Developing Topological Representations for Robotic Exploration of Known Environments

Alex Wallar Donald A. Sofge Daniela Rus

***Abstract*— test**

I. INTRODUCTION

In the context of mobile robotics, it is vital for a robot or a planner to understand the topological and metric connectivity of the environment it is operating in. In order to conduct higher level planning such as monitoring locations of interest or harvesting resources, a map of the environment is needed that is able to represent through which paths two areas in the environment are connected. There has been a lot of work in persistent coverage and autonomous surveillance which solve variations of the travelling salesman problem on graphical representations of an environment that assume knowledge of the paths between areas of interest in their search [1], [2], [3], [4] but do not provide a method of determining these paths. This work seeks to provide an simple graphical representation such that there are a low number of nodes in the representation in order for solutions to the travelling salesman problem and its variants are still feasible and a user of the presented algorithm will be able to simply look-up the path between areas of interest.

There has been a lot of work done in generating roadmaps of the environment (i.e. graphs that represent the connectivity of the environment). Kavraki et al developed the very popular probabilistic roadmap which randomly samples points in the configuration space of a robot and connects them in a graphical structure if it is feasible for the robot to move from one configuration two another avoiding collisions [5]. LaValle developed the rapidly-exploring random tree which incrementally develops a tree like structure of paths that would lead a robot from an initial to a goal configuration [6]. However, these approaches do not provide a concise graphical representation of the environment.

There has also been a lot of work in thinning free space to provide a more concise representation. Beenson et al developed a topological thinning algorithm using an Extended Voronoi Graph (EVG) that can show topologically distinct paths in free space [7]. However, the work lacks a concise graphical structure. Instead of providing a graph representing the connectivity, the work has provided a list of points which lie on the thinned representation of free space. Similarly, work has been done using morphology operations to produce a thin skeleton of the original volume [8], [9] but do not provide information about the connectivity of regions. Portugal and Rocha were able to use the work done by Beenson et al to determine the location of critical points that can be used as nodes in a graph [10], however this work did not provide a means of determining edges between the discovered nodes.

A. Wallar and D. Rus are with the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. D. Sofge is with the Naval Center for Applied Research in Artificial Intelligence at the Naval Research Laboratory

This paper presents an algorithm that is able to provide a graphical structure that represents both the topological and metric connectivity of the environment with a smaller number of nodes than work done previously by Portugal and Rocha and a simple look-up procedure to determine the path between nodes. Our method consists of three steps:

1) Produce a Voronoi graph of the environment using boundary points along the obstacles as centers of the Voronoi cells
2) Reduce the number of nodes by pruning redundant nodes that are not critical in defining the topological or metric connectivity of the environment
3) Prune the redundant leaf branches left over from the Voronoi graph

These steps are discussed in more detail in sections II, III, and IV respectively.

II. DEVELOPING THE INITIAL VORONOI DIAGRAM

We develop the initial graphical representation of the environment by determining points along the edge of all the obstacles in the environment. We use these points as centers of the Voronoi cells and provide them to the Voronoi diagram algorithm to determine the initial topological representation. This is the technique used by Choset and Burdick when developing generalized Voronoi graphs of occupancy grids [11]. Algo. 1 shows how these boundary points are chosen from the occupancy grid.

In order to extract a graphical structure from the Voronoi diagram, the points along the ridges are used as vertices. The vertices are connected to other points along the same ridge. Ridges that intersect obstacles and ridges totally contained by obstacles are discarded. This connected structure provides the first graphical representation of the environment.

III. PRUNING REDUNDANT NODES

In order to produce a graph with a relatively small number of vertices, we need to prune vertices from the graph generated by the Voronoi diagram that are not critical to defining the metric and topological structure of the environment. A critical node is defined as any node with one edge or with three or more edges. The initial pruning of redundant nodes is done by first determining which nodes in the original graph are critical and connecting them through a

Algorithm 1 BOUNDARYPOINTS($\mathcal{O}$)

Input:

- $\mathcal{O}$: $M \times N$ binary matrix representing obstacles in the environment, where if $\mathcal{O}_{i,j}$ is 1, (i, j) is inside an obstacle

Output:

- A set of pairs representing boundary points in the occupancy grid

$S \leftarrow \{\}$
for $i = 1$ **to** M **do**
  for $j = 1$ **to** N **do**
    if $\mathcal{O}_{i,j} = 1$ **then**
      for $k = -1$ **to** 1 **do**
        for $l = -1$ **to** 1 **do**
          if $\mathcal{O}_{i+k,j+l} == 0$ **then**
            $S \leftarrow S \cup \{(i, j)\}$
return S

Algorithm 2 RTVD($\mathcal{O}$)

Input:

- $\mathcal{O}$: $M \times N$ binary matrix representing obstacles in the environment

Output:

- A graph representing the topological connectivity of the environment, $\mathcal{O}$, with redundant nodes

$P \leftarrow$ BOUNDARYPOINTS($\mathcal{O}$)
$\mathcal{V} \leftarrow$ VORONOI(P)
$G = (V, E) \leftarrow$ DIGRAPH()
for $i \in$ CRITICALNODES($\mathcal{V}$) **do**
  for $j \in$ NEIGHBOURS(i) **do**
    $\Pi \leftarrow \{i\}$
    $S \leftarrow \{j\}$
    while $|S| > 0$ **do**
      $n \leftarrow$ POP(S)
      $\Pi \leftarrow \Pi \cup \{n\}$
      if $n \in$ CRITICALNODES($\mathcal{V}$) **then**
        $V \leftarrow V \cup i$
        $E \leftarrow E \cup (i, n, \Pi)$
        break
      else
        for $m \in$ NEIGHBOURS(n) **do**
          if $m \notin \Pi$ **then**
            $S \leftarrow S \cup \{m\}$
return H

series a of non-critical points. This chain of non-critical points will develop the path between the critical nodes. After pruning these nodes, a directed multi-graph structure, $G = (V, E, \Pi)$, which contains a set of critical nodes, a set of pairs representing the topological connectivity between nodes, and a set of sequences representing the geometric paths between the nodes is produced. Note that this graph structure is able to have multiple paths between the same pair of nodes.

To develop this initial reduced graph, we iterate through all of the critical nodes in the original graph determined by the Voronoi diagram. Then for each neighbour of each critical node, we determine if that node is also critical. If so, an edge is created and the iteratively stored path between them is added to the edge. If the neighbour is not a critical node, it gets added to path stemming from the critical node until another critical node is discovered. This process is shown in Algo. 2.

IV. PRUNING REDUNDANT LEAF BRANCHES

V. EXPERIMENTAL RESULTS

VI. CONCLUSION

REFERENCES

[1] J. Yu, M. Schwager, and D. Rus, “Correlated orienteering problem and it application to persistent monitoring tasks,” *arXiv preprint arXiv:1402.1896*, 2014.

[2] S. Sariel-Talay, T. R. Balch, and N. Erdogan, “Multiple traveling robot problem: A solution based on dynamic task selection and robust execution,” *Mechatronics, IEEE/ASME Transactions on*, vol. 14, no. 2, pp. 198–206, 2009.

[3] D. Mitchell, M. Corah, N. Chakraborty, K. Sycara, and N. Michael, “Multi-robot long-term persistent coverage with fuel constrained robots,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1093–1099.

[4] E. Stump and N. Michael, “Multi-robot persistent surveillance planning as a vehicle routing problem,” in *Automation Science and Engineering (CASE), 2011 IEEE Conference on*. IEEE, 2011, pp. 569–575.

[5] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566–580, Aug 1996.

[6] S. M. LaValle, “Rapidly-exploring random trees a new tool for path planning,” 1998.

[7] P. Beeson, N. K. Jong, and B. Kuipers, “Towards autonomous topological place detection using the extended voronoi graph,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 4373–4379.

[8] P. S. Heckbert, Ed., *Graphics Gems IV*. San Diego, CA, USA: Academic Press Professional, Inc., 1994.

[9] T.-C. Lee, R. L. Kashyap, and C.-N. Chu, “Building skeleton models via 3-d medial surface/axis thinning algorithms,” *CVGIP: Graph. Models Image Process.*, vol. 56, no. 6, pp. 462–478, Nov. 1994. [Online]. Available: http://dx.doi.org/10.1006/cgip.1994.1042

[10] D. Portugal and R. P. Rocha, “Extracting topological information from grid maps for robot navigation.” in *ICAART (1)*, 2012, pp. 137–143.

[11] H. Choset and J. Burdick, “Sensor-based exploration: The hierarchical generalized voronoi graph,” *The International Journal of Robotics Research*, vol. 19, no. 2, pp. 96–125, 2000.

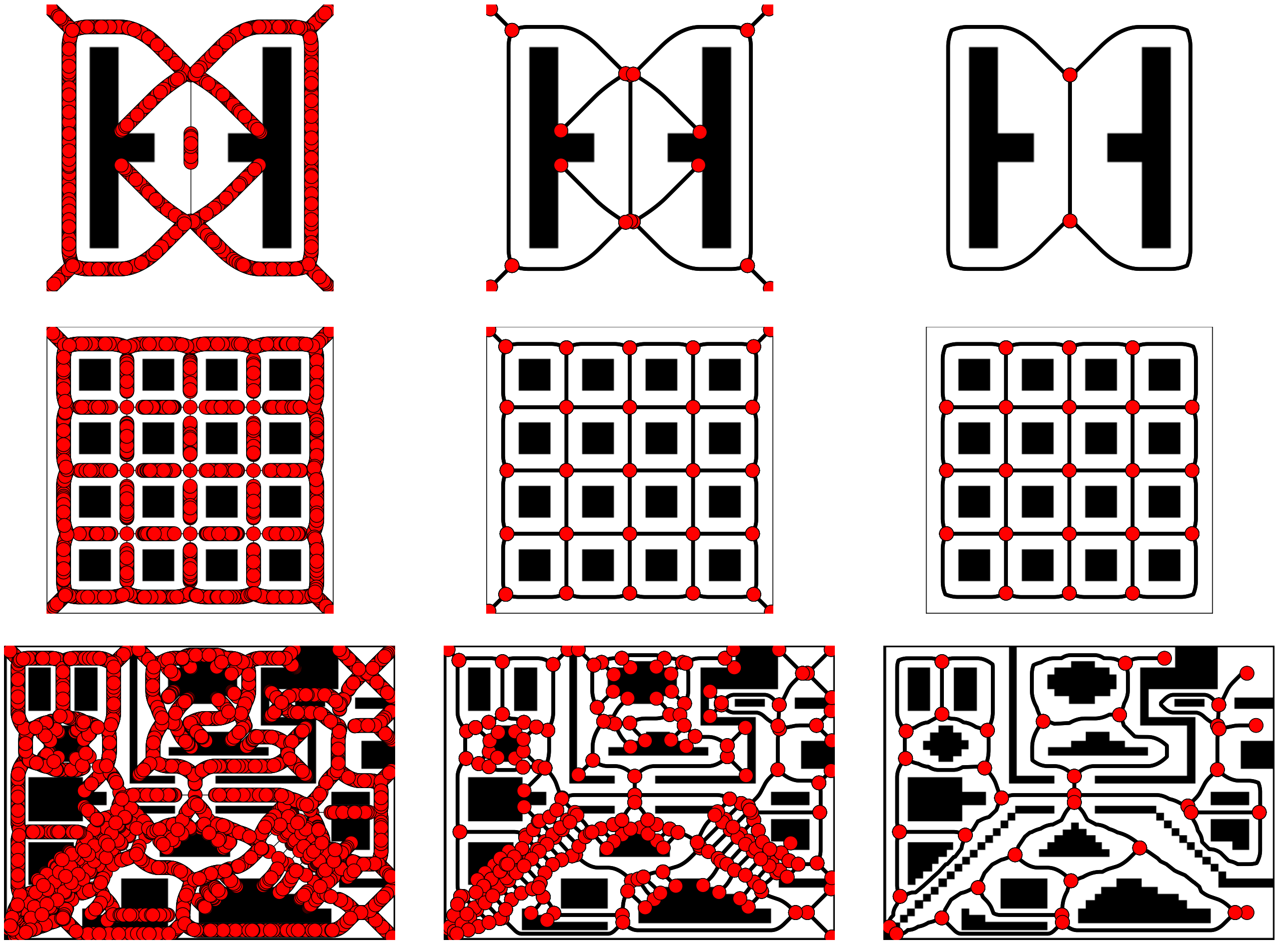

Fig. 1. The topological extraction algorithm running on each of the three scenes presented. The first column represents the graph generated from the initial Voronoi decomposition. The second column is a result of the first node reduction using RTVD and the last column is the final graph representing the connectivity of the environment generated by the second node reduction using TVD.

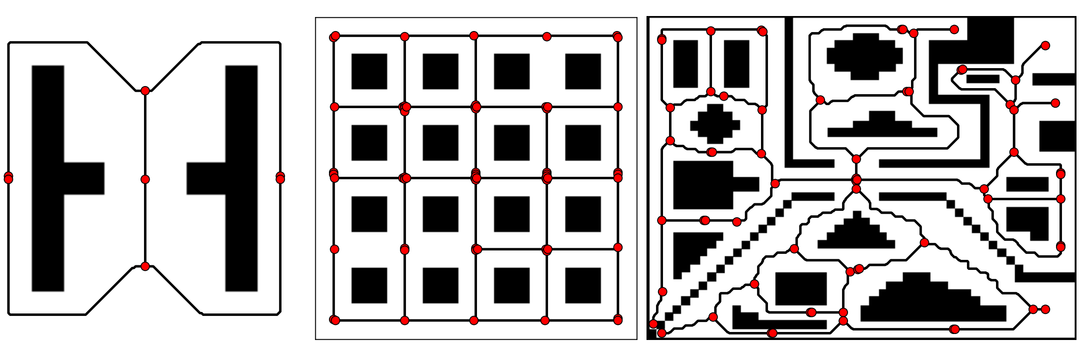

Fig. 2. The topological graph extracted using the algorithm by Portugal and Rocha shown in [10]

Algorithm 3 TVD($\mathcal{O}$)

Input:

- $\mathcal{O}$: $M \times N$ binary matrix representing obstacles in the environment

Output:

- A graph representing the topological connectivity of the environment, $\mathcal{O}$

$G = (V, E) \leftarrow$ MULTIDIGRAPH()
$H \leftarrow$ RTVD($\mathcal{O}$)
$H' \leftarrow$ REMOVECLOSENODES($H, \mathcal{O}$)
for $i \in$ CRITICALNODES(H') **do**
  for $j \in$ NEIGHBOURS(i) **do**
    $\Pi \leftarrow \{i\}$
    $S \leftarrow \{(i, j)\}$
    while $|S| > 0$ **do**
      $(n, m) \leftarrow$ POP(S)
      $\Pi \leftarrow \Pi \cup \Pi_{H'}(n, m)$
      if $n \in$ CRITICALNODES(H') **then**
        $V \leftarrow V \cup i$
        $E \leftarrow E \cup (i, n, \Pi)$
        break
      else
        for $k \in$ NEIGHBOURS(n) **do**
          if $k \notin \Pi$ **then**
            $S \leftarrow S \cup \{(n, k)\}$
return G

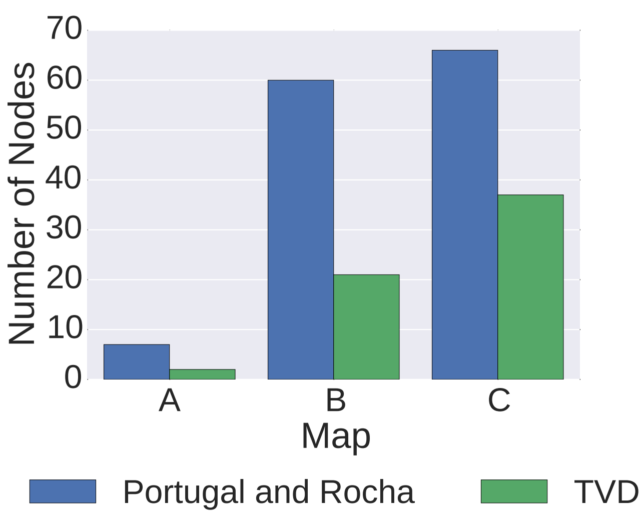

Fig. 3. Plot comparing the number of nodes generated using the proposed TVD algorithm and the algorithm developed in [10]