On Multi-Robot Area Coverage

Alireza Davoodi ALIREZA_DAVOODI@SFU.CA
School of Interactive Arts and Technology, Simon Fraser University
Pooyan Fazli POOYANF@CS.UBC.CA
Department of Computer Science, University of British Columbia
Philippe Pasquier PASQUIER@SFU.CA
School of Interactive Arts and Technology, Simon Fraser University
Alan K. Mackworth MACK@CS.UBC.CA
Department of Computer Science, University of British Columbia

Abstract

This paper presents an approach allowing a team of robots each with limited visibility to cover an area in the presence of different types of static obstacles. We introduce *Reduced-CDT*, an environment representation method based on *Constrained Delaunay Triangulation*. A new graph segmentation method called *Multi-Prim's* is used to decompose the *Reduced-CDT* and construct a forest of *partial spanning trees (PSTs)*. Each *PST* is then modified through a mechanism called the *Constrained Spanning Tour (CST)* to build a cycle which is assigned to an explorer robot. Subsequently, robots start navigating the cycles and consequently cover the whole area. The proposed approach is guaranteed to be complete and robust.

1. Introduction

Multi-robot area coverage has applications in different scenarios such as search and rescue operations, planetary exploration, intruder detection, environment monitoring and so on. In this task a team of robots are cooperatively trying to visit or cover the entire area, possibly containing obstacles, with their actuators or sensors. Robots could have different capabilities (e.g. different sensor ranges or actuator types). The goal is to build efficient paths for each robot which jointly cover the whole area. Several optimization critira can be considered in building these coverage paths for robots including minimizing the time, building balanced paths for each robot of the team, initial location of the robots, and etc. Several research communities including robotics/agents (Choset, 2001), sensor networks (Meguerdichian et al., 2001) and computational geometry (Carlsson et al., 1991) work on this class of problems. In computational geometry this problem stems from the *"Art Gallery"* problem and its variation for mobile guards called the *"Watchman Route"* problem (Chin & Ntafos, 1986). Most research done on the above problem definitions deal with simple polygons and unconstrained guard visibility. In the robotics community, previous research on multi-robot area coverage are mostly focused on grid-based approaches (Hazon & Kaminka, 2008). Grid-based methods have limitations since they do not consider the structure of the environment and have difficulties handling partially occluded cells or covering areas close to the boundaries in the continuous spaces. On the other hand graph-based approaches which employ structures such as a visibility graph or Voronoi diagram for environment representation do not suffer those restrictions. While traversing a visibility graph guarantees covering the whole environment in continuous spaces, it might include many redundant movements in large workspaces. To address the concerns mentioned above including the robots' limited visibility constraint, *Reduced-CDT* is introduced as a more appropriate representation of the environment. Due to the distributed characteristic of the underlying problem, another mechanism called *Multi-Prim's* is applied to decompose the environment among the explorer robots and support robustness through handling individual robot failure.

2. Multi-Robot Area Coverage with Limited Visibility

In this paper we present a centralized cooperative approach to cover a known environment using an arbitrary number of robots. Robots are assumed to have 360° field of view and a predefined circular visibility range. Our coverage method is composed of four main steps. First, it determines the location of guards required to visually cover a given 2D environment, considering the limited visibility of the robots' cameras. Then, it builds a graph-based representation of the environment using *Reduced-CDT*. An algorithm called *Multi-Prim's* is introduced to partition the graph vertices and construct a forest consisting of as many partial spanning trees as there are explorer robots. Afterward, a new method called *Constrained Spanning Tour* is used to build a cycle on each consequent tree of the forest, and finally, the cycles are allocated individually to the explorer robots.

2.1. Locating Guards

Our approach uses a variation of the algorithm proposed in (Kazazakis & Argyros, 2002) to locate a sub-optimal number of guards required to visually cover the whole environment. To this aim, the proposed approach decomposes the initial environment, i.e., a 2D, simple, non-convex polygon with static obstacles, into a collection of convex polygons. Then, a divide and conquer method is applied, to successively divide each of the resulting convex polygons into smaller convex sub-polygons until each of them can be visually covered by one guard. The decided static guards can jointly cover the whole environment while satisfying the visibility constraint of the robots.

2.2. Environment Representation

Algorithm 1 describes the steps of the construction of *Reduced-CDT* on a given environment. Given the set of obstacles O and their corresponding endpoints P, the algorithm first uses the method *LocateGuards*(), as explained in the previous section, in order to create the set $SG = \{g_1, g_2, ..., g_n\}$ of the static guards. The method $CDT(SG, O)$ is then applied to construct the *Constrained Delaunay Triangulation*, $G_{cdt}(V_{cdt}, E_{cdt})$. At the next step, the *Floyd-Warshall* algorithm is used to find the set $R = \{(r_{ij}, v_i, v_j) | v_i, v_j \in V_{cdt}\}$ of shortest paths, r_{ij}, and the set $F = \{(c_{ij}, v_i, v_j) | v_i, v_j \in V_{cdt}\}$ of minimum distances, c_{ij}, between each two vertices v_i and v_j. The minimum value of all the minimum distances in F is then selected provided that both the endpoints of the corresponding shortest path in R are in SG. The chosen path including all its vertices and edges forms the initial component called *Connected Component*, G_{cc}. Next, among all the guards that have not yet been added to the component, the algorithm finds the closest guard to the current component and eventually merges the corresponding shortest path with the current component. Following the same process, the algorithm keeps expanding the *Connected Component* until there is no more guard to be added to the current component. The resultant *Connected Component* is the final graph G_{rcdt}. This algorithm is of complexity order of $O((n+m)^3)$ where n and m are the number of guards and endpoints of obstacles, respectively.

Algorithm 1 Reduced-CDT

```
Input:
O = {o_1, o_2, ..., o_z} //Set of Obstacles
P = {p_1, p_2, ...., p_m} //Endpoints of Obstacles
α = Visibility Range
Ouput:
Graph G_rcdt (V_rcdt, E_rcdt) where V_rcdt = SG ∪ P̃, P̃ ⊂ P
set V_rcdt = φ and E_rcdt = φ
SG = LocateGuards(O, P, α)
G_cdt(V_cdt, E_cdt) = CDT(SG, O)
(F, R) = FloydWarshall(G_cdt)
(i, j) = arg min_(i,j) (c_ij | (c_ij, v_i, v_j) ∈ F)
r_ij = GetTheCorrespondingShortestPath(i, j)
CC(V_cc, E_cc) = InitialConnectedComponent(r_ij)
while ¬all the guards added do
    g = FindClosestGuardTo(CC)
    Expand(CC, g)
end while
V_rcdt = V_cc and E_rcdt = E_cc
```

2.3. Multi-Prim's Algorithm

The *Multi-Prim's* algorithm extends the *Prim's* algorithm used to build minimum spanning tree of a weighted graph. The *Multi-Prim's* algorithm has a weighted graph as an input and outputs a forest of partial spanning trees. This algorithm starts by initiating as many trees as there are explorer robots. A corresponding starting point for a robot is the nearest vertex of the *Reduced-CDT* to that robot, provided that they are mutually visible as well. Subsequently, robots try to sequentially expand their own trees (one edge at a time) using *Prim's* algorithm until all the vertices of the *Reduced-CDT* are visited at least once. The vertices of the *Reduced-CDT* are visited in a way that it satisfies the following three constraints: (1) find the nearest adjacent vertex and its corresponding edge and add both to the tree provided it does not create a cycle. (2) avoid adding a vertex which has already been visited by some robots, unless there is no other unvisited accessible vertex in the robot's field of view, and (3) the algorithm terminates when all the vertices of the graph have been visited by at least one robot.

2.4. Constrained Spanning Tour

The next step is to construct a cycle on each partial spanning tree resulting from *Multi-Prim's* algorithm. To this end, we introduce an algorithm called *Constrained Spanning Tour (CST)* which is an improved variation of the *Doubled-Minimum Spanning Tree (DMST)* algorithm. *DMST* takes a tree as an input and returns a cycle whose length is twice of the length of the tree. In order to form a shorter cycle, *CST* starts from an arbitrary initial point and traverses the vertices in the same way as the *Doubled-Minimum Spanning Tree* algorithm does except whenever it reaches a vertex visited before, it discards it and proceeds to the next vertex along the cycle to find an unvisited one. This process continues until it returns back to the starting point. But due to the existing obstacles in the environment, *CST* uses just the edges of the original graph (*Reduced-CDT* in our case) as a shortcut edge. This algorithm employs a back tracking mechanism to find out the best shortcut. If the shortcut does not belong to the original graph, the next best shortcut will be considered. It keeps following the same way to find the best shortcut edge. *CST* traverses the tree to return back to the initial point. In the worst case the result would be the same as the result of the *Doubled-Minimum Spanning Tree* algorithm.

3. Fault-Tolerant Multi-Robot Area Coverage

Robot failure during execution can be a concern in running the algorithm. Our approach addresses robustness issues through the concept of *Supportive* trees. Two trees of a graph are *Mutually Supportive* if there is at least one bridge connecting those two trees. A bridge can be either a vertex or an edge in common or an edge with endpoints each located on one of the trees. The algorithm uses the forest, built on *Reduced-CDT* by *Multi-Prim's* algorithm, to find all the *Mutually Supportive* trees. It is easily proved that there is at least one *Supportive* tree for each tree of the forest. Robots working on two *Mutually Supportive* trees are also *Mutually Supportive*. Consequently each robot has at least one *Supportive* robot. When a robot fails, all the vertices of its assigned tree are released. Then all of its *Supportive* robots expand their trees through *Multi-Prim's* algorithm to possess the released vertices and to cover the whole environment again.

4. Conclusion and Future Work

This paper presents a new approach for covering a known area cluttered with obstacles by a team of robots with limited visibility range. A new environment representation method called *Reduced-CDT* was introduced. *Multi-Prim's* algorithm is applied to decompose the *Reduced-CDT* graph into a forest of partial spanning trees. *Constrained Spanning Tour* is employed to construct a cycle on each partial spanning tree. Finally, each cycle is assigned to a robot. Since the complexity order of the whole coverage mechanism is mostly dominated by *Reduced-CDT*, the entire approach is a polynomial time algorithm of complexity of $O((n+m)^3)$. Based on the experiments we have conducted on different map-robot configurations, this approach can be efficiently used in dense environments and areas with narrow passages. Also, it is capable of dealing with robot failure. For future work, we intend to extend the current approach to support scalability and heterogeneous agents with different visibility ranges.

References

Carlsson, S., Nilsson, B. J., & Ntafos, S. (1991). Optimum guard covers and m-watchmen routes for restricted polygons. *Proceedings of 2nd Workshop on Algorithms and Data Structures* (pp. 367–378).

Chin, W., & Ntafos, S. (1986). Optimum watchman routes. *SCG '86: Proceedings of the second annual symposium on Computational geometry* (pp. 24–33). New York, NY, USA: ACM.

Choset, H. (2001). Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence* (pp. 113 – 126).

Hazon, N., & Kaminka, G. A. (2008). On redundancy, efficiency, and robustness in coverage for multiple robots. *Robot. Auton. Syst.*, *56*, 1102–1114.

Kazazakis, G. D., & Argyros, A. A. (2002). Fast positioning of limited visibility guards for inspection of 2d workspaces. *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2843–2848).

Meguerdichian, S., Koushanfar, F., Potkonjak, M., & Srivastava, M. B. (2001). Coverage problems in wireless ad-hoc sensor networks. *in IEEE INFOCOM* (pp. 1380–1387).