Contour Tracking and Information-guided Navigation in Wireless Sensor Networks

Jie Gao

Stony Brook University

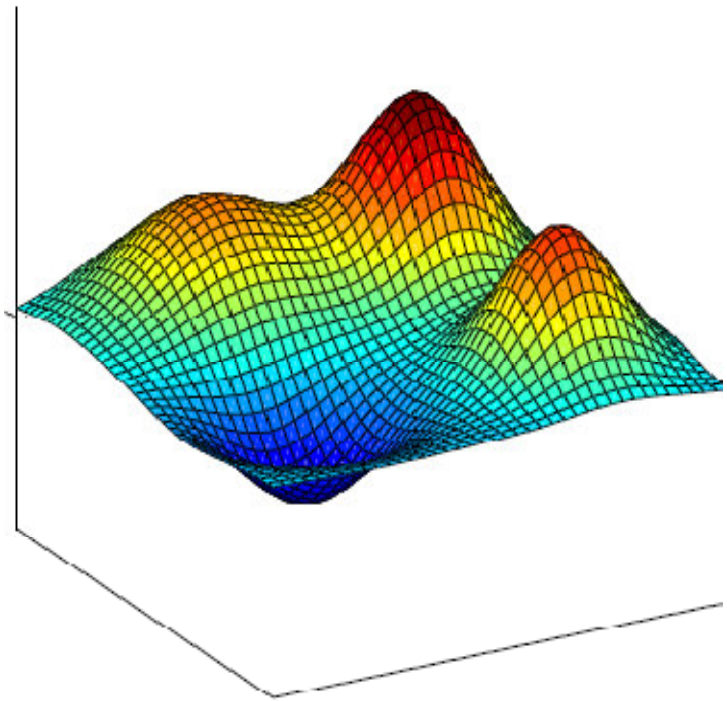
Joint work with

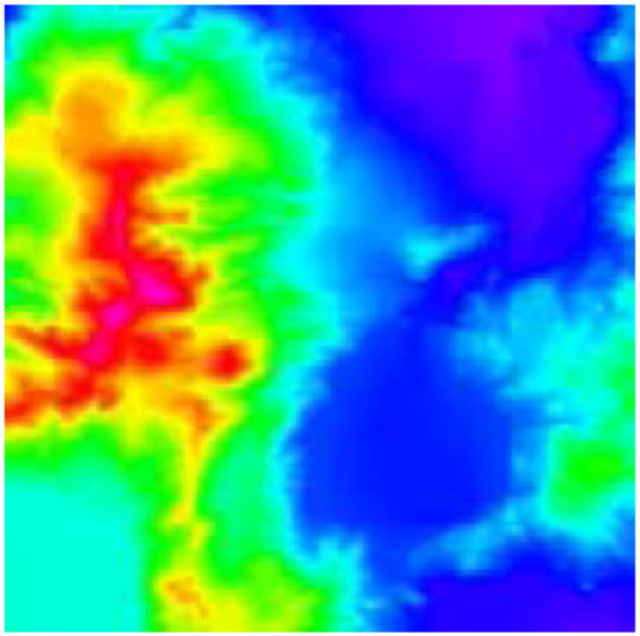
Xianjin Zhu, Rik Sarkar,

Leonidas Guibas, Joseph S. B. Mitchell

Sensing and reacting to a spatial signal

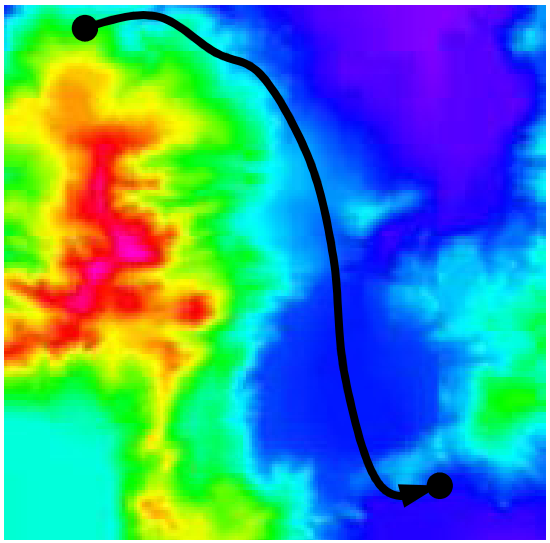
- Dense sensors distributed to monitor a spatial signal (temperature, traffic density, pollution level, etc).



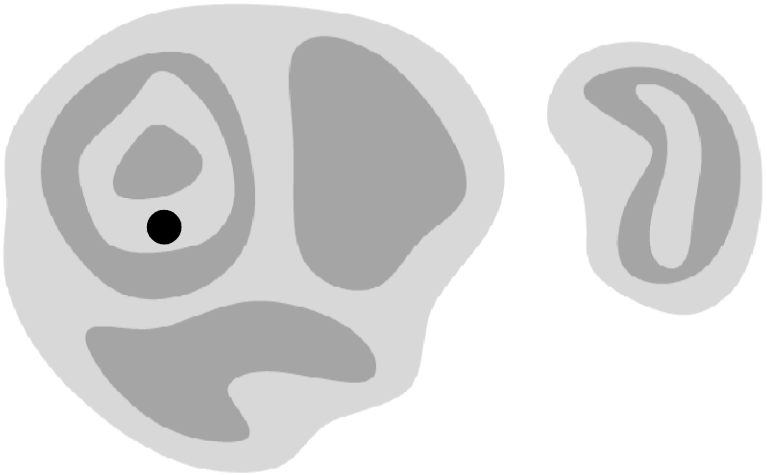


“A smart environment”

- Real-time situation understanding
- Information-guided navigation
  - Real-time response and emergency rescue
  - Direct vehicles to alleviate traffic jam



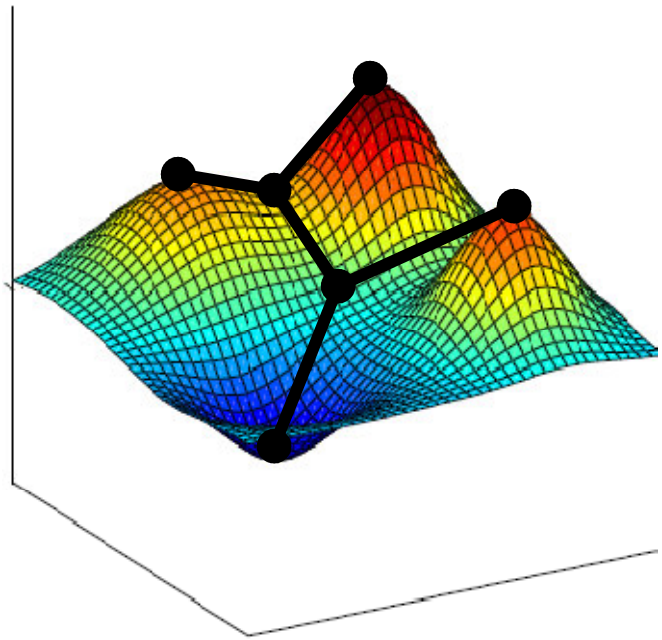
Find a low-value path?

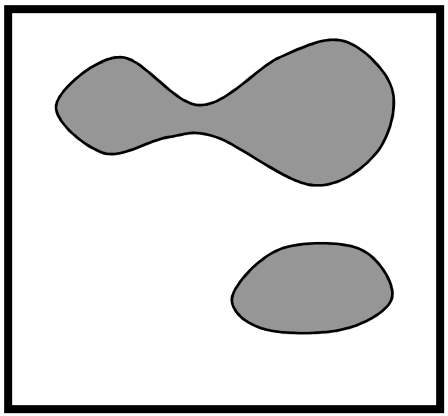


Am I surrounded?

Goal: track the spatial structure

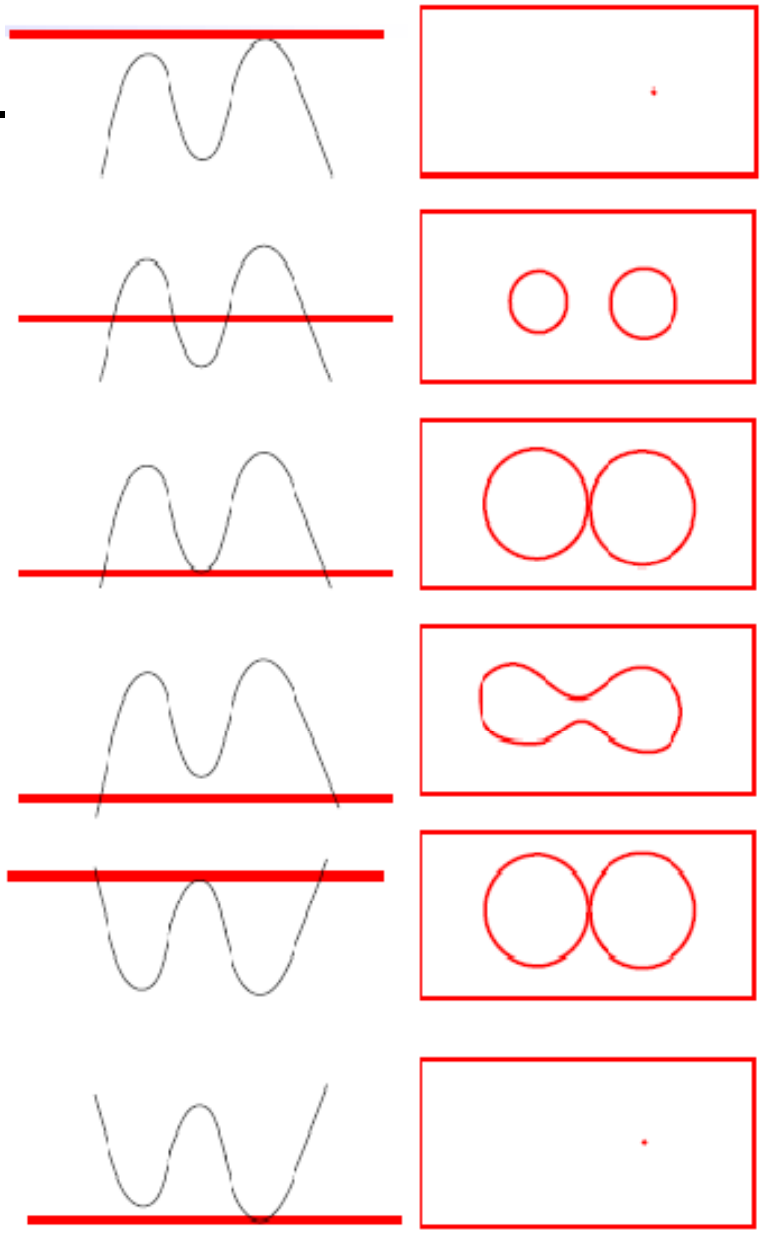
- Track contours and their evolvement over time.
- Track local max/min, saddles, and relationship between them.





Contour tree

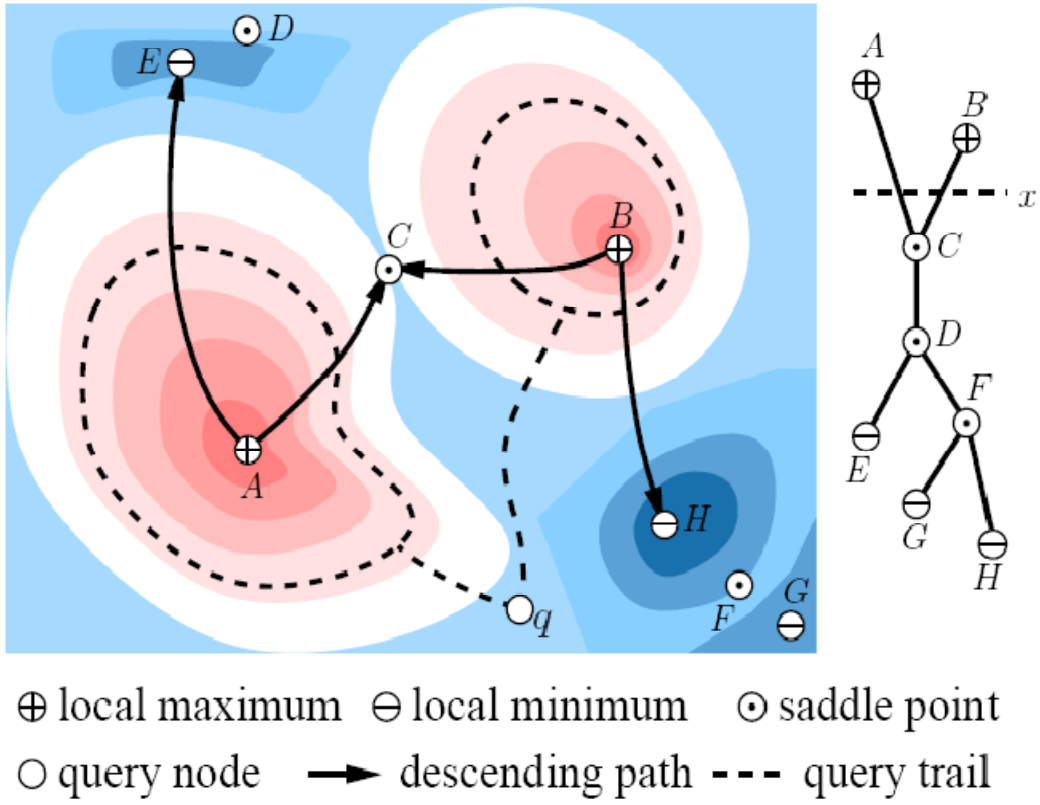
- A tree on critical points (local max/min, saddles).
- Capture the contour merging/splitting, emergence, disappearance.



Information guided navigation

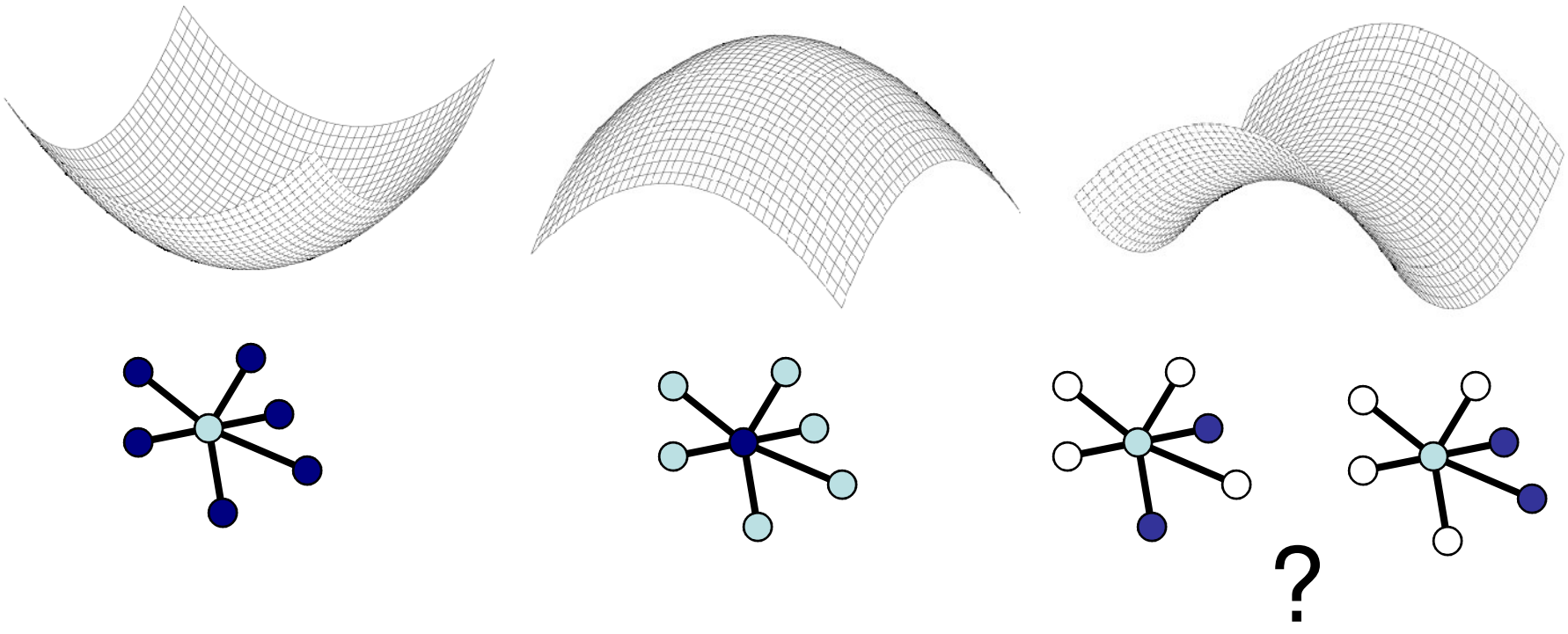
- **Iso-contour queries**: find the contours at level x.
- **Low-value routing**: find a path from s to t with value below x.

Gradient descent methods + contour tree maintenance.



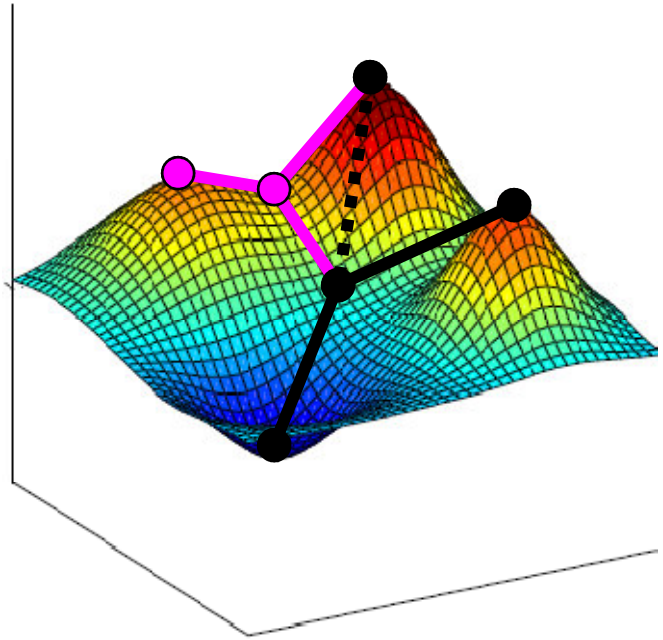
Challenges of maintaining a contour tree

1. No location or with location errors.
   - Can we locally identify max/min and saddles?



Challenges of maintaining a contour tree

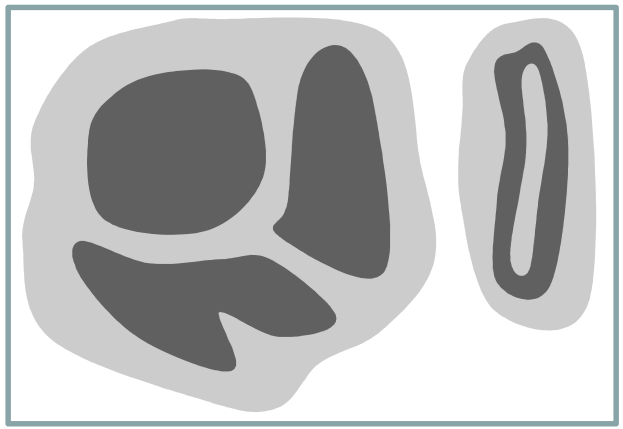
1. No location or with location errors.
2. Low communication cost.
   - Connect the critical points.
   - Emergence of a local max/min can change the tree in a non-local fashion.



This talk: two (simpler) problems

- Maintain the iso-contour at value x in a time-varying signal field.
- Construct the contour tree in a static field.

Contour tracking problem

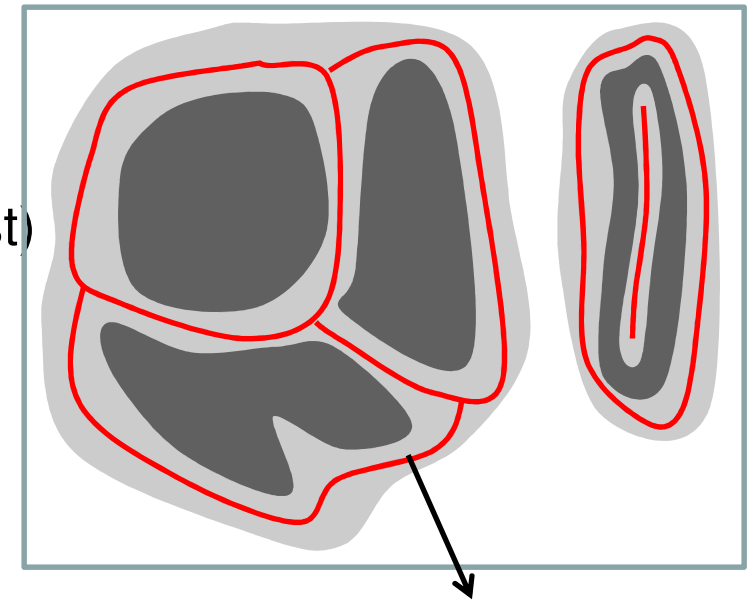


- Binary sensor model:
  - Value 1 = in the range of interest
  - Value 0, otherwise.

- Color of senor i
  - **BLACK**: with value 1 and all its neighbor have value 1
  - **WHITE**: with value 0 and all its neighbor have value 0
  - **GRAY**: otherwise.
- Black regions and white regions are separated by gray.
- Robustness: a salt-and-pepper signal field is not worth tracking.

Contour tracking problem

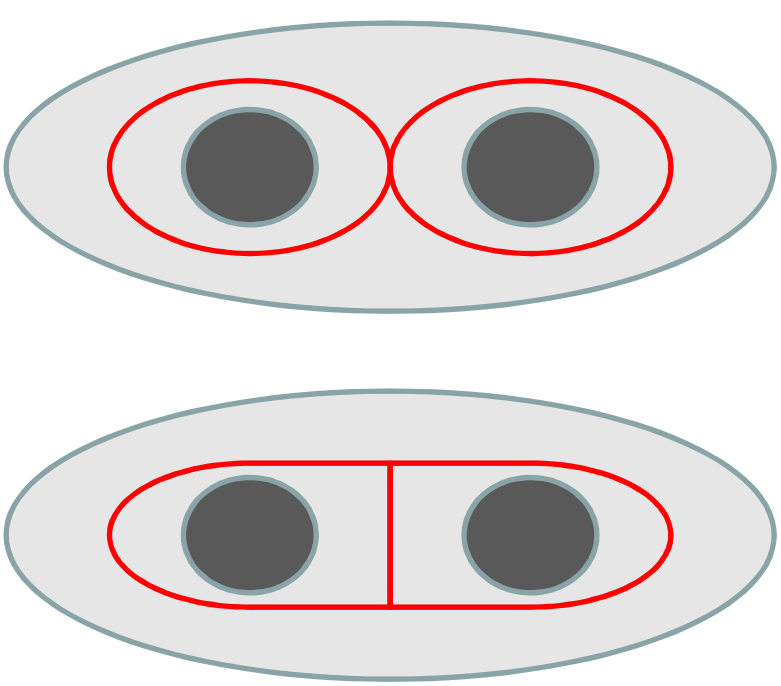
- Contour network: capture topological features of dynamic contours
  - Inside k-gray band: within k hops from BLACK regions (tightly enclosing BLACK & reduced communication cost)
- Deformation retract:
  - A continuous map that shrinks a space to a "thinner" subspace, while maintain the same topology

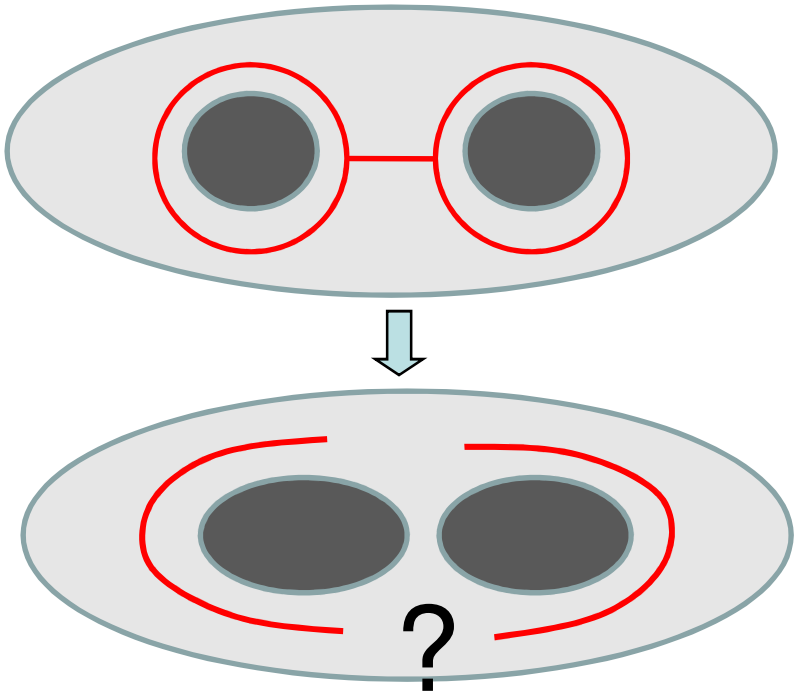


Challenges

A valid contour network may have very different local features.

When a part of the network is removed, how to repair it?

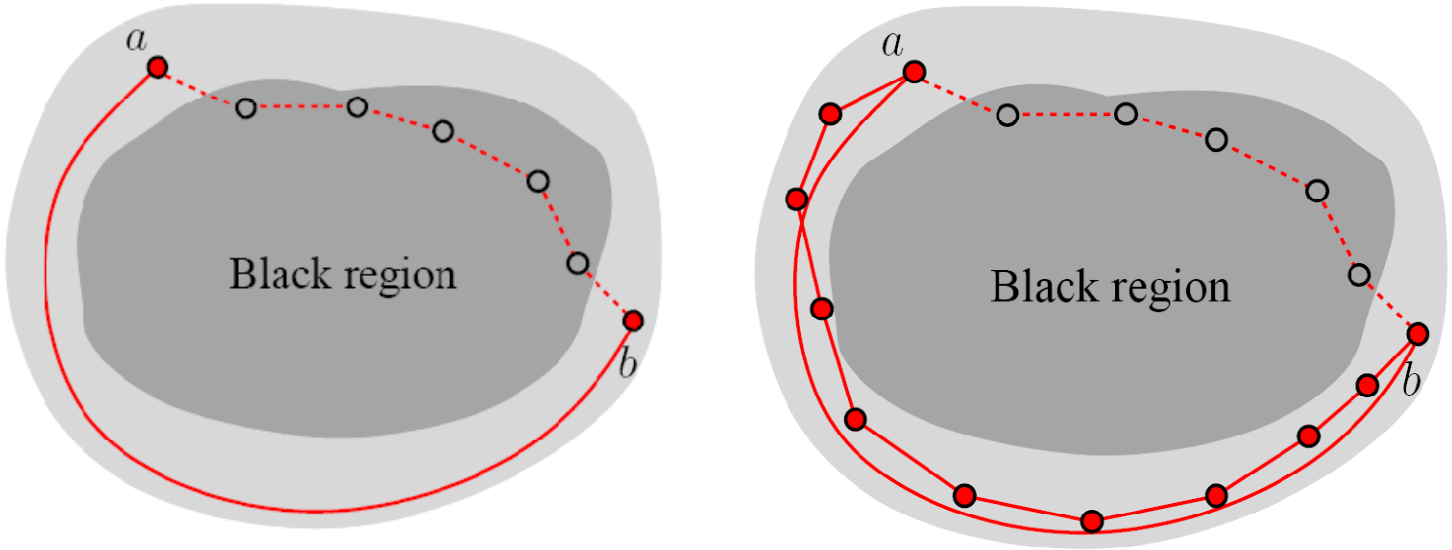




How to reconnect the partial contours?

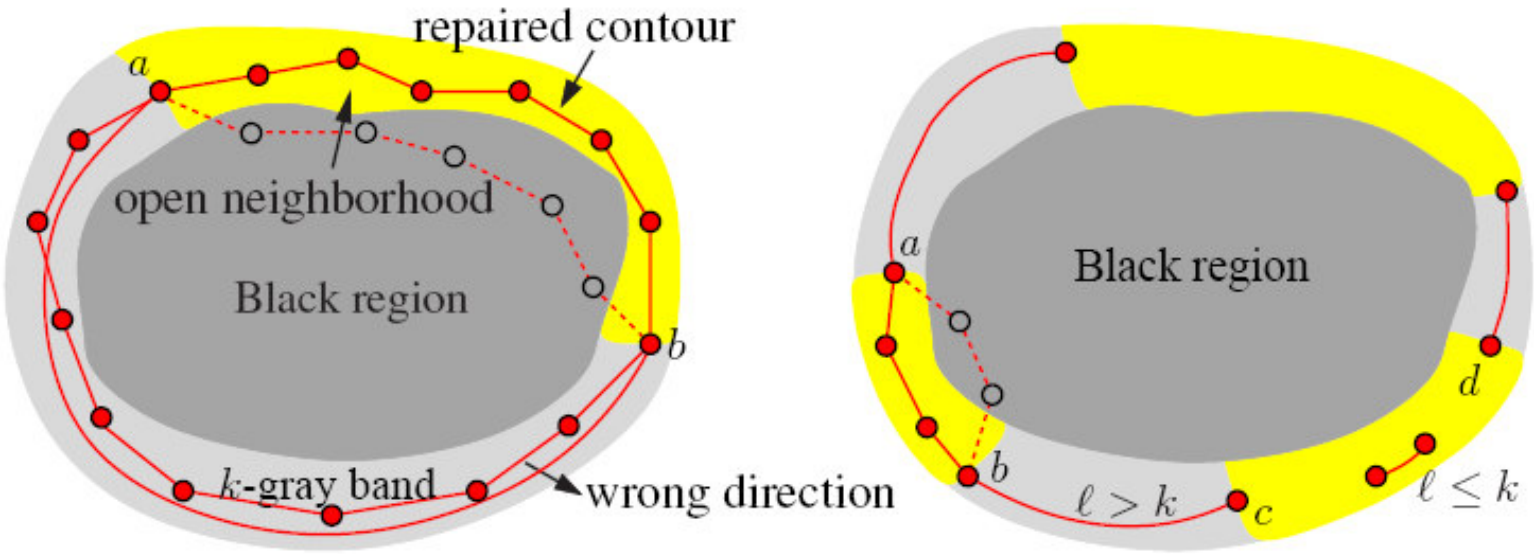
Repair a single contour cycle

- Who is responsible for the repair?
- Open red nodes take responsibility to repair
  - Red nodes (on the contour) that lose one or more current red neighbors become OPEN.
- Competition is resolved by IDs (lower ID wins).



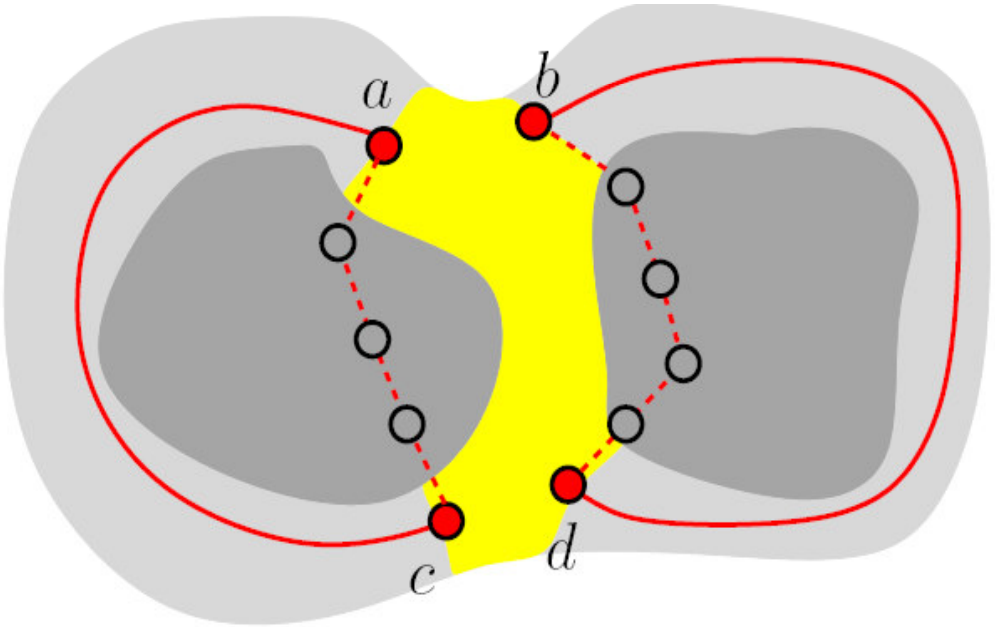
Repair a single contour cycle

- Make sure we re-connect the open endpoints in the correct direction.
- The k-hop neighbors of the closed red segments block the propagation of repair messages.



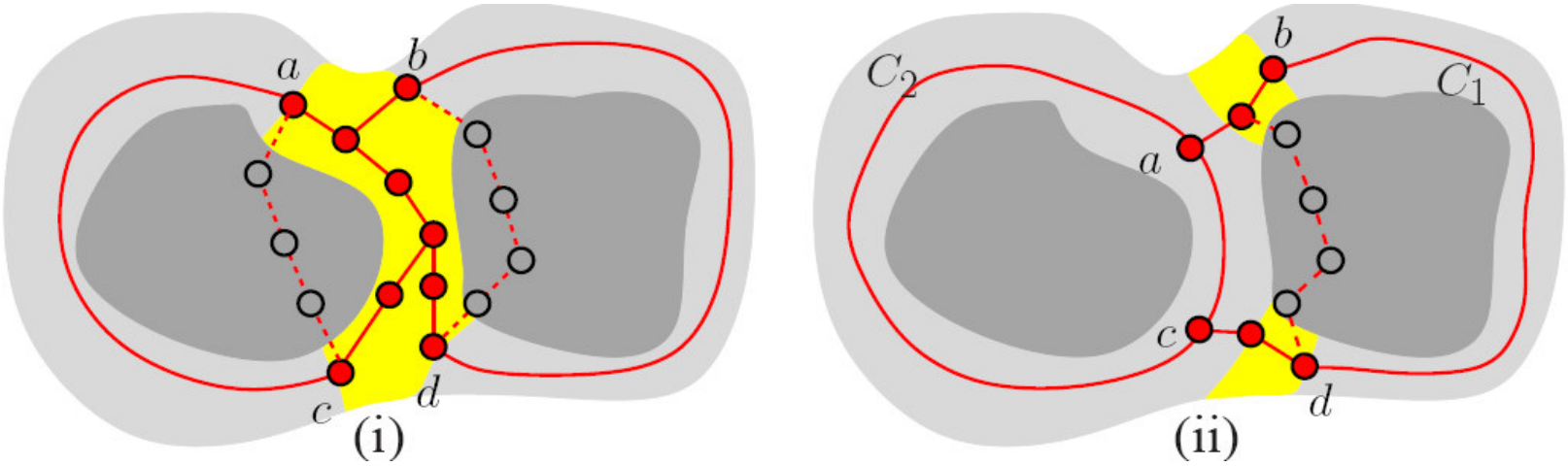
Contour merging

- Simultaneous repair, merging and splitting
  - A repair message may encounter multiple OPEN RED nodes, how to connect them?

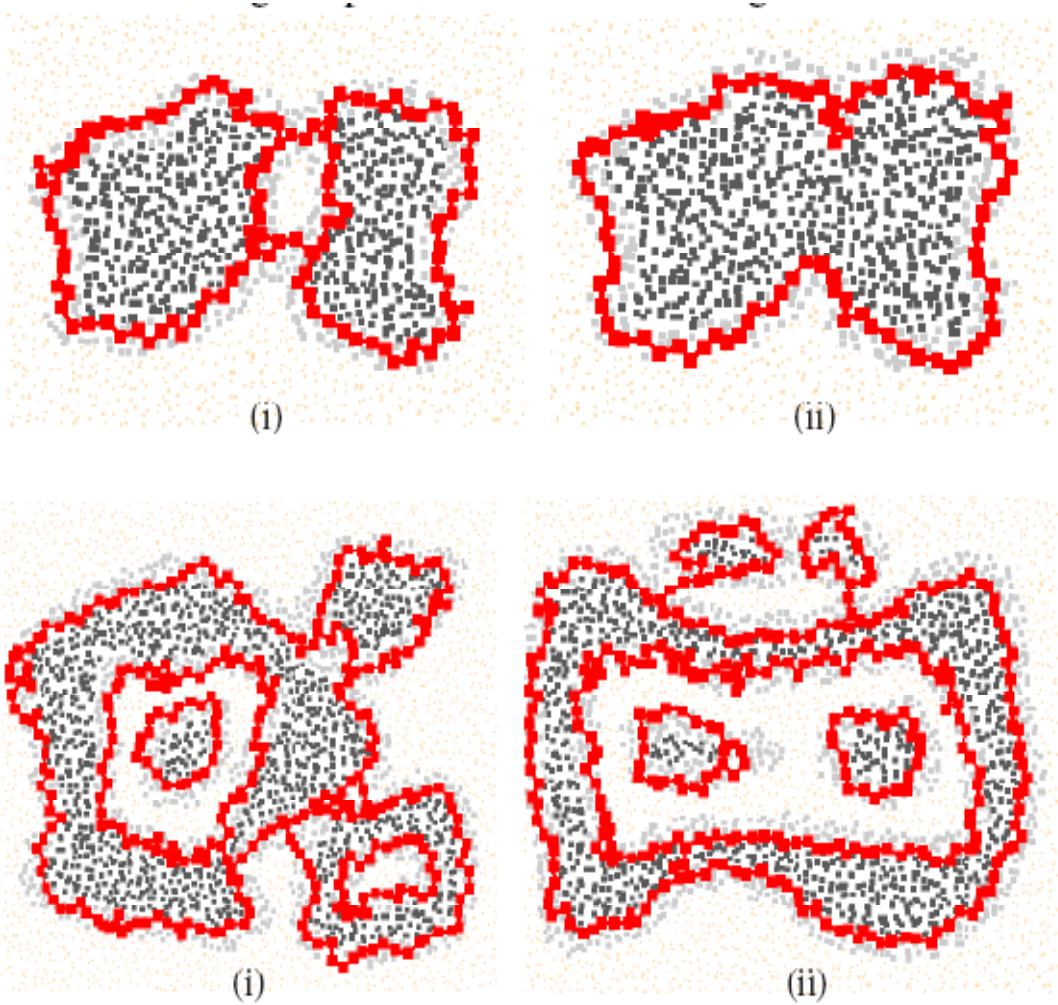


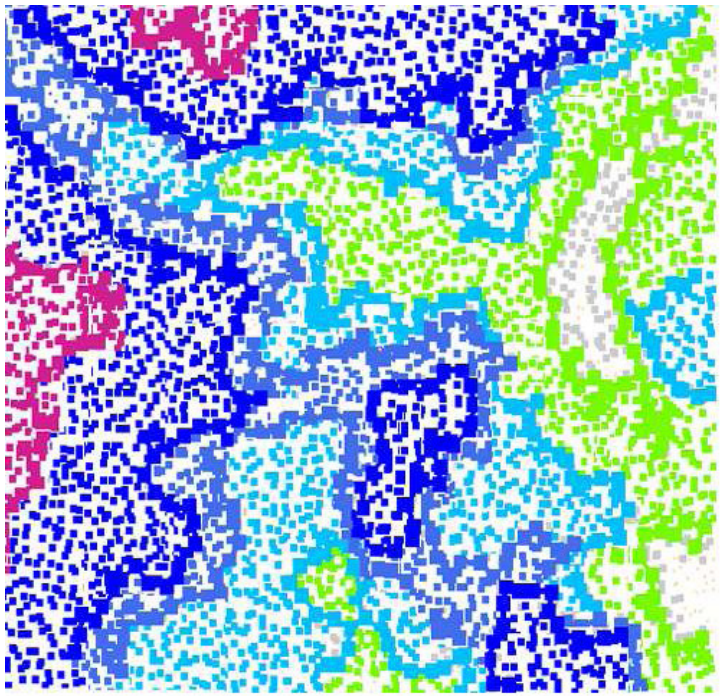
Contour merging

- Simultaneous repair, merging and splitting
  - Connect by a (non self-intersecting) spanning tree.
  - Say, the shortest path tree.



Contour tracking examples

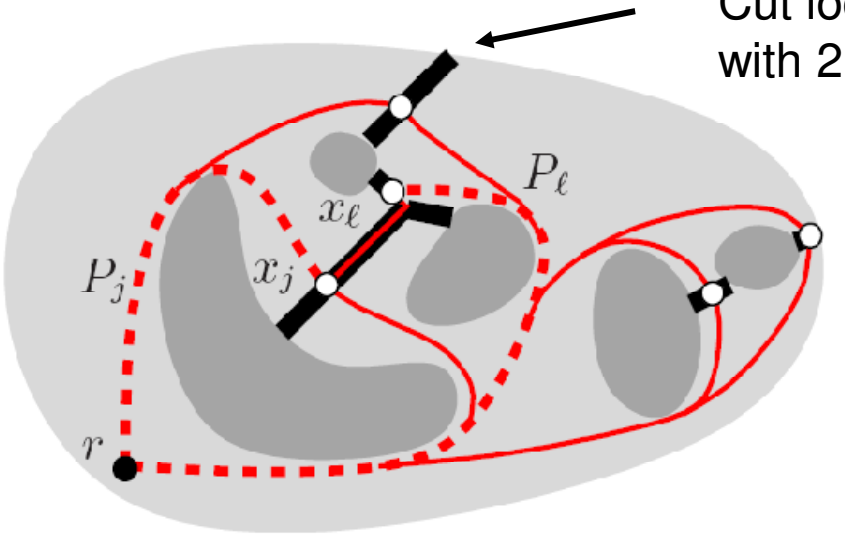




Video clip

Theoretical results

- Theorem: when the system stabilizes, the contour network is a deformation retract of the k-gray region.
- First, how to find a deformation retract.



Cut locus: collection of points with 2 or more shortest paths to r.

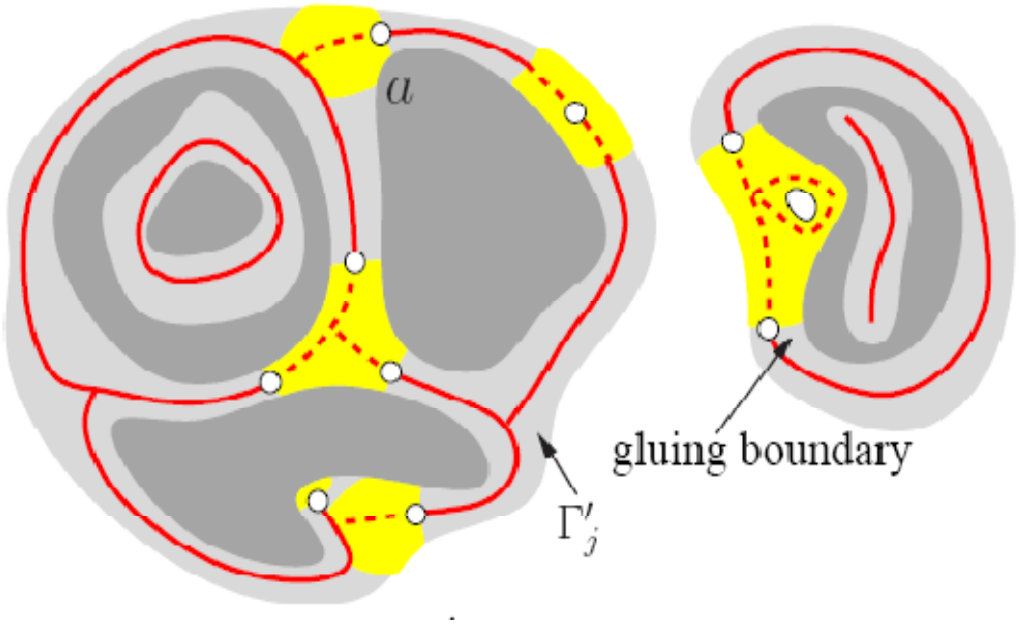
Cut along the cut locus ➔ simple region whose boundary has alternating cut locus segments and hole boundaries.

Connect via shortest paths to each cut locus segments.

Connect these attaching points by a tree on each cut locus component.

Theoretical results

- Algorithm: inside each open neighborhood, combine the contour network inside this region & shortest paths to bounding red segments.



Note: this repair uses collection of shortest paths, thus doesn't create extra crossings.

Proof sketch: define a continuous map to shrink the k-gray band to the contour network.

A few notes

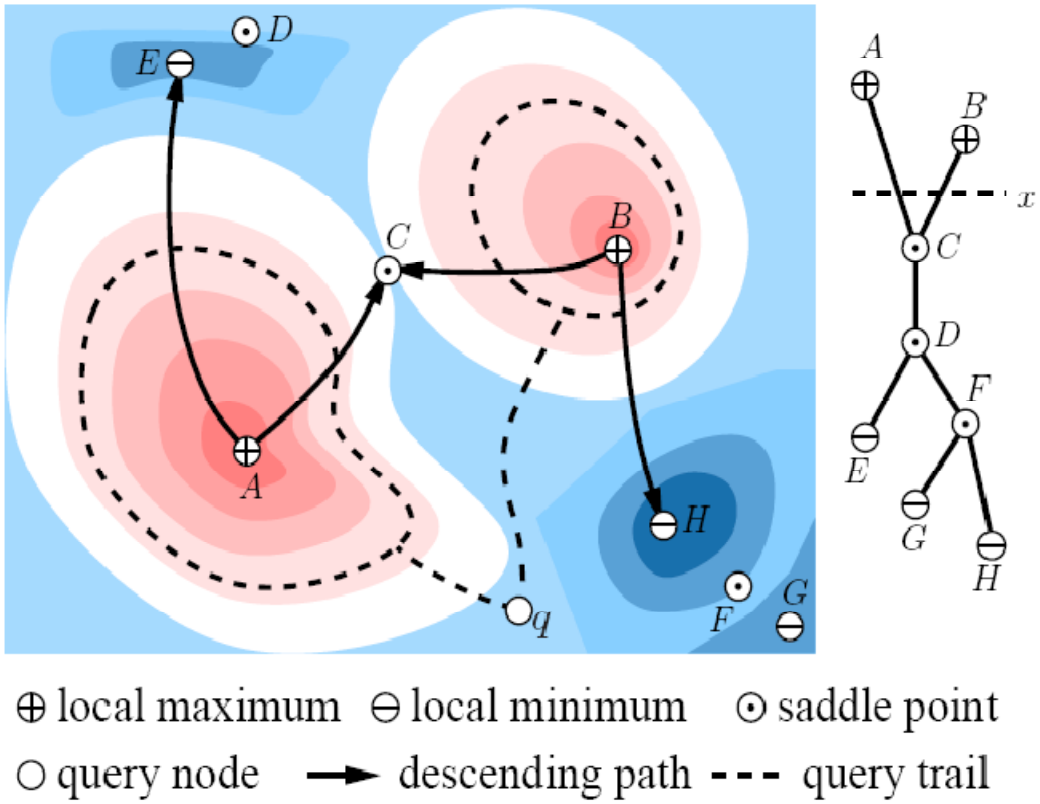
- A light-weight distributed algorithm that captures the topological features of time-varying contours.
- The communication cost is “output-sensitive”, proportional to the amount of contour changes.
- The algorithm provides a foundation for further processing of spatial sensor data, e.g., contour compression and aggregation [Gandhi, Hershberger, Suri 2007].

This talk: two (simpler) problems

- Maintain the iso-contour at value x in a time-varying signal field.
- Construct the contour tree in a static field.

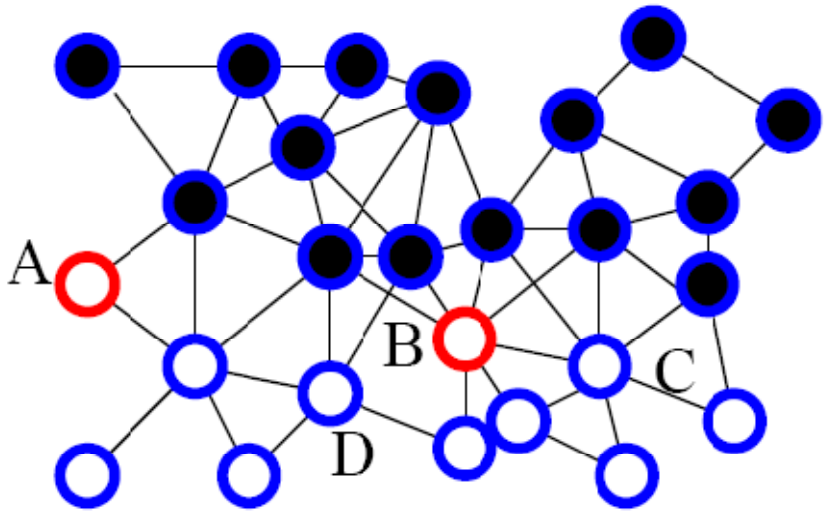
Recall: contour tree

- Tree capturing the contour merging and splitting as the value decreases.
- First: how to define a saddle point?



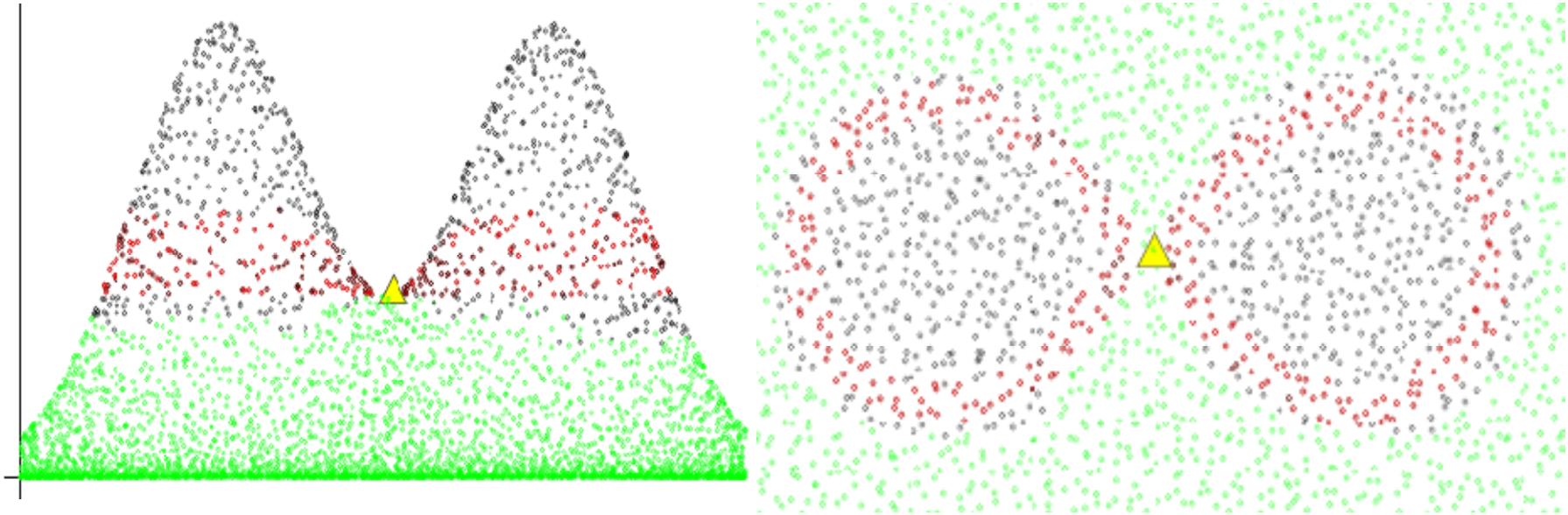
Use distributed sweeps

- Start from local maximum.
  - A node is swept if all its higher neighbors have been swept.
  - The sweep carries the ID of the local MAX.



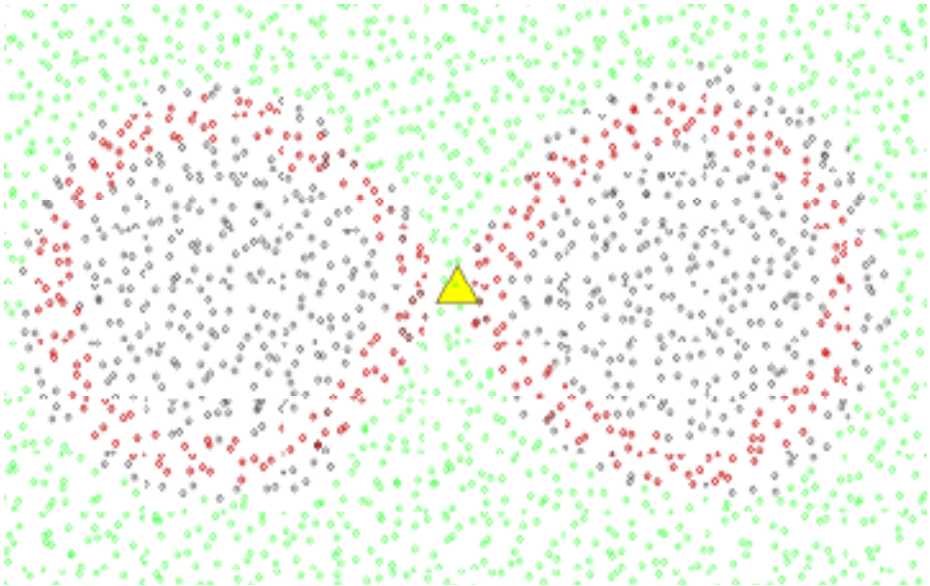
Saddle point

- Saddle: the node w/ highest value that receives sweep msgs from different critical points.
  - Property: a saddle v is a cut node of the graph on vertices with values above it.



Algorithm to find a saddle

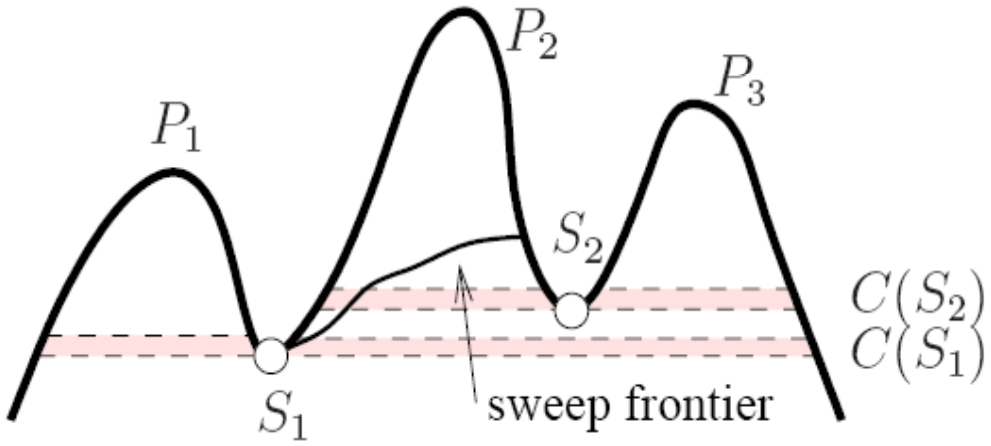
- A node receiving two different sweep msgs
  - Propose to be a saddle candidate
  - Flood the saddle contour (nodes with value above f(v) who has a neighbor below f(v)).



Saddle candidates retire if they meet a candidate with higher value.

Ambiguity?

- Sweep progresses in an asynchronous manner.
  - Contour traversal will verify that all nodes on the saddle contour have been swept properly.



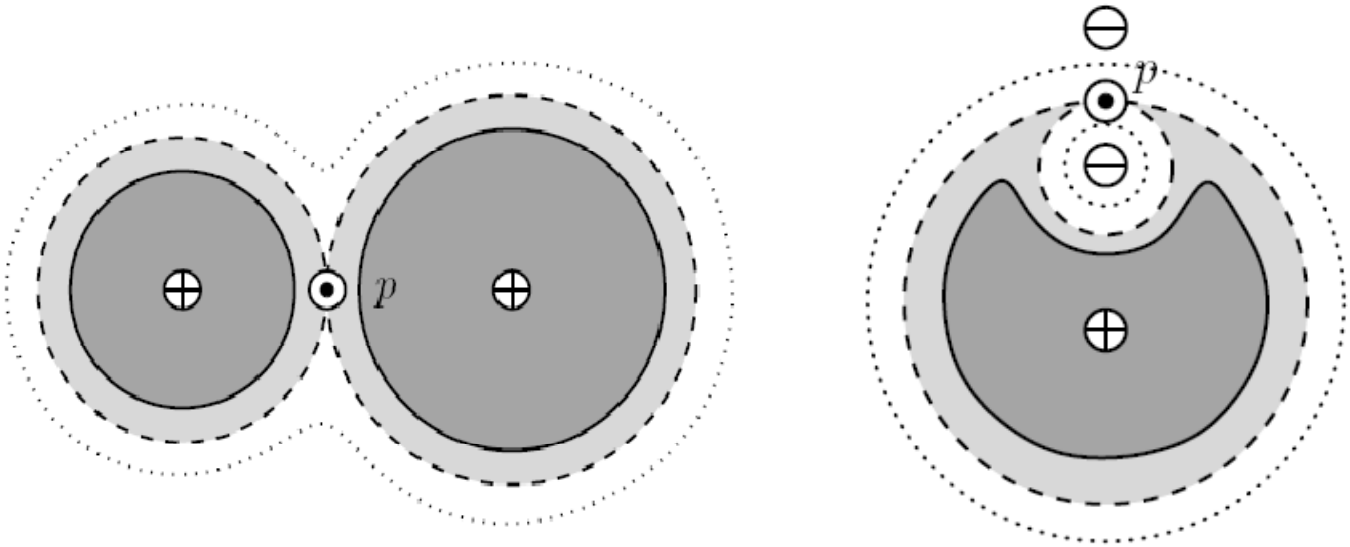
S1 proposes to be saddle for P1, P2 and only wins if all nodes in the contour saddle have been swept by P1 and P2.

S2 will stop sweeping --- this will prevent S1 to win.

S2 wins and starts a new sweep with ID of S2.

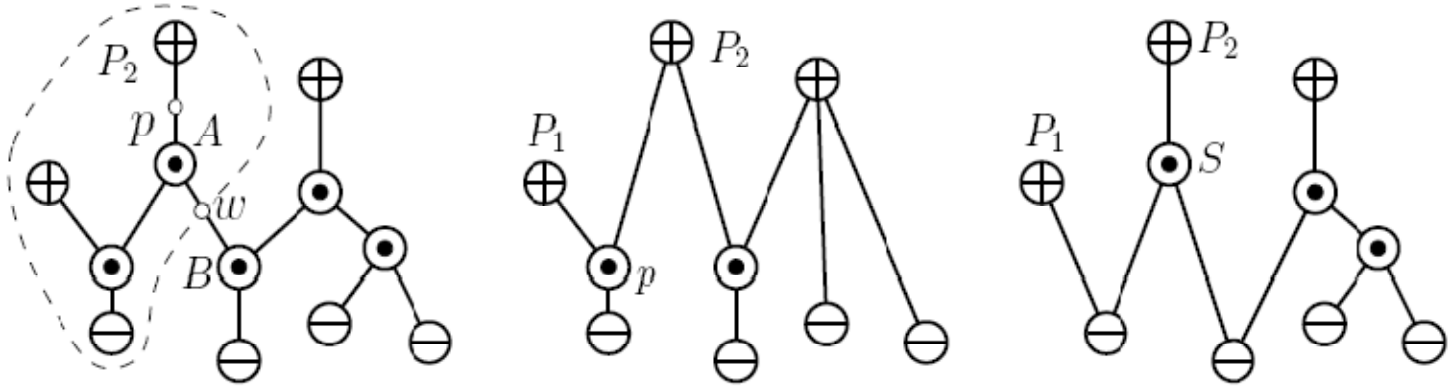
Sweeps

- Top-down sweeps from all local MAX
  - Find all merge saddles.
- Bottom-up sweeps from all local MIN
  - Find all split saddles.



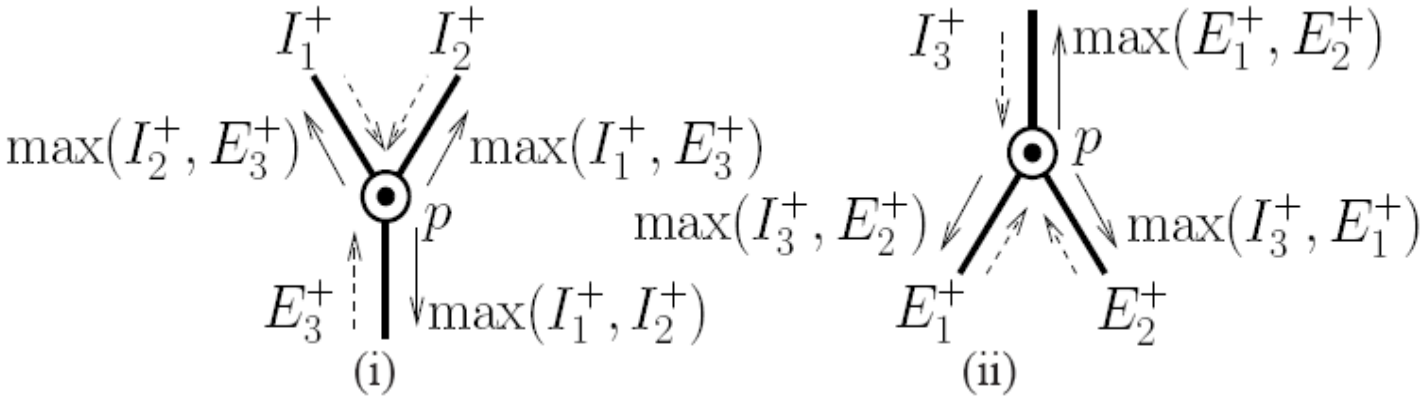
Merge tree and split tree

- Merge tree: on all merge saddles.
- Split tree: on all split saddles.
- Put them together: the contour tree.



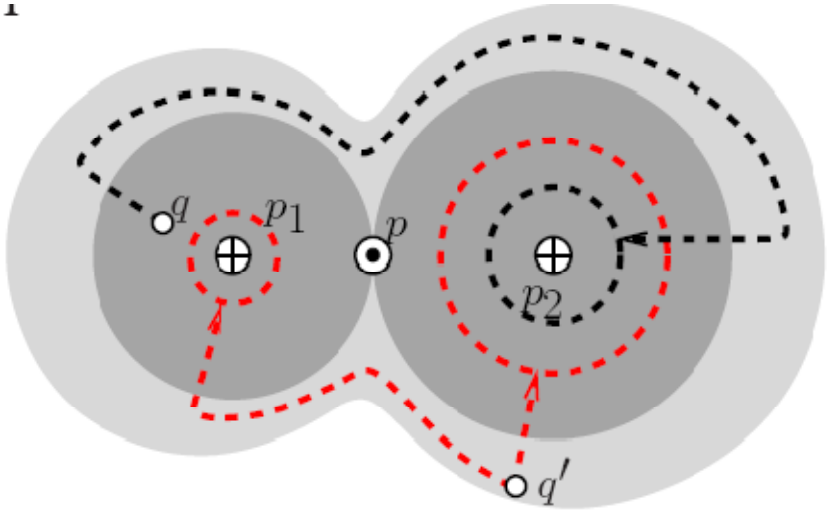
Info stored on each node

- Each node w stores
  - MIN/MAX value inside and outside the contour component through w.
  - The same info about neighboring saddles.
- Such info can be propagated by another top-down and bottom-up sweep.



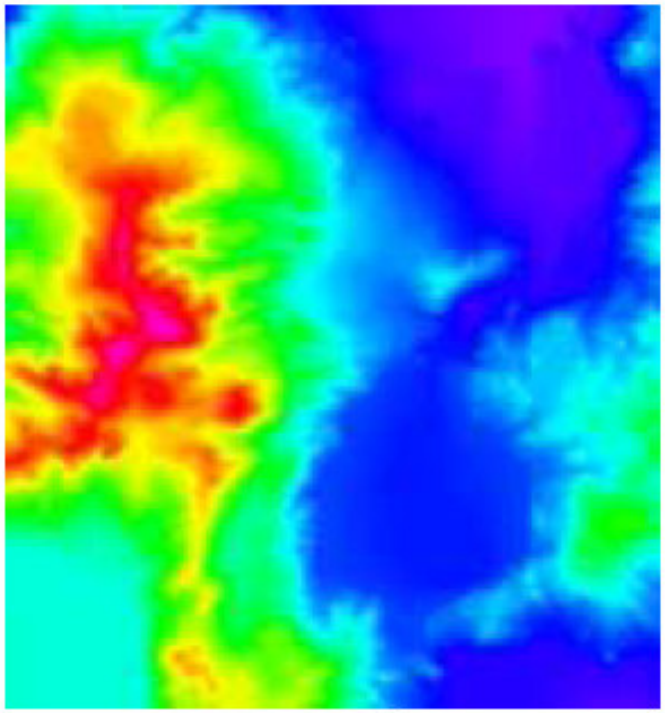
Gradient descent to find all contours

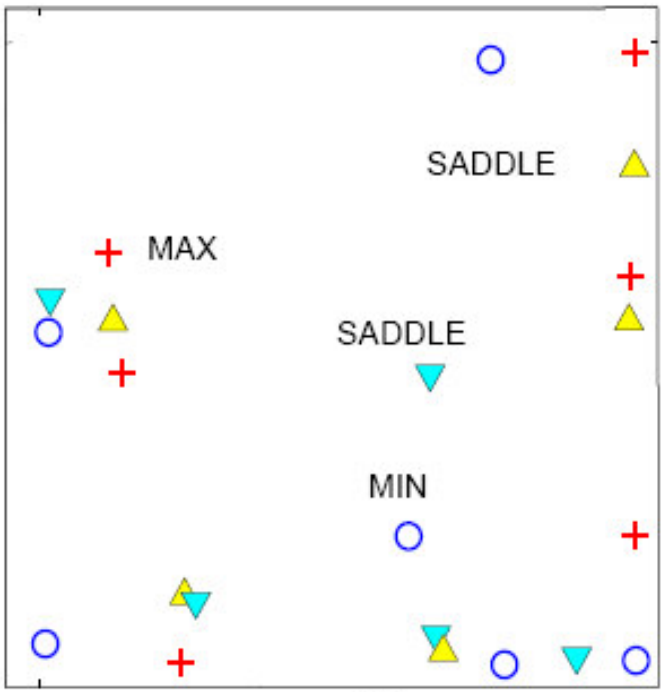
- At each node w checks
  - Is x inside the [MIN, MAX] of the interior/exterior of the contour through w? If so, send a msg following gradient descend/ascend.
  - Is x inside the [MIN, MAX] of each of the upcoming saddle component? If so, send 1 or 2 msgs.



A few notes

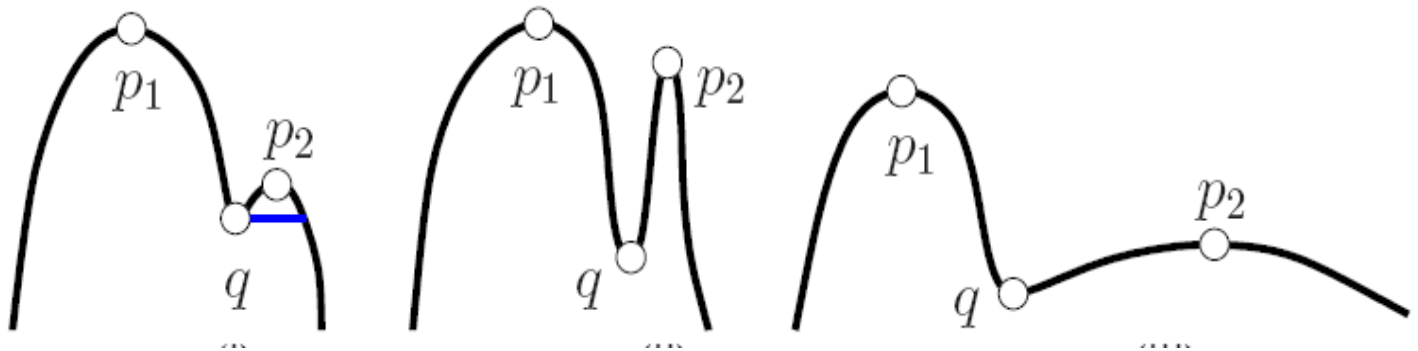
- Construction cost: 4 global sweeps, O(n).
- Gradient descent: discover all components in an iso-contour.
- Low-value routing: consult the contour tree to find a path.





Noises

- Smooth out the signal by some local averaging.
- Define a noisy max/min by its height, and/or spread.
- Remove noises.



Summary and future work

- Track spatial structures.

- On-going work:
  - Maintain a time-varying contour tree.

    (maintain saddle contours over time)
  - Online robot navigation and discovery of the spatial structure.

References

- Xianjin Zhu, Rik Sarkar, Jie Gao, Joseph S. B. Mitchell, Lightweight Contour Tracking in Wireless Sensor Networks, to appear in INFOCOM'08.

- Rik Sarkar, Xianjin Zhu, Jie Gao, Leonidas Guibas, Joseph S. B. Mitchell, Iso-Contour Queries and Gradient Descent with Guaranteed Delivery in Sensor Networks, to appear in INFOCOM'08.

Questions or comments?

- http://www.cs.sunysb.edu/~jgao
- Email: jgao@cs.sunysb.edu