Boundary Recognition in Sensor Networks by Topological Methods

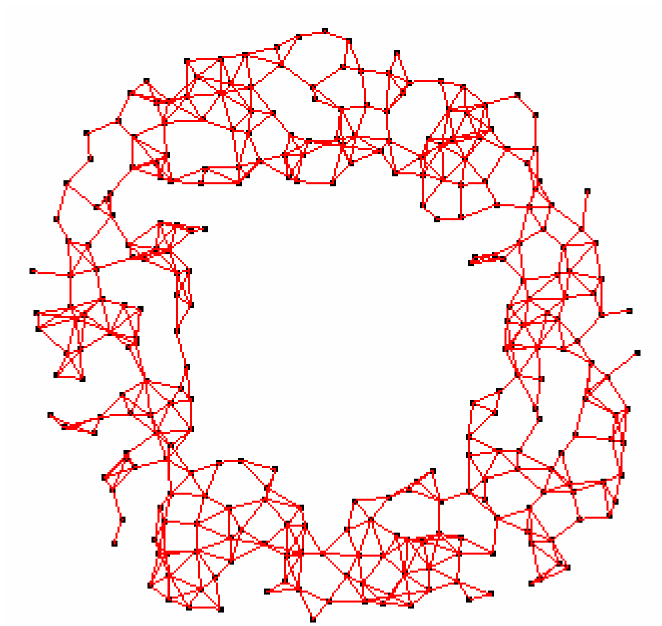
Yue Wang, Jie Gao, Joseph S.B. Mitchell
Stony Brook University

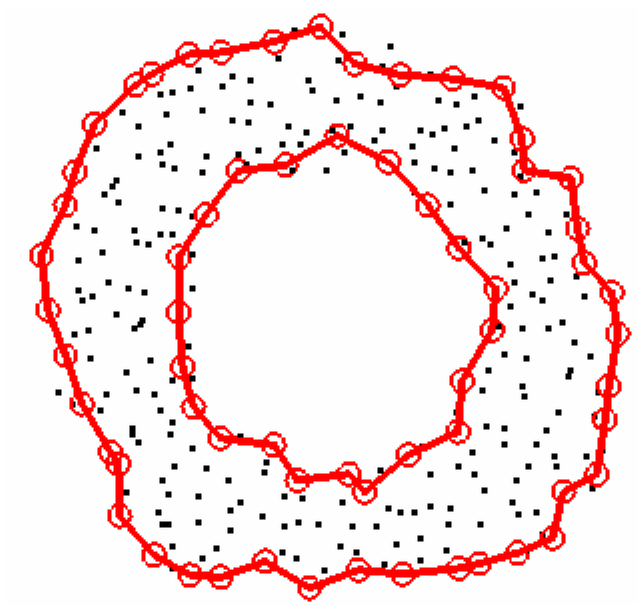
MobiCom 2006
Sep. 26 - 28, Los Angeles, USA

The Problem: Boundary Detection in Sensor Networks

- Input: Communication graph
  - no knowledge of the node locations
  - not require the unit disk graph model

• Output: The boundary nodes
- outer boundary and inner boundaries (holes)
- connects them into meaningful boundary cycles





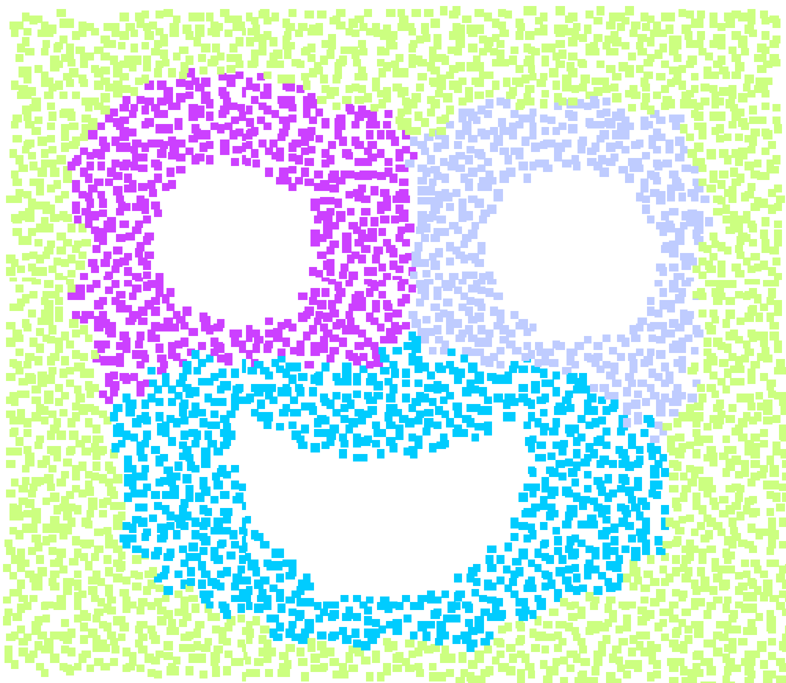
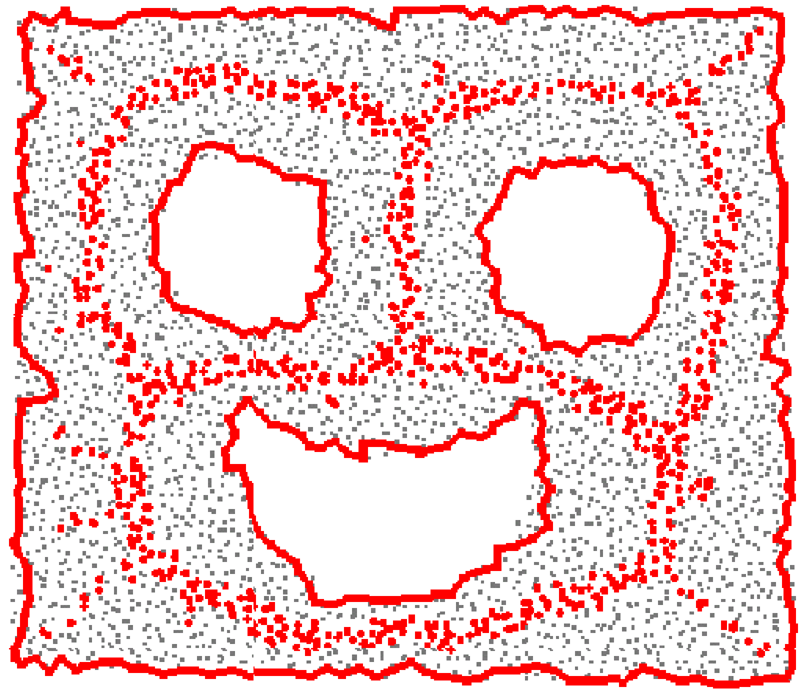
Applications of boundary detection I

- Monitor the physical environment
  - Sensors with readings > a threshold considered "dead".
  - Find the contour of abnormal chemical contamination, or overheated sensors.
- Indicate the health of sensor network
  - Detect the breakdown of network
  - Insufficient coverage or connectivity

Applications of boundary detection II

- Aid sensor deployment
  - Guarantee the newly added sensors in the expected region
- Help to design topology-adaptive network protocols
  - Virtual coordinates for routing (e.g., medial axis).

Quasi-Voronoi diagram and the medial axis diagram

Medial axis can be used to construct virtual coordinates for efficient point-to-point routing.

Related Work

- Geometric methods [FGG06]
  - Assume nodes know their locations.
- Statistical methods [FKPFB04][FKKL05]
  - Assume a probabilistic distribution of sensor deployment, e.g., Poisson distribution.
  - Typically requires a high node density.
- Topological methods [GM05][KFPF06][F05]
  - Use connectivity only.
  - [GM05] homology centralized approach.
  - [KFPF06] Quasi-Unit disk graph model.
  - [F05] Discover the "breaking" of flooding contours.

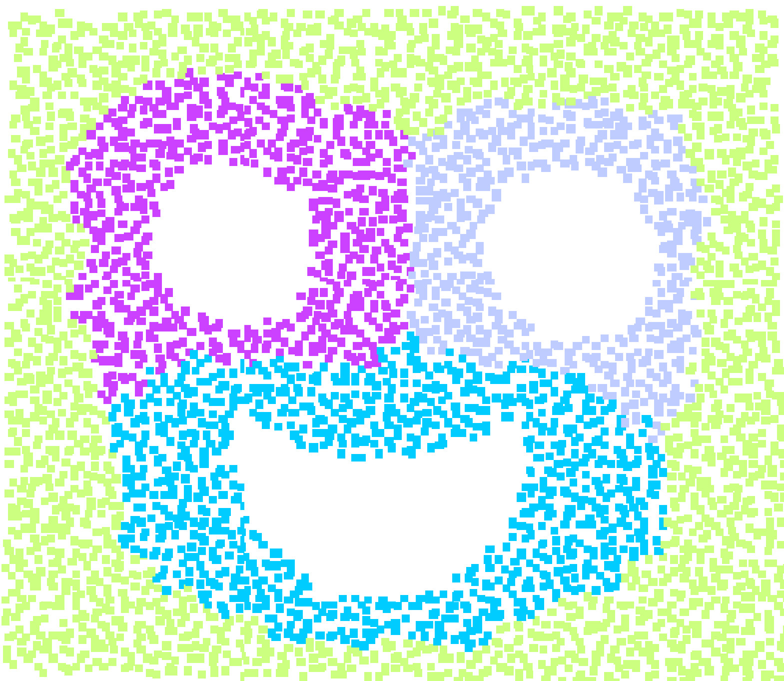
Our contribution

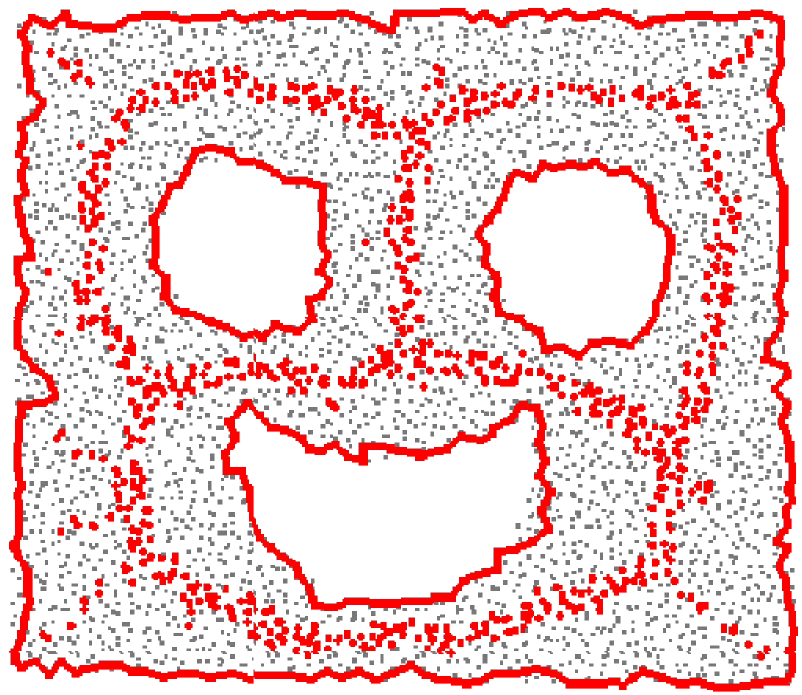
- A practical algorithm that
  - No location information.
  - Not assume unit disk graph model (that is too idealistic).
  - Works well in low-density network (average degree 6~7).
- Scenario: network initialization.

Our Contribution, cont.

- We output boundary nodes connected into cycles.
- Provide numerous topological and geometric features
  - E.g., # holes (genus), the nearest hole to any given sensor, medial axis…
- Theoretical guarantee:
  - prove the algorithm correctly finds all boundaries in the continuous case

Our Contribution, cont.

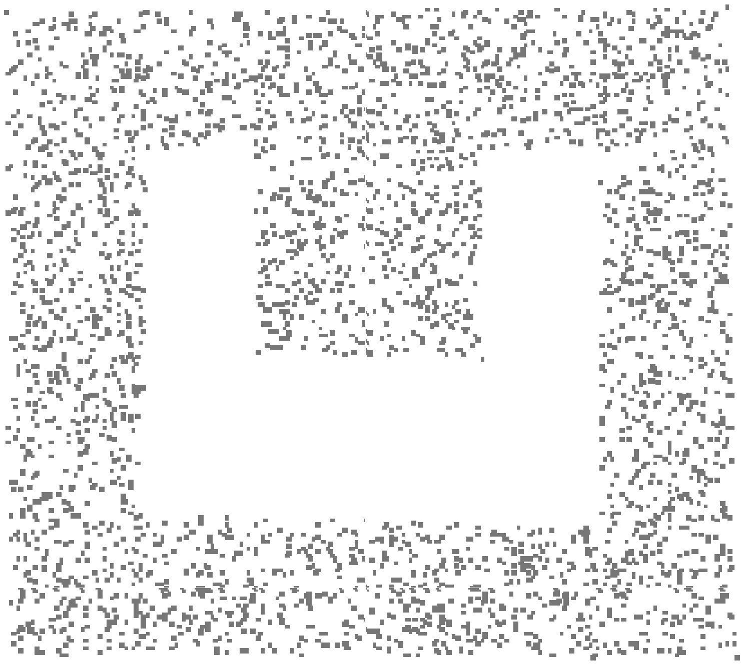




Topological Boundary Recognition

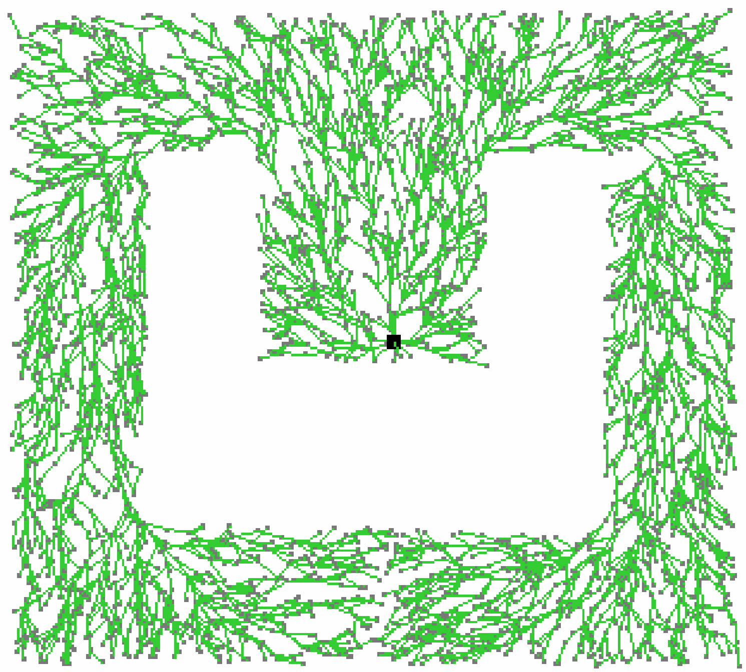
- Basic Idea:
  - Exploit special structure of the shortest path tree to detect the existence of holes.
- Step 1: Build a shortest path tree by flooding
- Step 2: Find cuts in the shortest path tree
- Step 3: Detect a coarse inner boundary
- Step 4: Find extremal nodes
- Step 5: Find the outer boundary and refine the coarse inner boundary
- Step 6: Restore the inner boundary

Step 1: Build a shortest path tree by flooding



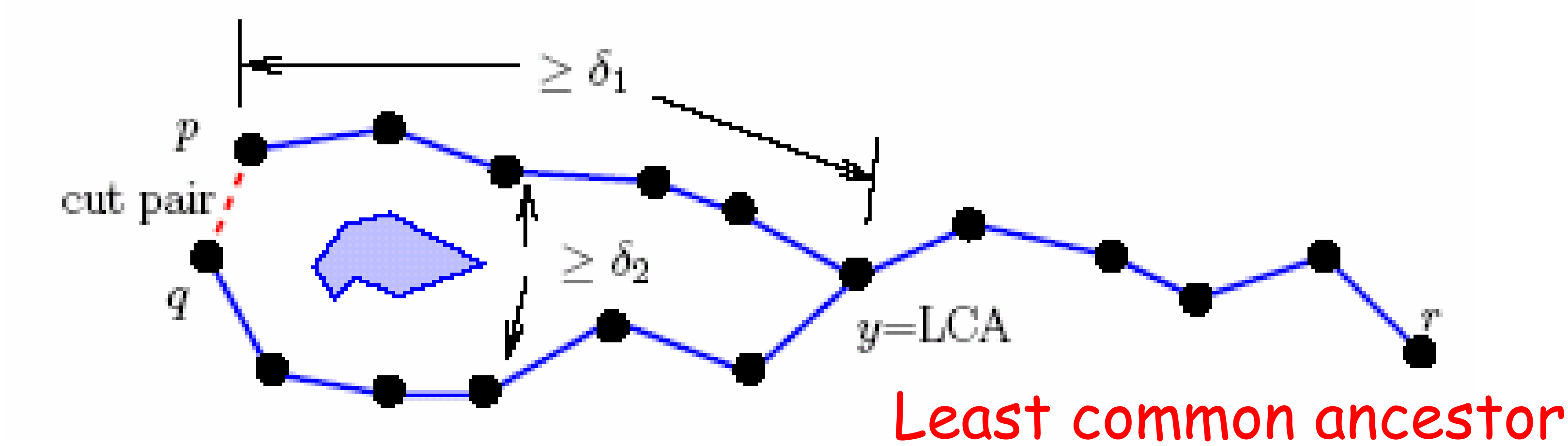
- flood the network from an arbitrary root node.

Step 2: Find cuts in the shortest path tree



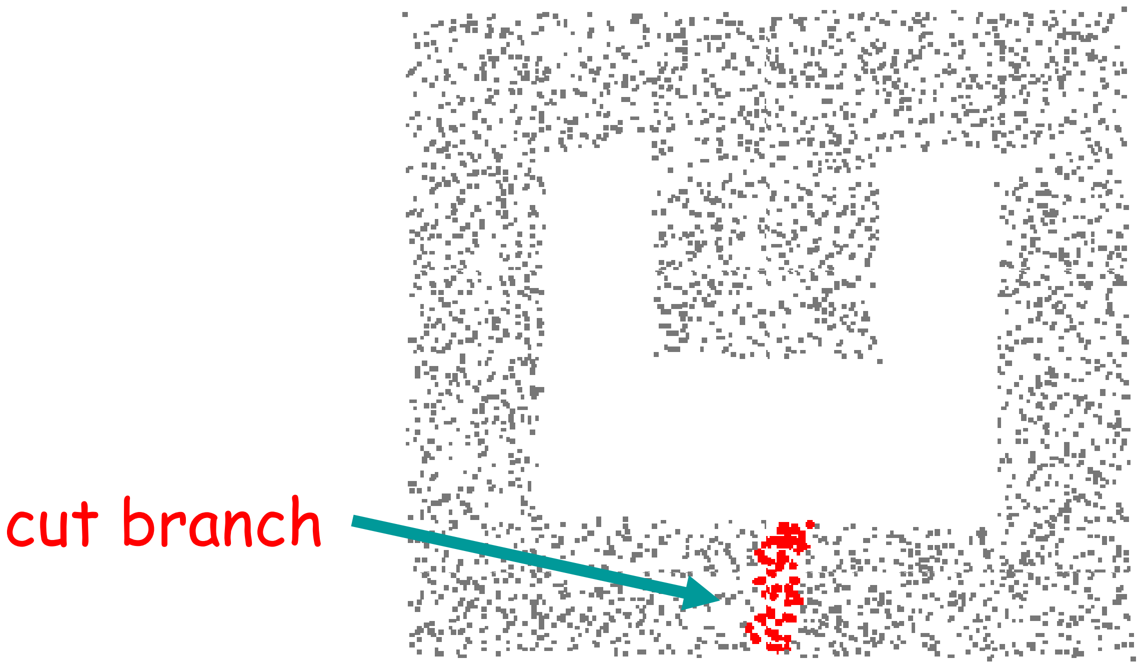
- The "flow" of the shortest path tree forks near a hole, continues along opposite sides of the hole and then meets again past the hole.

Step 2: Find cuts in the shortest path tree – cont.



- A cut pair (p, q) is two neighboring nodes, s.t.,
  - The (hop) distance between p or q and LCA is above a threshold δ_1
  - The maximum (hop) distance between a node on the path from p to y and the path from q to y in the shortest path tree is above a threshold δ_2

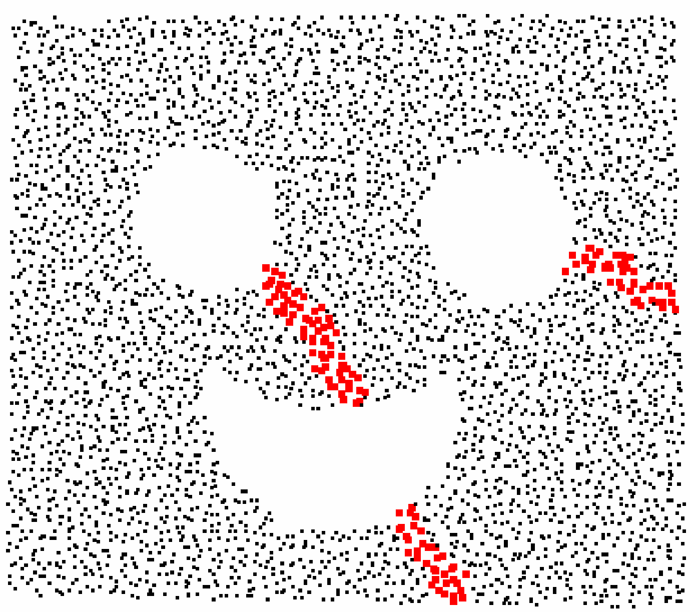
Step 2: Find cuts in the shortest path tree – cont.

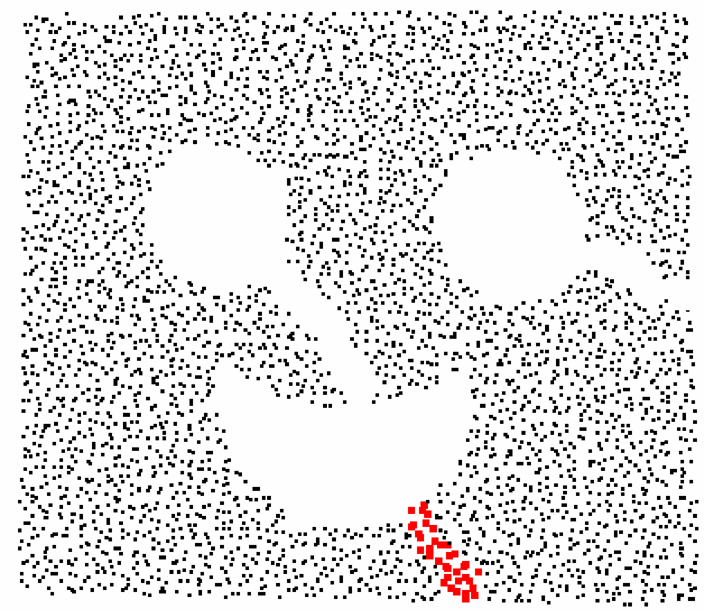


- The cut pair will locally connect themselves into connected components
- Each cut branch corresponds to a cut connected component

Step 2: Find cuts in the shortest path tree - cont.

- Relation between #The cut branches and # holes

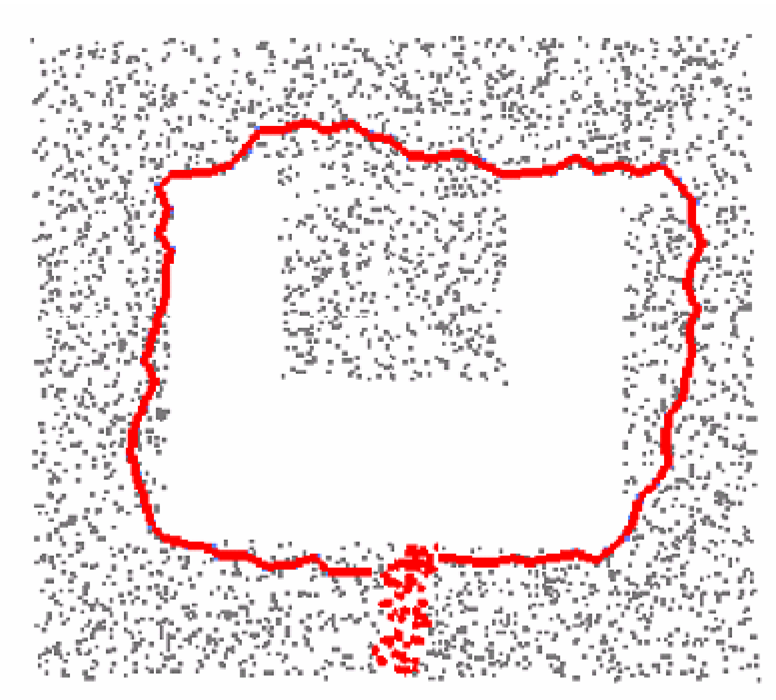


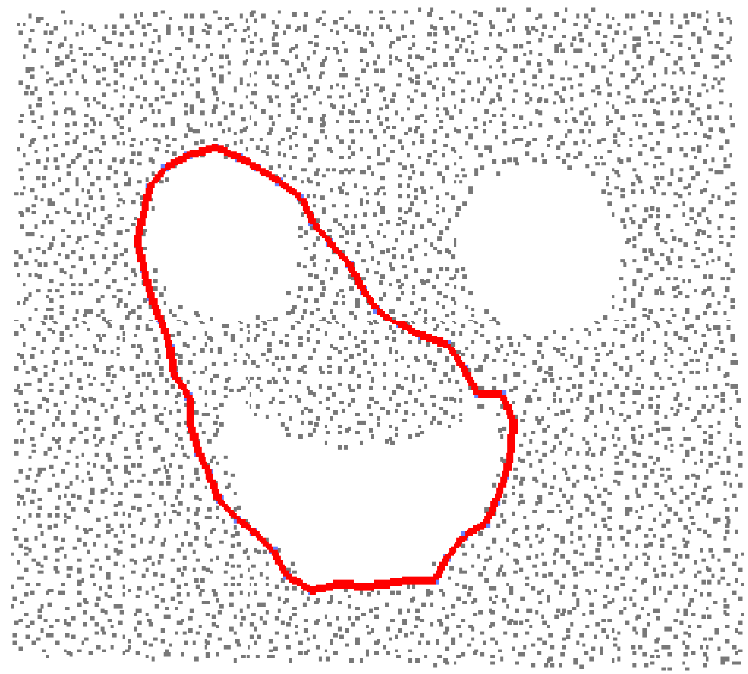


- Multiple holes and multiple cut branches: merge the holes into a single hole

  remove nodes on cut branches, until there is only one composite hole left.

Step 3: Detect a coarse inner boundary R



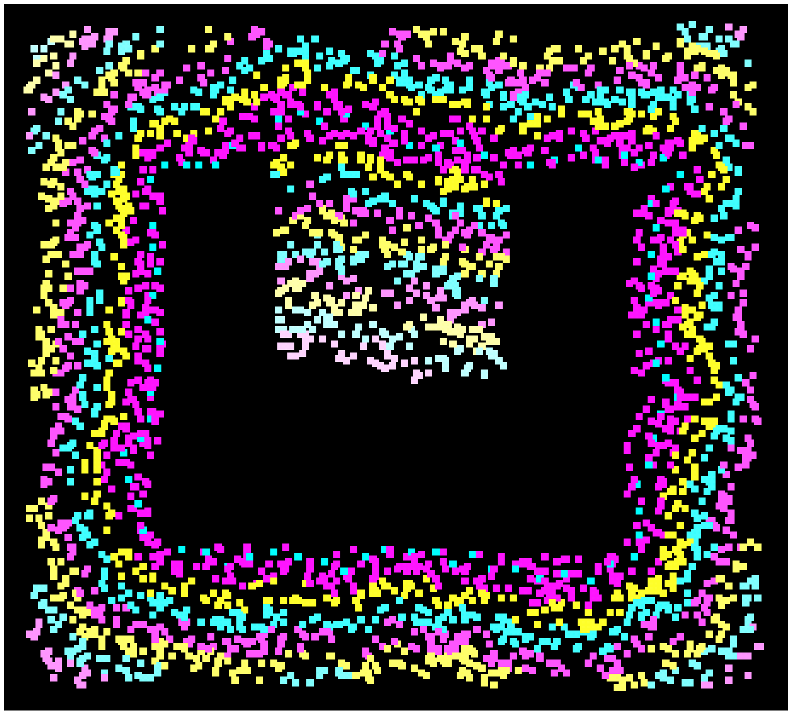


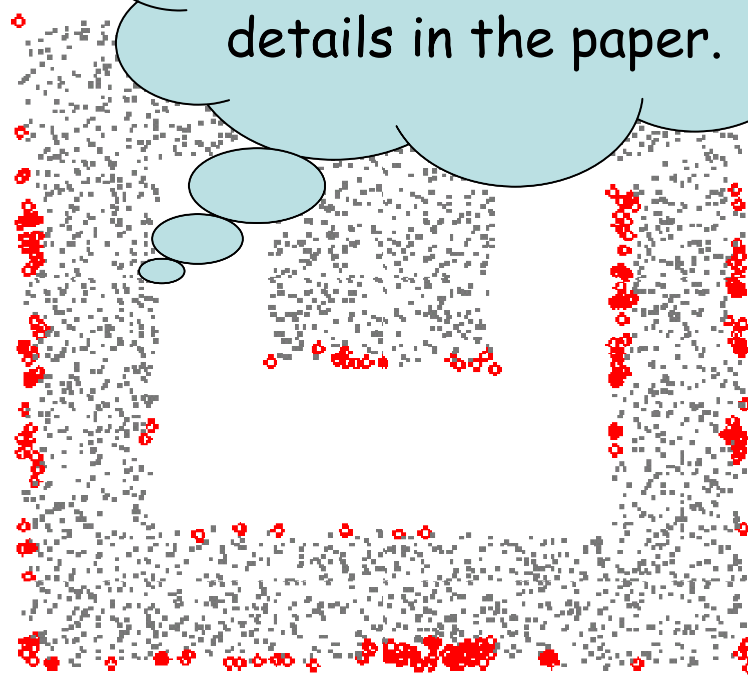
A coarse inner boundary R is a shortest cycle enclosing the interior hole in the sensor field.

- Not tight for concave hole.

Step 4: Find extremal node

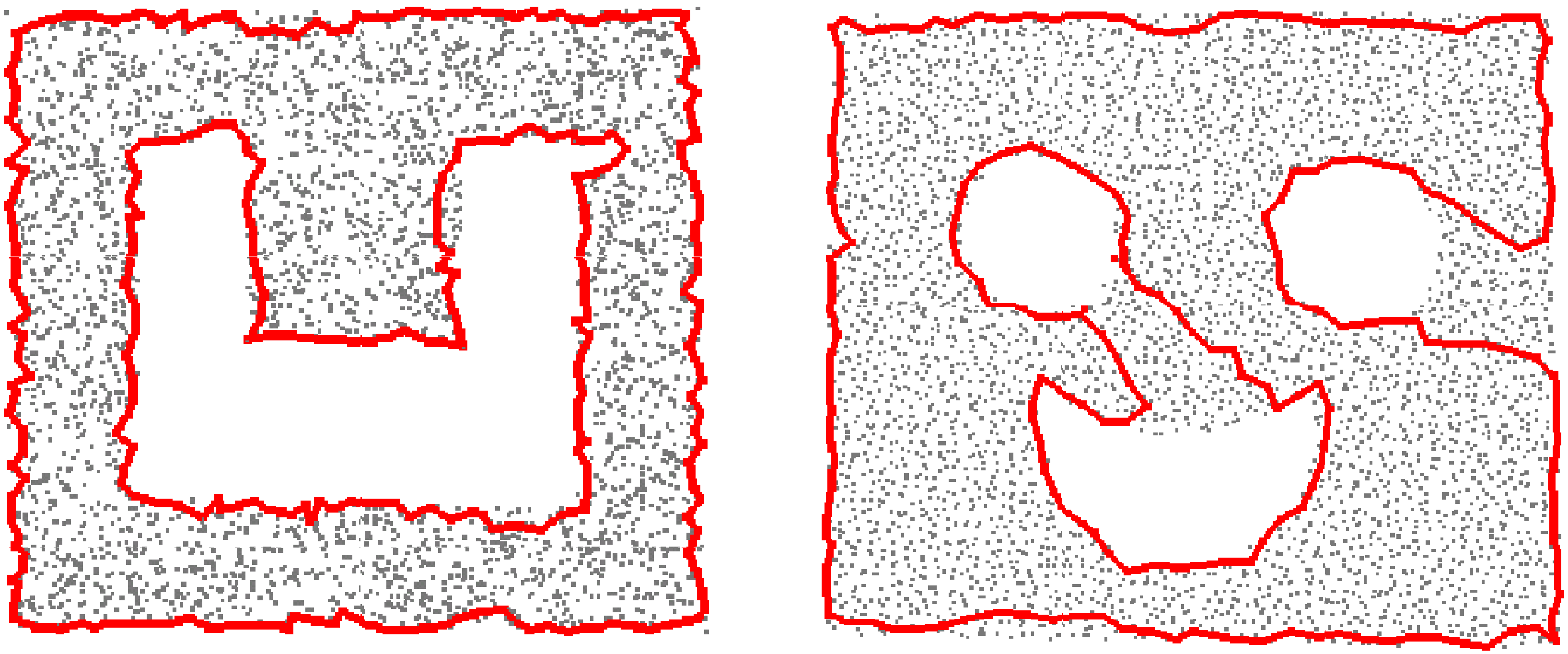
Differentiate extremal nodes on different sides of *R*: details in the paper.





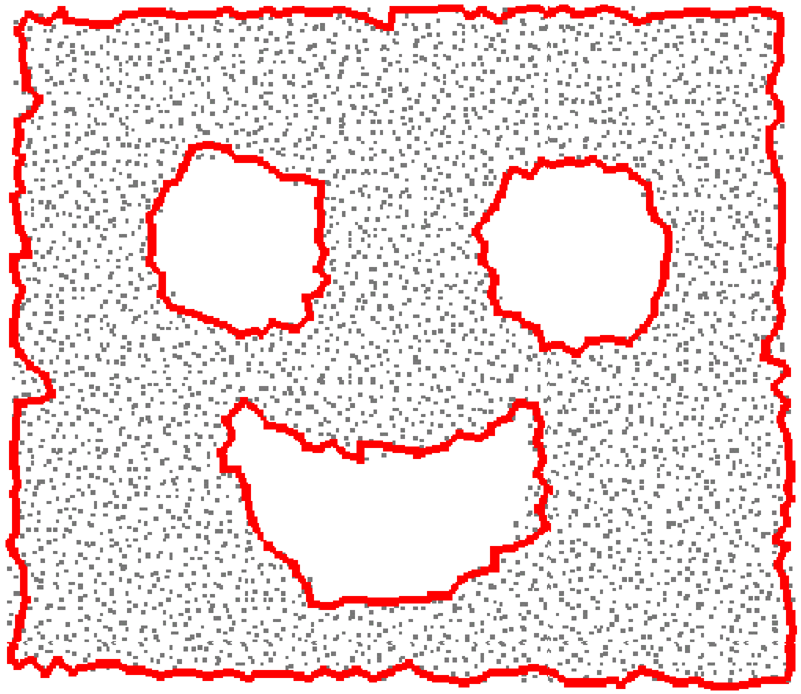
- An extremal node: hop count to nodes in R is locally maximal.
  - on the outer boundary or the ridges of the real inner boundary of a concave hole

Step 5: Find the outer boundary and refine the coarse inner boundary



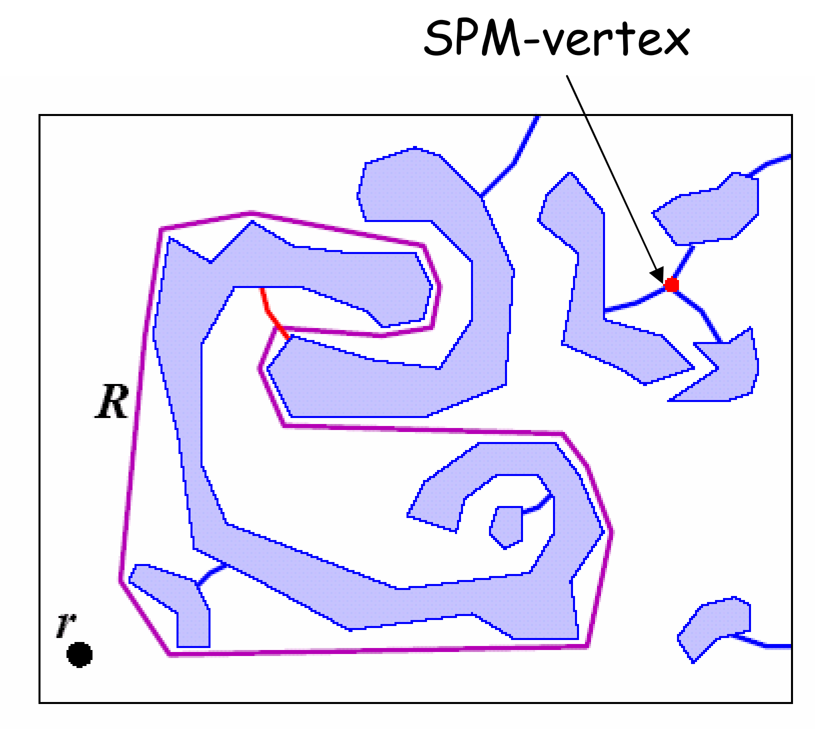
- Force (inner and outer) boundaries to go through extremal nodes.

Step 6: Restore the boundary



- Undelete the cut nodes we removed earlier and restore the correct boundary

Thm: Alg is Correct in the Continuous Case



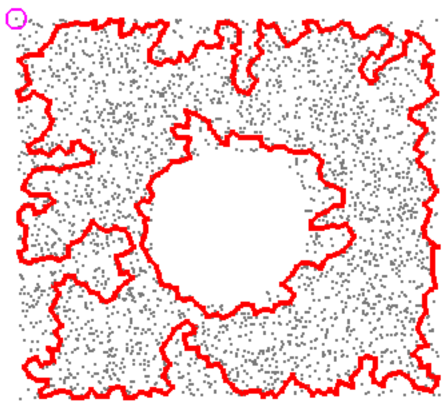
A shortest path map SPM(r)

- In a continuous domain with polygon holes, our algorithm finds the correct boundaries.
- Any shape can be approximated by polygons.

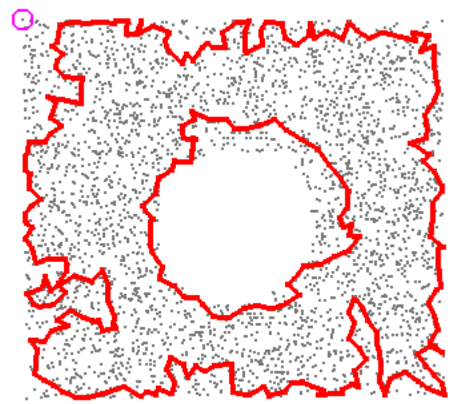
Effect of Node Distribution and Density

- Random distribution
  - Very good results for graphs with avg deg >= 10
  - For very low deg (< 10), take 2-hop / 3-hop neighbors as "fake" 1-hop neighbors
- Grid with random perturbation
  - Gives good results for graphs with avg deg >=6
- Low density, sparse graph
  - Performs well even in such cases

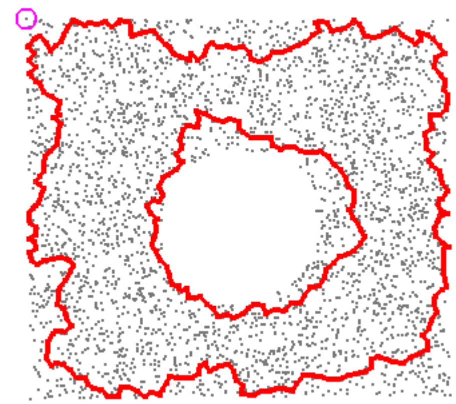
Random Distribution of Sensors



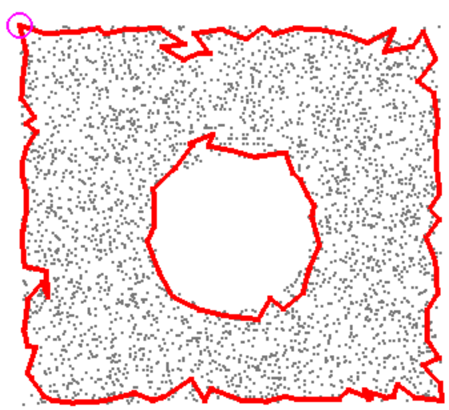
Avg deg = 7
1-hop



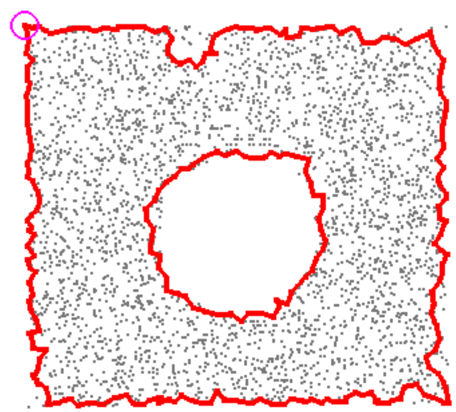
Avg deg = 7
2-hop (9)



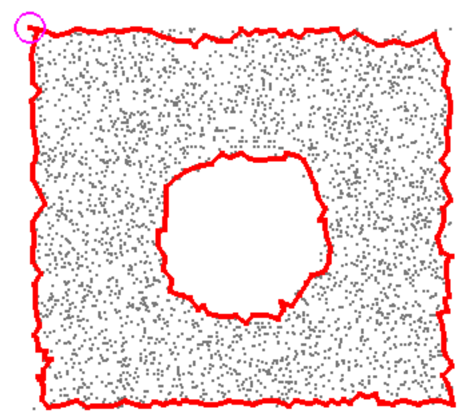
Avg deg = 7
3-hop (12)



Avg deg = 10

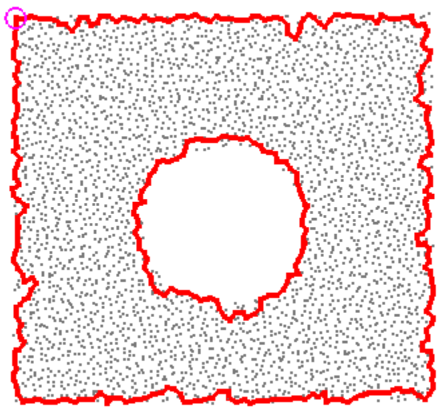


Avg deg = 13

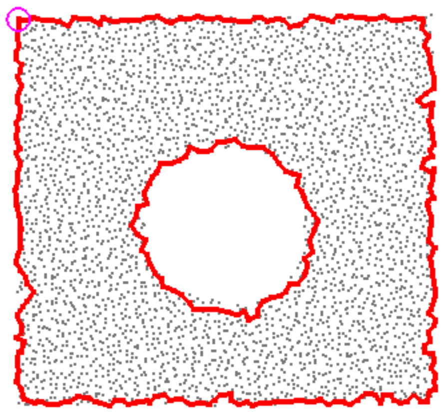


Avg deg = 16

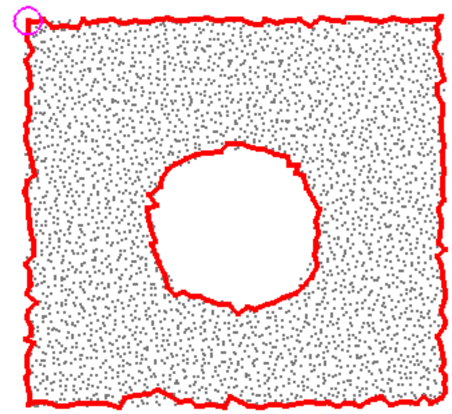
Grid with Random Perturbation



Avg deg = 6

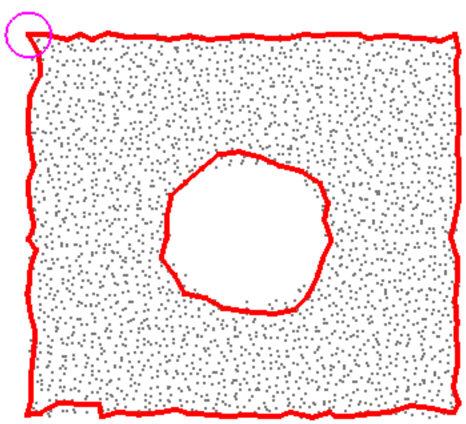


Avg deg = 8

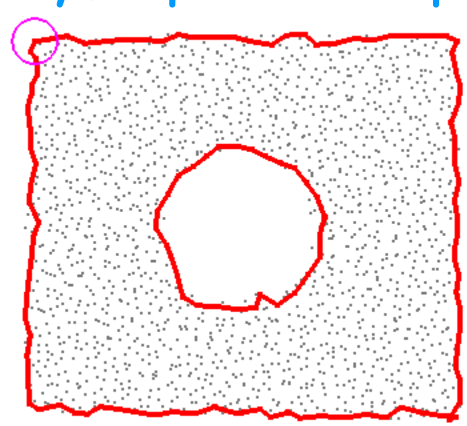


Avg deg = 12

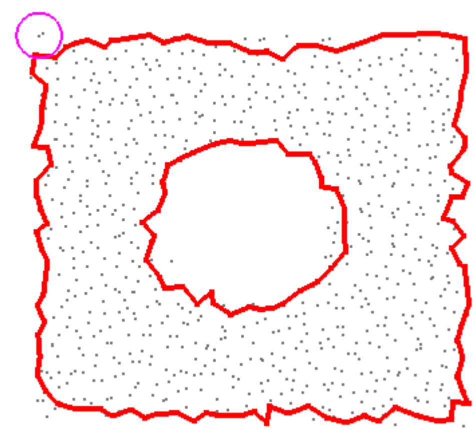
Low Density, Sparse Deployment



2628 nodes
Avg deg = 25



1742 nodes
Avg deg = 16



842 nodes
Avg deg = 7

Complexity of the algorithm

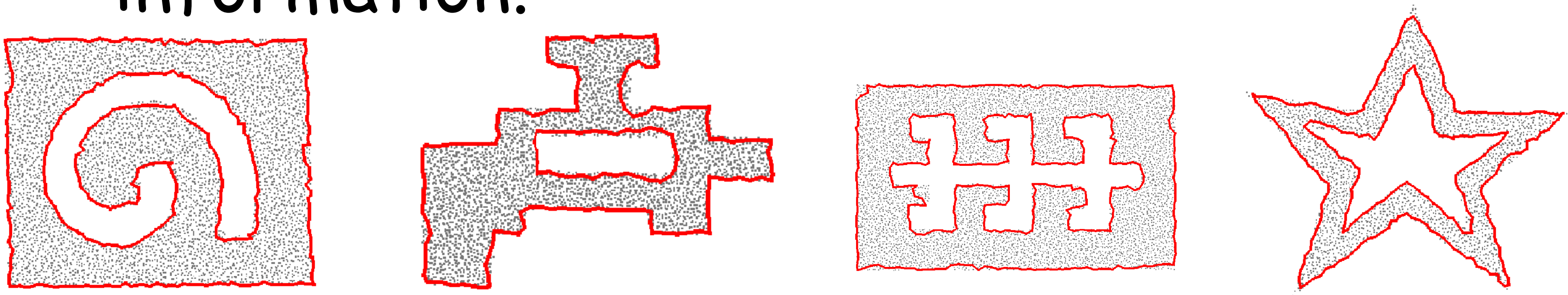
- 2 rounds of flooding.
  - Step 1, flooding for a shortest path tree.
  - Step 4, flood from coarse inner boundary.
- Most other operations are local shortcutting.

Discussion

- The precision of our method
  - There may be more than one shortest paths between two nodes based only on the hop-count
    - lower deg nodes are more likely to be on the boundary
    - use iterative method to find more extremal nodes
  - the correct orderings of the extremal nodes
- Incorporate partial location, location or angular information.
- no hole case?
  - Algorithm will output no hole.
  - Artificially cut a hole to find outer boundary.

Summary

- A practical algorithm for boundary detection with only connectivity information.



- Future work:
  - topology-adaptive algorithms by using boundaries for localization, routing, etc.