SLAM Algorithm

Simultaneous Localization And Mapping

Albin Frischenschlager, 0926427

December 17, 2013

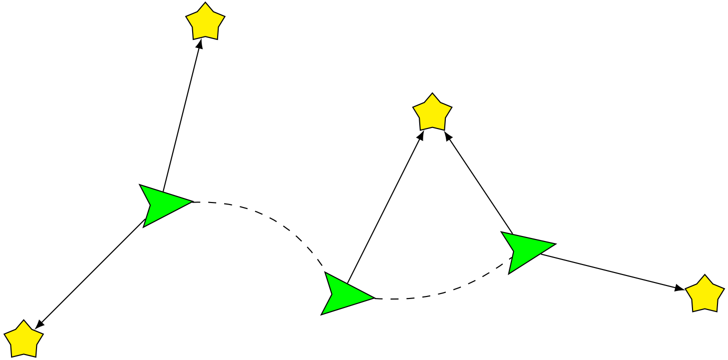
Outline

What is ...

- **Robot** - a device, that moves through the environment
- **Landmark** - characteristic, reobservable point in the environment
- **Localization** - estimating the robot's location
- **Mapping** - building a map

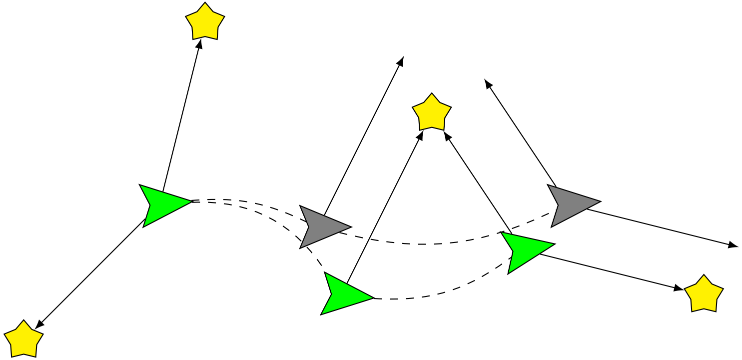
Localization Example

- Estimate the robot's pose given landmarks



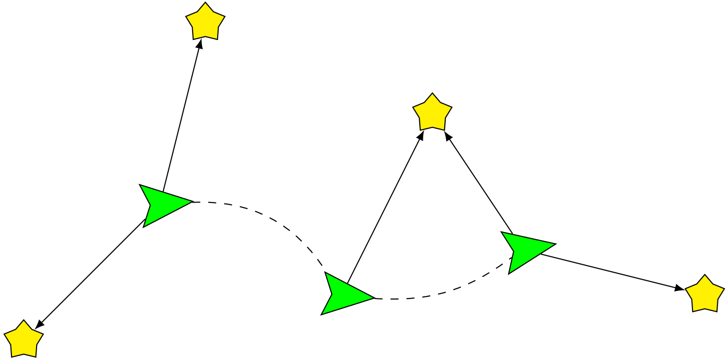
Localization Example

- Estimate the robot's pose given landmarks



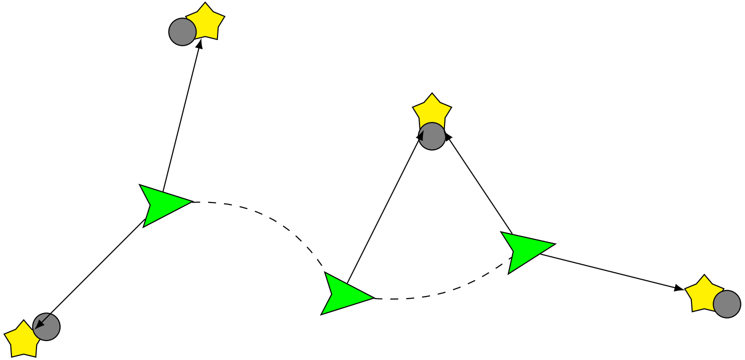
Mapping Example

- Estimate the landmarks given the robot's poses



Mapping Example

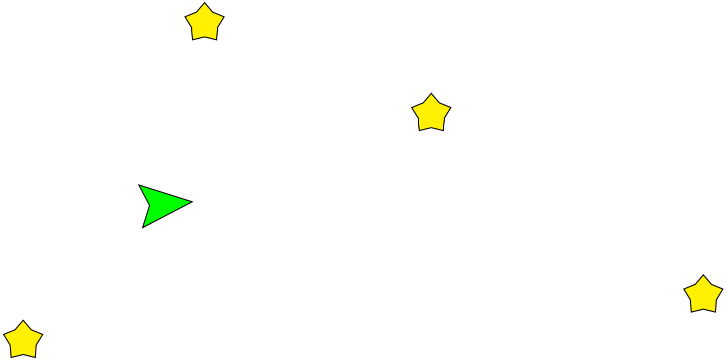
- Estimate the landmarks given the robot's poses



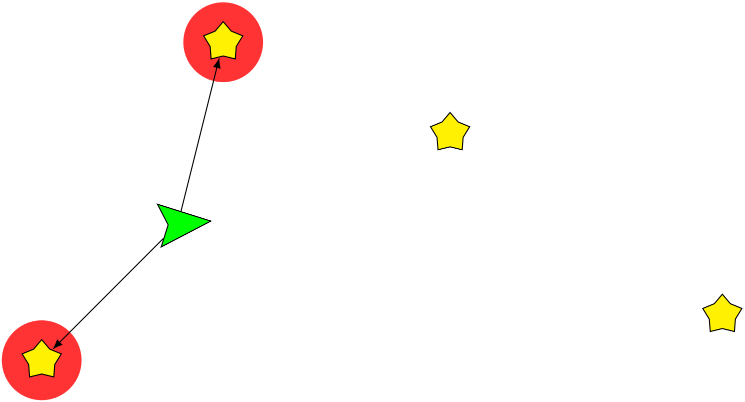
What is SLAM?

- **SLAM** - Computing the robot's pose and the map of the environment at the same time $\rightarrow$ i.e. do localization and mapping simultaneously
- it's a chicken-or-egg problem:
  - a map is needed for localization and
  - a pose is needed for mapping
- it's a hard problem $\rightarrow$ map and pose estimates correlate
- it's a important problem

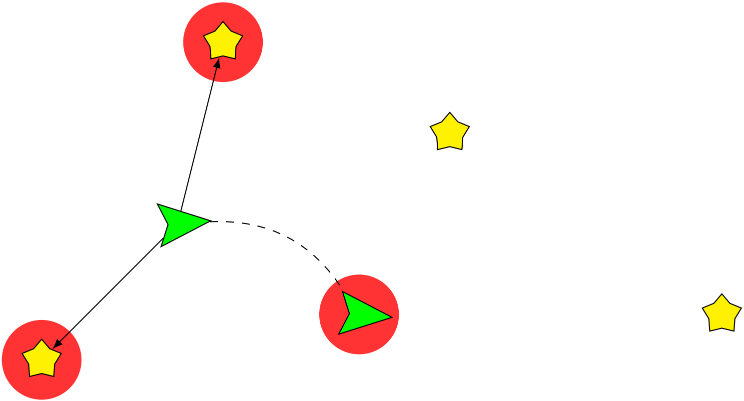
SLAM Example



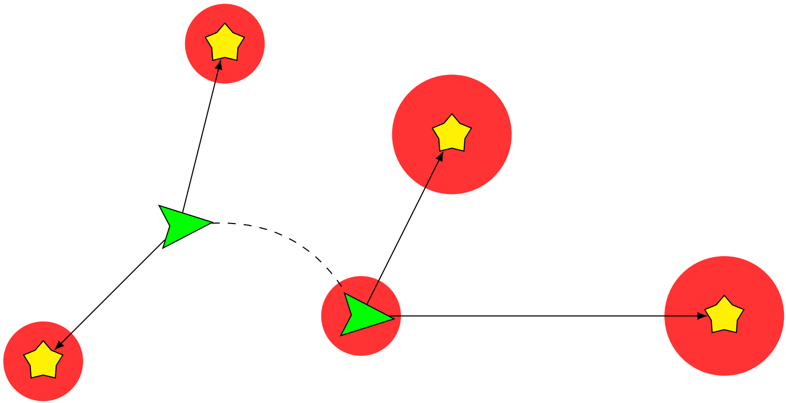
SLAM Example



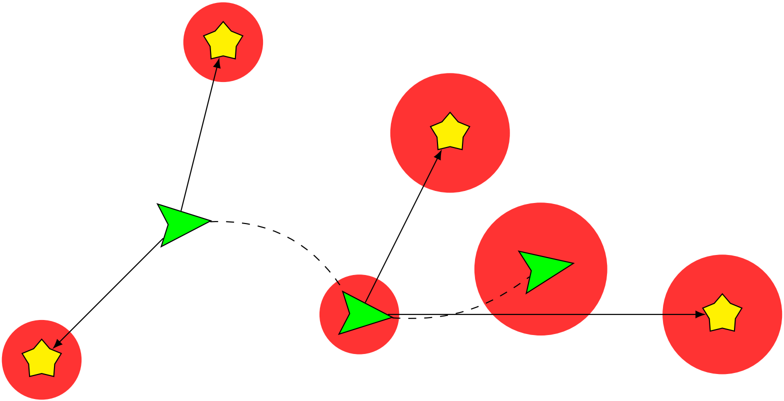
SLAM Example



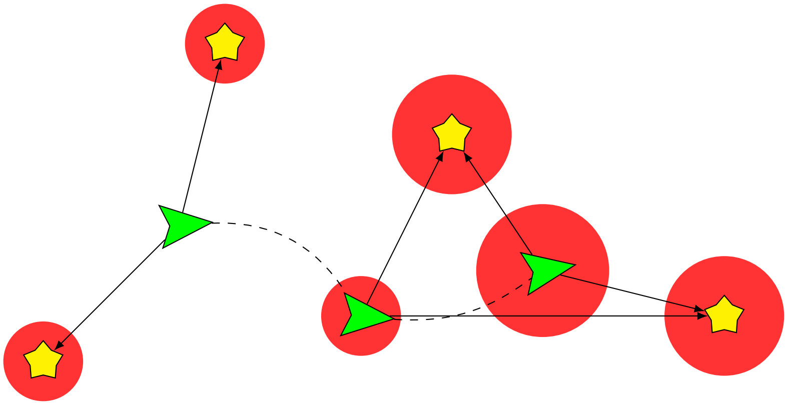
SLAM Example



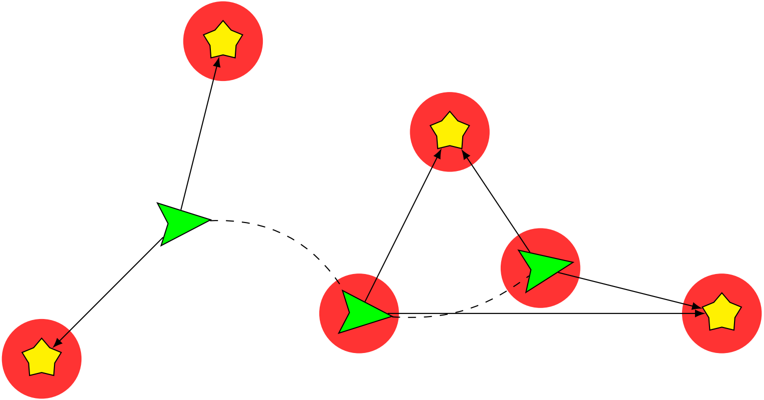
SLAM Example



SLAM Example



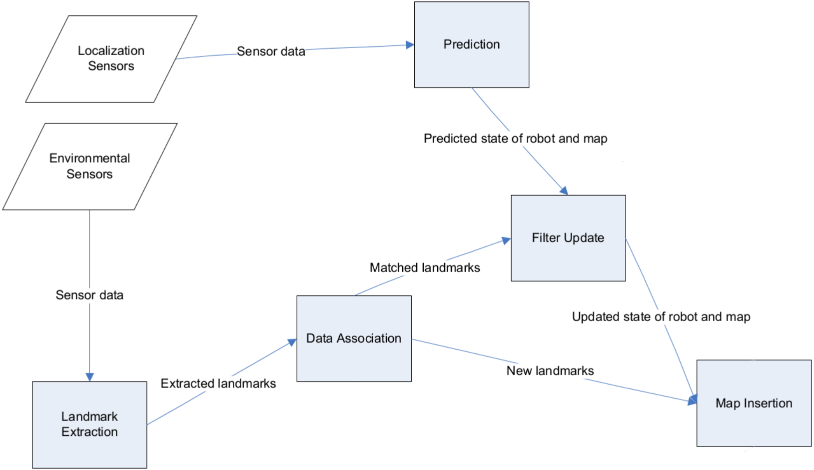
SLAM Example



SLAM Algorithm

- There isn't 'the' SLAM algorithm
- SLAM is just a problem, but luckily there a possibilities to solve it

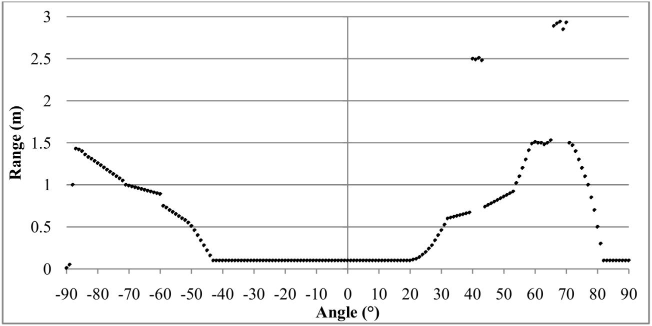
SLAM flowchart



Landmark Extraction

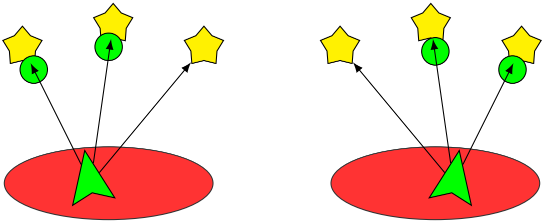
- Extract from the environmental sensors characteristic points
- Input can be a camera image, array of measurements, ...
- Algorithms for array of measurements:
  - Spike
  - RANSAC (Random Sampling Consensus)
  - Scan-Matching
  - Geometric polygon extraction

Landmark Extraction



Data Association

- Matching observed landmarks with those previously stored in the map
- Wrong association can have catastrophic consequences (divergence)
- (gated) Nearest-neighbor approach using Euclidean distance or Mahalanobis distance



Prediction and Filter Update

- Kalman Filter
  - Extended Kalman Filter
  - Information Filter
  - Unscented Kalman Filter
  - Sparse Extended Information Filter
- Particle Filter

Extended Kalman Filter

- Estimate state of a (non-linear) dynamic system, given:
  - model of the system
  - control inputs
  - model of the sensors
  - meassurements with noise from the sensors
- Set of mathematical equations in a recursive fashion
- Two steps:
  - Prediction
  - Correction

EKF in SLAM

- System state: $\bar{\mathbf{x}} = \begin{bmatrix} \bar{\mathcal{R}} \\ \bar{\mathcal{M}} \end{bmatrix} = \begin{bmatrix} \bar{\mathcal{R}} \\ \bar{\mathcal{L}}_1 \\ \vdots \\ \bar{\mathcal{L}}_n \end{bmatrix}$
- Covariances matrix:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathcal{RR}} & \mathbf{P}_{\mathcal{RM}} \\ \mathbf{P}_{\mathcal{MR}} & \mathbf{P}_{\mathcal{MM}} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{\mathcal{RR}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{R}\mathcal{L}_n} \\ \mathbf{P}_{\mathcal{L}_1\mathcal{R}} & \mathbf{P}_{\mathcal{L}_1\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{L}_1\mathcal{L}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{\mathcal{L}_n\mathcal{R}} & \mathbf{P}_{\mathcal{L}_n\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{L}_n\mathcal{L}_n} \end{bmatrix}$$

- Goal: keep the map $\bar{\mathbf{x}}$, $\mathbf{P}$ up to date at all times

Prediction

- Movement function (part of the system model): $\mathbf{x} \leftarrow f(\mathbf{x}, \mathbf{u}, \mathbf{n})$
  - $\mathbf{u} \ldots$ control vector
  - $\mathbf{n} \ldots$ pertubation vector
  - $p(\mathbf{n}) \sim N(0, \mathbf{Q})$
- EKF prediction step in SLAM:

$$\bar{\mathcal{R}} \leftarrow f_{\mathcal{R}}(\bar{\mathcal{R}}, \mathbf{u}, 0)$$

$$\mathbf{P}_{\mathcal{RR}} \leftarrow \frac{\partial f_{\mathcal{R}}}{\partial \mathcal{R}} \mathbf{P}_{\mathcal{RR}} \frac{\partial f_{\mathcal{R}}}{\partial \mathcal{R}}^{\mathsf{T}} + \frac{\partial f_{\mathcal{R}}}{\partial \mathbf{n}} \mathbf{Q} \frac{\partial f_{\mathcal{R}}}{\partial \mathbf{n}}^{\mathsf{T}}$$

$$\mathbf{P}_{\mathcal{RM}} \leftarrow \frac{\partial f_{\mathcal{R}}}{\partial \mathcal{R}} \mathbf{P}_{\mathcal{RM}}$$

$$\mathbf{P}_{\mathcal{MR}} \leftarrow \mathbf{P}_{\mathcal{RM}}^{\mathsf{T}}$$

Correction

- Observation function (part of the sensor model): $\mathbf{y} = h(\mathbf{x}) + \mathbf{v}$
  - $\mathbf{v}\ldots$ meassurement noise
  - $p(\mathbf{v}) \sim N(0, \mathbf{R})$
- EKF correction step in SLAM for every landmark $\mathcal{L}_i$:

$$\bar{\mathbf{z}} = \mathbf{y}_i - h_i(\bar{\mathcal{R}}, \bar{\mathcal{L}}_i) \text{ 'innovation'}$$

$$\mathbf{Z} = \begin{bmatrix} \mathbf{H}_{\mathcal{R}} & \mathbf{H}_{\mathcal{L}_i} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_i} \\ \mathbf{P}_{\mathcal{L}_i\mathcal{R}} & \mathbf{P}_{\mathcal{L}_i\mathcal{L}_i} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{\mathcal{R}}^{\mathsf{T}} \\ \mathbf{H}_{\mathcal{L}_i}^{\mathsf{T}} \end{bmatrix} + \mathbf{R}$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_i} \\ \mathbf{P}_{\mathcal{L}_i\mathcal{R}} & \mathbf{P}_{\mathcal{L}_i\mathcal{L}_i} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{\mathcal{R}}^{\mathsf{T}} \\ \mathbf{H}_{\mathcal{L}_i}^{\mathsf{T}} \end{bmatrix} \mathbf{Z}^{-1} \text{ 'Kalman gain'}$$

$$\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}} + \mathbf{K}\bar{\mathbf{z}}$$

$$\mathbf{P} \leftarrow \mathbf{P} - \mathbf{K}\mathbf{Z}\mathbf{K}^{\mathsf{T}}$$

  - $\mathbf{H}_{\mathcal{R}} = \frac{\partial h_i(\bar{\mathcal{R}}, \bar{\mathcal{L}}_i)}{\partial \mathcal{R}}$, $\mathbf{H}_{\mathcal{L}_i} = \frac{\partial h_i(\bar{\mathcal{R}}, \bar{\mathcal{L}}_i)}{\partial \mathcal{L}_i}$

Map Insertion

$$\mathcal{L}_{n+1} = g(\bar{\mathcal{R}}, \mathbf{y}_{n+1})$$

$$\mathbf{G}_{\mathcal{R}} = \frac{\partial g(\bar{\mathcal{R}}, \mathbf{y}_{n+1})}{\partial \mathcal{R}}$$

$$\mathbf{G}_{\mathbf{y}_{n+1}} = \frac{\partial g(\bar{\mathcal{R}}, \mathbf{y}_{n+1})}{\partial \mathbf{y}_{n+1}}$$

$$\mathbf{P}_{\mathcal{L}\mathcal{L}} = \mathbf{G}_{\mathcal{R}} \mathbf{P}_{\mathcal{R}\mathcal{R}} \mathbf{G}_{\mathcal{R}}^{\top} + \mathbf{G}_{\mathbf{y}_{n+1}} \mathbf{R} \mathbf{G}_{\mathbf{y}_{n+1}}^{\top}$$

$$\mathbf{P}_{\mathcal{L}\mathbf{x}} = \mathbf{G}_{\mathcal{R}} \mathbf{P}_{\mathcal{R}\mathbf{x}} = \mathbf{G}_{\mathcal{R}} \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{M}} \end{bmatrix}$$

$$\bar{\mathbf{x}} \leftarrow \begin{bmatrix} \bar{\mathbf{x}} \\ \mathcal{L}_{n+1} \end{bmatrix}$$

$$\mathbf{P} \leftarrow \begin{bmatrix} \mathbf{P} & \mathbf{P}_{\mathcal{L}\mathbf{x}}^{\top} \\ \mathbf{P}_{\mathcal{L}\mathbf{x}} & \mathbf{P}_{\mathcal{L}\mathcal{L}} \end{bmatrix}$$

Sources

- SLAM for Dummies by Søren Riisgaard and Morten Rufus Blas
- A simultaneous localization and mapping implementation using inexpensive hardware by Todd Michael Aycock
- Online Video: http://www.mrpt.org/tutorials/slam-algorithms/slam-simultaneous-localization-and-mapping-for-beginners-the-basics
- Online Course: http://www.joansola.eu/JoanSola/eng/course.html

Thank you for your attention!