allows them to escape from such perceptions. The impulse is realized imposing uniformly random control speeds in $[-\bar{v}, \bar{v}]$ and $[-\bar{\omega}, \bar{\omega}]$.

The agents can navigate along orbits of arbitrary length, created by a sequence of perceptions resulting in locations that yield identical perceptions. We call these *long-term* loops. To avoid them, we programmed a wiggling behavior that consists of initially n_w^i consecutive instants in which the agents are subjected to random impulses (as in the short-time case). Since these loops are usually observed when the hole is partially healed, the wiggle is triggered only after t_w^i instants and repeated with a period of T_w instants. At each repetition of the wiggle, its duration is increased by one instant.

IV. Experiments and Results

To evaluate the performance of our algorithm, we used HDHSim [33], a discrete-time simulator introduced in [12], explicitly designed for hole detection and healing problems. Our analysis served two purposes: (i) comparing the performance of the impressionist algorithm against the one in [12] (denoted as *Exact*), that accesses the full set of information and (ii) studying its impressionistic traits, that is how its performance varies with different spatial and temporal resolutions. As in [16], the former is represented by the number K of quantization levels of the bearing, while the latter is measured through the *Cognition Speed* (CS), that is defined as the ratio of perception frequency to motion speed, i.e., $(T_P \bar{v})^{-1}$. To further validate our approach, we also compared it against a benchmark based on random walk behavior (see Section IV-A).

We adopted the same default configuration of [12]: a failure of 7 adjacent nodes out of 125 randomly placed in the ROI. The swarm is composed of 50 agents released from a single RP. We carried out 100 simulations of $T = 1000$ time steps (of 1s each), for each experiment. The simulation parameters are reported in Tab. I.

TABLE I: Parameters used in the simulations. The default values are highlighted in bold font.

Parameter	Symbol	Value
Node sensing range	$r_{s,n}$	30 m
Node comm. range	$r_{c,n}$	60 m
Agent sensing range	$r_{s,a}$	{5, **10**, 15} m
Agent comm. range	$r_{c,a}$	60 m
Cruise driving speed	$\bar{v}$	5 m/s
Cruise steering speed	$\bar{\omega}$	0.1 rad/s
Sensors rotation speed	ω_s	0.1 rad/s
No. bearing sectors	K	{2, 3, **4**, 5, 6}
Cognitive speed	CS	{0.4, **0.2**, 0.1, 0.05} m^{-1}
Coverage threshold	Th_c	0.95

We measured performance by the average coverage over time, defined as the fraction of hole area covered by deployed agents, and by the average number of deployed agents. We will refer as *convergence*, the attainment of a stable value by the coverage trend before the end of the simulation. To ensure a fair comparison in terms of number of deployed agents, we endowed the Exact method with the same commitment behavior of our algorithm, without altering the measure of efficacy of its navigation and deployment logic.

A. Random Walk Benchmark

To benchmark our algorithm, we designed an approach in which random walk is the primary healing strategy. Since a pure random walk over the whole ROI would not be particularly informative, we limited its application in the vicinity of the hole. To do so, we leveraged the impressionist navigation strategy every time the agent is not perceiving any level zero.

The rotating motion of the random walk is obtained sampling the turning angle from a wrapped Cauchy distribution with zero mean and concentration ρ, using the inversion sampling method in [34]. For the straight motion, we sampled the number of instants, instead of the distance to cover as in [35], from a log-normal distribution with zero-mean and 0.5 standard deviation. The sampled number was then scaled by a factor λ to match the dimension of the ROI.

This *Random Walk* (RW) algorithm uses the same positioning logic, commitment strategy, and ACR computation methods as the impressionist approach. We included the logic in Section III-C to counteract frontier crossing and short-time loops. Due to its random nature, this method is not affected by long-term loops.

As the impressionist algorithm, RW works with K circular sectors and 3 to 5 levels of quantization of distance measurements (depending on the computation strategy of the ACR).

B. Family-wise Analysis

The impressionist and RW algorithms differ in the navigation logic, but share the four ACR computation strategies. For this reason, we identified them as two *families* of four variants each (e.g., impressionist + ACR-i with $i = 1, 2, 3, 4$) and we compared these variants to individuate the most promising combinations.

1) Parameters Optimization: To properly detect which variants were the most effective in hole healing, we firstly performed parameters optimization using the *Tree-structured Parzen Estimator* (TPE) implementation in the Optuna framework [36]. We set a limit of 200 iterations to maximize the value of coverage at convergence. To identify the initial data necessary to run TPE, we applied few steps of random search algorithm, as suggested in [37]. The parameters involved in the optimization, along with their range of variation (in the format [*min, max, step*] or list of values), and the resulting best values are reported in Tab. II for the navigation parameters and in Tab. III for the ACR parameters.

2) Comparison: For each of the eight variants, we carried out 100 simulations on the same set of randomly generated scenarios, with the optimized parameters and the default configuration of Tab. I. Fig. 4a shows the average coverage over time for the four variants of the impressionist family. There is no statistically significant difference among variants with the same number of quantization levels for distance measurements: strategies with $L = 3$ (ACR-1 and ACR-2) result in slightly less coverage at convergence with longer transient. For $L = 5$ variants, ACR-3 reaches convergence sooner than ACR-4, slowed by the delay in the update caused by the sensors rotation (Fig. 4a - zoom). For this reason, we selected the impressionist + ACR-3, *Imp-3* henceforth, for the rest of the evaluation. RW + ACR-1 has longer transient compared to other variants, because it lacks additional guidance from deployed agents, as shown in Fig. 4b. Since there is no statistically significant difference among the other variants, we selected RW + ACR-3 (*RW-3*) as default, to match the impressionist algorithm. We compared the best performing variants of the two families with the Exact algorithm (Fig. 4c). All methods achieve similar high values of coverage at convergence, with Imp-3 being the quickest to converge and Exact the slowest. This speed up is caused by the greater dispersion of the swarm in Imp-3 and RW-3, that allows tackling the hole from different directions, thus parallelizing the healing. To demonstrate this, we computed the Global Shannon Entropy (GSE) as in [38]. We divided the ROI in q cells and, for each time step, we counted the number x_i of agents within the i-th cell for $i = 1, \ldots, q$. The GSE of the empirical distribution is

$$\mathrm{GSE} = -\frac{1}{\log(q)} \sum_{i=1}^{q} p_i \log(p_i), \qquad p_i = \frac{x_i}{\sum_{i=1}^{q} x_i}. \tag{3}$$

The closer GSE is to 1, the smaller is the departure from the complete spatial randomness, hence the more the swarm is dispersed. Conversely, the closer to 0, the more clustered the agents are. The entropy values in Tab. IV show that on average, during the navigation, the Exact swarm is much more compact than the other solutions. The excess in the entropy of RW-3, in comparison to Imp-3, is caused by the additional randomness in the movements.

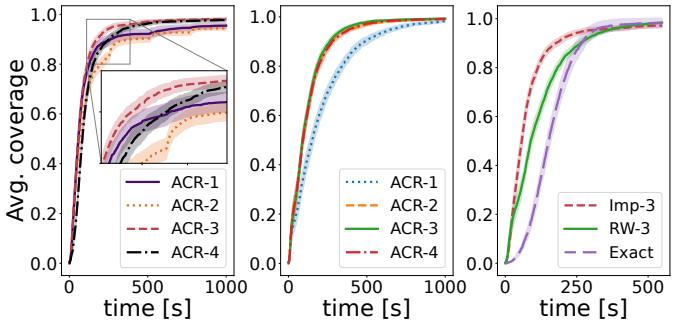
Since the methods achieve similar values of coverage at convergence, we can compare the number of deployed agents (Tab. IV). The Exact method uses less agents, with respect

TABLE II: Optimization space for navigation logic parameters.

Logic	Parameter			Best Value			
	Name	Symbol	Range	ACR-1	ACR-2	ACR-3	ACR-4
Impressionist	Wiggle period	T_w	[50, 300, 50]	250	250	200	150
	Wiggle initial strength	n_w^i	[1, 10, 3]	10	10	4	10
	Wiggle initial instant	t_w^i	[0, 1000, 250]	250	250	0	1000
Random Walk	RW concentration	ρ	[0, 1, 0.2]	0.4	0.8	0.8	0.8
	RW scaling factor	λ	[1, 51, 10]	11	21	21	21

TABLE III: Optimization space for ACR computation parameters.

Logic	Parameter			Best Value	
	Name	Symbol	Range	Impressionist	Random Walk
ACR-2	Agents weight	w_a	[0, 1, 0.2]	0.2	0
	Nodes weight	w_n	[0, 1, 0.2]	0.8	0.4
ACR-4	Resolution	f_{ov}	[0.79, 0.52, 0.39, 0.26, 0.2]	0.2	0.52



(a) Impressionist (b) Random Walk (c) Cf. Exact

Fig. 4: Average coverage over time for the impressionist (a) and RW (b) families. Comparison among best variants for each family and Exact algorithm (c). The shaded bands represent the 95% confidence interval.

to Imp-3 and RW-3, to heal the hole. Their higher number of agents used results from the greater overlap in the agents' sensing disks, caused by their greedy deployment strategy.

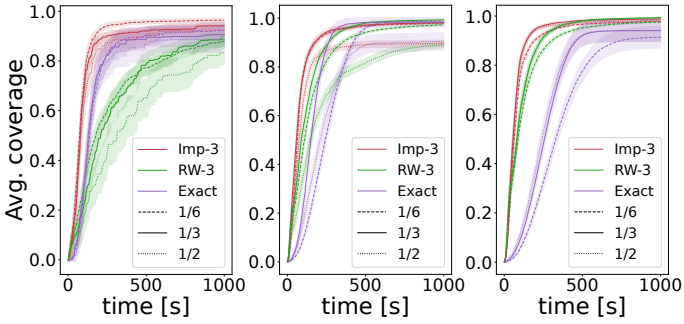
C. Single Hole Analysis

We expanded the comparison by analyzing the impact that different hole size and *sensing ratios* have on the performance. We define the sensing ratio R_s as the ratio between the agents' sensing range and that of the nodes, i.e., $r_{s,a}/r_{s,n}$. Specifically, we varied m, the number of adjacent nodes which failure creates the hole. For each value of $m = \{1, 7, 15\}$, we performed a set of 100 simulations by keeping fixed the nodes' sensing range and varying R_s in $\{1/6, 1/3, 1/2\}$. For each combination of sensing ratio and hole size, we carefully calibrated the swarm size to avoid undue inflation caused by an excessive number of agents, while ensuring that healing was not compromised due to the lack of agents (regardless of the quality of the algorithm).

TABLE IV: Average number of deployed agents and average maximum GSE variation in the default scenario.

	Deployed Agents		Max. GSE Variation	
	Avg.	95% CI	Avg.	95% CI
Imp-3	40.10	0.82	0.59	0.02
RW-3	40.86	0.81	**0.78**	0.02
Exact	**33.61**	0.82	0.25	0.03

The results in Fig. 5 show that an influential factor in the healing process is the relationship between the dimension of the agents' sensing disk and that of the *hole*. The smaller the disk is with respect to the dimension of the hole, the slower the healing process will be. This is caused by the limited contribution that small agents individually bring to the healing of massive holes. The difference in the transient duration is more pronounced when the swarm is clustered and the healing cannot be parallelized further, as for the Exact algorithm (Fig. 5b and Fig. 5c - $R_s = 1/6$). Conversely, the bigger the agents' sensing disk is with respect to the hole, the more difficult is to find a viable placement position, resulting in a decrease of the coverage at convergence, as shown by Fig. 5a and Fig. 5b for $R_s = 1/2$. Bigger agents tend to leave sparse tiny area uncovered, difficult to detect and reach. This drawback is emphasized in the RW: the probability of crossing an uncovered area with random movements decreases with the dimension of the area. This results in longer transients and lower coverage at convergence, as reported in Fig. 5a. For all hole sizes, high values of coverage were achieved, independently from the sensing ratio and especially for Imp-3. This means that the algorithm is suitable to various sensing hardware. By comparing the results in Fig. 5 with those of Section IV-B, we concluded that the three algorithms exhibit similar high coverage, with the Imp-3 occasionally achieving



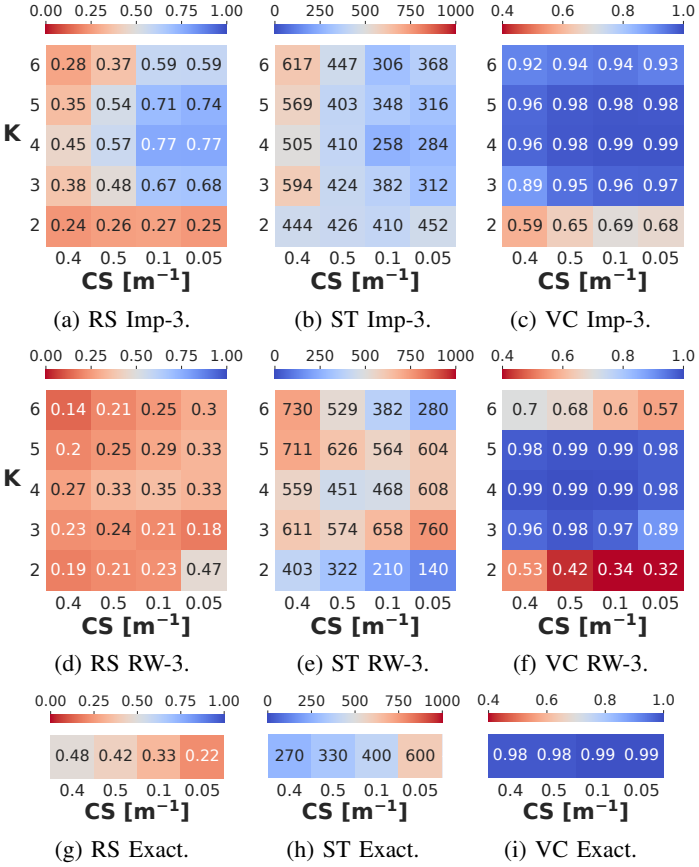
(a) $m = 1$ (b) $m = 7$ (c) $m = 15$

Fig. 5: Average coverage over time for different hole sizes and R_s. The shaded bands represent the 95% confidence interval.

even higher values. Notably, our algorithm demonstrates a much shorter transient period than both the benchmark and the algorithm in [12], albeit at the expense of employing a slightly higher number of agents.

D. Impressionist Analysis

We investigated the effect of different temporal and spatial resolutions, to determine the presence of thresholds at which the global behavior deteriorates. We considered K in $\{2, 3, 4, 5, 6\}$ and CS in $\{0.4, 0.2, 0.1, 0.05\}\text{m}^{-1}$, that corresponds to receiving data every $\{0.5, 1, 2, 4\}$s, respectively. For each combination, we performed 100 simulations using the default configuration and the parameters in Tab. I. To summarize the resulting trends of the average coverage over time, we used three indicators: (i) *Rise Speed* (RS): the average slope of the transient; (ii) *Settling Time* (ST): the time required to reach convergence; (iii) *Value at Convergence* (VC): the value reached by the trend at convergence. Fig. 6c shows that our algorithm achieves high coverage regardless of the degree of bearing quantization and perception frequency, except for $K = 2$. This is a very restrictive setting, in which an agent can only perceive if something is in front of or behind it, resulting in a drop to 59% of coverage in the worst case. However, with as little as 3 sectors, and the appropriate CS, the algorithm can reach up to 97% of coverage. Higher number of sectors helps to compute more precise ACR values, but has detrimental effect on the navigation logic, slightly decreasing the attained coverage. In case of too high CS, the frequent perceptions might conflict, stalling the agents, and thus resulting in gradual incline transients and longer time to convergence (Fig. 6a and Fig. 6b). The RW-3 presents slower transients with respect to Imp-3 (cf. Fig. 6a with Fig. 6d), achieving similar values of coverage for K in $\{3, 4, 5\}$ and much lower for too wide or too narrow resolutions, as shown in Fig. 6f. As discussed in Section IV-B2, the Exact method usually presents longer transients (Fig. 6g and Fig. 6h), but it achieves very high coverage, as reported in Fig. 6i, independently from the cognition speed. Overall, the Exact method performs better with more frequent perception, while our algorithm achieves similar results even with three quantization levels for the bearing and receiving updates only every four seconds.



(a) RS Imp-3. (b) ST Imp-3. (c) VC Imp-3.

(d) RS RW-3. (e) ST RW-3. (f) VC RW-3.

(g) RS Exact. (h) ST Exact. (i) VC Exact.

Fig. 6: Summary of the average coverage over time for different values of K and CS. SR is measured in % increase in coverage per second, ST in seconds. Better outcomes are closer to blue.

V. Conclusions

In this work, we addressed the problem of hole detection and healing in case of limited access to a low amount of information. We proposed a new swarm-based impressionist algorithm, in which the agents rely on severely quantized range and bearing information as their only way to perceive the environment. They use this data to navigate toward the hole and place in the first viable position to restore the coverage. Experimental results show that our solution reaches similar or higher coverage than the state-of-the-art and a benchmark based on random walk. Moreover, it reaches convergence much faster, at the cost of using slightly more agents. The investigation of its impressionist traits revealed that it can reinstate up to 97% of the coverage with as little as three quantization levels for the bearing and five for the distance, while perceiving the environment with a frequency four times lower: being ideal for time-sensitive and prohibitive missions.

Acknowledgment

Work partially supported by: (i) the University of Pisa, in the framework of the PRA 2022 101 project "Decision Support Systems for territorial networks for managing ecosystem services"; (ii) the European Commission under the NextGenerationEU program, Partenariato Esteso PNRR PE1

- "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI", and PNRR - M4 C2, Investment 1.5 "Creating and strengthening of innovation ecosystems, building territorial R&D leaders", project "THE - Tuscany Health Ecosystem", Spoke 6 "Precision Medicine and Personalized Healthcare"; (iii) the Italian Ministry of Education and Research (MIUR) in the framework of the FoReLab project (Departments of Excellence) and of the "Reasoning" project, PRIN 2020 LS Programme, Project number 2493 04-11-2021.

REFERENCES

[1] C. Qiu, H. Shen, and K. Chen, "An energy-efficient and distributed cooperation mechanism for k-coverage hole detection and healing in wsns," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1247–1259, 2017.

[2] B. Khalifa, Z. Al Aghbari, and A. M. Khedr, "A distributed self-healing coverage hole detection and repair scheme for mobile wireless sensor networks," *Sustainable Computing: Informatics and Systems*, vol. 30, p. 100428, 2021.

[3] A. M. Khedr, W. Osamy, and A. Salim, "Distributed coverage hole detection and recovery scheme for heterogeneous wireless sensor networks," *Computer Communications*, vol. 124, pp. 61–75, 2018.

[4] S. Zhai, Z. Tang, D. Wang, Z. Li, X. Chen, D. Fang, and F. Chen, "Coverage hole detection and recovery in wireless sensor networks based on rssi-based localization," in *2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC)*, vol. 2, pp. 250–257, IEEE, 2017.

[5] W. Li and Y. Wu, "Tree-based coverage hole detection and healing method in wireless sensor networks," *Computer networks*, vol. 103, pp. 33–43, 2016.

[6] M. Davoodi, E. Delfaraz, S. Ghobadi, and M. Masoori, "Hole detection and healing in hybrid sensor networks," *arXiv preprint arXiv:2106.10659*, 2021.

[7] A. Soundarya and V. Santhi, "An efficient algorithm for coverage hole detection and healing in wireless sensor networks," in *2017 1st International conference on electronics, materials engineering and nano-technology (IEMENTech)*, pp. 1–5, IEEE, 2017.

[8] Y. Xu, Z. Zeng, and O. Ding, "An energy efficient hole repair node scheduling algorithm for wsn," *Wireless Networks*, vol. 23, pp. 103–116, 2017.

[9] R. Kadu and K. Malpe, "Movement-assisted coverage improvement approach for hole healing in wireless sensor networks," in *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–4, IEEE, 2017.

[10] N. Kukunuru, D. RajyaLakshmi, and A. Damodaram, "Hybrid approach for detecting and healing the coverage-hole in wireless sensor network," in *2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014)*, pp. 110–115, IEEE, 2014.

[11] C. So-In, T. G. Nguyen, and N. G. Nguyen, "An efficient coverage hole-healing algorithm for area-coverage improvements in mobile sensor networks," *Peer-to-Peer Networking and Applications*, vol. 12, no. 3, pp. 541–552, 2019.

[12] G. Simionato, F. A. Galatolo, and M. G. Cimino, "Swarms of artificial platelets for emergent hole detection and healing in wireless sensor networks," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 75–83, 2023.

[13] u-blox. Product Summary: XPLR-AOA-2. 2022. Available online: https://content.u-blox.com/sites/default/files/XPLR-AOA-2_ProductSummary_UBX-21017999.pdf (accessed on 30 July 2023).

[14] C. Bilaloğlu, M. Şahin, F. Arvin, E. Şahin, and A. E. Turgut, "A novel time-of-flight range and bearing sensor system for micro air vehicle swarms," in *International Conference on Swarm Intelligence*, pp. 248–256, Springer, 2022.

[15] J. F. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano, "3-d relative positioning sensor for indoor flying robots," *Autonomous Robots*, vol. 33, pp. 5–20, 2012.

[16] F. Berlinger, J. T. Ebert, and R. Nagpal, "Impressionist algorithms for autonomous multi-robot systems: Flocking as a case study," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11562–11569, IEEE, 2022.

[17] B. Pang, Y. Song, C. Zhang, and R. Yang, "Effect of random walk methods on searching efficiency in swarm robots for area exploration," *Applied Intelligence*, vol. 51, pp. 5189–5199, 2021.

[18] N. Seenu, L. Manohar, N. M. Stephen, K. C. Ramanathan, and M. Ramya, "Autonomous cost-effective robotic exploration and mapping for disaster reconnaissance," in *2022 10th International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing (ICETET-SIP-22)*, pp. 1–6, IEEE, 2022.

[19] S. Mehta and A. Malik, "A swarm intelligence based coverage hole healing approach for wireless sensor networks," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 7, no. 26, pp. e8–e8, 2020.

[20] J. Wang, C. Ju, H.-j. Kim, R. S. Sherratt, and S. Lee, "A mobile assisted coverage hole patching scheme based on particle swarm optimization for wsns," *Cluster Computing*, vol. 22, no. 1, pp. 1787–1795, 2019.

[21] L. Yan, Y. He, and Z. Huangfu, "A fish swarm inspired holes recovery algorithm for wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 27, no. 1, pp. 89–101, 2020.

[22] M. R. Senouci, A. Mellouk, and K. Assnoune, "Localized movement-assisted sensordeployment algorithm for holedetection and healing," *IEEE Transactions on parallel and distributed systems*, vol. 25, no. 5, pp. 1267–1277, 2013.

[23] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3299–3304, IEEE, 2017.

[24] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.

[25] A. Weinstein, A. Cho, G. Loianno, and V. Kumar, "Visual inertial odometry swarm: An autonomous swarm of vision-based quadrotors," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1801–1807, 2018.

[26] S. Zhang, D. Martinez, and J.-B. Masson, "Multi-robot searching with sparse binary cues and limited space perception," *Frontiers in Robotics and AI*, vol. 2, p. 12, 2015.

[27] K. Y. Chin, Y. Khaluf, and C. Pinciroli, "Minimalistic collective perception with imperfect sensors," *arXiv preprint arXiv:2209.12858*, 2022.

[28] N. Hoff, R. Wood, and R. Nagpal, "Effect of sensor and actuator quality on robot swarm algorithm performance," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4989–4994, IEEE, 2011.

[29] T. Aust, M. S. Talamali, M. Dorigo, H. Hamann, and A. Reina, "The hidden benefits of limited communication and slow sensing in collective monitoring of dynamic environments," in *International Conference on Swarm Intelligence*, pp. 234–247, Springer, 2022.

[30] B. Wang, *Coverage control in sensor networks*. Springer Science & Business Media, 2010.

[31] H. Zhang, J. C. Hou, *et al.*, "Maintaining sensing coverage and connectivity in large sensor networks.," *Ad Hoc Sens. Wirel. Networks*, vol. 1, no. 1-2, pp. 89–124, 2005.

[32] K. McGuire, C. De Wagter, K. Tuyls, H. Kappen, and G. C. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Science Robotics*, vol. 4, no. 35, p. eaaw9710, 2019.

[33] G. Simionato and F. A. Galatolo, "https://github.com/giadasimionato/hdhsim.git," 2023.

[34] N. I. Fisher, *Statistical analysis of circular data*. cambridge university press, 1995.

[35] C. Dimidov, G. Oriolo, and V. Trianni, "Random walks in swarm robotics: an experiment with kilobots," in *International conference on swarm intelligence*, pp. 185–196, Springer, 2016.

[36] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

[37] V. N. Thatha, A. S. Rao, N. R. Reddy, and M. Silparaj, "Application of hyper parameter optimization algorithms using big data," in *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–4, IEEE, 2022.

[38] K. M. Kam, L. Zeng, Q. Zhou, R. Tran, and J. Yang, "On assessing spatial uniformity of particle distributions in quality control of manufacturing processes," *Journal of Manufacturing Systems*, vol. 32, no. 1, pp. 154–166, 2013.