Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-1 (2016) 195–200

Pseudo Spanning Tree-based Complete and Competitive Robot Coverage using Virtual Nodes

T.D. Ranjitha and K.R. Guruprasad

Department of Mechanical Engineering, National Institute of Technology Karnataka, Surathkal, Mangaluru, 57025, India.
Email: T.D. Ranjitha: ranjithatd.26@gmail.com
K.R. Guruprasad: krgprao@gmail.com

Abstract: In this paper, we propose a new robot coverage algorithm using approximate cellular decomposition. The algorithm uses spanning tree on adjacency graph formed by the decomposed cells. We introduce a concept of a *virtual node* corresponding to a partially occupied cell, leading to a *pseudo spanning tree*. Unlike the existing approximate cellular decomposition based coverage algorithms reported in the literature, the proposed algorithm ensures complete coverage of even partially occupied cells, with minimal (or no) overlapping/retracing of the path. The algorithm is illustrated using examples comparing its coverage performance with that of the STC and Competitive-STC algorithms, two spanning tree based coverage algorithms using approximate cellular decomposition reported in the literature.

Keywords: Coverage path planning, Pseudo spanning tree, robot motion planning.

1. INTRODUCTION

In many real-world applications, such as automatic floor cleaning, lawn moving, autonomous land-mine detection, etc, a mobile robot needs to cover an area of interest completely. A coverage algorithm should cover an area *completely*, with minimal (or no) retraces (or overlap). A survey of various coverage algorithms is provided in Choset (2000). A more recent survey is provided in Galceran and Carreras (2013). While the off-line algorithms require a priori knowledge of the entire environment, no such information is required for on-line or sensor-based coverage algorithms. Most coverage algorithms use either approximate or exact cellular decomposition of the space. Coverage algorithms based on cellular decomposition make the robot pass through each cell only once by avoiding the obstacles. They rely upon an adjacency graph, and use graph search techniques such as breadth-first search or depth-first search to cover all the cells. In exact cellular decomposition based methods such as in Choset (2001); Xu et al. (2014), and Strimel and Veloso (2014), the obstacle-free region is decomposed into cells using on board sensors. Approximate cellular decomposition methods decompose the region (including obstacle) into fine grids of size equal to the robot (or sensor/coverage tool) footprint. If the robot visits each of the cells, then the coverage is complete, and if a cell is not visited more than once, then there is no duplicate coverage (overlap or retracing of the path). In most graph search based coverage algorithms such as in Choset (2001) and Gonzalez et al. (2005), retracing of path cannot be completely avoided. In a spanning tree-based coverage (STC) algorithm proposed in Gabriely and Rimon (2001), graph is constructed based on considering combination of four cells (size of robot footprint) as nodes, providing a retrace path through uncovered cells, thus avoiding coverage overlaps. An improved version known as competitive-STC algorithm was presented by the authors in Gabriely and Rimon (2003), where the resolution of cells is reduced to that of the robot footprint, while compromising on coverage overlap. However, in STC based algorithms, partially occupied cells are considered as occupied cells and hence are left uncovered. Thus, when an approximate cellular decomposition based algorithm covers all the free cells, it is only *resolution complete*. Though the algorithm presented in Gonzalez et al. (2005) attempts to cover even the partially occupied cells, by its design it cannot guarantee no retracing of the path.

In this paper, we propose a new coverage path planning algorithm, namely, *Pseudo spanning Tree-based Coverage* (**PTC**) algorithm, to provide a truly complete coverage by introducing a concept of *virtual node* corresponding to partially occupied cells and creating a *virtual edge* which ensures coverage of even the partially occupied cells without (or with minimal) retracing the path. Though we do not provide formal proofs of the completeness and competitiveness at this time, we demonstrate the proposed algorithm using illustrative examples.

2. THE PROBLEM SETTING

We consider a robot equipped with a coverage tool having a square of sides D as its footprint. The region of interest Q, a topologically connected subset of $\mathbb{R}^2$, is enclosed within a rectangular area Q_R of size $2mD \times 2nD$, where m, n are integers. The enclosing rectangle is divided into $m \times n$ square cells, called major cells, of sides $2D$. There are finite number of obstacles $O_1, O_2, \ldots, O_N$ in Q, with $O = \{O_1, O_2, \ldots, O_N\}$. Further, major cells are divided into four $D \times D$ sized sub-cells. The space beyond the region to be covered, that is, $Q_R \setminus Q$ is considered to be obstacle. A sub-cell may be completely

10.1016/j.ifacol.2016.03.052

free, completely occupied by obstacles, or partially occupied by obstacle. Now we shall look at a few useful definitions [1].

Definition 1. If a region Q is divided into cells, and a robot path passes through all completely free cells and not through partially free cells, then such a coverage path is said to be **resolution-complete**.

Definition 2. If a region Q is divided into cells, and a robot path passes through all cells, completely or partially free, such that the coverage tool covers entire $Q \setminus O$, then such coverage path is said to be **truly-complete**.

Resolution-complete coverage converges to truly-complete coverage in the limit as the cell size tends to zero. However, in approximate cellular decomposition based coverage algorithms, the size of the cells are same as that of the robot (or coverage tool) footprint.

Definition 3. A coverage path planning algorithm is said to be **competitive** [2] if it avoids "unnecessary" coverage overlap, while achieving a given coverage objective.

Note that even the optimal solution achieving a truly complete coverage (that is, 100% coverage), may not achieve non-overlapping coverage (that is, 0% overlap). A competitive, truly complete coverage algorithm minimizes the coverage overlap (that is, $\geq$ 0% overlap) while maximizing the coverage (that is, achieving 100% coverage).

Problem statement The problem addressed in this paper is of finding a truly complete, competitive coverage path for a robot, that is, a path which will pass through all partially or fully free subcells without unnecessary (or avoidable) retracing of paths, thus ensuring minimal possible overlap, so that the area $Q \setminus O$ is completely covered in true sense by the coverage tool.

3. NODES AND ADJACENCY GRAPH

Here we discuss the nodes and corresponding adjacency graphs, which forms the basis for the proposed algorithm.

3.1 Major nodes, sub nodes, and virtual nodes

Figure 1 shows a single major cell with the subcells embedded in it after the decomposition of the area. The major cell is shown by a square with dark boundary of size $2D \times 2D$. The dark circle at the center is a major node corresponding to it. The sub cell is indicated by the dotted square of size D and the black diamond is the corresponding sub-node. The concepts of major and sub cells and the corresponding nodes were introduced in Gabriely and Rimon (2001). Now we introduce a new concept called *virtual major nodes.* Node corresponding to a major cell which is partially occupied by obstacles is called a *virtual major node* or simply a *virtual node.* In contrast, a major node corresponding to a completely free major cell is called a *real major node*, or simply a *real node.*

[1] These definitions are provided in Guruprasad and Ranjitha (2015). We provide them here for brevity.

[2] This definition of competitiveness is stronger than the standard definition of "competitiveness" and weaker than "optimality"

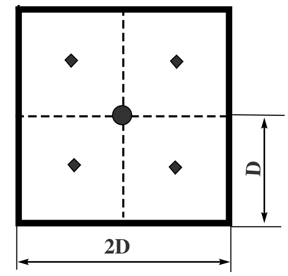

Fig. 1. The cell within the solid boundary ($2D \times 2D$) is a major cell and the black circle is the corresponding major-node. The cells within dotted boundary ($D \times D$) indicate the sub cells with the black diamonds as the corresponding sub-nodes.

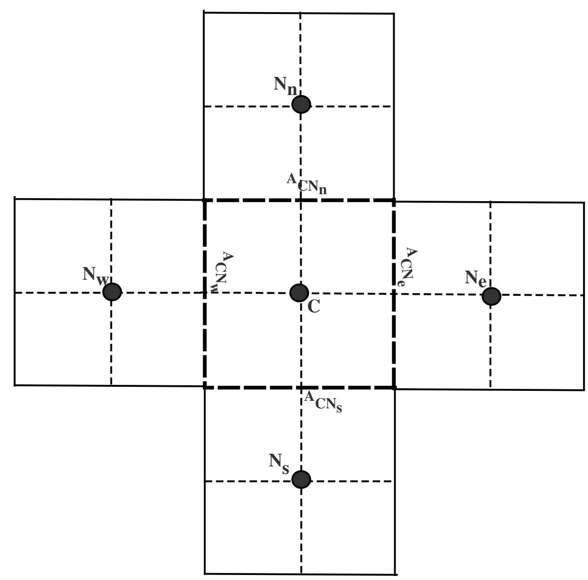

Fig. 2. Illustration of adjacent major nodes and the corresponding major cell boundaries.

3.2 The adjacency graph

Now we discuss the adjacency graph over the set of major/virtual nodes. Two major-nodes (virtual or real) are adjacent if they share a common cell boundary. Unlike in STC algorithm Gabriely and Rimon (2001), the common cell boundary plays an important role in defining the adjacency graph and also construction of the spanning tree in the proposed PTC algorithm. Figure 2 illustrates neighboring major cells and Table 1 lists the major-nodes and the associated common cell boundaries. Two adjacent Major cells, say C and N_e, are

Minor nodes	Corresponding major cell boundary
C and N_w	A_{CNw}
C and N_e	A_{CNe}
C and N_n	A_{CNn}
C and N_s	A_{CNs}

Table 1. Adjacent major nodes corresponding major-cell boundaries for the example shown in Figure 2

connected by an edge in the adjacency graph, if the robot