Patrolling a Region with a Structured Swarm of Robots with Limited Individual Capabilities

Sándor P. Fekete
Computer Science Department
TU Braunschweig
Braunschweig, Germany
Email: s.fekete@tu-bs.de

Seoung Kyou Lee
Computer Science Department
Rice University
Houston, TX, USA
Email: sl28@rice.edu

Alejandro López-Ortiz
David R. Cheriton School of Computer
University of Waterloo
Waterloo, ON N2L 3G1, Canada
Email: alopez-o@uwaterloo.ca

Daniela Maftuleac
David R. Cheriton School of Computer
University of Waterloo
Waterloo, ON N2L 3G1, Canada
Email: dmaftule@uwaterloo.ca

James McLurkin
Computer Science Department
Rice University
Houston, TX, USA
Email: jmclurkin@rice.edu

***Abstract*—We present methods for patrolling and surveillance in an environment with a distributed swarm of robots with limited capabilities. We make use of a distributed triangulation of the work space, in which a set of n stationary sensors provides coverage control; in addition, there are k mobile robots that can move between the sensors.**

Building on our prior work on structured exploration of unknown spaces with multi-robot systems, we can make use of a triangulation that is constructed in a distributed fashion and guarantees good local navigation properties, even when sensors and robots have very limited capabilities. This physical data structure allows triangles to sense, compute, and communicate the information required to guide navigating robots while circulating in the region. We present: 1) A description of a real-world robot platform (the r-one robots) with limited capabilities that allows coverage, communication and mobility 2) A summary of how to achieve coverage by building a triangulation of the workspace, and the ensuing properties. 3) A description of a simple local policy (LRV, for *Least Recently Visited*) for achieving coverage by the patrolling robots. 4) A description of a alternative local policy (AR, for *Age Rank*) that identifies important vertices, thus ensuring smaller refresh times for those, and allowing more flexible response to events. 5) Experimental results for both policies.

I. Introduction and Related Work

Many practical applications of multi-robot systems, such as search-and-rescue, exploration, mapping and surveillance require robots to disperse across a large geographic area. Large populations of robots can maintain coverage of the environment after the dispersion is complete. These large populations will require the individual robots to be low-cost, precluding the use of expensive sensors or actuators. A solution is to deploy a heterogeneous group of robots, with many small robots to map the environment and perform simple local communication, and only a few capable robots to patrol or respond to events. After deployment of the mapping robots, controlling the more powerful robots amounts to a a *coverage control* problem: how to move the navigating robots in order to ensure small latency in visiting all areas of the surveyed environment.

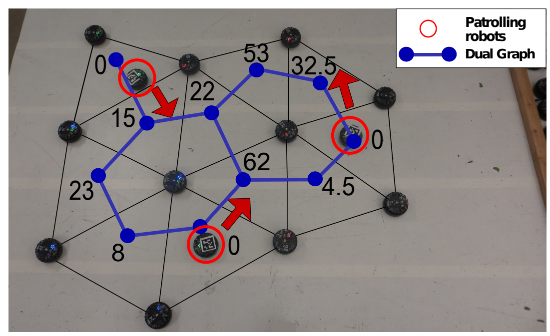

Fig. 1: Sample triangulated network with three patrolling robots (red circles). Blue lines indicate the dual graph of the triangulation. The number in each triangle represents the refresh timestamp of each triangle. The LRV policy lets patrolling robots move toward the adjacent triangles having largest timestamp (red arrows).

There are many approaches to coverage control [7], [3], [4], [5]. Using a heterogeneous system of robots allows us to deploy a large number of simple robots to explore the environment in a structured way [12]. In this work, the robots *triangulate* the environment, producing a *physical data structure* – a set of triangles formed by the positions of the robots. This physical data structure allows triangles to sense, perform simple computations, and communicate the information required to guide navigating robots for achieving coverage control. Fig. 1 shows an example experiment.

Making use of the dual graph of the triangulation, we can perform simple navigation tasks based on purely local information, while achiving constant stretch for navigation tasks, i.e, staying within a bounded factor of the minimum achievable travel time with full information and perfect control; see our paper [12] for more details. The next level of challenges considers how to move the more flexible, mobile robots around the dual graph in order to ensure that each subarea gets visited at regular intervals.

This paper presents:

- A summary of underlying structural properties that we can achieve for the stationary set of sensors.
- A discussion of the local *Least Recently Visited* policy achieving full coverage of the guarded region by the mobile sensors.
- The alternative *Age Rank* local policy that focuses on patroling only the "important" portions, with the idea of staying central and being more flexible to respond to events.
- Hardware and simulation experiments.

Basic Assumptions

We are interested in policies for a population of n stationary sensors and k mobile robots, and focus our attention on approaches applicable to small, low-cost devices with limited sensors and capabilities. In this work, we assume that robots do not have a map of the environment, nor the ability to localize themselves relative to the environment geometry, *i.e.* SLAM-style mapping is beyond the capabilities of our platform. We exclude solutions that use centralized control, as the communication and processing constraints do not allow these approaches to scale to large populations. We also do not assume that GPS localization or external communication infrastructure is available, which are limitations present in an unknown indoor environment. We assume that the communication range is much smaller than the size of the environment, so a multi-hop network is required for communication. Finally, we assume that the devices know the geometry of their local communications network. This *local network geometry* provides each mobile robot with relative pose information about its neighboring sensors.

We assume that a large group of simple robots has dispersed into the environment and triangulated the environment; see [2] for an illustrative video, and [12] for a technical paper. The main objective is to perform patrolling by the mobile robots, i.e., a protocol that makes sure the robots circulate within the region, visiting portions at regular intervals, and are able to respond to events when they happen. Clearly, these objectives can be carried out in different ways: (1) We can aim for strictly local policies that ensure *all* dual vertices get visited infinitely often. (2) We can aim for strictly local policies that ensure the "important" vertices get visited frequently, so that the response time to an event can be kept small. As we will demonstrate, there is a simple policy (called *Least Recently Visited*, LRV) that can ensure property (1), albeit at the price of potentially exponentially coverage times. On the other hand, we propose a local heuristic (called *Age Rank*, AR) that uses only local computation for identifying central dual vertices with relatively large time stamp for ensuring preferential refresh times; this comes at the possible price of lower visiting frequencies for lesser dual vertices, but aims at getting in their vicinity more often.

Related Work

Our results rely on the computational power of many small robots distributed throughout the environment. There is a large body of work on using distributed sensors networks for robot navigation, we cannot cover them all here, but note Batalin's approach, which is similar to our own [1]. Our network is composed of triangles, which provide useful geometric properties. Approaches like Spears $et.al.$ in [16] build a triangulated configuration using potential fields, but the network does not have a physical data structure, so the robots never recognize that they form triangles. Our approach allows us to use triangles as computational elements, which support practical distributed computations. Geraerts [10] or Kallmann [11], use a triangulated environment for path planning, but require global information and localization. Our approach is fully distributed, using only local information and communications.

Optimizing the refresh frequency when patrolling a graph amounts to finding a shortest roundtrip that visits all vertices, i.e., the well-known *Traveling Salesman Problem* (TSP). The TSP is known to be NP-hard, even for full information and central control. Making use of approximation algorithms for the TSP, policies with a limited amount of global information have been proposed, e.g., based on building a minimum spanning tree [8]; however, this does require some global communication, even with only local connections.

An alternative option is to make use of only local information that can be tracked by the dual vertices. This can be based on time elapsed since the last visit by a robot. This has been considered in the context of token passing in decentralized adhoc networks. As Malpani et al. [13] showed, the policy *Least Recently Visited* (LRV) ensures that finite refresh times can be guarantee. However, Cooper et al. [6] demonstrated that the policy may result in refresh times that are exponential in the size of the graph. More on this will be discussed in Section III, where we discuss LRV in the context of mobile robots.

In the context of patrolling, it may be more relevant to circulate among a set of central locations from which every other portion of the region can be reached relatively quickly when the need arises, rather than visiting all nodes at larger interval. We will propose a policy that makes use of only local information, and has some resemblance to the well-known *Page Rank*; see [14], and [15] for a study of this and other centrality measures.

II. Model and Preliminaries

A. *Robot Model*

We have a system of n triangulation robots and p navigation robots. The communication network is an undirected graph $G = (V, E)$. Each robot is modeled as a vertex, $u \in V$, where V is the set of all robots and E is the set of all robot-to-robot communication links. The neighbors of each vertex u are the set of robots within line-of-sight communication range r_{max} of robot u, denoted $N(u) = \{v \in V \mid \{u, v\} \in E\}$. We assume all network edges are also navigable paths. Robot u sits at the origin of its local coordinate system, with the $\hat{x}$-axis aligned with its current heading. Robot u cannot measure distance to its neighbors, but can only measure the *bearing* and *orientation*.

B. *Environment Triangulation*

We make use of our MATP triangulation algorithm [9], [2], which explores an unknown region. The exploration proceeds in a structured, breadth-first fashion, leaving a triangulated network in its wake. These triangles, and their associated dual graph, can be modeled as computational elements, but the actual processing and communication occurs on the robots. This allows us to model information stored in a distributed fashion on triangles, and reason about communication via the dual graph.

III. Local Patrolling Policies

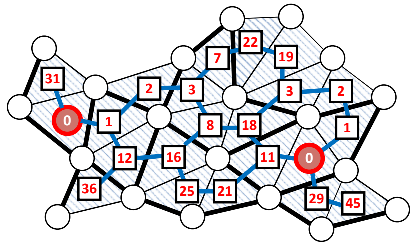

Fig. 2: An example of triangulation with a dual graph. Thin circles, red circles, blue lines, and bold black lines represent stationary robots, patrolling robots, dual graph, and primal graph, respectively. The number of each square indicates the current refresh time of each triangle. There is no explicit data structure storing the dual graph. Instead, information of each triangle is stored in the stationary robots that form the triangle.

We use the basic triangulation to support a simple distributed patrolling algorithm. Each triangle stores the time elapsed since its last visit, the *refresh time*, $RT_t(\Delta_i)$. The goal is to minimize the refresh time of each triangle.

A basic local patrolling algorithm (called LRV, for *least recently visited*) simply moves each robot into the adjacent triangle with the largest $RT_t(\Delta_i)$. While simple, this policy is known to produce complete coverage. In the rest of this section for clarity of presentation, we refer to the dual graph G induced by the triangulation.

The LRV policy has been studied in various contexts. Fig. 5a demonstrates that it exhibits relatively good behavior in practice; it shows data for the LRV policy under the patrolling experiment shown in Fig 1. The experiment starts with one navigating robot; we add another at 1500 and 3600 seconds. As we deploy more navigating robots, the maximum $RT_t(\Delta_i)$ decreases, seen in the red line that shows the 400s moving average.

However, theoretically it is known that the worst-case behavior of LRV can be exponential in the number of nodes in the graph for certain graphs. That is, for every n there exists a graph with n vertices in which the largest refresh time for a node is $\exp(\Theta(n))$ [6]. Figure 3 depicts the basic structure of one such graph, which filters a fixed percentage (1/3 to be precise) of all left-to-right paths that go past the diamond-like structures or gadgets. If we connect $\Theta(n)$ such gadgets in series, we will require a total of $(3/2)^{\Theta(n)}$ paths, starting from the left for at least one of them to reach the rightmost point in the series. Because of this, we supplement the LRV policy with a novel policy that aims to patrol central areas more often. This policy uses a simple local policy update method to compute a centrality score for each node. This score leads the robot in the direction of central nodes that have not been visited in the recent past. It is vaguely inspired by the *Pagerank* score assigned to nodes in the web graph [14] and as such we term it *Age Rank*. A high-level description of the distributed scoring mechanism is as follows. Each node v has an Age

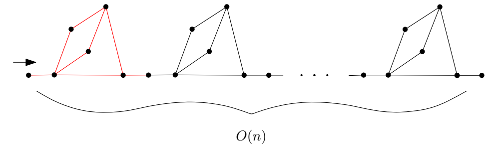

Fig. 3: Graph with n vertices with a chain of $\Theta(n)$ gadgets. A single gadget is colored in red for illustration purposes. Patrolling takes exponential time in the worst case [6].

Rank local score, which is initialized to the refresh time, that is, $AR_0(v) := RT_0(v)$. Then at each step, for every node v in the graph it transmits to its neighbours an additional score of $c/\delta(v)AR_t(v)$ where $\delta(v)$ is the degree of v and $0 < c < 1$. Simultaneously the node receives the transmitted amounts from its neighbours.

$$AR_{t+1}(v) = (1-c)\,AR_t(v) + \sum_{u \in N(v)} (c\,AR_t(u))/\delta(u),$$

or more precisely (using matrix notation), we have

$$\overline{AR}_{t+1} = c(A^T - A)\overline{AR}_t + (1-c)\overline{AR}_t,$$

where A denotes the row normalized adjacency matrix and $\overline{AR}_t$ denotes the column vector of age rank scores at time t. Here A is the adjacency matrix such that every row is divided by the degree of the row node and thus each row adds up to 1.

IV. Experimental Results

Fig. 4: Snapshot of the r-one robot used in the experiments.

Fig 1 shows our experimental setup for testing LRV patrolling. We built eleven triangles using eleven low-cost robots as shown in Fig. 4. We place between one and three robots to patrol along the dual graph in the triangulation. For the next triangle, each patrolling robot then selects a triangle that has maximum timestamp among all adjacent triangles. Fig. 5a shows data from the patrolling experiment shown in Fig. 5a. The experiment starts with one navigating robot; we add another at 1500 and 3600 seconds. As we deploy more navigating robots, the maximum $RT_{\Delta_i}(t)$ decreases, seen in the red line that shows the 400s moving average.

We also simulated the patrolling performance using 1, 3, 10, and 20 robots in an triangulated environment with a large number (101) of triangles, as shown in Fig 5b. Each color indicates the refresh time of each triangle. Fig. 5c shows the comparison of the maximum refresh time among all triangles at each round. As we put more patrolling robots, the maximum refresh time rapidly decreases. In addition, we could measure the time to cover all triangles at least once. For $k = 1$, it takes about 5000 rounds while only about 700 rounds is required for $k = 20$.

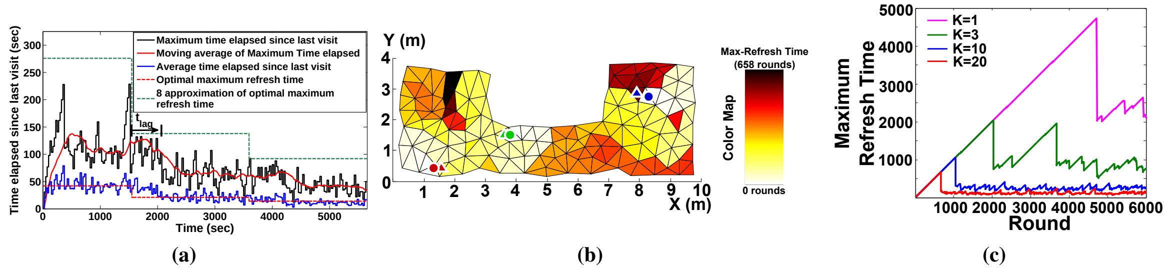

Fig. 5: (a) LRV Patrolling experiment results from a triangulation with 11 triangles. The setup is shown in Fig. 1. The experiment starts with one patrolling robot, and adds additional patrolling robots after 1500 seconds and 3600 seconds, respectively. **(b)** Distributed patrolling simulation screen shot. We test with 1, 3, 10, and 20 patrolling robots. Moving circles and equilateral triangles represent robots and the target triangle of each robot, respectively. The color of each triangle indicates current refresh time. As the current refresh time of a triangle increases, its color is close to black. **(c)** Resulting maximum refresh time at each round with $k = 1, 3, 10, 20$. As we add more robots, the maximum refresh time for each round decreases.

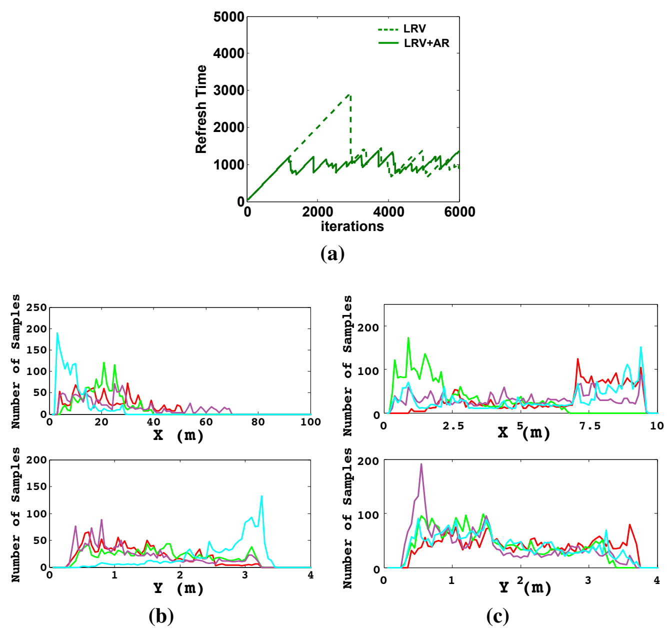

Fig. 6: (a) A patrolling simulation with the modified policy based on Age Rank (AR), and $k = 4$ robots. The setup is the same as in in Fig. 5b. We show a comparison of the original LRV policy with the one enhanced by AR (LRV+AR). Shown are the refresh times for both policies for getting within dual distance of $d = 1$, showing that the enhanced local policy is better for getting close to all dual triangles. **(b)** and **(c)** Position distribution of each robot for the first 1500 rounds with LRV method and AgeRank+LRV method.

A simulation based on a combination of LRV and Age Rank is shown in Fig 6a. Shown is a setting with $k = 4$ robots, and for getting within a dual distance of $d = 1$ of each triangle. Clearly, this produces even better results.

Figs. 6b and 6c show the distribution of the position of each robot for the first 1500 iterations with only LRV and combined method, respectively. Until 1500 iterations, no robot (with only LRV method) explores to the right side of environment. On the other hands, combined method spreads all robots evenly in the environment, and gives more chance to navigate the region with high RT value.

V. Conclusion

We have demonstrated how a combination of a weak stationary swarm with a set of mobile robots with limited capabilities gives rise to problems of distributed coverage. A simple policy (based on purely local information without any sophisticated communication between devices) allows complete coverage of the mapped area, but may lead to high refresh times for all portions; we gave an alternative approach (called *Age Rank*) that focuses only on central vertices, and uses only local computations.

It is clear that there are a variety of other policies that can be used, provided that the involved stationary and mobile devices can employ more sophisticated communication protocols. It is an interesting task to balance the required assumptions with the resulting performance guarantees.

References

[1] Maxim Batalin and Gaurav S. Sukhatme. Using a sensor network for distributed Multi-Robot task allocation. In *IEEE International Conference on Robotics and Automation*, pages 158–164, New Orleans, Louisiana, April 2004.

[2] Aaron Becker, Sándor P. Fekete, Alexander Kröller, Seoung Kyou Lee, James McLurkin, and Christiane Schmidt. Triangulating unknown environments using robot swarms. In *Proc. 29th Annu. ACM Sympos. Comput. Geom.*, pages 345–346, 2013. Video available at http://imaginary.org/film/triangulating-unknown-environments-using-robot-swarms.

[3] Subhrajit Bhattacharya, Nathan Michael, and Vijay Kumar. Distributed coverage and exploration in unknown non-convex environments. In *Distributed Autonomous Robotic Systems*, pages 61–75. Springer, 2013.

[4] Andreas Breitenmoser, Mac Schwager, J-C Metzger, Roland Siegwart, and Daniela Rus. Voronoi coverage of non-convex environments with a group of networked robots. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4982–4989. IEEE, 2010.

[5] CH Caicedo-Nuez and Milos Zefran. A coverage algorithm for a class of non-convex regions. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4244–4249. IEEE, 2008.

[6] Colin Cooper, David Ilcinkas, Ralf Klasing, and Adrian Kosowski. Derandomizing random walks in undirected graphs using locally fair exploration strategies. *Distributed Computing*, 24(2):91–99, 2011.

[7] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *Robotics and Automation, IEEE Transactions on*, 20(2):243–255, 2004.

[8] Yehuda Elmaliach, Noa Agmon, and Gal A. Kaminka. Multi-robot area patrol under frequency constraints. *Ann. Math. Artif. Intell.*, 57(3-4):293–320, 2009.

[9] Sándor P. Fekete, Tom Kamphans, Alexander Kröller, J.S.B. Mitchell, and Christiane Schmidt. Exploring and triangulating a region by a swarm of robots. In *Proc. APPROX 2011*, volume 6845 of *LNCS*, pages 206–217. Springer, 2011.

[10] Roland Geraerts. Planning short paths with clearance using explicit corridors. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1997–2004. IEEE, 2010.

[11] Marcelo Kallmann. Path planning in triangulations. In *Proceedings of the IJCAI workshop on reasoning, representation, and learning in computer games*, pages 49–54, 2005.

[12] SeoungKyou Lee, Aaron Becker, Sandor P. Fekete, Alexander Kroller, and James McLurkin. Exploration via structured triangulation by a multi-robot system with bearing-only low-resolution sensors. In *International Conference on Robotics and Automation (ICRA), to appear*, 2014.

[13] Navneet Malpani, Yu Chen, Nitin H. Vaidya, and Jennifer L. Welch. Distributed token circulation in mobile ad hoc networks. *IEEE Trans. Mob. Comput.*, 4(2):154–165, 2005.

[14] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web, 1999.

[15] Nicola Perra and Santo Fortunato. Spectral centrality measures in complex networks. *Phys. Rev. E*, 78:036107, Sep 2008.

[16] W. M. Spears, D. F. Spears, J. C. Hamann, and R. Heil. Distributed, Physics-Based control of swarms of vehicles. *Autonomous Robots*, 17(2):137–162, 2004.