# 7630 – Autonomous Robotics

## Introduction to Mapping

Cédric Pradalier

March $18^{th}$, 2013

# Objectives

## Mapping

- Occupancy grids
- Relative mapping
- Kalman-Filter-based mapping

## Later lecture: SLAM

- SLAM = Simultaneous Localisation and Mapping
- EKF-SLAM
- FastSLAM: Particle filter for SLAM
- MonoSLAM: SLAM with a single camera

# Outline

# Bayesian Fusion

## Assumptions

- A variable $X$ is being estimated
- A set of observations $\{Z_i\}$ correlated with $X$ are being made.

$$P(X\, Z_1\, \ldots\, Z_n) = P(X) \prod_{i=1}^{n} P(Z_i|X) \tag{1}$$

Bayesian inference:

$$P(X|Z_1\, \ldots\, Z_n) \propto P(X) \prod_{i=1}^{n} P(Z_i|X) \tag{2}$$

# Bayesian Fusion: Gaussians

## Assumptions

- All distributions are Gaussian: $P(X) \to \mathcal{N}(\mu_0, \Sigma_0)$, $P(Z_i|X) \to \mathcal{N}(\nu_i, \sigma_i)$.

$$P(X|Z_1 \ldots Z_n) \propto P(X) \prod_{i=1}^{n} P(Z_i|X) \tag{3}$$

- Hence $P(X|Z_1 \ldots Z_n)$ is Gaussian $\mathcal{N}(\mu_n, \Sigma_n)$.

Bayesian inference:

$$\log P(X|Z_1 \ldots Z_n) = \log P(X) + \sum_{i=1}^{n} \log P(Z_i|X) + Cst$$

# Bayesian Fusion: Gaussians

## Assumptions

- $P(X) \rightarrow \mathcal{N}(\mu_0, \Sigma_0)$, $P(Z_i|X) \rightarrow \mathcal{N}(\nu_i = h(x), \sigma_i)$.

Bayesian inference:

$$\begin{aligned}
\log P(X|Z_1 \ldots Z_n) &= \log P(X) + \sum_{i=1}^{n} \log P(Z_i|X) + Cst \\
(\mu_n - x)^T \Sigma_n^{-1} (\mu_n - x) &= (\mu_0 - x)^T \Sigma_0^{-1} (\mu_0 - x) \\
&+ \sum_{i=1}^{n} (\nu_i - h(x))^T \sigma_i^{-1} (\nu_i - h(x)) + Cst
\end{aligned}$$

# Bayesian Fusion Special Cases

$h(x) = x$ and $n = 1$

$$\begin{aligned}
&(\mu_1 - x)^T \Sigma_1^{-1} (\mu_1 - x) \\
&\quad = \quad (\mu_0 - x)^T \Sigma_0^{-1} (\mu_0 - x) + (\nu_1 - x)^T \sigma_1^{-1} (\nu_1 - x) + Cst
\end{aligned}$$

$$\begin{aligned}
&\mu_1^T \Sigma_1^{-1} \mu_1 + 2\mu_1^T \Sigma_1^{-1} x + x^T \Sigma_1^{-1} x \\
&\quad = \quad \mu_0^T \Sigma_0^{-1} \mu_0 + 2\mu_0^T \Sigma_0^{-1} x + x^T \Sigma_0^{-1} x \\
&\quad + \quad \nu_1^T \sigma_1^{-1} \nu_1 + 2\nu_1^T \sigma_1^{-1} x + x^T \sigma_1^{-1} x + Cste
\end{aligned}$$

Hence

$$\begin{aligned}
\Sigma_1^{-1} &= \Sigma_0^{-1} + \sigma_1^{-1} \\
\mu_1 &= \Sigma_1 \left( \Sigma_0^{-1} \mu_0 + \sigma_1^{-1} \nu_1 \right)
\end{aligned}$$

# Bayesian Fusion Special Cases

$h(x) = x$ and $n = 1$

$$\begin{aligned} \Sigma_1^{-1} &= \Sigma_0^{-1} + \sigma_1^{-1} \\ \mu_1 &= \Sigma_1 \left( \Sigma_0^{-1} \mu_0 + \sigma_1^{-1} \nu_1 \right) \end{aligned}$$

Can be rewritten as:

$$\begin{aligned} K &= \Sigma_0 \left( \Sigma_0 + \sigma_1 \right)^{-1} \\ \Sigma_1 &= (I - K)\Sigma_0 \\ \mu_1 &= \mu_0 + K \left( \nu_1 - \mu_0 \right) \end{aligned}$$

# Bayesian Fusion Special Cases

$h(x) = Hx$ and $n = 1$

$$
\begin{aligned}
K &= \Sigma_0 H^T \left(H\Sigma_0 H^T + \sigma_1\right) \\
\Sigma_1 &= (I - KH)\Sigma_0 \\
\mu_1 &= \mu_0 + K\left(\nu_1 - H\mu_0\right)
\end{aligned}
$$

$h(x)$ non linear and $n = 1$

$$
\begin{aligned}
H &= \frac{\partial h}{\partial x} \longrightarrow \text{Jacobian} \\
K &= \Sigma_0 H^T \left(H\Sigma_0 H^T + \sigma_1\right)^{-1} \\
\Sigma_1 &= (I - KH)\Sigma_0 \\
\mu_1 &= \mu_0 + K\left(\nu_1 - H\mu_0\right)
\end{aligned}
$$

# Bayesian Fusion Special Cases

## Maximum A Posteriori (MAP)

- We are just interested in the maximum of $P(X|Z_1 \ldots X_n)$, i.e. $\mu_n$.

$$\begin{aligned}
\arg\max_X P(X|Z_1 \ \ldots \ Z_n) &= \arg\min_X \log P(X|Z_1 \ \ldots \ Z_n) \\
&= \arg\min_X (\mu_0 - x)^T \Sigma_0^{-1} (\mu_0 - x) \\
&+ \sum_{i=1}^{n} (\nu_i - h(x))^T \sigma_i^{-1} (\nu_i - h(x))
\end{aligned}$$

- Weighted least-square minimisation (or regression), non-linear if h is non-linear.

# Recursive Bayesian Fusion

## Assumptions

- A variable $X$ is being estimated
- A set of observations $\{Z_i\}$ correlated with $X$ are being made over time.
- We denote $Bel_k(X) = P(X|Z_1 \ldots Z_k)$.

Bayesian inference:

$$
\begin{aligned}
Bel_k(X) &= P(X|Z_1 \ldots Z_k) \propto P(X) \prod_{i=1}^{k} P(Z_i|X) \\
&= \left[ P(X) \prod_{i=1}^{k-1} P(Z_i|X) \right] P(Z_k|X) \\
&= P(Z_k|X) \cdot Bel_{k-1}(X)
\end{aligned}
$$

# Recursive Bayesian Fusion, with Gaussian Assumption

## Assumptions

$$Bel_k(X) = P(Z_k|X) \cdot Bel_{k-1}(X)$$

- $Bel_k$ is Gaussian $\rightarrow \mathcal{N}(x_k, P_k)$
- $P(Z_k|X)$ is Gaussian $\rightarrow \mathcal{N}(h(X), R)$

This is Gaussian Bayesian fusion with $n = 1$:

$$\begin{aligned}
H &= \frac{\partial h}{\partial x} \longrightarrow \text{Jacobian} \\
K &= P_{k-1}H^T \left(HP_{k-1}H^T + R\right)^{-1} \\
P_k &= (I - KH)P_{k-1} \\
x_k &= x_{k-1} + K\left(z_k - Hx_{k-1}\right)
\end{aligned}$$

# Exercise

## Problem

Measure the position of a WiFi router (2D) from range measurement $z_i$ taken from a set of positions $p_i$. The WiFi range measurements are expected to have a $10m$ precision.

## Questions

- What is the estimated state?
- Using a Gaussian assumption, what are the update equations?
- What is the precision?
- How is the precision of the belief evolving with the number of observations? Can in worsen?

# Outline

# Outline

# Mapping using Occupancy Grids

## Principle

- Discrete representation of the map as an array of cell.
- In each cell, compute the likelihood of occupancy.
- Requires localisation, but no data association.
- Well suited for time-of-flight range sensors, as they give information about occupied and free space.

# Mapping using Occupancy Grids

Let's work on the following paper:

- Efficient GPU-based Construction of Occupancy Grids Using several Laser Range-finders M. Yguel, O. Aycard, C. Laugier. Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems (2006)

Also...

- `http://hal.inria.fr/docs/00/18/20/08/PDF/egcog.pdf`
- And many others...

# Occupancy Grids

Questions:

- What is the update law?
- What are log quotients and how do they help?
- Why do we need ray-tracing algorithms?

# Outline

# Mapping using Relative Maps

## Principle

- Observe and estimate parameters independent of the robot pose.
- For ground robots: distances and angles
- The theory of invariants predict the number and dimensionality of observable given an observation process.
- Does not require localisation, but helps for data association.

# Mapping using Relative Maps

Papers:

- New approach to map building using relative position estimates. Michael Csorba ; Hugh F. Durrant-Whyte Proc. SPIE 3087, Navigation and Control Technologies for Unmanned Systems II, 115 (June 26, 1997)

Also in the article package:

- Invariant filtering for simultaneous localization and mapping. Deans, Matthew C. ; Hebert, Martial H. IEEE International Conference on Robotics and Automation, 2000.
- Simultaneous localization and mapping using the geometric projection filter and correspondence graph matching. C. Pradalier, S. Sekhavat, Advanced Robotics, 17 (7), 675-690, 2003

# Relative Maps: Challenges

Questions:

- Initialization?
- Building a cartesian representation?

# Outline

# Mapping using Kalman Filters

## Map representation

- List of objects described by parameters.
- Examples: point landmark $(x, y)$ or $(x, y, z)$, lines, planes, ...

## Kalman filtering: hypothesis

- Known robot/sensor position
- Gaussian noise
- No prediction step (or no uncertainty in the prediction step)
- Only perception step: $P(Z|L\,X)$

# Observation model

## Conditional Independence

- Assuming data association:

$$P(Z_1 \dots Z_p | L_1 \dots L_n X) = \prod_{i=1}^{p} P(Z_i | L_{k(i)} X) \tag{4}$$

- leads to $n$ kalman filter, one for each landmark.

# Homework: Step 3

Objectives:

- Same environment than for the (future) localisation homework.
- Assuming the robot position is known (TF), build a map of the landmarks.

Challenges:

- Initialization.
- Is it a false observation or a new landmark?
- Has this landmark disappeared?
- Data association (in general).

# Homework: Step 1 & 2

Objectives:

- Experiment with out-of-the-box mapping solutions
- Actually running SLAM (but in simulation you can check the difference between Mapping and SLAM).
- Test on TurtleBots

Challenges:

- Integration of existing blocks...

# Recursive Bayesian Fusion

## Assumptions

- A variable $X$ is being estimated
- A set of observations $\{Z_i\}$ correlated with $X$ are being made over time.
- We denote $Bel_k(X) = P(X|Z_1 \ldots Z_k)$.

Bayesian inference:

$$\begin{aligned} Bel_k(X) &= P(X|Z_1 \ldots Z_k) \propto P(X) \prod_{i=1}^{k} P(Z_i|X) \\ &= \left[ P(X) \prod_{i=1}^{k-1} P(Z_i|X) \right] P(Z_k|X) \\ &= P(Z_k|X) \cdot Bel_{k-1}(X) \end{aligned}$$

# Recursive Bayesian Fusion, Gaussian Assumption

## Assumptions

$$Bel_k(X) = P(Z_k|X) \cdot Bel_{k-1}(X)$$

- $Bel_k$ is Gaussian $\rightarrow \mathcal{N}(x_k, P_k)$
- $P(Z_k|X)$ is Gaussian $\rightarrow \mathcal{N}(h(X), R)$

This is Gaussian Bayesian fusion with $n = 1$:

$$\begin{aligned} H &= \frac{\partial h}{\partial x} \longrightarrow \text{Jacobian} \\ K &= P_{k-1}H^T \left(HP_{k-1}H^T + R\right)^{-1} \\ P_k &= (I - KH)P_{k-1} \\ x_k &= x_{k-1} + K\left(z_k - Hx_{k-1}\right) \end{aligned}$$

The update stage of the Kalman filter (more later)

# Recursive Bayesian Filter

## Assumptions

- A variable $X_k$ is being estimated at time $k$
- A set of observations $\{Z_i\}$ correlated with $X$ are being made over time.
- A model of changes of $X_k$ is available as $P(X_k|X_{k-1}, U_{k-1})$.
- We denote $Bel_k(X_k) = P(X_k|Z_1 \ldots Z_k, U_1 \ldots U_k) = P(X_k|\mathcal{Z}_k, \mathcal{U}_k)$.

# Recursive Bayesian Filter

Bayesian inference:

$$
\begin{aligned}
Bel_k(X_k) &= P(X_k|\mathcal{Z}_k,\mathcal{U}_k) \propto P(Z_k|X_k\mathcal{U}_k,\mathcal{Z}_{k-1})P(X_k|\mathcal{U}_k,\mathcal{Z}_{k-1}) \\
&= P(Z_k|X_k)P(X_k|\mathcal{U}_k,\mathcal{Z}_{k-1}) \\
&= P(Z_k|X_k)\int_{X_{k-1}} P(X_k|\mathcal{U}_k,\mathcal{Z}_{k-1},X_{k-1})P(X_{k-1}|\mathcal{U}_k,\mathcal{Z}_{k-1}) \\
&= P(Z_k|X_k)\int_{X_{k-1}} P(X_k|U_{k-1},X_{k-1})P(X_{k-1}|\mathcal{U}_{k-1},\mathcal{Z}_{k-1}) \\
&= P(Z_k|X_k)\int_{X_{k-1}} P(X_k|U_{k-1},X_{k-1})Bel_{k-1}(X_{k-1})
\end{aligned}
$$

Important: know how to derive that.

# Recursive Bayesian Filter: Kalman Filter

Assumptions

$$
\begin{aligned}
Bel_{k-1}(X_k) &= \int_{X_{k-1}} P(X_k|X_{k-1}) Bel_{k-1}(X_{k-1}) \\
Bel_k(X_k) &= P(Z_k|X) \cdot Bel_{k-1}(X_k)
\end{aligned}
$$

- $Bel_k$ is Gaussian $\rightarrow \mathcal{N}(x_k, P_k)$
- $P(X_k|X_{k-1})$ is Gaussian $\rightarrow \mathcal{N}(f(X), Q)$
- $P(Z_k|X_k)$ is Gaussian $\rightarrow \mathcal{N}(h(X), R)$

# Kalman Filter: Prediction Stage

## Assumptions

$$Bel_{k-1}(X_k) = \int_{X_{k-1}} P(X_k|X_{k-1}) Bel_{k-1}(X_{k-1} \rightarrow \mathcal{N}(\bar{x}_k, \bar{P}_k)$$

- $P(X_k|X_{k-1})$ is Gaussian $\rightarrow \mathcal{N}(f(X_{k-1}, U_{k-1}), Q)$

$$\begin{aligned}
A &= \frac{\partial f}{\partial x} \longrightarrow \text{Jacobian} \\
B &= \frac{\partial f}{\partial u} \longrightarrow \text{Jacobian} \\
\bar{x}_k &= f(x_{k-1}, u_{k-1}) \\
\bar{P}_k &= AP_{k-1}A^T + (BQ_uB^T) + Q
\end{aligned}$$

# Kalman Filter, Observation stage

## Assumptions

$$Bel_k(X_k) = P(Z_k|X_k) \cdot Bel_{k-1}(X_k)$$

- $Bel_{k-1}(X_k)$ is Gaussian $\rightarrow \mathcal{N}(\bar{x}_k, \bar{P}_k)$
- $Bel_k(X_k)$ is Gaussian $\rightarrow \mathcal{N}(x_k, P_k)$
- $P(Z_k|X)$ is Gaussian $\rightarrow \mathcal{N}(h(X), R)$

This is Gaussian Bayesian fusion:

$$\begin{aligned}
H &= \frac{\partial h}{\partial x} \longrightarrow \text{Jacobian} \\
K &= \bar{P}_k H^T \left(H\bar{P}_k H^T + R\right)^{-1} \\
P_k &= (I - KH)\bar{P}_k \\
x_k &= \bar{x}_{k-1} + K\left(z_k - H\bar{x}_{k-1}\right)
\end{aligned}$$

# Outline

# Objectives

## Kalman Filter

- Understand the important matrices
- Design your own filter
- Use cases for robotics

## Batch Estimation

- Link with filtering
- Design your own estimator
- Use cases

# System 1: Argos Float

## Description

- Float buoy
- GPS measurement every second

## Objective

- Continuous localisation

# System 2: GPS Navigation system

## Description

- GPS Navigation system in a **car**
- GPS measurement every second

## Objective

- Continuous localisation

# System 3: Integrated GPS Navigation system

## Description

- GPS Navigation system in a **car**
- Rear wheel displacement measure (e.g. differential)

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} \frac{\Delta s_L + \Delta s_R}{2} \\ 0 \\ \frac{\Delta s_R - \Delta s_L}{e} \end{bmatrix} \tag{5}$$

- GPS measurement every second

## Objective

- Continuous localisation

# System 4: Integrated GPS Navigation system

## Description

- GPS Navigation system in a **car**
- Speed and steering measurement

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \frac{v\tan(\beta)}{L} \end{bmatrix} \tag{6}$$

- GPS measurement every second

## Objective

- Continuous localisation

# System 5: Indoor Navigation system

## Description

- Indoor robot (e.g. Roomba)
- Differential wheel measurement
- Known feature observation with orientation, in body frame

$$Z_i = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} \tag{7}$$

## Objective

- Continuous localisation

# System 6: Indoor Navigation system

## Description

- Indoor robot (e.g. Roomba)
- Differential wheel measurement
- Known feature observation: position only in body frame

$$Z_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{8}$$

## Objective

- Continuous localisation

# System 6b: Indoor Navigation system

## Description

- Indoor robot (e.g. Roomba)
- Differential wheel measurement
- Known feature observation: range and bearing in body frame

$$Z_i = \begin{bmatrix} \rho_i \\ \beta_i \end{bmatrix} \tag{9}$$

## Objective

- Continuous localisation

# System 7: Indoor Navigation system

## Description

- Indoor robot (e.g. Roomba)
- Differential wheel measurement
- Known feature observation, bearing only in body frame

$$Z_i = \left[ \ \beta_i \ \right] \tag{10}$$

## Objective

- Continuous localisation

# System 8: Indoor Navigation system

## Description

- Indoor robot (e.g. Roomba)
- Differential wheel measurement
- Known feature observation, range only in body frame

$$Z_i = \left[ \begin{array}{c} \rho_i \end{array} \right] \quad (11)$$

## Objective

- Continuous localisation

# System 9: Underwater system

## Description

- Torpedo-shaped robot
- Lift-drag based motion model
- Known feature observation, range only (e.g. sonar pinger)

$$Z_i = \left[ \begin{array}{c} \rho_i \end{array} \right] \quad (12)$$

## Objective

- Continuous localisation

# System 10: Feature mapper

## Description

- Indoor robot
- Known localisation
- Observation of $n$ features with known Ids.
- Observation type: range, bearing, position, pose...

## Objective

- Map estimation

# System 11: Feature-based SLAM

## Description

- Indoor robot
- Unknown localisation
- Observation of $k$ features with known Ids.
- Observation type: range, bearing, position, pose...

## Objective

- Localisation and Map estimation

# System 12: Extrinsic calibration

## Description

- Indoor robot with $k$ sensors.
- Known localisation
- Joint observation of one feature with $k$ sensors.
- Observation type: range, bearing, position, pose...

## Objective

- Sensor position with respect to reference frame (e.g. sensor 1)
- Detection of loose sensors.

# System 13: auto-calibration

## Description

- Indoor robot with 1 sensor.
- Differential motion model integrating wheel diameter

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} \frac{r_L \Delta\theta_L + r_R \Delta\theta_R}{2} \\ 0 \\ \frac{r_R \Delta s_R - r_L \Delta s_L}{e} \end{bmatrix} \tag{13}$$

- Observation type: range, bearing, position, pose... with respect to known map.

## Objective

- Auto-estimate wheel diameter $r_L$ and $r_R$ and inter-wheel spacing $e$
- Puncture detection.