CSCE 774 ROBOTICS SYSTEMS

Exploration

Ioannis Rekleitis

Three Main Challenges in Robotics

1. Where am I? (Localization)
   - Sense
   - relate sensor readings to a world model
   - compute location relative to model
   - assumes a perfect world model
2. What the world looks like? (Mapping)
   - sense from various positions
   - integrate measurements to produce map
   - assumes perfect knowledge of position

- Together 1 and 2 form the problem of *Simultaneous Localization and Mapping* (SLAM)

3. How do I go from **A** to **B**? (Path Planning)
   - More general: Which action should I pick next?

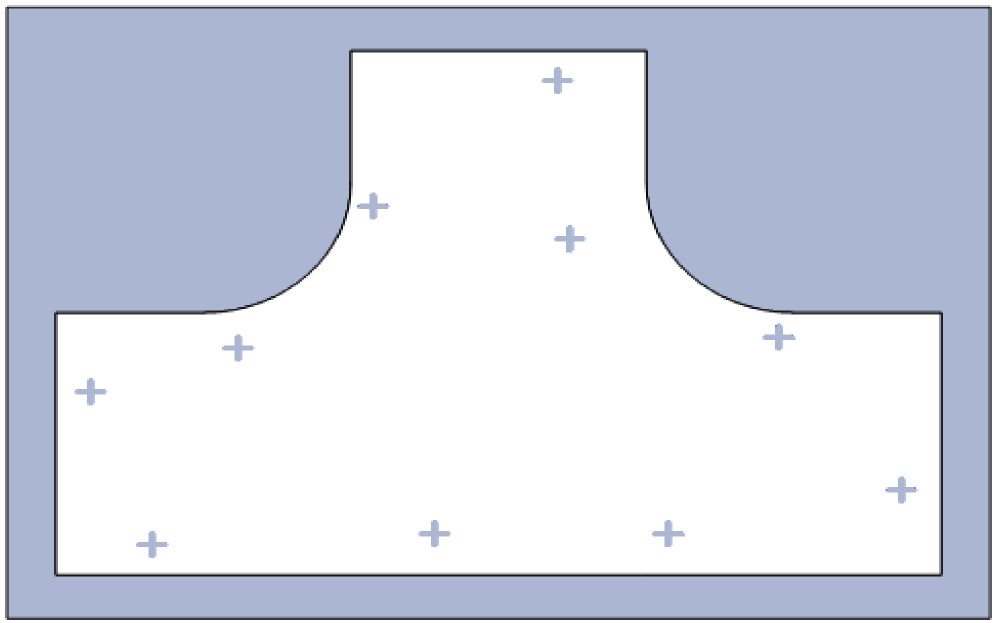
Mapping

- What the world looks like
- Improve the accuracy of the map
- Ensure that all the important parts of the environment are mapped – Exploration!

Environment Representation (Map)

- Grid Based Maps
- Feature Based Maps
- Topological Maps
- Hybrid Maps

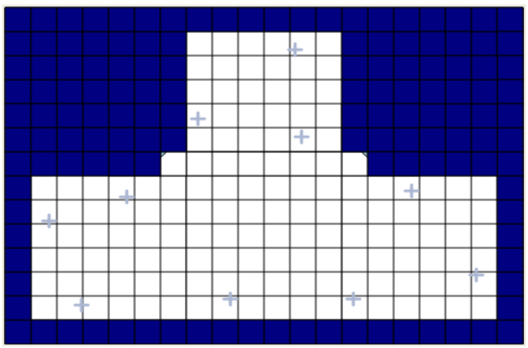
Consider this Environment:



Three Basic Map Types

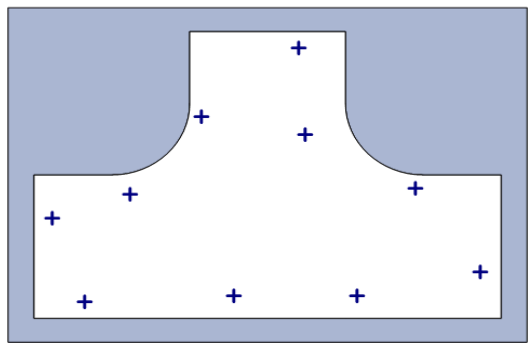
Grid-Based:

Collection of discretized obstacle/free-space pixels



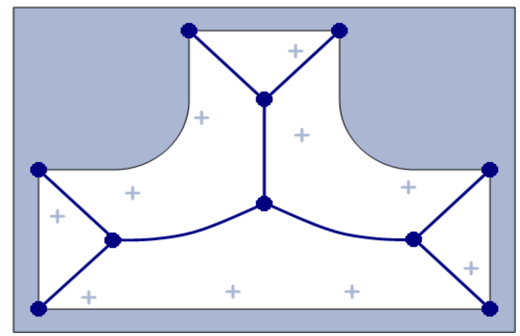
Feature-Based:

Collection of landmark locations and correlated uncertainty

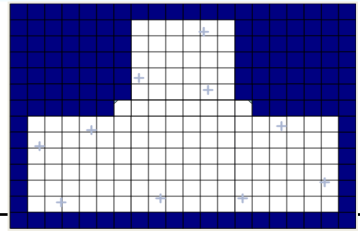
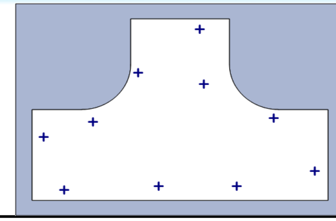
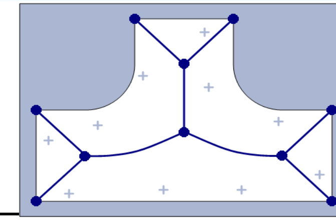


Topological:

Collection of nodes and their interconnections



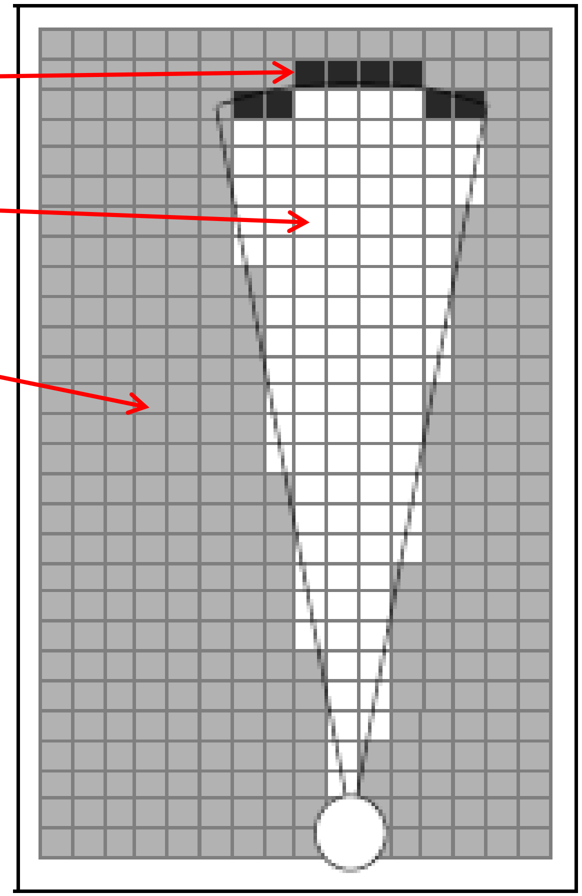
Three Basic Map Types

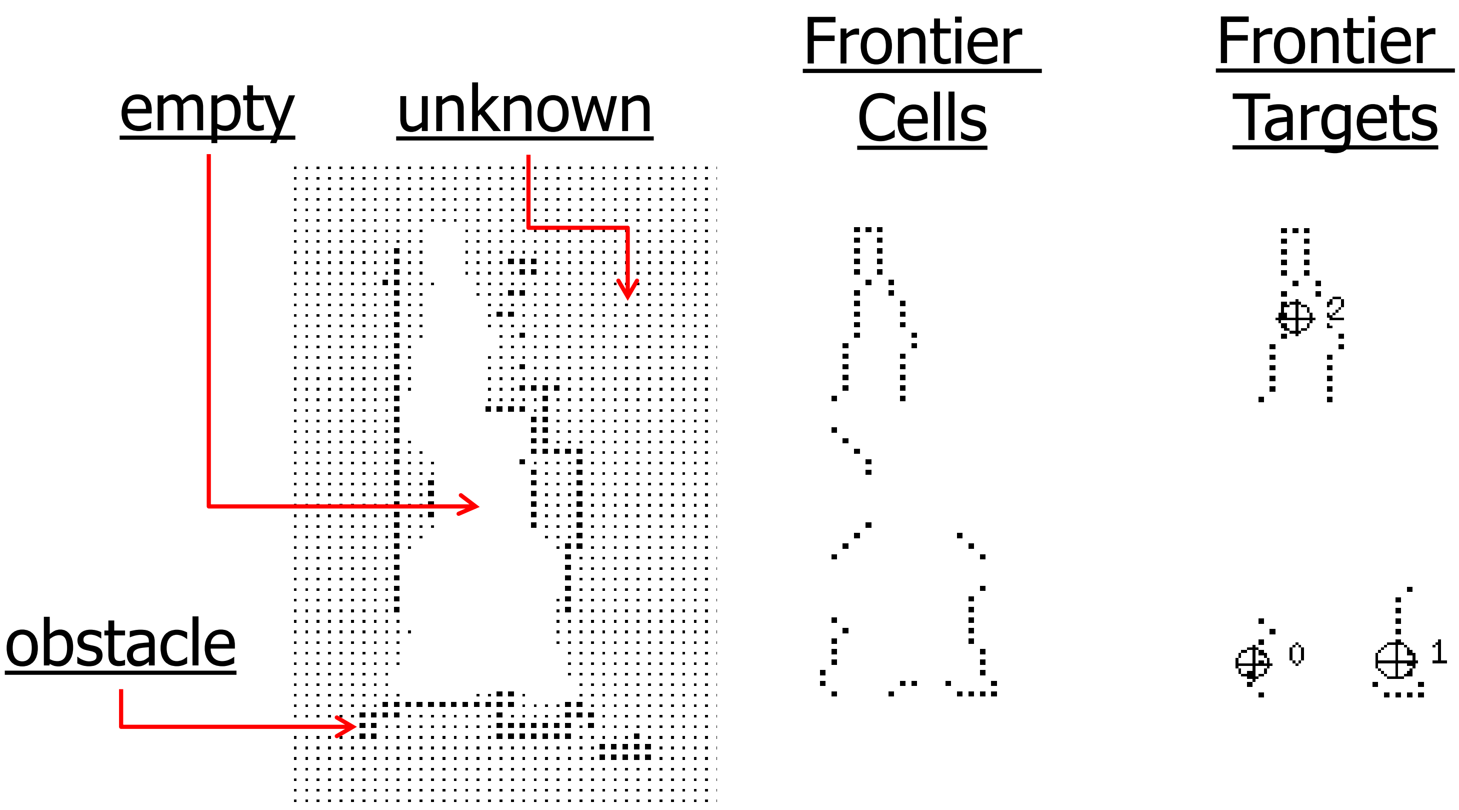
	Grid-Based	Feature-Based	Topological
Construction	Occupancy grids	Kalman Filter	Navigation control laws
Complexity	Grid size and resolution	Landmark covariance (N^3)	Minimal complexity
Obstacles	Discretized obstacles	Only structured obstacles	GVG defined by the safest path
Localization	Discrete localization	Arbitrary localization	Localize to nodes
Exploration	**Frontier-based exploration**	**No inherent exploration**	**Graph exploration**

Grid Based Maps

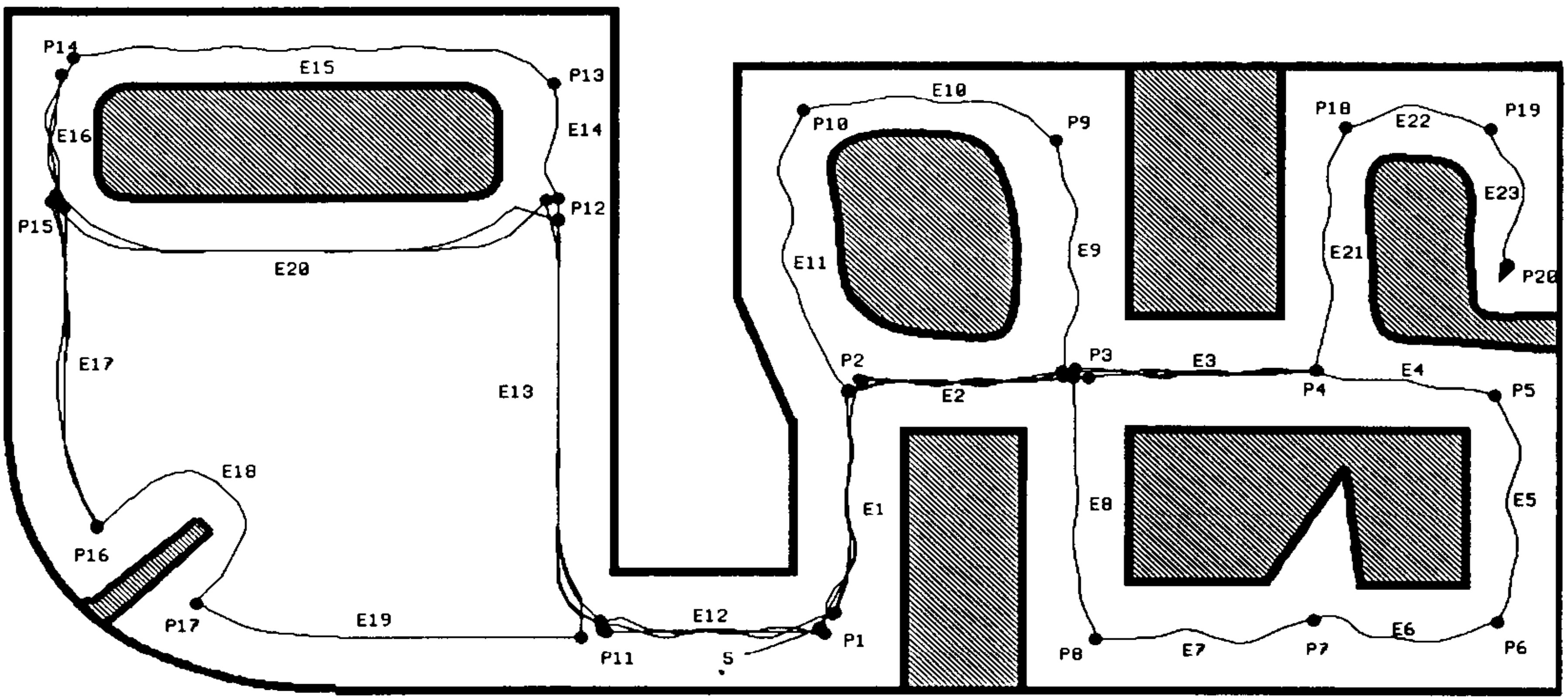
- Occupied cells
- Free cells
- Unknown cells



Frontier based Exploration (Grid Maps)

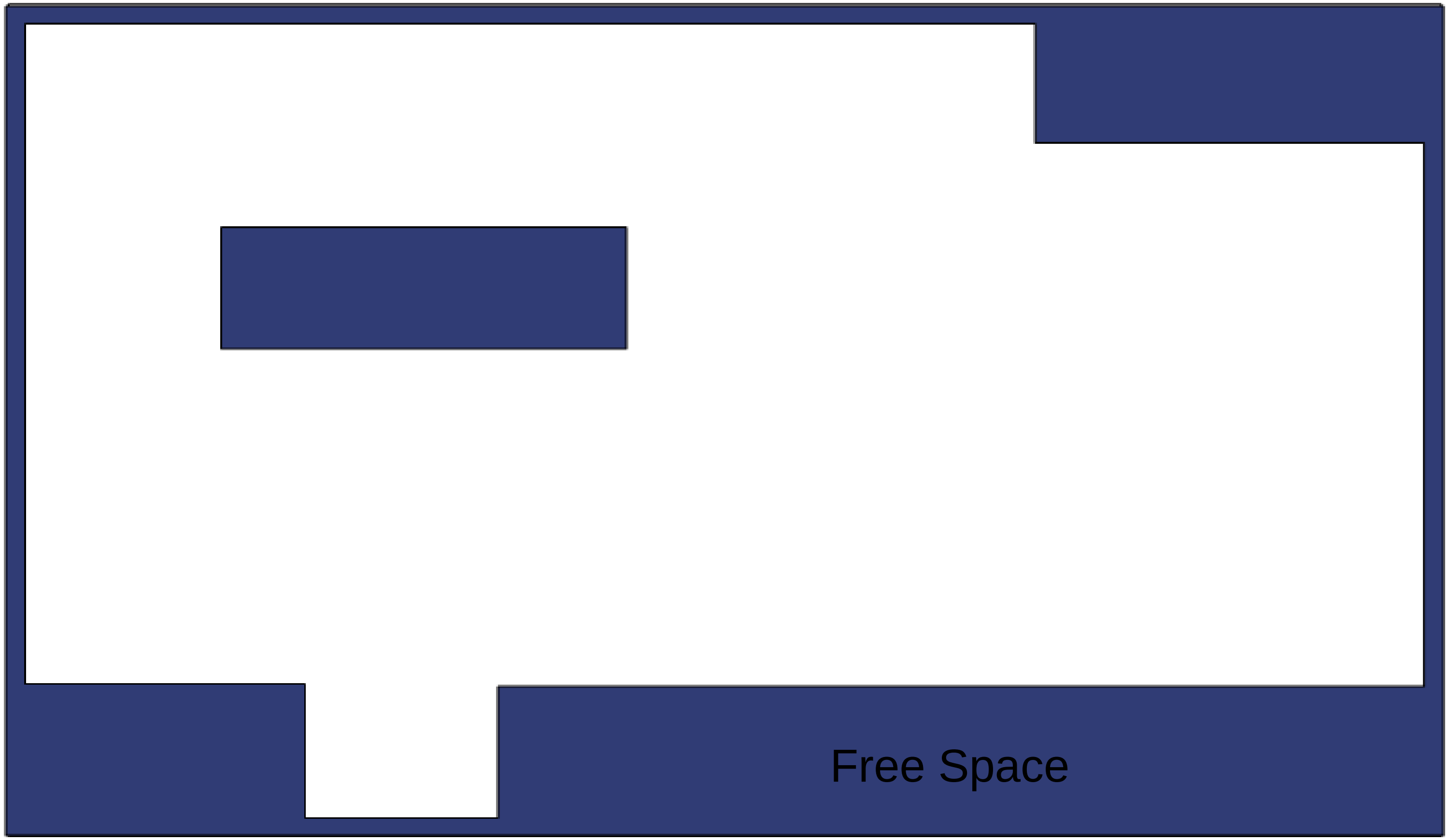


Topological Representations



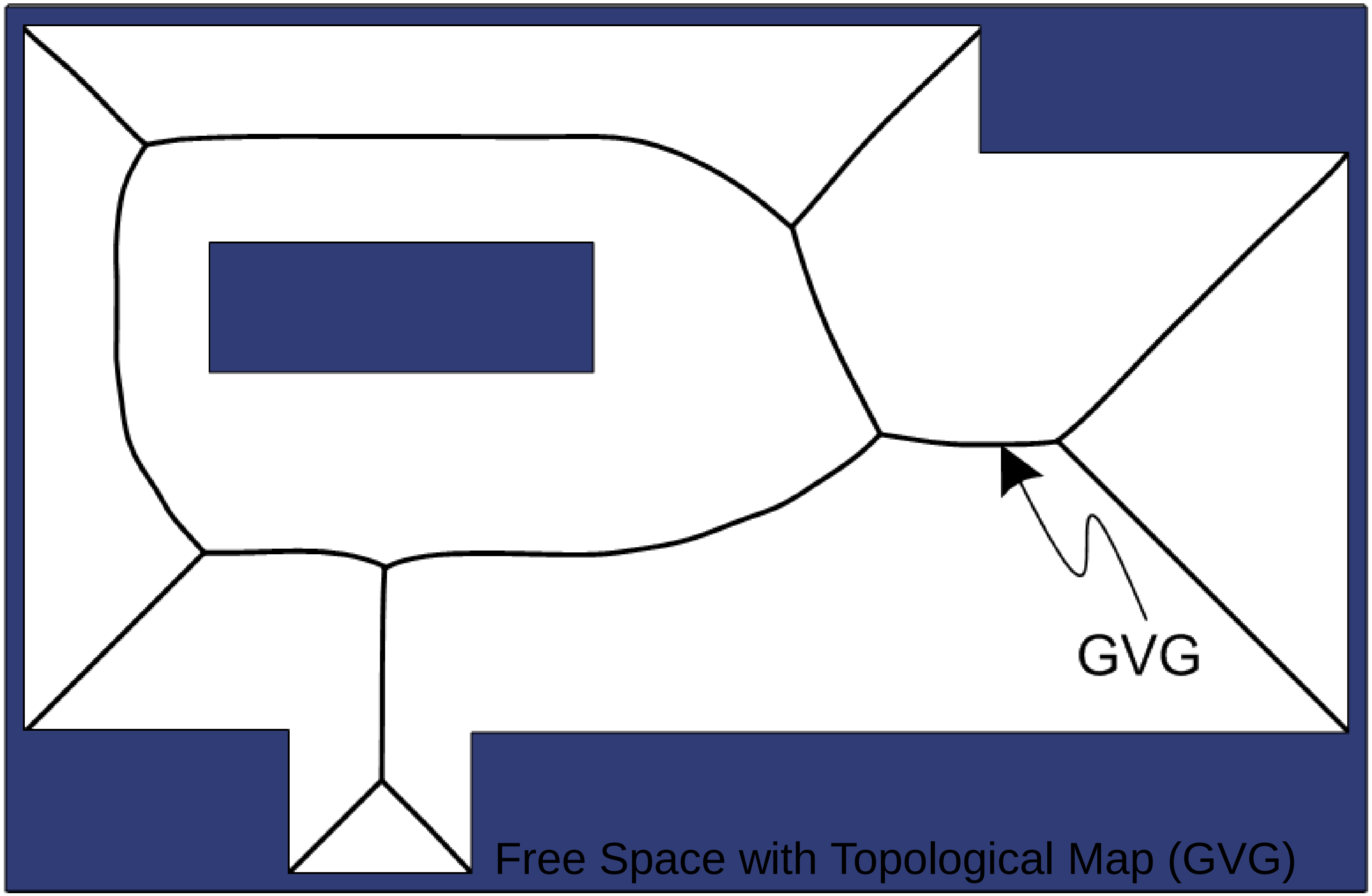
- B. J. Kuipers and Y.-T. Byun. “A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations”. In *Journal of Robotics and Autonomous Systems*, 8: 47-63, 1991.

Generalized Voronoi Graph (GVG)



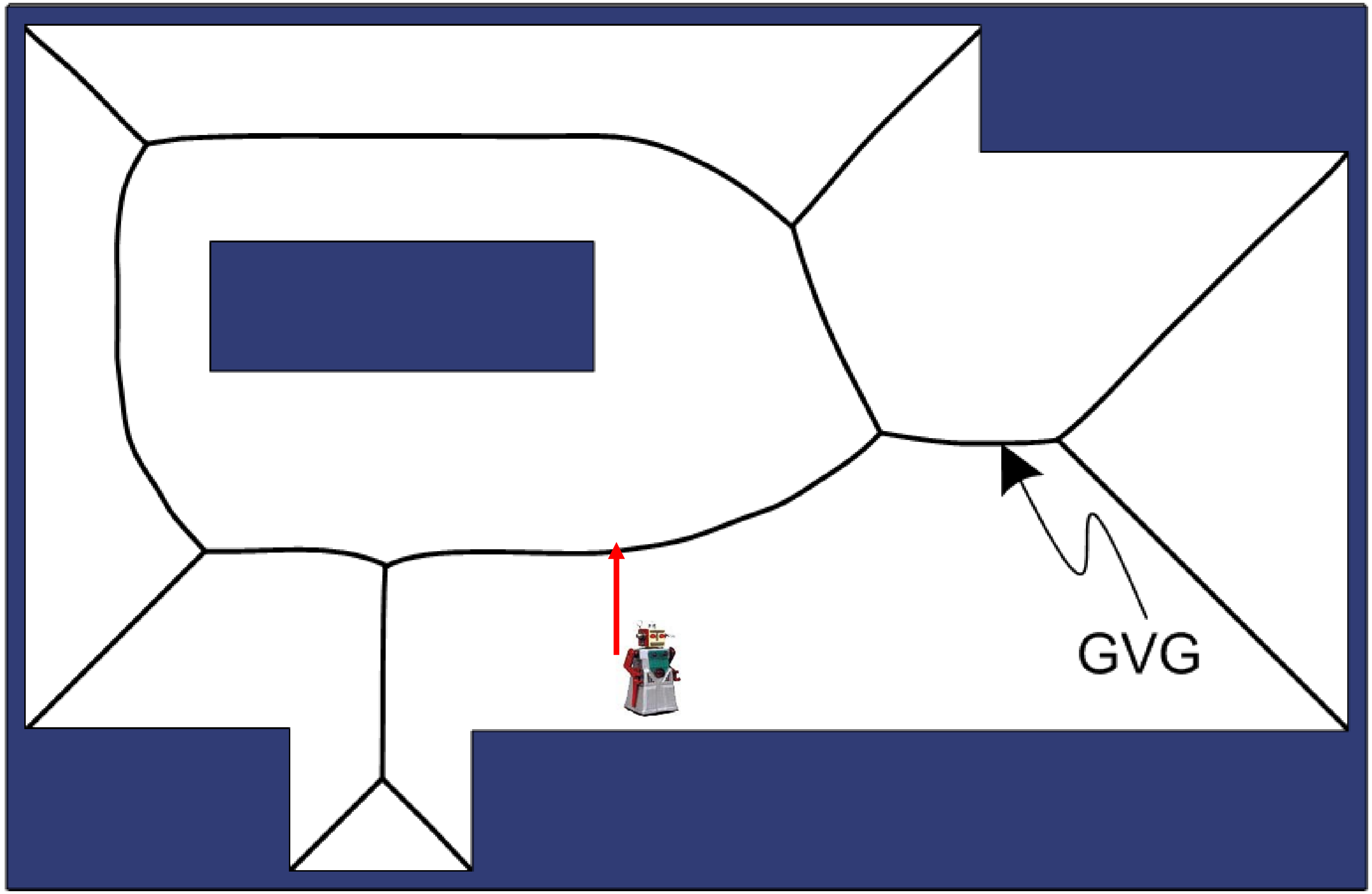
H. Choset, J. Burdick, “Sensor based planning, part ii: Incremental construction of the generalized voronoi graph”. In IEEE Conference on Robotics and Automation, pp. 1643 – 1648, 1995.

Generalized Voronoi Graph (GVG)



Generalized Voronoi Graph (GVG)

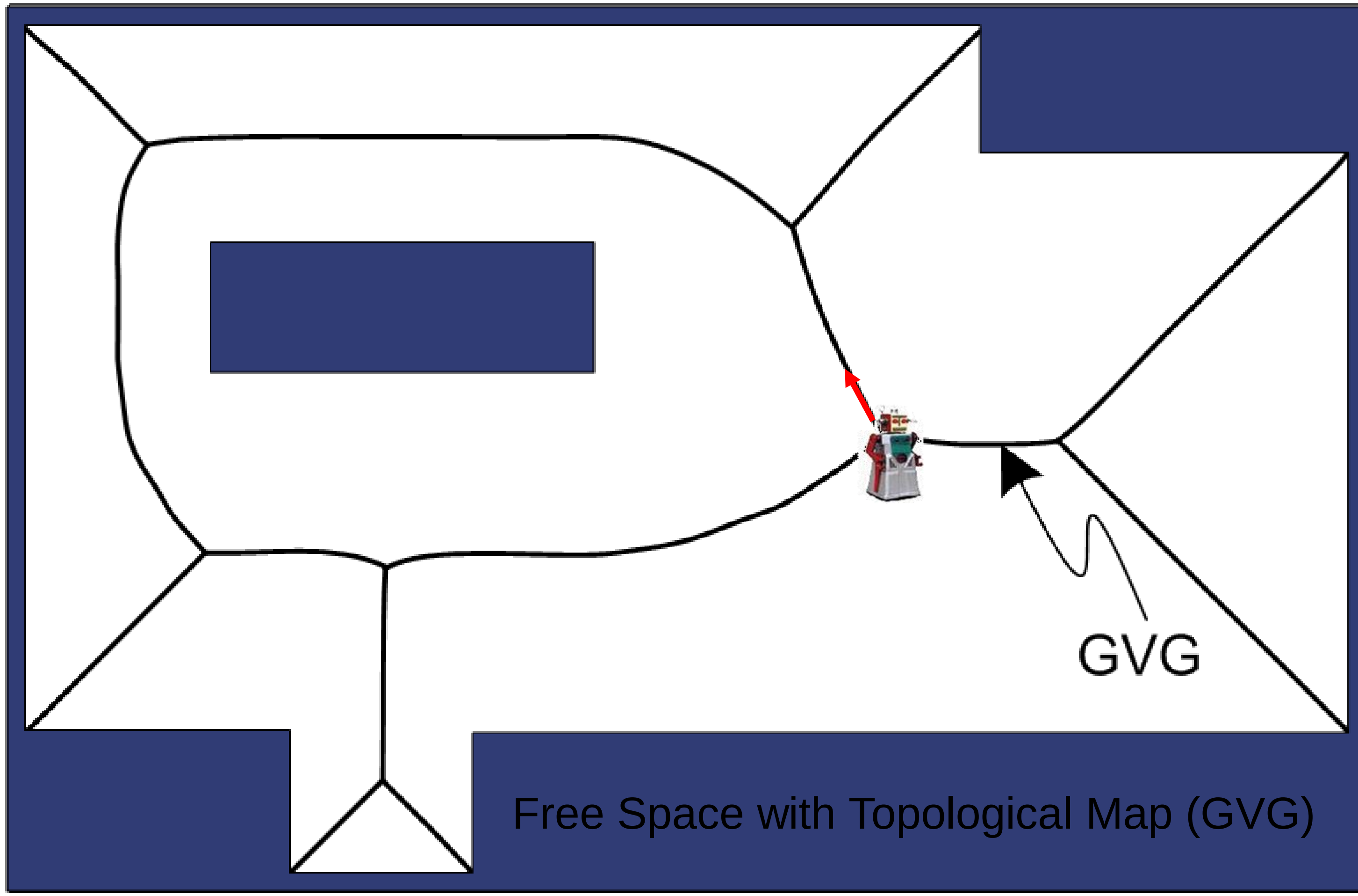
- Access GVG



Free Space with Topological Map (GVG)

Generalized Voronoi Graph (GVG)

- Access GVG
- Follow Edge
- Home to the MeetPoint
- Select Edge

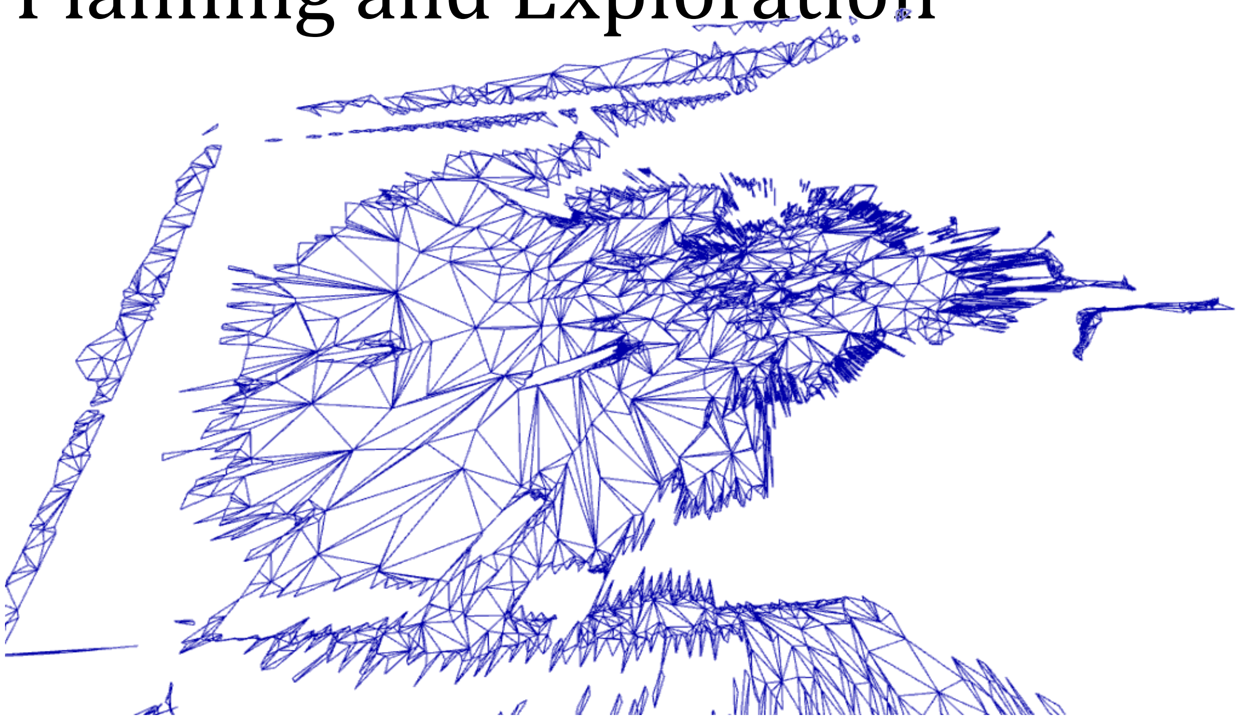


Exploration via Graph Search

- Exhaustive Depth First Search
- Breadth-First Search
- Heuristics

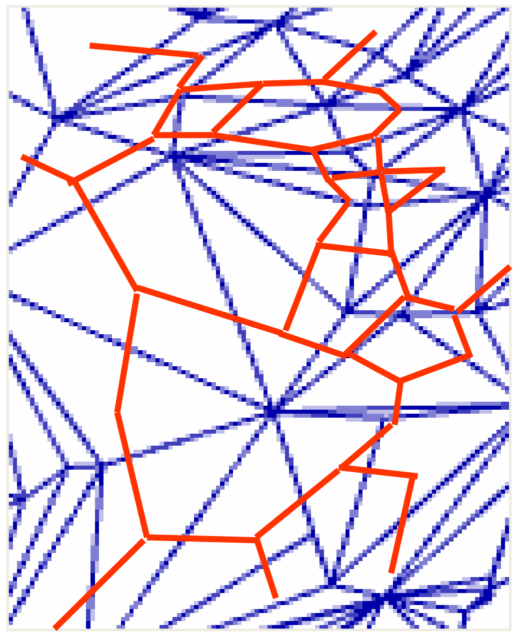
Irregular Triangular Mesh (ITM)

- Terrain Representation
- Underlying Topological Structure
- Path Planning and Exploration



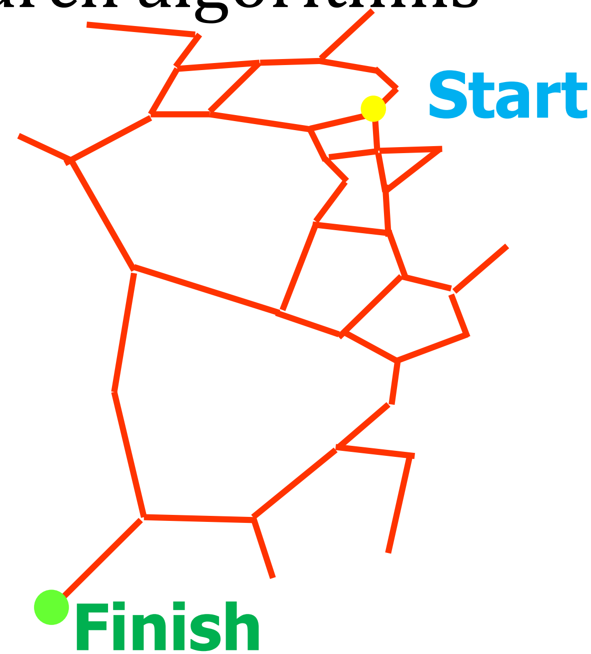
From 2.5D Representation to Topological

- Convert ITM into Connected Graph



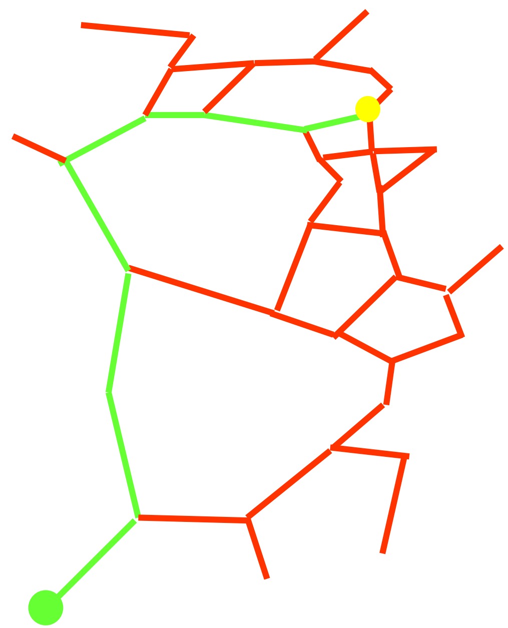
Planning

- Convert ITM into Connected Graph
- Planning using Graph Search Algorithms:
  - Dijkstra, A* search algorithms



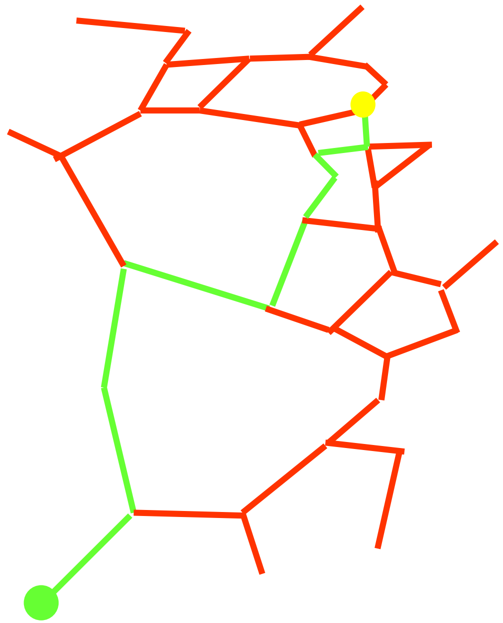
Planning

- Convert ITM into Connected Graph
- Path Planning using Graph Search Algorithms:
  - Dijkstra, A* search algorithms
- Different Cost Functions Q
  - Number of triangles $Q = 1$



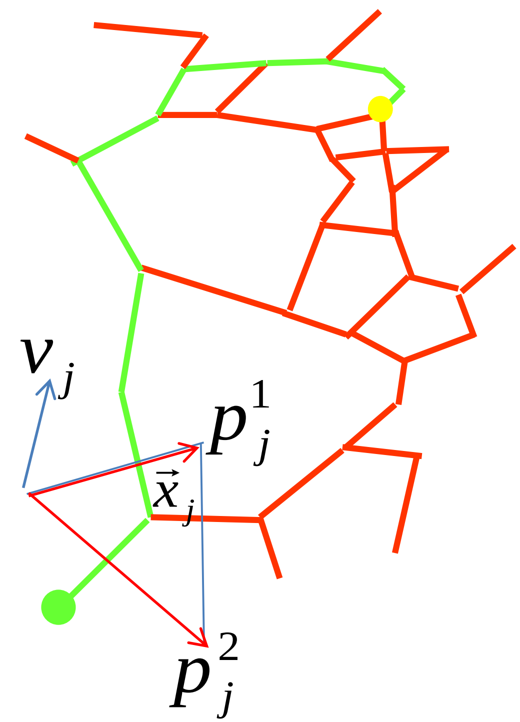
Planning

- Convert ITM into Connected Graph
- Path Planning using Graph Search Algorithms:
  - Dijkstra, A*
- Different Cost Functions Q
  - Number of triangles
  - Euclidian distance $Q = \left\| \vec{x}_i - \vec{x}_j \right\|$



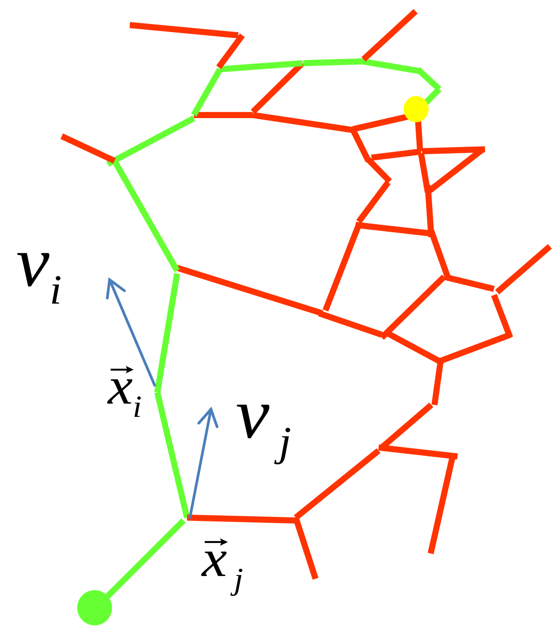
Planning

- Convert ITM into Connected Graph
- Path Planning using Graph Search Algorithms:
  - Dijkstra, A*
- Different Cost Functions Q
  - Number of triangles
  - Euclidian distance
  - Slope of each triangle $v_j = \dfrac{p_j^1 \times p_i^2}{\|p_j^1\|\|p_j^2\|}$



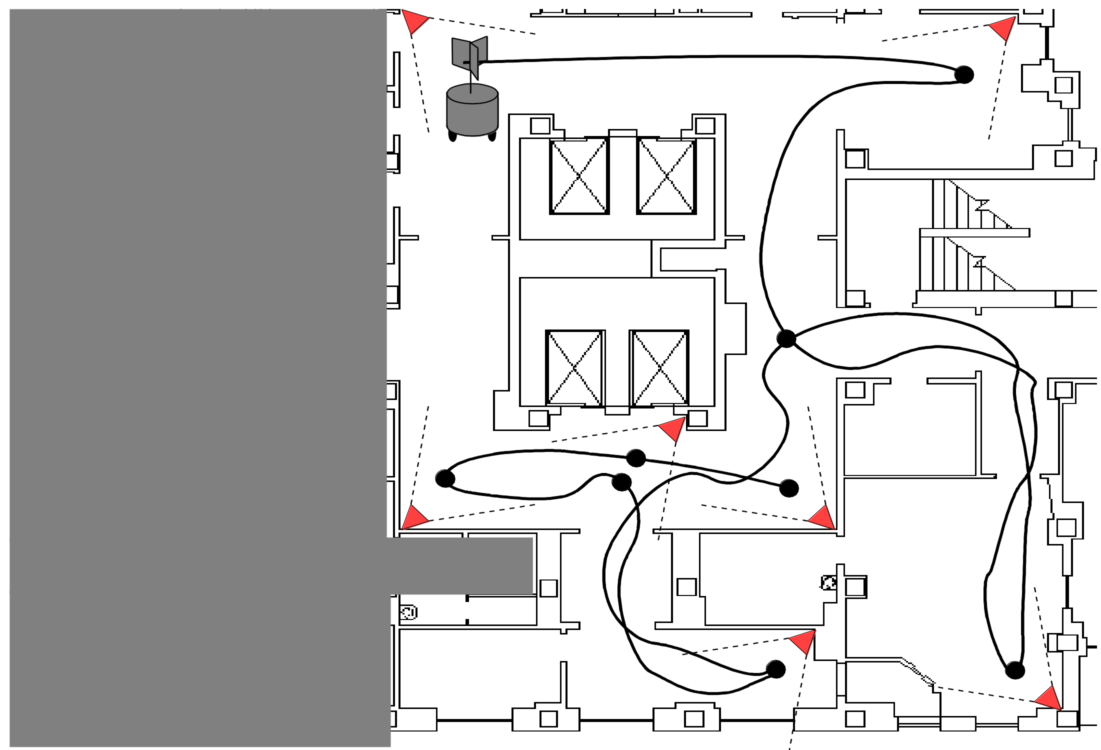
Planning

- Convert ITM into Connected Graph
- Path Planning using Graph Search Algorithms:
  - Dijkstra, A*
- Different Cost Functions Q
  - Number of triangles
  - Euclidian distance
  - Slope of each triangle
  - Cross triangle slope



Exploration Planning Problem

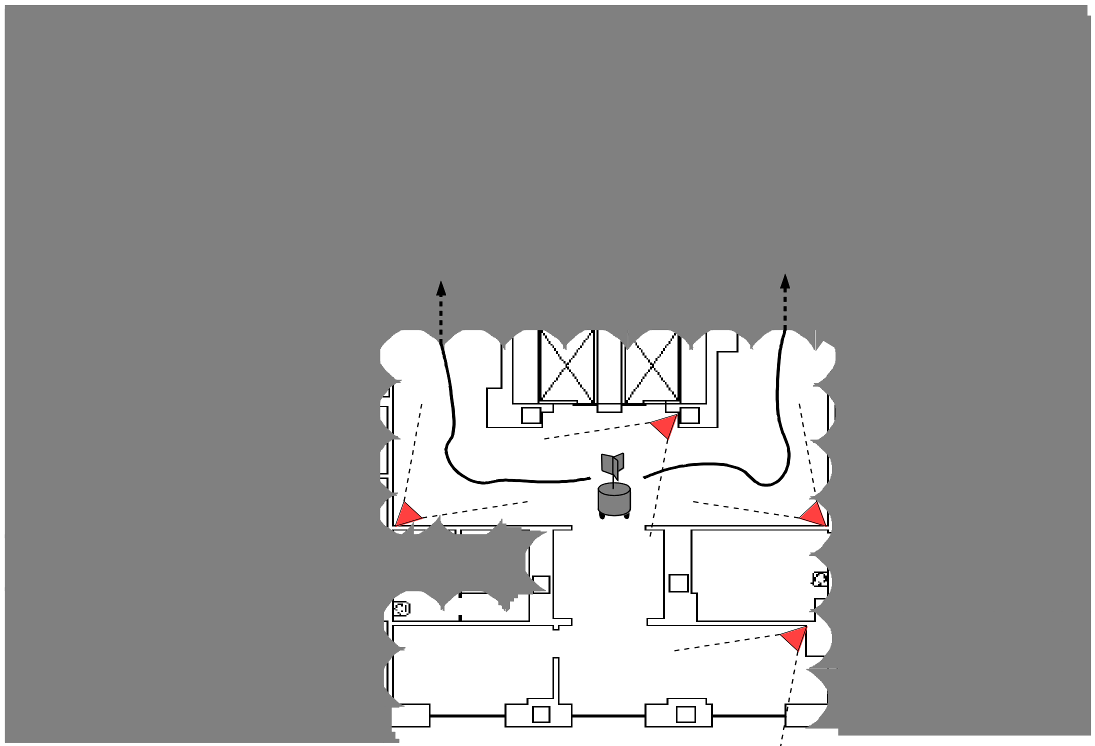
Two fundamental problems for path planning during exploration and mapping:



Exploration Planning Problem

Two fundamental problems for path planning during exploration and mapping:

- – Planning for re-localization
- – Planning the exploration of new territory



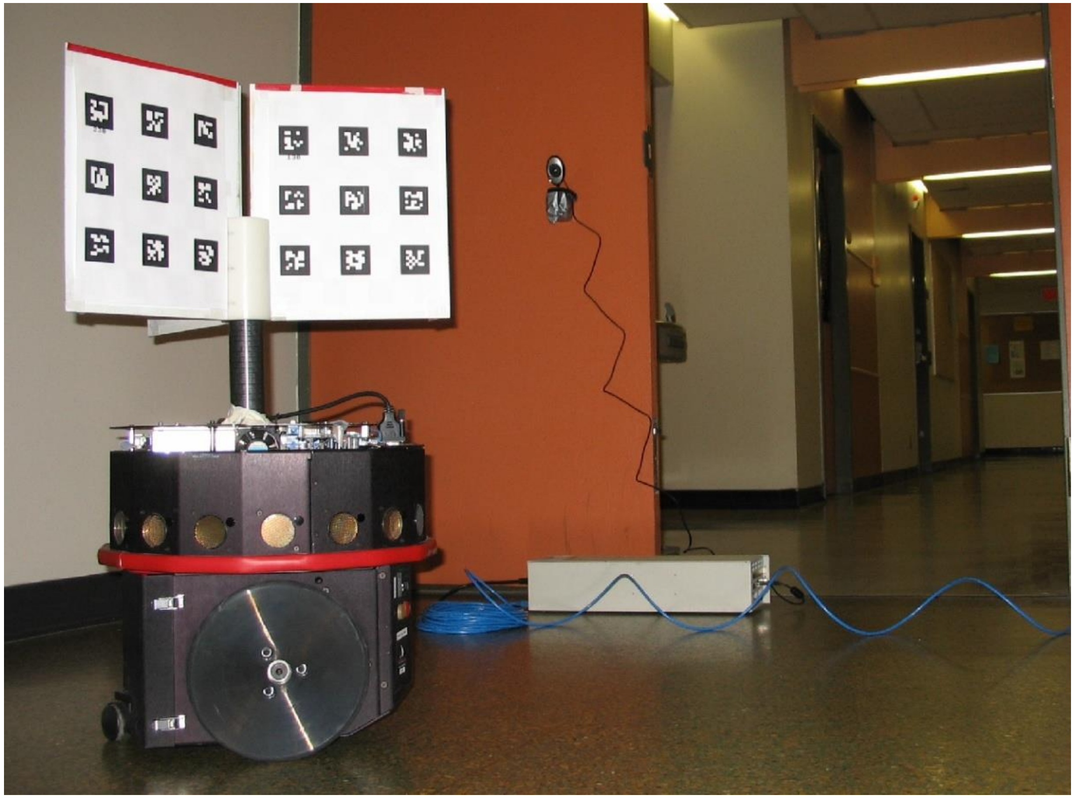
Previous Localization Planning

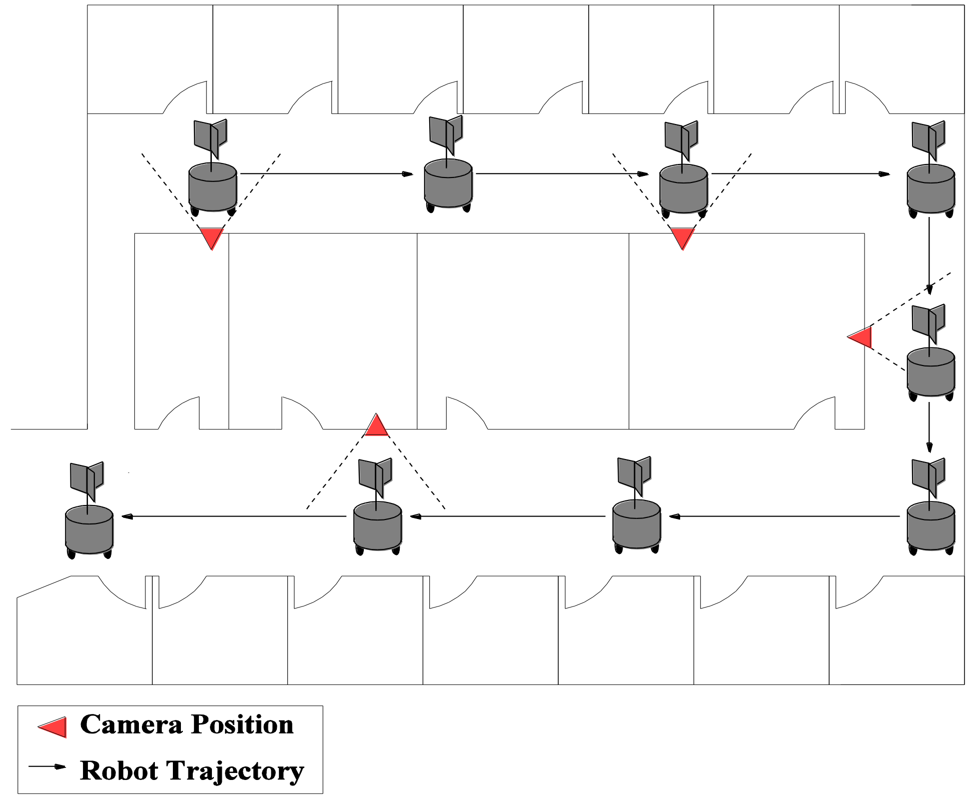
- Reduce measure of map or position entropy
- Variety of graph search planning algorithms (breadth first, A*-search, RRT)
- Evaluate paths with simulation, or Cramer-Rao bounds for expected uncertainty
- e.g. [Fox et al RAS 1998], [Sim and Roy ICRA 2005], [He et al ICRA 2008], [Censi et al ICRA 2008]

Previous Exploration Planning

- Includes motion into unexplored regions
- Typically requires prior knowledge of environment properties or rough layout
- Computation of exploration effects is a challenge
- e.g. [Bourque and Dudek IROS 1999], [Bourgault et al IROS 2002], [Kollar and Roy IJRR 2008]

Exploring a Camera Sensor Network





D. Meger, I. Rekleitis, and G. Dudek. “Heuristic Search Planning to Reduce Exploration Uncertainty”, IROS 2008.

Heuristic Search Planning Method

- Solution to exploration planning for camera sensor networks
  - Composed of two alternated steps: exploration and re-localization
  - Combined distance and uncertainty cost function
  - Heuristic search for good paths

Exploration and Uncertainty Reduction

- Decision (exploration vs exploitation)
- Target Node
- Path Planning through the known graph
- Exploration Strategies

Exploration and Uncertainty Reduction

- Decision (exploration vs. exploitation)
  - **Epsilon-Greedy**
  - **Epsilon-First**
  - **Adaptive**
  - **Bounded Uncertainty**
- Target Node
- Path Planning through the known graph
- Exploration Strategies

Exploration and Uncertainty Reduction

- Decision (exploration vs. exploitation)
- Target Node (Exploration)
  - **Random**
  - **Shortest distance**
  - **Maximum Uncertainty**
  - **Minimum Uncertainty**
- Path Planning through the known graph
- Exploration Strategies

Exploration and Uncertainty Reduction

- Decision (exploration vs. exploitation)
- Target Node (Relocalization)
  - **Maximum Uncertainty**
- Path Planning through the known graph
- Exploration Strategies

Exploration and Uncertainty Reduction

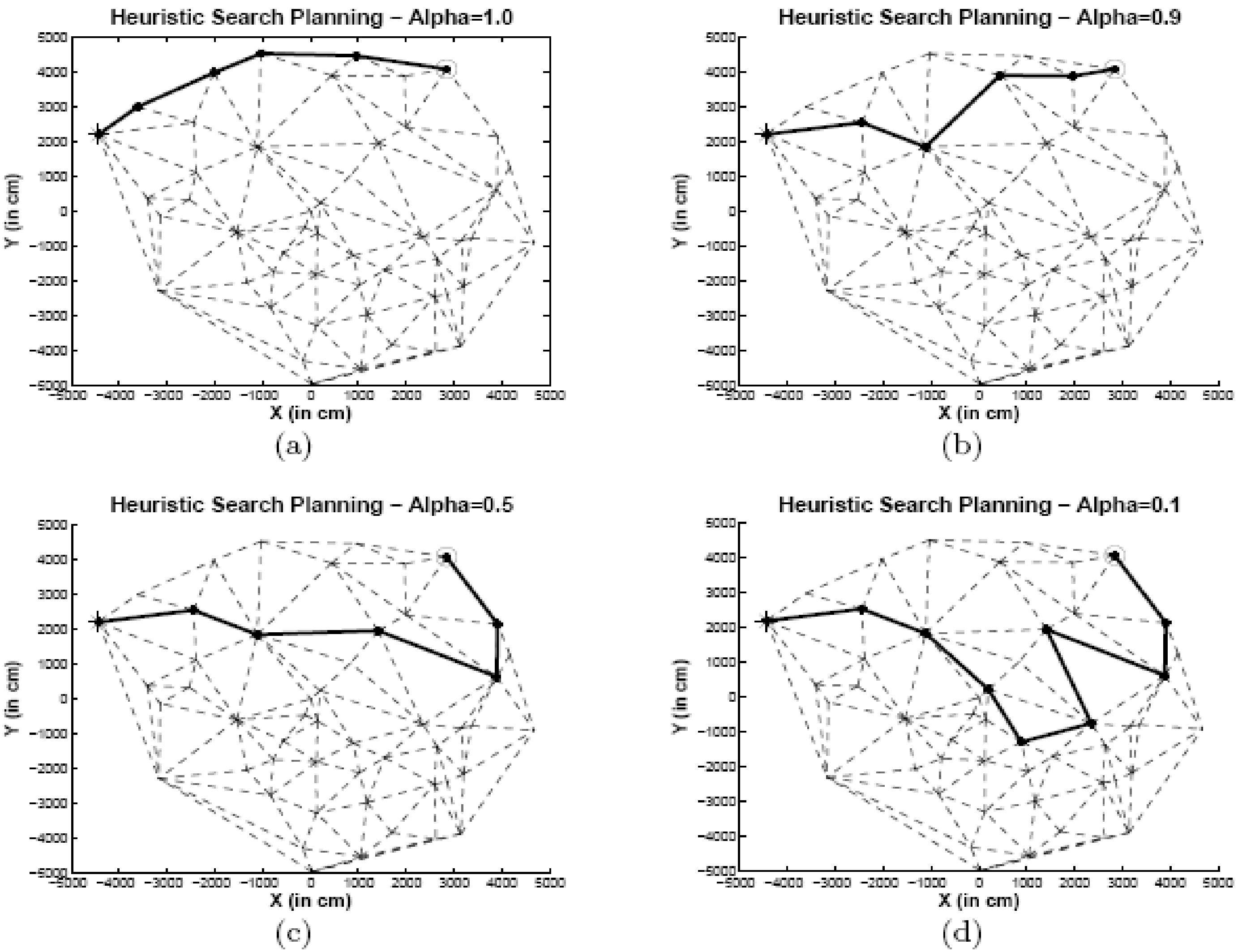
- Decision (exploration vs. exploitation)
- Target Node
- Path Planning through the known graph
  - Work with D. Meger and G. Dudek [IROS 2008]
  - A* based strategy
  - Cost: $C(p) = \omega_d length(p) + \omega_u trace(P(p))$
  - Distance-based "cost-to-go" heuristic function *h* used to compute estimated cost

$$C(n) = f(n) + h(n)$$

Estimated cost through n — Cost so far — Estimated cost to go

- Exploration Strategies

Effect of α Parameter for Relocalization



Heuristic Search

- Graph search to optimize cost function

$$C(p) = \omega_d length(p) + \omega_u trace(\Sigma(p))$$

- Heuristic search allows considering only a fraction of the paths, ordered by expected cost
- Distance-based "cost-to-go" heuristic function *h* used to compute estimated cost

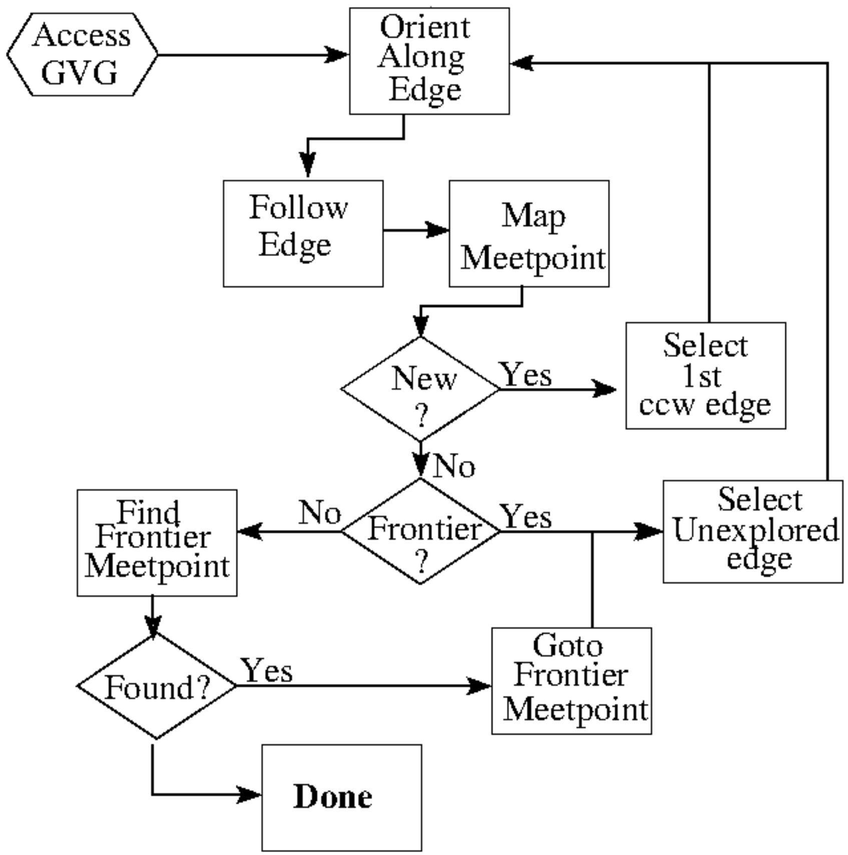
$$C(n) = f(n) + h(n)$$

Estimated cost through n — Cost so far — Estimated cost to go

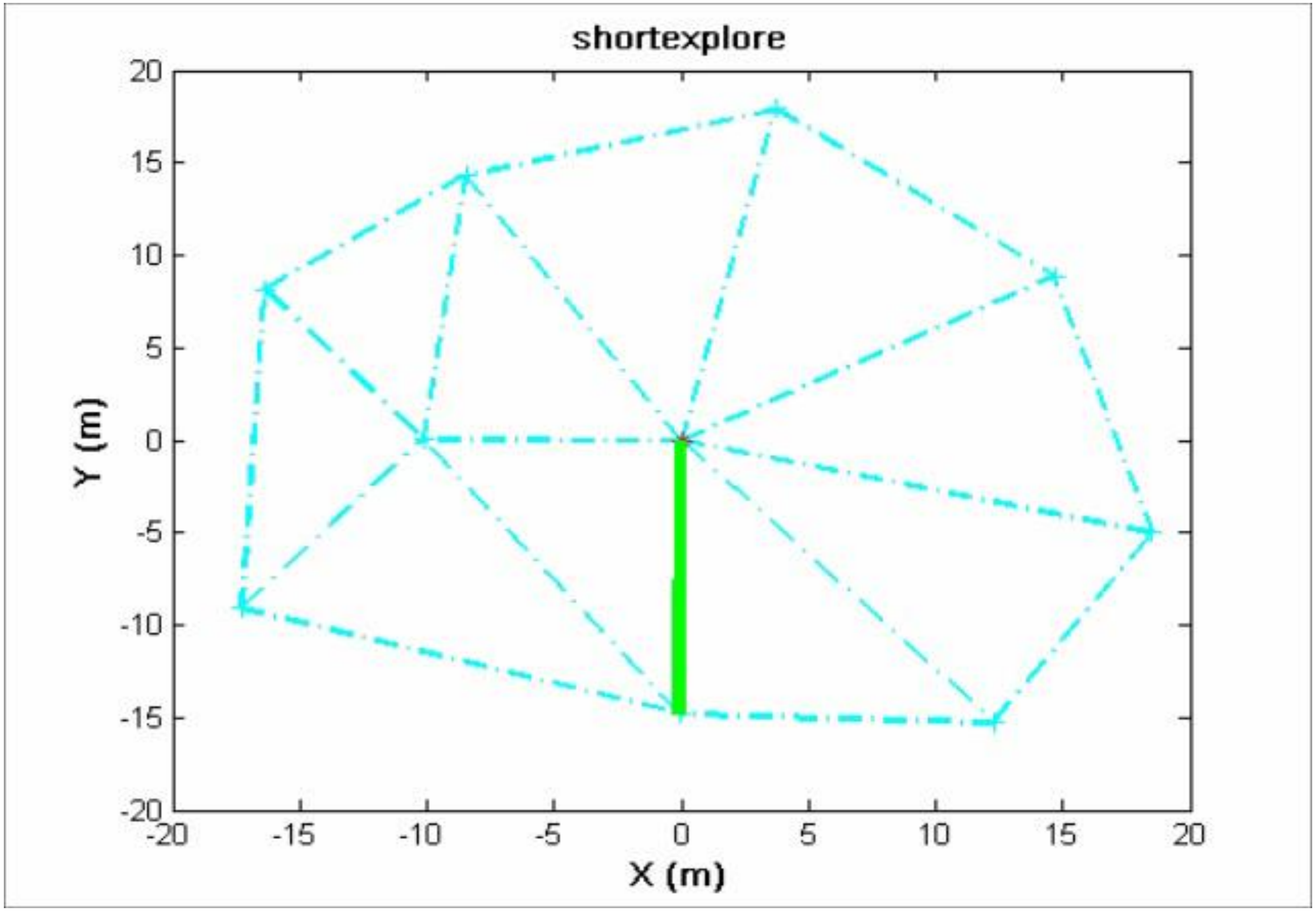
Exploration and Uncertainty Reduction

- Decision (exploration vs. exploitation)
- Target Node
- Path Planning through the known graph
- Exploration Strategies
  - One Step Exploration
  - Ear based exploration

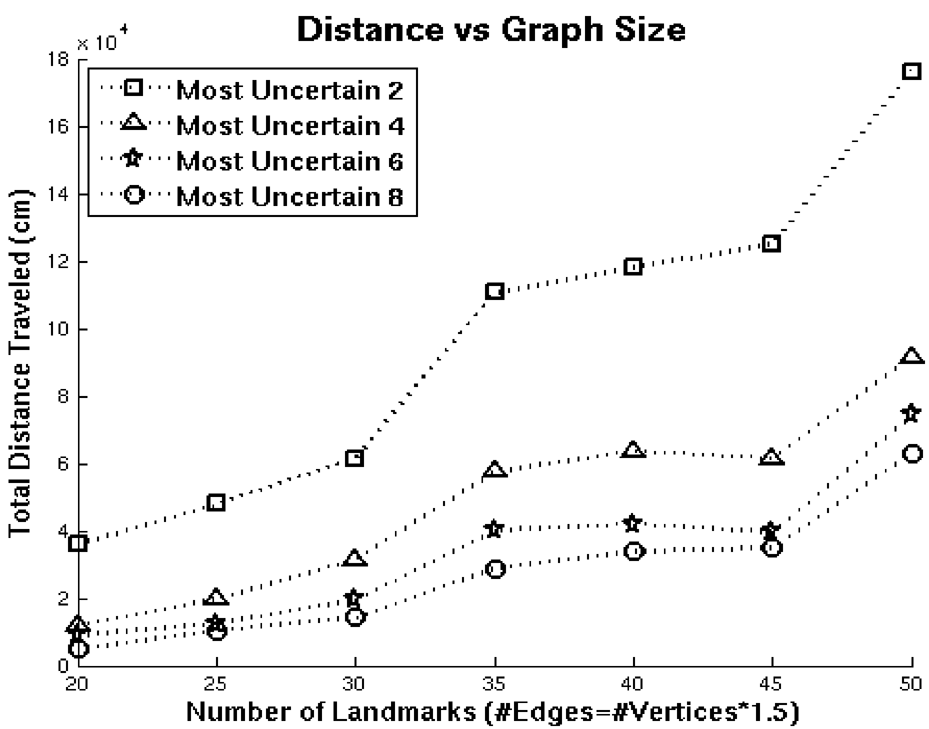
Ear-Based Exploration Algorithm

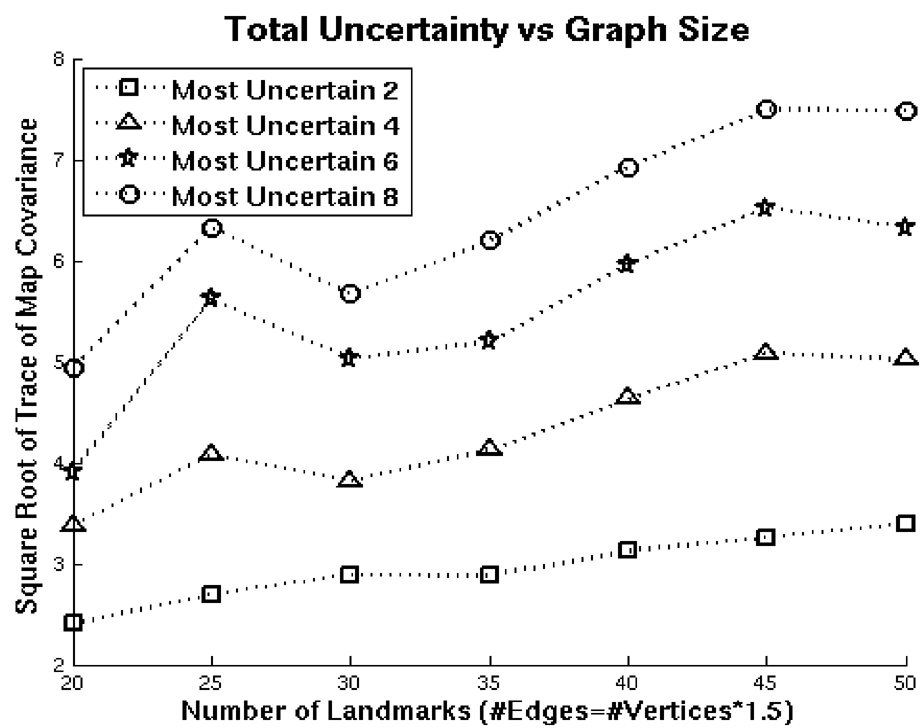


Shortest Node P(exploit)=0.3

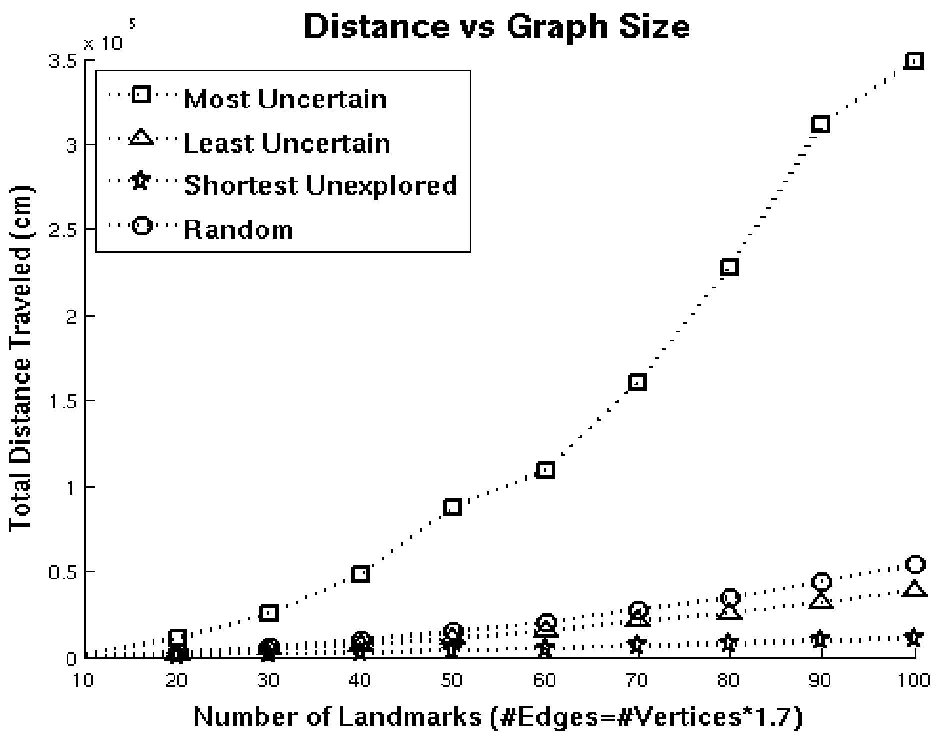


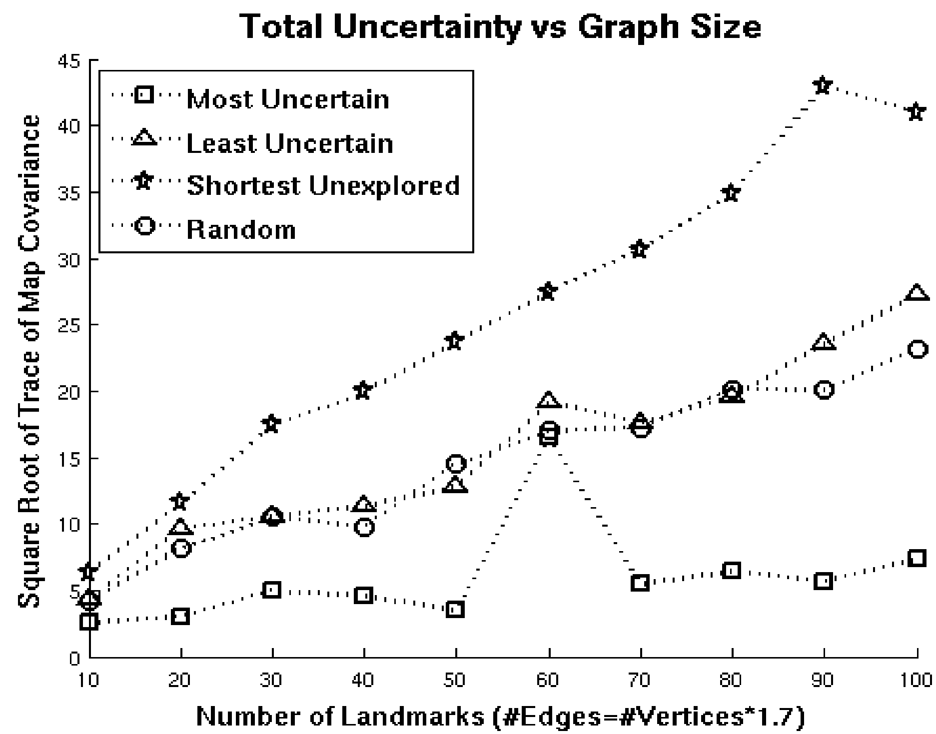
Experimental Results
Bounded Uncertainty





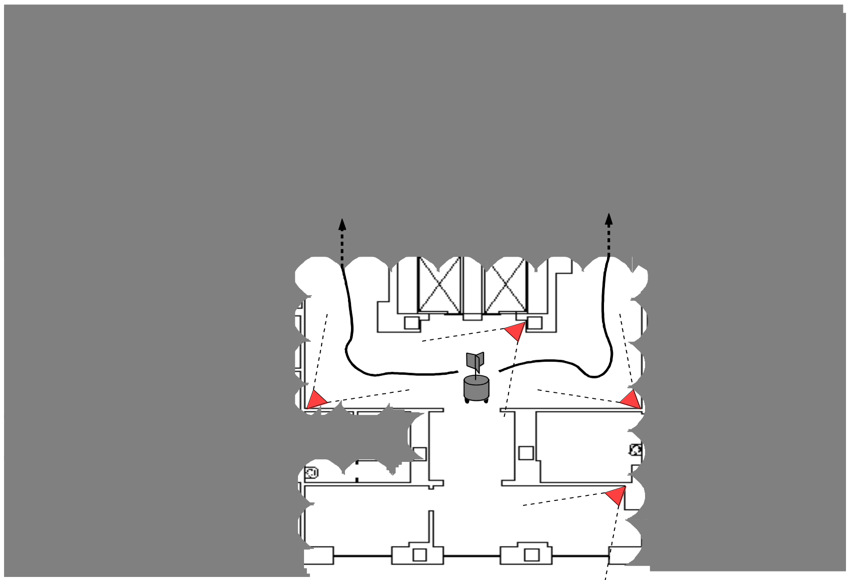
Experimental Results
Different Strategies





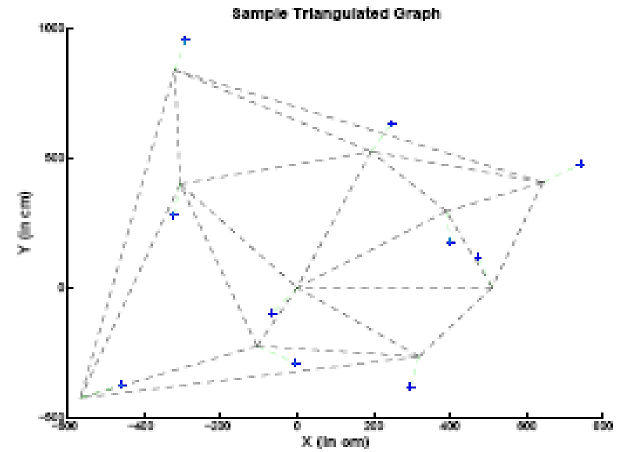
Planning Exploratory Steps

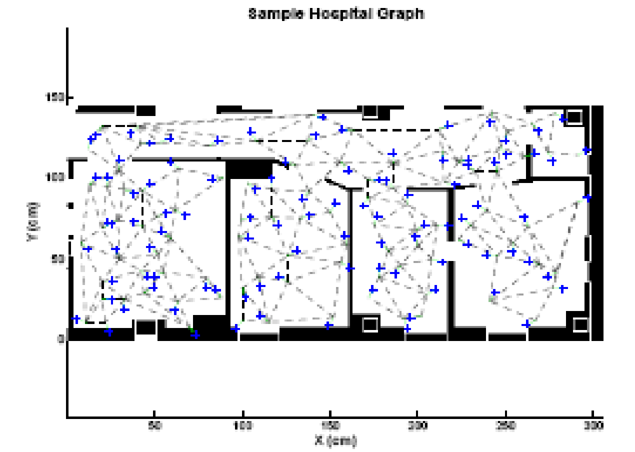
- Choose motion in unexplored space to locate additional camera nodes
- Planner cannot simulate these paths
- Evaluated 2 strategies: 1) nearest camera and 2) a randomly selected camera



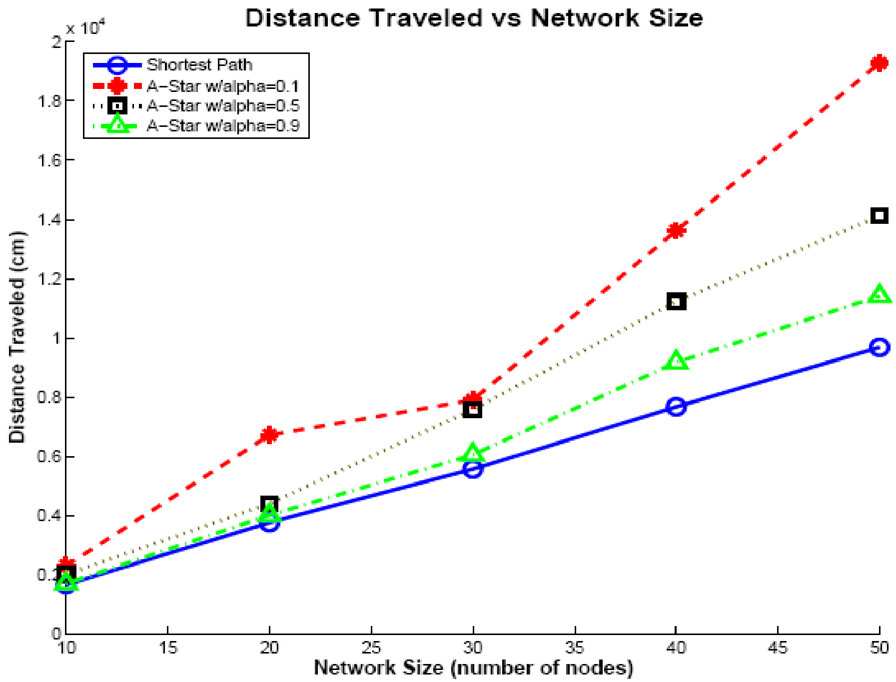
Simulation Results

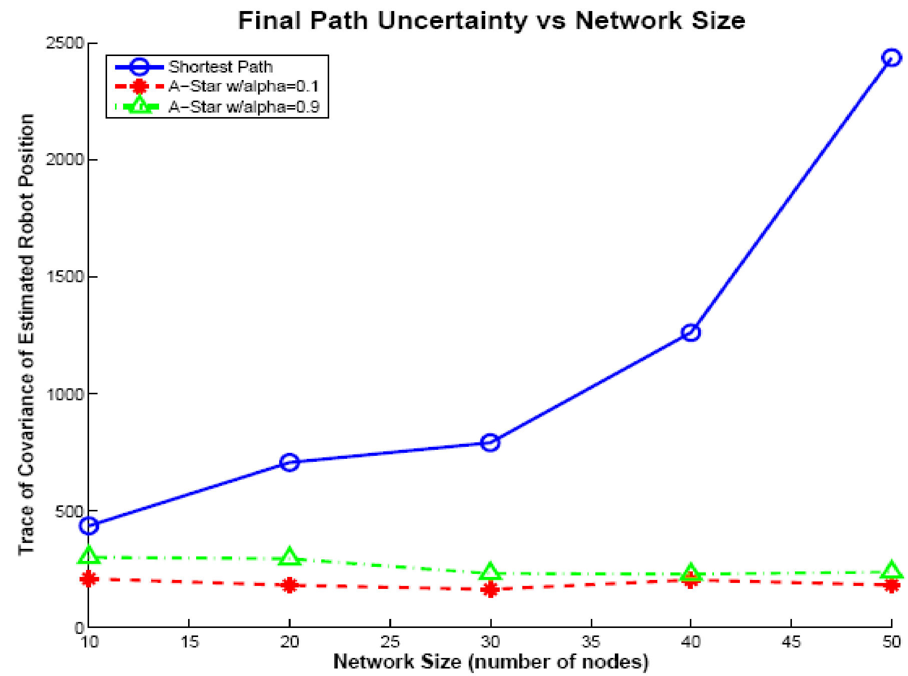
- Compared planners over many trials
- 3 realistic network types (2 shown)
- 3 methods for comparison:
  - Depth-first
  - Return to origin
  - Return to nearest explored



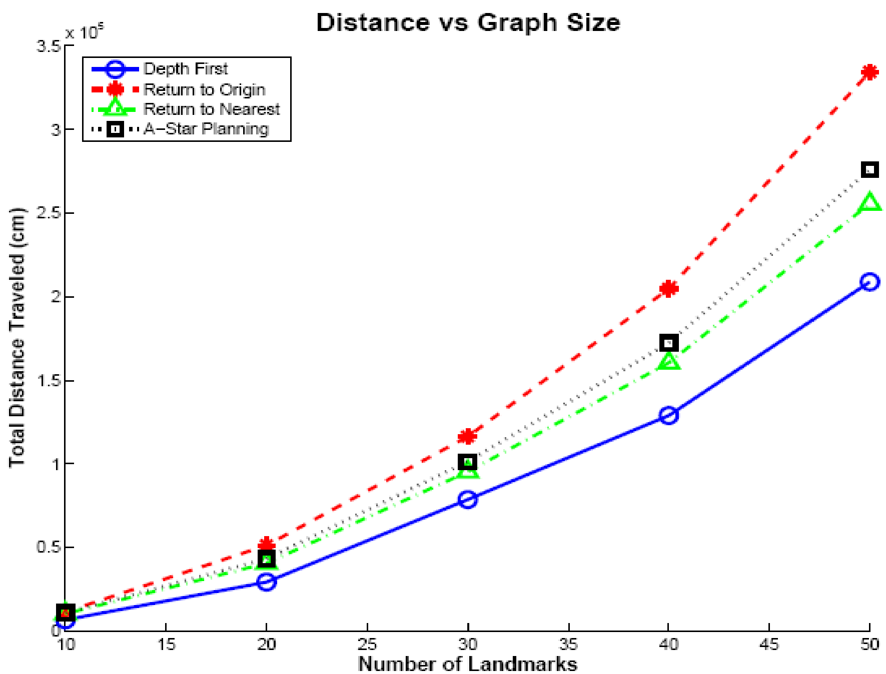


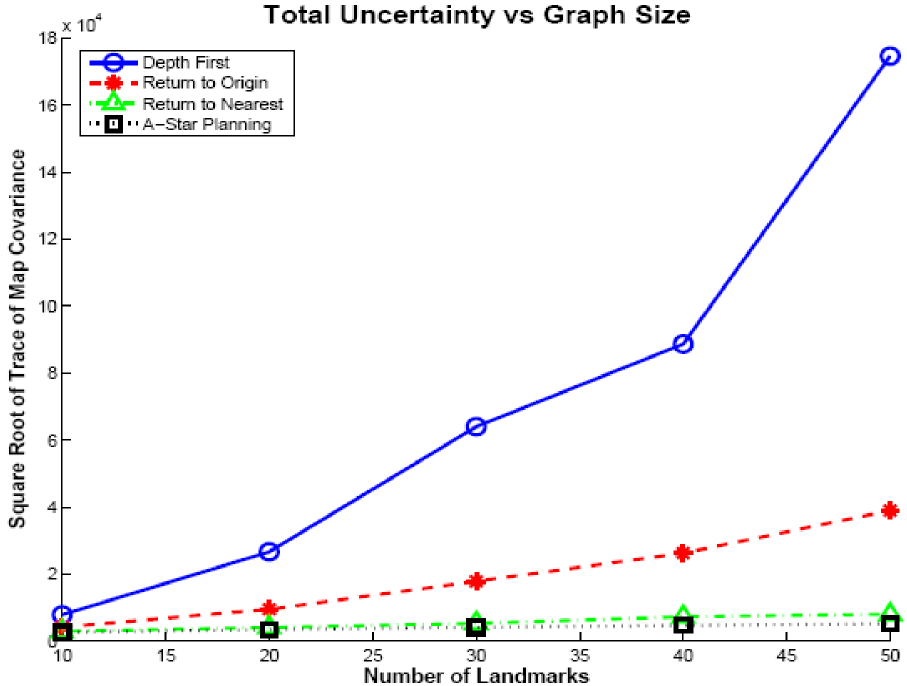
Simulated Relocalization Results



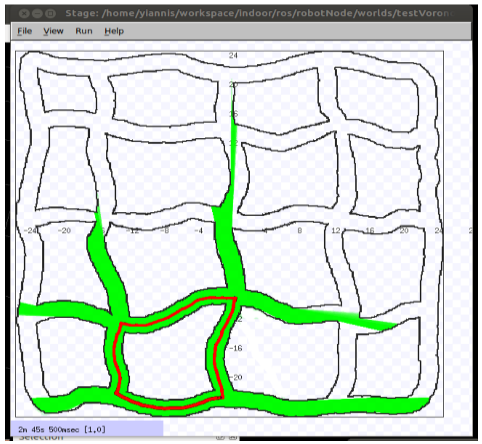


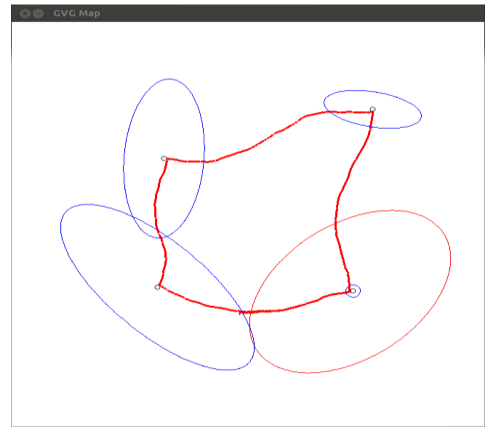
Simulated Exploration Results

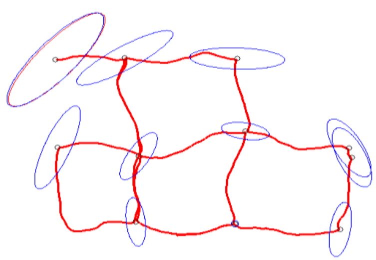


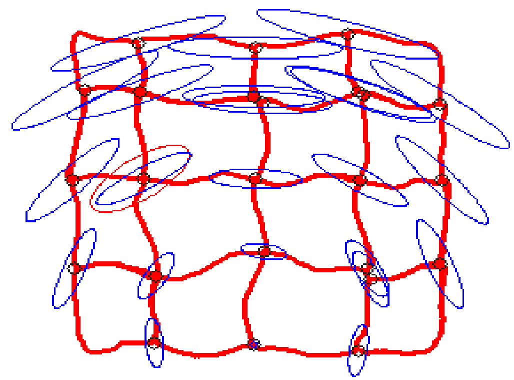


Exploration of the GVG



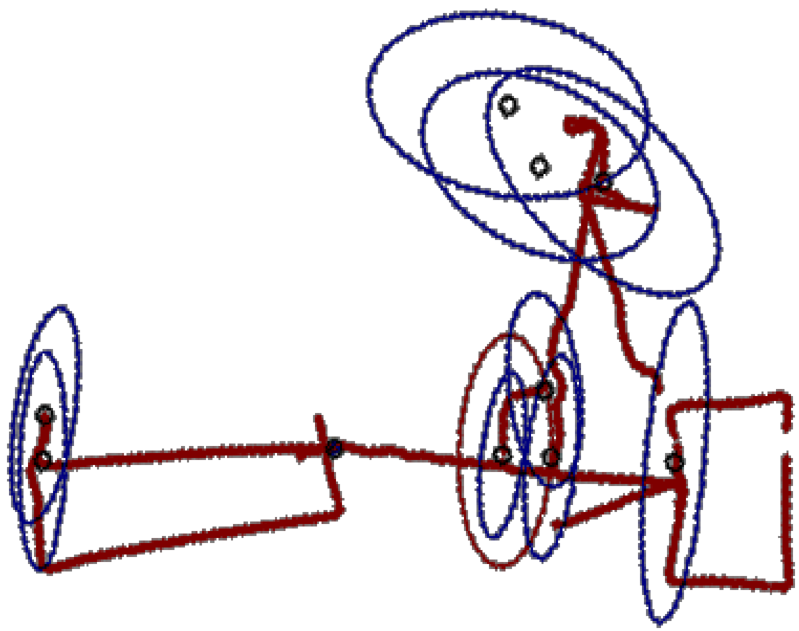


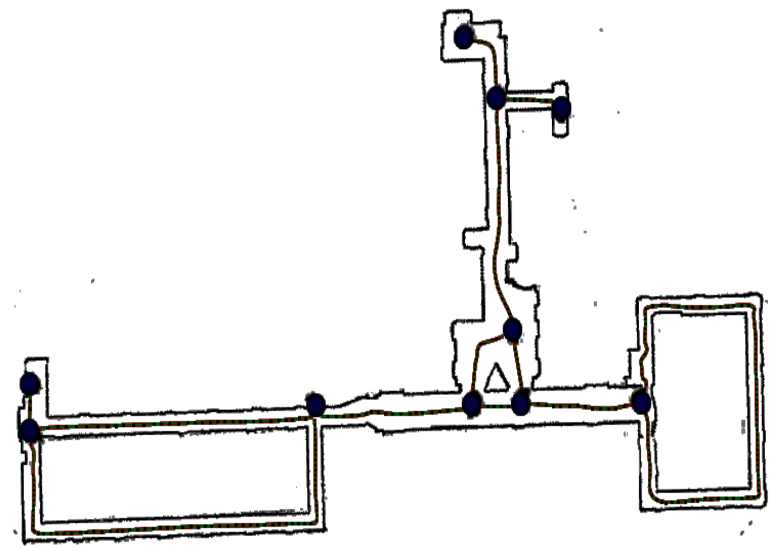




Simulation in StageRos

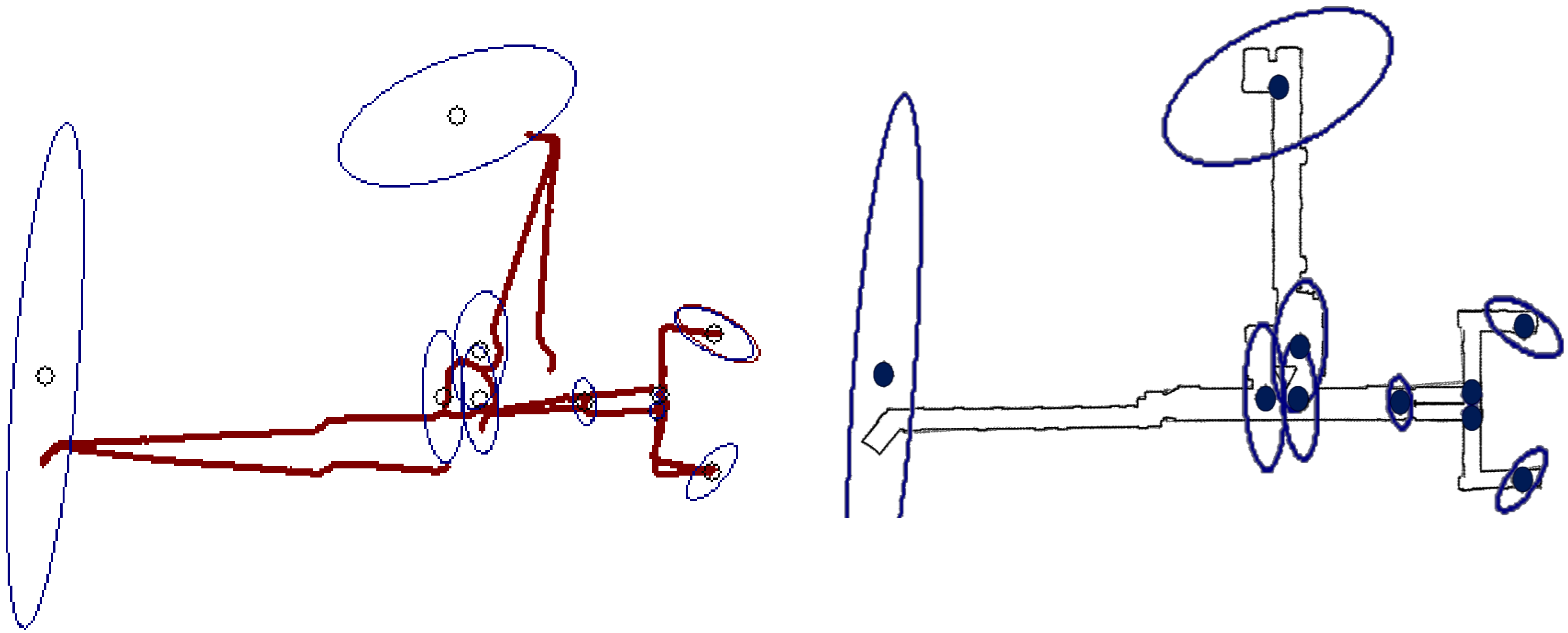
Exploration of the GVG





Real environment, McConnell 4th floor

Exploration of the GVG



Real environment, McConnell 3rd floor

Video of the Ear-based Exploration

Key Points

- Mapping requires exploration
- Exploration strategies depend on the representation
- Topological representations are the most convenient for exploration
- Two objectives:
  - Explore new territory
  - Improve the accuracy by relocalization

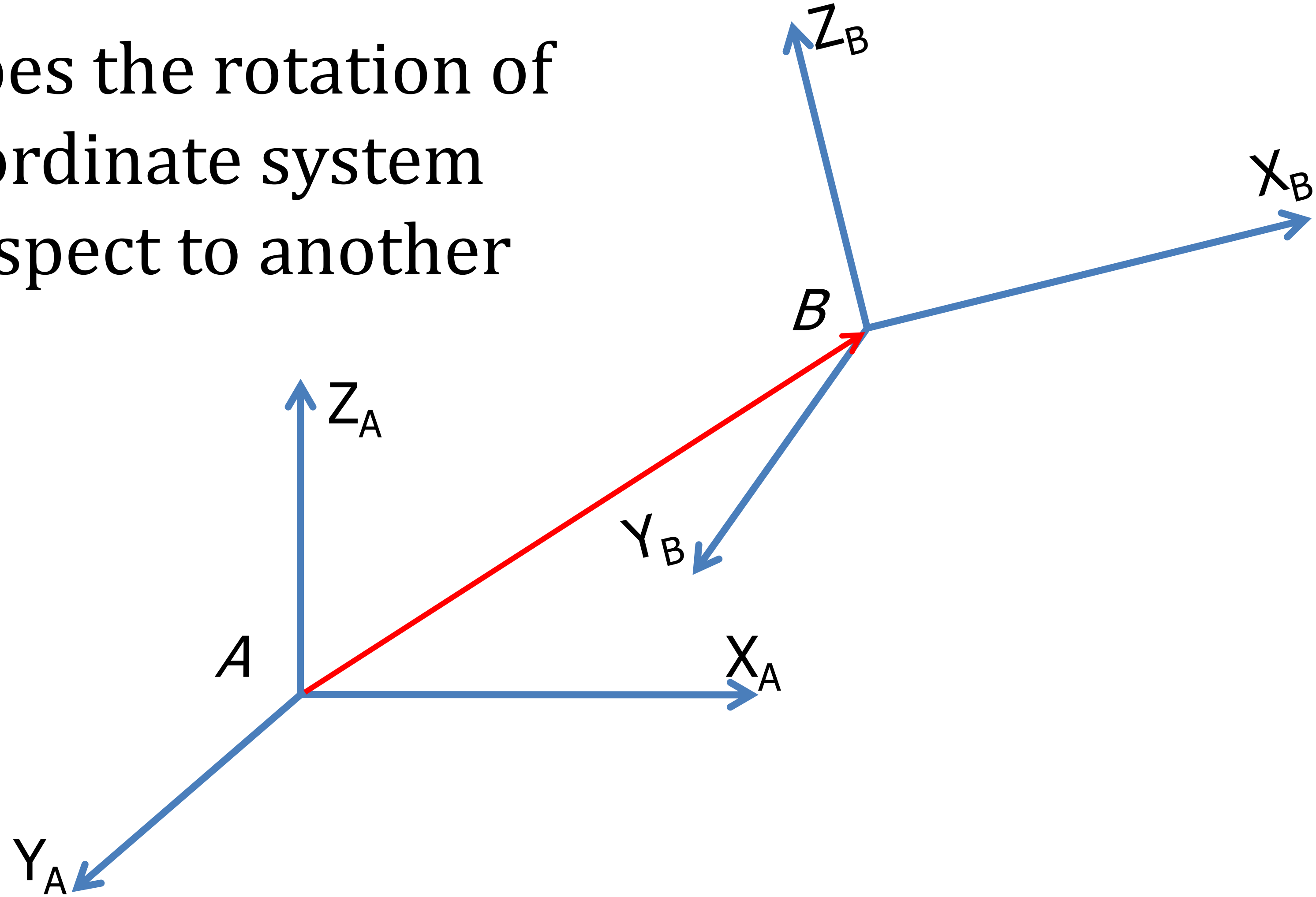
References

- B. J. Kuipers and Y.-T. Byun. “A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations”. In *Journal of Robotics and Autonomous Systems*, 8: 47-63, 1991.
- H. Choset, J. Burdick, “Sensor based planning, part ii: Incremental construction of the generalized voronoi graph”. In IEEE Conference on Robotics and Automation, pp. 1643 – 1648, 1995.
- B. Yamauchi, “Frontier-based exploration using multiple robots”, In Second International Conference on Autonomous Agents, Minneapolis, MN, 1998, pp. 47–53.
- Makarenko, A.A. Williams, S.B. Bourgault, F. Durrant-Whyte, “An experiment in integrated exploration”, In IEEE/RSJ International Conference on Inte.lligent Robots and System, vol.1, pp 534-539, 2002.
- Stachniss, C. Hahnel, D. Burgard, W. , “Exploration with active loop-closing for FastSLAM”. In IEEE/RSJ International Conference on Intelligent Robots and Systems. vol.2, pp 1505-1510, 2004.
- R. Sim and N. Roy, “Global a-optimal robot exploration in slam”. In *International Conference on Robotics and Automation, pp. 661*– 666, *2005*.
- T. Kollar and N. Roy, “Using reinforcement learning to improve exploration trajectories for error minimization”. *In of the IEEE International Conference on Robotics and Automation, 2006.*
- R. Martinez-Cantin, N. de Freitas, A. Doucet, and J. Castellanos, “Active policy learning for robot planning and exploration under Uncertainty”. In Robotics: Science and Systems, 2007.
- D. Meger, I. Rekleitis, and G. Dudek. “Heuristic Search Planning to Reduce Exploration Uncertainty”. In *IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp 3382-3399, 2008.

- QUESTIONS?

Orientation Representations

- Describes the rotation of one coordinate system with respect to another

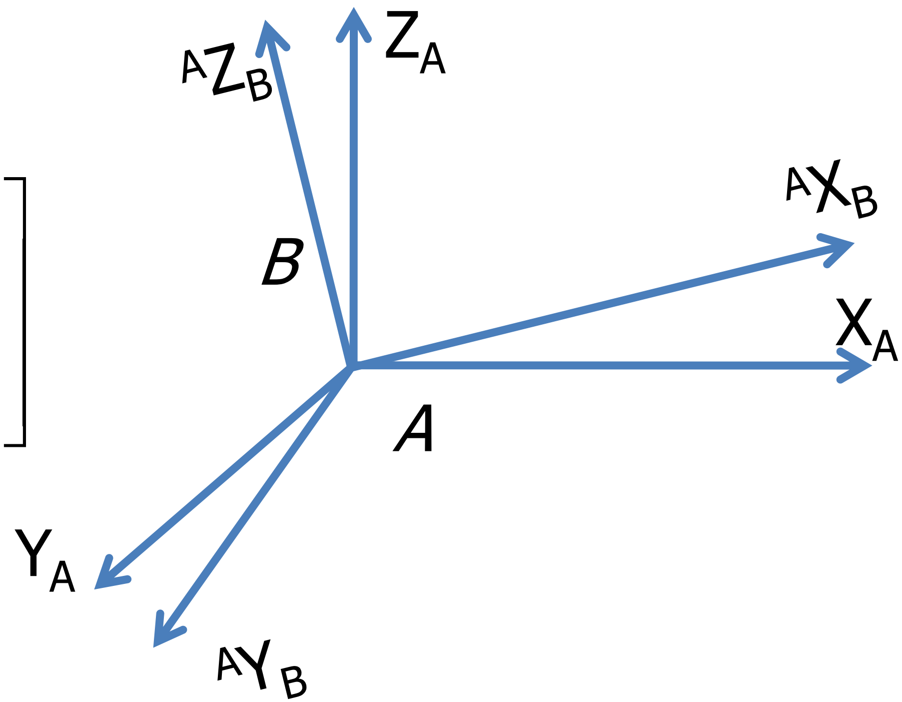


Rotation Matrix

- Write the unit vectors of B in the coordinate system of A.
- Rotation Matrix:

$$
{}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_{B} & {}^{A}\hat{Y}_{B} & {}^{A}\hat{Z}_{B} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}
$$

$$
= \begin{bmatrix} \hat{X}_{B} \cdot \hat{X}_{A} & \hat{Y}_{B} \cdot \hat{X}_{A} & \hat{Z}_{B} \cdot \hat{X}_{A} \\ \hat{X}_{B} \cdot \hat{Y}_{A} & \hat{Y}_{B} \cdot \hat{Y}_{A} & \hat{Z}_{B} \cdot \hat{Y}_{A} \\ \hat{X}_{B} \cdot \hat{Z}_{A} & \hat{Y}_{B} \cdot \hat{Z}_{A} & \hat{Z}_{B} \cdot \hat{Z}_{A} \end{bmatrix}
$$



Properties of Rotation Matrix

$$ {}^{B}_{A}R = {}^{A}_{B}R^{T} $$

$$ {}^{A}_{B}R^{T}\,{}^{A}_{B}R = I_{3} $$

$$ {}^{A}_{B}R = {}^{B}_{A}R^{-1} = {}^{B}_{A}R^{T} $$

Coordinate System Transformation

$$M = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0_{3\times 1} & 1 \end{bmatrix}$$

where R is the rotation matrix and T is the translation vector

Rotation Matrix

- The rotation matrix consists of 9 variables, but there are many constraints. The minimum number of variables needed to describe a rotation is three.

Rotation Matrix-Single Axis

$$R_x(\theta)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\theta) & -\sin(\theta)\\ 0 & \sin(\theta) & \cos(\theta)\end{bmatrix}$$

$$R_y(\theta)=\begin{bmatrix}\cos(\theta) & 0 & \sin(\theta)\\ 0 & 1 & 0\\ -\sin(\theta) & 0 & \cos(\theta)\end{bmatrix}$$

$$R_z(\theta)=\begin{bmatrix}\cos(\theta) & -\sin(\theta) & 0\\ \sin(\theta) & \cos(\theta) & 0\\ 0 & 0 & 1\end{bmatrix}$$

Fixed Angles

- One simple method is to perform three rotations about the axis of the original coordinate frame:
  - X-Y-Z fixed angles

$$
{}^{A}_{B}R(\theta,\phi,\psi)=R_z(\psi)R_y(\phi)R_x(\theta)
$$

$$
=\begin{bmatrix} \cos(\psi)\cos(\phi) & \cos(\psi)\sin(\phi)\sin(\theta)-\sin(\psi)\cos(\theta) & \cos(\psi)\sin(\phi)\cos(\theta)+\sin(\psi)\sin(\theta) \\ \sin(\psi)\cos(\phi) & \sin(\psi)\sin(\phi)\sin(\theta)+\cos(\psi)\cos(\theta) & \sin(\psi)\sin(\phi)\cos(\theta)+\cos(\psi)\sin(\theta) \\ -\sin(\phi) & \cos(\phi)\sin(\theta) & \cos(\theta)\cos(\psi) \end{bmatrix}
$$

- There are 12 different combinations

Inverse Problem

- From a Rotation matrix find the fixed angle rotations:

$$ {}_B^A R(\theta,\phi,\psi) = {}_B^A R \Rightarrow $$

$$ \begin{bmatrix} \cos(\psi)\cos(\phi) & \cos(\psi)\sin(\phi)\sin(\theta)-\sin(\psi)\cos(\theta) & \cos(\psi)\sin(\phi)\cos(\theta)+\sin(\psi)\sin(\theta) \\ \sin(\psi)\cos(\phi) & \sin(\psi)\sin(\phi)\sin(\theta)+\cos(\psi)\cos(\theta) & \sin(\psi)\sin(\phi)\cos(\theta)+\cos(\psi)\sin(\theta) \\ -\sin(\phi) & \cos(\phi)\sin(\theta) & \cos(\theta)\cos(\psi) \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} $$

thus:

$$ \phi = A\tan 2\left(-r_{31}, \sqrt{\left(r_{11}^2 + r_{21}^2\right)}\right) $$

$$ \psi = A\tan 2\left(r_{21}/\cos(\phi), r_{11}/\cos(\phi) \right) $$

$$ \theta = A\tan 2\left(r_{32}/\cos(\phi), r_{33}/\cos(\phi) \right) $$

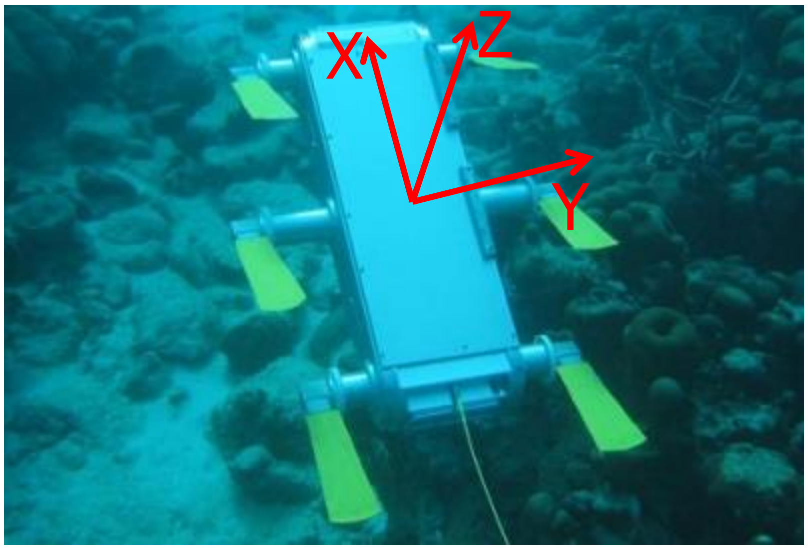
Euler Angles

- **ZYX**: Starting with the two frames aligned, first rotate about the Z_B axis, then by the Y_B axis and then by the X_B axis. The results are the same as with using XYZ fixed angle rotation.

- There are 12 different combination of Euler Angle representations

Euler Angles

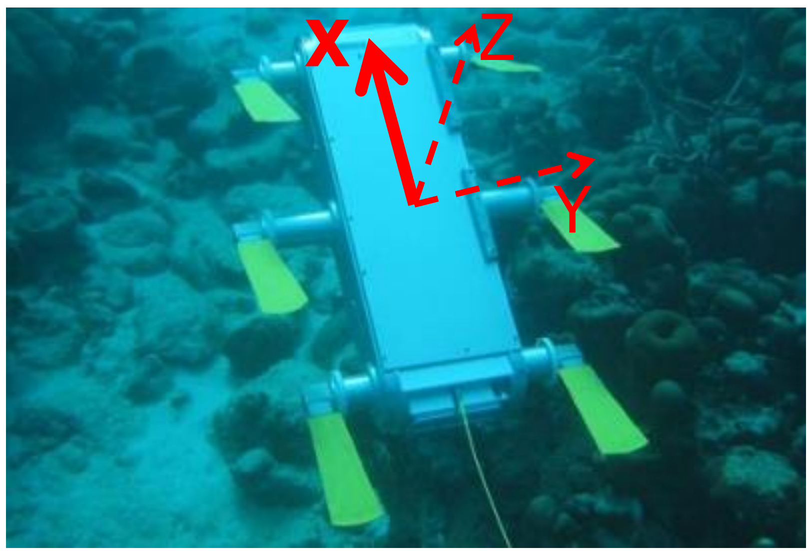
- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw



Euler Angles

- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

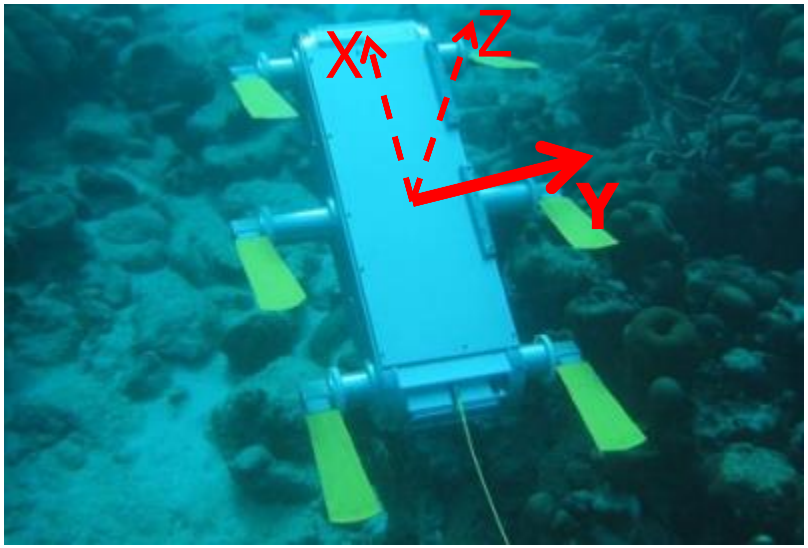
Roll



Euler Angles

- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

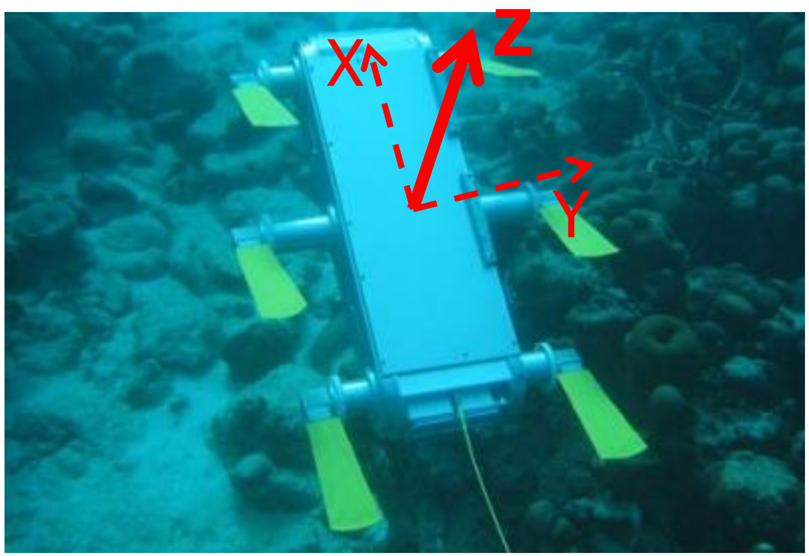
Pitch



Euler Angles

- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

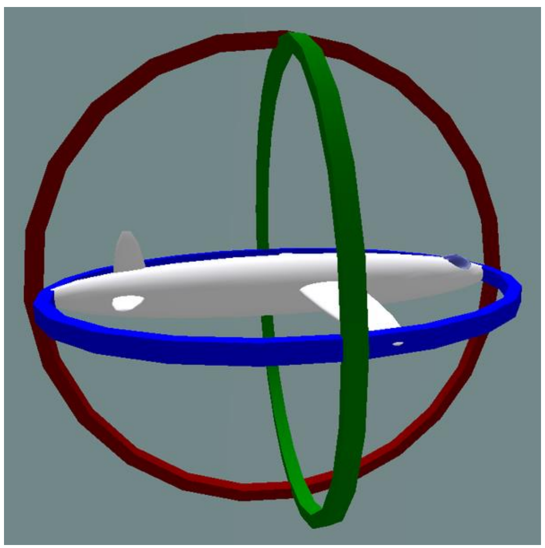
Yaw

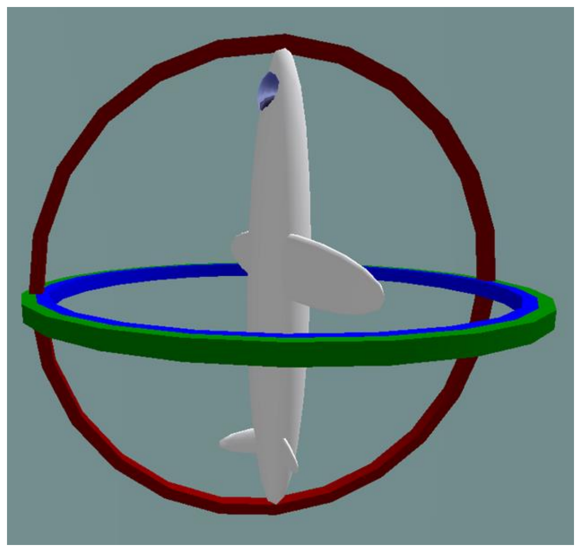


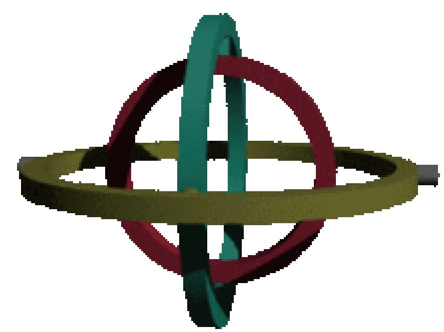
Euler Angle concerns: Gimbal Lock

Using the **ZYZ** convention

- (90°, 45°, −105°) ≡ (−270°, −315°, 255°) multiples of 360°
- (72°, 0°, 0°) ≡ (40°, 0°, 32°) singular alignment (Gimbal lock)
- (45°, 60°, −30°) ≡ (−135°, −60°, 150°) bistable flip

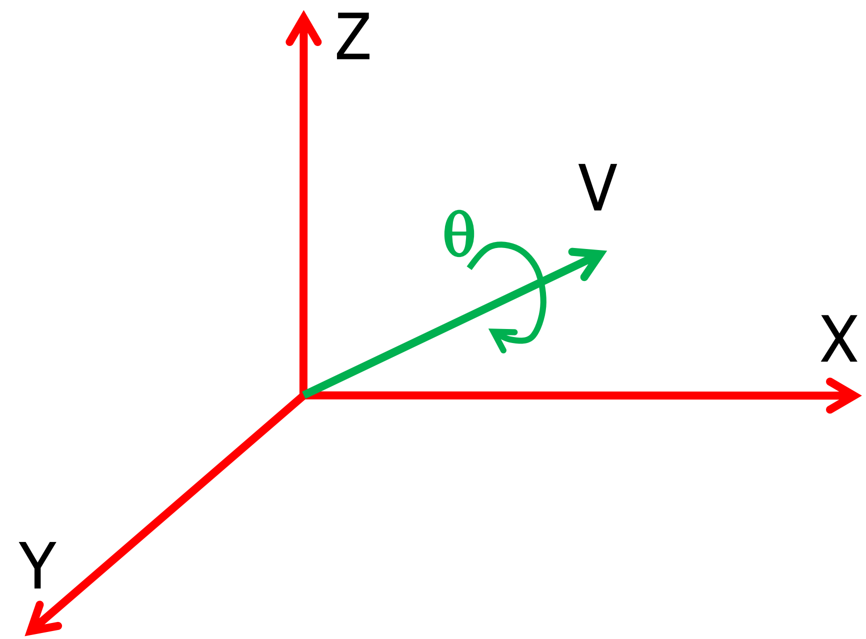






Axis-Angle Representation

- Represent an arbitrary rotation as a combination of a vector and an angle



Quaternions

- Are similar to axis-angle representation
- Two formulations
  - Classical
  - Based on JPL's standards

    W. G. Breckenridge, "Quaternions - Proposed Standard Conventions," JPL, Tech. Rep. INTEROFFICE MEMORANDUM IOM 343-79-1199, 1999.
- Avoids Gimbal lock

- See also: M. D. Shuster, "A survey of attitude representations," Journal of the Astronautical Sciences, vol. 41, no. 4, pp. 439–517, Oct.–Dec. 1993.

Quaternions

Classic notation	JPL-based
$\bar{q} = q_4 + q_1 i + q_2 j + q_3 k$	$\bar{q} = q_4 + q_1 i + q_2 j + q_3 k$
$i^2 = j^2 = k^2 = ijk = -1$ $ij = -ji = k,\ jk = -kj = i,\ ki = -ik = j$	$i^2 = j^2 = k^2 = -1$ $-ij = ji = k, -jk = kj = i, -ki = ik = j$
$\bar{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, q_0 = \cos\left(\theta/2\right), \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \sin\left(\theta/2\right)\cos\left(\beta_x\right) \\ \sin\left(\theta/2\right)\cos\left(\beta_y\right) \\ \sin\left(\theta/2\right)\cos\left(\beta_z\right) \end{bmatrix}$	$\bar{q} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix}, \mathbf{q} = \begin{bmatrix} k_x \sin\left(\theta/2\right) \\ k_y \sin\left(\theta/2\right) \\ k_z \sin\left(\theta/2\right) \end{bmatrix}, q_4 = \cos\left(\theta/2\right)$
	$\lVert \bar{q} \rVert = 1, \bar{q} \otimes \bar{p}, \mathbf{q} \times \mathbf{p}, \bar{q}_I, \lfloor \mathbf{q} \times \rfloor$

Vector Notation

See also: N. Trawny and S. I. Roumeliotis, “Indirect Kalman Filter for 3D Attitude Estimation,” University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep. 2005-002, March 2005.

Coordinate frames on PR2