Light-weight Contour Tracking in Wireless Sensor Networks

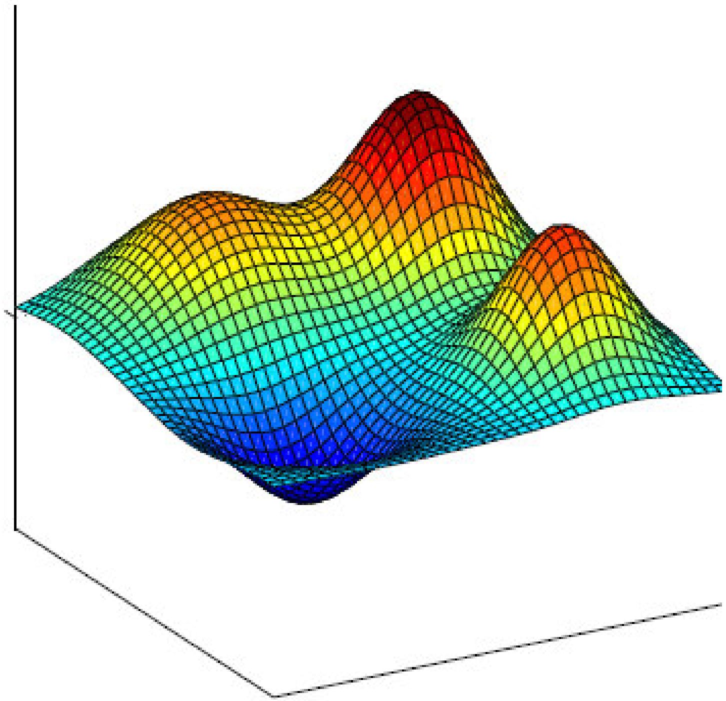
Xianjin Zhu **Rik Sarkar** Jie Gao Joseph S. B. Mitchell

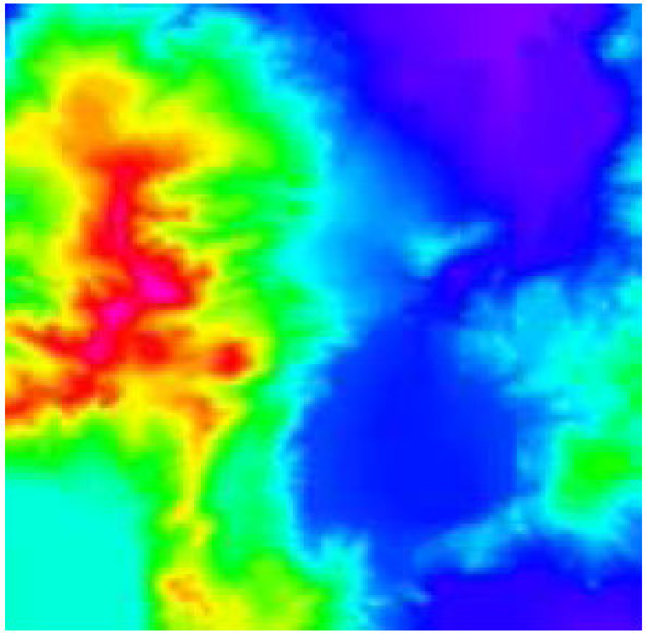
STONY BROOK UNIVERSITY

INFOCOM' 08

Motivation

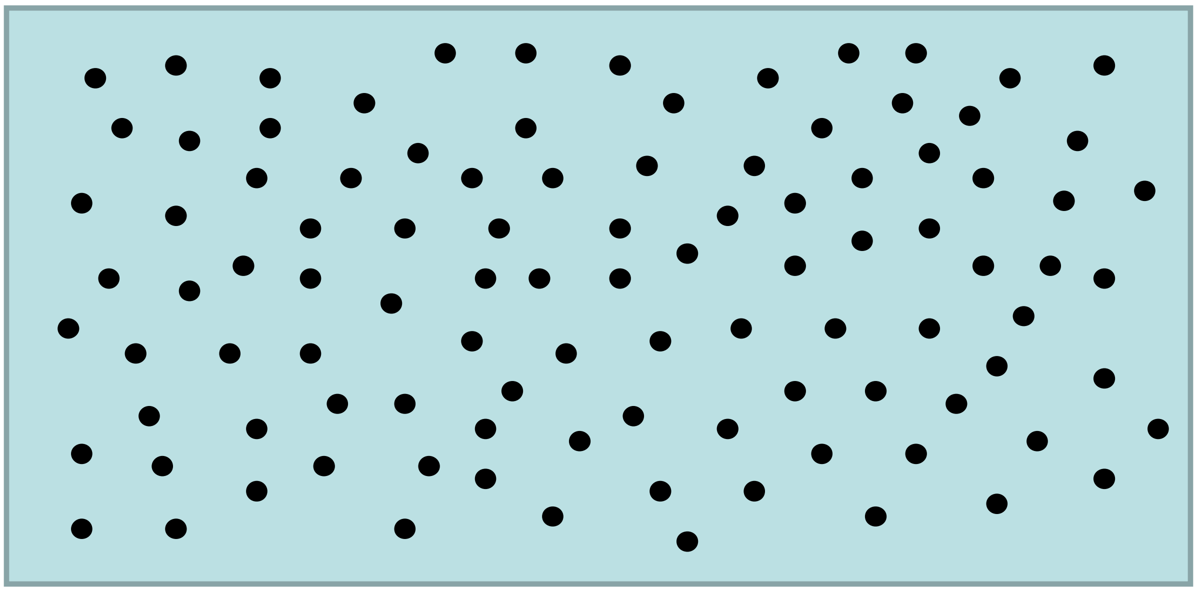
- Sensor network: sense and monitor the physical world (temperature, traffic density, pollution level, etc).





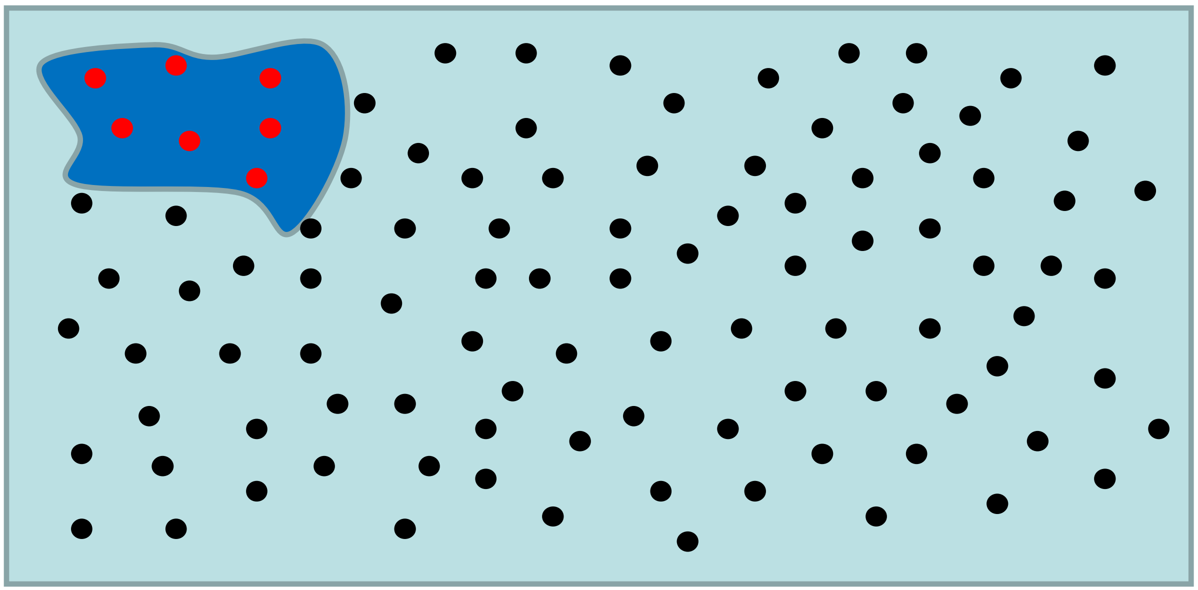
Motivation

- Time-varying 2-D signal field
- Example application scenario
  - Chemical pollution



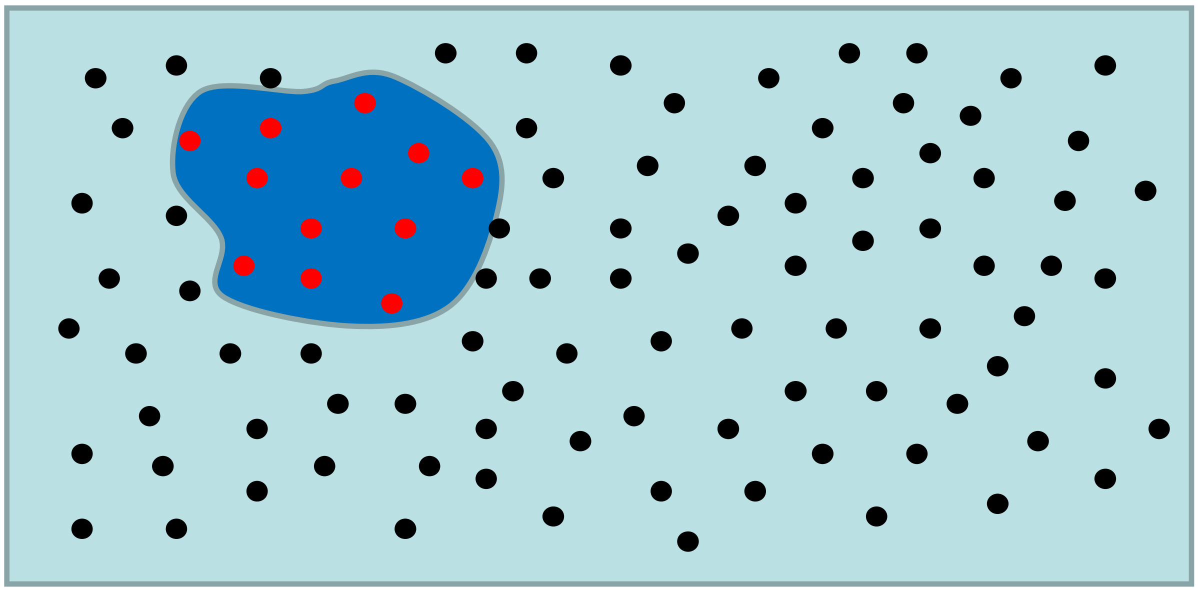
Motivation

- Time-varying 2-D signal field
- Example application scenario
  - Chemical pollution



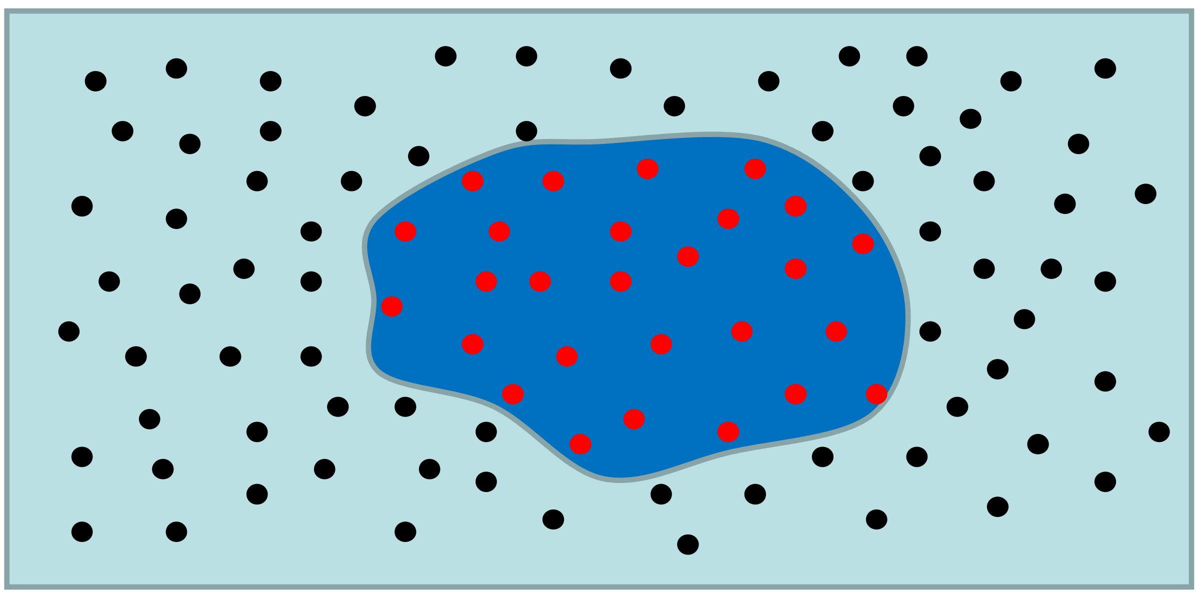
Motivation

- Time-varying 2-D signal field
- Example application scenario
  - Chemical pollution



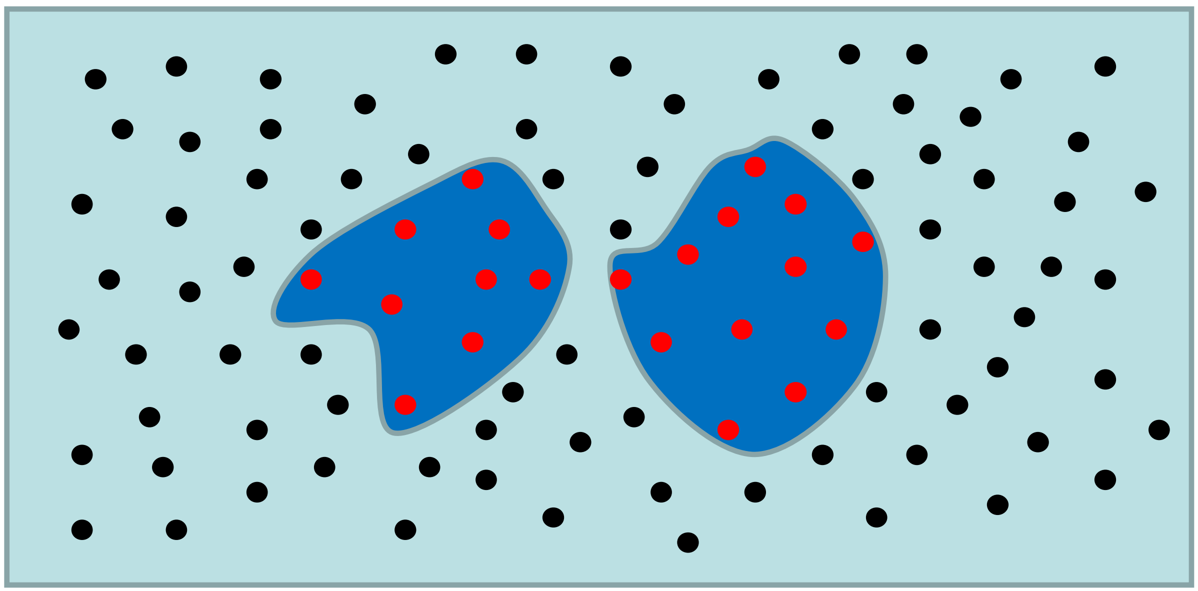
Motivation

- Time-varying 2-D signal field
- Example application scenario
  - Chemical pollution



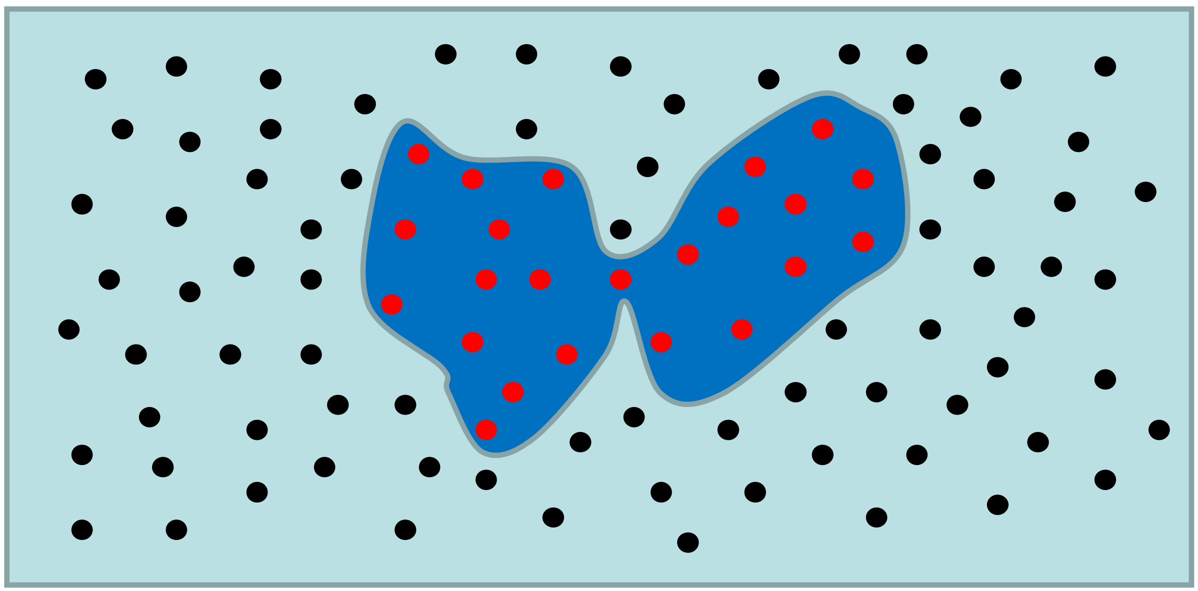
Motivation

- Time-varying 2-D signal field
- Example application scenario
  - Chemical pollution



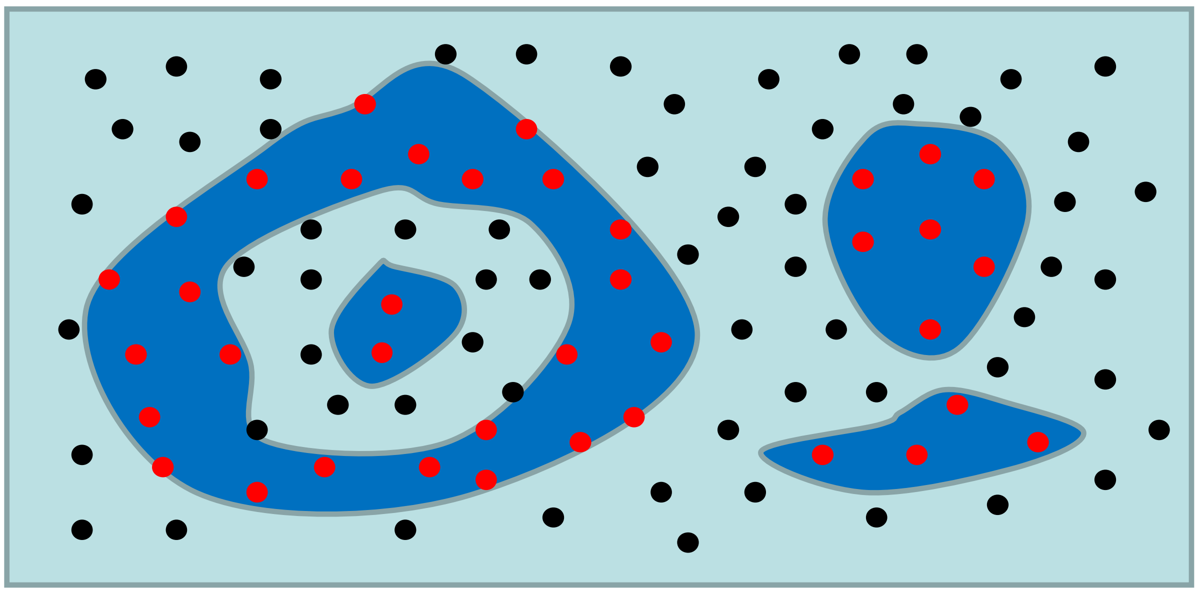
Motivation

- Time-varying 2-D signal field
- Example application scenario
  - Chemical pollution



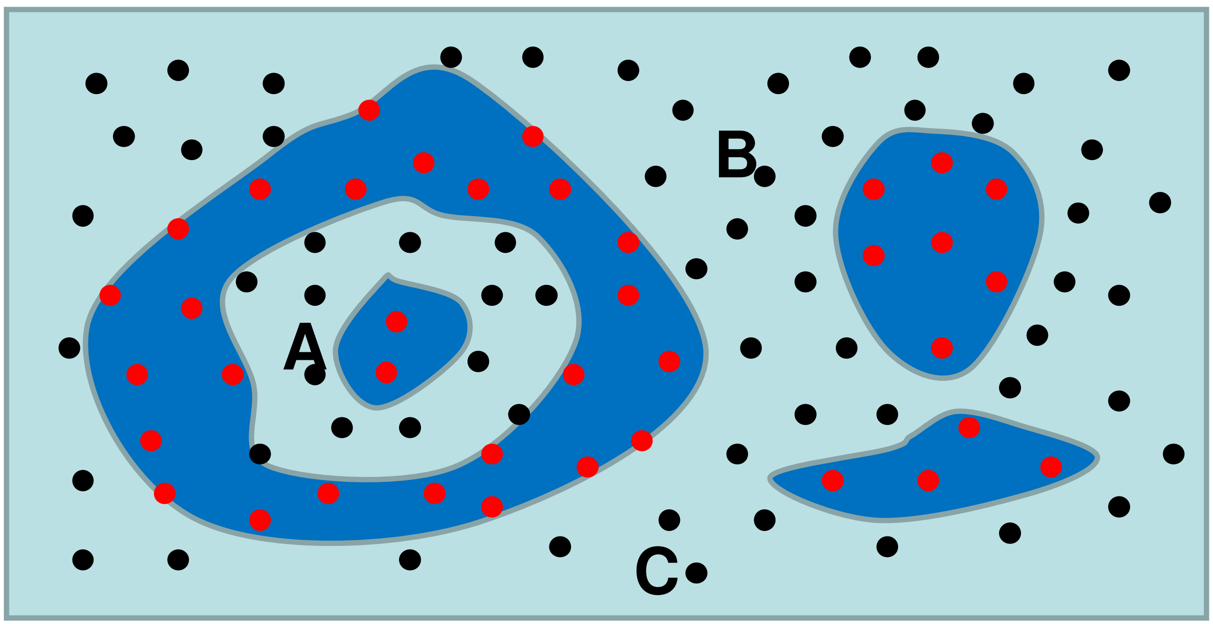
Motivation

- Time-varying 2-D signal field
- Example application scenario
  - Chemical pollution



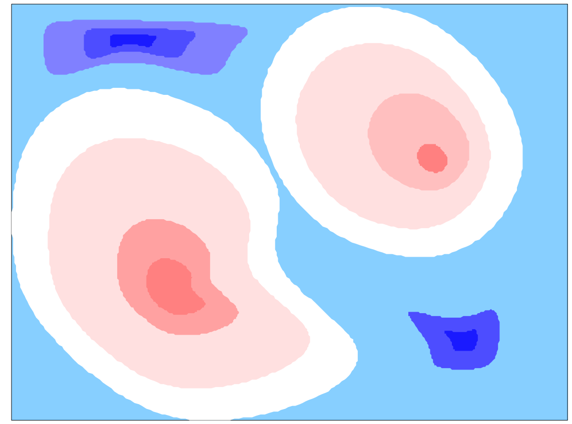
Motivation

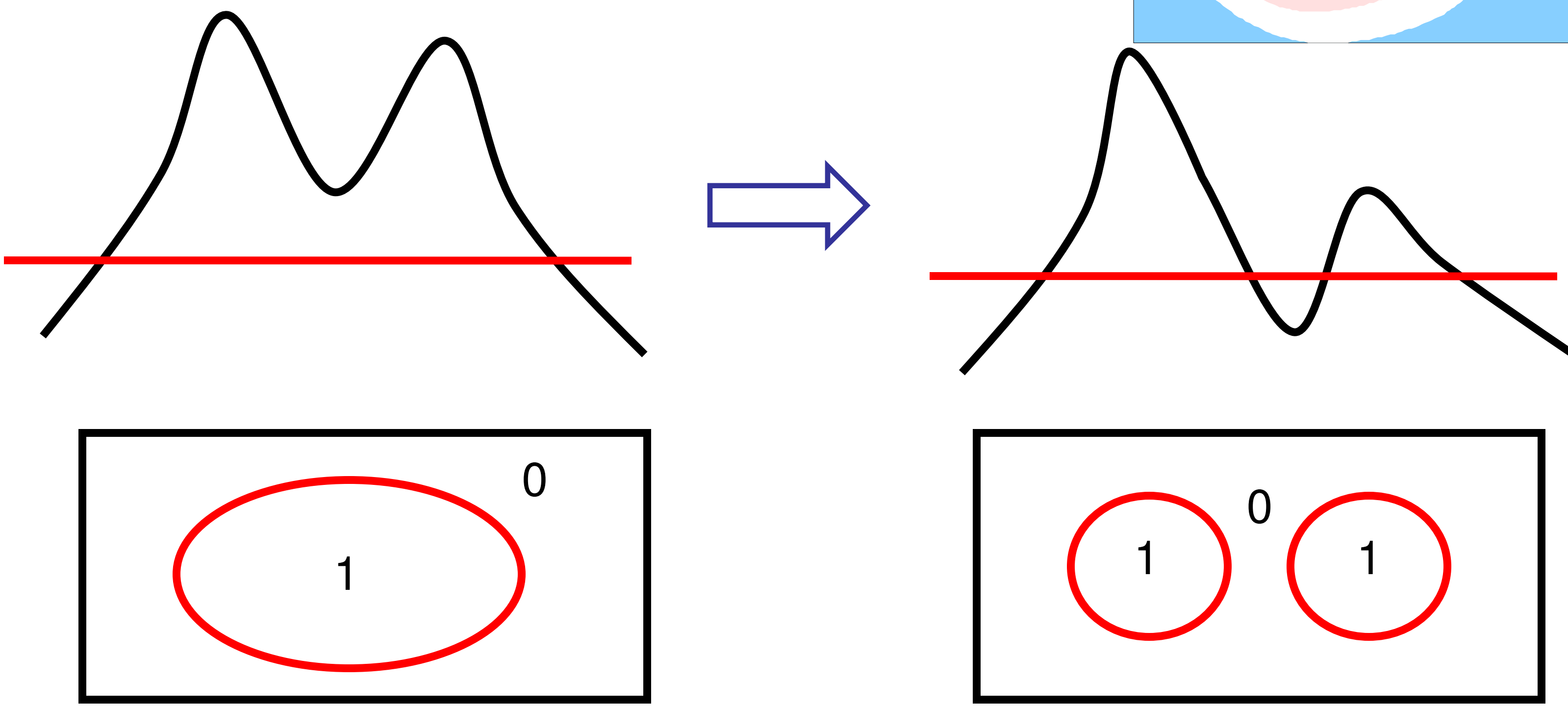
- Topological features are important
- Queries:
  - Is there a safe path from B to A? C to B?
  - Is a location surrounded by chemical contaminations?



Contour tracking problem

- Track contours at a threshold of interest - below/above thresh (0/1).
- Capture their topological features as contours evolve over time





Related works

- Target tracking [Guibas 2002, Zhao et al. 2002, Aslam et al. 2003, Liu et al. 2004, Kim et al. 2005, He et al. 2006, Shrivastava et al. 2006]
  - Track individual targets
  - Few works on tracking a continuously deforming blob or groups of targets
- Boundary detection [Fekete et al. 2005, Wang et al. 2006, Funke et al. 2006, Kroller et al. 2006]
  - Can be used in static scenario
  - Periodically running boundary detection in dynamic scenario is inefficient.

Our contributions

- Light-weight distributed algorithm to track time-varying contours

- Capture topological features

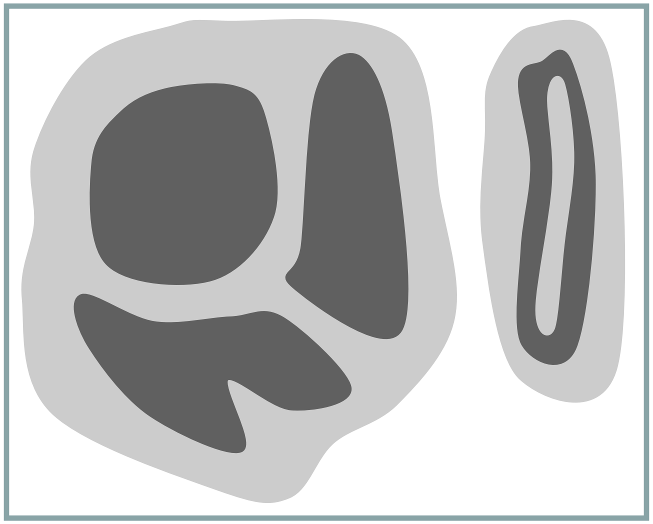
- Low communication cost
  – proportional to the change in the input

Outline: our approach

- Network setting & concepts
- Challenges
- Contour tracking algorithm
- Theoretical results
- Simulation results

Network setting & concepts

- Binary sensor model:
  - Color
    - **BLACK**: all neighbors high
    - **WHITE**: all neighbors low
    - **GRAY**: neither BLACK nor WHITE (mixed high and low)

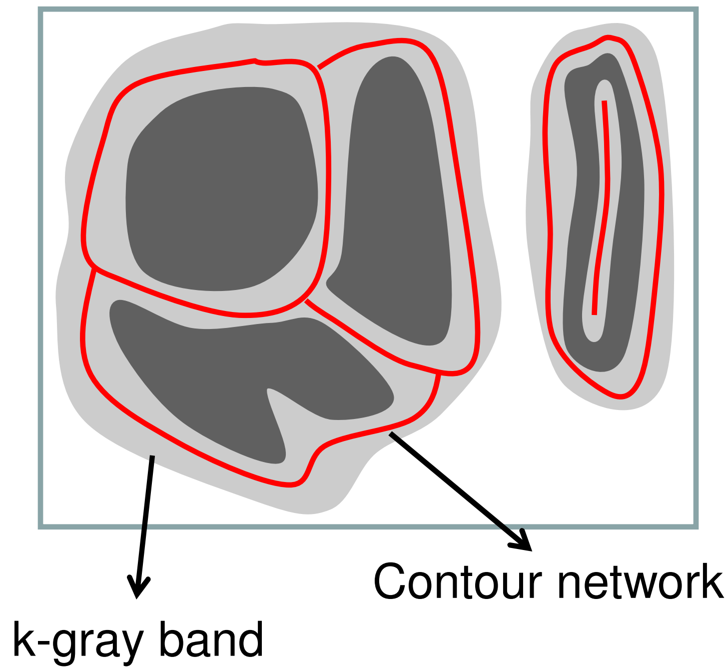


- We want to track the *Black* boundaries.
- Robustness: resilient to outliers and ambiguity.
- Black regions and white regions are separated by gray.
- Black regions are bounded by contours of threshold.

Goal of the algorithm

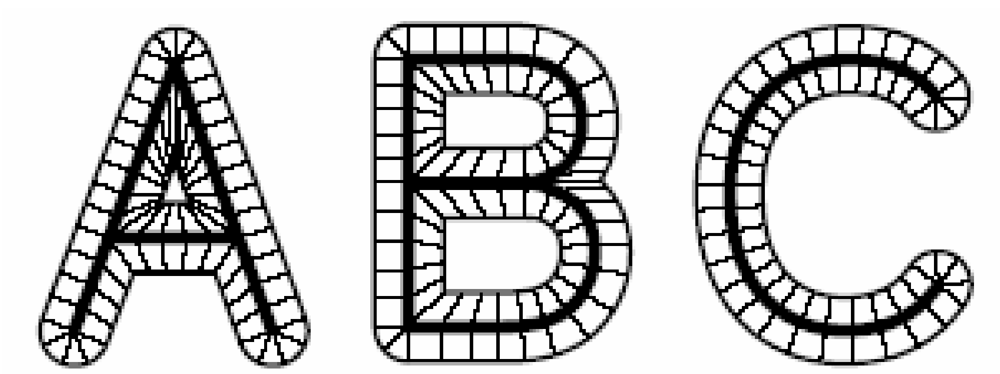
- K-gray band:
  - a set of gray nodes at most k-hop from BLACK region

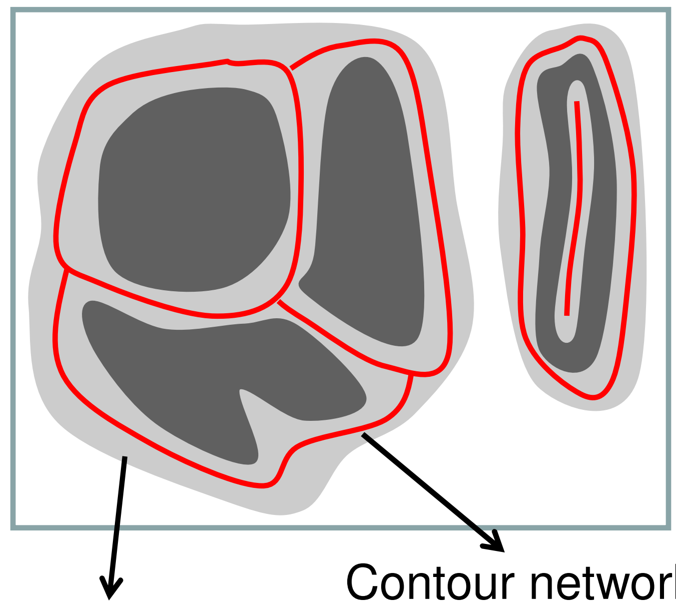
- Contour network: Graph to capture topological features of contours



Goal of the algorithm

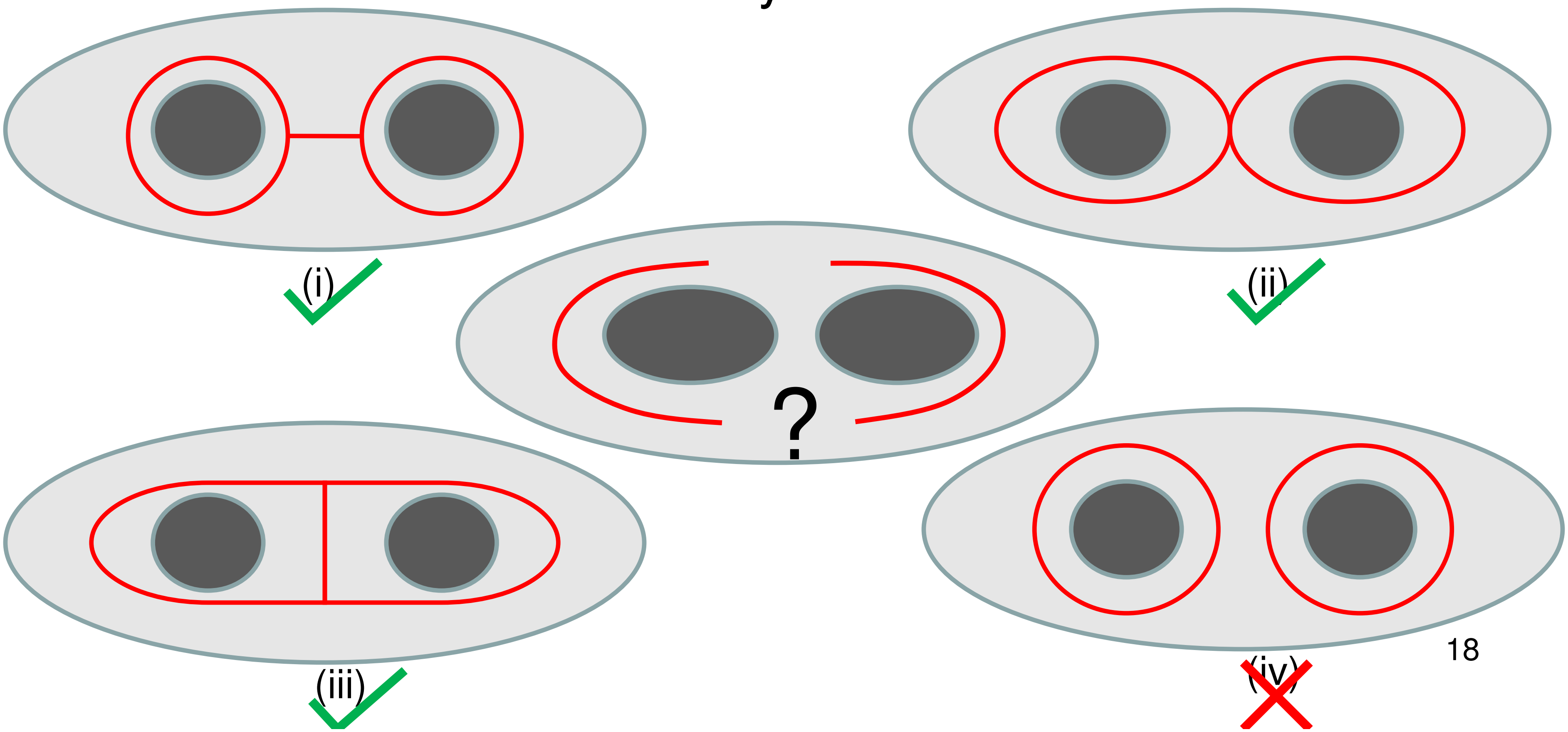
- Deformation retract:
  - A thinner version in subspace, with same homotopy. There is a continuous deformation taking the space to the retract.
  - Contour network: skeleton of k-gray band





Challenges

- Hard to tell if a contour network is valid from local view
- Same contour topology may have multiple valid deformation retract with totally different local view



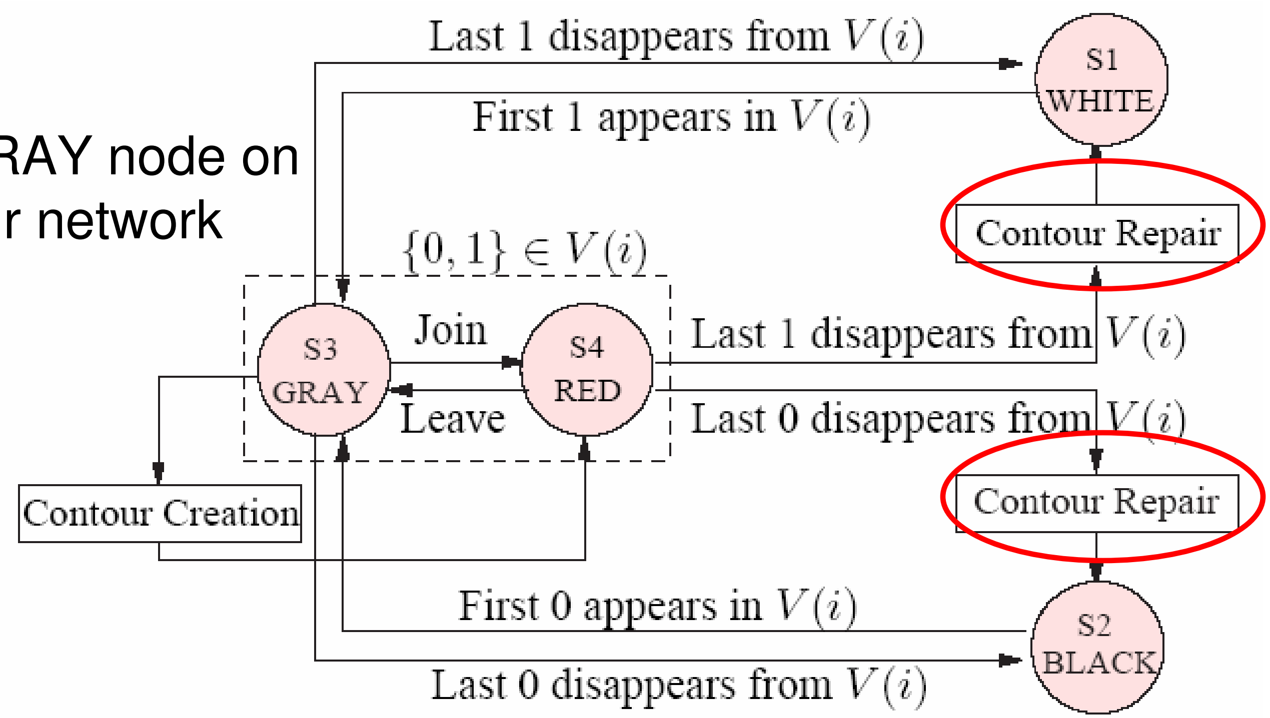
Challenges

Network setting	Goal
Each sensor node only has local information Limited resources	Maintain graphs with identical topological information
↓	↓
Naturally require distributed efficient algorithm	global property

Contour tracking algorithm

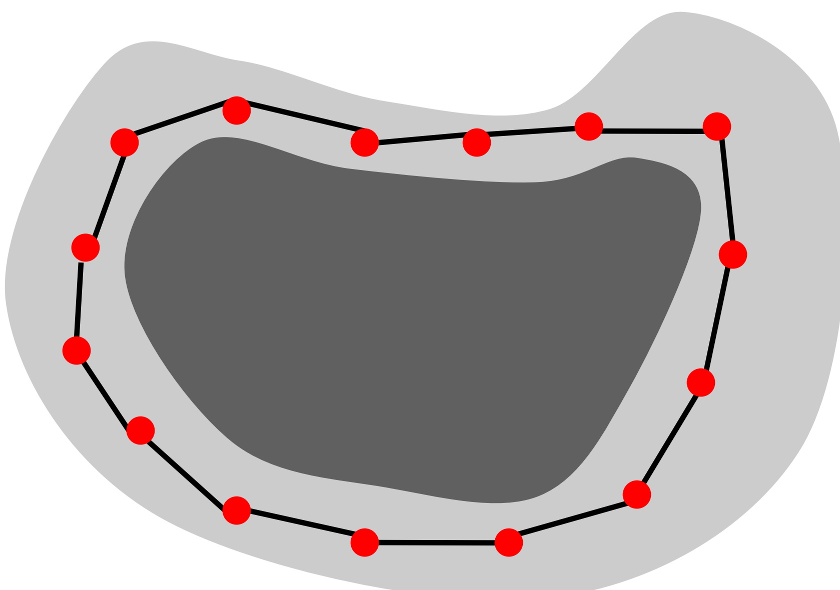
- When change occurs:
  - freeze the valid segments in the old contour graph
  - only repair the contour network where it is broken
- Automaton runs at each sensor

RED: a GRAY node on the contour network



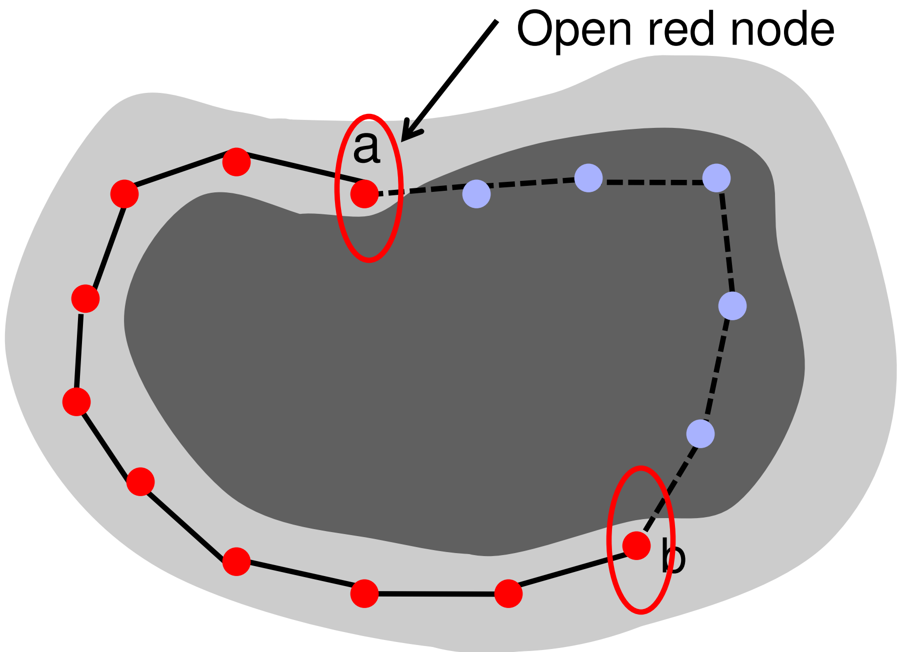
Contour Repair

- The simplest case: repair of a single contour cycle



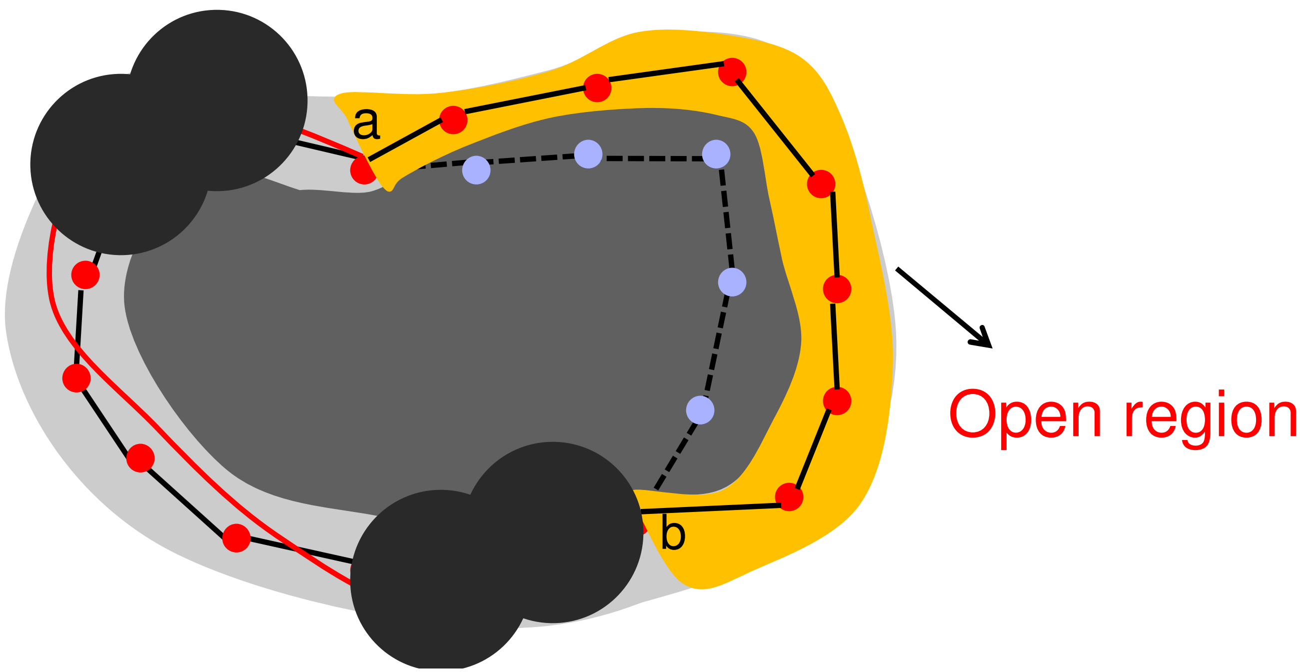
Contour Repair

- The simplest case: repair of a single contour cycle
- Open red nodes take responsibility of repair
  - Red nodes at edge of broken contour



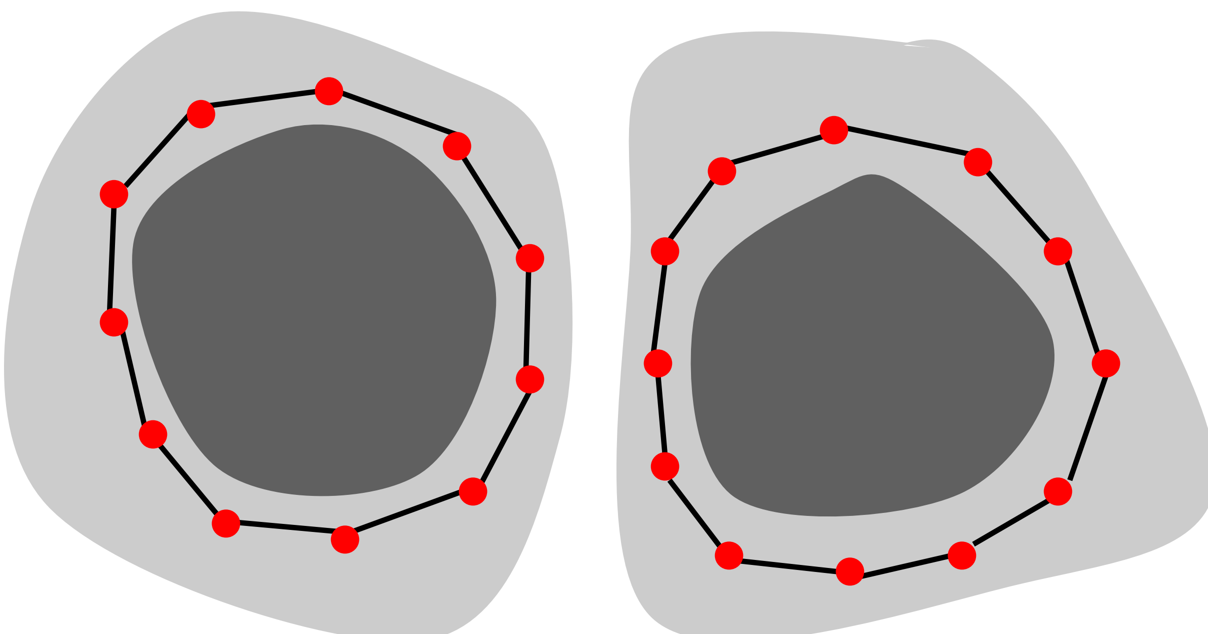
Contour Repair

- Which direction to send repair messages?
  - sensor nodes have no sense of orientation
- The k-hop neighbors of the closed red segments block the propagation of repair messages.



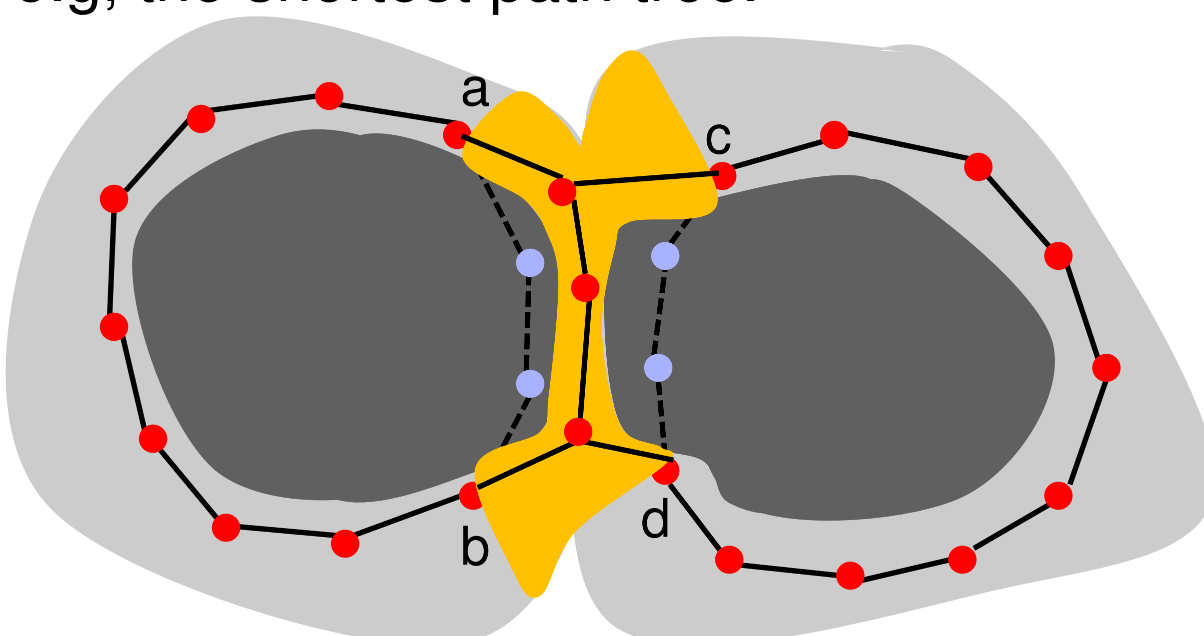
Contour Repair

- Simultaneous repair, merging and splitting
  - May encounter multiple RED nodes, which RED node to connect to?



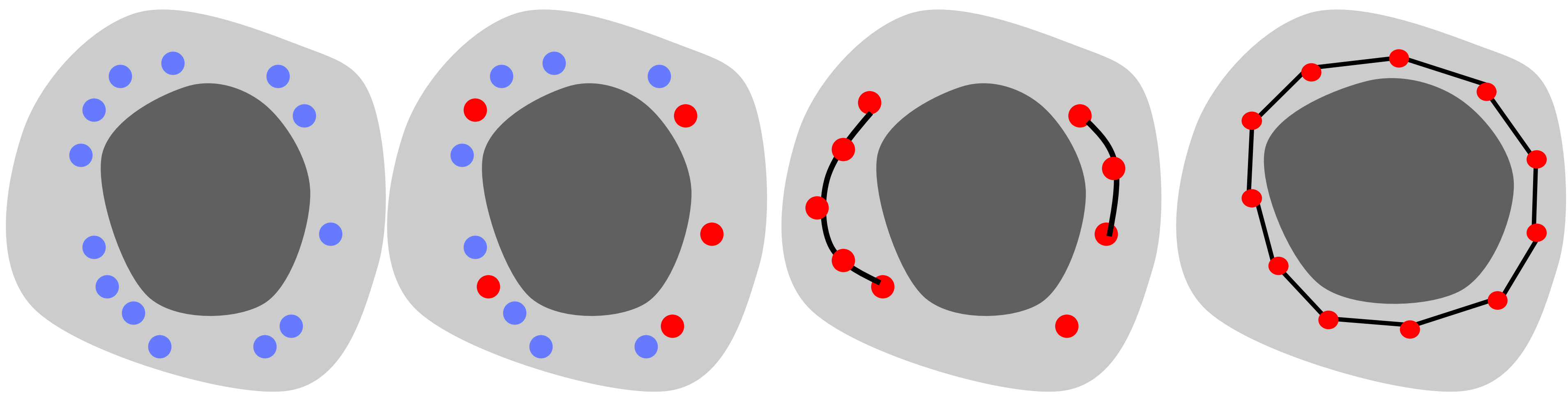
Contour Repair

- Simultaneous repair, merging and splitting
  - May encounter multiple open RED nodes, which RED node to connect to?
  - Connect by a (non self-intersecting) spanning tree, e.g, the shortest path tree.



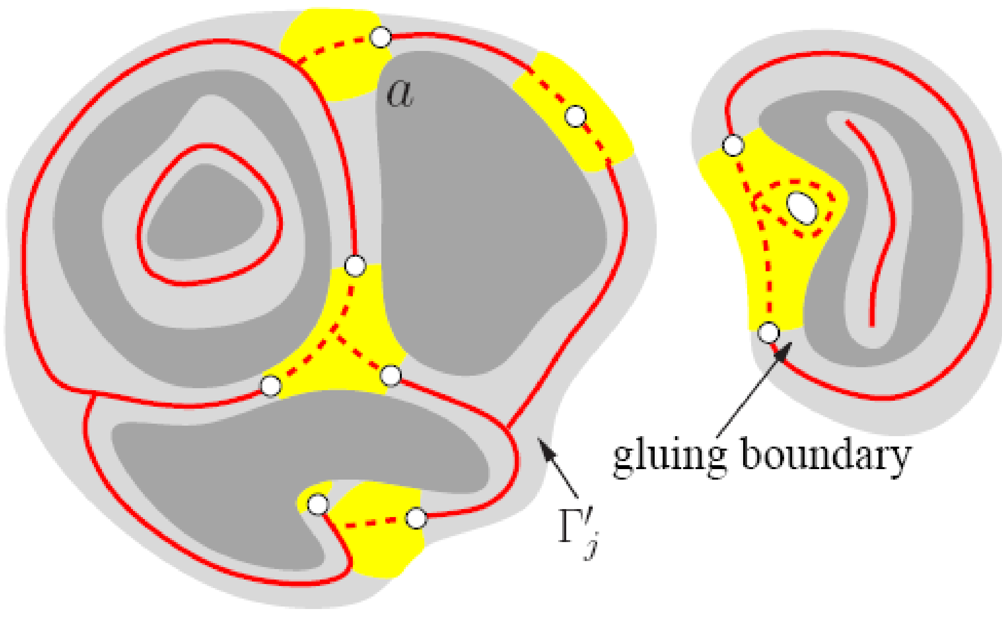
Contour creation

- Triggered at some BLACK nodes
  - have a GRAY neighbor but cannot see RED nodes in its k-hop neighborhood
- GRAY neighbors start to create a new contour
  - Select leaders within k-hop neighborhood
  - Form short chain with length > k
  - Start contour repair



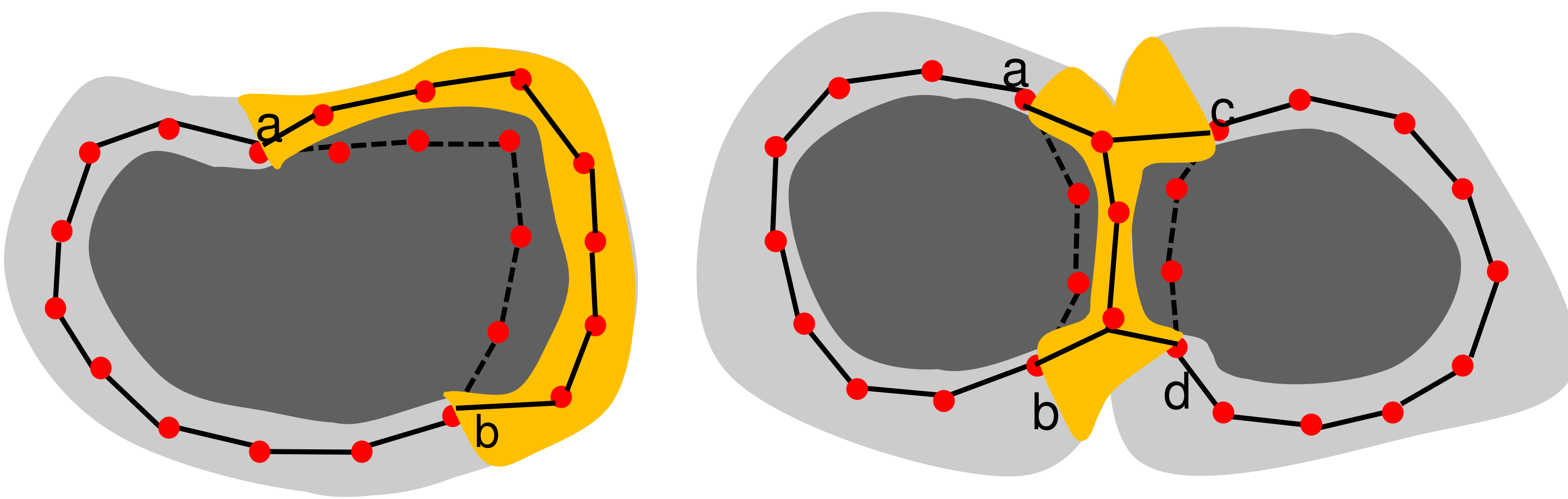
Summary of algorithm

- Valid segments of contour network are still usable
- Repair only happens where the contour network is broken
- Repair connects all red nodes within a open neighborhood through a spanning tree.
- Augmented algorithm deals with small holes inside the k-gray band



Theoretical results

- Theorem: The contour network is a deformation retract of the k-gray band, when the system stabilizes.
- Sketch of proof:
  - Existing contour network is a deformation retract of the segment of k-gray band it resides in.
  - Repaired new contour network is a deformation retract of open neighborhood.
  - The boundaries align correctly.



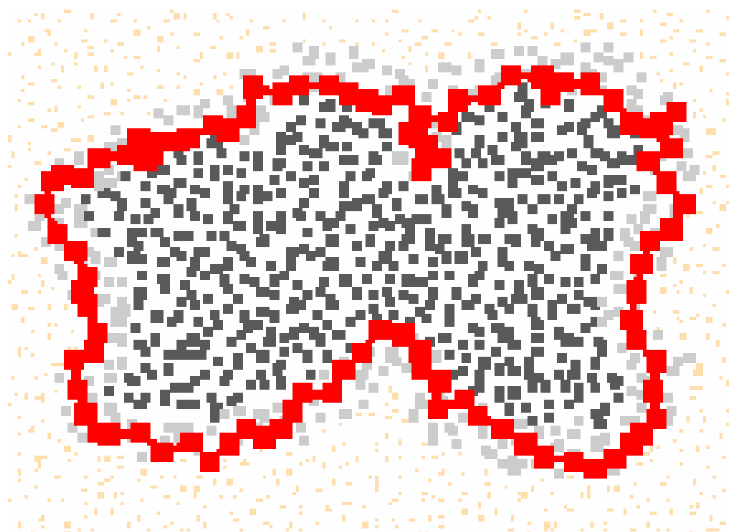
Simulations

- Setup:
  - 4000 nodes distributed in a 500*500 field
  - Simulate dynamic changes among a sequence of stabilized states of a contour field
  - Changes happen in a random order between two stabilized states

Simulations

- Contour merge/split

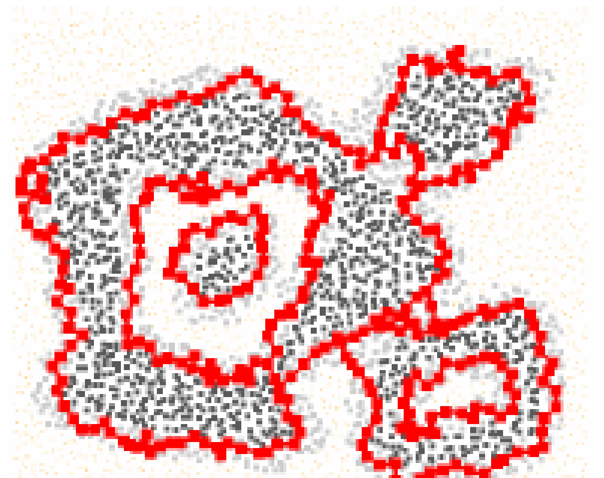
Two black regions move closer. Their gray bands meet each other and (multiple) “bridges” are built up.



Black regions themselves merge together.

Simulations

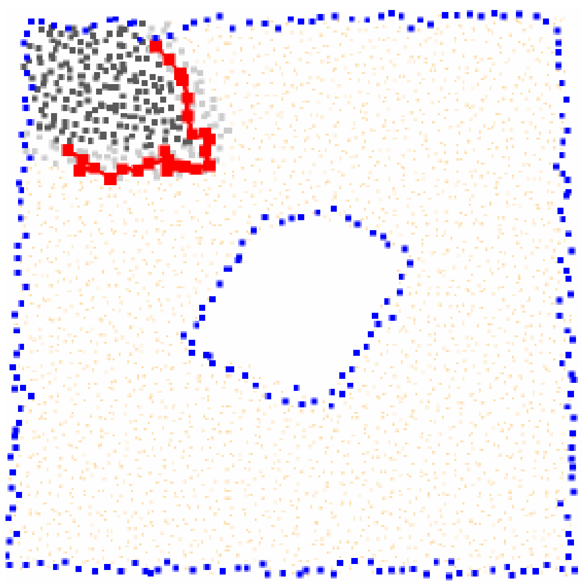
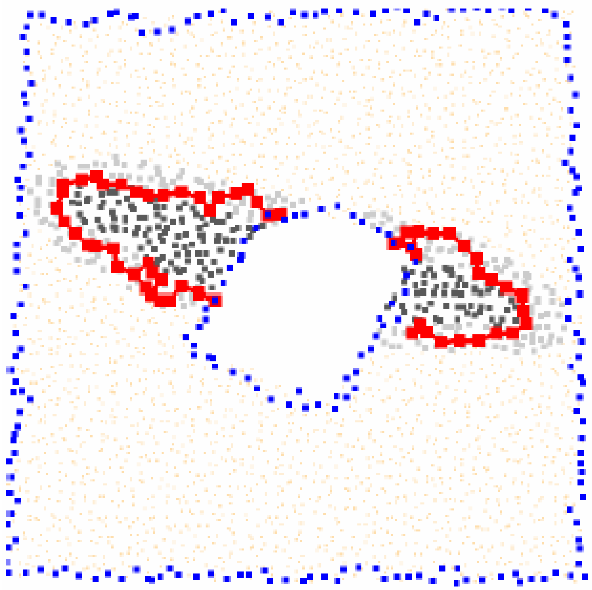
- Nested contours


(i)


(ii)

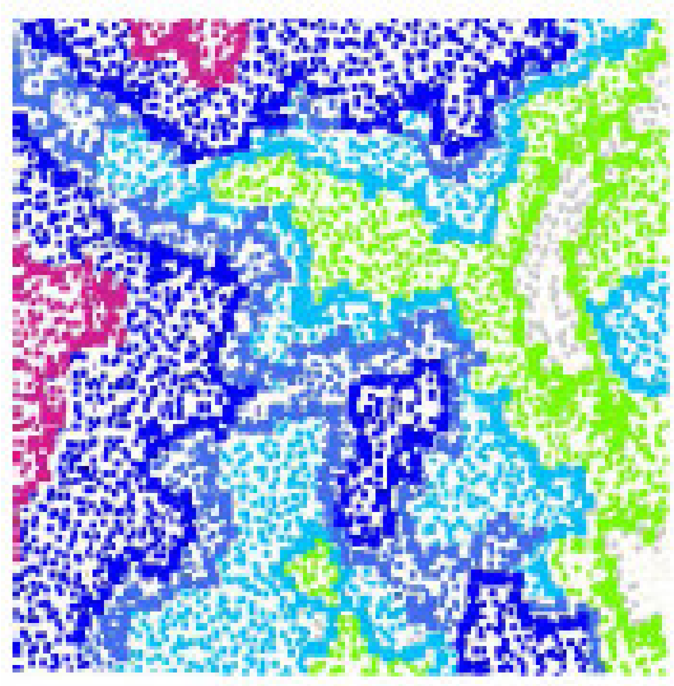
Simulations

- Contours pass through a hole

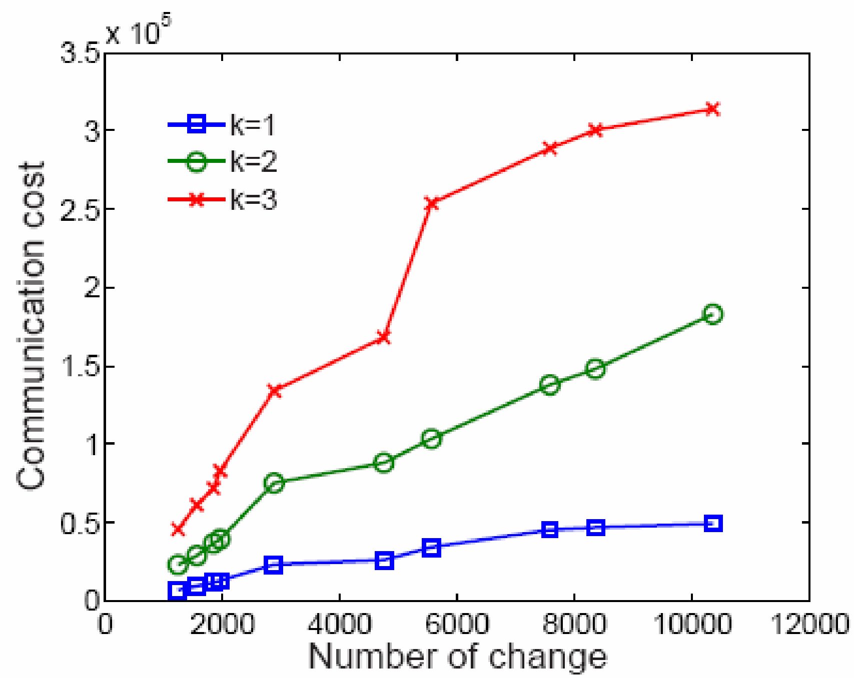
Simulations

- Multi-level contours
  - Apply the single-level contour tracking algorithm at each level independently



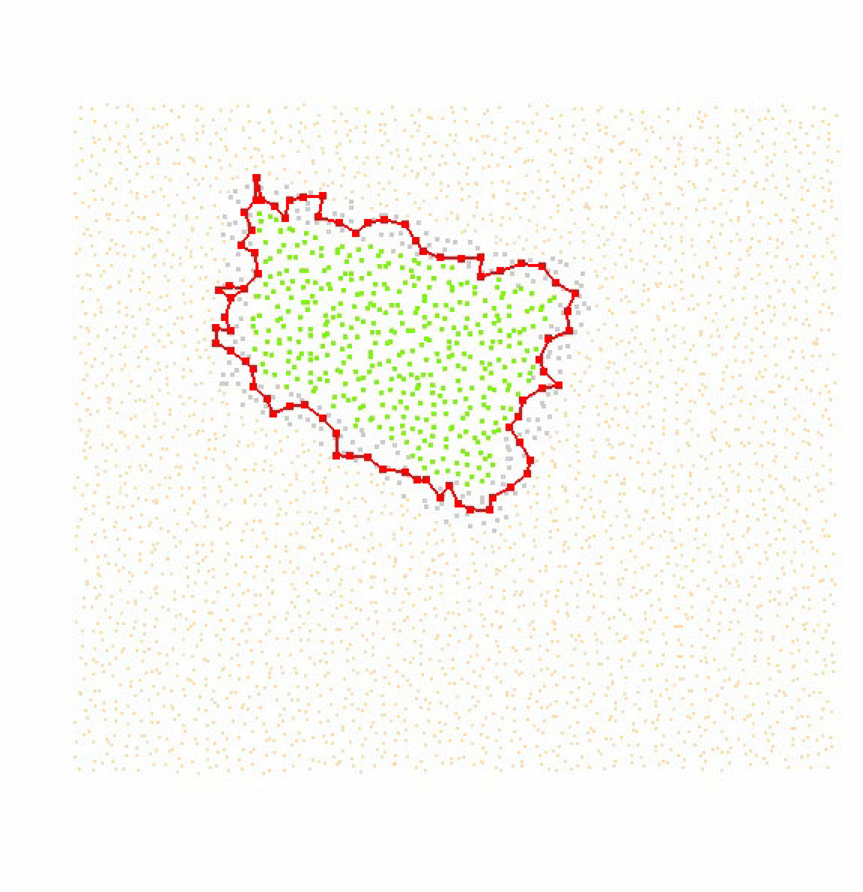
Simulation

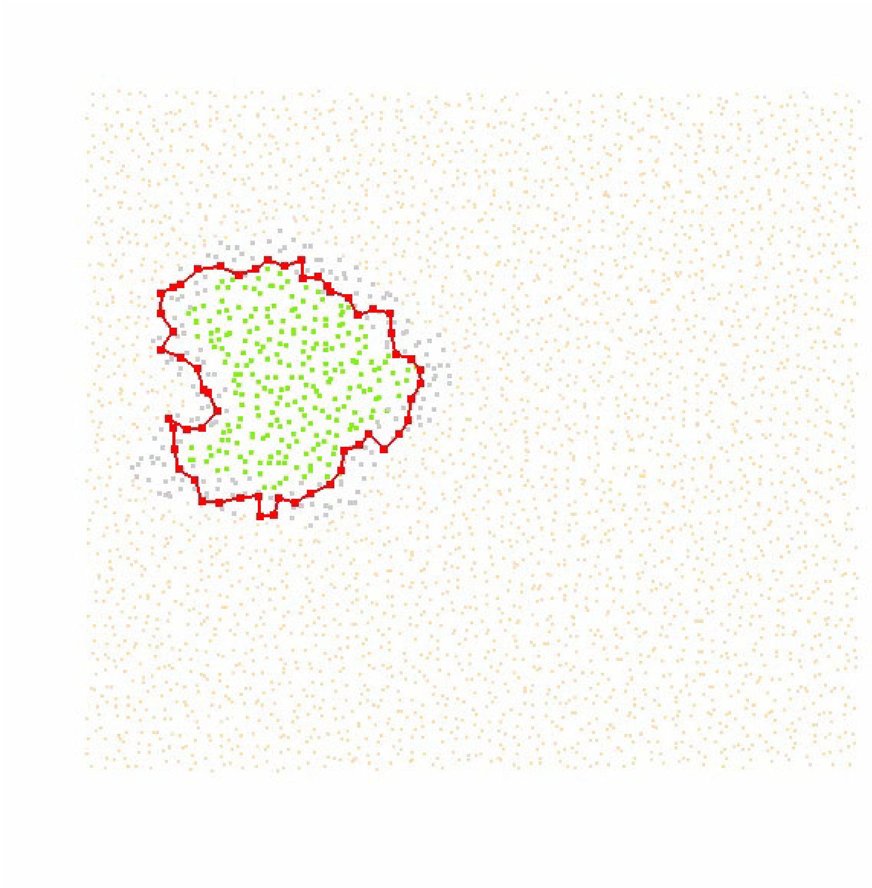
- Communication cost:
  - Proportional to number of changes



Simulations

- Video clips



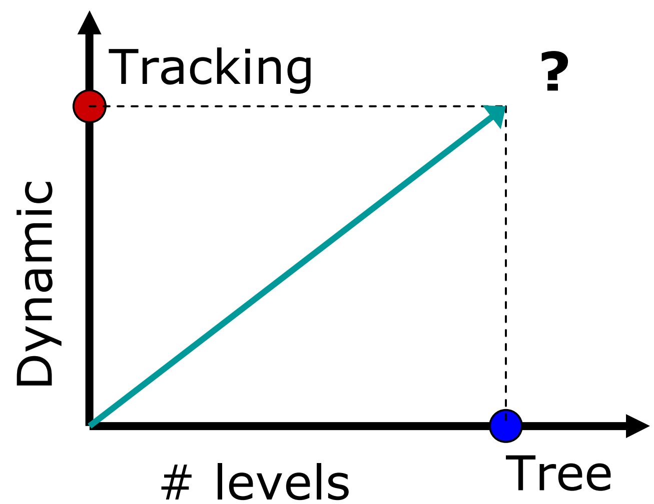


Conclusions

- 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 et al. 2007].

Future Works

- Explore the applications of contour tracking
  - Real-time response and emergency rescue
  - Direct vehicles to alleviate traffic jam
- Combine with our concurrent contour tree work



Thank you !

- Questions & Comments?