ME490

Programming for Autonomous System

: Simultaneous Localization and Mapping (SLAM)

Fall, 2018

Jinwhan Kim

KAIST

Introduction

Simultaneous Localization & Mapping (SLAM)

a.k.a.

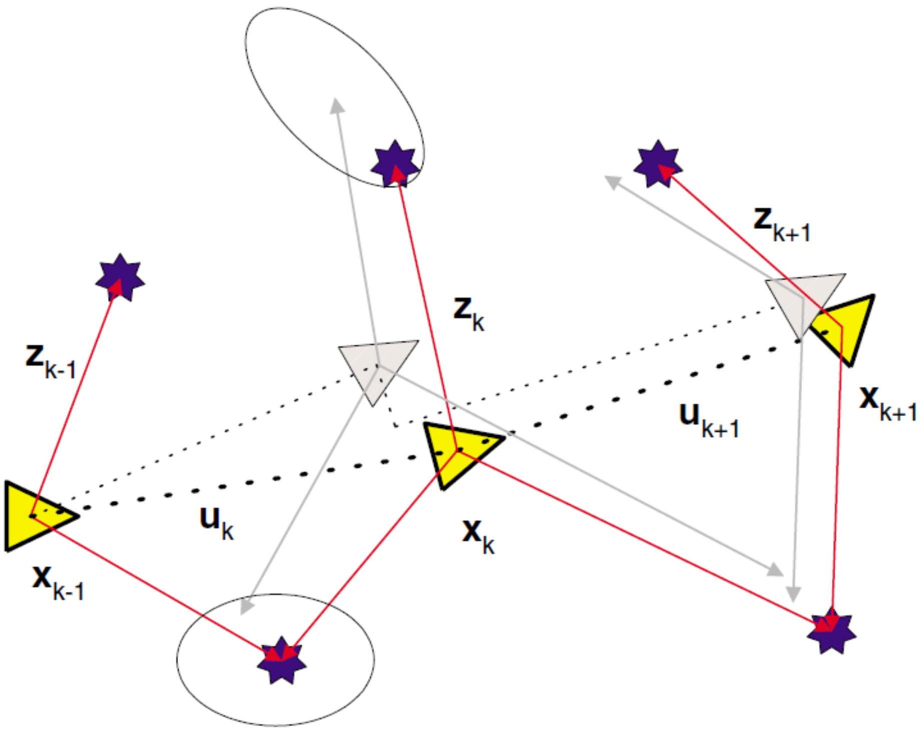
Concurrent Mapping & Localization (CML)

Localization, Mapping and SLAM

- **Localization** is the process of finding one's position in a known environment.
  - "Give me a map, then I can figure out where I am."
- **Map building (or Mapping)** is the process of creating a spatial representation of features with respect to an observer in an unknown environment.
  - "Tell me where I am, then I can build a map around me."
- **S**imultaneous **L**ocalization **A**nd **M**apping (**SLAM**) is the process of acquiring a spatial model (map) of its environment while simultaneously localizing itself relative to this map.
  - "Let me do both at the same time."

Localization Problem

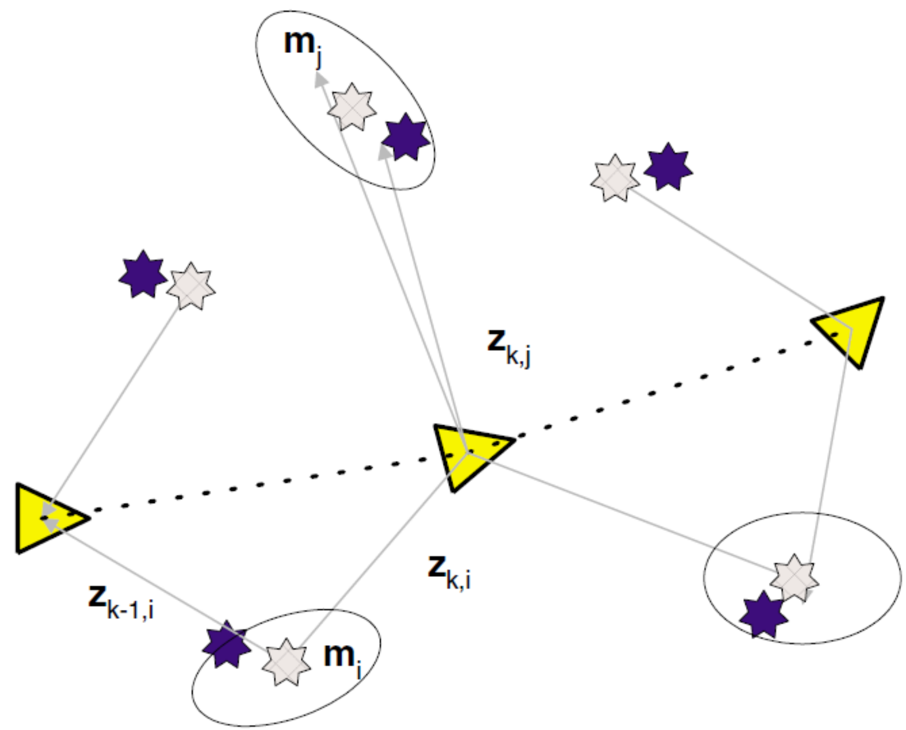
- Given:
  - A map (geometric or landmark-based)
  - Observations of nearby features on the map
  - Control actions of a robot (or vehicle)
- Estimate
  - Robot's trajectory (or pose)



from the lecture slides for SLAM Summer School 2002 by H. Durrant-Whyte

Mapping Problem

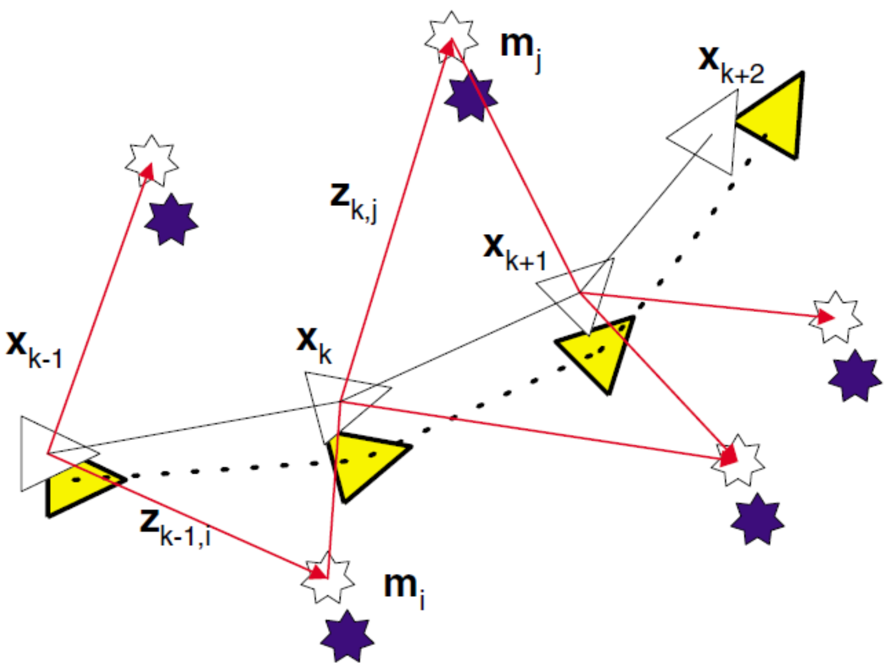
- Given:
  - Robot's locations
  - Observations of nearby features
  - Control actions of a robot (or vehicle)
- Estimate
  - The map of the environment associated with the observed features.



from the lecture slides for SLAM Summer School 2002 by H. Durrant-Whyte

SLAM Problem

- Given:
  - Control actions of a robot (or vehicle)
  - Observations of nearby features
- Estimate
  - Robot's trajectory
  - The map of the environment associated with the observed features.

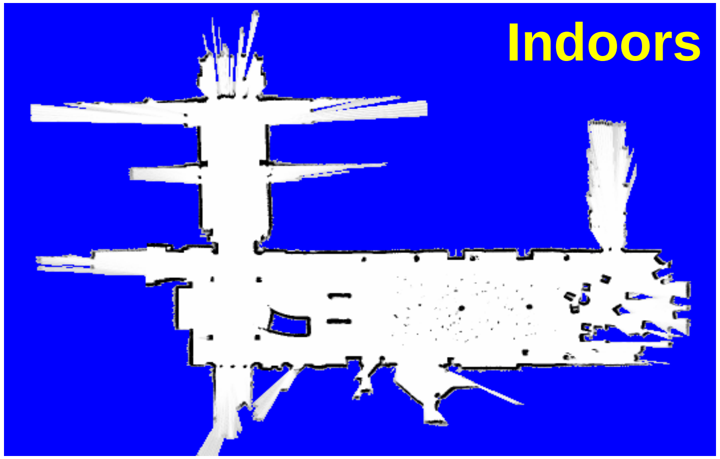


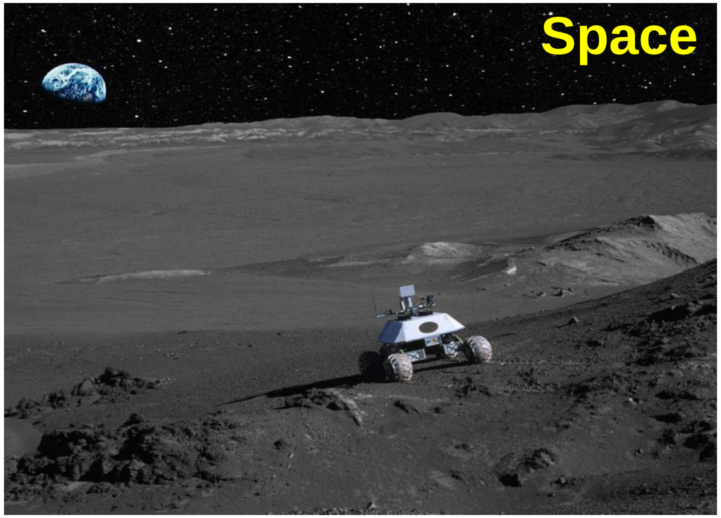
from the lecture slides for SLAM Summer School 2002 by H. Durrant-Whyte

Why are we interested in SLAM?

- **SLAM** enables a robot to start in an unknown location in an unknown environment, and then incrementally build a map of the environment and find its own position relative to the map simultaneously.
- We humans already have this SLAM capability, which allows for roaming around a new place without a map (often) without getting lost.
- SLAM is a crucial capability for robots (or unmanned vehicles) to be operated truly autonomously without any external intervention by humans.

SLAM Applications





from the lecture slides for "probabilistic robotics" by S. Thrun

Navigation (Localization)

Navigation

The word navigation is derived from the Latin roots **navis** meaning "ship" and **agere** meaning "to move" or "to direct."

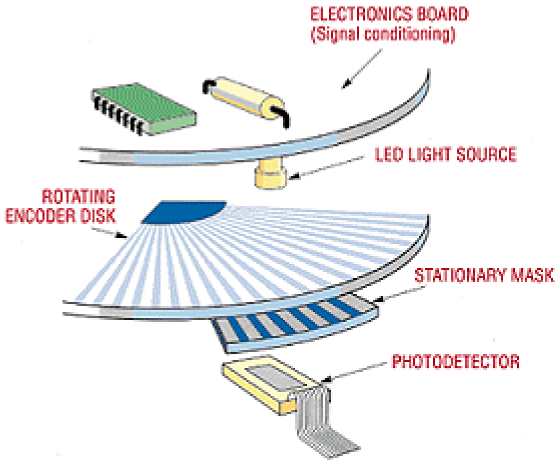
- In a broad sense, it is the art of getting from one place to another including planning and execution of the maneuvers necessary to move between desired locations.
- In a more technical sense, navigation is the process of determining one's motion in space (e.g., position) at a given time.
- Cf. Guidance and control

Navigation Sensors

- Encoders
  - Measure the rotational motion of a wheel shaft
- Directional rangefinders
  - Measure distance from the observer to target objects (e.g., radar, sonar, lidar)
- Compasses and inclinometers
- Inertial measurement systems
  - Gyros measure angular rate
  - Accelerometers measure linear accelerations
- Positioning systems
  - GPS, LORAN, Decca, VOR/DME, etc.
- Cameras

(Rotary) Encoder

- A device that converts rotational motion into a sequence of digital pulses.
  - Signals are integrated to obtain the position of a robot
- Encoder types
  - Absolute encoders measure the absolute angular position of a wheel shaft
  - Incremental encoders measure the change in angular position of a wheel shaft.



http://www.kavlico.com/index_home.html

Directional Rangefinders

- A device to measure the distance between the observer and nearby objects using sound or electromagnetic waves. (e.g., radar, lidar, sonar)

Speed of wave propagation

$$d = c(t - t_d)/2$$

Time of flight   Latency

  - Speed of sound: c ≈ 0.3 m/ms
  - Speed of light: c ≈ 0.3 m/ns

Compass and Inclinometer

- **A compass** is a device to show the its direction (orientation) with respect to the Earth-fixed reference frame.
- **An inclinometer** is a device to measure the slope of an object with respect to the Earth's gravity.

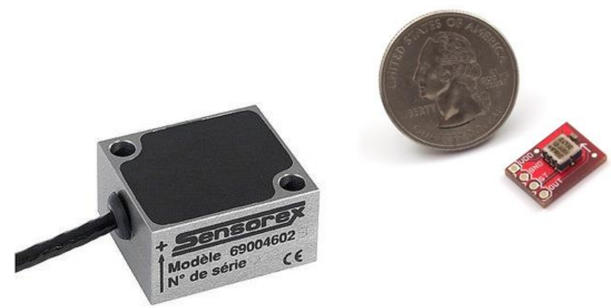
http://www.geonavmarine.com

http://www.honeywell.com

Accelerometer

- A device that measures specific force (force divided by mass, a.k.a. proper acceleration) which is the acceleration with respect to free-fall.

$$\text{Specific Force} = \frac{F_{\text{experienced}} - F_{\text{gravity}}}{\text{Mass}}$$

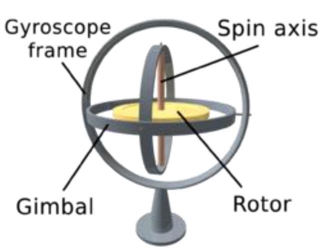


- Accelerometers are used to measure the acceleration of a vehicle and also to measure its inclination with respect to the gravity axis.

Gyroscopes (Gyros)

- Mechanical or optical systems that can be used where the magnetic field is not available
- Gyros are for measuring rotational motions (e.g., angular rate, attitude/orientation)
- Basically, angle is obtained by integrating angular rate, which necessarily leads to drift errors.

Mechanical gyro



http://en.wikipedia.org/wiki/Gyroscope

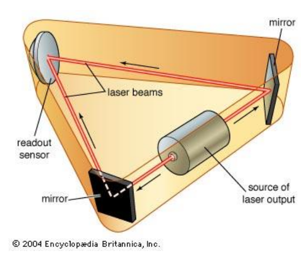
MEMS gyro

http://www.invensense.com/

Fiber optic gyro (FOG)

http://www.alcielo.com/

Ring-laser gyro



Encyclopædia Britannica

Integrated Inertia Sensor Systems

- **INS**: Inertial navigation system
  - Sensor-navigation computer integrated system
- **IMU**: Inertial measurement unit
  - 3 gyros and 3 accelerometers
- **AHRS**: Attitude and heading, reference system
  - 3 gyros often with magnetometer

http://www.raytheon-anschuetz.com/

http://www.microstrain.com/

Global Positioning Systems

- NAVSATR GPS
  - The first satellite was launched in 1978.
  - Currently, constellation of 27 satellites in medium earth orbit
  - Selective Availability was turned off in 2000
- Other systems
  - GLONASS (Russia), Galileo (EU), BeiDou (China)

http://tf.nist.gov/service/gpstrace.htm

Cameras

- Cameras had not been actively used for vehicle navigation, since necessary technologies were not available until recently.
- With the advance of computer vision technology, cameras are now widely used for vehicle navigation, mapping and many other applications.
- A camera image is two-dimensional and basically provides bearing-only information.
- Cameras for robot navigation
  - Monocular or stereo cameras
  - Optimal or infrared cameras
  - Event camera
  - RGB-D camera




Vehicle Navigation Approaches

- Dead-reckoning
  - Encoder (e.g., odometry)
  - Inertial navigation
- Navigational beacons
  - GPS
  - Baseline systems (e.g., acoustic LBL, SBL, USBL)
- Landmark navigation
- Map-based navigation

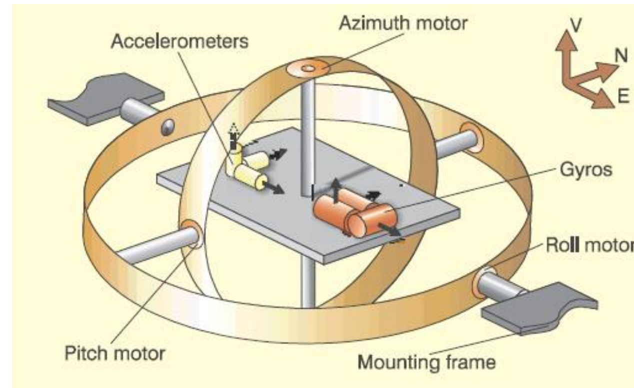
http://gcoinc.files.wordpress.com/2011/06/where_am_i.png

Inertial Navigation

- **Inertial navigation** is a self-contained, nonjammable **dead-reckoning** technique in which measurements provided by **accelerometers** and **gyroscopes** are used to track the position and orientation of an object relative to a known starting point, orientation and velocity.
- **No external references are required** in order to determine the object's position, orientation, or velocity once initialized.

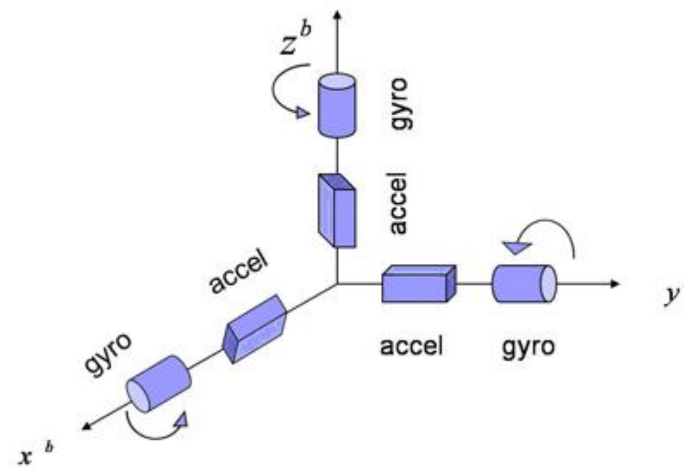
Gimbaled Inertial Platform

- The gimbals are a set of three rings, each with a pair of bearings initially at right angles.
- Measure a vehicle's roll, pitch, and yaw angles directly at the bearings of the gimbals.
- Simple integration of linear acceleration for position computation because the directions of the linear accelerometers do not change.



Strapdown Inertial Platform

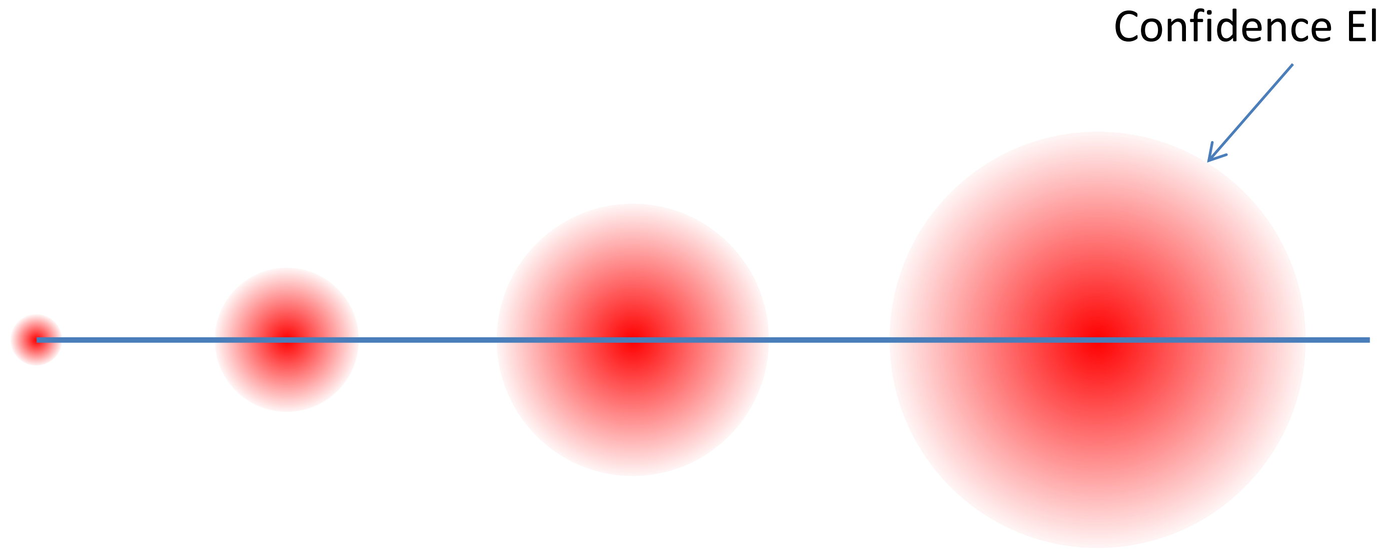
- The sensors (with no moving parts) are strapped to the vehicle.
- 3 accelerometers for measuring accelerations and 3 gyroscopes for angular velocities.
- Advantages:
  - Wider range, higher update rate, increased reliability, lower power consumption, reduced cost …
- Requires more complex computation
- More commonly used nowadays



Inertial Measurement Unit (IMU)

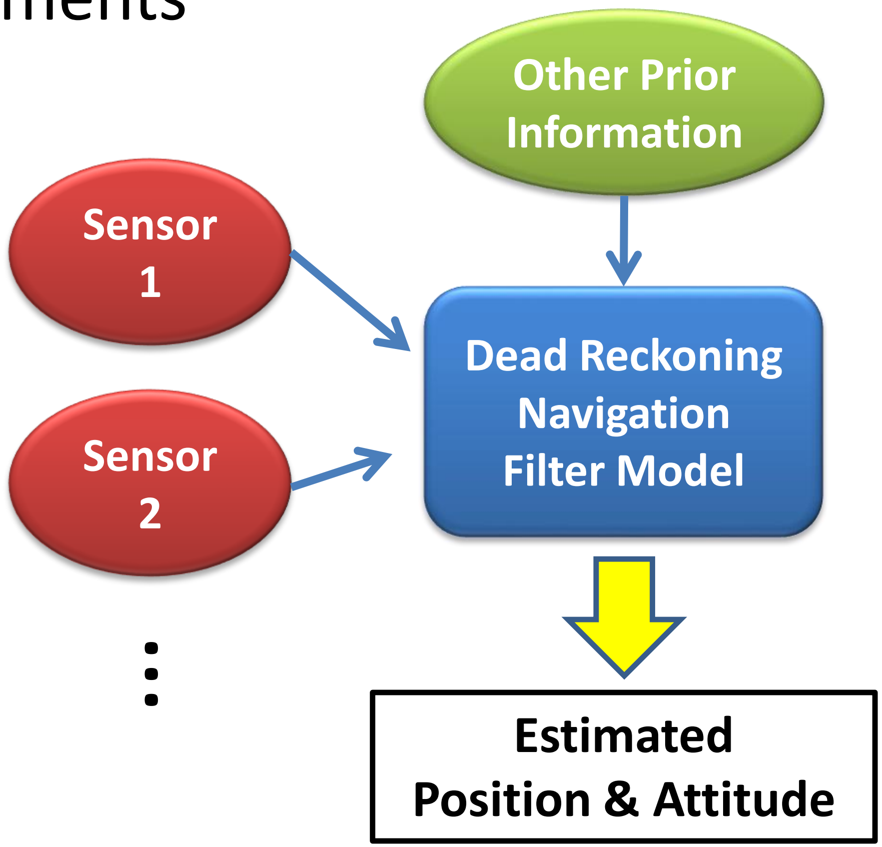
Dead-Reckoning Uncertainty

- The vehicle might start to move from a known location. However, position uncertainty during dead-reckoning grows in time without bounds due to sensor biases and measurement noise.



Sensor Fusion for Navigation

- Reference measurements
  - Position
    $\mathbf{z}_{\text{pos}} = \mathbf{r}^n_{b/n}$
  - Attitude
    $\mathbf{z}_{\text{ang}} = \mathbf{\Theta}^n_{b/n}$
  - Velocity
    $\mathbf{z}_{\text{vel}} = \mathbf{v}^n_{b/n}$
  - Additional info. if available



Bayes' Rule (Bayes' Theorem)

- By the conditional probability and the theorem of total probability,

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} = \frac{p(y \mid x)p(x)}{\sum_{x'} p(y \mid x')P(x')} \qquad \text{(Discrete)}$$

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} = \frac{p(y \mid x)p(x)}{\int p(y \mid x')p(x')\, dx'} \qquad \text{(Continuous)}$$

- Another representation

$$\underbrace{p(x \mid y)}_{\text{Posterior probability}} = \underbrace{\eta}_{\text{Normalizer}}\, \underbrace{p(y \mid x)}_{\text{Likelihood}}\, \underbrace{p(x)}_{\text{Prior probability}}$$

Bayesian Estimation

- Construct the probability density function(pdf) of the required state vector using all available information.

State vector to be estimated
(e.g., position, velocity, etc.)

Measurement information
(e.g., sensor measurments)

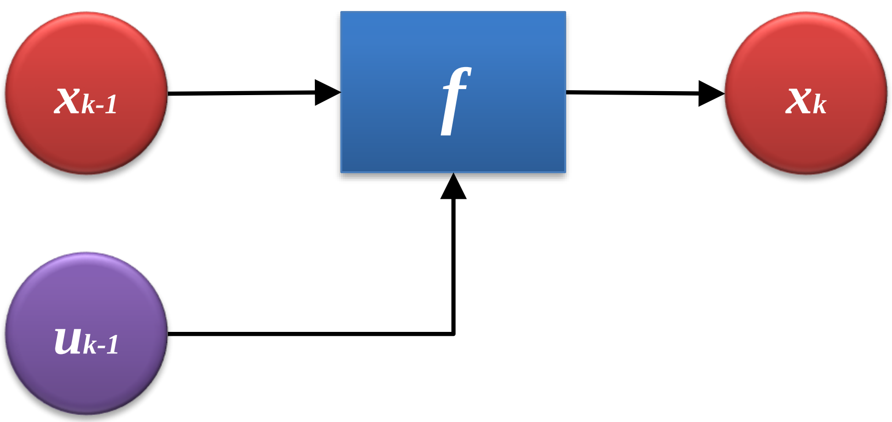
$$p(\mathbf{x} \mid \mathbf{z})$$

Statistical description of the state vector based on available information

- Available information
  – System dynamics: State transition probability
  – Sensor information: Measurement probability

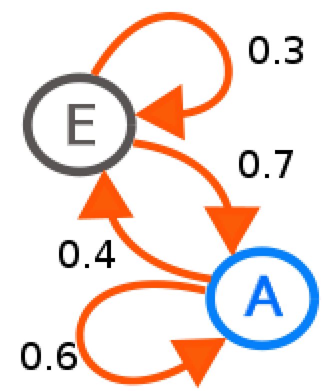
State Transition Probability

- The one-step transition probability is the probability of transitioning from one state to another in a single step.



$$p(\mathbf{x} \mid \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$$

cf., Markov Chain



Measurement Probability

- The information on the sensor characteristics is provided as the pdf in the following form:

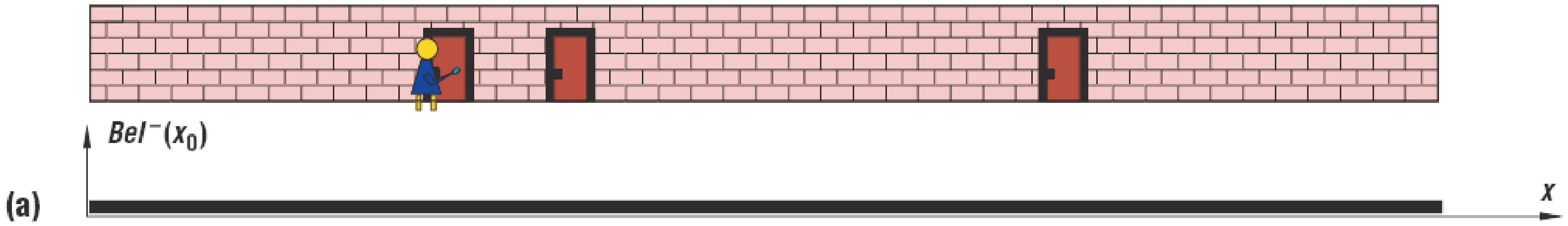
$$p(\mathbf{z} \mid \mathbf{x})$$

Markov Localization

- Markov localization is a probabilistic algorithm that maintains a probability distribution over the space of multiple hypotheses.
  - The probabilistic representation allows for assigning a different weight to each hypothesis.
  - Note: Kalman filtering maintains a single hypothesis as to where in the world a robot might be.

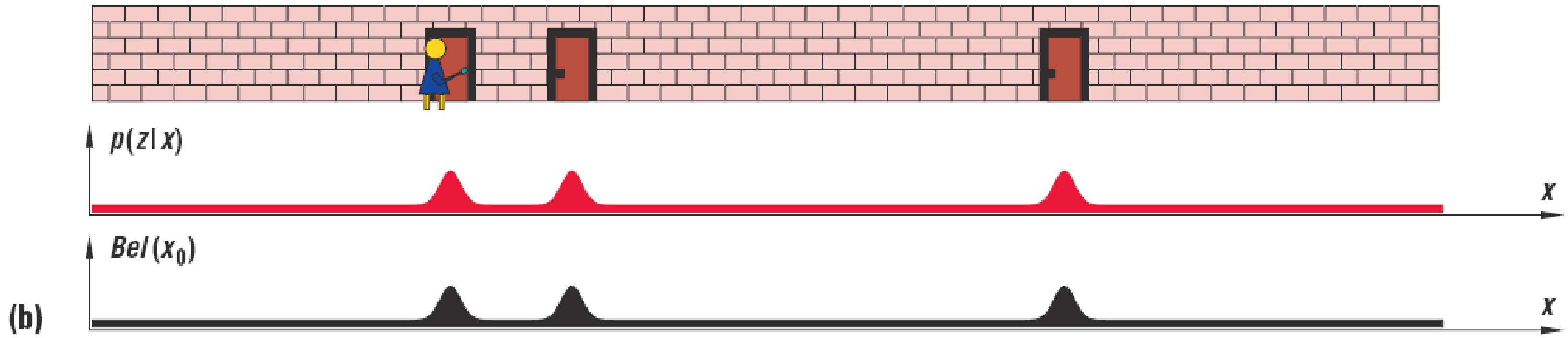
$$\underbrace{p(\mathbf{x}_k \mid \mathbf{Z}_k)}_{\text{Updated state estimate}} = \eta \, \underbrace{p(\mathbf{z}_k \mid \mathbf{x}_k)}_{\text{Measurement model}} \int \underbrace{p(\mathbf{x}_k \mid \mathbf{x}_{k-1})}_{\text{State transition probability}} \, \underbrace{p(\mathbf{x}_{k-1} \mid \mathbf{Z}_{k-1})}_{\text{Previous state estimate}} \, d\mathbf{x}_{k-1}$$

Illustration: Markov Localization



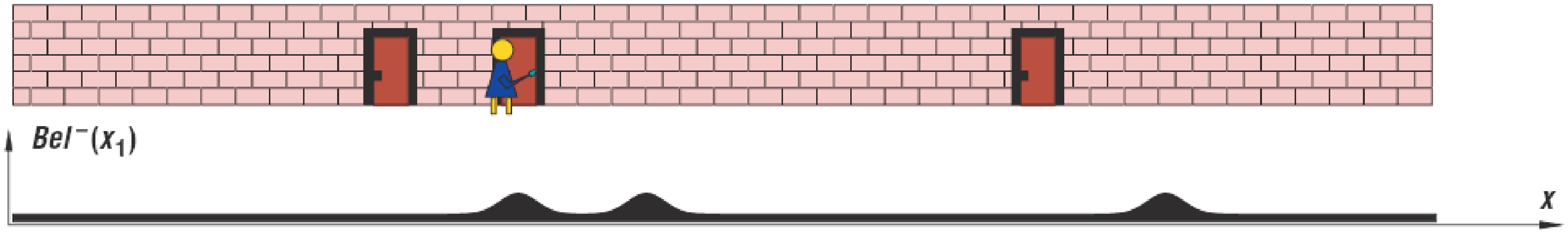
This illustration is from http://faculty.kfupm.edu.sa/COE/mayez/

Illustration: Markov Localization



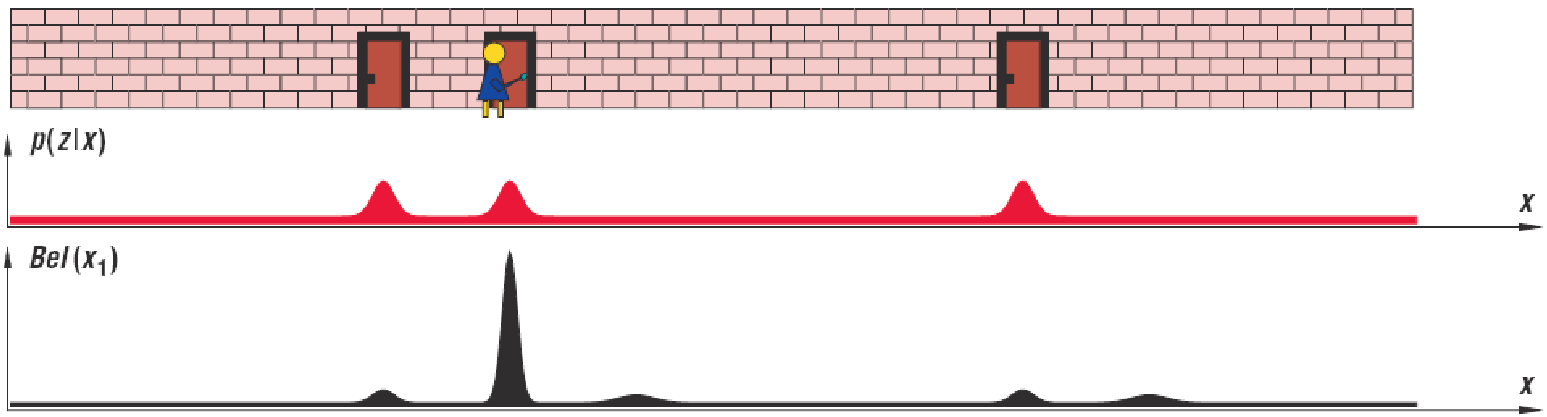
This illustration is from http://faculty.kfupm.edu.sa/COE/mayez/

Illustration: Markov Localization

(c) 

This illustration is from http://faculty.kfupm.edu.sa/COE/mayez/

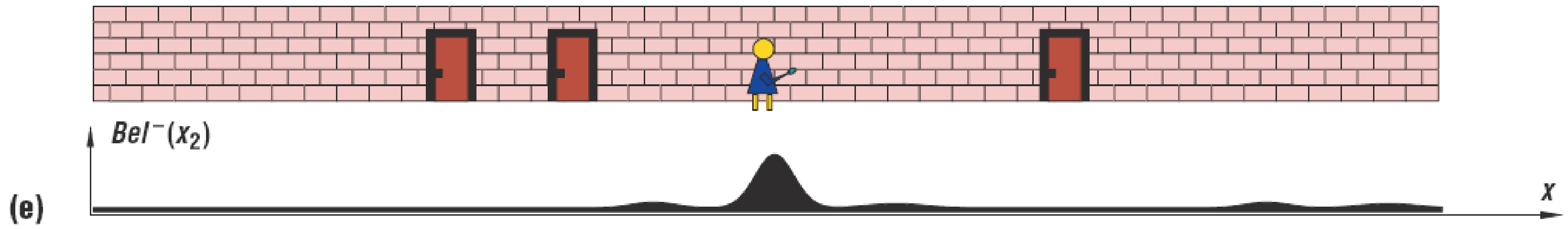
Illustration: Markov Localization



(d)

This illustration is from http://faculty.kfupm.edu.sa/COE/mayez/

Illustration: Markov Localization



This illustration is from http://faculty.kfupm.edu.sa/COE/mayez/

Markov Localization

- Implementation (discrete representation)

Prediction: Sum over all previous possible positions and actions

$$p(\mathbf{x}_k^i \mid \mathbf{Z}_{k-1}) = \sum_{j=1}^{n} p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^j)\, p(\mathbf{x}_{k-1}^j \mid \mathbf{Z}_{k-1})$$

Correction: Given perception $\mathbf{z}_k$, evaluate the probability of being in location $\mathbf{x}_k^i$

$$p(\mathbf{x}_k^i \mid \mathbf{Z}_k) = \eta\, p(\mathbf{z}_k \mid \mathbf{x}_k^i)\, p(\mathbf{x}_k^i \mid \mathbf{Z}_{k-1})$$

Algorithm Implementation

- Pseudocode

Algorithm Markov Localization$(p(\mathbf{x}_{k-1} \mid, \mathbf{z}_{k-1}), \mathbf{u}_{k-1}, \mathbf{z}_k, \mathbf{m})$

for $i = 1$ to N

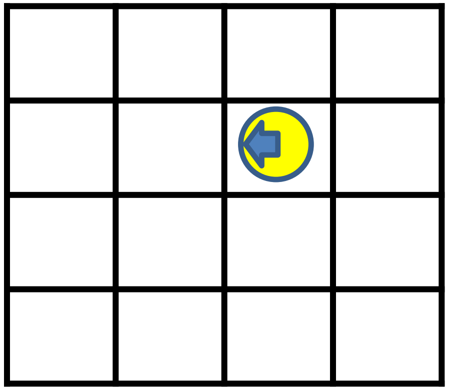
$$p(\mathbf{x}_k^i \mid \mathbf{Z}_{k-1}) = \sum_{j=1}^{N} p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^j, \mathbf{u}_{k-1}, \mathbf{m})\, p(\mathbf{x}_{k-1}^j \mid \mathbf{Z}_{k-1})$$

$$p(\mathbf{x}_k^i \mid \mathbf{Z}_k) = \eta\, p(\mathbf{z}_k \mid \mathbf{x}_k^i, \mathbf{m})\, p(\mathbf{x}_k^i \mid \mathbf{Z}_{k-1})$$

end

return

  - Example) Markov localization of a robot equipped with encoders and a compass moving in a grid world.



Mapping

Why Mapping?

- Learning maps is one of the fundamental problems in mobile robotics.
- Maps allow robots to efficiently carry out their tasks, allow localization, etc.
- Successful robot systems rely on maps for localization, path planning, etc.
- Mapping often involves to simultaneously estimate the pose of the vehicle and the map. (i.e., chicken and egg problem)

From the lecture notes of CS226B by Thrun & Teichman at Stanford

Robot Mapping

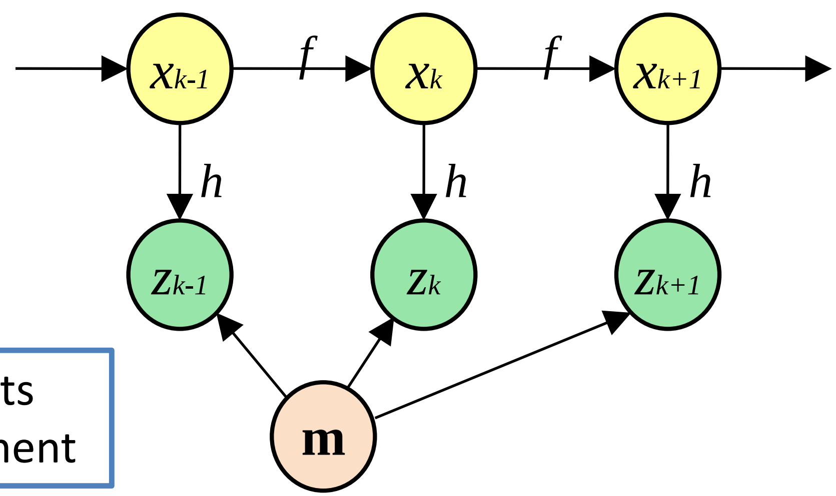
- Mapping with known poses
  - The robot's poses are assumed to be known (or provided) during mapping.
  - Posterior over maps given information

Map of the environment

$$p(\mathbf{m} \mid \mathbf{X}_{\mathrm{k}}, \mathbf{Z}_{\mathrm{k}})$$

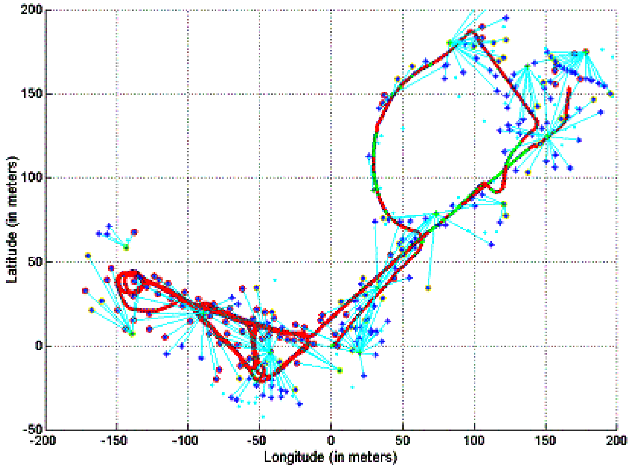
Robot Trajectory

Measurements of the environment



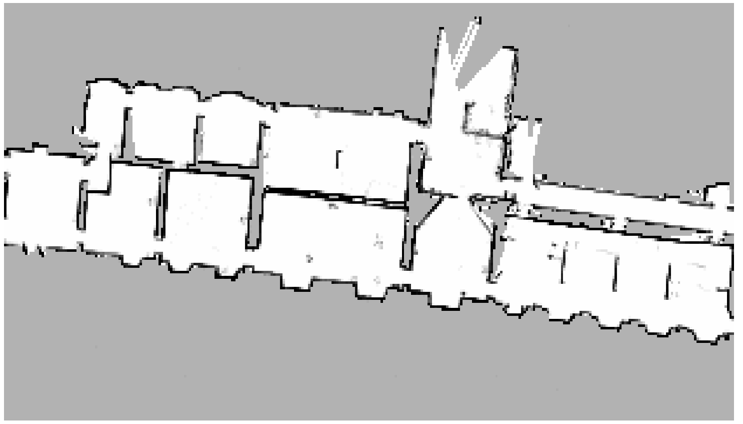
Map Representation

- **Landmark (feature)** based
  - A number of feature points associated each with its position coordinates are used.



http://www-personal.acfr.usyd.edu.au/nebot/publications/slam

- **Grid (location)** based
  - Each index is labeled with a specific grid location and its property is defined.
  - i.e., Occupancy grid maps

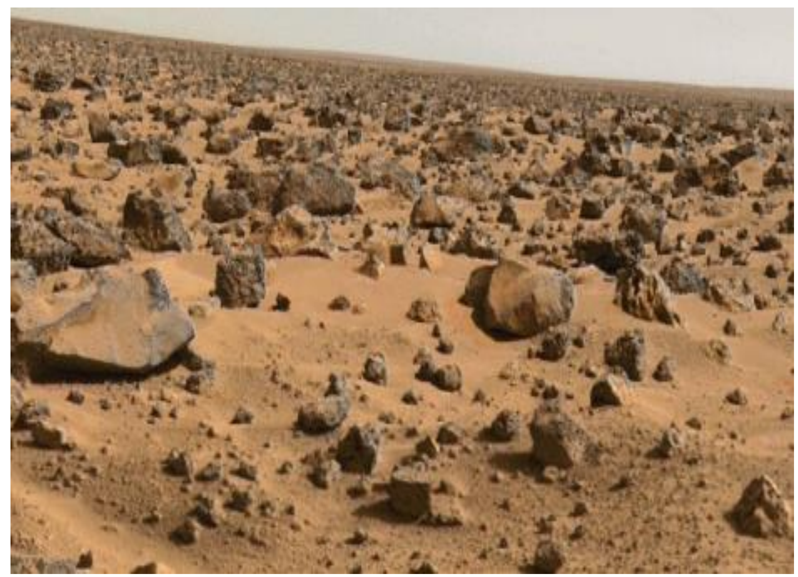


http://www.ics.forth.gr/webfair/technology.html

Feature-Based Mapping

- Compact representation: Suppose the environment consists of a set of isolated features (or landmarks).
- Treat a landmark as a point location (x_k, y_k) in 2D.
- For mapping, a Kalman filter is a natural choice.

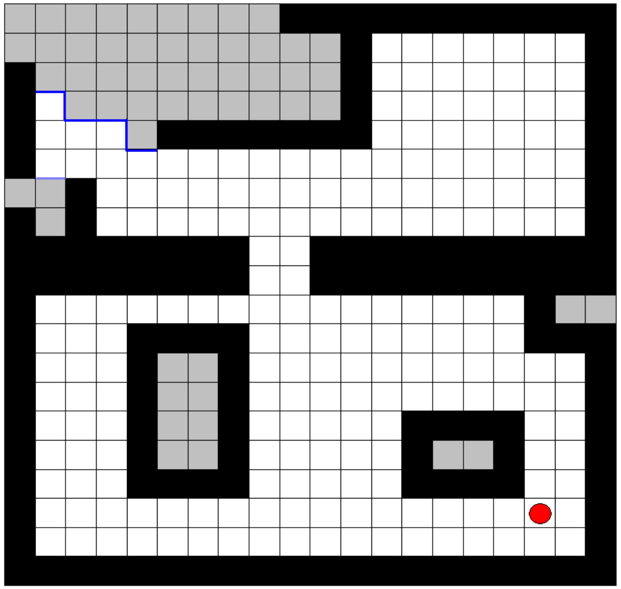
Courtesy by E. Nobot



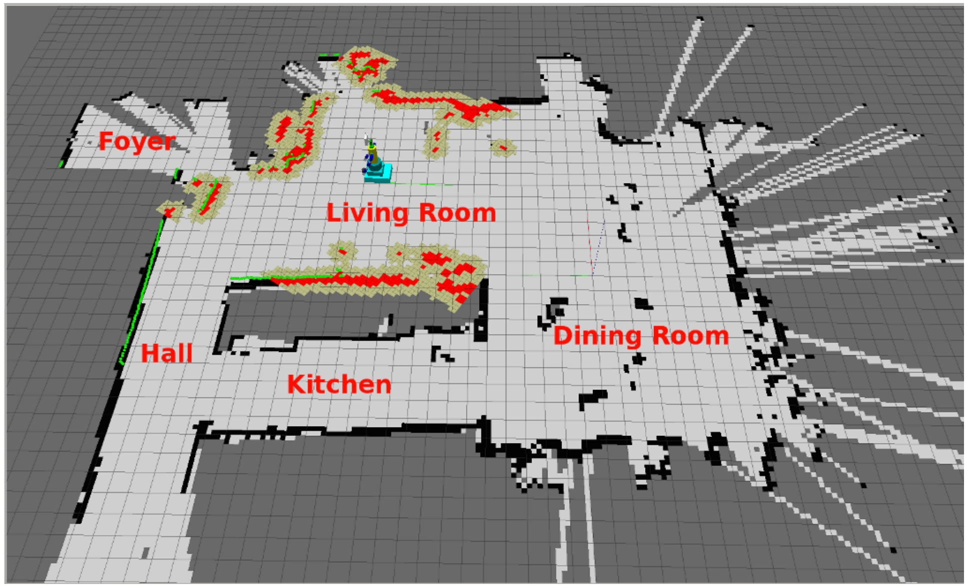
Courtesy by NASA

Grid-Based Mapping

- Discretize the world into a number of cells.
- Work well with noisy range sensor measurements.
- The map structure is rigid and has a fixed resolution.
- Maps are usually two-dimensional but can cover 3D.



http://cse17-iiith.virtual-labs.ac.in/exploration/



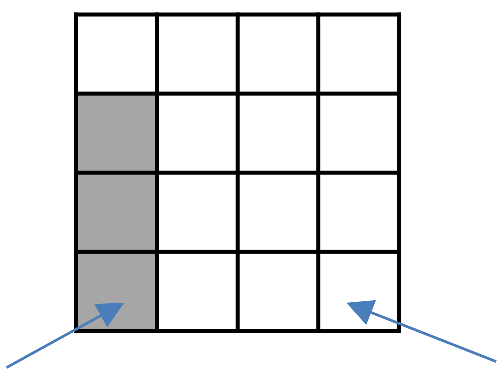
http://www.pirobot.org/blog/0015/

Landmarks vs. Grids

Landmarks	Occupancy Grids
• Parametric model • (Generally) point landmarks are assumed. • Feature models can be arbitrarily precise. • Space and time increase with the contents of the environment.	• Non-parametric model • No assumption about types of features. • (Typically) map resolution is fixed. • Space and time increase with the size of the environment to be mapped.

Occupancy Probability

- The environment is represented as a set of discretized cells, and the area that corresponds to a cell is either **free** or **occupied**. The individual cells are labeled m_i.
- A **binary random variable** to model the occupancy is assigned to each cell which defines whether or not the cell is occupied by an object.



$p(m_i) = 1$ $\qquad$ $p(m_i) = 0$

$p(m_i) = 1$: Occupied

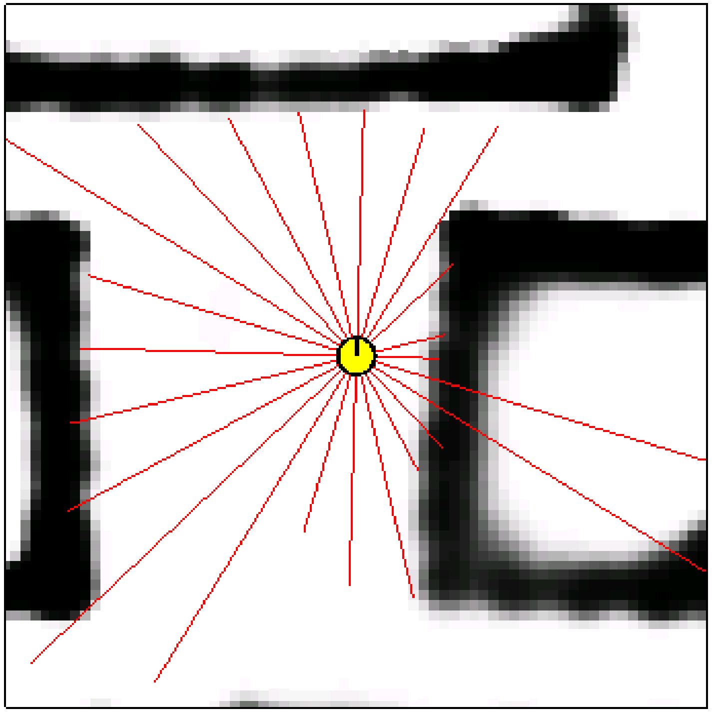
$p(m_i) = 0$: Empty

$p(m_i) = 0.5$: Unknown

Sensors for Occupancy Grid Mapping

- Range sensors (proximity sensors) such as lidar and sonar are commonly used.

https://www.ti.uni-bielefeld.de/html/research/equipment.html



Resulting Occupancy Map

from the lecture for "mobile robotics: method & algorithms" by R. Eustice

SLAM

SLAM Formulation

- Time propagation (prediction step)

$$\underbrace{p(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{k-1})}_{\textbf{Predicted state \& map estimate}} = \int \underbrace{p(\mathbf{x}_k \mid \mathbf{x}_{k-1})}_{\text{State transition probability}} \underbrace{p(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}_{k-1})}_{\text{Previous state \& map estimate}} \, d\mathbf{x}_{k-1}$$

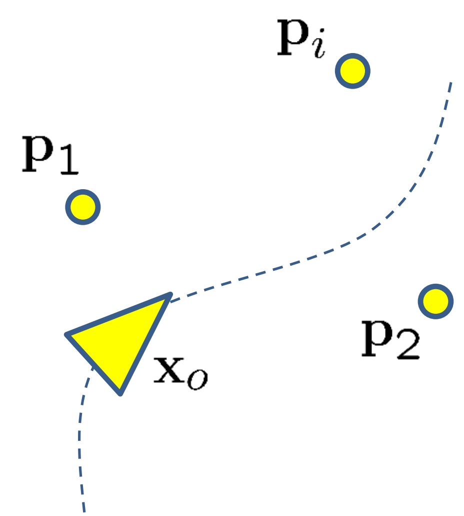
- Measurement update (correction step)

$$\underbrace{p(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_k)}_{\textbf{Updated state \& map estimate}} = \eta \, \underbrace{p(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m})}_{\text{Measurement model}} \underbrace{p(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{k-1})}_{\text{Predicted state \& map estimate}}$$

SLAM Filter Formulation

- State augmentation
  - Vehicle dynamics model: $\dot{\mathbf{x}}_o = \mathbf{f}(\mathbf{x}_o, \mathbf{u}, \mathbf{w})$
  - Landmark state vector: $\mathbf{m} = [\,\mathbf{p}_1^T \quad \mathbf{p}_2^T \quad \cdots\,]^T$
  - Augmented state vector:

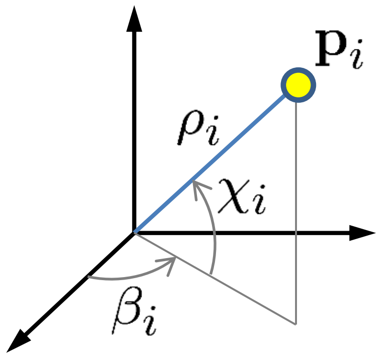
$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{x}}_o \\ \dot{\mathbf{p}}_1 \\ \dot{\mathbf{p}}_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}_o, \mathbf{u}, \mathbf{w}) \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \end{bmatrix}$$



SLAM Filter Formulation

- Measurement equation
  - Relative measurements to nearby landmarks

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_o \\ \mathbf{z}_{p_1} \\ \mathbf{z}_{p_2} \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_o) \\ \mathbf{g}(\mathbf{x}_o, \mathbf{p}_1) \\ \mathbf{g}(\mathbf{x}_o, \mathbf{p}_2) \\ \vdots \end{bmatrix} + \mathbf{v}$$



  - Often provided in the form of range and bearing

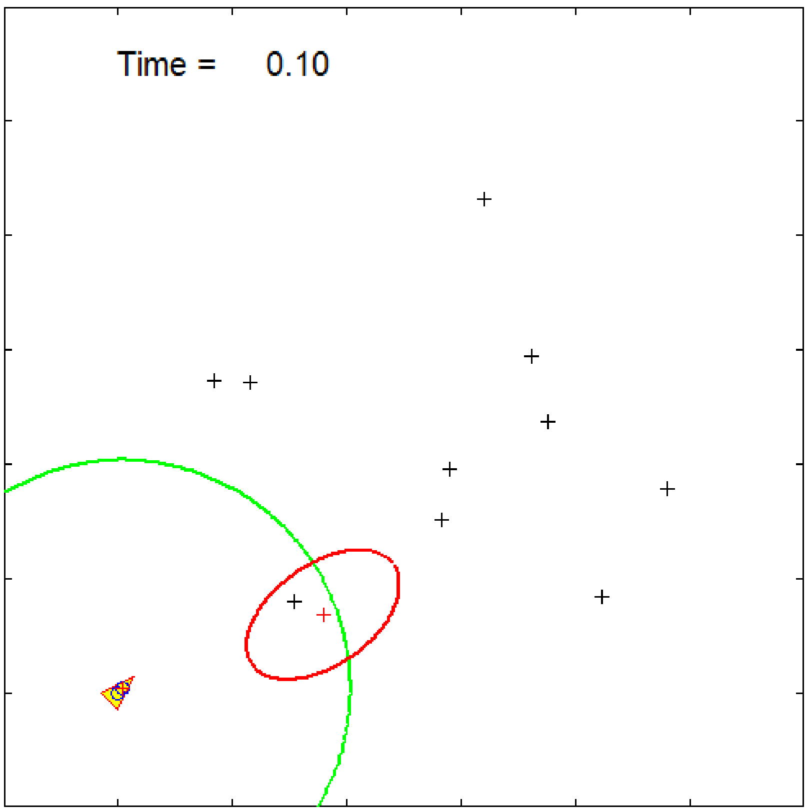
$$\mathbf{g}(\mathbf{x}_o, \mathbf{p}_i) = \begin{bmatrix} \rho_i & \beta_i \end{bmatrix}^T \quad \text{or} \quad \begin{bmatrix} \rho_i & \beta_i & \chi_i \end{bmatrix}^T$$

2D: range & bearing

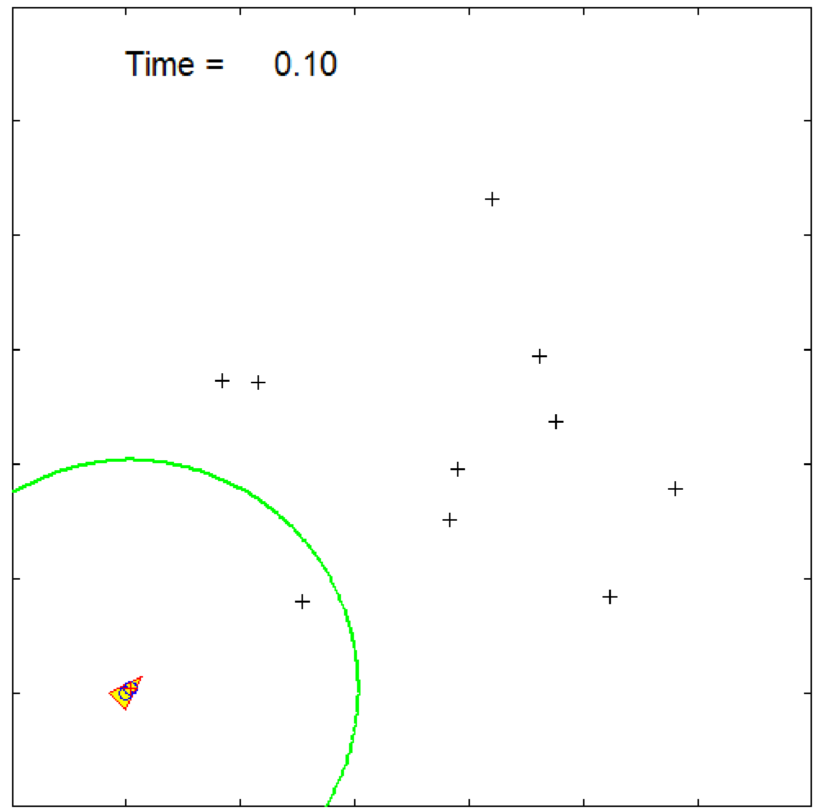
3D: range, azimuth & elevation

Example: EKF SLAM

- w/ range & bearing



- w/ only bearing



FastSLAM

- **A particle filter** (which is known to be effective in many ways) is applied to SLAM problems.
- A naïve implementation of particle filters for SLAM will not work due to its prohibitive computation.
- Given the robot pose, the location of all features can be determined independently.
- This conditional independence allows for employing the **Rao-Blackwellized particle filter** algorithm for SLAM problems, which is a variant of particle filters.

Basic Idea of FastSLAM

- Factorization of the SLAM posterior

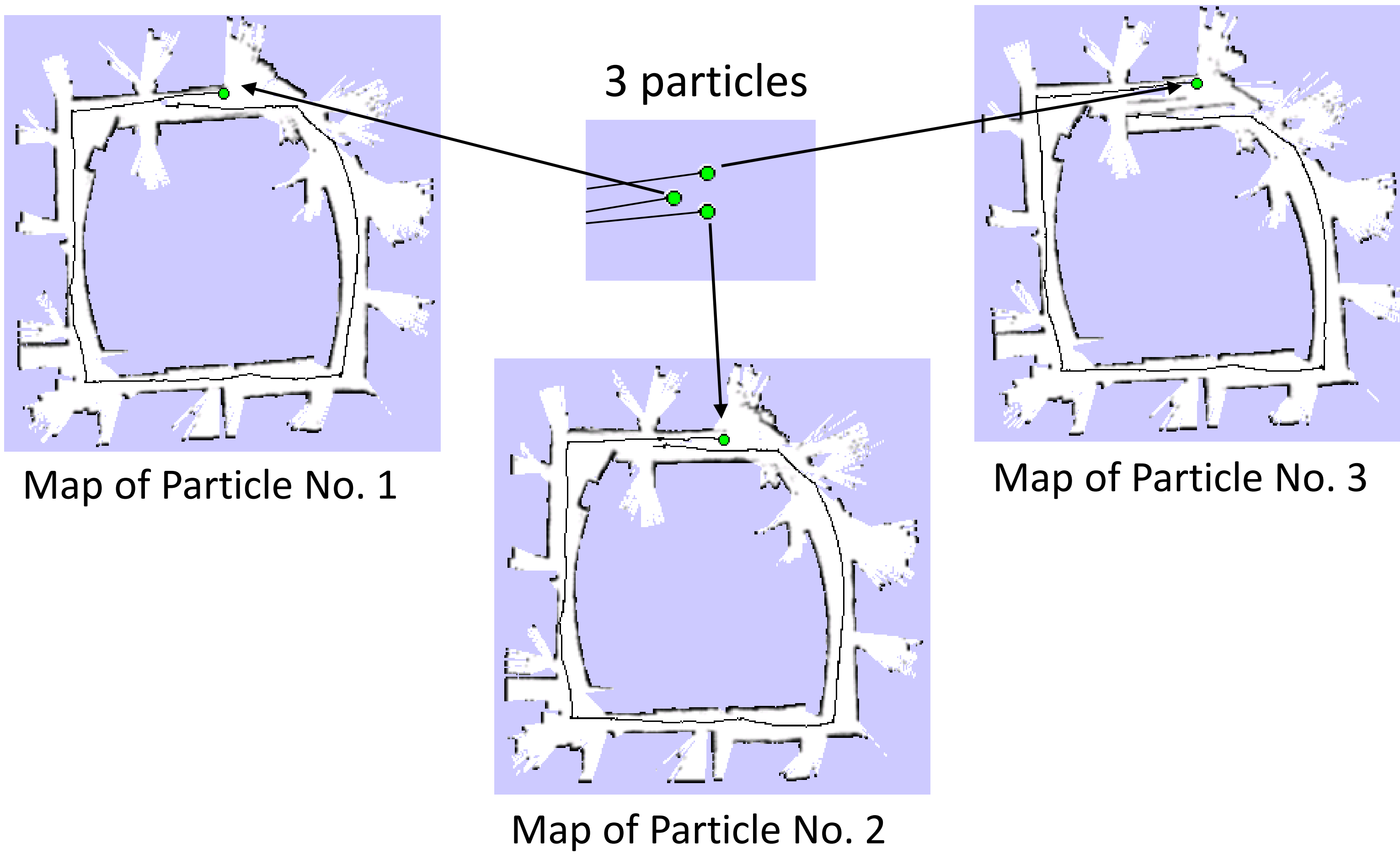
$$\underbrace{p(\overset{\text{Robot's pose}}{\mathbf{x}_k}, \overset{\text{Map}}{\mathbf{m}} \mid \overset{\text{Observations}}{\mathbf{Z}_k})}_{\text{SLAM Posterior}} = \underbrace{p(\mathbf{m} \mid \mathbf{x}_k, \mathbf{Z}_k)}_{\text{Mapping problem by }\textbf{Kalman filtering}} \; \underbrace{p(\mathbf{x}_k \mid \mathbf{Z}_k)}_{\text{Localization problem by }\textbf{particle filtering}}$$

  – **FastSLAM** employs **a particle filter** for estimating the pose posterior and estimates the conditional landmark using **Kalman filters**.

Grid-Based SLAM

- The extension of FastSLAM to occupancy grid maps, which combines Monte-Carlo localization and occupancy grid mapping.
- This approach works on the grid-based map representation and thus does not require any specific landmarks in the environment.
- Each particle represents a potential trajectory of the robot and carries its own map of the environment.
- The importance weight of each particle is determined based on the likelihood of measurements in the particle’s own map.

Illustration: Grid-Based SLAM

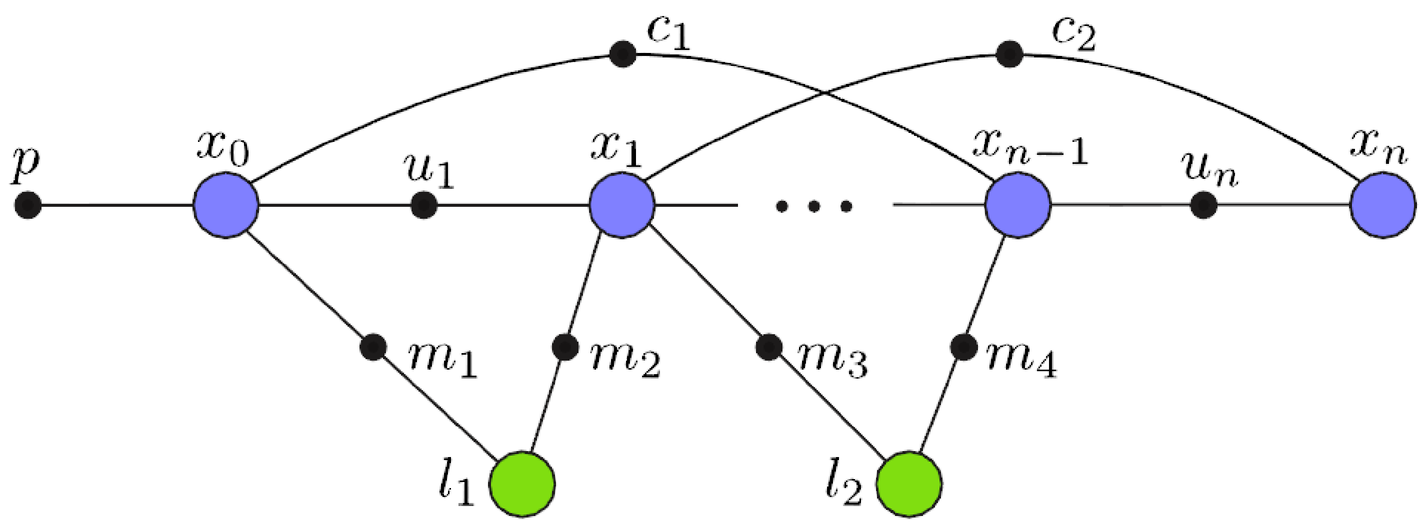


from the lecture slides for “probabilistic robotics” by S. Thrun

Graph SLAM

- Graph SLAM was born from the intuition that the SLAM problem can be interpreted as a sparse graph of nodes and edges between nodes.
- The nodes in the graph represent the robot poses and features in the map (e.g., landmarks).
- Each edge in the graph corresponds to an spatial constraint.
  - Motion between two robot poses
  - Measurement between a pose and a feature in the map
  - Loop closing measurement

Graph Representation



M. Kaess et al., "iSAM2: Incremental smoothing and mapping using Bayes tree," Int. J. of Robot. Res., 2011

p : A prior

x : Robot poses

l : Landmark maps

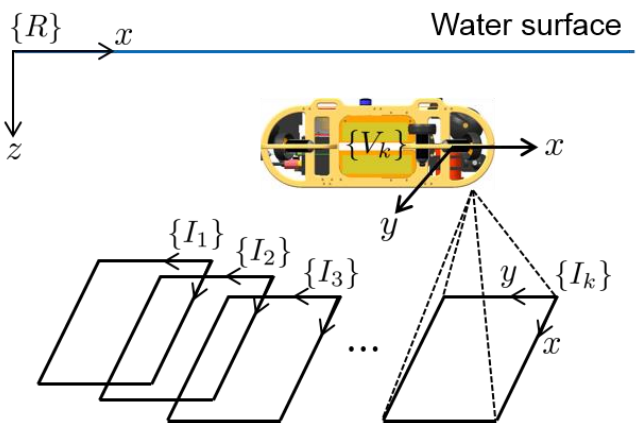
u : Odometry measurements

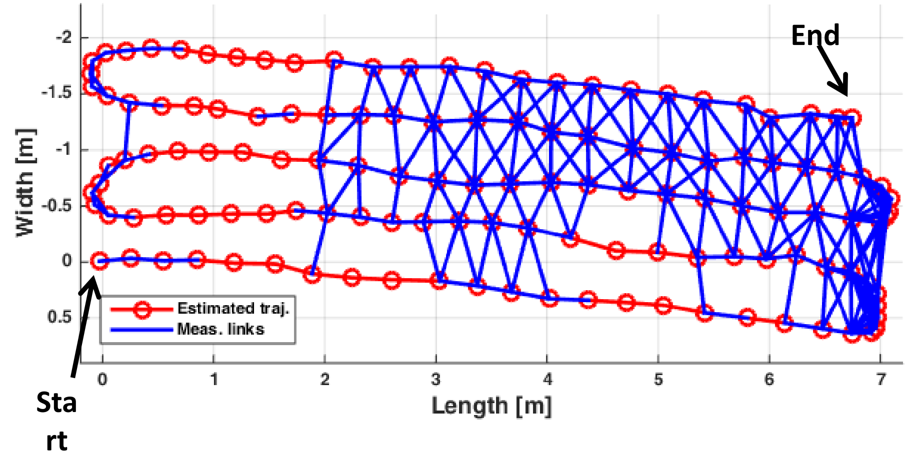
m : Landmark measurements

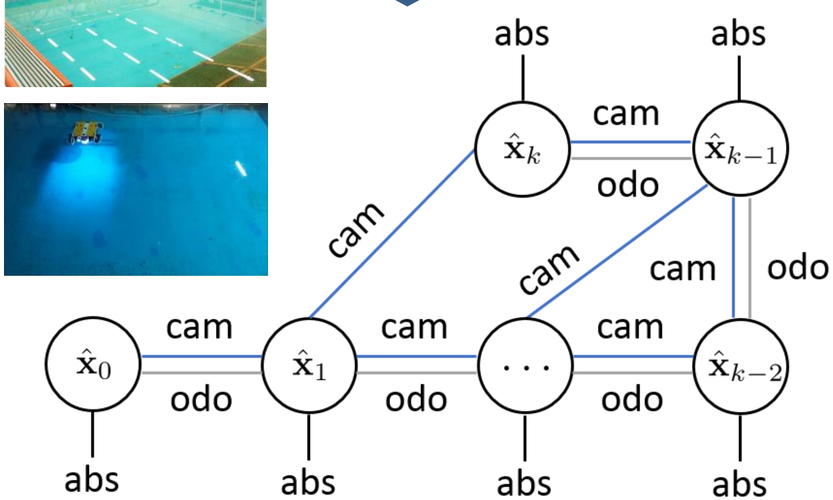
c : loop closing constraints

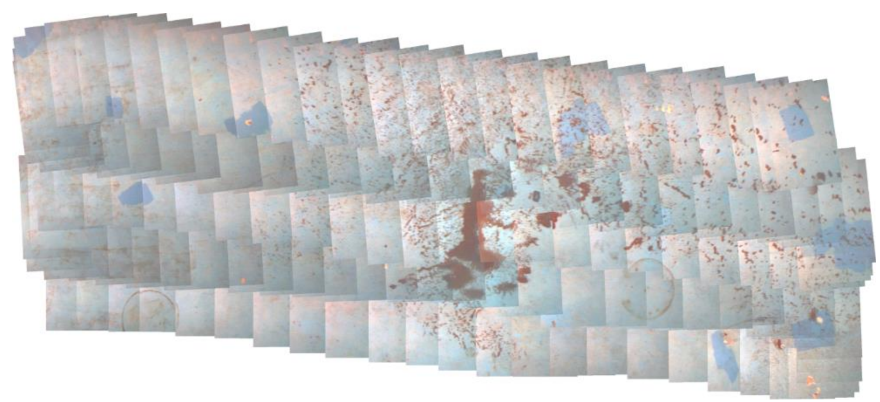
- ✓ Landmark-based SLAM: w/o loop closing constraints
- ✓ Pose-graph SLAM: w/o landmarks and landmark measurements

Example: Pose-Graph Visual SLAM

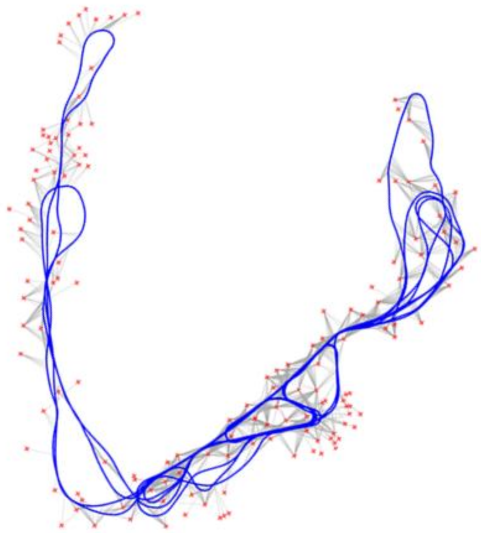








Example: SLAM by iSAM



Odometry only

Blue: GPS data ,   Yellow: iSAM

M. Kaess et al., “iSAM: Incremental smoothing and mapping,” IEEE Trans. On Robotics, 2008

Further Research Issues on SLAM

- The SLAM problem has been the most important research topic in the field of robotics since the mid 80s.
- The problem has been formulated and solved in various ways, and the standard approach to SLAM is now well understood.
- However, a number of challenging research issues still remain that need to be tackled.
- Some of the candidate research issues on SLAM:
  - Solutions to larger problems
  - SLAM in dynamic environments
  - Trajectory optimization (i.e., active SLAM)
  - Cooperative SLAM with multiple robots