Distributed Systems

Localization

Rik Sarkar

University of Edinburgh

Fall 2014

Find location of a device

- On a map
  - Use in navigation
- Use relative location for services
  - Print to the nearest printer
  - Find a friend nearby
- Location based sensing and services
  - Find the the nearest printer
  - Find the status of coffee machine on floor 3
  - Find temperature in room 202

GPS does not always work

- Does not work indoors
- Takes too much energy
- Expensive
  - Inappropriate for mass sensing with cheap sensors

GPS does not always work

- Does not work indoors
- Takes too much energy
- Expensive
  - Inappropriate for mass sensing with cheap sensors
- Wireless communication can be used for localization
  - E.g. cell signal based localization, wifi based localization (on google maps)

Localization

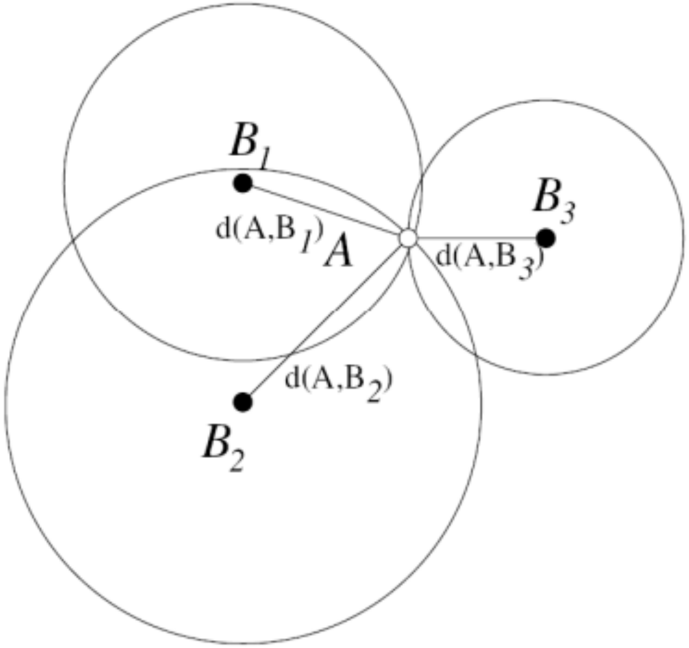
- Possible Output
  - Global location (like GPS)
  - Relative location (with respect to other nodes)
- Possible input
  - Connectivity: neighbors of each node
  - Distance to a neighbor
    - Or a specialized beacon (e.g. cell tower)
  - Angle to a neighbor
    - Angle to a beacon

Distance Measure

- Received signal strength indicator (RSSI)
  - Signals get weaker with distance
  - Problems: Not uniformly. Noisy. Signal can reflect from objects and create multi-path effects

Time of arrival

- Used in GPS
- Needs synchronization
  - Can be relaxed with round trip time

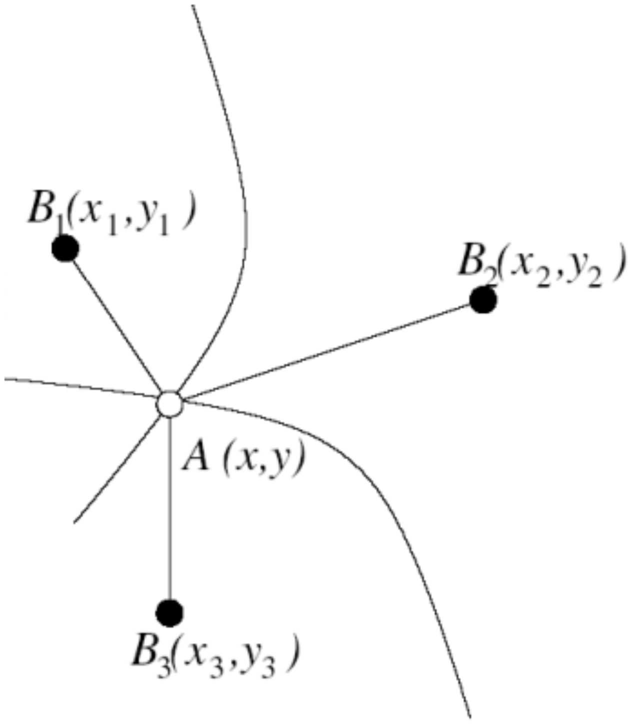


Time Difference of arrival

- Beacons B1 and B2 transmit simultaneously
- Time difference at A gives a hyperbola

$$\sqrt{(x-x_1)^2+(y-y_1)^2}-\sqrt{(x-x_2)^2+(y-y_2)^2}=\delta_d$$

- B2 & B3 gives another hyperbola
- Intersection is A

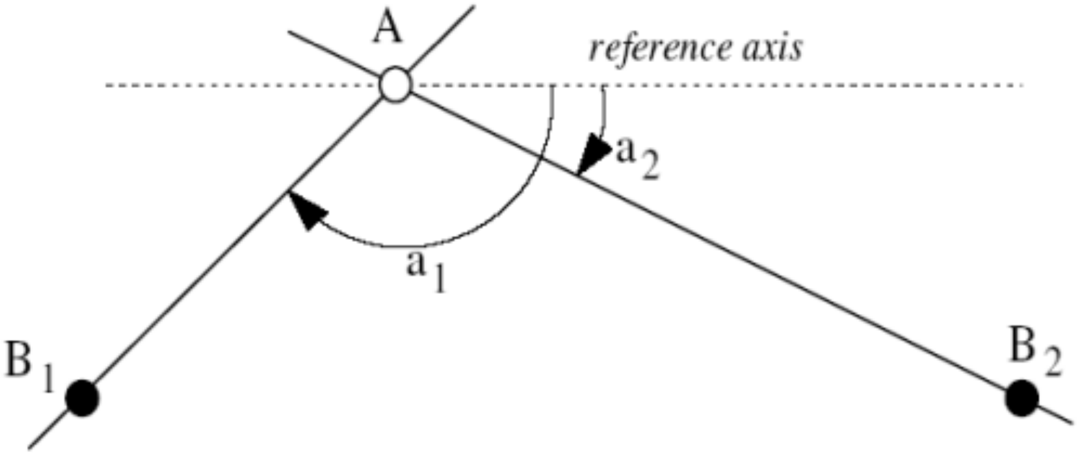


Angle of arrival

- Using directional antenna
  - – Transmit only in a direction
  - – Propagations in other directions cancelled by electronic phase cancellations
  - – Alternatively, use laser communicators

Angle of arrival

- Determine location from angle of 2 beacons

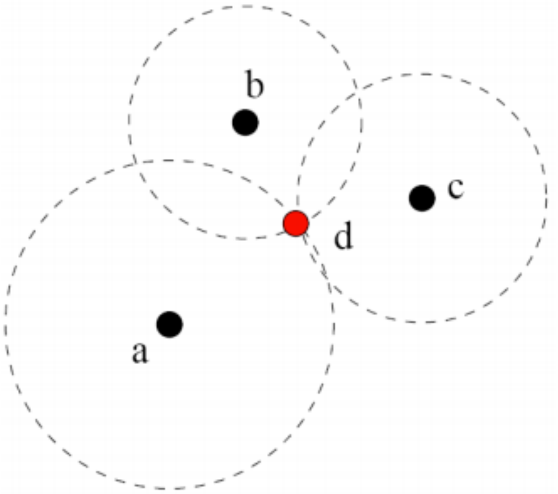


Localization algorithms

- Anchor based
  - Some anchor/beacon nodes know their locations
  - Eg. Using GPS, or manually given
- Anchor free
  - Relative location only
  - More difficult. No point of reference
- Range based
  - Use distance information (no method is perfect)
- Range free
  - No distance info. Only connectivity i.e. neighbors
  - Use hop-count as a measure of distances

Triangulation, trilateration

- Anchors advertise their coordinates and transmit signals
- Other nodes use this to measure distances



Triangulation, trilateration

- Problem: distance measures are not accurate
- We now have an optimization problem
- E.g. minimize mean square error

Indoor localization

- Important in applications
  - In which room is the user?
  - On which floor is the user?

- No good solutions from GPS or cellular localization
- Needs to be really accurate
  - Room 1 vs room 2

Indoor localization systems

- RADAR fingerprinting
  - With RF signals
  - Have 3 (or more) base stations/anchors/access points
  - Offline (pre-processing/training) phase: collect detailed map of base station signals strengths in the building (its fingerprint)
  - Online (localization) phase: match current signal strength to offline readings
  - Significant cost: fingerprinting

Indoor localization systems

- Cricket
  - Anchor nodes with RF & ultrasound
  - Measure distance from time difference of light & sound
  - Use trilateration

Maximal likelihood estimation

- k beacons at (x_i, y_i)
- Say, node 0 to be localized is at (x_0, y_0)
- Suppose the measured distance between node 0 and beacon i is r_i
- Then the error is:

$$r_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

Linearization

- Ideally, error = 0
- So,

$$({x_0}^2 + {y_0}^2) + x_0(-2x_i) + y_0(-2y_i) - r_i^2 = -x_i^2 - y_i^2$$

- Subtract the kth equation to remove square terms:

$$2x_0(x_k - x_i) + 2y_0(y_k - y_i) = r_i^2 - {r_k}^2 - x_i^2 - y_i^2 + x_k^2 + y_k^2$$

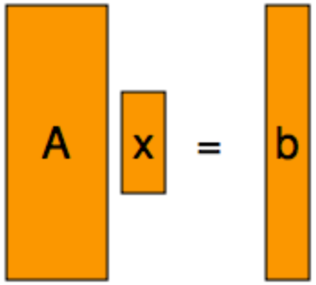
Minimizing mean square error

- When we have many anchors, the system is overconstrained

$$Ax = b$$

$$b = \begin{bmatrix} r_1^2 - r_k^2 - x_1^2 - y_1^2 + x_k^2 + y_k^2 \\ r_2^2 - r_k^2 - x_2^2 - y_2^2 + x_k^2 + y_k^2 \\ \vdots \\ r_{k-1}^2 - r_k^2 - x_{k-1}^2 - y_{k-1}^2 + x_k^2 + y_k^2 \end{bmatrix} \qquad A = \begin{bmatrix} 2(x_k - x_1) & 2(y_k - y_1) \\ 2(x_k - x_2) & 2(y_k - y_2) \\ \vdots & \vdots \\ 2(x_k - x_{k-1}) & 2(y_k - y_{k-1}) \end{bmatrix}$$

$$x = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$



Solve using least square equation

-

$$x = (A^T A)^{-1} A^T b$$

Iterative multilateration

- Suppose not all nodes are in range of beacons
- Find a node that “sees” 3 beacons
- Localize
- This node now can act as a beacon
  - And help localize other nodes

- Problem:

Iterative multilateration

- Suppose not all nodes are in range of beacons
- Find a node that “sees” 3 beacons
- Localize
- This node now can act as a beacon
  - And help localize other nodes

- Problem:
  - Errors accumulate
  - Can get stuck due to impossible to localize configurations

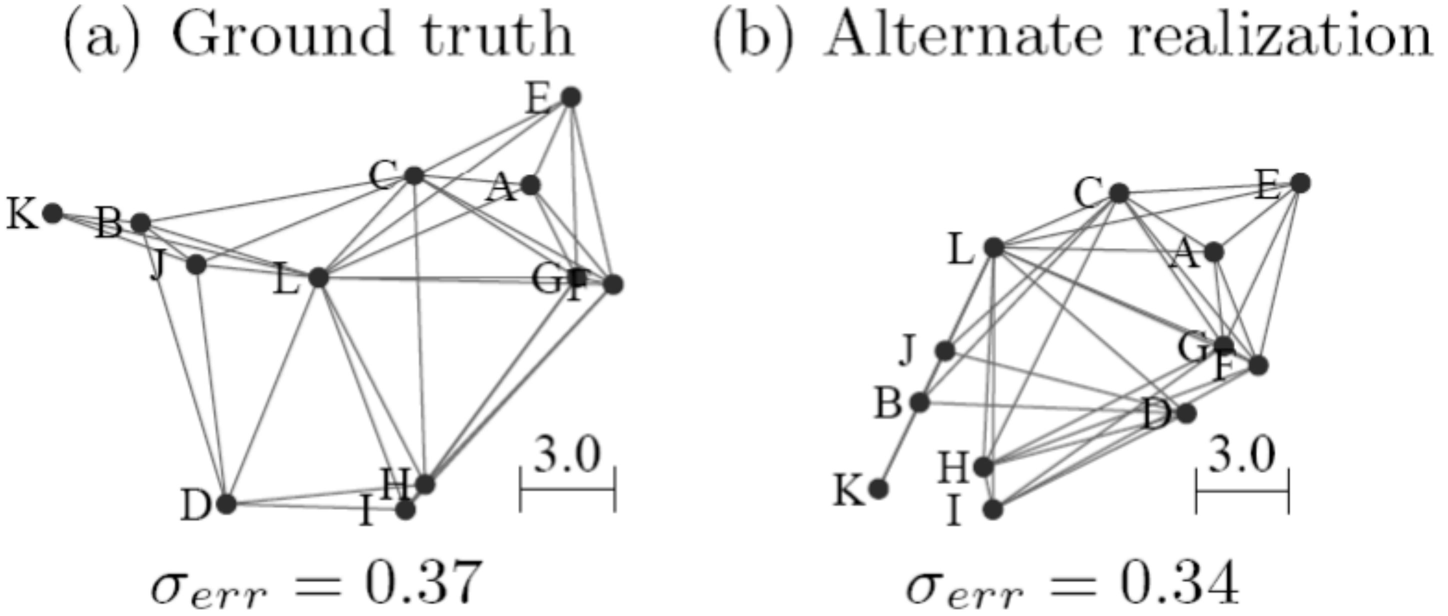
Iterative multilateration

- Suppose not all nodes are in range of beacons
- Find a node that "sees" 3 beacons
- Localize
- This node now can act as a beacon
  - And help localize other nodes

- Problem:
  - Errors accumulate
  - Can get stuck due to impossible to localize error configurations

Ambiguity

- Same set of distance measures can have multiple localizations:



(a) Ground truth

(b) Alternate realization

$\sigma_{err} = 0.37$

$\sigma_{err} = 0.34$

Error of the measured distances from the calculated distances

What about connectivity

- Suppose we know only connectivity
  - Neighbors
  - But not distances (edge lengths)
- Is it possible to find correct locations?

Localization of UDG

- Suppose the network is a unit disk graph
- Given this graph, can we find a valid localization?
  - i.e. a set of locations that satisfies the UDG

Localization of UDG

- Suppose the network is a unit disk graph
- Given this graph, can we find a valid localization?
  - i.e. a set of locations that satisfies the UDG

- The problem is NP-hard
  - Meaning there is no easy way short of trying out all possible locations

Localization of UDG

- Suppose the network is a unit disk graph
- Given this graph, can we find a valid localization?
    - i.e. a set of locations that satisfies the UDG

- The problem is NP-hard
    - Meaning there is no easy way short of trying out all possible locations

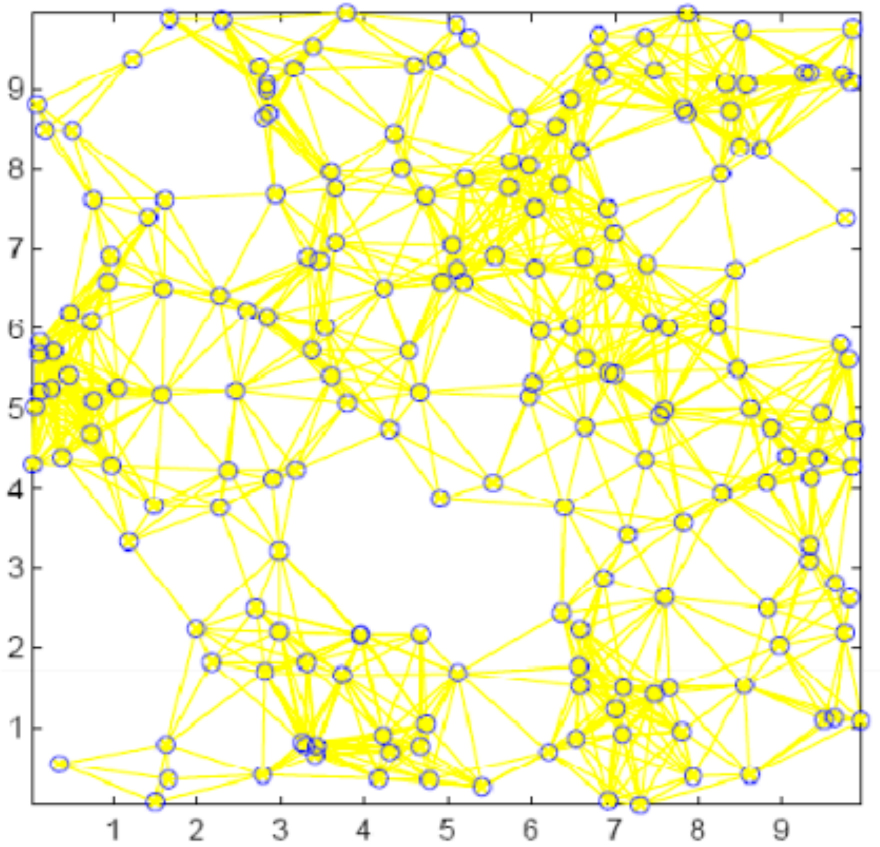
Multi-dimensional scaling

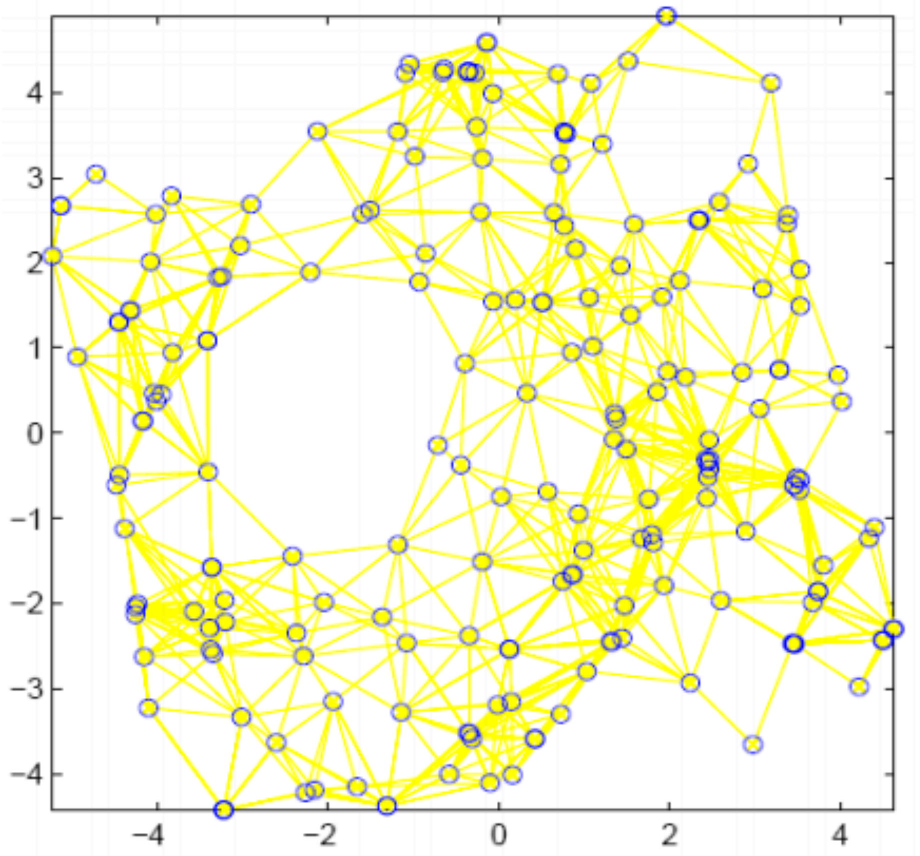
- A global optimization method
- Dimension reduction
- Input:
  - Matrix P: pairwise distances between nodes
- Output:
  - nodes in R^d such that their distances approximate distances in P
- Notes:
  - In a fixed-d network, if we give exact distances, MDS will return correct locations (upto translation, rotation)
  - Also works when distances are not Euclidean (ie. Has error) – mds recovers a "best fit"
  - Used in social sciences, other areas for visualization and similarity based clustering

MDS based localization

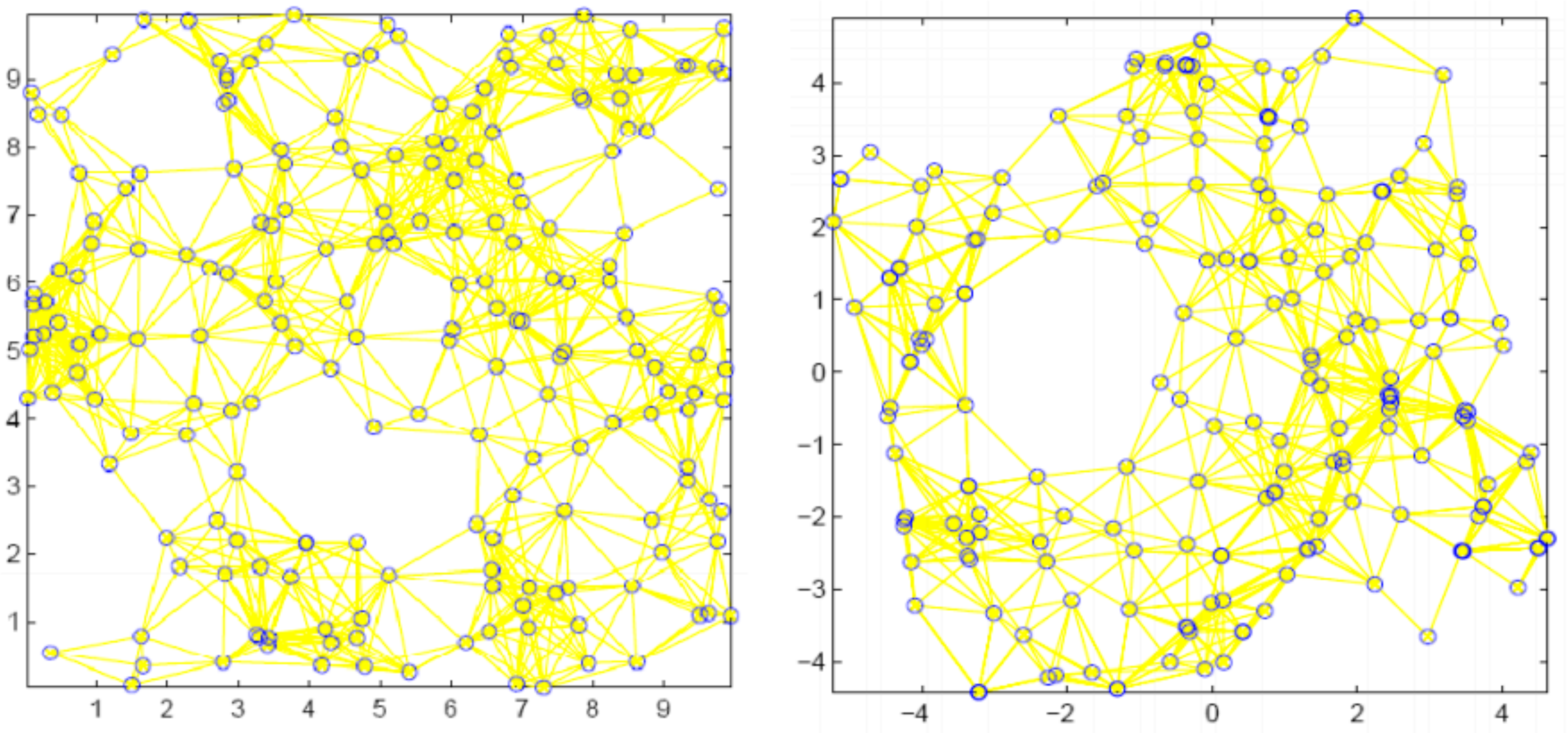
- Compute all-pairs shortest paths
- Use that to fill input matrix P
    - Approximates the actual distance
- Apply MDS to get embedding in R^2
    - (Or R^3 for 3d nets)

UDG with random placement

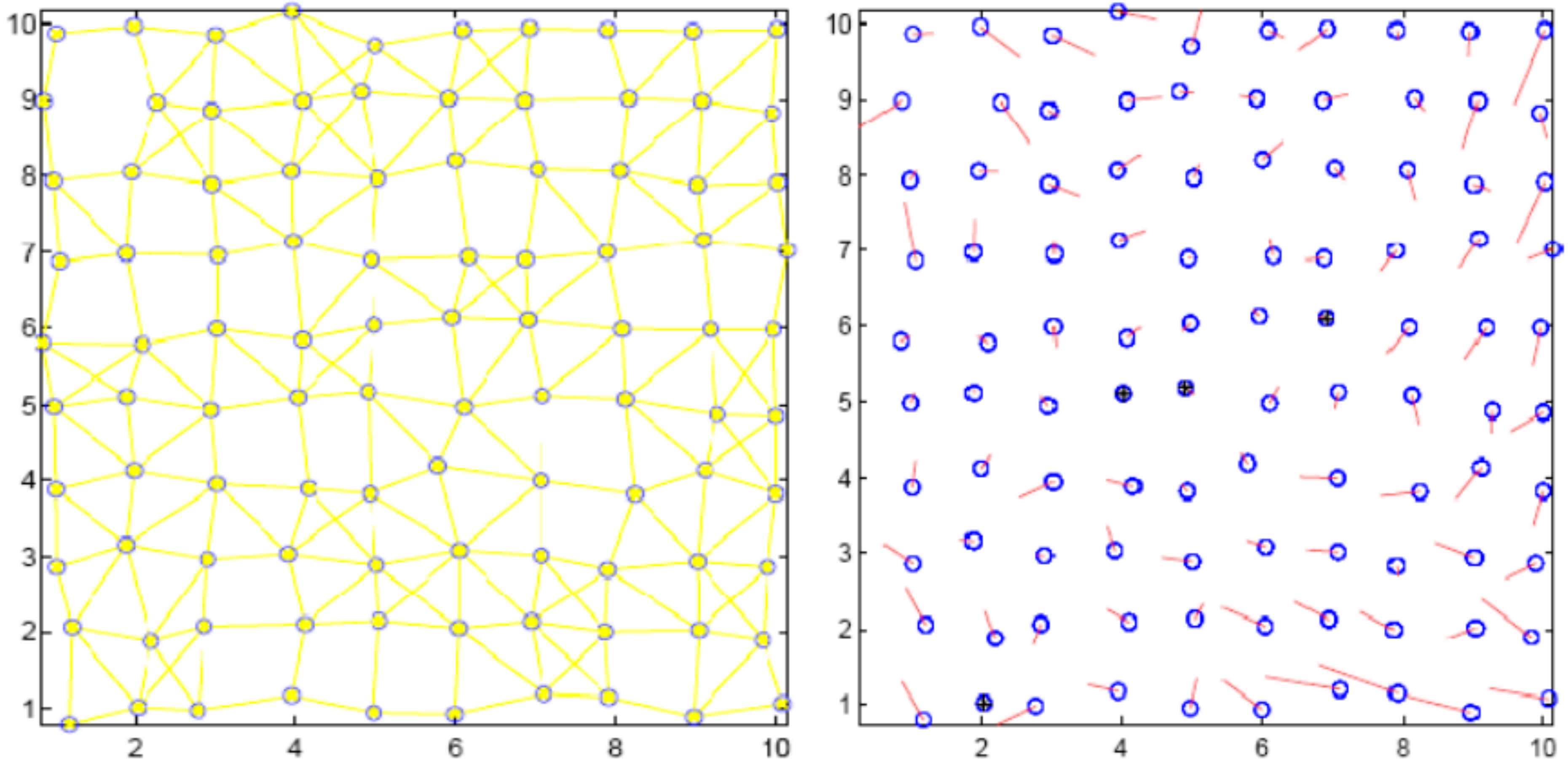




UDG with random placement



Grid placement with 10% error



MDS

- Centralized
  - One node needs the entire input matrix
- Expensive computation
  - Eigen value computations
- When shortest path length is not a good approximation, results can be bad