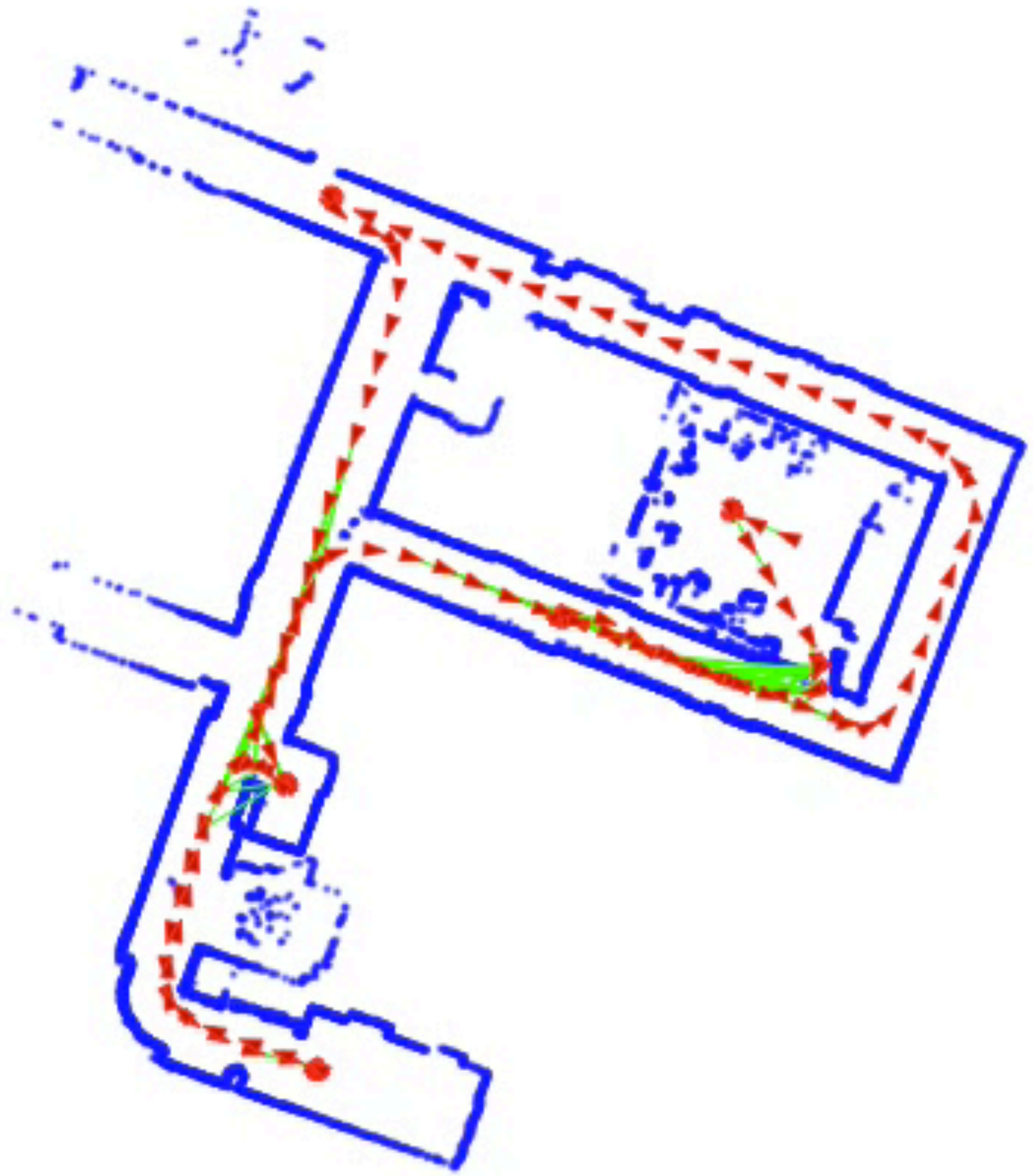


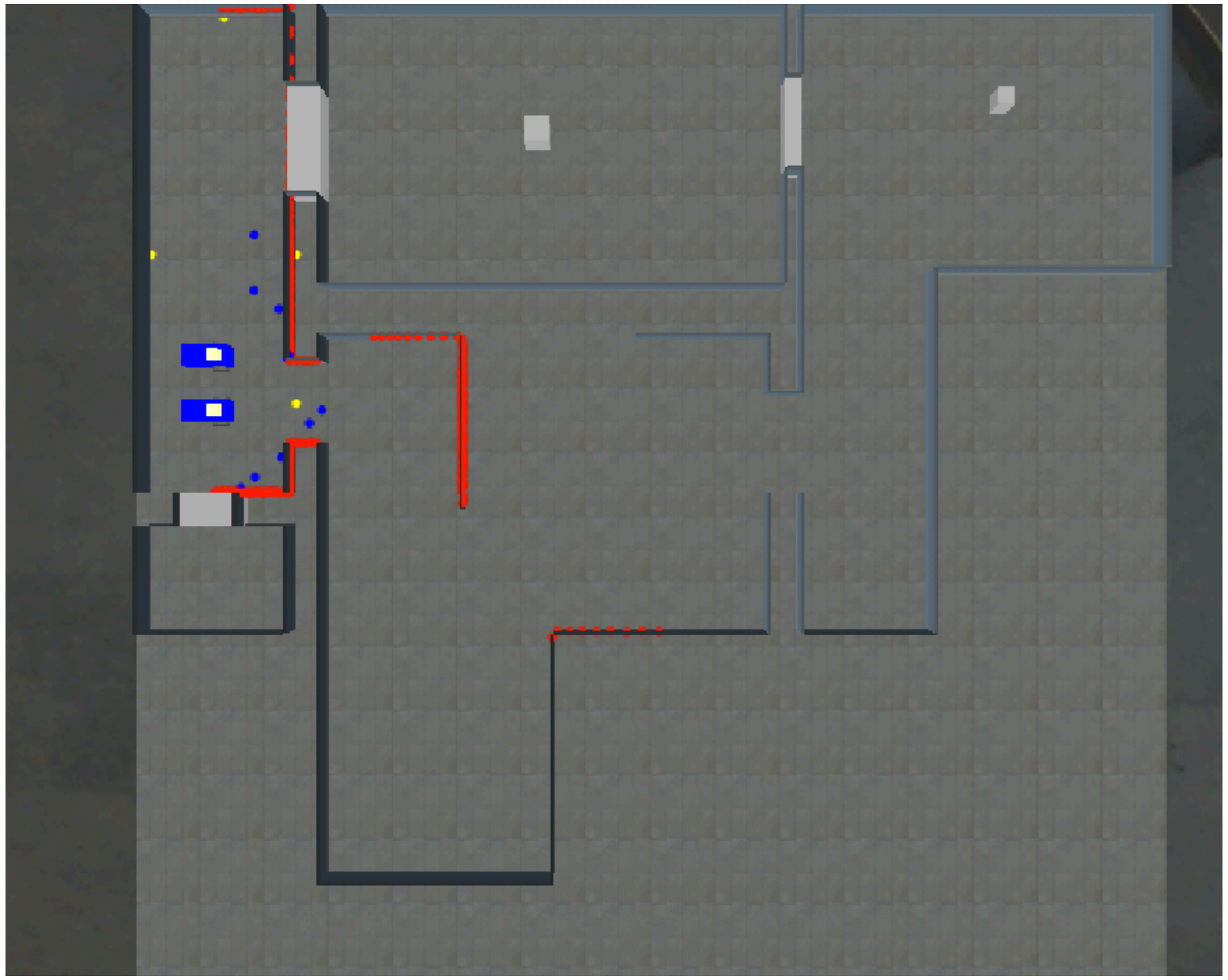
Soar Robot Unleashed

Keegan R. Kinkade

21 February 2012

Soar Robot Environment

- Soar Robot Project
  - Differentially driven robot utilizing LIDAR (**LI**ght **D**etection **A**nd **R**anging)
  - Encoders estimate robot's trajectory
- What additional machinery is required to place agent in world?



Input Link

^io.input-link

Area Description

- ^id
- ^type
- ^gateway
  - ^id
  - ^x, ^y, ^distance
  - ^direction
  - ^to (2x)
  - ^door
    - ^id
    - ^state
- ^wall
  - ^direction
  - ^open
  - ^to
  - ^x, ^y, ^distance
- ^light

Topological Mapping

About the Robot

- ^self
  - ^name
  - ^area
  - ^headlight
  - ^battery
  - ^pose
    - ^x
    - ^x-velocity
    - ^y
    - ^y-velocity
    - ^yaw
    - ^yaw-velocity

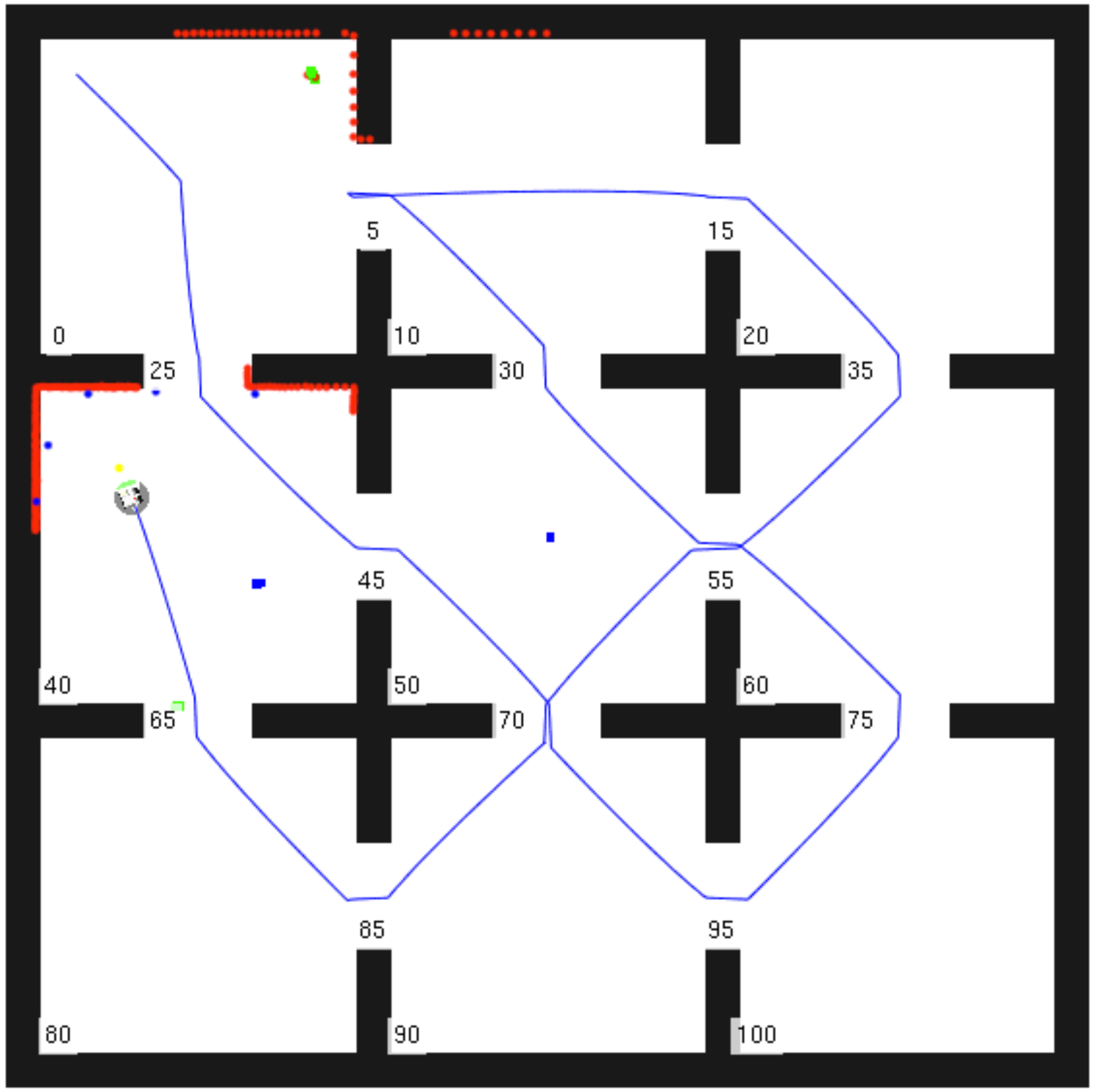
SLAM

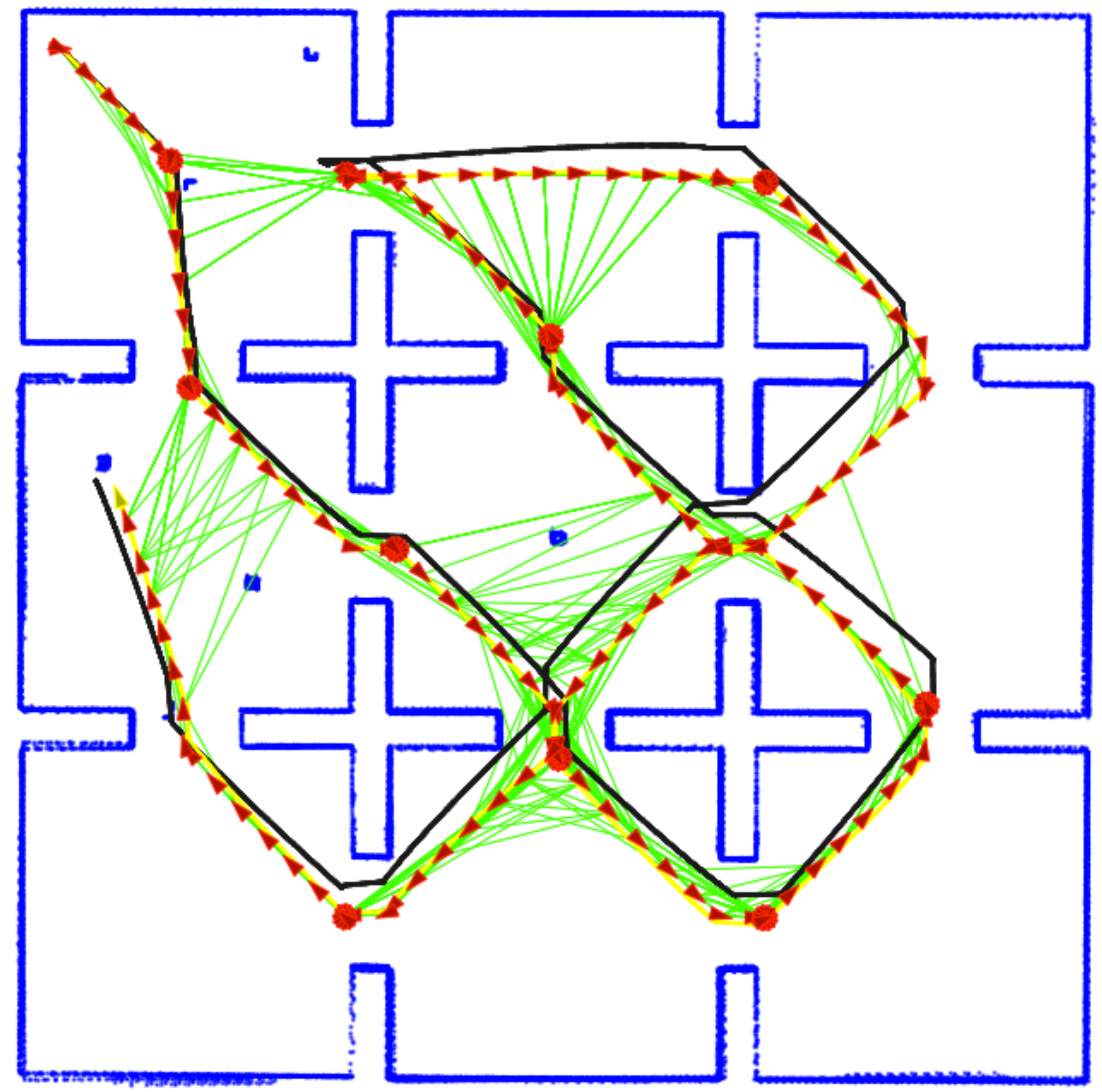
LIDAR

- ^lidar
- ^range
- ^id
- ^distance
- ^rel-bearing

Objects

- ^object
  - ^id
  - ^visible
  - ^x
  - ^y
  - ^distance
  - ^...



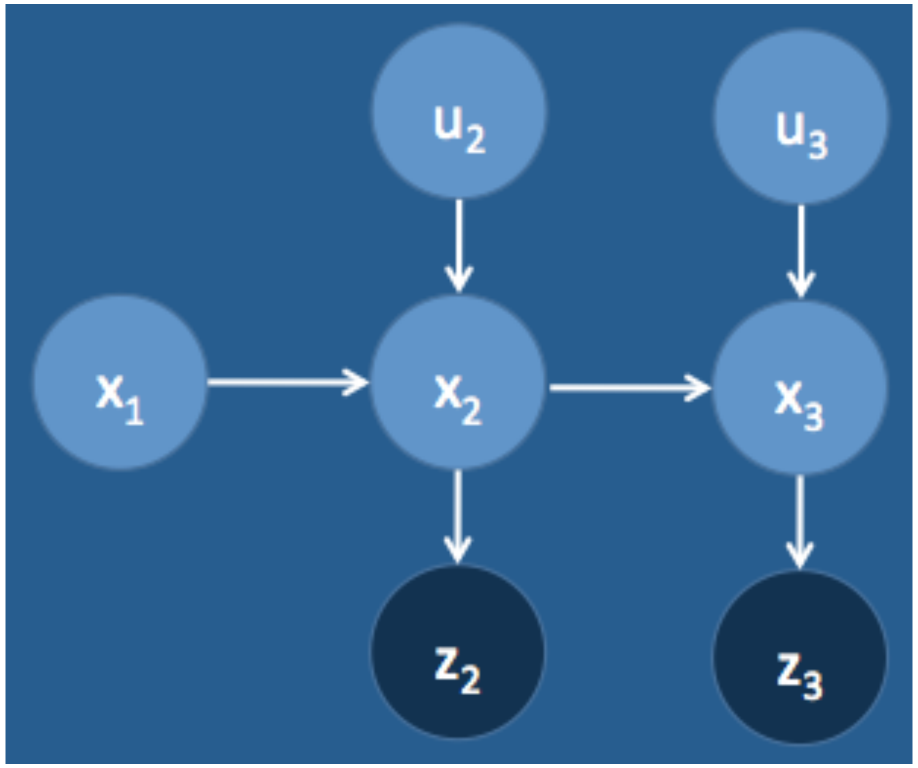


Simultaneous Localization & Mapping

Simultaneous Localization and Mapping

- While robot is exploring unknown environment:
  - **GIVEN:** Robot's movement commands and observations of unknown environment
  - **ESTIMATE:** Map of features and robot's path through environment
- Probabilistically motivated problem:
  - Errors within robot's movement and observations

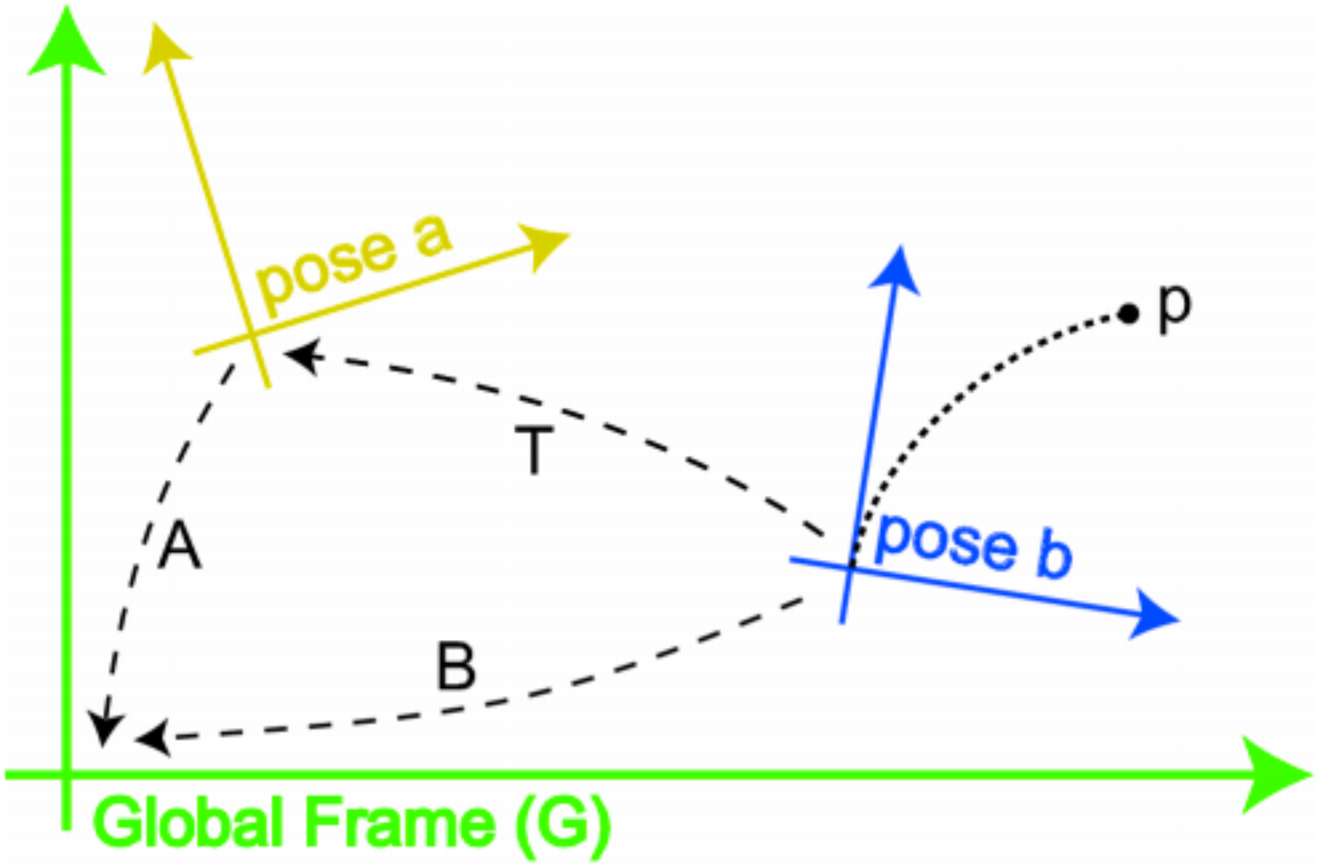
$$p(s, f \mid u, z, d)$$



Pose / Feature Graphs

- Poses and Features represented as 'Nodes'
- Connected by 'Edges' composed of Rigid Body Transformations
- Additional observations create an overdetermined system
- Feature p in Global Frame:

$$p' = Ap$$
$$= ABp$$



$$T = \begin{bmatrix} cos(\Delta\theta) & -sin(\Delta\theta) & \Delta x \\ sin(\Delta\theta) & cos(\Delta\theta) & \Delta y \\ 0 & 0 & 1 \end{bmatrix}$$

Non-Linear SLAM

- Edge → Observation: $z_i = f_i(x)$
- Observation Residual: $r_i = z_i - f_i(x)$
- Scale Residual by Observation Confidence:

$$\chi_i^2 = (z_i - f_i(x))^T \Sigma_i^{-1} (z_i - f_i(x))$$

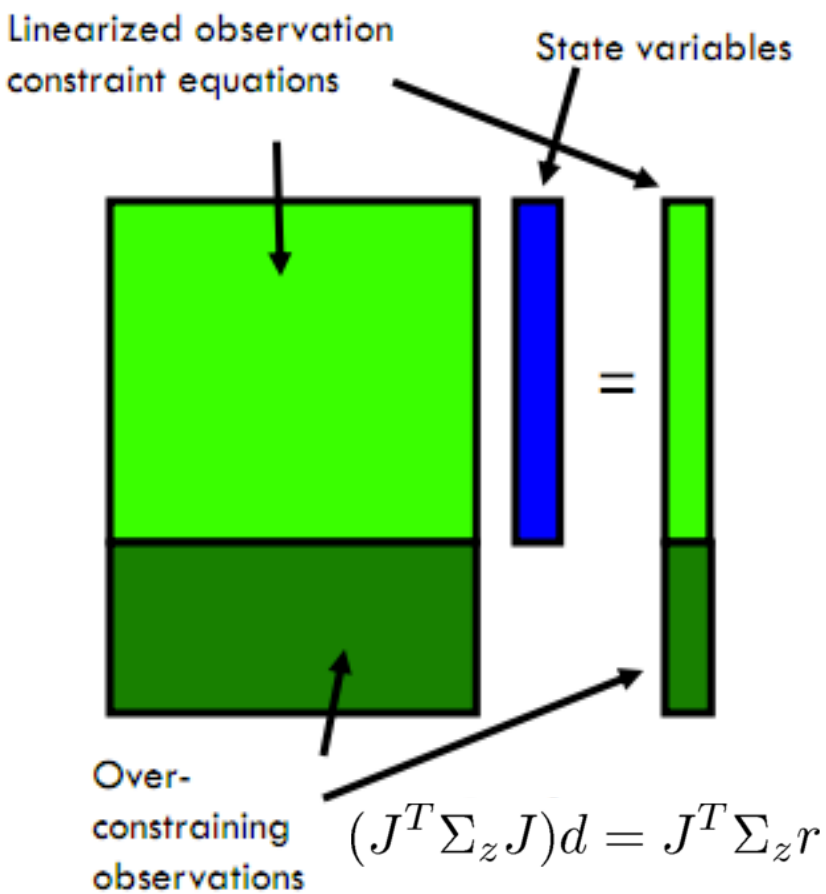
- Typical Observations != Linear
- Stack linearized observations using Jacobian:

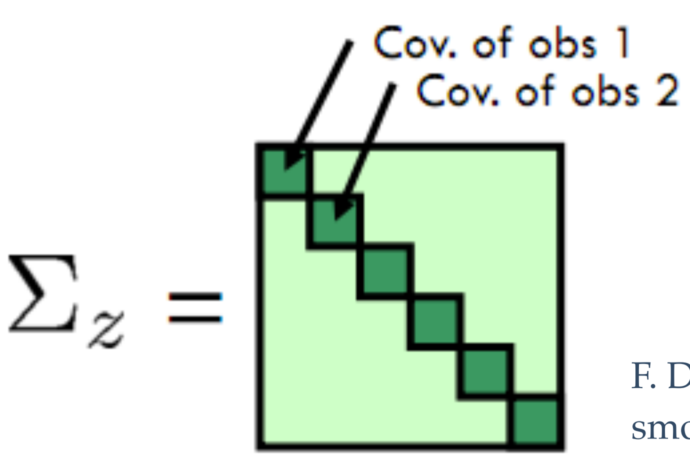
$$\chi^2 \approx (Jd - r)^T \Sigma^{-1} (Jd - r)$$, where

  - r is the observation residual
  - d is the linearization residual ($x - x_0$)

- Differentiate χ^2 with respect to d, solve for d which minimizes the χ^2 error

Square root SAM





- Naive Solution:

$$d = (J^T \Sigma_z^{-1} J)^{-1} J^T \Sigma_z^{-1} r$$

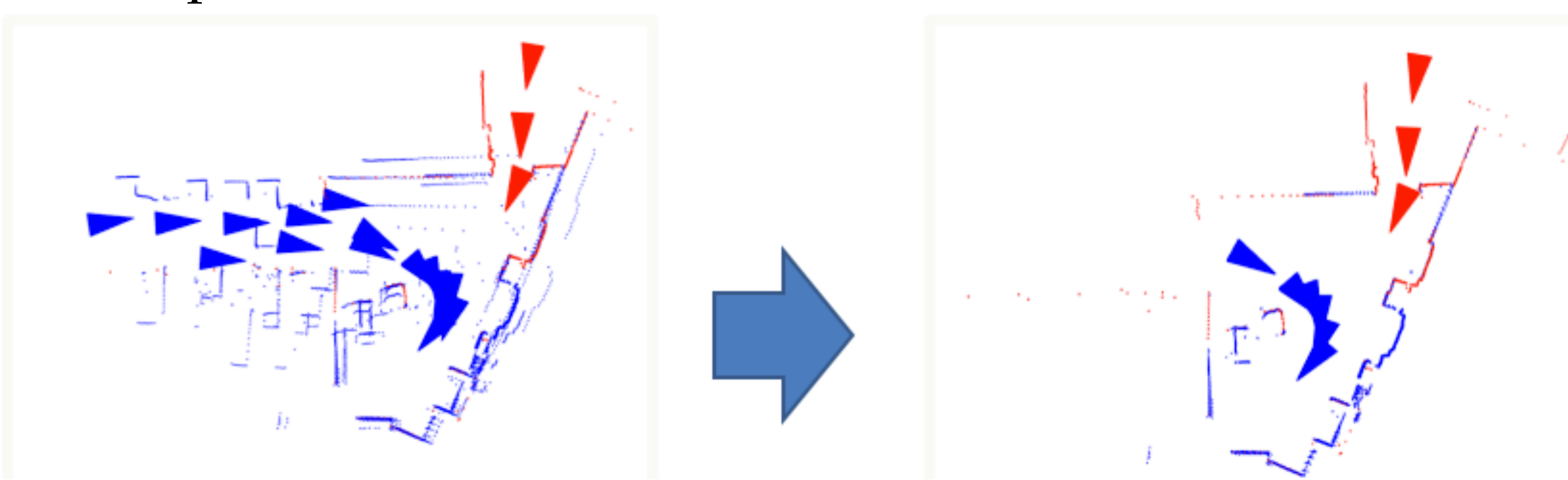
Information Matrix

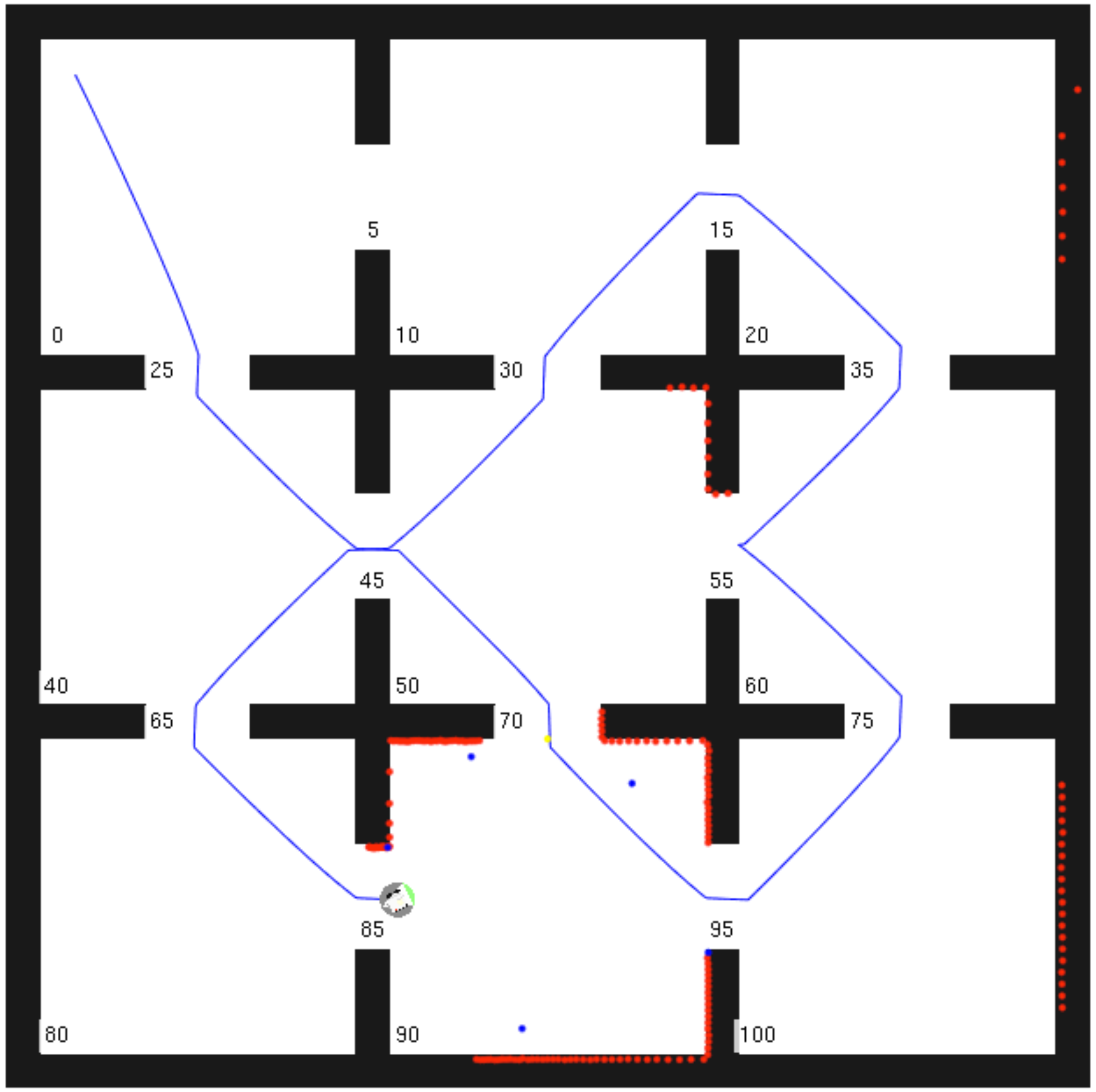
- Typically impractical due to matrix inversion complexity ~ $O(N^3)$
  - Other SLAM methods attempt to approximate this
- Solution: Exploit sparsity within the information matrix
  - Reorder nodes to induce additional sparsity
  - Back solve using Cholesky decomposition

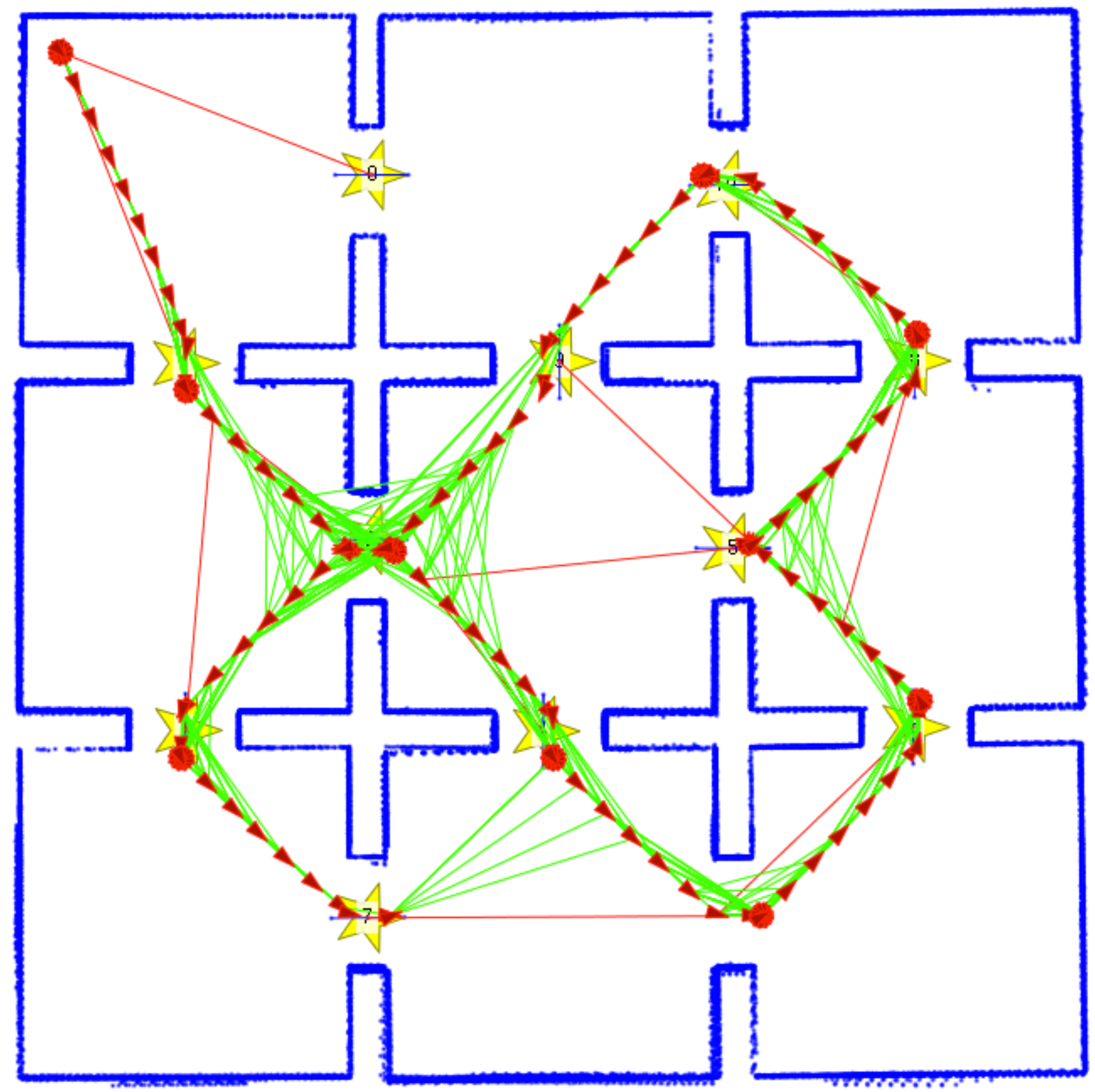
F. Dellaert and M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. International Journal of Robotics Research, 25(12):1181–1203, December 2006

Loop Closure

- Using LIDAR to close the loop:
  - Determine if robot is in the location of a previous pose
  - Attempt to create RBT between two poses using their corresponding LIDAR scans
    - Resolution Scan Matching algorithm to align points of individual scans



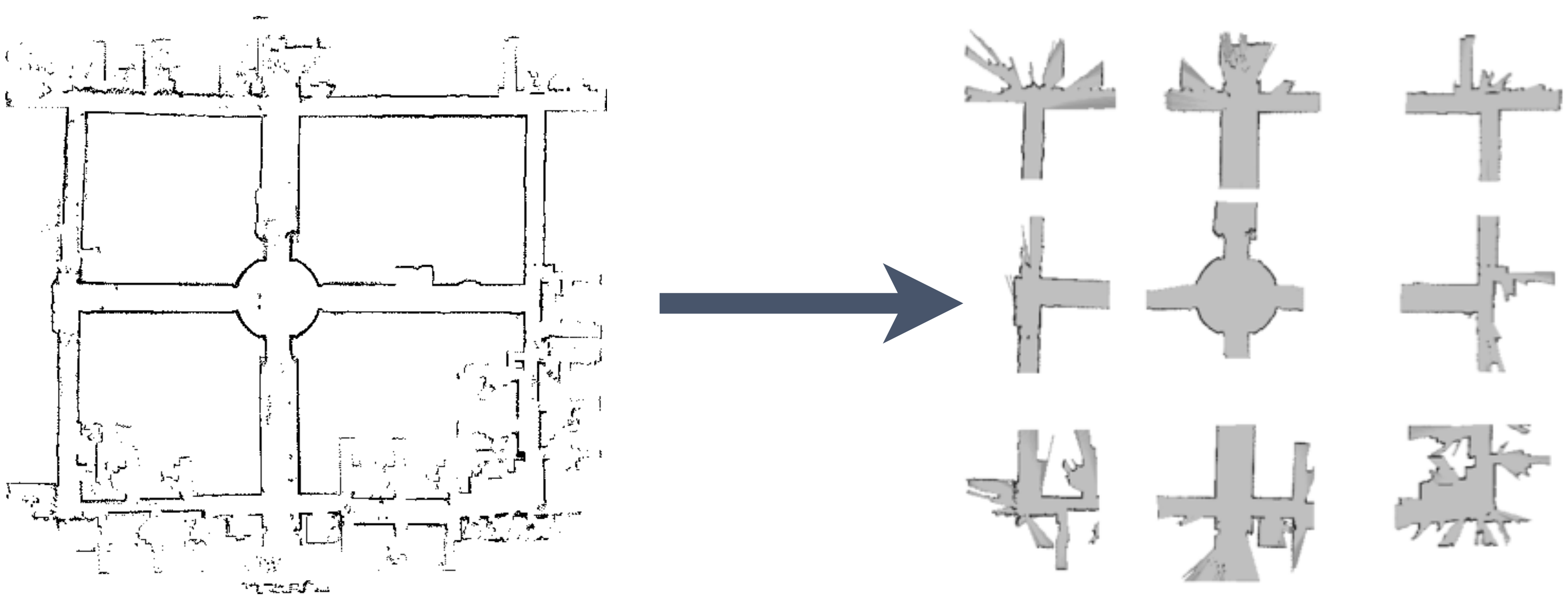




Topological Mapping

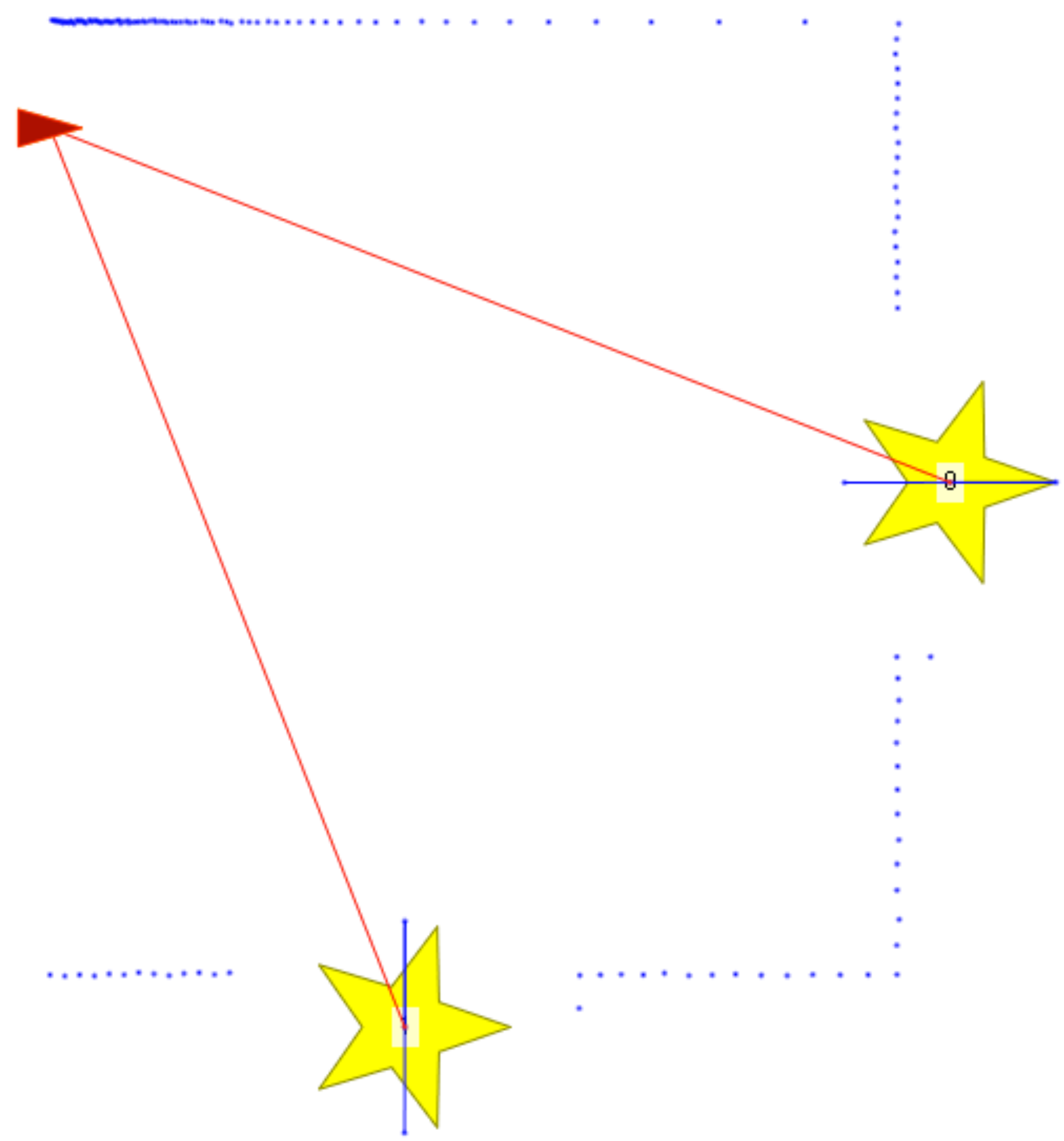
Topological Mapping

- While robot is exploring unknown environment:
  - **GIVEN:** SLAM output: Current estimate of robot poses and corresponding LIDAR scans.
  - **ESTIMATE:** Map of rooms and how those rooms are connected.

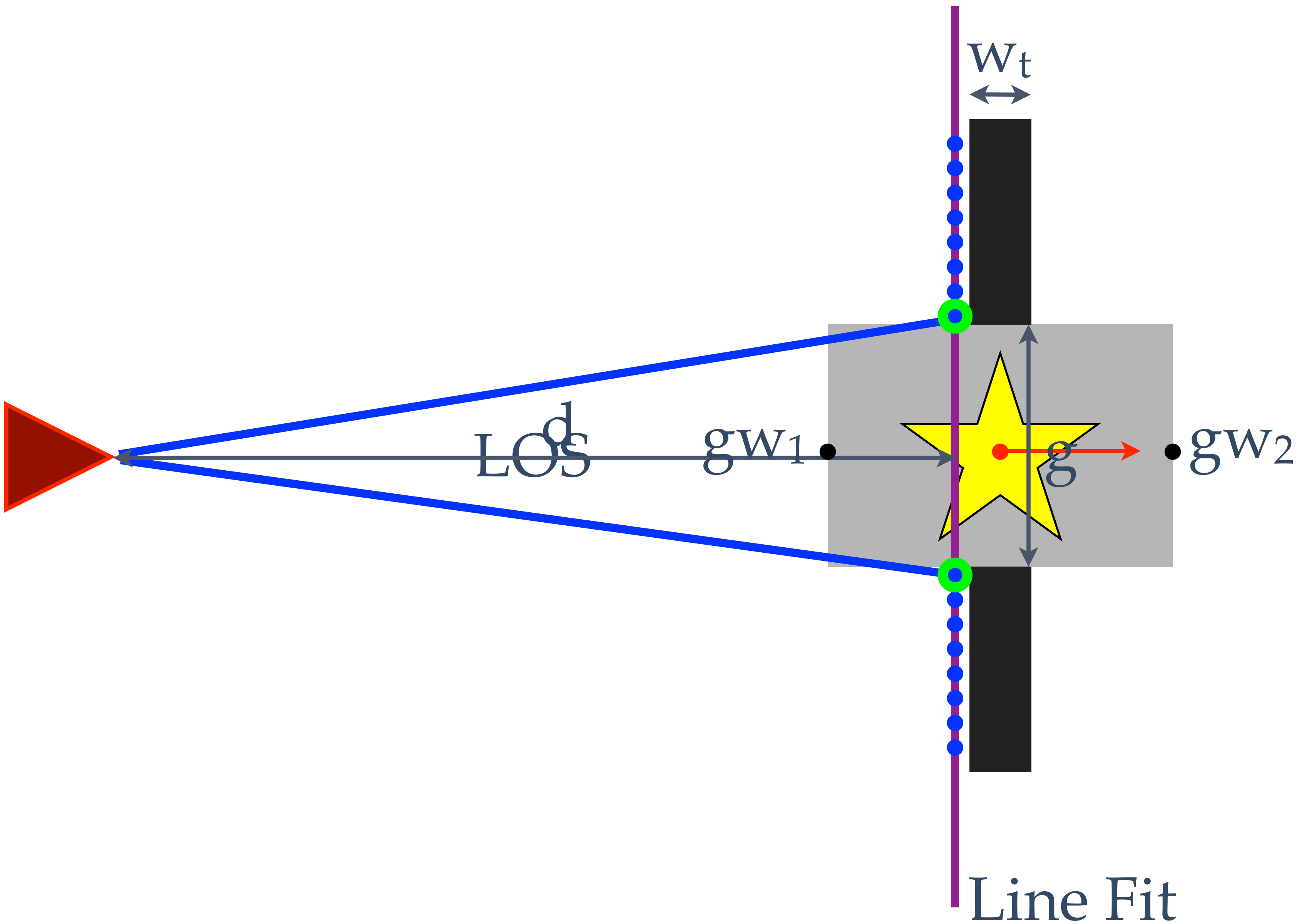


Topological Features: Doors

- For each new pose added:
  - Find ‘Drop-Off’ Points
  - Test ‘Drop-Off’ Points
    - Gap
    - Distance
    - Line Fit
    - Line of Sight
  - Set Door Direction ⊥ Line Fit
  - Set Location WRT Wall Thickness



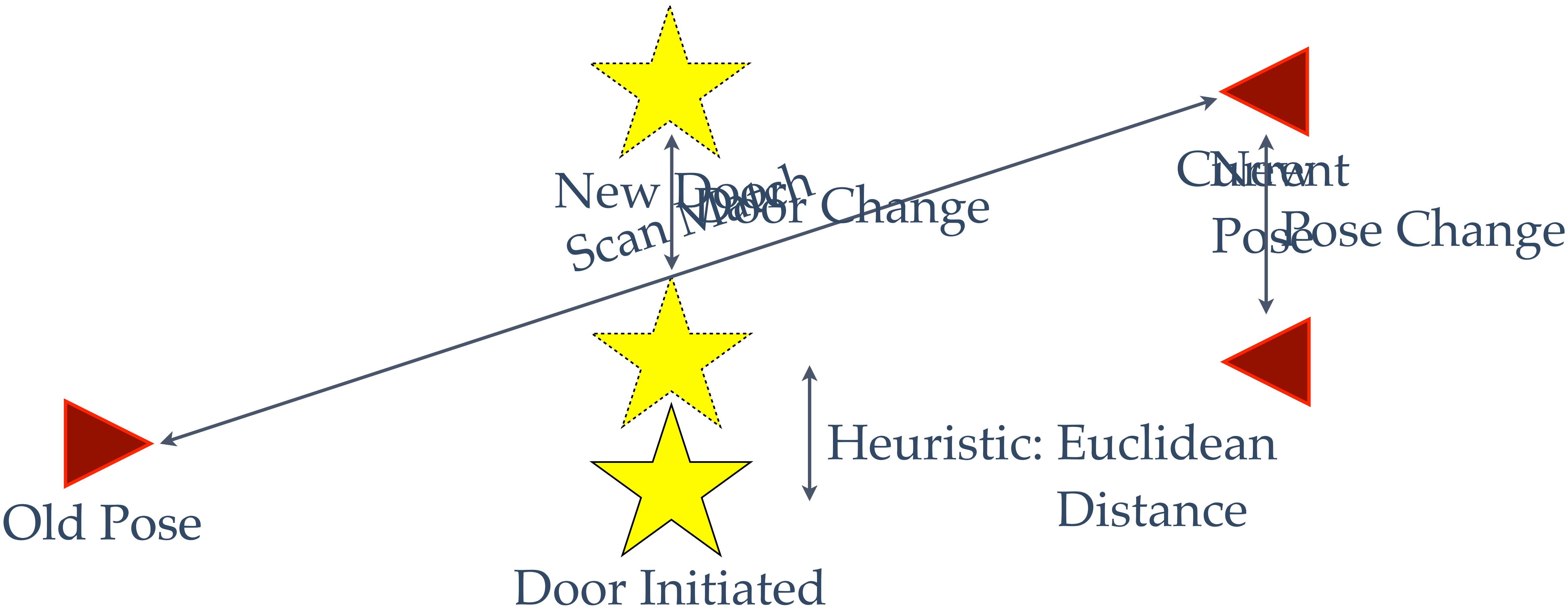
Door Finder Example



Data Association

- How do we determine if robot finds old / new door?
- Double Gated Nearest-Neighbor Search
- Greedy: O(mn)
- Issues?

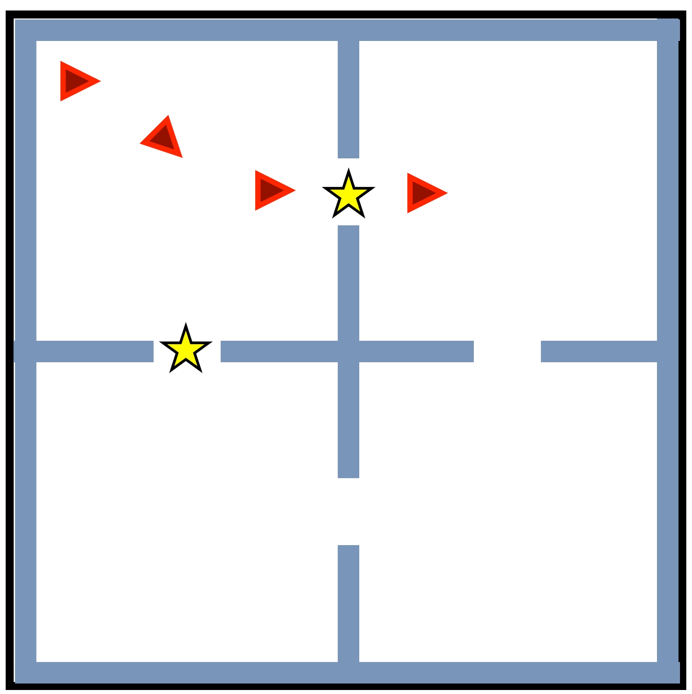
Data Association Heuristic



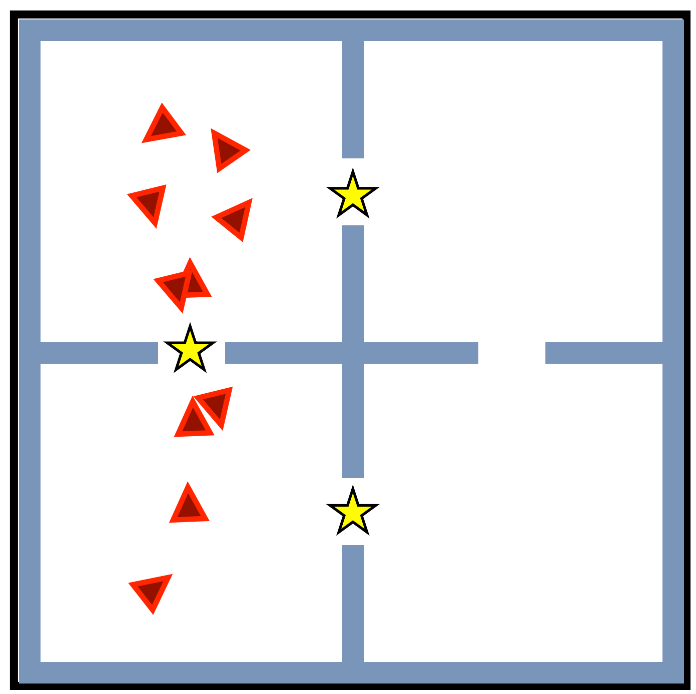
Room Connections

- Which doors connect what rooms?
    - 3-Unique Possibilities

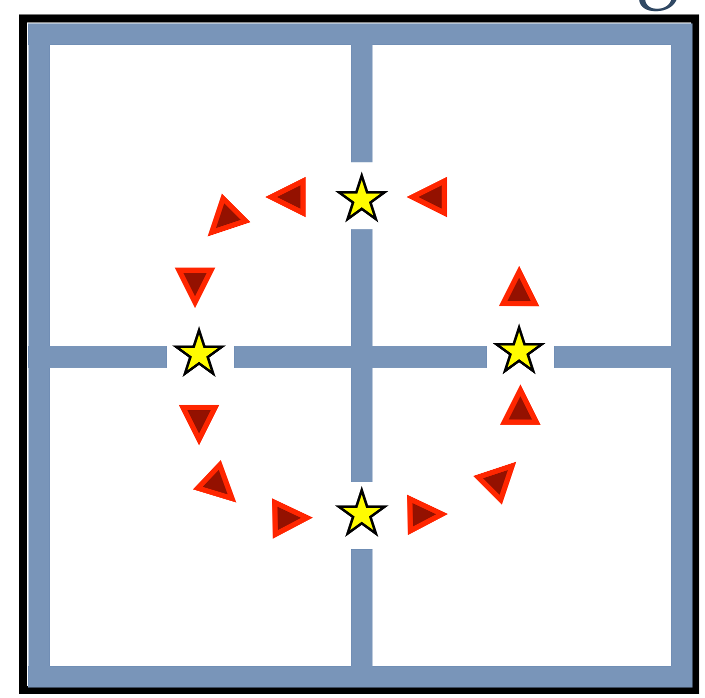
1.) Enter New Room

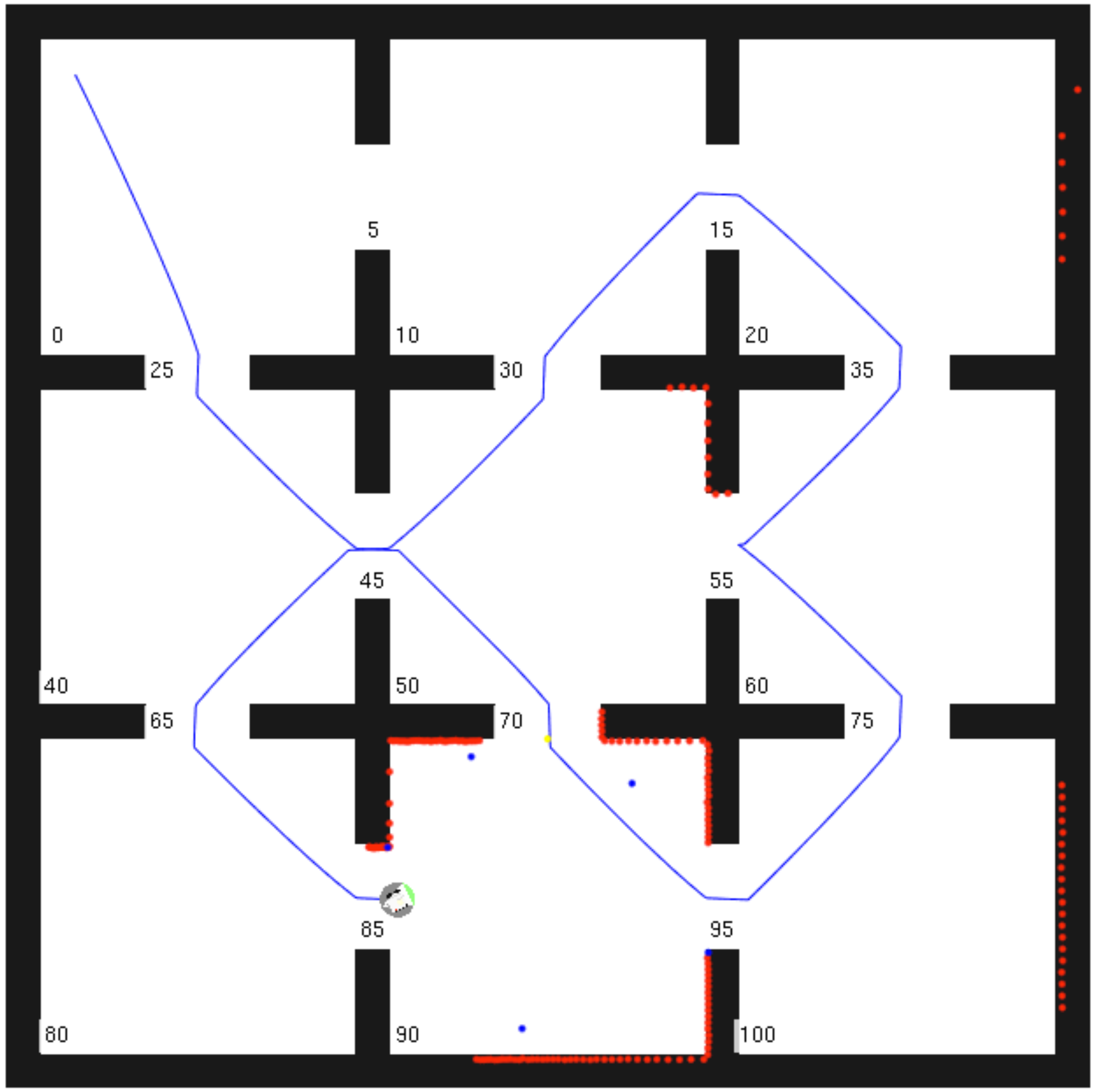


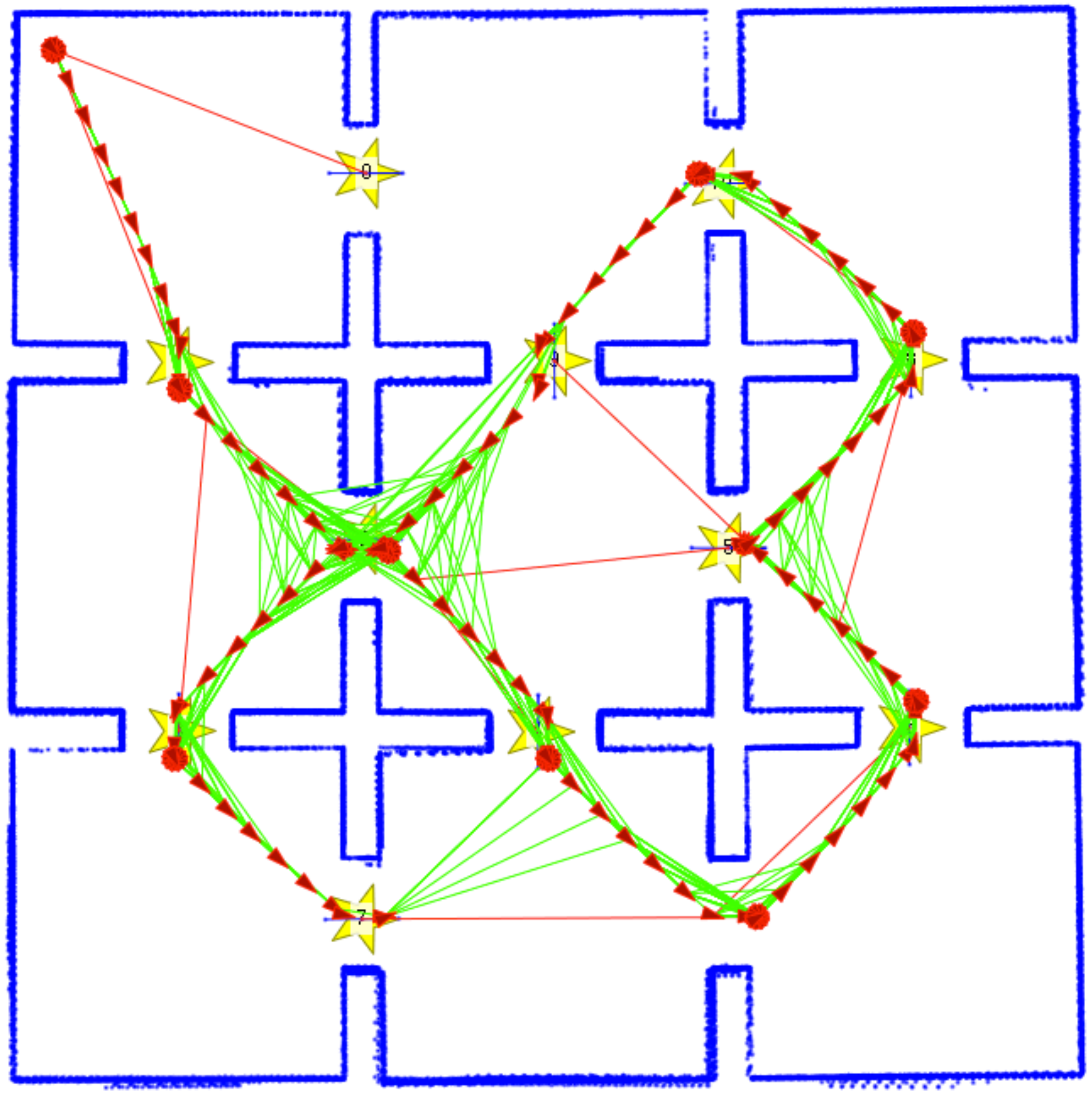
2.) Enter Old Room



3.) Enter Old Room Without Knowledge







Full System Demo

Input Link

^io.input-link

Area Description

- ^id
- ^type
- ^gateway
  - ^id
  - ^x, ^y, ^distance
  - ^direction
  - ^to (2x)
  - ^door
    - ^id
    - ^state
- ^x
- ^y
- ^light

About the Robot

- ^self
  - ^name
  - ^area
  - ^headlight
  - ^battery
  - ^pose
    - ^x
    - ^x-velocity
    - ^y
    - ^y-velocity
    - ^yaw
    - ^yaw-velocity

LIDAR

- ^lidar
- ^range
- ^id
- ^distance
- ^rel-bearing

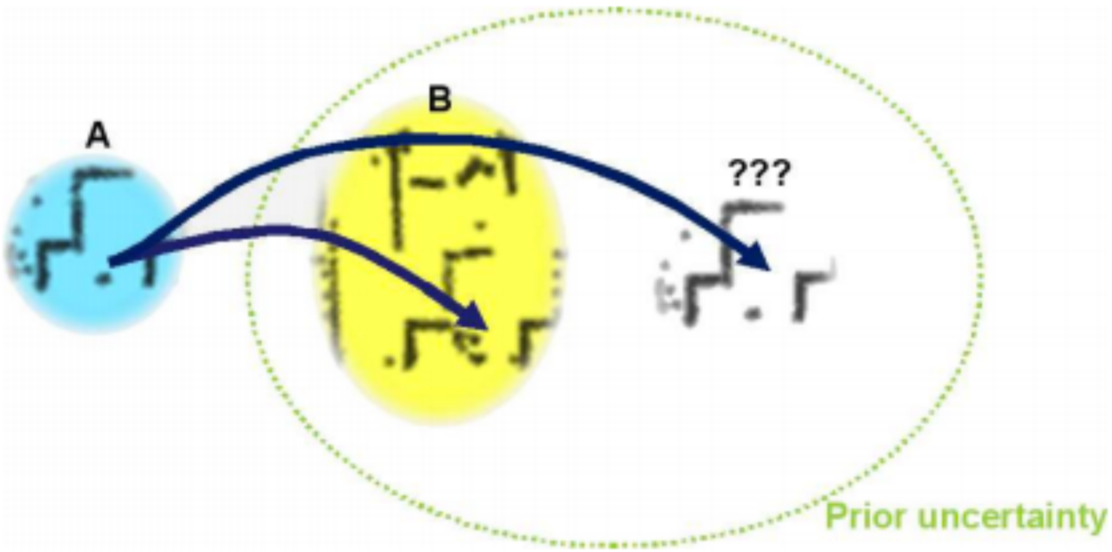
Objects

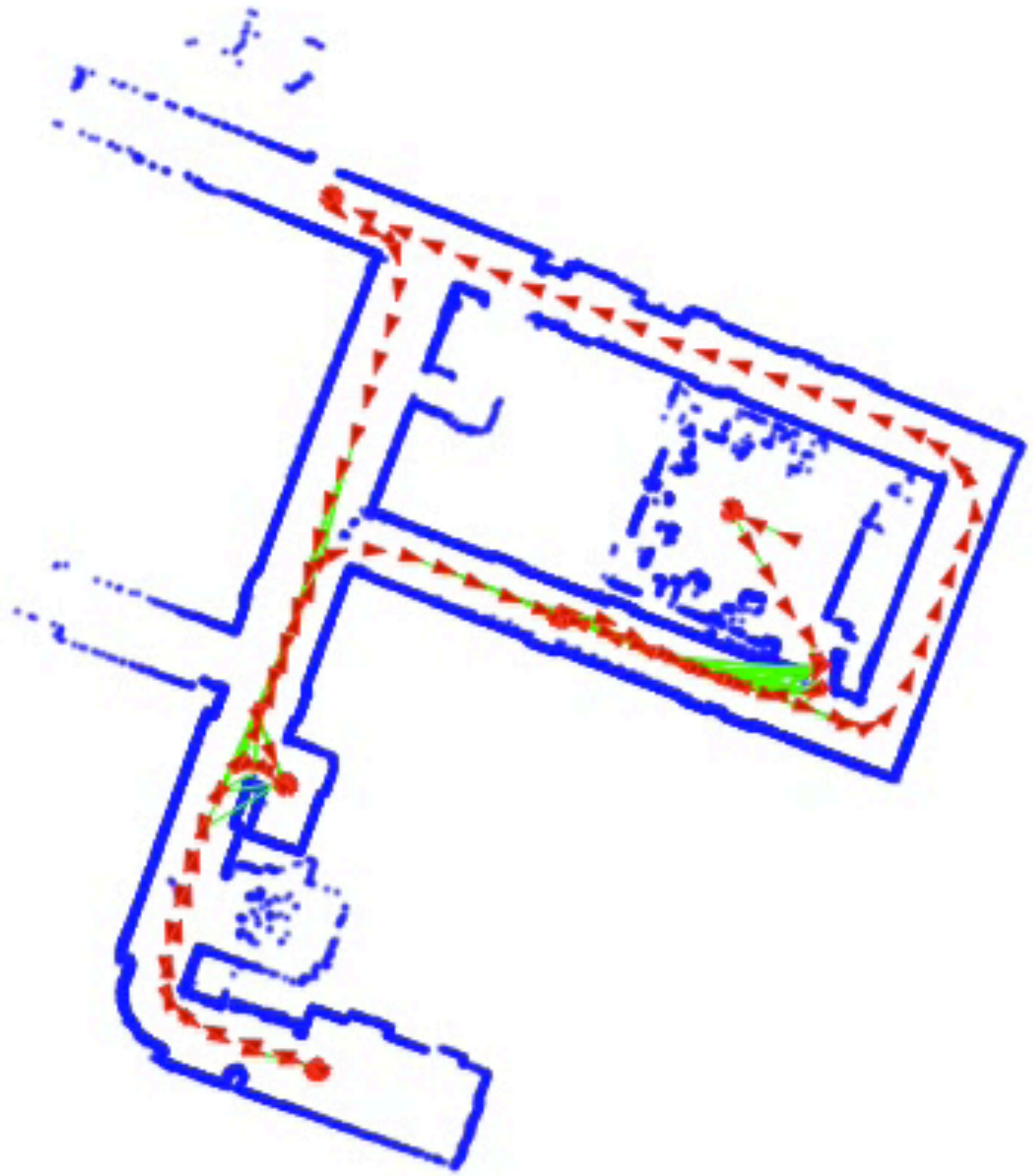
- ^object
  - ^id
  - ^visible
  - ^x
  - ^y
  - ^distance
  - ^...

Simulated

Future Work

- Additional Feature Extraction
- Cooperation:
  - Loop Closing
  - Topological Mapping





Soar Robot Unleashed

Keegan R. Kinkade

21 February 2012