# LECTURE 8-1

Localization and Map Making

*Introduction to AI Robotics (Sec. 11.5 – 11.8)*

# Quote of the Week

*"Genius is 1 percent inspiration and 99 percent perspiration. As a result, genius is often a talented person who has simply done all of his homework."*

Thomas Edison

# ANNOUNCEMENTS

- Quiz 15 on Sec. 11.1 – 11.4, Lec. 8-1 on *Thursday, 5/6/10*
- Lab 8 memo and code is due on Angel by midnight on *Thursday, 5/6/10*

# OBJECTIVES

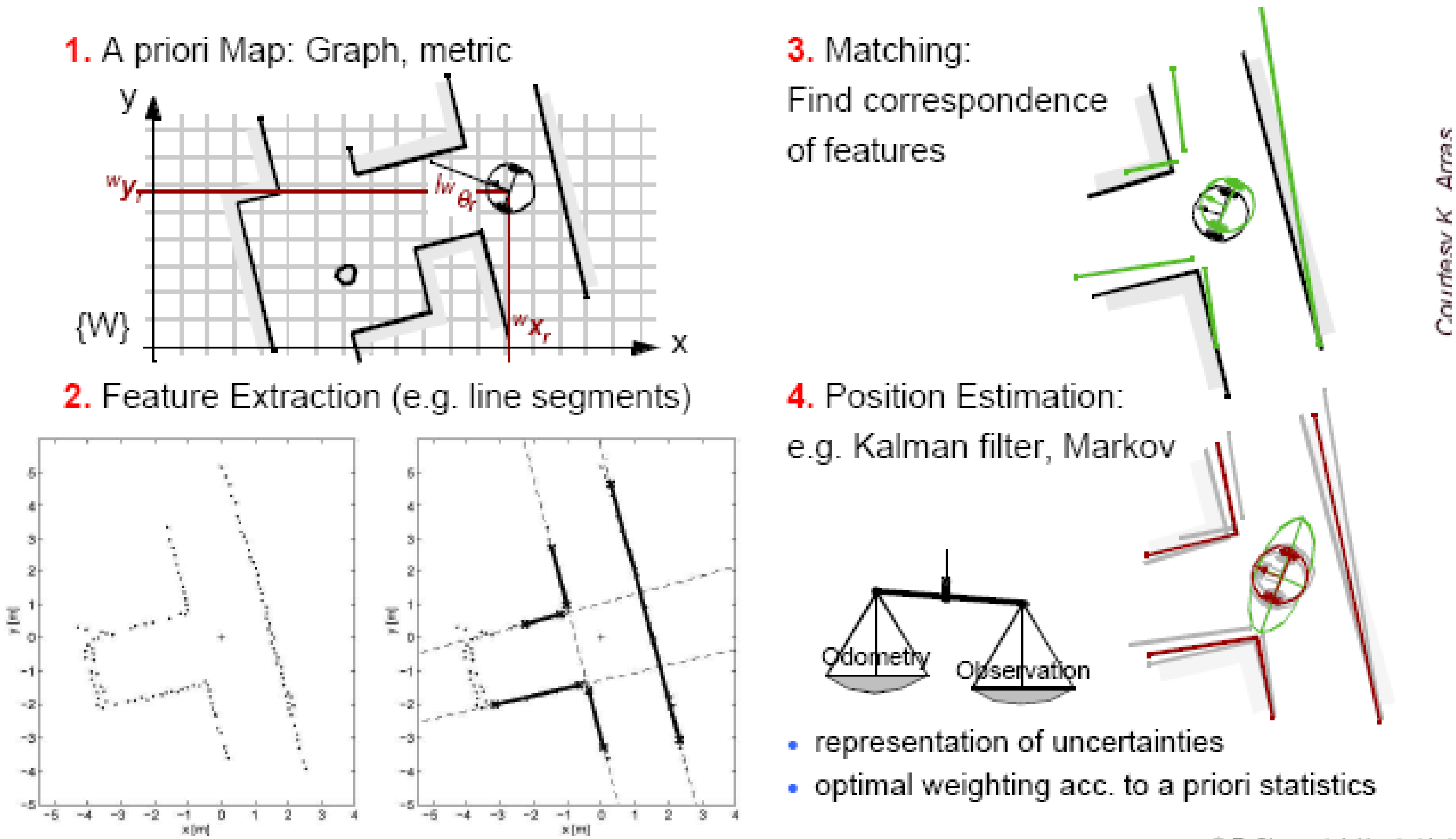
Upon completion of this lecture the student should be able to:

- Describe the difference between iconic and feature-based localization
- Be able to update an occupancy grid using either Bayesian, Dempster-Shafer or HIMM methods
- Describe the two types of formal exploration strategies

# HISTOGRAMMIC IN MOTION MAPPING (HIMM)

- HIMM was designed to improve obstacle avoidance and is a quasi-evidential technique where it scores certainty based upon sonar data
- Only elements along the acoustic axis of the sonar are updated
- The uncertainty score for cell is represented by a byte
- The scattered sampling means that a particular element may get only one or two updates
- Growth rate operators (GRO) can be used with a mask to increase the evidence of occupancy

# METHODS FOR LOCALIZATION: QUANTITATIVE METRIC APPROACH

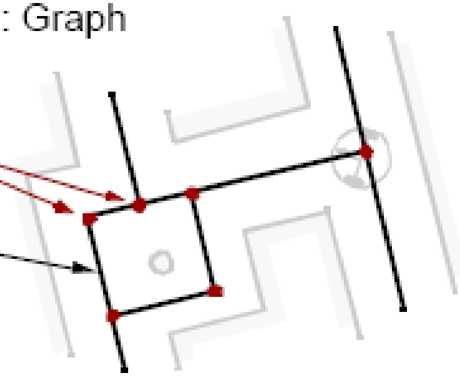


# TOPOLOGICAL LOCALIZATION QUANTITATIVE APPROACH

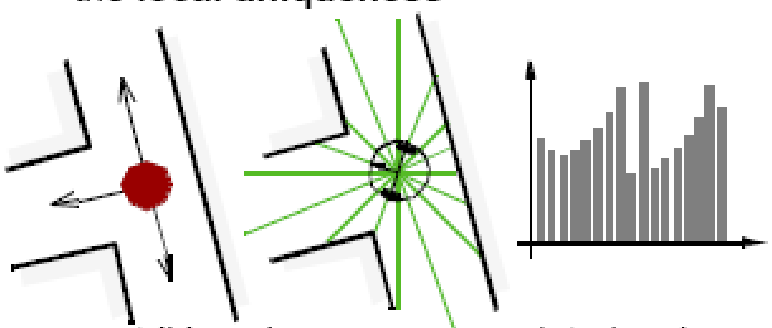
**1.** A priori Map: Graph

locally **unique** points

edges



**2.** Method for determining the **local uniqueness**

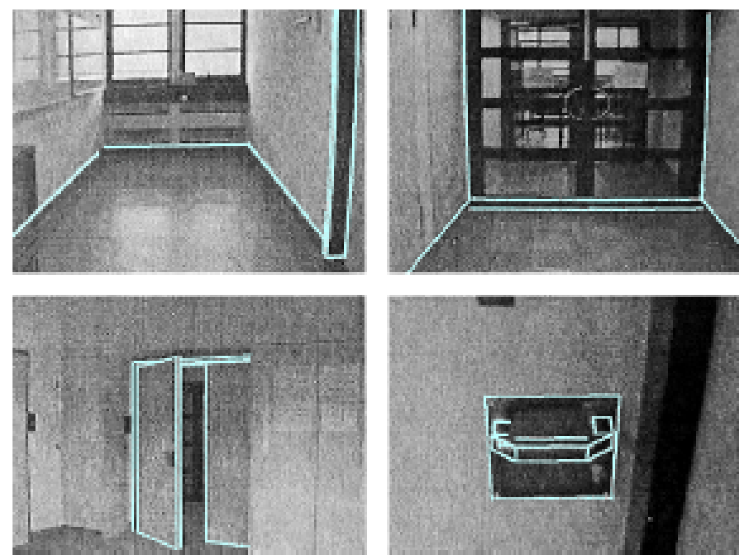


e.g. striking changes on raw data level or highly distinctive features

**3.** Library of **driving behaviors**

e.g. wall or midline following, blind step, enter door, application specific behaviors

Example: Video-based navigation with natural landmarks



Courtesy of [Lanser et al. 1996]

# PROBABILISTIC MAP-BASED LOCALIZATION

- One geometric approach to ***multi-hypothesis representation*** identifies the possible positions of a robot
- ***Probabilistic techniques*** identifies probabilities with the possible robot positions
- Two classes of probabilistic localization are:
  - ***Markov localization***
  - ***Kalman filter localization***

# PROBABILISTIC LOCALIZATION CLASSES

- ***Markov localization***
  - Uses an explicitly defined probability distribution across all robot positions
- ***Kalman filter localization***
  - Uses a Gaussian probability density representation of robot position and scan matching for localization
  - Unlike Markov, it does not independently consider each possible robot pose
  - Kalman results from the Markov axioms if the robot's position uncertainty is assumed to be Gaussian

# UPDATING ROBOT POSITION

- The process of updating robot position based upon ***proprioceptive*** and ***exteroceptive*** sensor values are separated logically into a general two-step process
  - ***Action Update***
    - Proprioceptive
    - Represents the application of some action model
  - ***Perception Update***
    - Exteroceptive
    - Represents the application of some perception model

# MARKOV AND KALMAN FILTER LOCALIZATION

- ***Markov localization*** is the robot's belief state usually represented as separate probability assignment for every possible pose on the map
  - Special case of probabilistic state estimation applied to mobile robot localization
- ***Kalman filter localization*** represents the robot's belief state using a single, well-defined Gaussian probability density function
  - It retains a $\mu$ and $\sigma$ parameterization of the robot's belief about position with respect to the map

# MARKOV VS. KALMAN FILTER LOCALIZATION

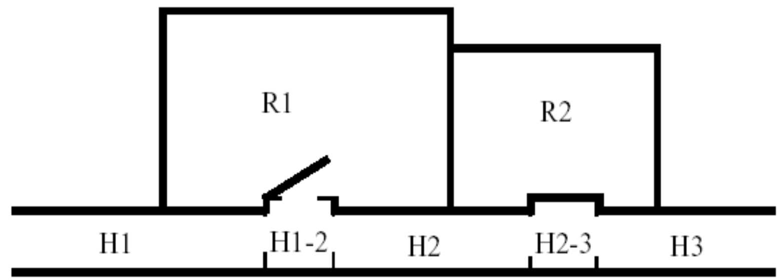
- **Markov**
  - Allows localization starting from any unknown position
  - Recovers from ambiguous situations because the robot can track multiple, complete disparate possible positions
  - Requires discrete representation of the space (geometric grid or topological graph)
  - Required memory and computational power can limit precision and map size

- ***Kalman Filter***
  - Tracks the robot from a known position
  - Is both precise and efficient
  - Can be used in continuous world representations
  - If robot uncertainty becomes too large and not unimodal, it can fail to capture the multitude of possible robot positions and can become irrevocably lost

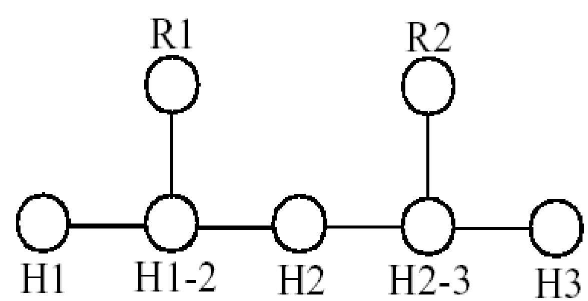
# MARKOV LOCALIZATION TOPOLOGICAL MAP

- Identical in abstraction and information to the environment map
- Decision involves assignment of nodes and connectivity between nodes
- Node boundaries are marked by doorways, hallways, and foyers
- Note there is no geometric information on the nodes

Environment map

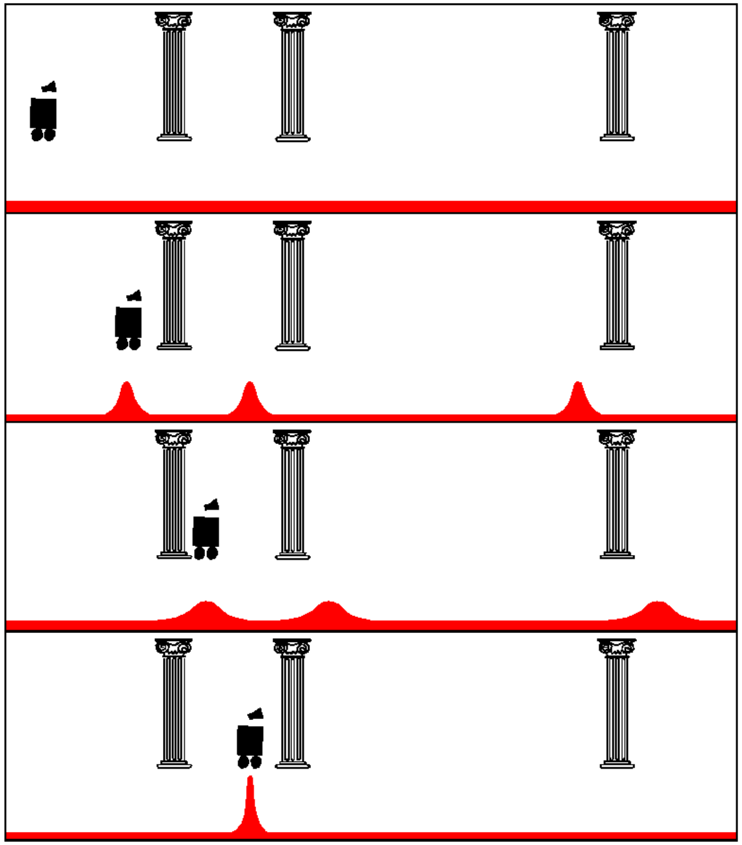



Topologic map



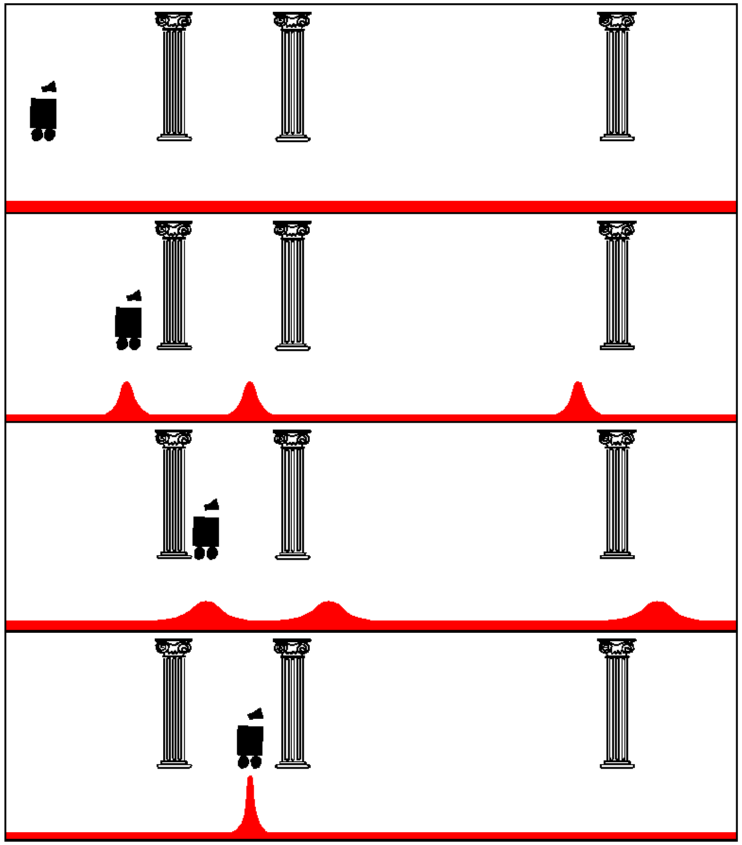
# MARKOV LOCALIZATION GRID MAP

1. Start
   - No knowledge at start, thus there is a uniform probability distribution
2. Robot perceives first pillar
   - Seeing only one pillar, the probability being at pillar 1, 2 or 3 is equal.
3. Robot moves
   - Action model enables the estimate of the new probability distribution based
     on the previous one and the motion.
4. Robot perceives second pillar
   - Based on all prior knowledge the probability being at pillar 2 becomes dominant

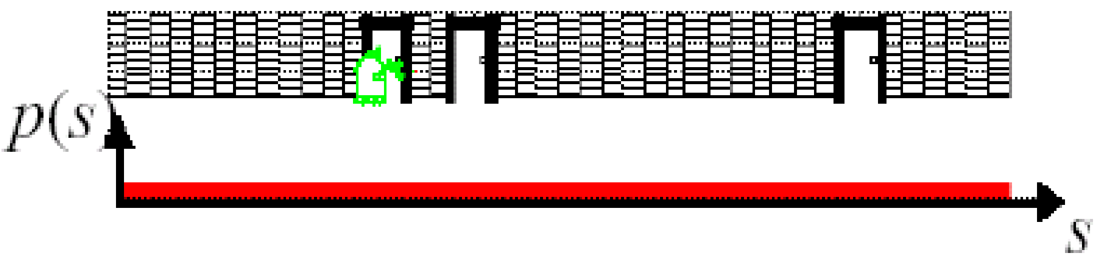


# MARKOV LOCALIZATION GRID MAP

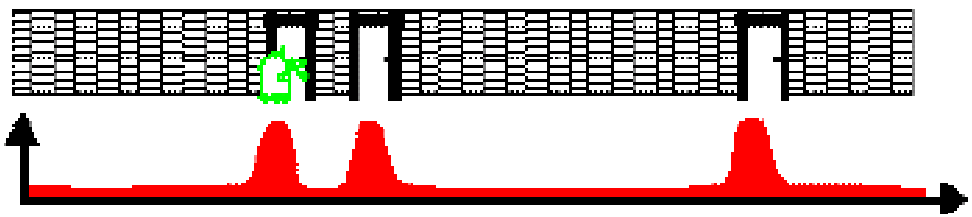
- As the robot encounters one pillar and then a second pillar, the probability density function over possible positions becomes multimodal, unimodal and then sharply defined
- The ability of a Markov localization system to ***localize the robot from an initially lost belief state*** is its key distinguishing feature
- This is a challenging application because of the dynamic nature of the environment



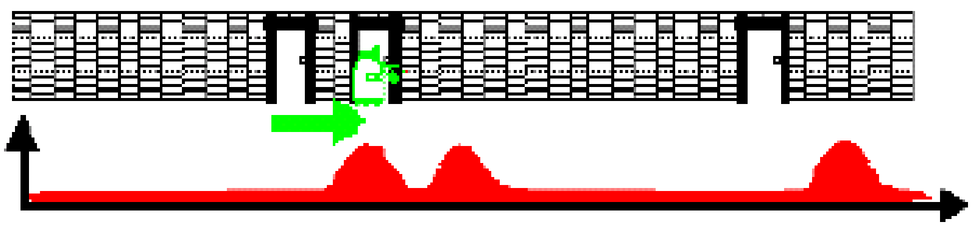
# MARKOV LOCALIZATION EXAMPLES



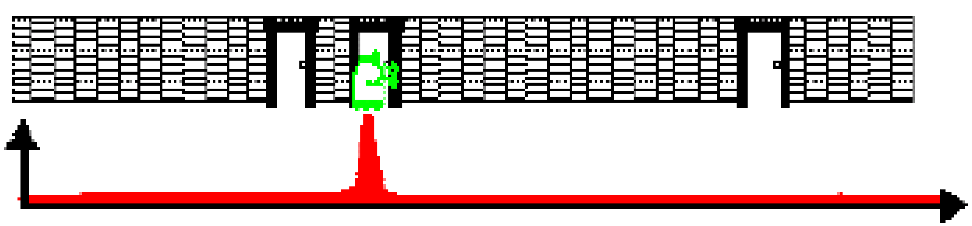

The robot is placed somewhere in the environment but it is not told its location



The robot queries its sensors and finds out it is next to a door



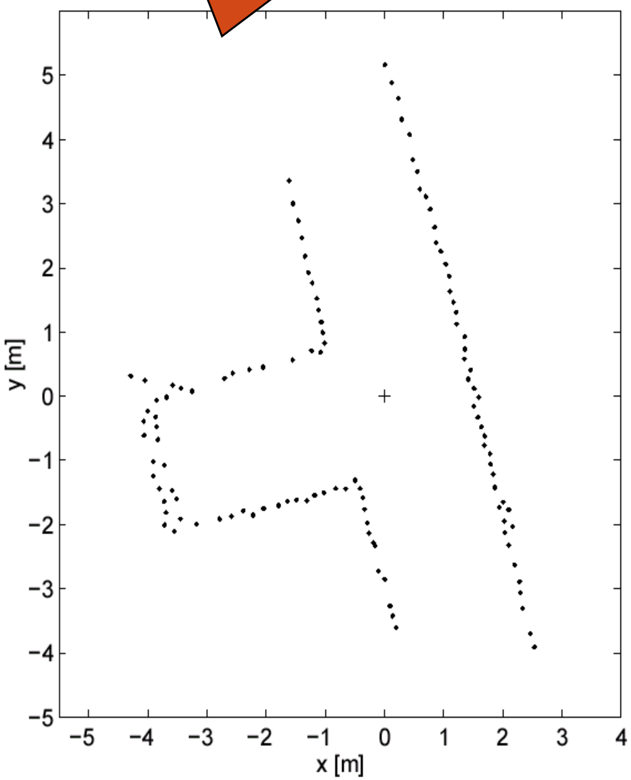
The robot moves one meter forward. To account for inherent noise in robot motion the new belief is smoother

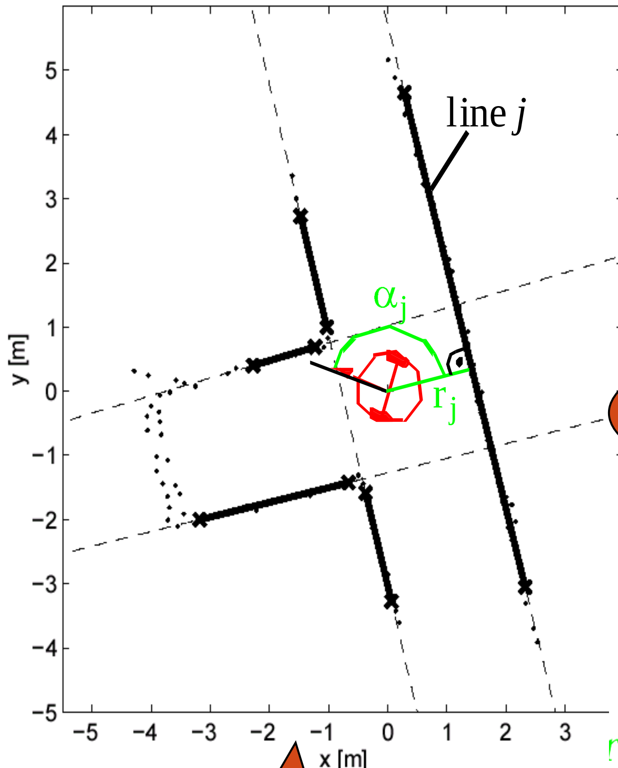


The robot queries its sensors and again it finds itself next to a door

# KALMAN FILTER LOCALIZATION

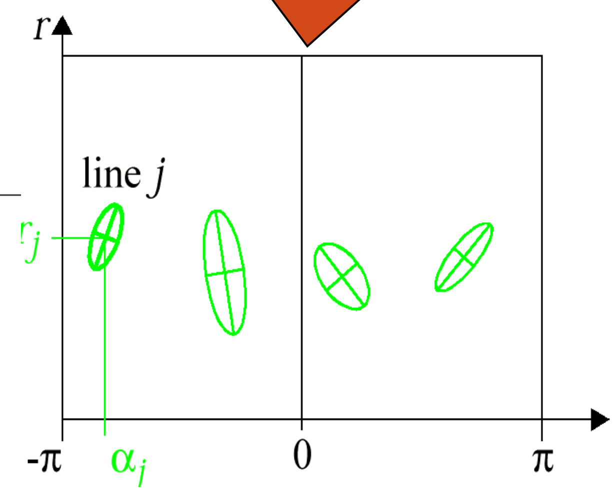
raw data from the laser scanner at time k+1, extracted lines



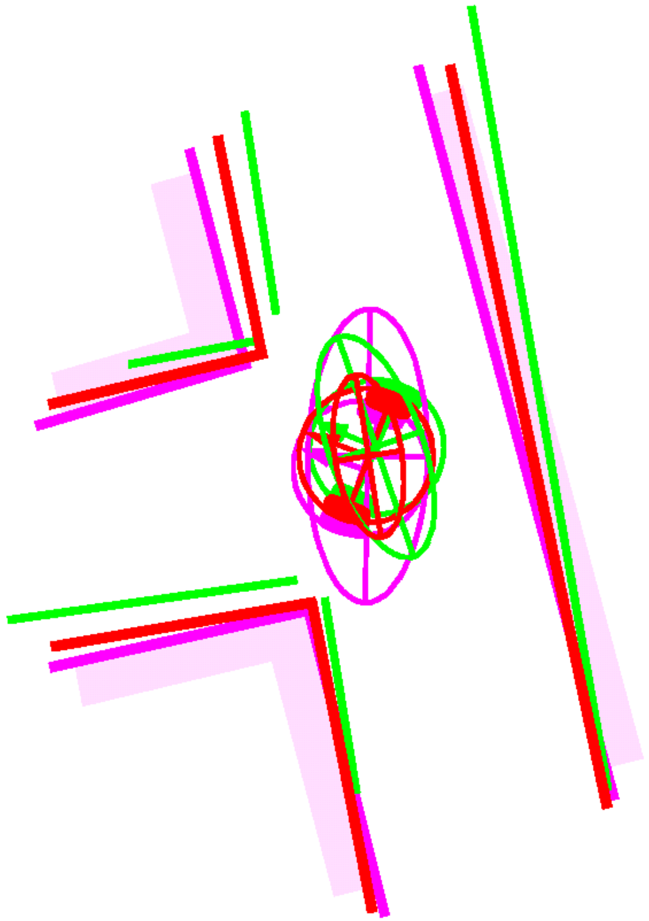

lines extracted from the raw data

the extracted lines uncertainties represented in the model space



# KALMAN FILTER LOCALIZATION CASE STUDY

- Kalman filter estimation of the new robot position
  - By fusing the prediction of robot position (magenta) with the information gained by the measurements (green)
  - we get the updated estimate of the robot position (red)
- this final pose estimate corresponds to the weighted sum of the
  - pose estimates of each matching pairing of observed and predicted features
  - robot position estimation based on odometry and observation positions



# MAP BUILDING

*Techniques:*

- Manual
  - Drawn by hand
  - Static/predictable environment
  - Costly
- Automatically
  - Robot learns environment
  - Dynamically/unpredictable changing
  - Different look due to different perception

*Requirements:*

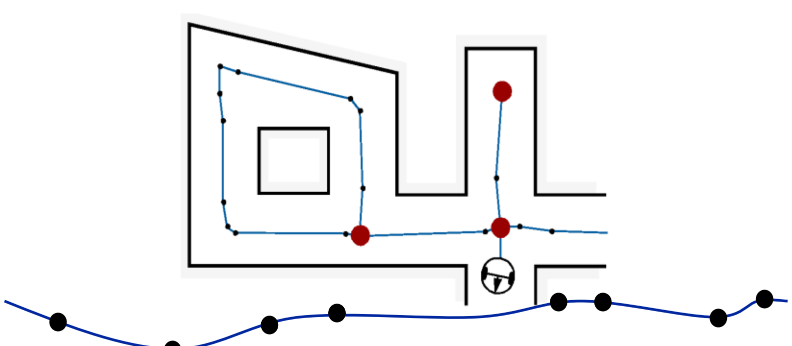
- Incorporates newly sensed information into the existing world model
- Contains information to estimate the robot's position
- Provides Information to do path planning and navigation tasks

# MAP BUILDING: MEASURE OF QUALITY

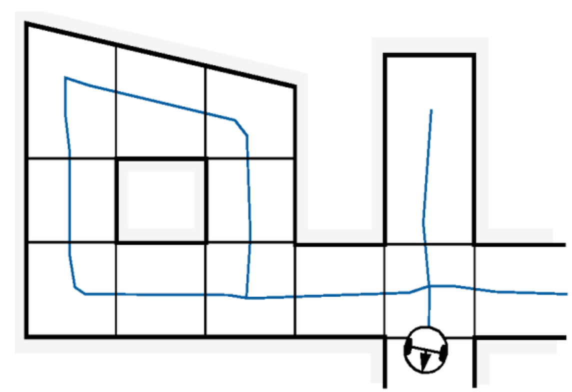
- Most environments are a mixture of *predictable* and *unpredictable* features (hybrid approach)
- The measure of quality is based upon
  - Topological correctness
  - Metric correctness

# ROAD MAP, GRAPH CONSTRUCTION CELL DECOMPOSITION

- *Road Map, Graph construction*
  - Identify a set of routes within the free space
  - Where to put the nodes?
  - Topology-based:
    - at distinctive locations
  - Metric-based:
    - where features disappear or get visible

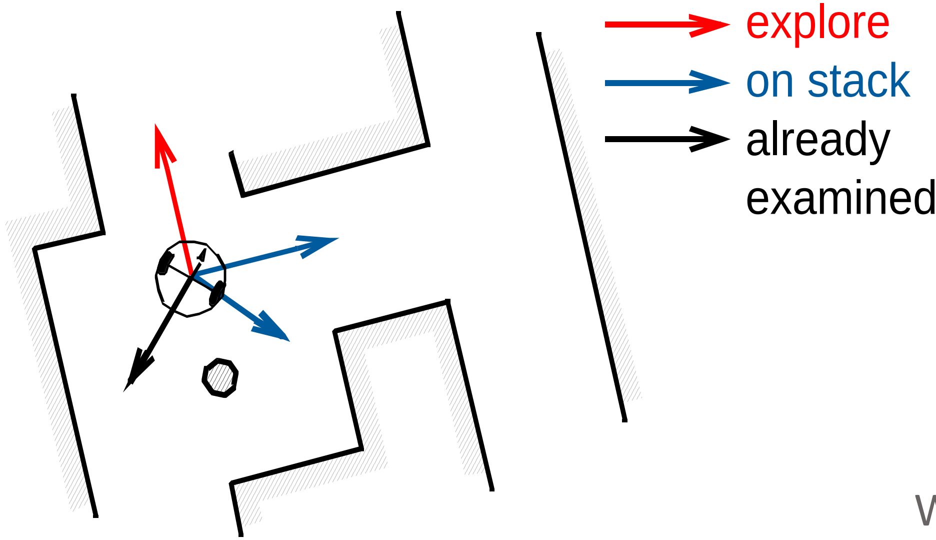


- *Cell decomposition*
  - Discriminate between free and occupied cells
  - Where to put the cell boundaries?
  - Topology- and metric-based:
    - where features disappear or get visible



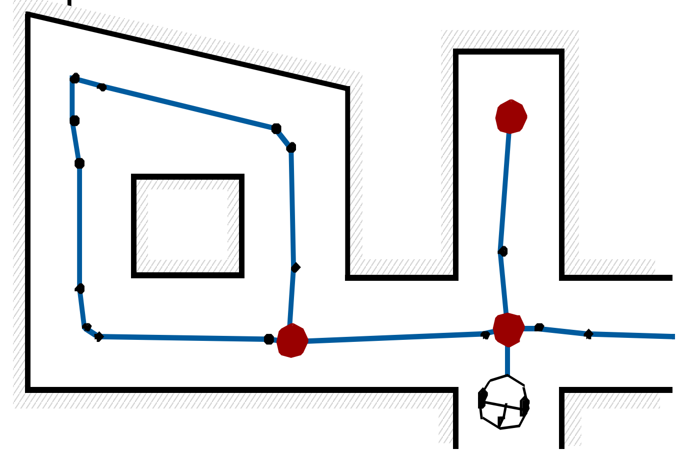
# MAP BUILDING: EXPLORATION AND GRAPH CONSTRUCTION

1. Exploration



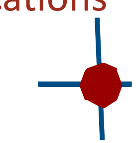

- provides correct topology
- must recognize already visited location
- backtracking for unexplored openings

2. Graph Construction



Where to put the nodes?

- Topology-based: at distinctive locations



- Metric-based: where features disappear or get visible

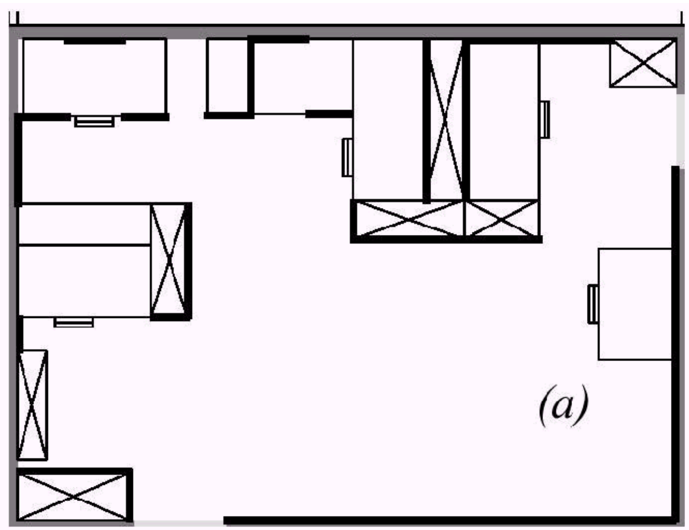
# CONTINUOUS REPRESENTATION

- A continuous-valued map is one method for exact ***decomposition*** of the environment
- Continuous maps are only in 2D representations as further dimensionality can result in computational explosion
- Combine the exactness of continuous representation with the compactness of ***closed-world assumption***
- The representation will specify all environmental objects in the map

# CONTINUOUS REPRESENTATION

- a low-memory map is a 2D representation in which polygons represent all obstacles
- many simulations run exclusively in the computer memory and polygons are not used to describe a real-world environment
- When real environments must be captured, there are trends for *selectivity* and *abstraction*
  - *The human captures only objects that can be detected by the robot's sensors*
  - *This represents a subset of the features of the real world objects*

# CONTINUOUS REPRESENTATION




Architecture map

Infinite line representation