

ECGR4161/5196 – July 26, 2011

Read Chapter 5

Exam 2 contents:

- Labs 0, 1, 2, 3, 4, 6
- Homework 1, 2, 3, 4, 5
- Book Chapters 1, 2, 3, 4, 5
- All class notes

Challenges in Map Representation

3 Fundamentals of Map Representation

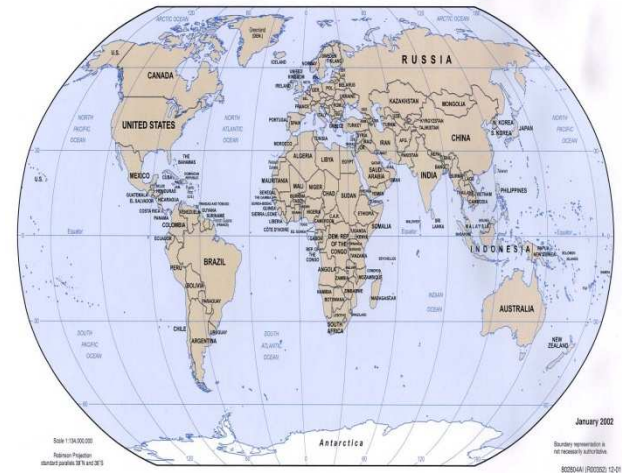
1. Map precision must match precision of robot's precision to achieve goals
2. Map precision and object representation must match sensor data precision
3. Map complexity has a direct impact on computational complexity

Major Issues with Map Representation

- Memory constraints on large and precise map representations
- How do you define objects and nodes in a map? (Indoor vs. Outdoor)
- What is the object? (Sensor shortcomings, vision research state of the art, etc.)
- Real world is dynamic (humans, nature, transient objects, etc.)
- Sensor fusion problem yet to be solved

References:

Siegwart, Roland. *Autonomous Mobile Robots*. Cambridge, Massachusetts. The MIT Press, 2011, 284-296.



Monte Carlo Localization for Mobile Robots

Probabilistic approach to localization problems

Global position estimation– ability to determine the robot's position in a previously learned map given no other info than the robot is somewhere on the map

Local tracking is keeping track of that position over time

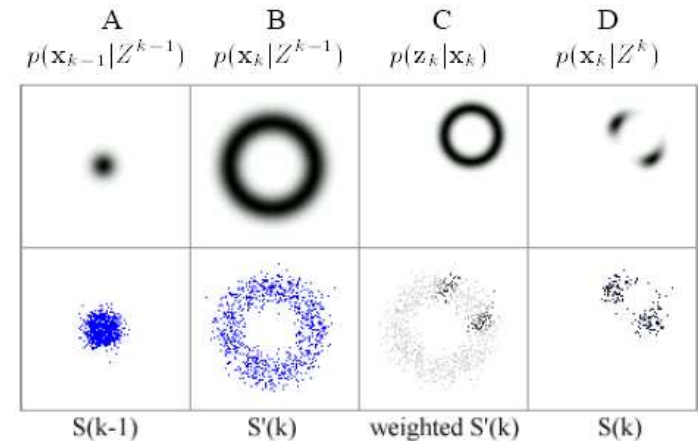


Fig. 1: The probability densities and particle sets for one iteration of the algorithm. See text for detail.

$\{z_k, i = 1..k\}$ The position measurements up to time k

$\mathbf{x} = [x, y, \theta]^T$ The 3D state vector

$p(\mathbf{x}_k | Z^{k-1})$ Predicts current position using motion only (given by integration)

$p(\mathbf{x}_k | Z^{k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1} | Z^{k-1}) d\mathbf{x}_{k-1}$ \mathbf{u}_{k-1} Is some type of known control input

$p(z_k | \mathbf{x}_k)$ Likelihood the robot is at location \mathbf{x}_k given z_k was observed

$p(\mathbf{x}_k | Z^k) = \frac{p(z_k | \mathbf{x}_k) p(\mathbf{x}_k | Z^{k-1})}{p(z_k | Z^k)}$ Posterior density over \mathbf{x}_k (Bayes Theorem)

A: Uncertainty – represented by cloud of particles

B: Robot moved 1 meter. Robot is somewhere within 1 meter radius

C: Observing landmark .5 meters away in top-right corner

D: Re-sampling from the weighted set – starting point for next iteration

Jeffrey Skelnik

jbskelni@uncc.edu

Monte Carlo Localization (Particle Filtering)

- Based on a collection of samples (also known as particles)
 - Each sample consists of a possible location along with the probability (importance weights) that the robot is at that location
 - More samples \rightarrow More accuracy at the expense of computational time
- Algorithm:

1- inputs: Distance u_t , sensor reading z_t , sampleset $S_t = \{ (x_t^{(i)}, w_t^{(i)}) \mid i = 1, \dots, n \}$

2- for $i = 1$ to n do *// First update the current set of samples*

– $x_t = \text{updateDist}(x_{t-1}, u_t)$ *// Compute new location*

– $w_t^{(i)} = \text{prob}(z_t \mid x_t^{(i)})$ *// Compute new probability*

3- $S_{t+1} = \text{null}$ *// Then resample to get the next generation of samples*

4- for $i = 1$ to n do

– Sample an index j from the distribution given by the weights in S_t

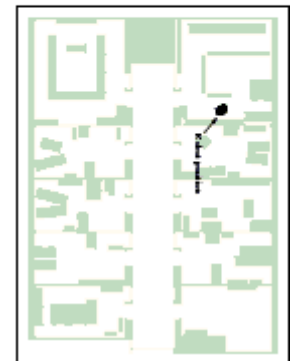
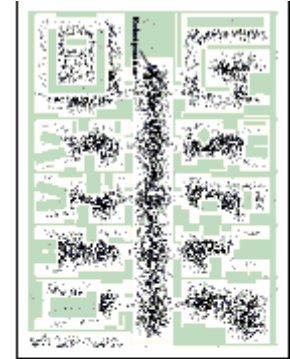
– Add $(x_t^{(j)}, w_t^{(j)})$ to S_{t+1} *// Add sample j to the set of new samples*

5- return S_{t+1}

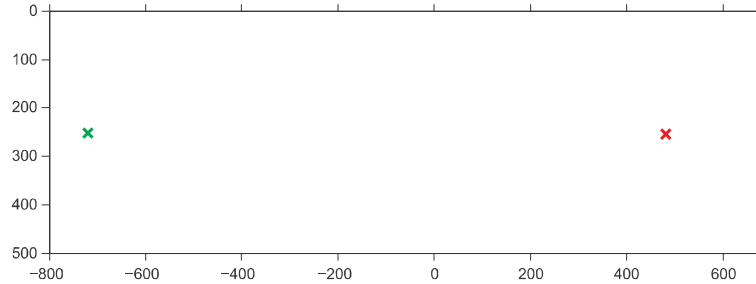
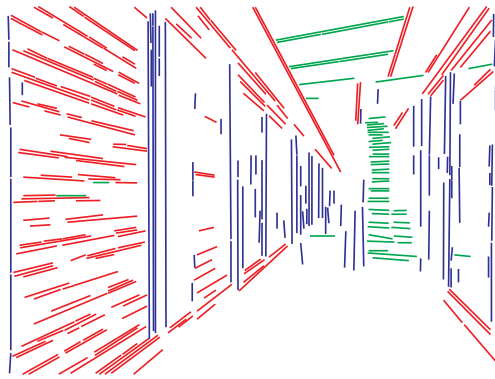
- http://www.youtube.com/watch?v=dItTmkR4hsQ&feature=player_embedded#at=12

<http://www.mcs.alma.edu/LMICSE/LabMaterials/AI/MonteCarloLoc/MonteCarlo.htm>

http://www.cs.washington.edu/ai/Mobile_Robotics//mcl/



Localization: Using Visual Cues and Vanishing Points



Base image compared with best fit lines



Corridor
a)

Flaws within the system...
No two images are ever the same
Needs a preexisting definition of “land marks” to search for
Reflections brought out by surfaces



Corridor
c)

- 1) Saurer, Olivier, Marc Pollefeys, and Friedrich Fraundorfer. *Visual Localization Using Global Visual Features and Vanishing Points*. *Computer Vision and Geometry Group*. Web. 25th July 2011. <http://www.clef2010.org/resources/proceedings/clef2010labs_submission_89.pdf>.

Markov Localization a probabilistic algorithm

$$Bel(L_t = l) = \frac{P(s_t | L_t = l, s_{1,\dots,t-1}, a_{1,\dots,t-1}) P(L_t = l | s_{1,\dots,t-1}, a_{1,\dots,t-1})}{P(s_t | s_{1,\dots,t-1}, a_{1,\dots,t-1})}$$

Equation (1) Markov Localization algorithm from [1]

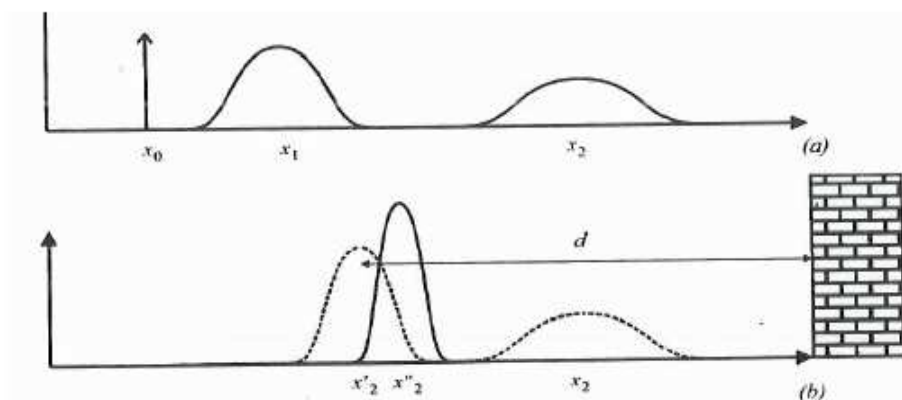
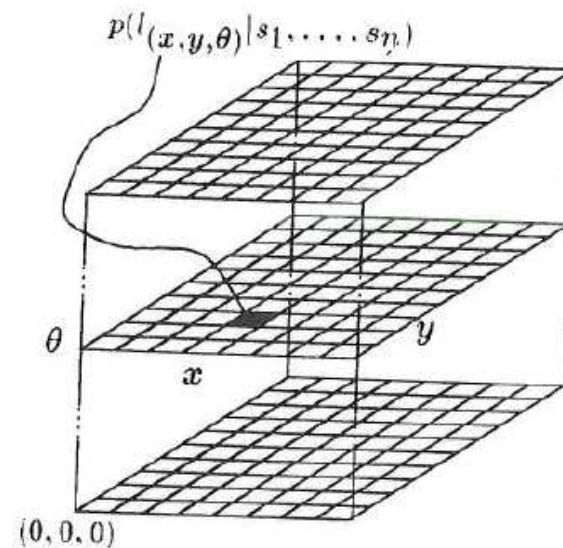


Fig (2) In probabilistic robotic (a) prediction phase (b) Perception phase from [2]

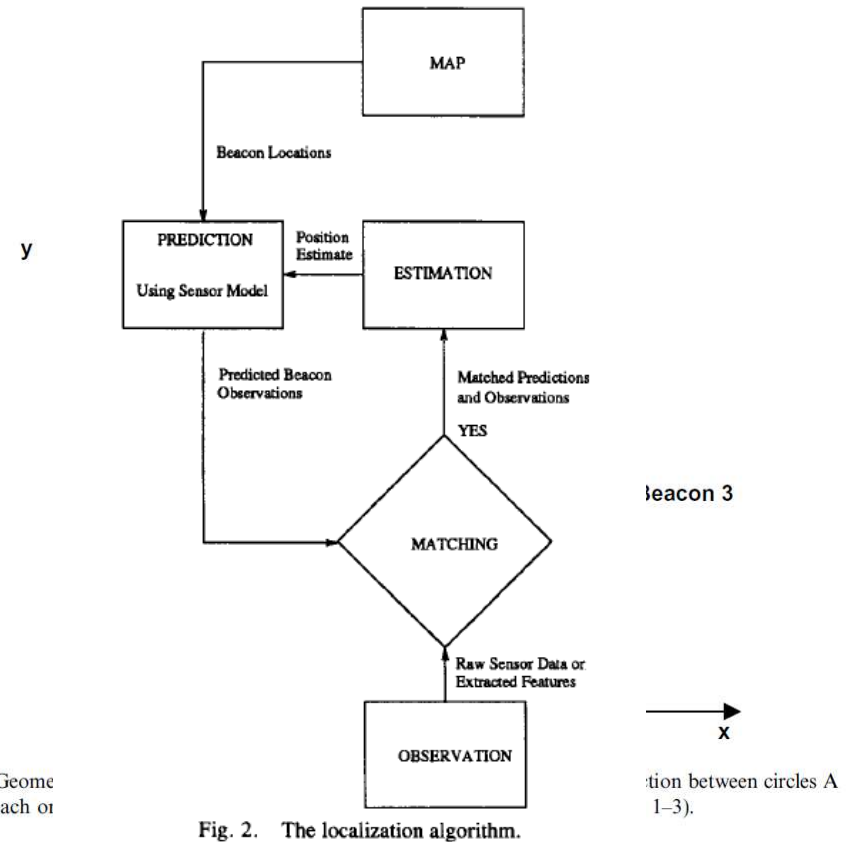


Fig(1) 3D array used in Markov Localization [2]

- References: 1. <http://www.cs.washington.edu/homes/fox/diss/diss.html>
 2. Chapters 5 of the introduction to Autonomous Mobile Robots

Local Positioning - Passive Sonar Beacons

- Local Positioning Systems use intentionally placed local (as opposed to global) 'beacons' to identify a reference within a space
- Reference Systems have the chief advantage of resolving error (incremental errors do not accumulate as in proprioceptive 'dead reckoning')
- Simple triangulation provides absolute localization data
- Passive references are the simplest to employ
- Sonar is just one example lasers, light, or radio also work
- Not effective in unprepared locales



[Video:](#)
[Light-Reflector Passive Beacons](#)

Refs:

- 1) Microcontroller Based System for 2D Localization; Casanova, Quijada, Garcia-Bermejo, Gonzalez
- 2) Mobile Robot Localization by Tracking Geometric Beacons; Durant-Whyte

Turtle Bot Localization Using Kinect and I-Robot Base.

The Turtle Bot utilizes a few sensors to determine its position based on a previously formed map or model of the area.

It has a single axis gyro, edge sensors, Kinect sensor with IR sensors and a camera, and encoders to determine position, speed, and obstacles.

It can be programmed to just avoid obstacles, identify object shapes, or stay within map/model boundaries.



Localization Methods for a Mobile Robot in Urban Environment

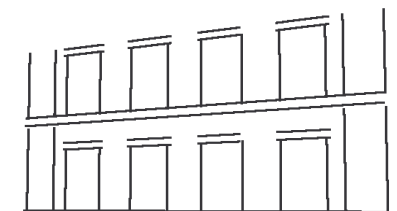
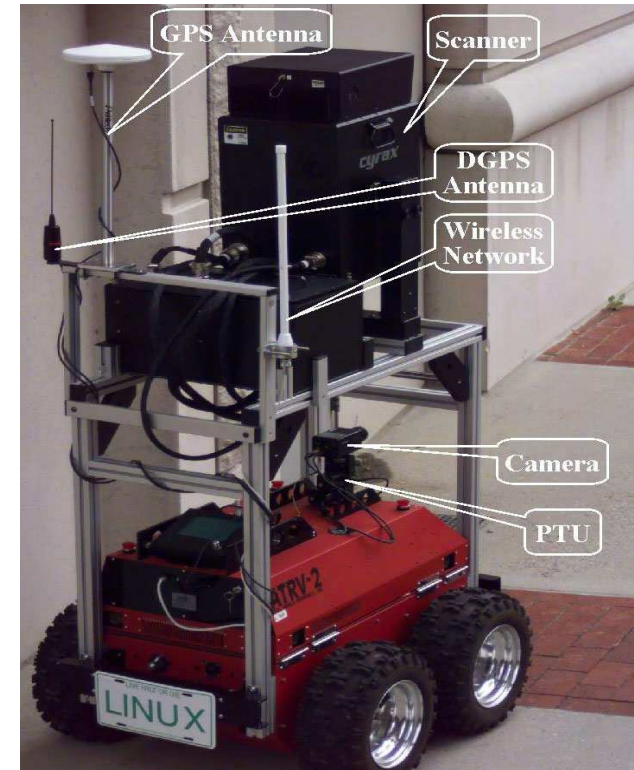
Outdoor urban environments. pose unique set of challenges Different from both indoor and outdoor open environment.

Open space localization :-

Odometry- minute errors accumulate over time, The error can not be bounded without the use of sensor such as GPS.

Digital Compass- helps to find orientation
The GPS receiver allows the position uncertainty from accumulating beyond acceptable limits.

Visual Localization :- On-demand updates, only when the open-space configuration fails.
On demand allows more time for image processing operations which increases the robustness of the overall system.



Reference:- ieeexplore.ieee.org/iel5/8860/29531/01339385.pdf

Monte Carlo Localization for Mobile Robots

key advantages:

- represent multi-modal distributions and thus can *globally* localize a robot and drastically reduces the amount of memory required

Prediction Phase:

use a motion model to predict the current position of the robot in the form of a predictive PDF $p(\mathbf{x}_k | \mathbf{Z}^{k-1})$

Current state \mathbf{x}_k is dependent on the previous state \mathbf{x}_{k-1} & known control input \mathbf{u}_{k-1} .

the motion model is specified as a conditional density $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$

predictive density over \mathbf{x}_k is then obtained by integration:

$$p(\mathbf{x}_k | \mathbf{Z}^{k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{Z}^{k-1}) d\mathbf{x}_{k-1}$$

Update Phase:

use a *measurement model (MM)* to incorporate information from the sensors to obtain the posterior PDF $p(\mathbf{x}_k | \mathbf{Z}^k)$

assume that the measurement \mathbf{z}_k is conditionally independent of \mathbf{Z}^{k-1} given \mathbf{x}_k , and that the MM is given in terms of a likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$

The posterior density over \mathbf{x}_k is obtained using Bayes theorem:

$$p(\mathbf{x}_k | \mathbf{Z}^k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Z}^{k-1})}{p(\mathbf{z}_k | \mathbf{Z}^{k-1})}$$

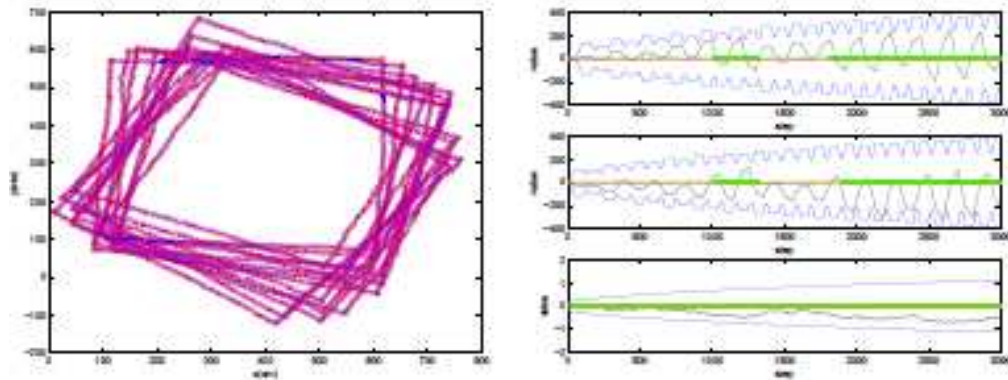
[1] [Frank Dellaert](#), [Dieter Fox](#), Wolfram Burgard, and [Sebastian Thrun](#), "Monte Carlo Localization for Mobile Robots," *IEEE International Conference on Robotics and Automation (ICRA99)*, May, 1999.



Kalman Filters

Definition:

A Kalman Filter (KF) is a state estimator that works on a prediction-correction basis. It depends on the reference coordinate system (x, y, ϕ) [1].



(a) True and Estimated trajectory of square driving robot with measurements

(b) Posterior estimation error (solid black), 95% confidence interval (solid blue), run with and without measurements (solid green and solid red, respectively)

Figure 1: Position Tracking with Corrections

Principle:

The KF is used as a control for a dynamic system and for state estimation. It provides the information that cannot directly be measured by estimating the values of the variables from indirect and noisy measurements [1]

Sources:

[1] Robot Localization and Kalman Filters (http://www.negenborn.net/kal_loc/thesis.pdf)