EE565:Mobile Robotics
Lecture 10

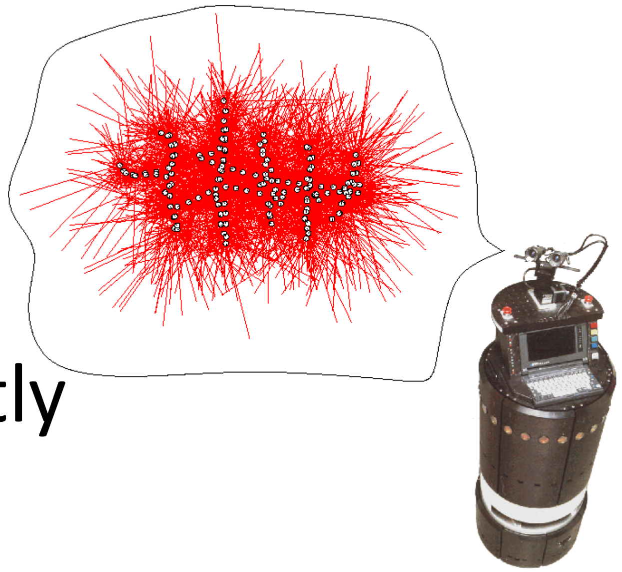
Welcome

Dr. Ahmad Kamal Nasir

Today's Objectives

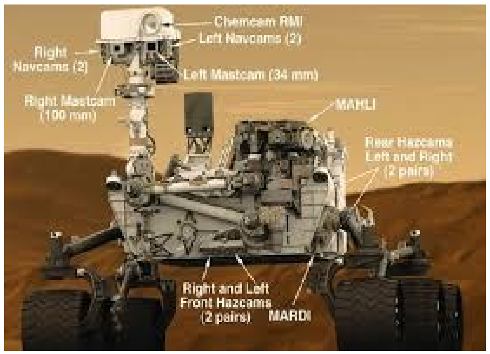
- Mapping
  - Feature mapping
  - Grid Mapping
- Introduction to SLAM
- Feature/Landmark SLAM
- Grid Mapping (GMapping)

Why Mapping?



- Learning maps is one of the fundamental problems in mobile robotics
- Maps allow robots to efficiently carry out their tasks, allow localization …
- Successful robot systems rely on maps for localization , path planning, activity planning etc.

Motivation and Challenges



- **Advantages:** Faster objective completion time, re-assign task in case a robot fails, tasks can be done which are beyond the capability of single robot
- **Challenges:**
  - Map merging, large dynamic sparse outdoor environment
  - Controlling and managing of multi-robot system is challenging because the system requires handling of multiple robots with heterogeneous capabilities
  - Standard software architecture to avoid re-implementation of basic communication and non-interoperability
- **Application:** Multi-robot map building in absence of priori map such as sea ports, destroyed nuclear plants…

Problems in Mapping

- Sensor interpretation
  - How do we extract relevant information from raw sensor data?
  - How do we represent and integrate this information over time?

- Robot locations have to be estimated
  - How can we identify that we are at a previously visited place?
  - This problem is the so-called data association problem.

The General Problem of Mapping

- Formally, mapping involves, given the sensor data,

$$d = \{u_1, z_1, u_2, z_2, \ldots, u_n, z_n\}$$

to calculate the most likely map

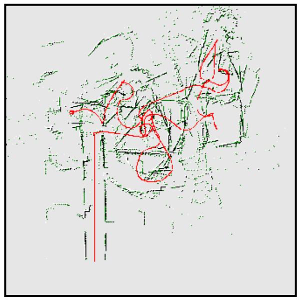
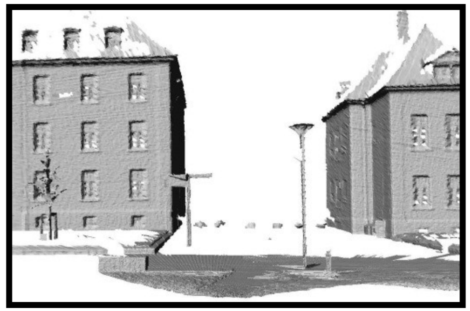
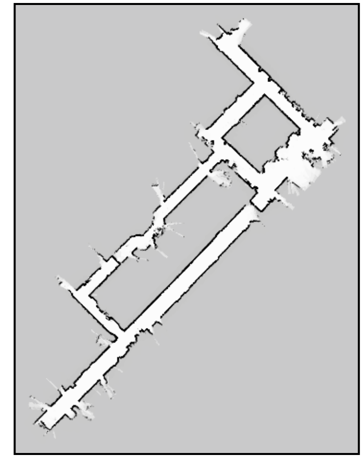
$$m^* = \arg\max_m P(m \mid d)$$

Mapping as a Chicken and Egg Problem

- So far we learned how to estimate the pose of the vehicle given the data and the map.
- Mapping, however, involves to simultaneously estimate the pose of the vehicle and the map.
- The general problem is therefore denoted as the simultaneous localization and mapping problem (SLAM).
- Throughout this section we will describe how to calculate a map given we know the pose of the vehicle.

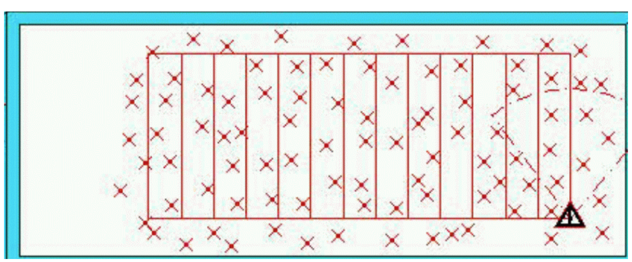
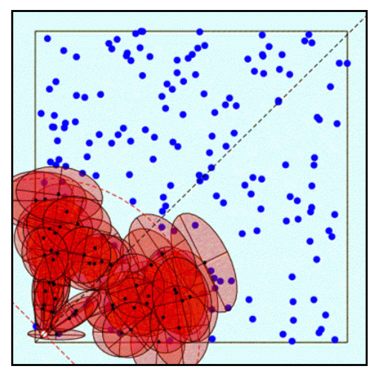
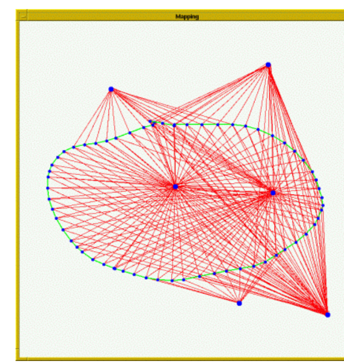
Types of SLAM-Problems

- Grid maps or scans

[Lu & Milios, 97; Gutmann, 98: Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01;…]

- Landmark-based

[Leonard et al., 98; Castelanos et al., 99: Dissanayake et al., 2001; Montemerlo et al., 2002;…

Full vs. Online SLAM

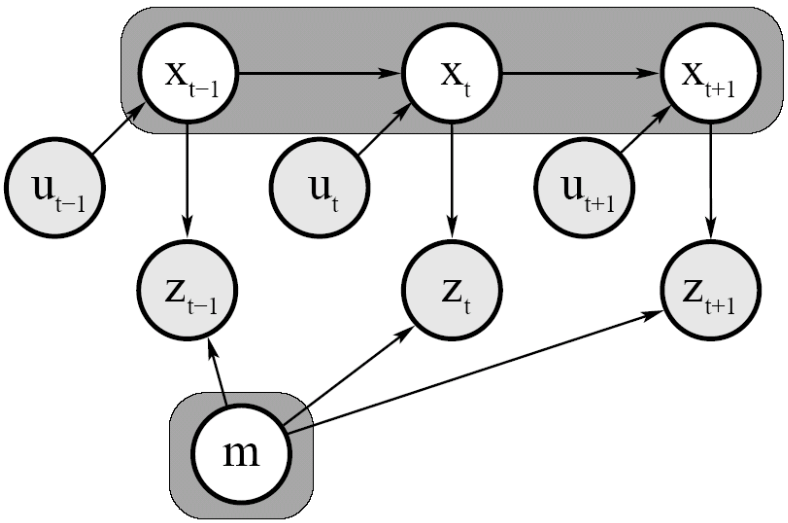
- Full SLAM calculates the robot state over all time up to time t

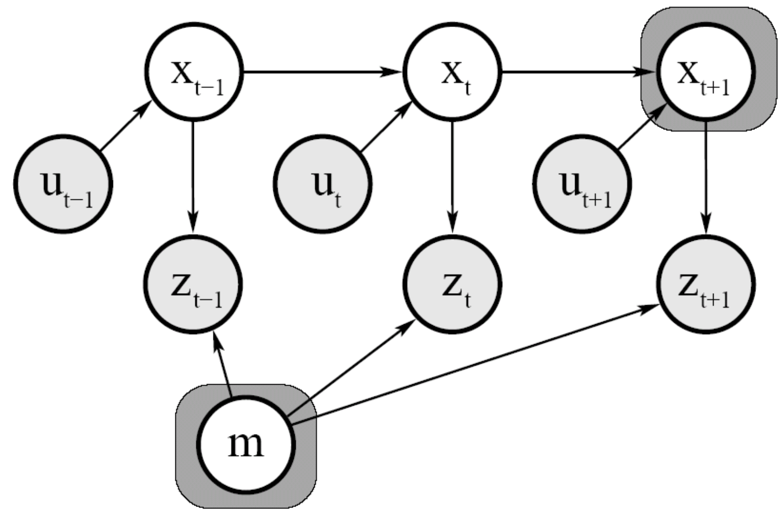
$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

- Online SLAM calculates the robot state for the current time t

$$p(x_t, m \mid z_{1:t}, u_{1:t}) = \int\int\ldots\int p(x_{1:t}, m \mid z_{1:t}, u_{1:t})\, dx_1 dx_2 \ldots dx_{t-1}$$

Full vs. Online SLAM





Full SLAM

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

Online SLAM

$$p(x_t, m \mid z_{1:t}, u_{1:t}) = \int\int \dots \int p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) \, dx_1 dx_2 \dots dx_{t-1}$$

Two Example SLAM Algorithms

- Extended Kalman Filter (EKF) SLAM
  - Solves online SLAM problem
  - Uses a linearized Gaussian probability distribution model
- FastSLAM
  - Solves full SLAM problem
  - Uses a sampled particle filter distribution model

Extended Kalman Filter SLAM

- Solves the Online SLAM problem using a linearized Kalman filter
- One of the first probabilistic SLAM algorithms
- Not used frequently today but mainly shown for its explanatory value

Kalman Filter Components

Linear discrete time dynamic system (motion model)

State — Control input — Process noise

$$x_{t+1} = F_t x_t + B_t u_t + G_t w_t$$

State transition function — Control input function — Noise input function with covariance Q

Measurement equation (sensor model)

Sensor reading — State — Sensor noise with covariance R

$$z_{t+1} = H_{t+1} x_{t+1} + n_{t+1}$$

Sensor function

EKF Equations

Propagation (motion model):

$$\hat{x}_{t+1/t} = F_t \hat{x}_{t/t} + B_t u_t$$

$$P_{t+1/t} = F_t P_{t/t} F_t^T + G_t Q_t G_t^T$$

Update (sensor model):

$$\hat{z}_{t+1} = H_{t+1} \hat{x}_{t+1/t}$$

$$r_{t+1} = z_{t+1} - \hat{z}_{t+1}$$

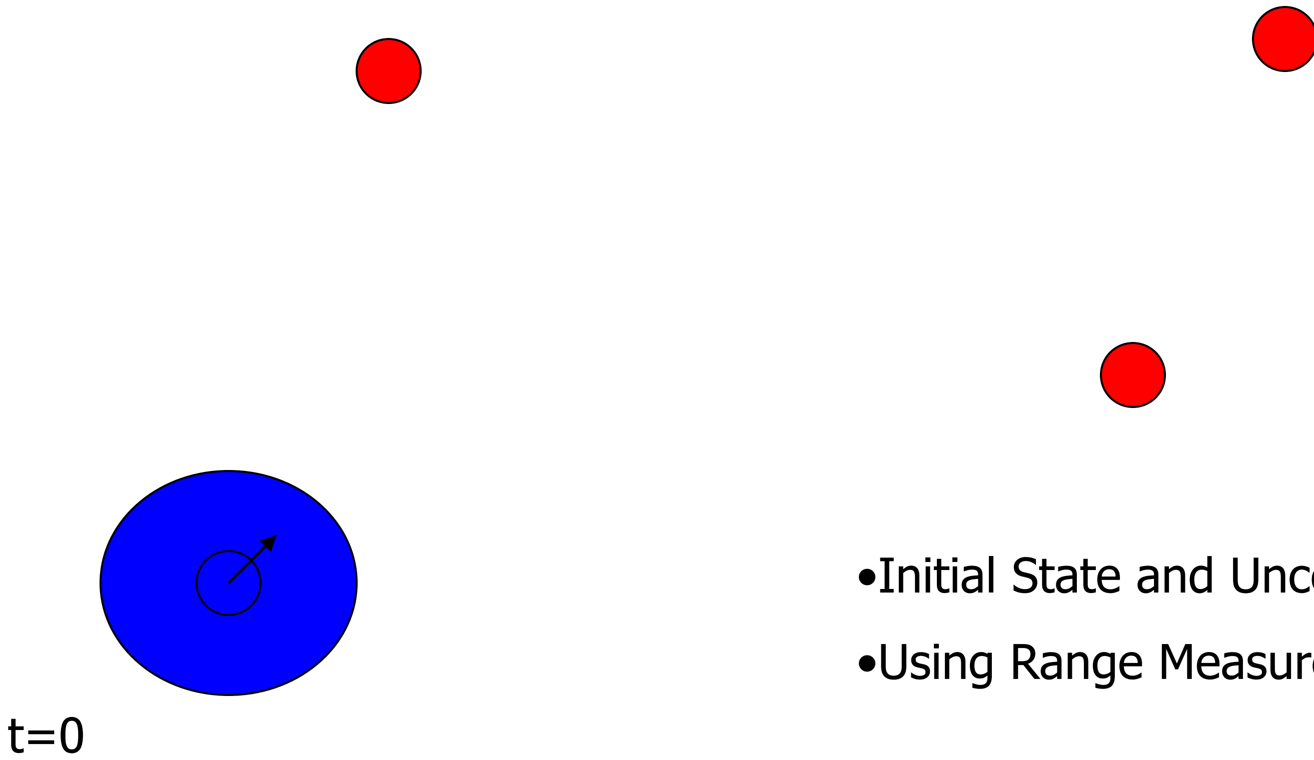
$$S_{t+1} = H_{t+1} P_{t+1/t} H_{t+1}{}^T + R_{t+1}$$

$$K_{t+1} = P_{t+1/t} H_{t+1}{}^T S_{t+1}{}^{-1}$$

$$\hat{x}_{t+1/t+1} = \hat{x}_{t+1/t} + K_{t+1} r_{t+1}$$

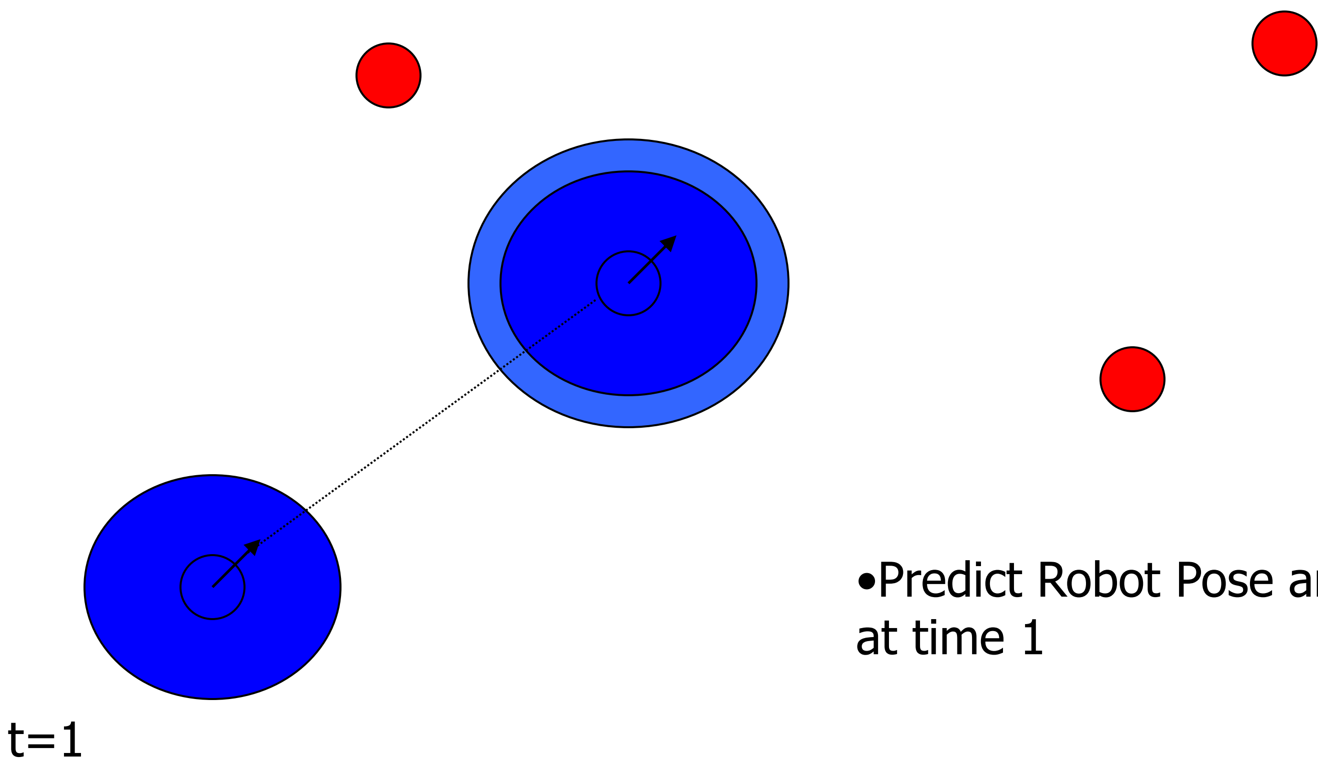
$$P_{t+1/t+1} = P_{t+1/t} - P_{t+1/t} H_{t+1}{}^T S_{t+1}{}^{-1} H_{t+1} P_{t+1/t}$$

EKF Example



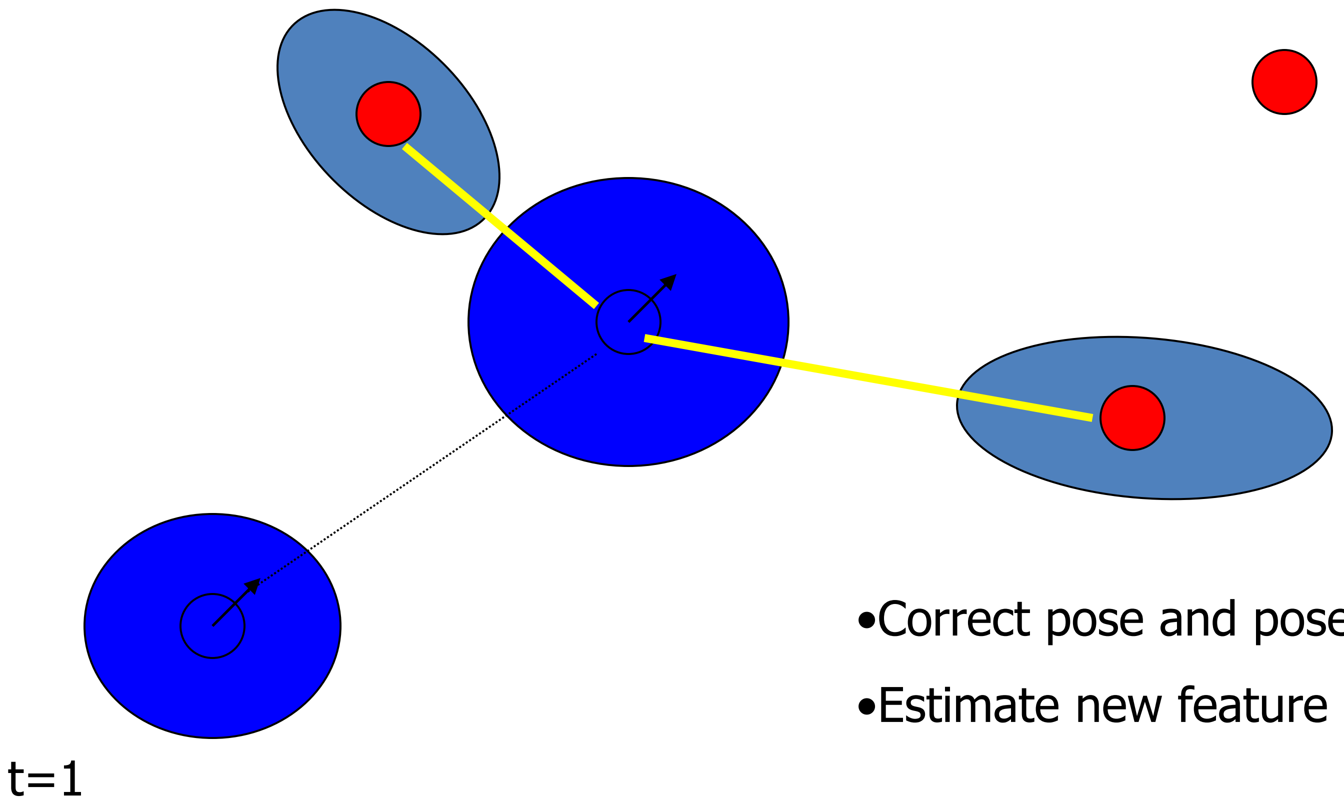
- Initial State and Uncertainty
- Using Range Measurements

EKF Example



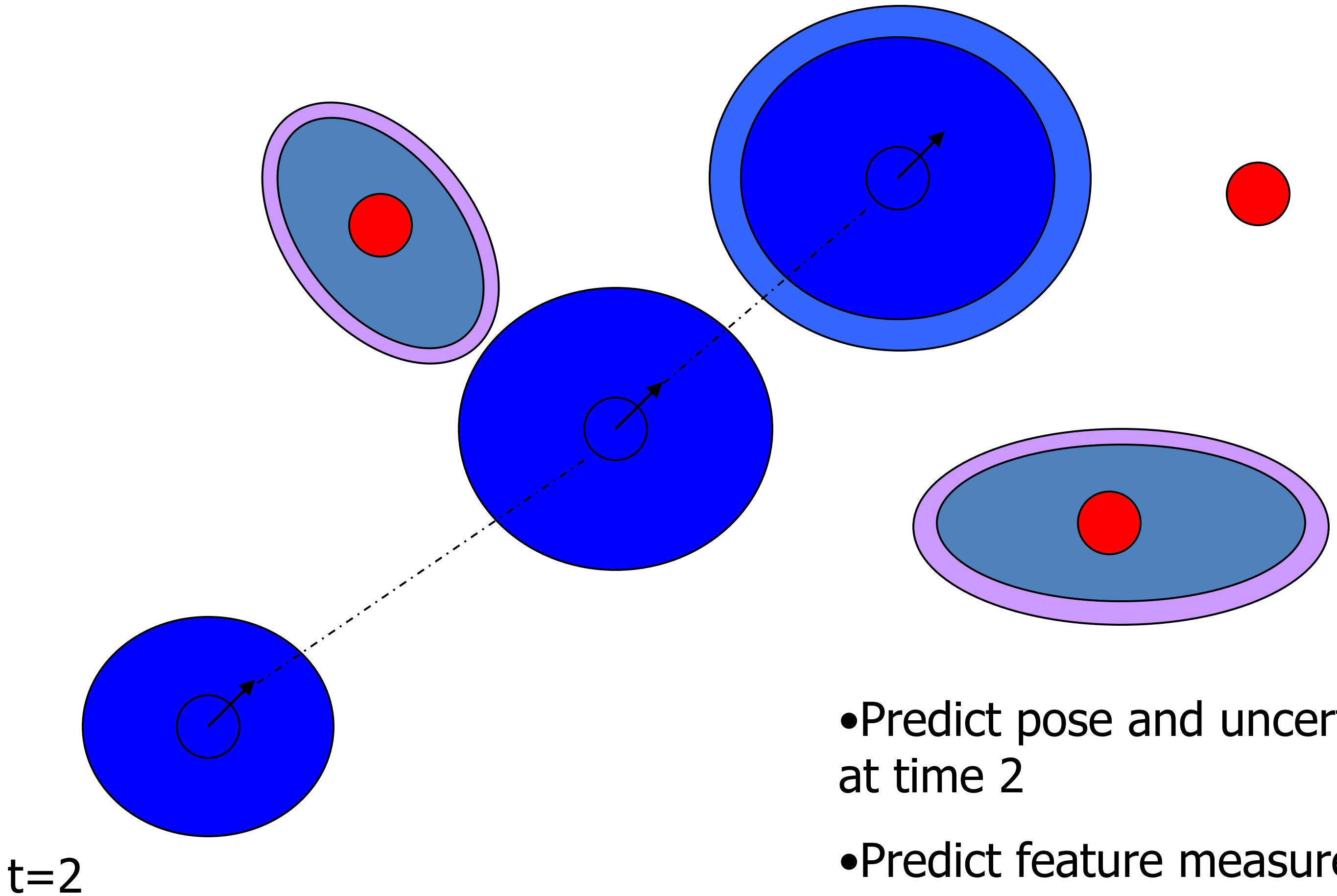
- Predict Robot Pose and Uncertainty at time 1

EKF Example



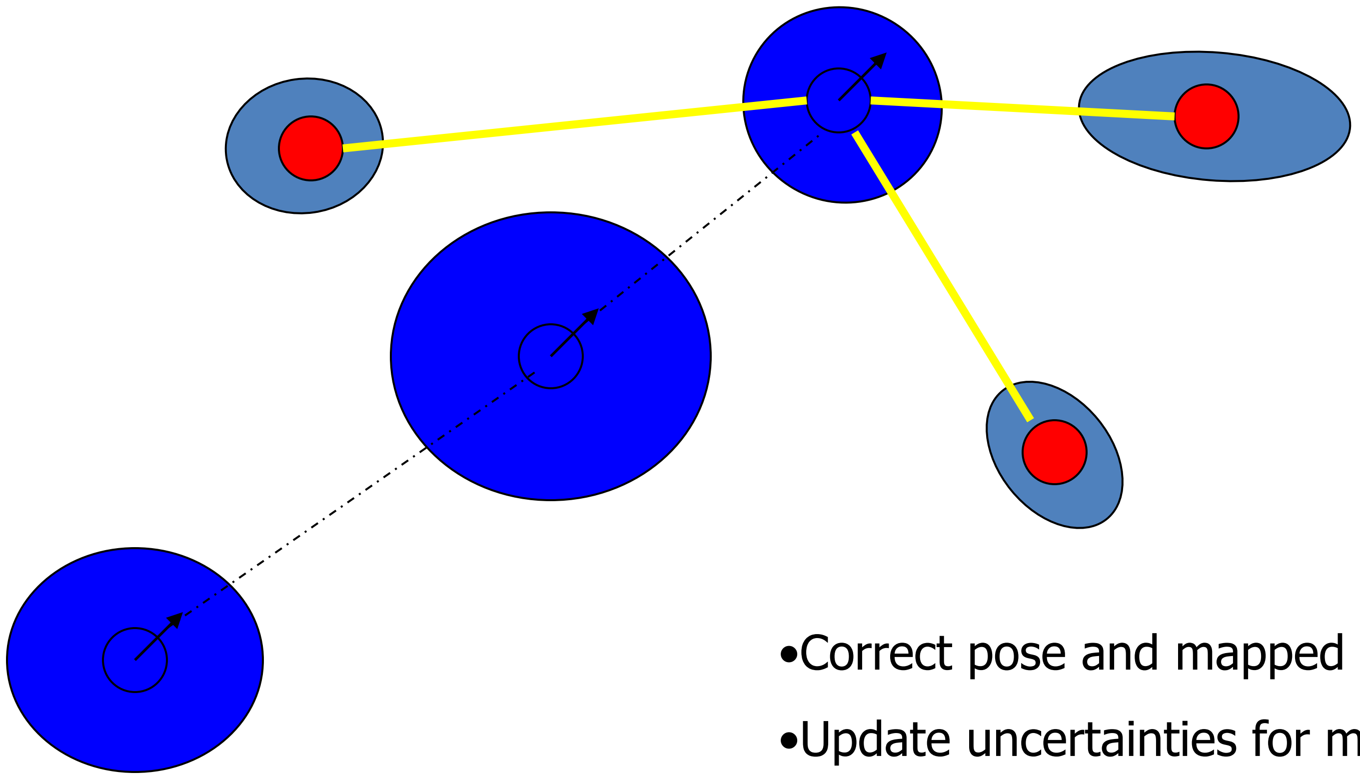
- Correct pose and pose uncertainty
- Estimate new feature uncertainties

EKF Example



- Predict pose and uncertainty of pose at time 2
- Predict feature measurements and their uncertainties

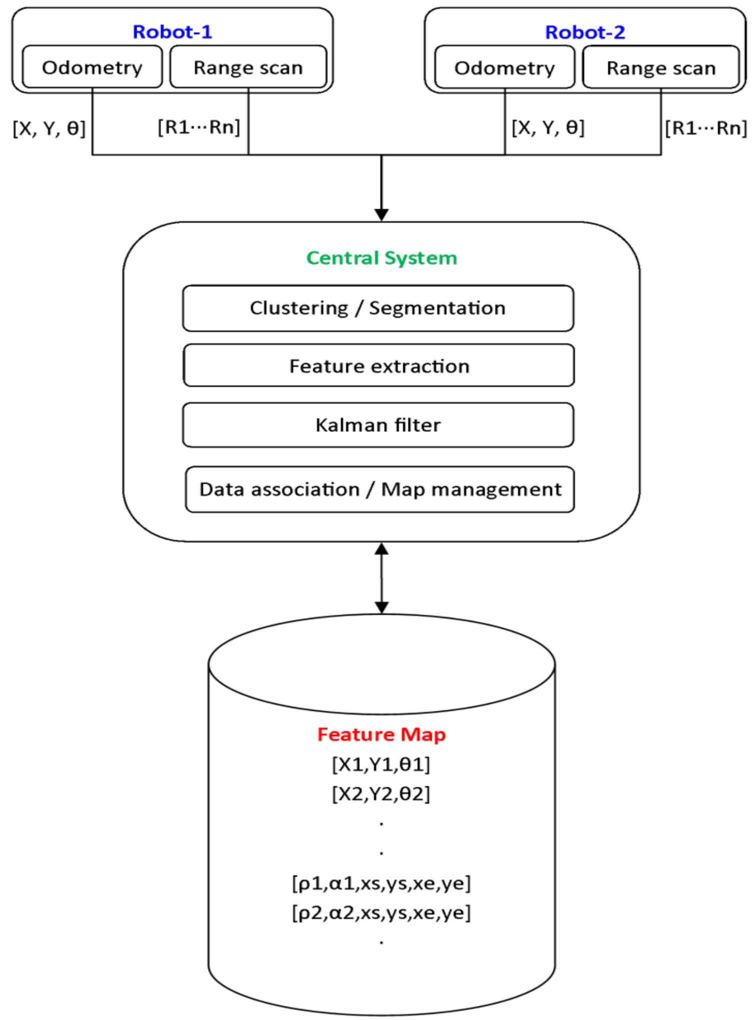
EKF Example



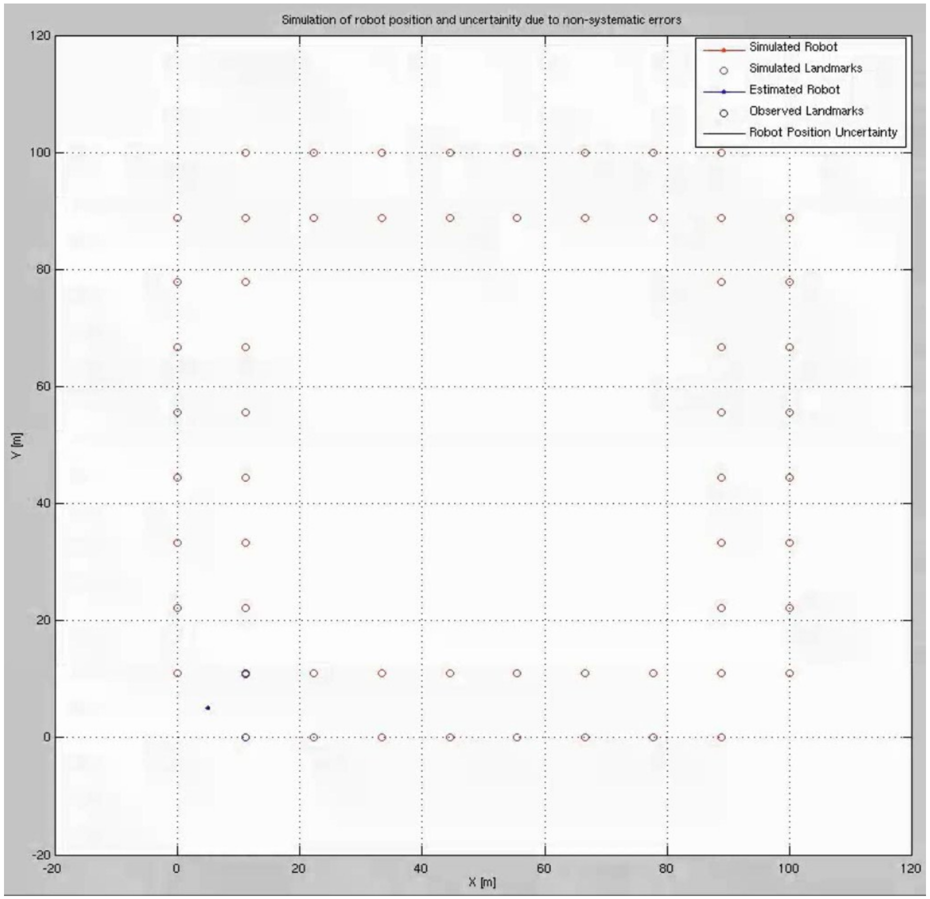
t=2

- Correct pose and mapped features
- Update uncertainties for mapped features
- Estimate uncertainty of new features

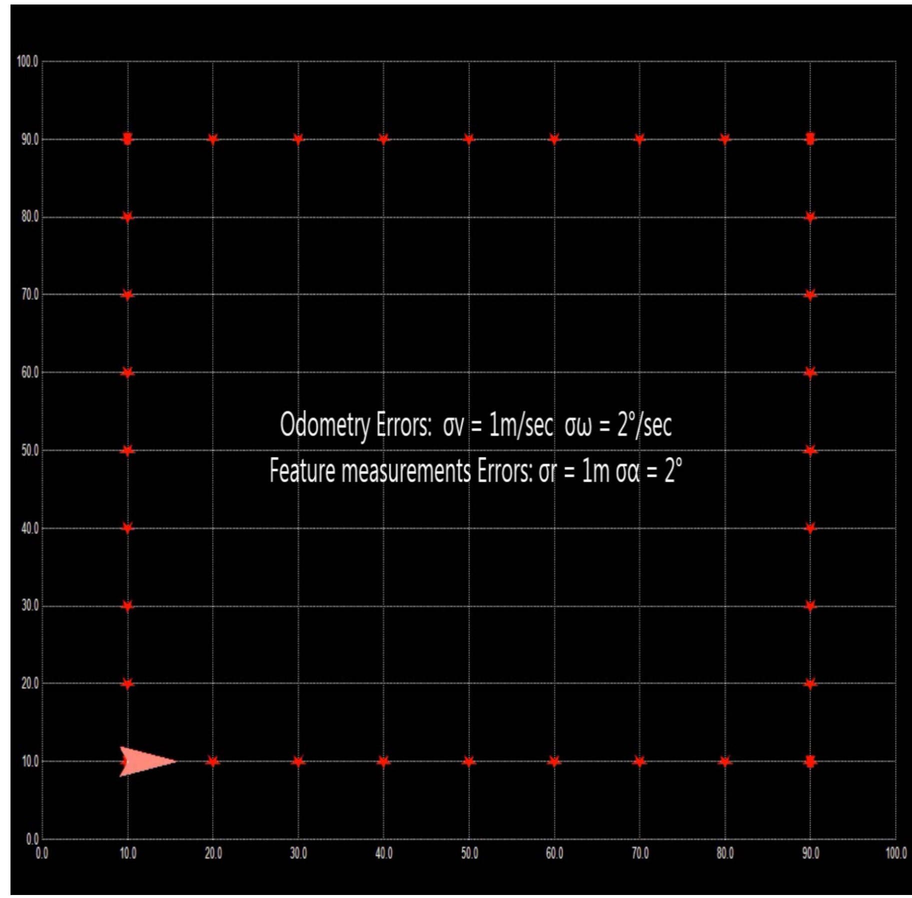
Implementation



SLAM

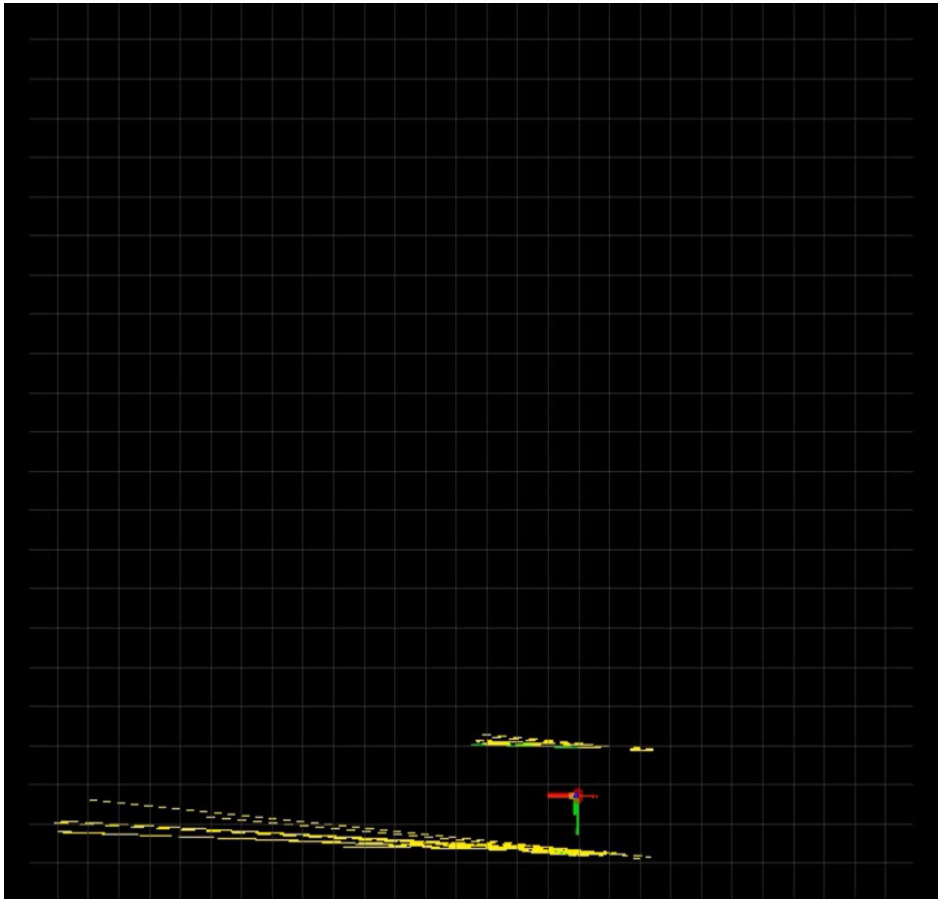


Effect of odometeric errors on robot uncertainty

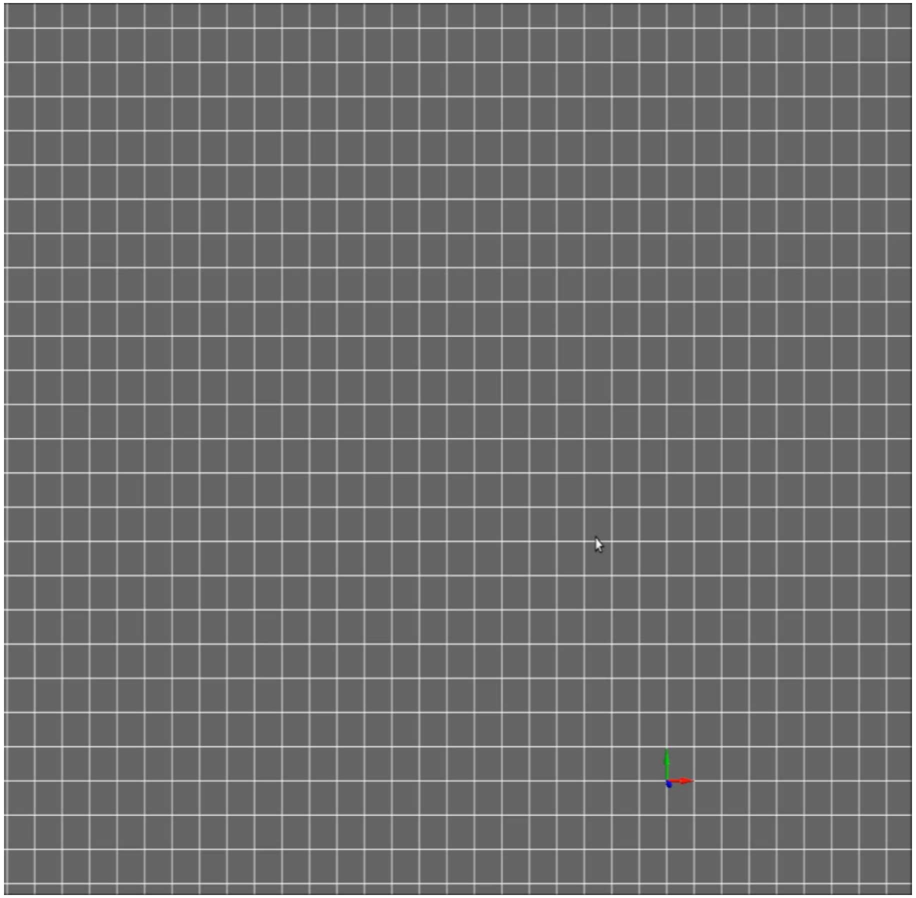


Feature based SLAM to reduce robot uncertainty

Feature based SLAM

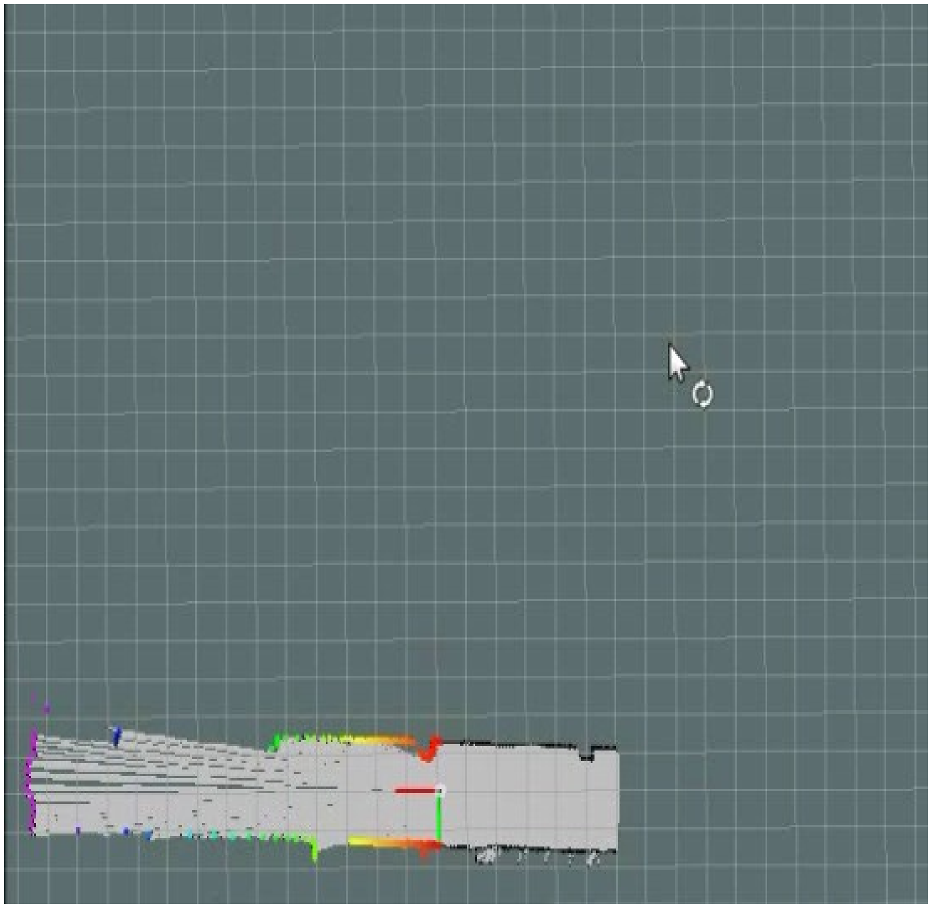


2D Line feature based SLAM using Laser Scanner

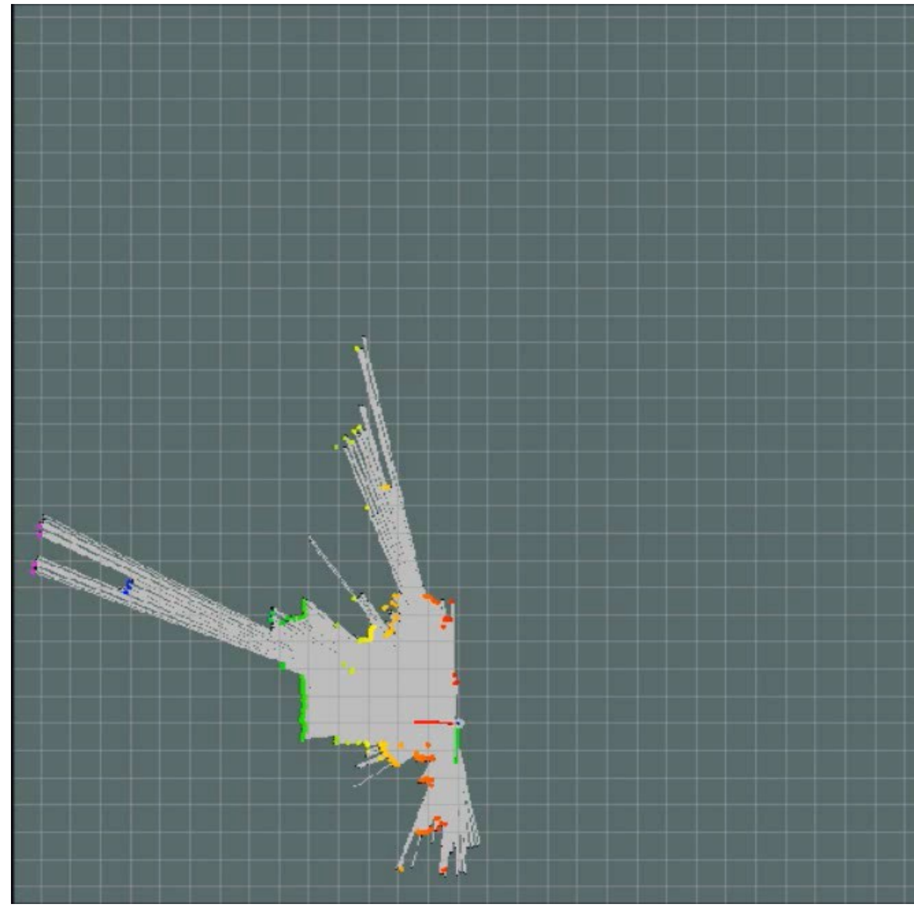


3D plane map using Kinect

Occupancy Grid based SLAM

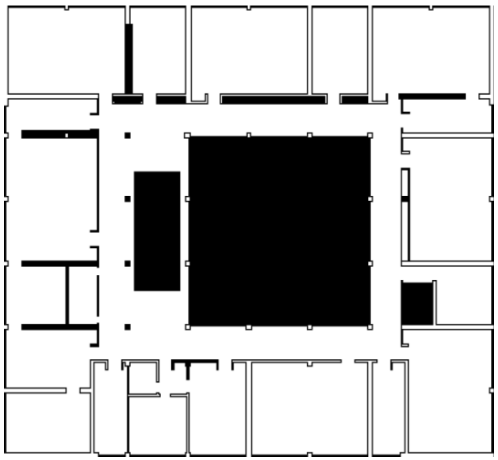


Grid based SLAM Experiment on H-F1

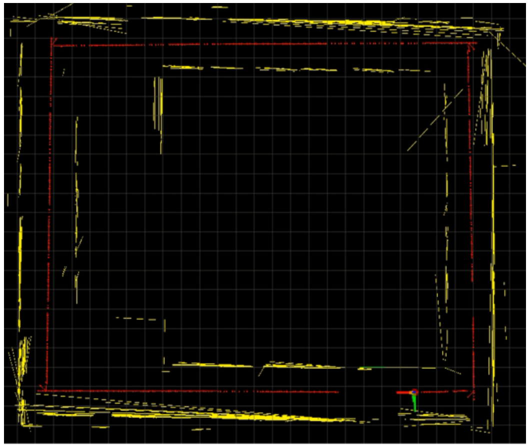


Grid based SLAM Experiment on H-F0

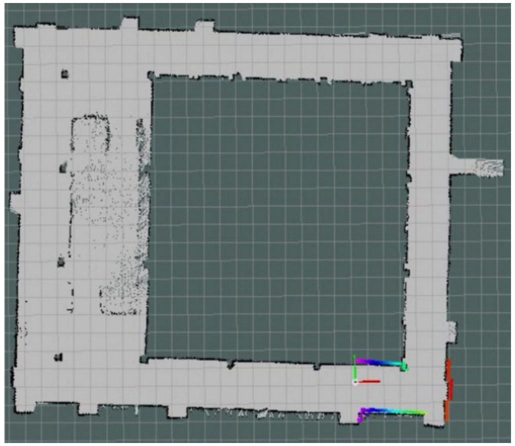
Mapping Results



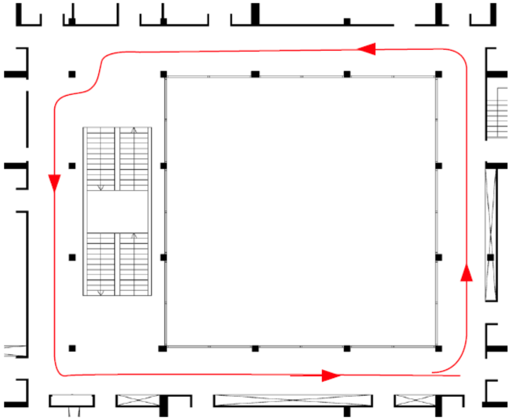
Original map



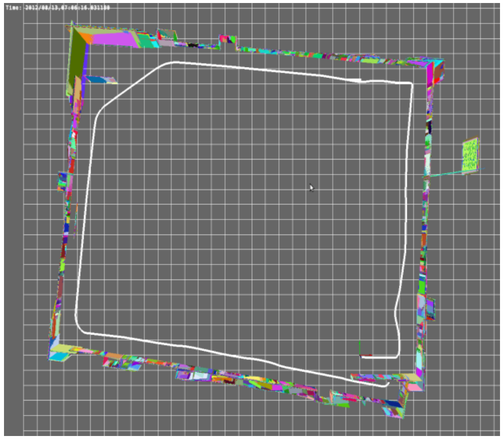
Line feature map



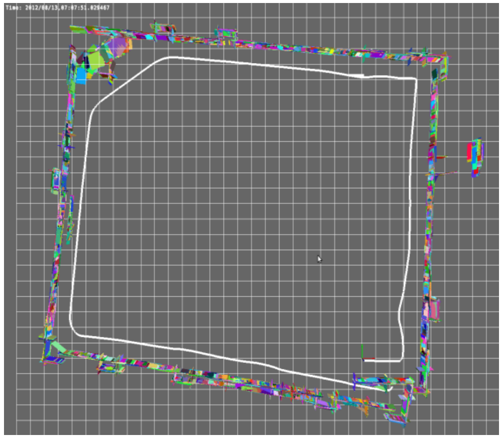
Grid map



Planned trajectory



Map using Hough transform



Map using RANSAC

SLAM ormulization

Robot state: $\boldsymbol{x_r = [x, y, \theta]^T}$

Line features: $\boldsymbol{m_l = [r, \alpha]^T}$

Plane features: $\boldsymbol{m_p = [r, \theta, \varphi]^T}$

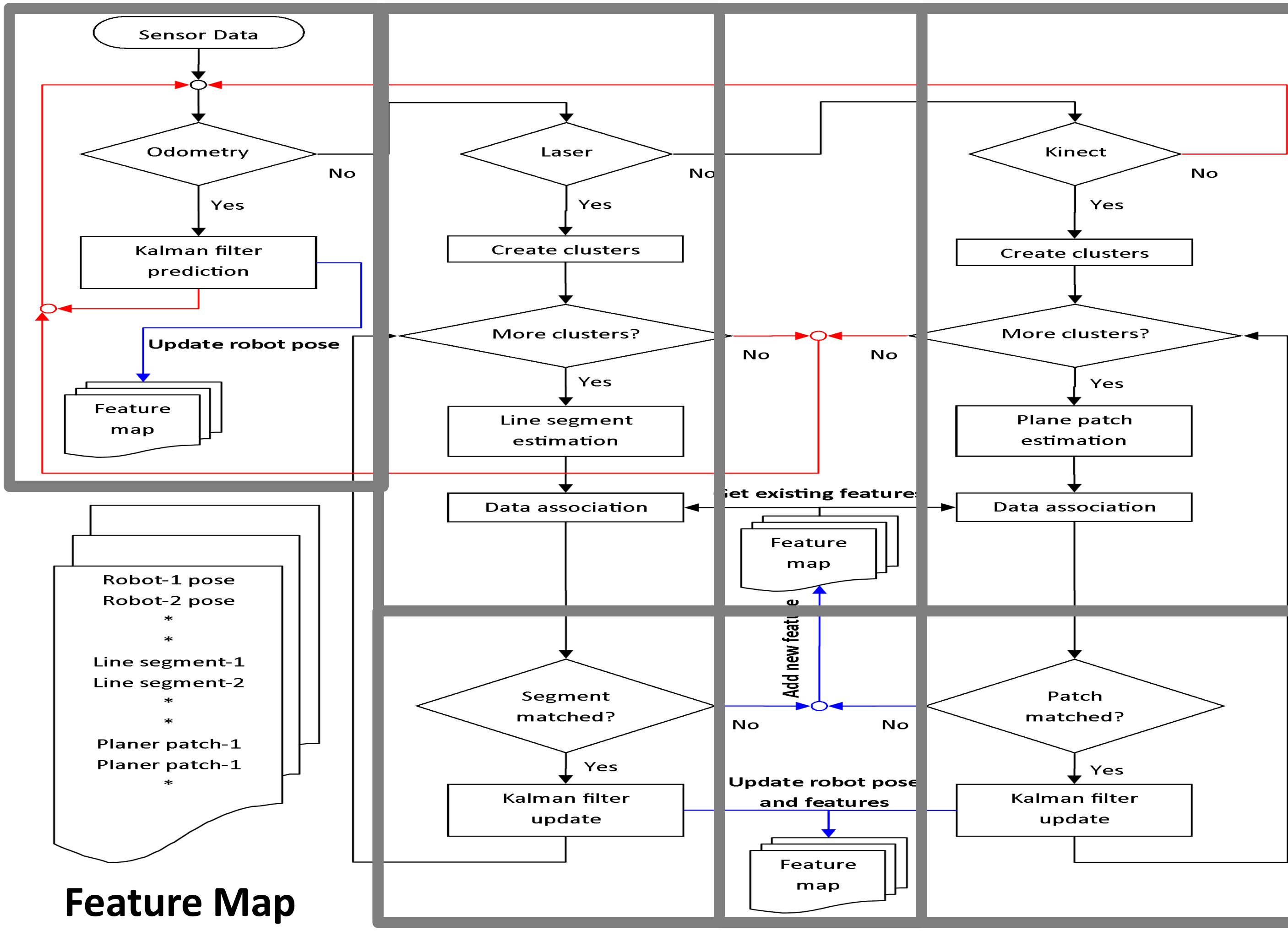
$$x = \begin{bmatrix} x_{r_1} \\ x_{r_2} \\ \vdots \\ m_{l_1} \\ \vdots \\ m_{p_1} \\ \vdots \end{bmatrix}$$

Map (robot states + features)

$$P = \begin{bmatrix} P_{r_1 r_1} & P_{r_1 r_2} & \cdots & P_{r_1 m_{l1}} & \cdots & P_{r_1 m_{p1}} & \cdots \\ P_{r_2 r_1} & P_{r_2 r_2} & \cdots & P_{r_2 m_{l1}} & \cdots & P_{r_2 m_{p1}} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots \\ P_{r_1 m_{l1}} & P_{r_2 m_{l1}} & \cdots & P_{m_{l1} m_{l1}} & \cdots & P_{m_{l1} m_{p1}} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots \\ P_{r_1 m_{p1}} & P_{r_2 m_{p1}} & \cdots & P_{m_{p1} m_{l1}} & \cdots & P_{m_{p1} m_{p1}} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots \end{bmatrix}$$

Map covariance

Methodology



Feature Map

Core CSLAM Modules

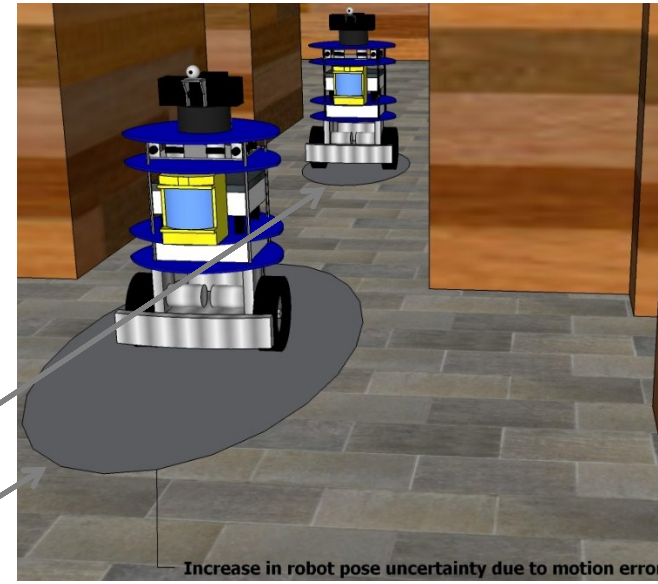
- Prediction
- Clustering/Segmentation
- Feature Extraction
- Correspondence/ Data association
- Map Update
- New Feature Augmentation
- Map Management

Prediction

$\boldsymbol{f(x_r, u_t, w_t)}$ (Robot kinematic motion model)

$\boldsymbol{m_l}$ represents all of the existing line features
$\boldsymbol{m_p}$ represents the set of all existing plane features

$\boldsymbol{F_{r1}} = \frac{\partial}{\partial x_{r_1}} f(x_r, u_t, w_t)$ Jacobian wrt. robot pose
$\boldsymbol{F_n} = \frac{\partial}{\partial w} f(x_r, u_t, w_t)$ Jacobian wrt. Noise
$\boldsymbol{Q}$ = Covariance of the noise input



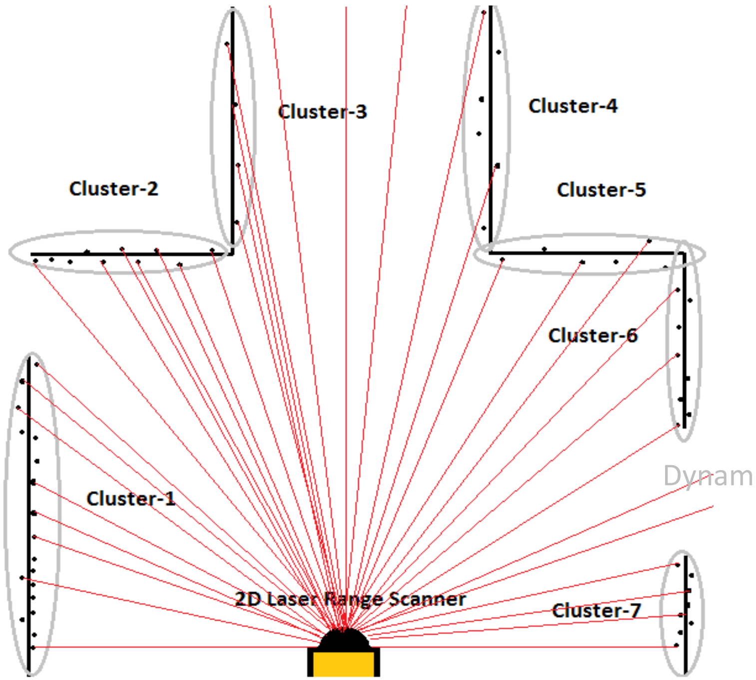
$$x_{t+1} = \begin{bmatrix} f(x_{r1}, u_t, w_t) \\ f(x_{r2}, u_t, w_t) \\ m_l \\ m_p \end{bmatrix} \quad P_{t+1} = \begin{bmatrix} F_{r_1} \cdot P_{r_1 r_1} \cdot F_{r_1}^T + F_n \cdot Q \cdot F_n^T & F_{r_1} \cdot P_{r_2} & F_{r_1} \cdot P_{r_1 m_l} & F_{r_1} \cdot P_{r_1 m_p} \\ P_{r_2} \cdot F_{r_1} & P_{r_2 r_2} & P_{r_2 m_l} & P_{r_2 m_p} \\ P_{r_1 m_l} \cdot F_{r_1} & P_{m_l r_2} & P_{m_l m_l} & P_{m_l m_p} \\ P_{r_1 m_p} \cdot F_{r_1} & P_{m_p r_2} & P_{m_p m_l} & P_{m_p m_p} \end{bmatrix}$$

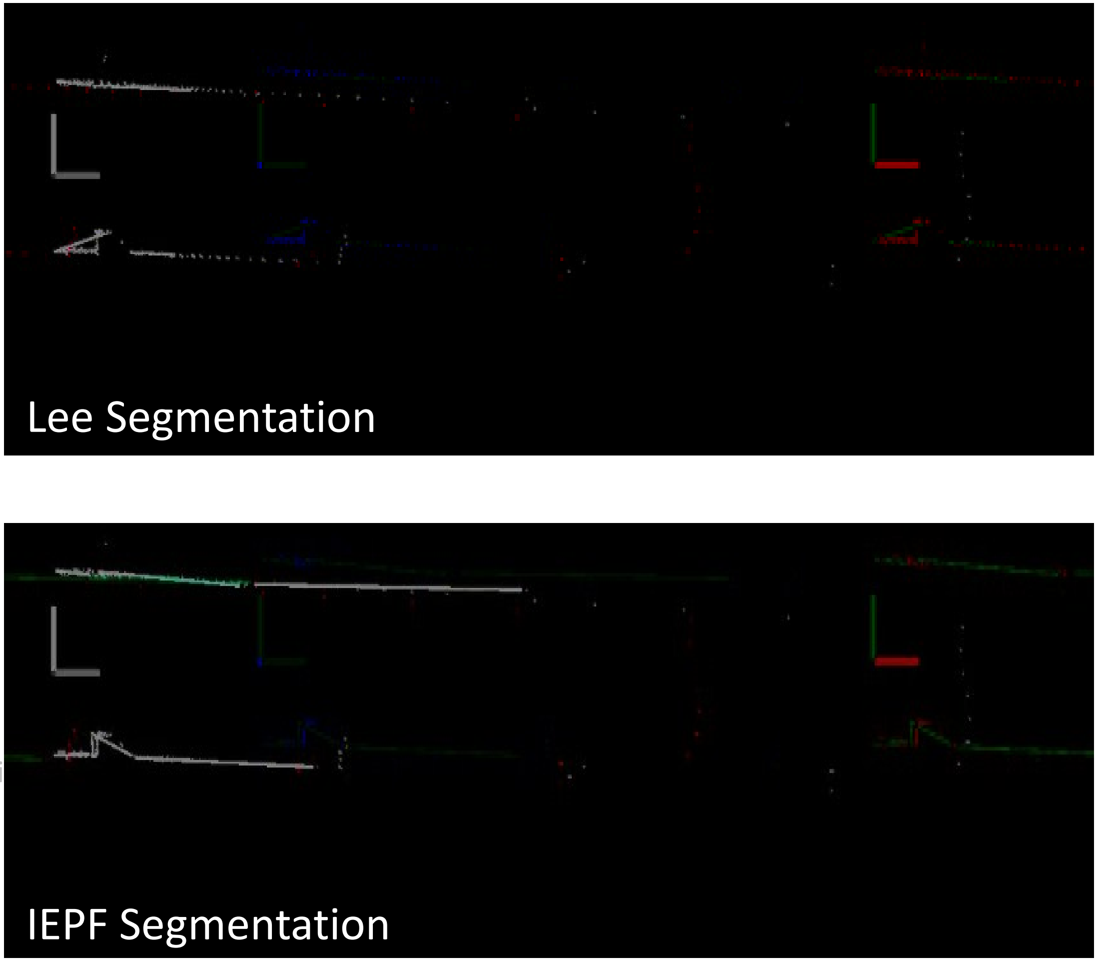
Clustering / Segmentation

$$D(r_i, r_{i+1}) = \sqrt{r_i^2 + r_{i+1}^2 - 2 r_i r_{i+1} \cos(\Delta\theta)}$$

$$D_{th} = C_0 + C_1 \min(r_i, r_{i+1})$$

$$C_1 = \sqrt{2(1 - \cos(\Delta\theta))} = \frac{D(r_i, r_{i+1})}{r_i}$$





Line Feature Extraction

$$\alpha = \frac{1}{2}\boldsymbol{atan2}\left(-2\sum_{i=0}^{n}(\bar{y}-y_i)(\bar{x}-x_i)\ ,\sum_{i=0}^{n}(\bar{y}-y_i)^2-(\bar{x}-x_i)^2\right)$$

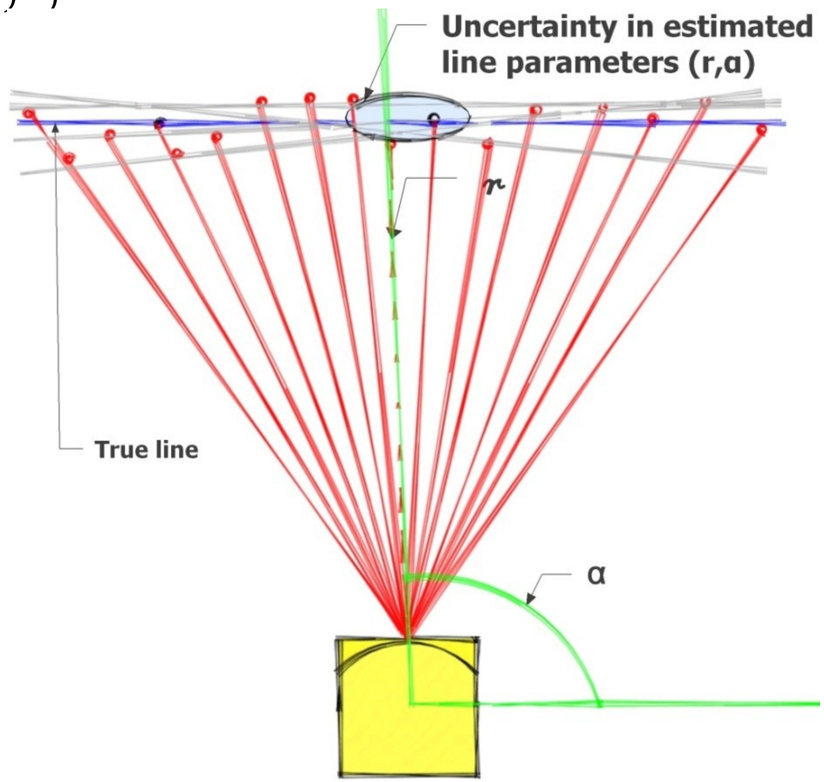
$$r = \bar{x}\cos(\alpha) + \bar{y}\sin(\alpha)$$

$$P_{\alpha r} = \begin{bmatrix} \sigma_\alpha^2 & \sigma_{\alpha r} \\ \sigma_{r\alpha} & \sigma_r^2 \end{bmatrix}$$

$$\sigma_\alpha^2 = \sum_{i=0}^{n}\left[\frac{\partial\alpha}{\partial\rho_i}\right]^2\sigma_{\rho_i}^2$$

$$\sigma_r^2 = \sum_{i=0}^{n}\left[\frac{\partial r}{\partial\rho_i}\right]^2\sigma_{\rho_i}^2$$

$$\sigma_{\alpha r} = \sigma_{r\alpha} = \sum_{i=0}^{n}\left[\frac{\partial\alpha}{\partial\rho_i}\cdot\frac{\partial r}{\partial\rho_i}\right]\cdot\sigma_{\rho_i}^2$$



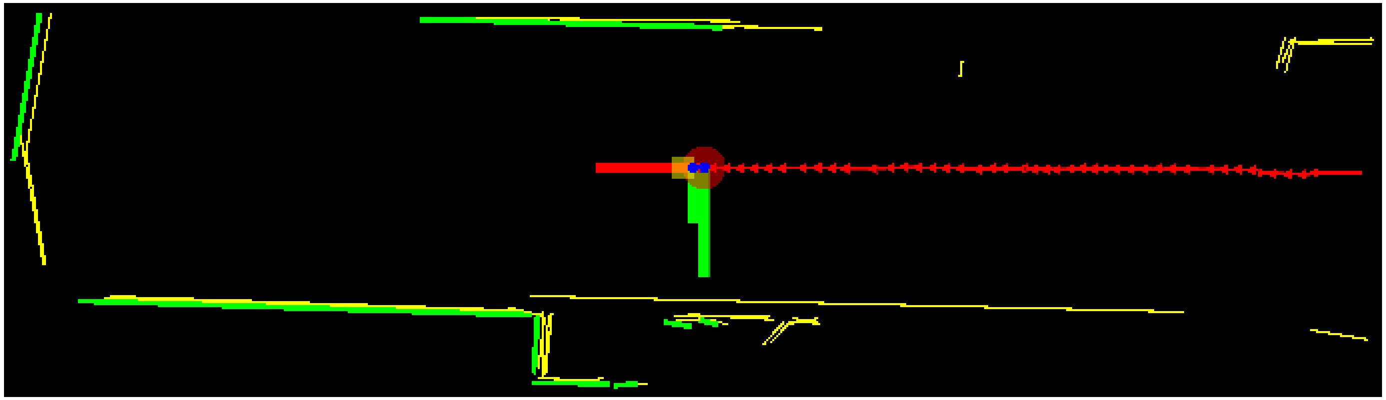
Correspondence / Data association

$\boldsymbol{z_i}$ is the innovation
$\boldsymbol{Z_i}$ is the covariance of the innovation
$\boldsymbol{n}$ is the threshold

$$z_i^{\,T} \cdot Z_i^{\,-1} \cdot z_i < n^2$$ Mahalanobis distance criterion

$$Z_i = S + R$$ S is the covariance of the expected feature

$$R = \begin{bmatrix} \sigma_r^2 & \sigma_{r\alpha} \\ \sigma_{\alpha r} & \sigma_\alpha^2 \end{bmatrix}$$ R is the covariance of the measured feature



Sensor Observation Model

The update step of the SLAM process in case of heterogeneous set of features is different for each type of feature. The line features only update the portion of the map containing the robot and line features and similarly for plane features.

$$f(x_r, y_g^i) = \begin{bmatrix} r_e^i \\ \alpha_e^i \end{bmatrix} = \begin{bmatrix} r_g^i - x_r \cdot cos(\alpha_g^i) - y_r \cdot sin(\alpha_g^i) \\ \alpha_g^i - \theta_r \end{bmatrix}$$

Sensor's observation model

$$z_i = y_m^i - y_e^i = \begin{bmatrix} r_m^i \\ \alpha_m^i \end{bmatrix} - \begin{bmatrix} r_e^i \\ \alpha_e^i \end{bmatrix}$$

New information from observed feature

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

Jacobian of sensor model wrt. robot pose

$$H_{l_i} = \begin{bmatrix} 1 & x_r \cdot sin(\alpha_e) - y \cdot cos(\alpha_e) \\ 0 & 1 \end{bmatrix}$$

Jacobian of sensor model wrt. feature

Map Update

$Z_i = S_j + R_i$ Covariance of the innovation

$$R = \begin{bmatrix} \sigma_r^2 & \sigma_{r\alpha} \\ \sigma_{\alpha r} & \sigma_\alpha^2 \end{bmatrix}$$

Measure feature covariance

$$K_{p_i} = \begin{bmatrix} P_{rr} & P_{rp_1} \\ P_{p_1 r} & P_{p_1 p_i} \\ \vdots & \vdots \\ P_{p_n r} & P_{p_n p_i} \end{bmatrix} \cdot \begin{bmatrix} H_r^T \\ H_{p_i}^T \end{bmatrix} \cdot [Z_i]^{-1}$$

Kalman gain

$x = x + K_{p_i} \cdot z_i$ Map update

$P = P - K_{p_i} \cdot Z_i \cdot {K_{p_i}}^T$ Map uncertainty reduced

Inverse sensor Observation Model

$$f\left(x_r, y_l^{n+1}\right) = \begin{bmatrix} r_g^{n+1} \\ \alpha_g^{n+1} \end{bmatrix} = \begin{bmatrix} r_l^{n+1} + x_r \cdot \cos\left(\alpha_l^{n+1} + \theta_r\right) + y_r \cdot \sin\left(\alpha_l^{n+1} + \theta_r\right) \\ \alpha_l^{n+1} + \theta_r \end{bmatrix}$$

$$P = \begin{bmatrix} P_{rr} & P_{rm} & P_{rr}^T \cdot Y_r^T \\ P_{rm} & P_{mm} & P_{rm}^T \cdot Y_r^T \\ Y_r \cdot P_{rr} & Y_r \cdot P_{rm} & Y_r \cdot P_{rr} \cdot Y_r^T + Y_{l_{n+1}} \cdot R \cdot Y_{l_{n+1}}^T \end{bmatrix}$$

Covariance of new feature

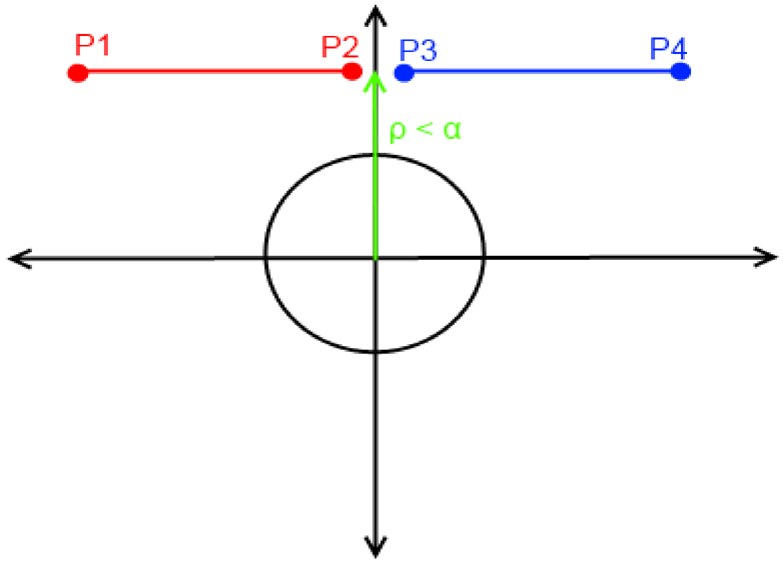
$$Y_r = \begin{bmatrix} \cos\left(\alpha_l^{n+1} + \theta_r\right) & \sin\left(\alpha_l^{n+1} + \theta_r\right) & y_r \cdot \cos\left(\alpha_l^{n+1} + \theta_r\right) - x_r \cdot \sin\left(\alpha_l^{n+1} + \theta_r\right) \\ 0 & 0 & 1 \end{bmatrix}$$

$$Y_{l_{n+1}} = \begin{bmatrix} 1 & y_r \cdot \cos\left(\alpha_l^{n+1} + \theta_r\right) - x_r \cdot \sin\left(\alpha_l^{n+1} + \theta_r\right) \\ 0 & 1 \end{bmatrix}$$

Map Management

State Vector (Map)

$$\begin{bmatrix} Robot_1[x\ y\ \theta]^T \\ Robot_2[x\ y\ \theta]^T \\ LineFeature_1[r\ \alpha\ x_1\ y_1\ x_2\ y_2]^T \\ \vdots \\ LineFeature_n[r\ \alpha\ x_1\ y_1\ x_2\ y_2]^T \end{bmatrix}$$

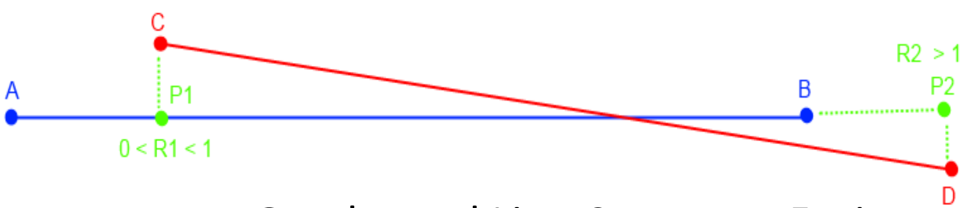


Line-Segments Fusion

$$AC = (C_x - A_x\, , C_y - A_y)$$

$$BA = (B_x - A_x\, , B_y - A_y)$$

$$R_1 = \frac{AC_x \cdot AB_x + AC_y \cdot AB\_y}{AB_{x^2} + AB_{y^2}}$$



Overlapped Line-Segments Fusion

Summary

- Mapping
  - Feature mapping
  - Grid Mapping
- Introduction to SLAM
- Feature/Landmark SLAM
- Grid Mapping (GMapping)

Questions