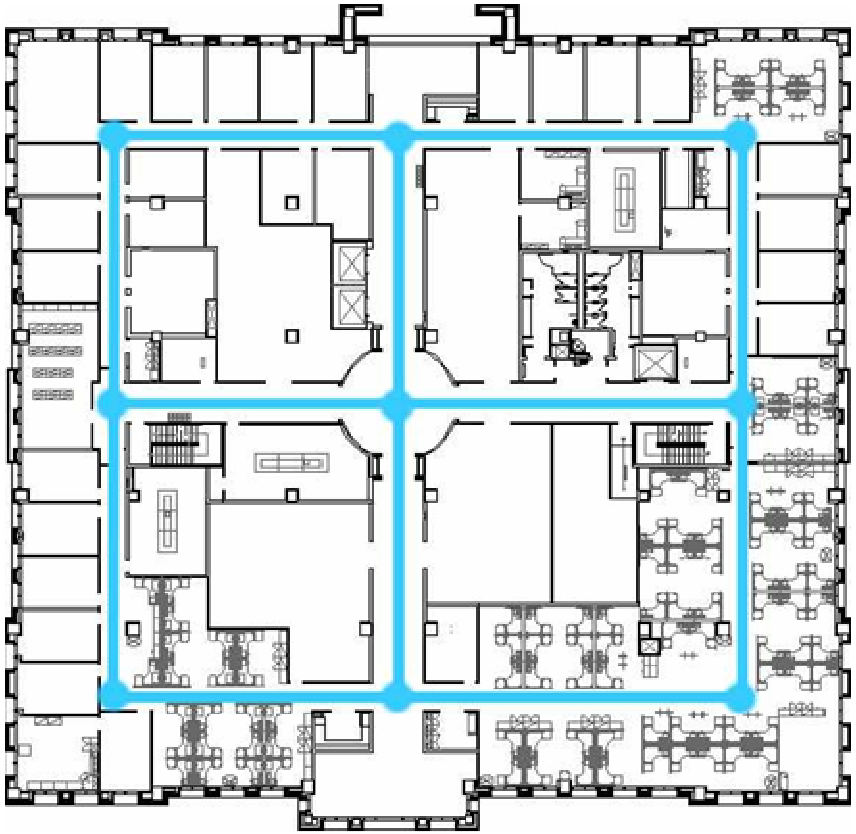
Towards Autonomous Topological Place Detection Using the Extended Voronoi Graph

Patrick Beeson, Nicholas K. Jong, and Benjamin Kuipers

Intelligent Robotics Lab
Department of Computer Sciences
The University of Texas at Austin

Problem Assumptions

- This work focuses on dead-ends, intersections, and doorways.
  - Smallest, atomic places sufficient for topological modeling.
  - Psychologically motivated

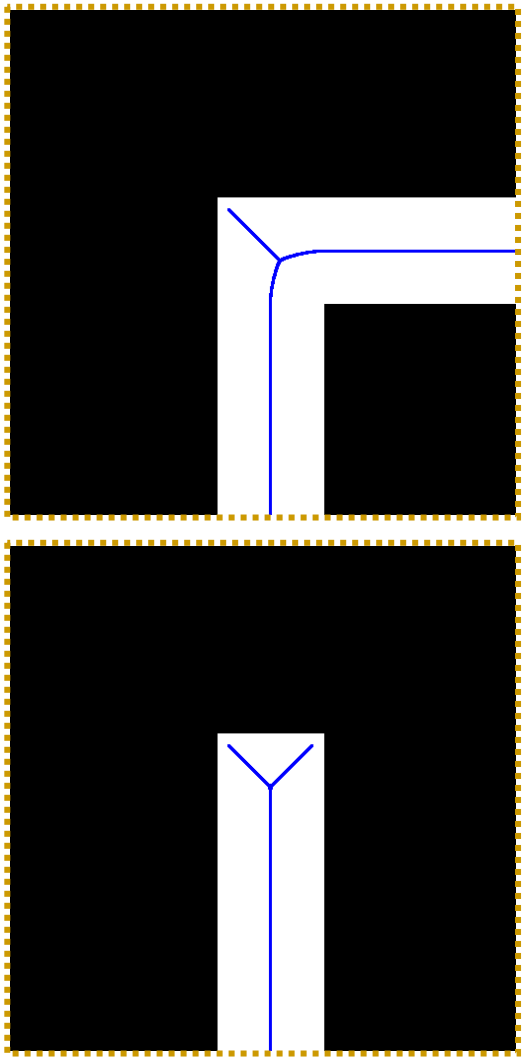


Previous Solutions

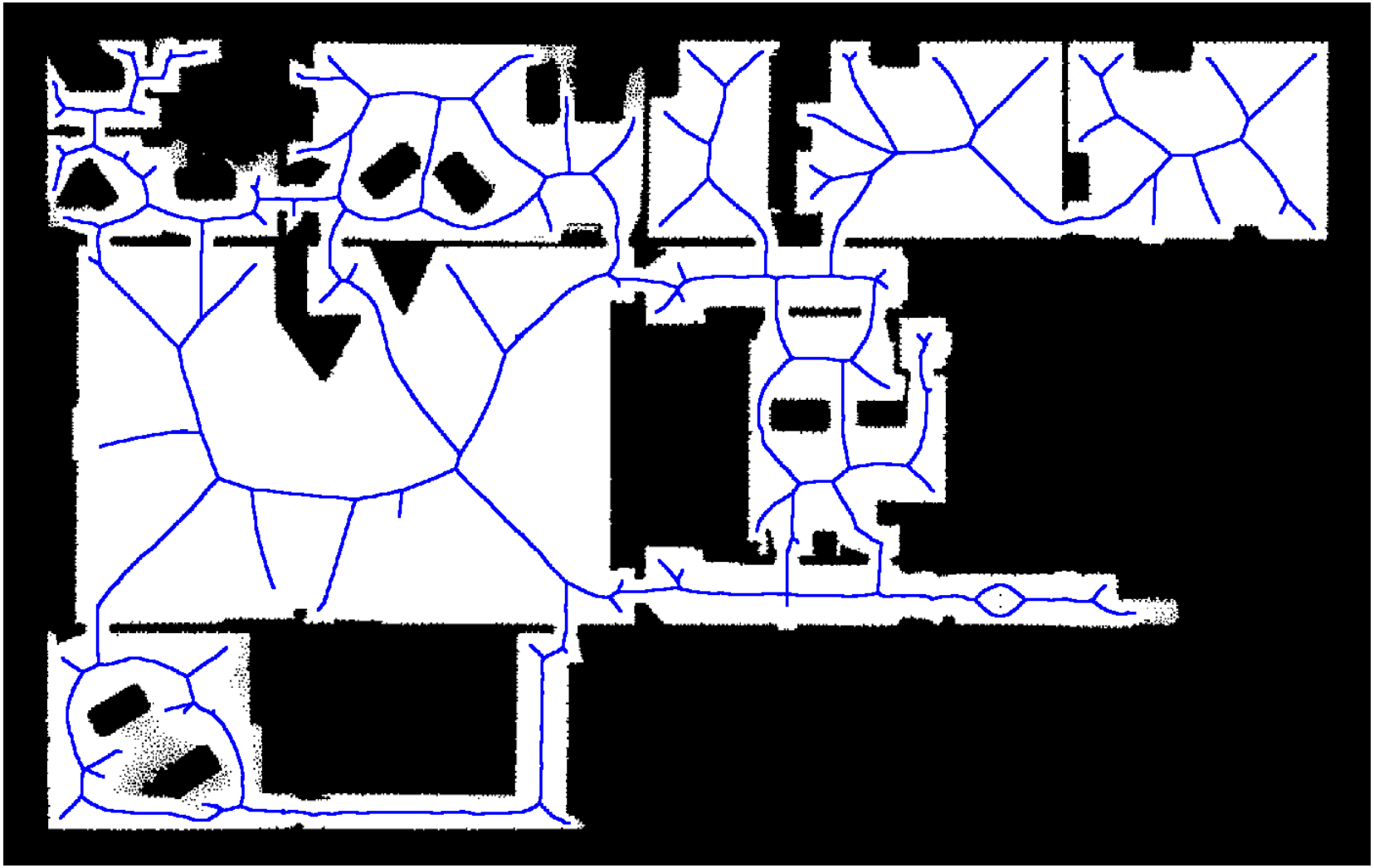
- Ad-hoc detection of simple intersections
  - Hand-coded
  - Environment dependent
  - Brittle
- Voronoi graph defines paths
  - Principled approach
    - Well-defined in enclosed environments
    - Work even in unconventional / non-perpendicular intersections
  - Experimentally validated in a number of environments
  - Choset and Nagatani, Trans. Robotics & Automation, 2001

Places at Voronoi Graph Junctions

- Voronoi graph – 1D set of points equidistant to the N (or more) closest obstacles in N dimensions.
- Usually:
  - Wheeled mobile robots
  - Planar range-sensors
  - 2D occupancy grid
- Places – branching points in the Voronoi graph
  - junctions



Voronoi Graph of a Global Metrical Map



Voronoi Problems

- Detecting Voronoi junctions has demonstrated some success for autonomous place detection.
- However, there are well known problems:
  - Unreliable
    - Spurious detections
    - Missed detections
    - Multiple places in complex intersections
  - Environment dependent
    - Only works in “enclosed” regions (at least N nearby obstacles)

Our Contributions

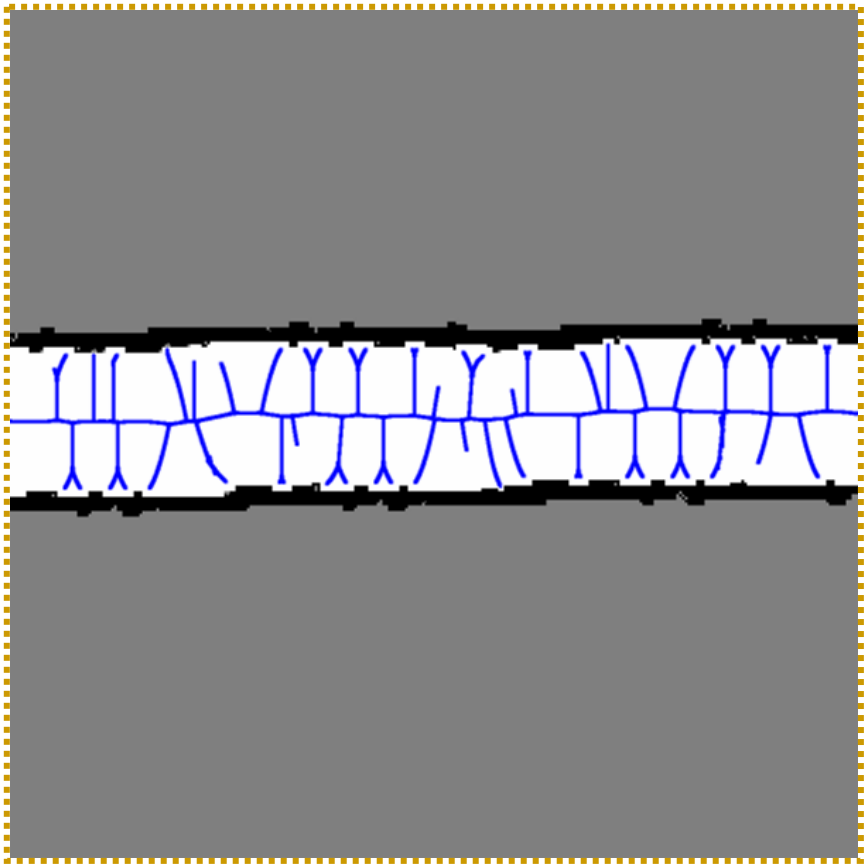
1. Provide a principled approach to pruning the Voronoi graph.
   - Reduces spurious Voronoi branches (i.e. paths)

2. Provide a simple extension to the standard Voronoi graph definition.
   - Allows non-enclosed (e.g. outdoor) environments

3. Provide a definition of intersections that is better than Voronoi junctions.
   - Reduces missed places
   - Aggregates regions at complex intersections into places

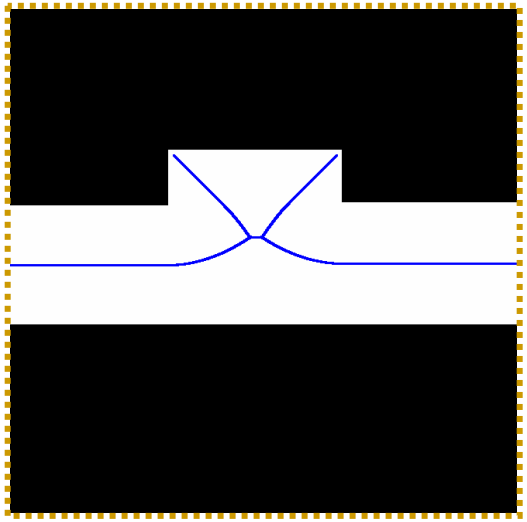
Pruning the Voronoi Graph

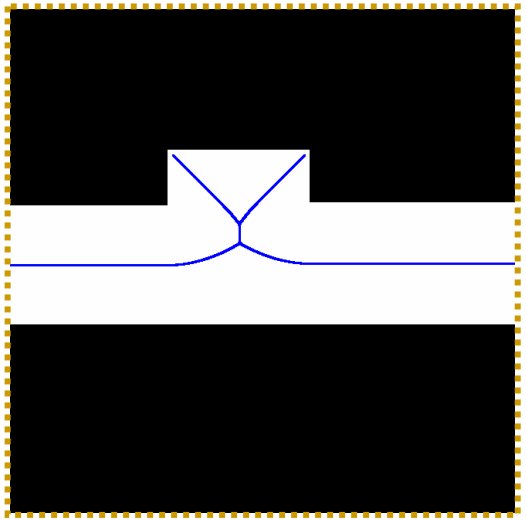
- Pruning is necessary to remove spurious branches.
- Choset's method
  - Remove terminal branches that only touch 2 obstacles.

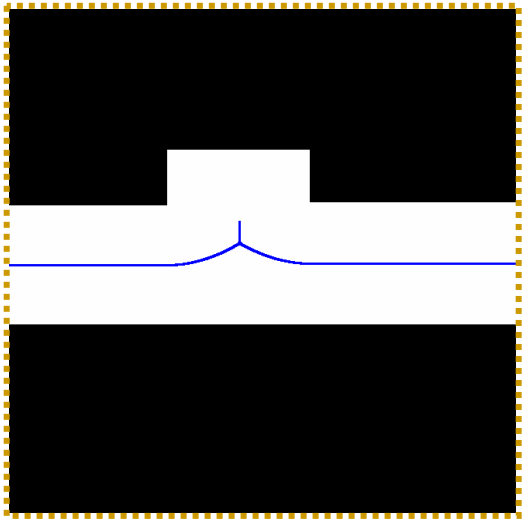


Pruning Failures

- Fixed-depth pruning fails for hierarchical branching.

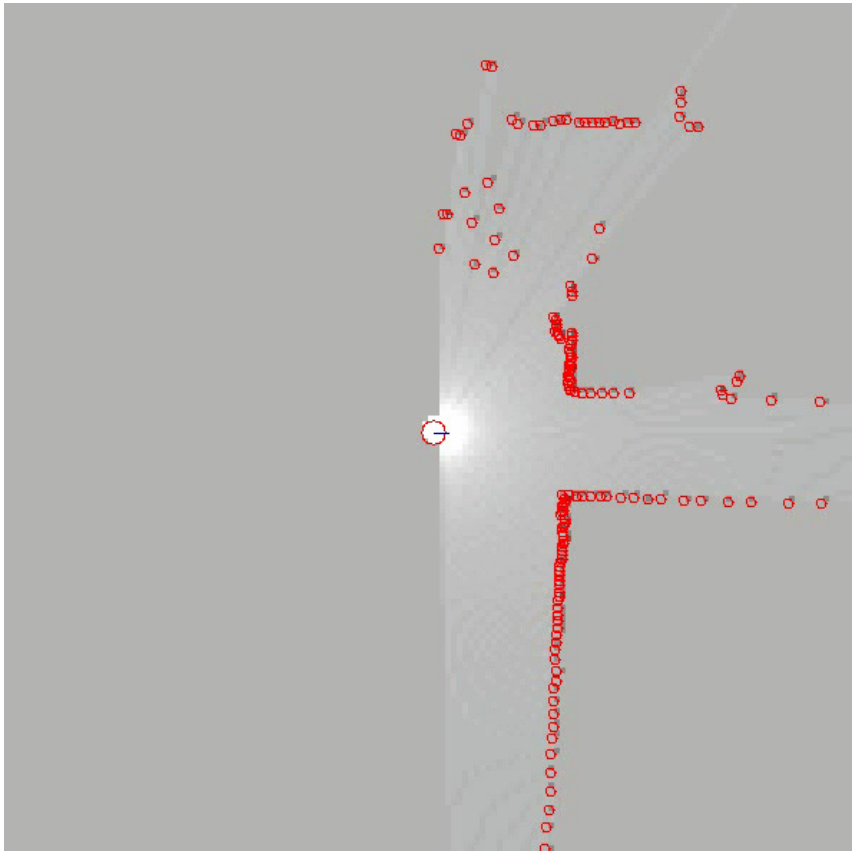






Our Solution to Pruning (Background)

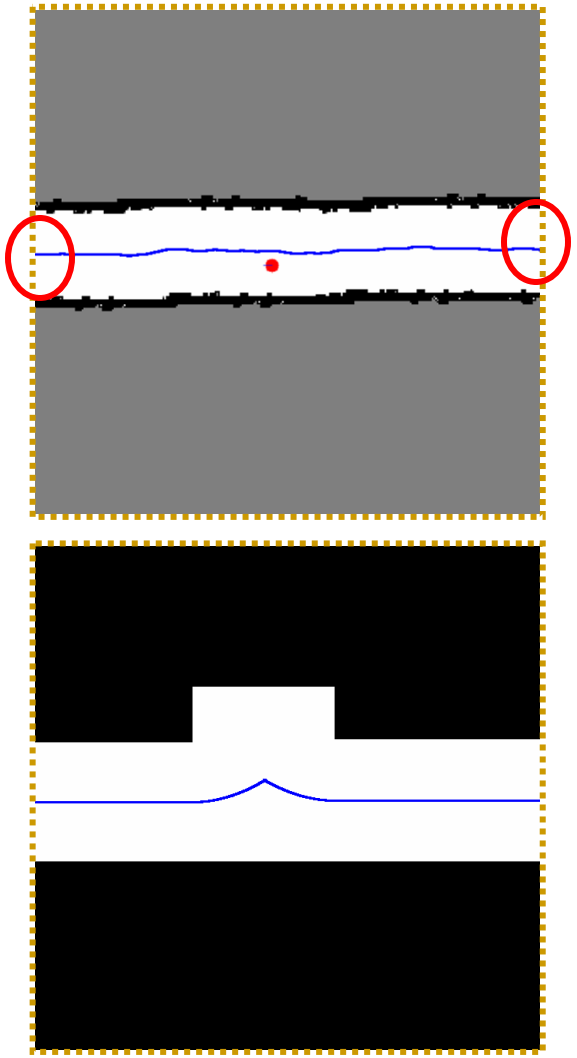
- There is always a fixed size, scrolling occupancy grid centered near the robot.
- (From previous work)
  - Small-scale space – local metrical model
  - Large-scale space – global topological model
  - If desired – global metrical model built using the topological skeleton



Video

Our Solution to Pruning

- Define exits as points in the Voronoi graph that:
  - touch the grid edge
  - touch “unknown” space (gray cells) in the grid
- Find the minimum spanning tree that connects the exits.
  - Dead-ends are a simple special case.
- The size of the scrolling, local occupancy grid will determine the size of relevant places and paths.



Our Contributions

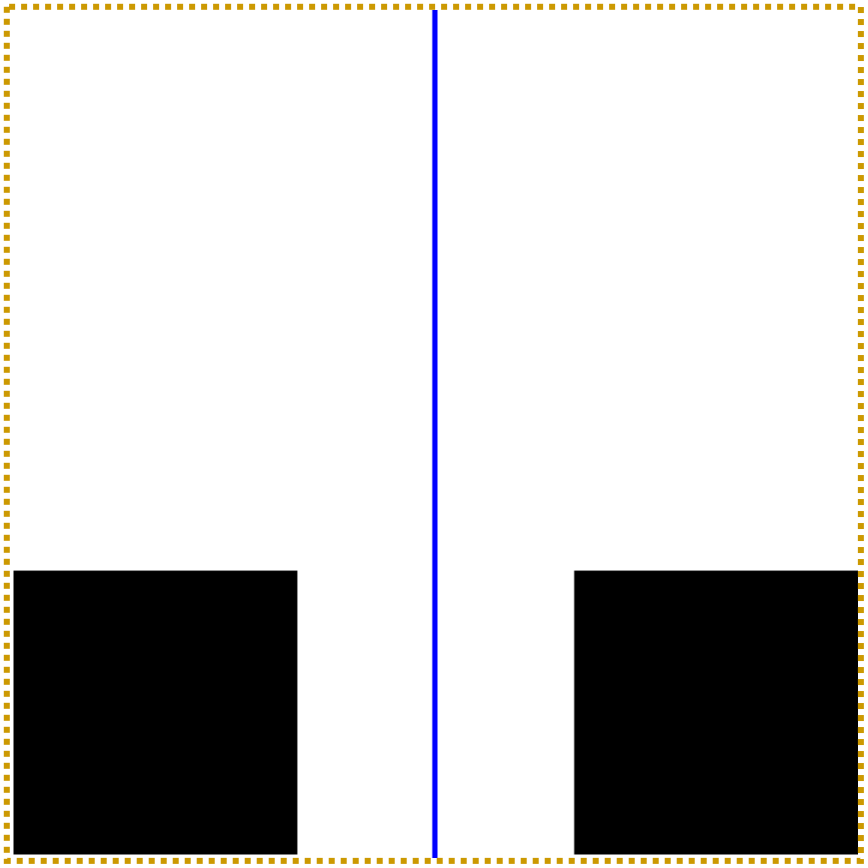
1. Provide a principled approach to pruning the Voronoi graph.
   - Reduces spurious Voronoi branches
2. Provide a simple extension to the standard Voronoi graph definition.
   - Allows non-enclosed (e.g. outdoor) environments
3. Provide a definition of intersections that is better than Voronoi junctions.
   - Reduces missed places
   - Aggregates regions at complex intersections into places

Extending the Voronoi Graph

- Voronoi graphs require N or more nearby obstacles.
  - In our case N=2.

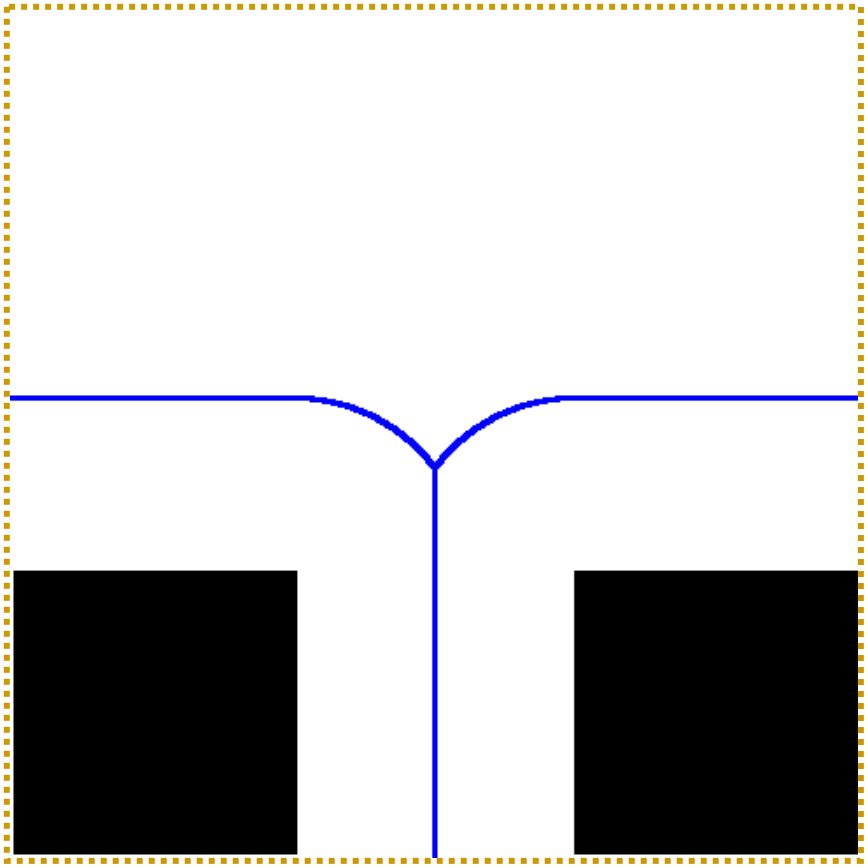
1. This limits the robot to exploring enclosed environments.

2. If following the Voronoi graph, the robot can become lost.

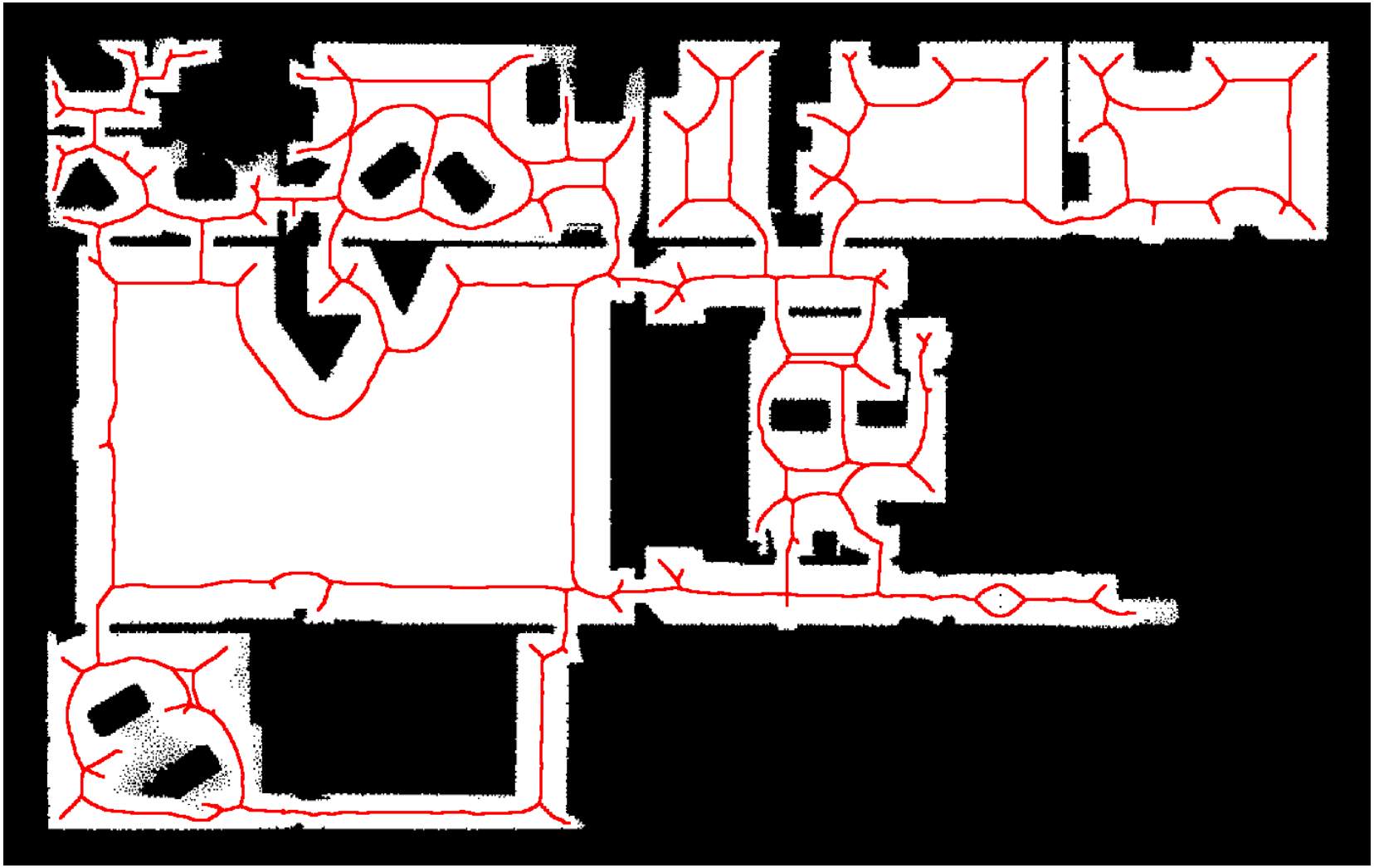


Extended Voronoi Graph

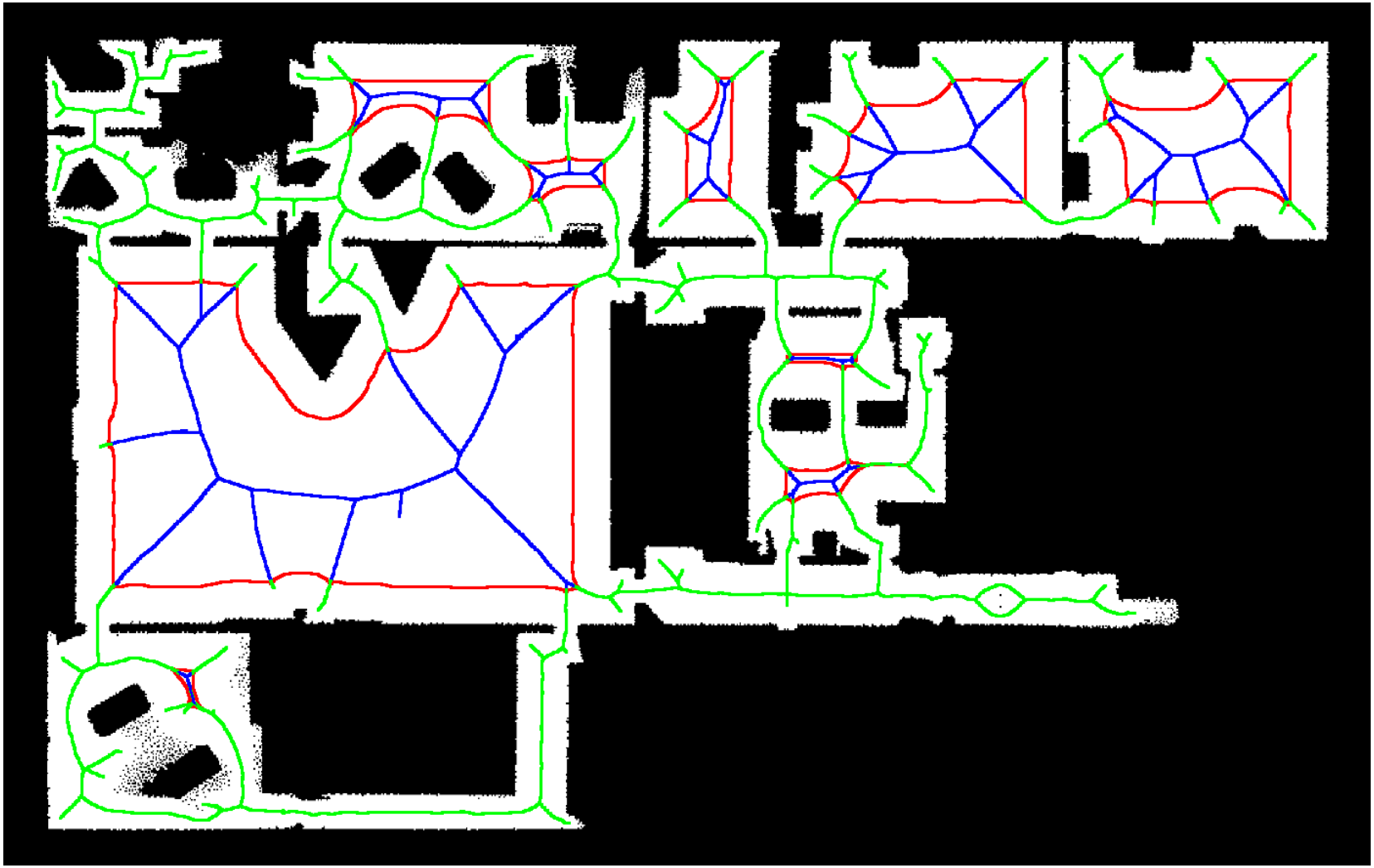
- UNION of:
- Subset of the regular Voronoi graph, where r < *d* < *S*.
  - *d* is distance to nearest obstacle.
  - r is based on the robot's physical body.
  - S is a maximum distance threshold
- The set of points where *d* = *S*.
  - Provides "coastal navigation" when necessary.



Extended Voronoi Graph



Voronoi Graph Comparison

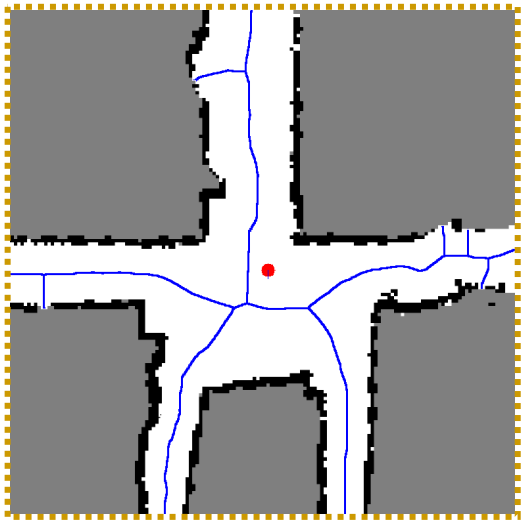


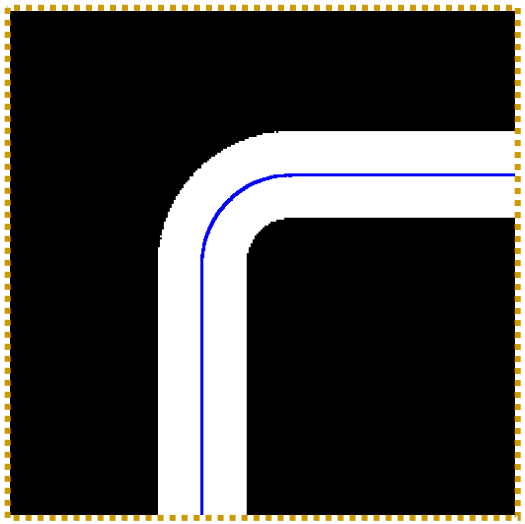
Our Contributions

1. Provide a principled approach to pruning the Voronoi graph.
   - Reduces spurious Voronoi branches
2. Provide a simple extension to the standard Voronoi graph definition.
   - Allows non-enclosed (e.g. outdoor) environments
3. Provide a definition of intersections that is better than Voronoi junctions.
   - Reduces missed places
   - Aggregates regions at complex intersections into places

Problems with Places at Junctions

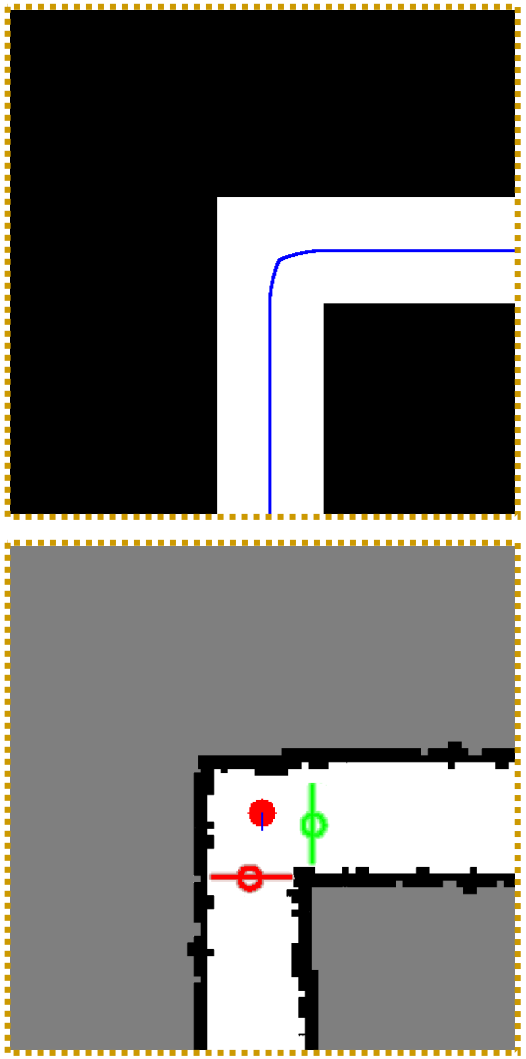
- Junctions are highly susceptible to environmental / sensor noise.
- Some places have multiple junctions.
- Some places do not have junctions.





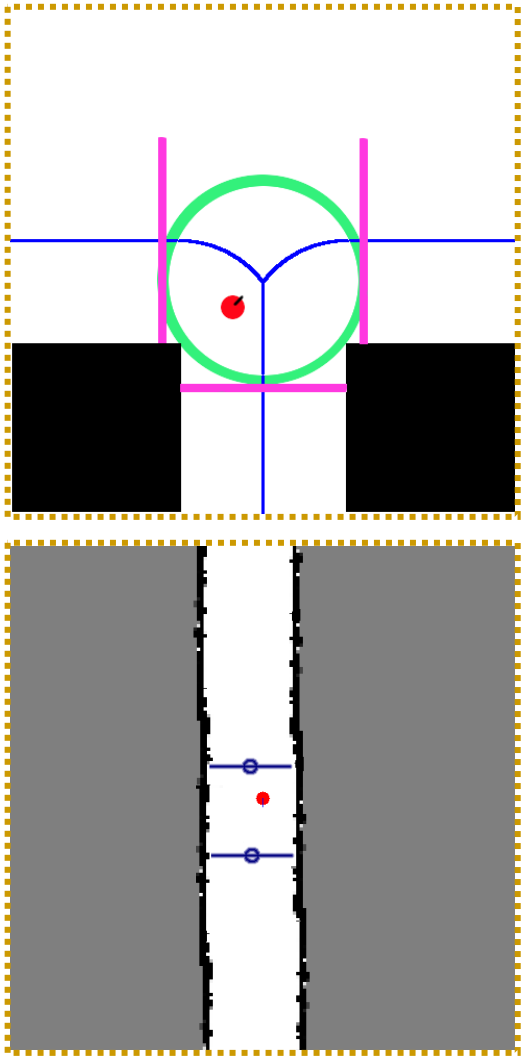
Problems with Paths as Branches

- Using branches to define paths results in multiple places in complex environments.
- Sometimes there are no places in simple environments.
- Our approach uses *gateways* to find paths in the local metrical model.

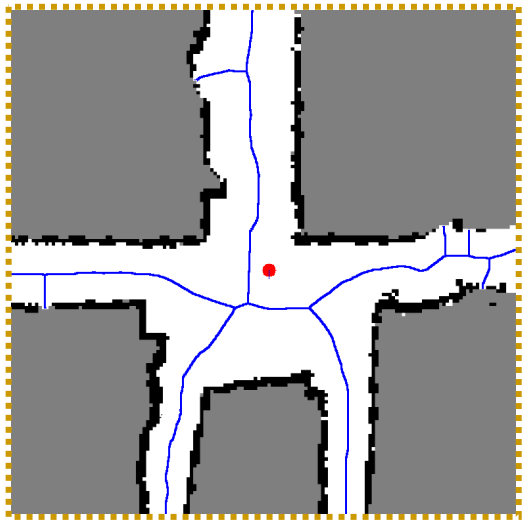


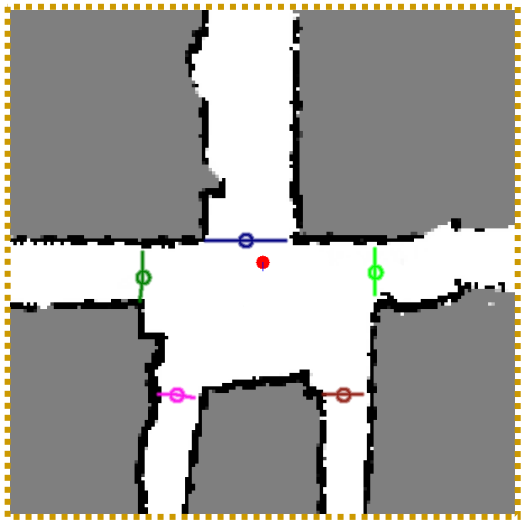
Finding Gateways

- "A gateway occurs where there is at least a partial visual separation between two neighboring areas and the gateway itself is a visual opening to a previously obscured area."
  - Chown et al., "PLAN" paper, 1995
- In previous work, we used gateways to determine the local topology at places.
- Multiple valid gateway definitions exist.
  - See paper for our formal definition.
- Gateways can also be determined at non-places.



Gateways in Complex Places

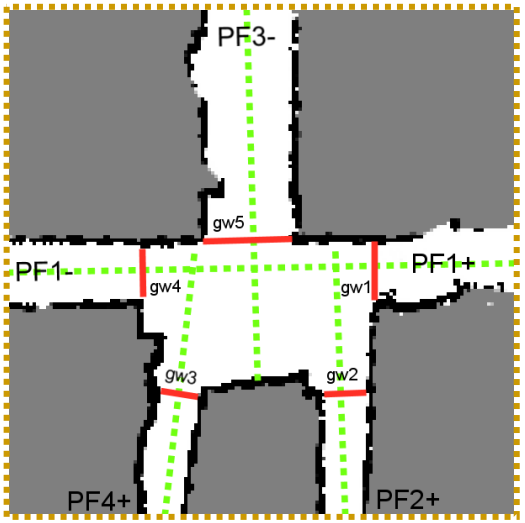


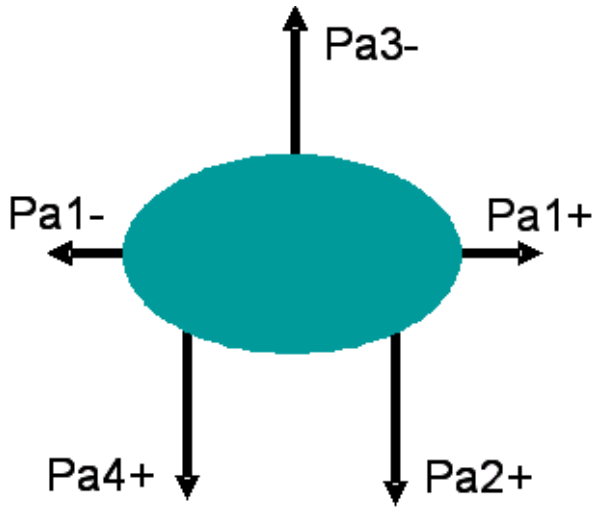


- Our gateway algorithm even works in complex intersections.

Defining Paths

- Each gateway is associated with exactly 1 path.
- If exactly 2 gateways are unambiguously aligned, they belong to the same path.
  - Path passes through the place.
- Otherwise, that path terminates at the place.





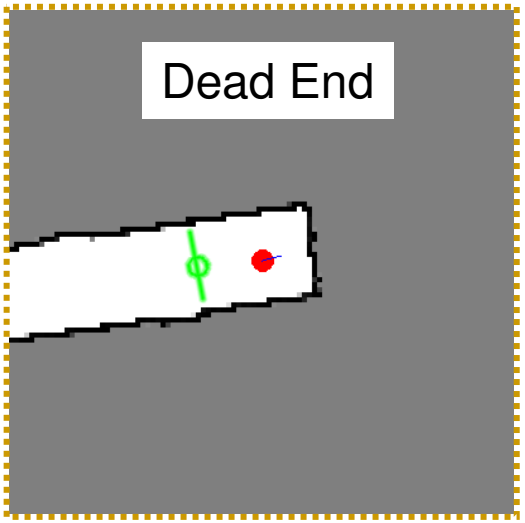
Detecting Places

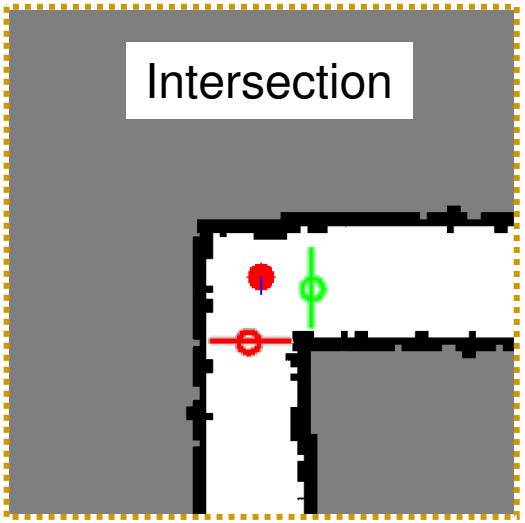
- The robot IS NOT at a place when:
  - Exactly 1 path

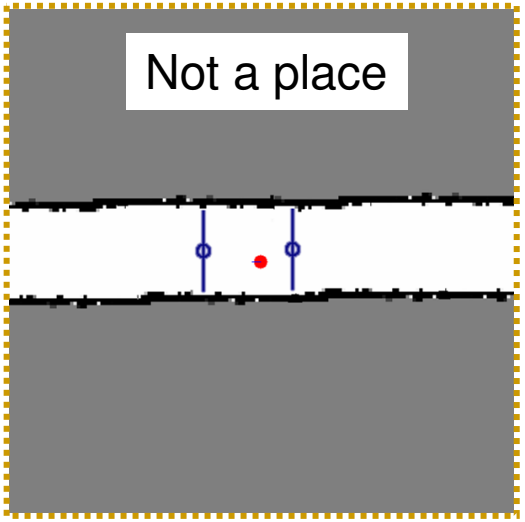
  AND

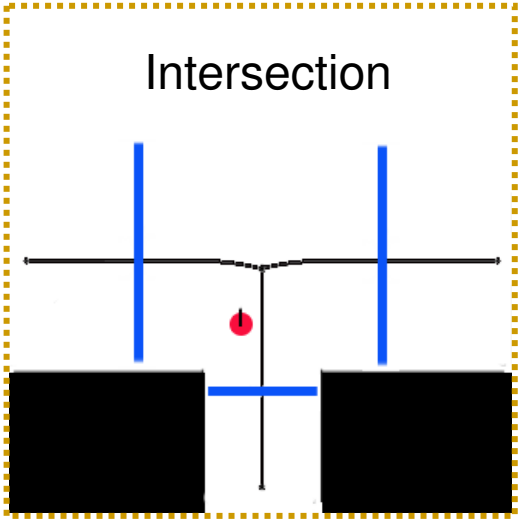
  - Exactly 2 gateways
- Otherwise, the robot IS at a place.
  - enclosed places
    - Number of paths = 0
  - dead-ends
    - Number of gateways = 1
  - intersections
    - Number of paths > 1

Place Detection Examples







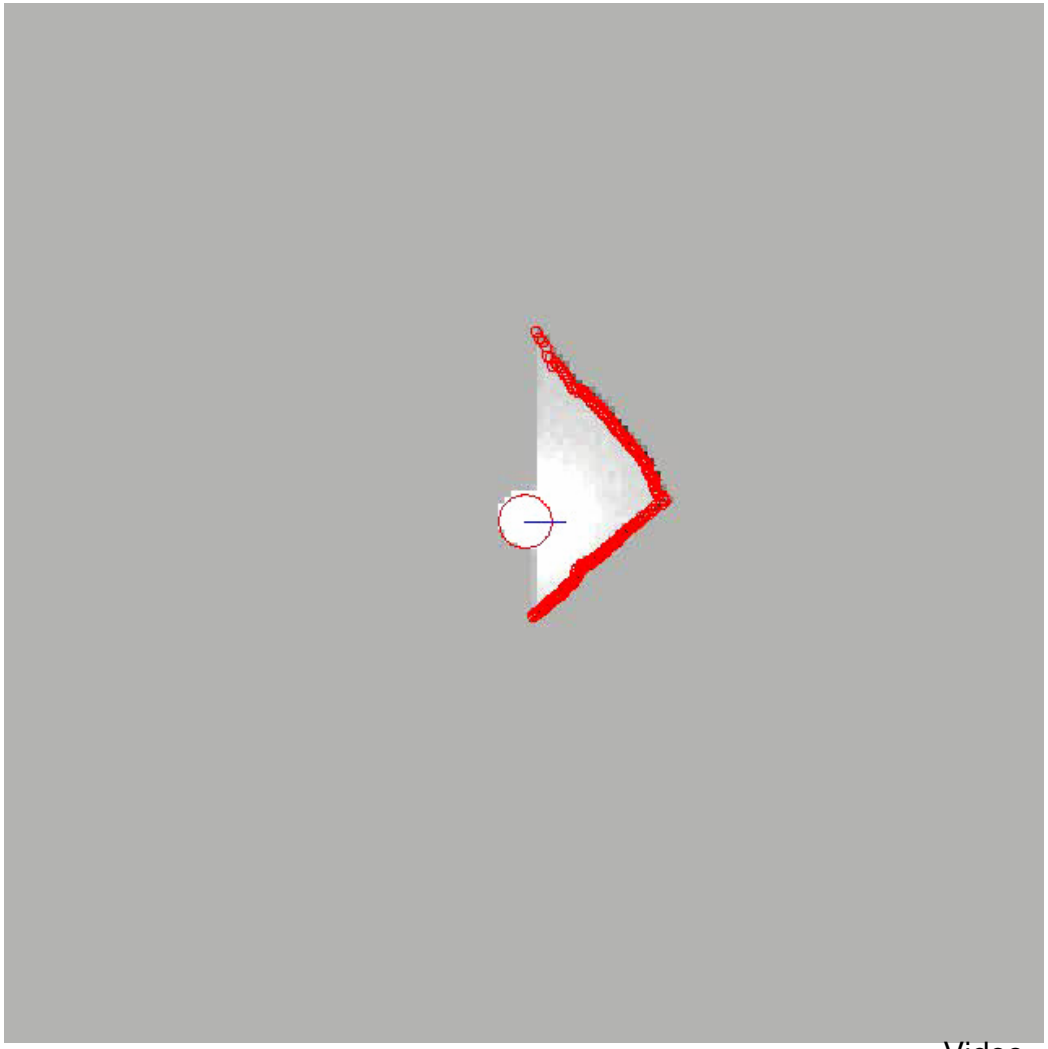


Conclusion

- Our solution achieves reliable autonomous topological place detection (finding intersections, dead-ends, and open doors).
- Our method handles more environments.
  - Extended Voronoi graph handles both enclosed and non-enclosed environments.
- Previous false positive and false negative place detections are overcome.
  - Better pruning technique
  - Define places using gateways and paths not Voronoi junctions
  - Using our gateway algorithm, a robot can even detect complex intersections as single places with non-trivial local topologies.

Thank You

http://www.cs.utexas.edu/~robot



Video