COMPETITIVE ON-LINE COVERAGE OF GRID ENVIRONMENTS BY A MOBILE ROBOT

Yoav Gabriely & Elon Rimon

Appeared in:

***Computational Geometry: Theory and Applications*, 24, 2003.**

Presented by:

Zachary L. Wilson

INTRODUCTION

- Mobile robot covering problem
  - Coverage done by tool, not robot itself
- On-line algorithm
- Traveling Salesman Problem and NP hardness
- 2 algorithms
  - Spiral-STC
  - Scan-STC

Preliminary Spiral-STC

- Most basic form of STC
- Work area is approximated using a grid of cells, each of size 2D (where D is the size of the tool used to cover a cell).
- Cells outside the area or obscured by obstacles are discarded
- Algorithm requires access to a position-and-orientation sensor and a range sensor.
- Constructs a spanning tree to follow as a coverage path
- Subcells of size D used for motion to avoid repeat coverage.

PRELIMINARY SPIRAL-STC CONT.

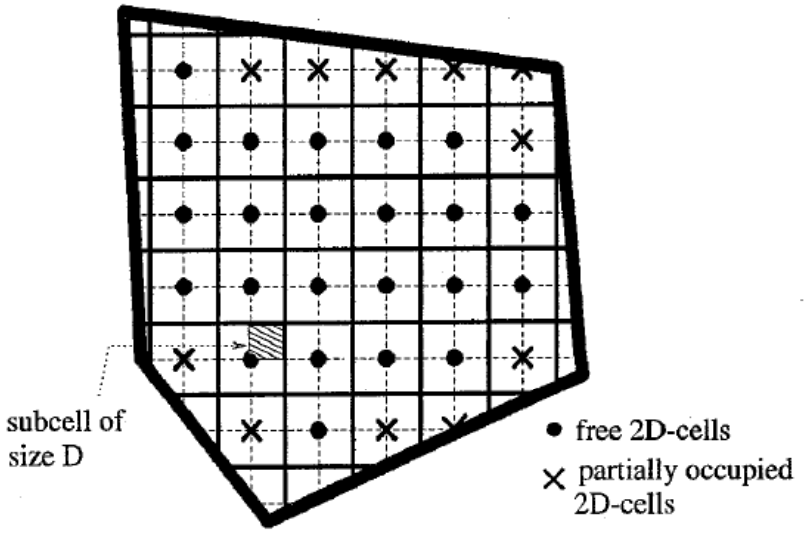

Fig. 1. Grid approximation of a given work-area.

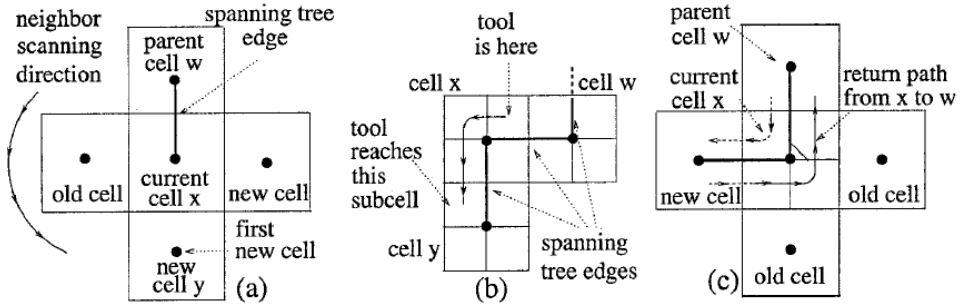

Fig. 2. (a) Counterclockwise scanning of four neighbors. (b) A move from x to a new cell y. (c) A return from x to a parent cell w.

Preliminary Spiral-STC cont.

- Spiral-STC is a recursive algorithm that:
  - Scans neighboring cells counterclockwise from the current cell's parent and while a neighboring unblocked, uncovered cell exists:
    - Generates a spanning-tree edge between the current cell and said neighbor,
    - Moves to the neighbor,
    - Re-calls Spiral-STC with the current cell as the parent and the now-occupied neighbor as the current cell.
  - Then, once all such neighbors are exhausted, the algorithm moves the platform it controls back to an uncovered subcell of the current cell's parent, and returns.

FULL SPIRAL-STC

Full spiral STC is like the preliminary algorithm, but augments the latter algorithm by allowing coverage of partially occupied cells.

- Allows for more coverage of the environment
- Causes some overlapped coverage

Introduces two types of edges, and a new phenomenon:

- Edges:
  - Double-sided edge
  - Single-sided edge
- Node doubling

FULL SPIRAL-STC CONT.

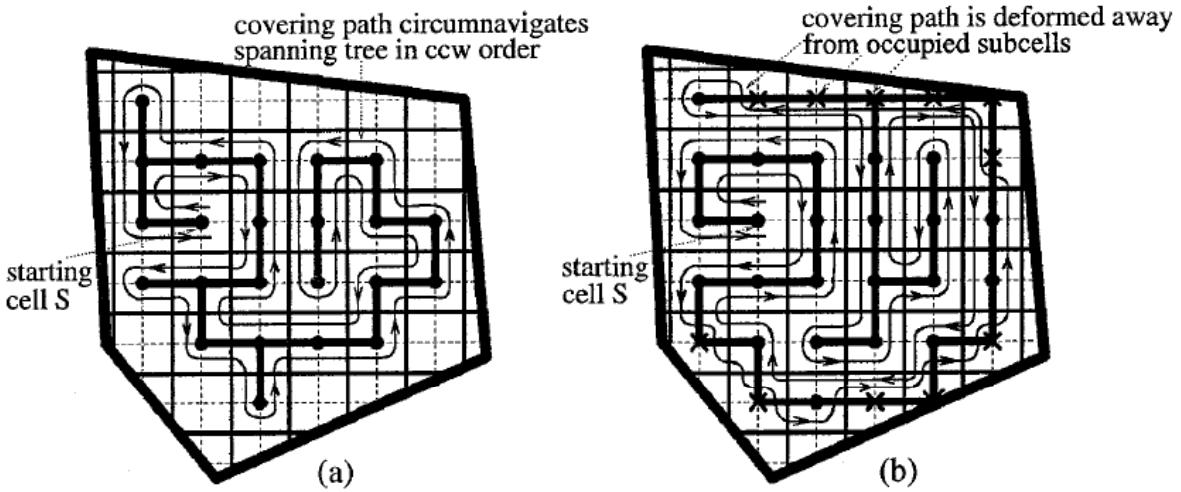

Fig. 3. An execution example of (a) *2D-Spiral-STC* and (b) the full *Spiral-STC*.

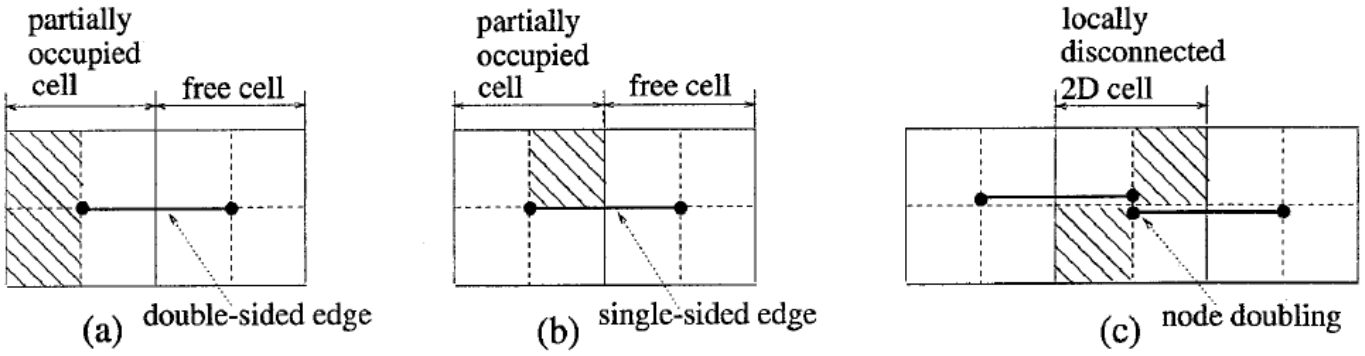

Fig. 4. (a) A double-sided and (b) a single-sided edge. (c) Node doubling at a disconnected cell.

FULL SPIRAL-STC CONT.

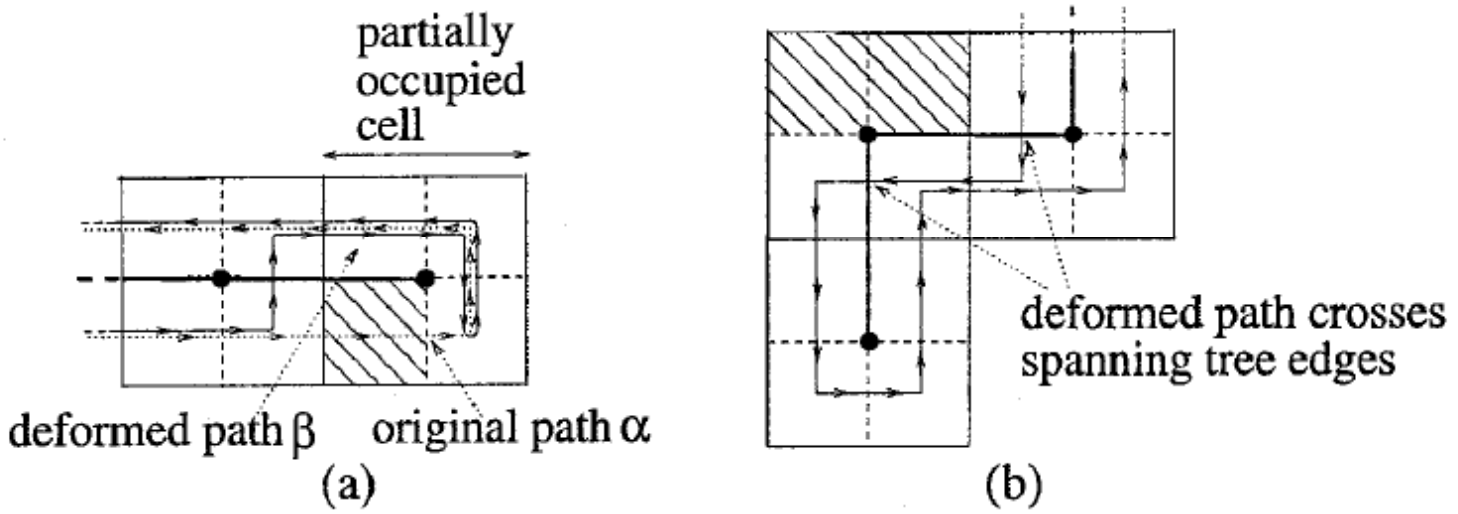

Fig. 5. (a) Path deformation along a single-sided edge. (b) Crossing of spanning-tree edges.

PRELIMINARY SCAN-STC

- Based upon Spiral-STC, but with added rules
  - If vertical scanning behavior is desired, algorithm will skip coverage of horizontal cells if the cells as shown below are unoccupied.

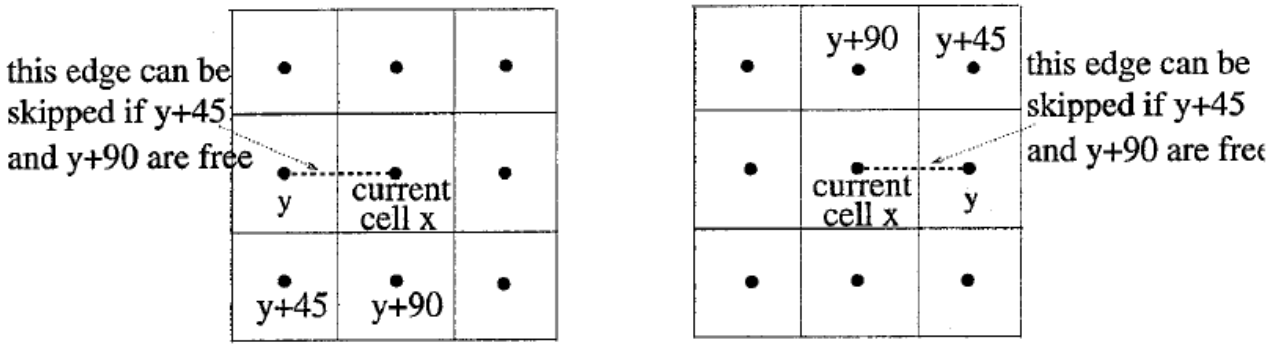

Fig. 6. The cells inspected by *2D-Scan-STC* when considering whether to skip the construction of a horizontal spanning-tree edge.

PRELIMINARY SCAN-STC CONT.

- Horizontal cells are only passed over if the robot is sure that:
  - It can continue onwards in its current vertical column.
  - It can access the next vertical column.
- This produces a more conventional "ladder-search" style pattern.

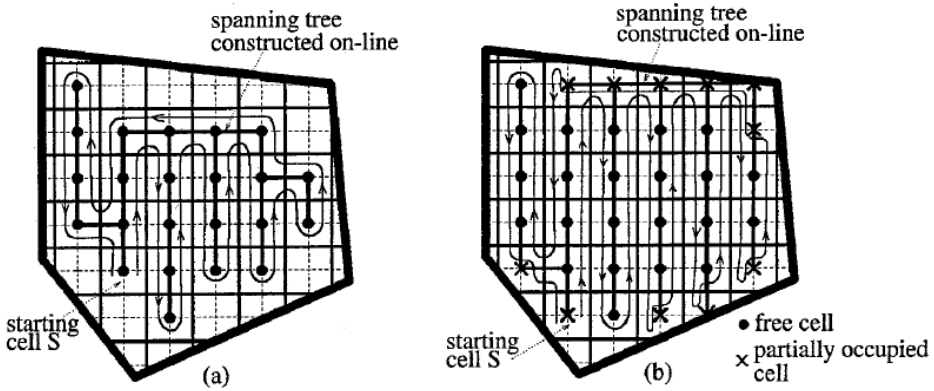

Fig. 7. An execution example of (a) *2D-Scan-STC* and (b) the full *Scan-STC*.

FULL SCAN-STC

- Provides the same advantages and disadvantages to the Scan-STC algorithm that the full Spiral-STC provides to the preliminary Spiral-STC.
  - Allows coverage of partial 2D cells.
  - Causes overlapping coverage.

Analysis

- Full Spiral-STC and Scan-STC algorithms cover every free accessible subcell
  - Algorithms construct a spanning tree that reaches every accessible free and partially occupied 2D-size cell
  - Deformed subcell path touches every unoccupied subcell

Analysis, cont.

- Coverage occurs using O(n) time and O(n) memory (n = total number of accessible free subcells)
  - m <= n is the number of boundary subcells
  - There are O(n) covering steps; every step involves entering a new cell or a repeat visit to a previous cell (bounded below)
  - Only O(n) spanning tree edges are stored in memory.

ANALYSIS, CONT.

- Path length <= (n+m) * D, where D is the tool size.
  - Repeated coverage is bounded by the number of boundary subcells, meaning that number of repetitive coverages is at most m.

RESULTS

- No rigorous measurement of average repeated coverage has been made; however:
  - Spiral-STC produced complete coverage with a path at most 18% longer than optimal in a given scenario.
  - Scan-STC produced complete coverage with a path at most 17% longer than optimal in the same scenario.

Conclusion

- Two full-coverage algorithms
  - Spiral-STC
  - Scan-STC
- Both algorithms operate in linear time with linear memory requirements.
- Issues:
  - Neither algorithm can guarantee at what time they will return to the starting point, only that they will do so. This can cause problems for localization methods which accumulate error.