Autonomous Mobile Robots: Simultaneous Localization and Mapping

Dr. Hamid D. Taghirad

Advanced Robotics and Automated Systems (ARAS)
K.N. Toosi University of Technology

RSI/ISM International Conference on Robotics and Mechantronics (ICRoM) 2013

February 13, 2013

Outline

Table of Contents

Autonomous Mobile Robots

- The ultimate goal of mobile robotics is to design autonomous mobile robots.

- "The ability to simultaneously localize a robot and accurately map its environment is a key prerequisite of truly autonomous robots."
- SLAM stands for Simultaneous Localization and Mapping.

Autonomous Mobile Robots

- The ultimate goal of mobile robotics is to design autonomous mobile robots.

- "The ability to simultaneously localize a robot and accurately map its environment is a key prerequisite of truly autonomous robots."
- SLAM stands for Simultaneous Localization and Mapping.

Autonomous Mobile Robots

- The ultimate goal of mobile robotics is to design autonomous mobile robots.

- "The ability to simultaneously localize a robot and accurately map its environment is a key prerequisite of truly autonomous robots."
- SLAM stands for Simultaneous Localization and Mapping.

Autonomous Mobile Robots

- The ultimate goal of mobile robotics is to design autonomous mobile robots.


- "The ability to simultaneously localize a robot and accurately map its environment is a key prerequisite of truly autonomous robots."
- SLAM stands for Simultaneous Localization and Mapping.

Autonomous Mobile Robots

- The ultimate goal of mobile robotics is to design autonomous mobile robots.

- "The ability to simultaneously localize a robot and accurately map its environment is a key prerequisite of truly autonomous robots."
- SLAM stands for Simultaneous Localization and Mapping.

Autonomous Mobile Robots

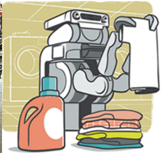
- The ultimate goal of mobile robotics is to design autonomous mobile robots.

- "The ability to simultaneously localize a robot and accurately map its environment is a key prerequisite of truly autonomous robots."
- SLAM stands for Simultaneous Localization and Mapping.

Autonomous Mobile Robots

- The ultimate goal of mobile robotics is to design autonomous mobile robots.



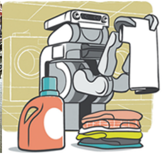



- "The ability to simultaneously localize a robot and accurately map its environment is a key prerequisite of truly autonomous robots."
- SLAM stands for Simultaneous Localization and Mapping.

Autonomous Mobile Robots

- The ultimate goal of mobile robotics is to design autonomous mobile robots.






- "The ability to simultaneously localize a robot and accurately map its environment is a key prerequisite of truly autonomous robots."
- SLAM stands for Simultaneous Localization and Mapping.

SLAM Problem

Basic Assumptions

- No *a priori* knowledge about the environment (i.e., no map)
- No independent position information (i.e., no GPS)

Given

- Noisy observations of the environment
- Noisy control signals (e.g., odometry)

Goal

- Estimate the map of the environment (e.g., locations of the features)
- Estimate the pose (position and orientation) OR trajectory of the robot

SLAM Problem

Basic Assumptions

- No *a priori* knowledge about the environment (i.e., no map)
- No independent position information (i.e., no GPS)

Given

- Noisy observations of the environment
- Noisy control signals (e.g., odometry)

Goal

- Estimate the map of the environment (e.g., locations of the features)
- Estimate the pose (position and orientation) OR trajectory of the robot

SLAM Problem

Basic Assumptions

- No *a priori* knowledge about the environment (i.e., no map)
- No independent position information (i.e., no GPS)

Given

- Noisy observations of the environment
- Noisy control signals (e.g., odometry)

Goal

- Estimate the map of the environment (e.g., locations of the features)
- Estimate the pose (position and orientation) OR trajectory of the robot

Map Representation

- Topological maps
- Grid maps
- Feature-based maps
- ...

Map Representation

- Topological maps
- Grid maps
- Feature-based maps
- . . .

Map Representation

- Topological maps
- Grid maps
- Feature-based maps
- . . .

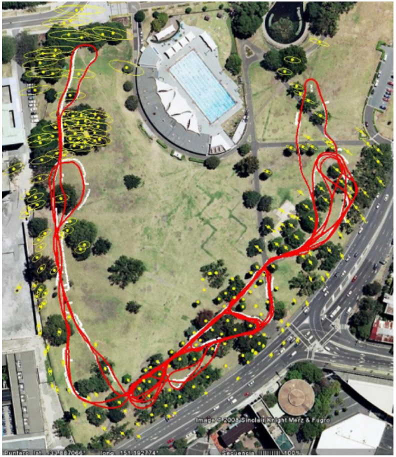


Figure : Victoria Park (Sydney)

Probabilistic Robotics

- "Uncertainty" is a part of Robotics
- Uncertain Models
- Uncertain Observations
- Uncertain Controls
- Uncertain Actions
- Uncertain World

Uncertainty arises from

- Partial Knowledge
- Noisy Measurements
- Incomplete Models, Modeling Limitations

Probabilistic Robotics

- "Uncertainty" is a part of Robotics
- Uncertain Models
- Uncertain Observations
- Uncertain Controls
- Uncertain Actions
- Uncertain World

Uncertainty arises from

- Partial Knowledge
- Noisy Measurements
- Incomplete Models, Modeling Limitations

Probabilistic Robotics

- "Uncertainty" is a part of Robotics
- Uncertain Models
- Uncertain Observations
- Uncertain Controls
- Uncertain Actions
- Uncertain World

Uncertainty arises from

- Partial Knowledge
- Noisy Measurements
- Incomplete Models, Modeling Limitations

Probabilistic Robotics

- "Uncertainty" is a part of Robotics
- Uncertain Models
- Uncertain Observations
- Uncertain Controls
- Uncertain Actions
- Uncertain World

Uncertainty arises from

- Partial Knowledge
- Noisy Measurements
- Incomplete Models, Modeling Limitations

Probabilistic Robotics

- "Uncertainty" is a part of Robotics
- Uncertain Models
- Uncertain Observations
- Uncertain Controls
- Uncertain Actions
- Uncertain World

Uncertainty arises from

- Partial Knowledge
- Noisy Measurements
- Incomplete Models, Modeling Limitations

Probabilistic Robotics

- "Uncertainty" is a part of Robotics
- Uncertain Models
- Uncertain Observations
- Uncertain Controls
- Uncertain Actions
- Uncertain World

Uncertainty arises from

- Partial Knowledge
- Noisy Measurements
- Incomplete Models, Modeling Limitations

Probabilistic Robotics

- "Uncertainty" is a part of Robotics
- Uncertain Models
- Uncertain Observations
- Uncertain Controls
- Uncertain Actions
- Uncertain World

Uncertainty arises from

- Partial Knowledge
- Noisy Measurements
- Incomplete Models, Modeling Limitations

Probabilistic Robotics

- "Uncertainty" is a part of Robotics
- Uncertain Models
- Uncertain Observations
- Uncertain Controls
- Uncertain Actions
- Uncertain World

Uncertainty arises from

- Partial Knowledge
- Noisy Measurements
- Incomplete Models, Modeling Limitations

Probabilistic Robotics

- "Uncertainty" is a part of Robotics
- Uncertain Models
- Uncertain Observations
- Uncertain Controls
- Uncertain Actions
- Uncertain World

Uncertainty arises from

- Partial Knowledge
- Noisy Measurements
- Incomplete Models, Modeling Limitations

Probabilistic Robotics

- "Uncertainty" is a part of Robotics
- Uncertain Models
- Uncertain Observations
- Uncertain Controls
- Uncertain Actions
- Uncertain World

Uncertainty arises from

- Partial Knowledge
- Noisy Measurements
- Incomplete Models, Modeling Limitations

Probabilistic Robotics

- "Uncertainty" is a part of Robotics
- Uncertain Models
- Uncertain Observations
- Uncertain Controls
- Uncertain Actions
- Uncertain World

Uncertainty arises from

- Partial Knowledge
- Noisy Measurements
- Incomplete Models, Modeling Limitations

Probabilistic Methods

Probabilistic approach tends to outperform deterministic approach

- Describe uncertainty in data, models and estimates
- Bayesian estimation
- Robot pose $\mathbf{s}_t$
- Robot pose is assumed to be a Markov process with initial distribution $p(\mathbf{s}_0)$
- Feature's location θ_i
- Map $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_N\}$
- Observation $\mathbf{z}_t$, and control input $\mathbf{u}_t$
- $\mathbf{x}_t = [\mathbf{s}_t \quad \boldsymbol{\theta}]^T$
- $\mathbf{x}_{1:t} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$

- Filtering distribution (Online SLAM): $p(\mathbf{s_t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- Smoothing distribution (Full SLAM): $p(\mathbf{s_{0:t}}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- MMSE estimate: $\hat{\mathbf{x}}_t = \mathbb{E}[\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$ $\hat{\mathbf{x}}_{0:t} = \mathbb{E}[\mathbf{x}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$
- Maximum a posteriori (MAP) Estimate

Probabilistic Methods

Probabilistic approach tends to outperform deterministic approach

- Describe uncertainty in data, models and estimates
- Bayesian estimation
- Robot pose $\mathbf{s}_t$
- Robot pose is assumed to be a Markov process with initial distribution $p(\mathbf{s}_0)$
- Feature's location θ_i
- Map $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_N\}$
- Observation $\mathbf{z}_t$, and control input $\mathbf{u}_t$
- $\mathbf{x}_t = [\mathbf{s}_t \quad \boldsymbol{\theta}]^T$
- $\mathbf{x}_{1:t} \triangleq \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$

- Filtering distribution (Online SLAM): $p(\mathbf{s}_t, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- Smoothing distribution (Full SLAM): $p(\mathbf{s}_{0:t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- MMSE estimate: $\hat{\mathbf{x}}_t = \mathbb{E}[\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$ $\hat{\mathbf{x}}_{0:t} = \mathbb{E}[\mathbf{x}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$
- Maximum a posteriori (MAP) Estimate

Probabilistic Methods

Probabilistic approach tends to outperform deterministic approach

- Describe uncertainty in data, models and estimates
- Bayesian estimation
  - Robot pose $\mathbf{s}_t$
  - Robot pose is assumed to be a Markov process with initial distribution $p(\mathbf{s}_0)$
  - Feature's location θ_i
  - Map $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_N\}$
  - Observation $\mathbf{z}_t$, and control input $\mathbf{u}_t$
  - $\mathbf{x}_t = [\mathbf{s}_t \quad \boldsymbol{\theta}]^T$
  - $\mathbf{x}_{1:t} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$

- Filtering distribution (Online SLAM): $p(\mathbf{s_t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- Smoothing distribution (Full SLAM): $p(\mathbf{s_{0:t}}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- MMSE estimate: $\hat{\mathbf{x}}_t = \mathbb{E}[\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$ $\hat{\mathbf{x}}_{0:t} = \mathbb{E}[\mathbf{x}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$
- Maximum a posteriori (MAP) Estimate

Probabilistic Methods

Probabilistic approach tends to outperform deterministic approach

- Describe uncertainty in data, models and estimates
- Bayesian estimation
- Robot pose $\mathbf{s}_t$
- Robot pose is assumed to be a Markov process with initial distribution $p(\mathbf{s}_0)$
- Feature's location θ_i
- Map $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_N\}$
- Observation $\mathbf{z}_t$, and control input $\mathbf{u}_t$
- $\mathbf{x}_t = [\mathbf{s}_t \quad \boldsymbol{\theta}]^T$
- $\mathbf{x}_{1:t} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$

- Filtering distribution (Online SLAM): $p(\mathbf{s_t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- Smoothing distribution (Full SLAM): $p(\mathbf{s_{0:t}}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- MMSE estimate: $\hat{\mathbf{x}}_t = \mathbb{E}[\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$ $\hat{\mathbf{x}}_{0:t} = \mathbb{E}[\mathbf{x}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$
- Maximum a posteriori (MAP) Estimate

Probabilistic Methods

Probabilistic approach tends to outperform deterministic approach

- Describe uncertainty in data, models and estimates
- Bayesian estimation
- Robot pose $\mathbf{s}_t$
- Robot pose is assumed to be a Markov process with initial distribution $p(\mathbf{s}_0)$
- Feature's location θ_i
- Map $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_N\}$
- Observation $\mathbf{z}_t$, and control input $\mathbf{u}_t$
- $\mathbf{x}_t = [\mathbf{s}_t \quad \boldsymbol{\theta}]^T$
- $\mathbf{x}_{1:t} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$

- Filtering distribution (Online SLAM): $p(\mathbf{s_t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- Smoothing distribution (Full SLAM): $p(\mathbf{s_{0:t}}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- MMSE estimate: $\hat{\mathbf{x}}_t = \mathbb{E}[\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$ $\hat{\mathbf{x}}_{0:t} = \mathbb{E}[\mathbf{x}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$
- Maximum a posteriori (MAP) Estimate

Probabilistic Methods

Probabilistic approach tends to outperform deterministic approach

- Describe uncertainty in data, models and estimates
- Bayesian estimation
- Robot pose $\mathbf{s}_t$
- Robot pose is assumed to be a Markov process with initial distribution $p(\mathbf{s}_0)$
- Feature's location θ_i
- Map $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_N\}$
- Observation $\mathbf{z}_t$, and control input $\mathbf{u}_t$
- $\mathbf{x}_t = [\mathbf{s}_t \quad \boldsymbol{\theta}]^T$
- $\mathbf{x}_{1:t} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$

- Filtering distribution (Online SLAM): $p(\mathbf{s_t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- Smoothing distribution (Full SLAM): $p(\mathbf{s_{0:t}}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- MMSE estimate: $\hat{\mathbf{x}}_t = \mathbb{E}[\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$ $\hat{\mathbf{x}}_{0:t} = \mathbb{E}[\mathbf{x}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$
- Maximum a posteriori (MAP) Estimate

Probabilistic Methods

Probabilistic approach tends to outperform deterministic approach

- Describe uncertainty in data, models and estimates
- Bayesian estimation
- Robot pose $\mathbf{s}_t$
- Robot pose is assumed to be a Markov process with initial distribution $p(\mathbf{s}_0)$
- Feature's location θ_i
- Map $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_N\}$
- Observation $\mathbf{z}_t$, and control input $\mathbf{u}_t$
- $\mathbf{x}_t = [\mathbf{s}_t \quad \boldsymbol{\theta}]^T$
- $\mathbf{x}_{1:t} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$

- Filtering distribution (Online SLAM): $p(\mathbf{s_t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- Smoothing distribution (Full SLAM): $p(\mathbf{s_{0:t}}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- MMSE estimate:
  $\hat{\mathbf{x}}_t = \mathbb{E}[\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$
  $\hat{\mathbf{x}}_{0:t} = \mathbb{E}[\mathbf{x}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$
- Maximum a posteriori (MAP) Estimate

Probabilistic Methods

Probabilistic approach tends to outperform deterministic approach

- Describe uncertainty in data, models and estimates
- Bayesian estimation

- Robot pose $\mathbf{s}_t$
- Robot pose is assumed to be a Markov process with initial distribution $p(\mathbf{s}_0)$
- Feature's location θ_i
- Map $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_N\}$
- Observation $\mathbf{z}_t$, and control input $\mathbf{u}_t$
- $\mathbf{x}_t = [\mathbf{s}_t \quad \boldsymbol{\theta}]^T$
- $\mathbf{x}_{1:t} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$

- Filtering distribution (Online SLAM): $p(\mathbf{s_t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- Smoothing distribution (Full SLAM): $p(\mathbf{s_{0:t}}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$
- MMSE estimate:
  $\hat{\mathbf{x}}_t = \mathbb{E}[\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$
  $\hat{\mathbf{x}}_{0:t} = \mathbb{E}[\mathbf{x}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}]$
- Maximum a posteriori (MAP) Estimate

State-Space Equations

- Robot motion equation:

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{u}_t, \mathbf{v}_t)$$

- Observation equation:

$$\mathbf{z}_t = g(\mathbf{s}_t, \theta_{n_t}, \mathbf{w}_t)$$

State-Space Equations

- Robot motion equation:

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{u}_t, \mathbf{v}_t)$$

- Observation equation:

$$\mathbf{z}_t = g(\mathbf{s}_t, \theta_{n_t}, \mathbf{w}_t)$$

- $f(\cdot,\cdot,\cdot)$ and $g(\cdot,\cdot,\cdot)$ are non-linear functions

State-Space Equations

- Robot motion equation:

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{u}_t, \mathbf{v}_t)$$

- Observation equation:

$$\mathbf{z}_t = g(\mathbf{s}_t, \theta_{n_t}, \mathbf{w}_t)$$

- $f(\cdot,\cdot,\cdot)$ and $g(\cdot,\cdot,\cdot)$ are non-linear functions
- $\mathbf{v}_t$ and $\mathbf{w}_t$ are zero-mean white Gaussian noises with covariances matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$

Bayes Filter

How to obtain the posterior distribution recursively in time? Bayes filter!

1. Prediction
2. Update

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)\, p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})\, d\mathbf{x}_{t-1} \tag{1}$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})}{\int p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_t} \tag{2}$$

Motion model

Observation model

There is a similar recursive formula for the smoothing density

Bayes Filter

How to obtain the posterior distribution recursively in time? Bayes filter!

1. Prediction
2. Update

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)\, p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})\, d\mathbf{x}_{t-1} \quad (1)$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})}{\int p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_t} \quad (2)$$

Motion model

Observation model

There is a similar recursive formula for the smoothing density

Bayes Filter

How to obtain the posterior distribution recursively in time? Bayes filter!

1. Prediction
2. Update

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t)\, p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t-1})\, d\mathbf{x}_{t-1} \quad (1)$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t},\mathbf{u}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t})}{\int p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t}) d\mathbf{x}_t} \quad (2)$$

Motion model

Observation model

There is a similar recursive formula for the smoothing density

Bayes Filter

How to obtain the posterior distribution recursively in time? Bayes filter!

1. Prediction
2. Update

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)\, p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})\, d\mathbf{x}_{t-1} \quad (1)$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})}{\int p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_t} \quad (2)$$

Motion model

Observation model

There is a similar recursive formula for the smoothing density

Bayes Filter

How to obtain the posterior distribution recursively in time? Bayes filter!

1. Prediction
2. Update

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t)\, p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t-1})\, d\mathbf{x}_{t-1} \tag{1}$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t},\mathbf{u}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t})}{\int p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t}) d\mathbf{x}_t} \tag{2}$$

Motion model

Observation model

There is a similar recursive formula for the smoothing density

Bayes Filter

How to obtain the posterior distribution recursively in time? Bayes filter!

1. Prediction
2. Update

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)\ p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})\ d\mathbf{x}_{t-1} \quad (1)$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)\ p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})}{\int p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})d\mathbf{x}_t} \quad (2)$$

Motion model

Observation model

There is a similar recursive formula for the smoothing density

Bayes Filter

How to obtain the posterior distribution recursively in time? Bayes filter!

1. Prediction
2. Update

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t)\, p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t-1})\, d\mathbf{x}_{t-1} \quad (1)$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t},\mathbf{u}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t})}{\int p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1},\mathbf{u}_{1:t}) d\mathbf{x}_t} \quad (2)$$

Motion model

Observation model

There is a similar recursive formula for the smoothing density

Bayes Filter Cont'd.

Example

For the case of linear-Gaussian models, Bayes filter equations would be simplified into the Kalman filter equations.

Bayes Filter Cont'd.

Example

For the case of linear-Gaussian models, Bayes filter equations would be simplified into the Kalman filter equations.

But . . .

In general, it is impossible to implement the exact Bayes filter because it requires the ability to evaluate complex high-dimensional integrals.

Bayes Filter Cont'd.

Example

For the case of linear-Gaussian models, Bayes filter equations would be simplified into the Kalman filter equations.

But . . .

In general, it is impossible to implement the exact Bayes filter because it requires the ability to evaluate complex high-dimensional integrals.

So we have to approximate . . .

- Extended Kalman Filter (EKF)
- Unscented Kalman Filter (UKF)
- Gaussian-Sum Filter
- Extended Information Filter (EIF)
- Particle Filter (A.K.A. Sequential Monte Carlo Methods)

Table of Contents

How it begins?

According to Bailey & Durrant-Whyte:

- ICRA'86
- Applying Estimation-theoretic and probabilistic methods to Mapping and Localization problems
- Smith, Self and Cheeseman: Landmark estimates become correlated
- They did not expect convergence at the time
- Minimize or ignore these correlations

How it begins?

According to Bailey & Durrant-Whyte:

- ICRA'86
- Applying Estimation-theoretic and probabilistic methods to Mapping and Localization problems
- Smith, Self and Cheeseman: Landmark estimates become correlated
- They did not expect convergence at the time
- Minimize or ignore these correlations

How it begins?

According to Bailey & Durrant-Whyte:

- ICRA'86
- Applying Estimation-theoretic and probabilistic methods to Mapping and Localization problems
- Smith, Self and Cheeseman: Landmark estimates become correlated
- They did not expect convergence at the time
- Minimize or ignore these correlations

Csorba and Dissanayake: SLAM is convergent!

Outline

1. Introduction
   - SLAM Problem
   - Probabilistic Methods
   - Bayesian Filtering
2. SLAM: Past and Present
   - History
   - EKF-SLAM
   - RBPF-SLAM (FastSLAM, . . .)
   - SEIF
   - Optimization Approach (GraphSLAM, SAM, . . .)
   - Softwares and Datasets
   - Final Remarks

EKF-SLAM

- State-Space Equations:

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{u}_t, \mathbf{v}_t)$$

$$\mathbf{z}_t = g(\mathbf{s}_t, \theta_{n_t}, \mathbf{w}_t)$$

EKF-SLAM

- State-Space Equations:

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{u}_t, \mathbf{v}_t)$$

$$\mathbf{z}_t = g(\mathbf{s}_t, \theta_{n_t}, \mathbf{w}_t)$$

- $f(\cdot, \cdot, \cdot)$ and $g(\cdot, \cdot, \cdot)$ are non-linear functions

EKF-SLAM

- State-Space Equations:

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{u}_t, \mathbf{v}_t)$$

$$\mathbf{z}_t = g(\mathbf{s}_t, \theta_{n_t}, \mathbf{w}_t)$$

- $f(\cdot,\cdot,\cdot)$ and $g(\cdot,\cdot,\cdot)$ are non-linear functions
- $\mathbf{v}_t$ and $\mathbf{w}_t$ are zero-mean white Gaussian noises with covariances matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$

EKF-SLAM

- State-Space Equations:

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{u}_t, \mathbf{v}_t)$$

$$\mathbf{z}_t = g(\mathbf{s}_t, \theta_{n_t}, \mathbf{w}_t)$$

- $f(\cdot,\cdot,\cdot)$ and $g(\cdot,\cdot,\cdot)$ are non-linear functions
- $\mathbf{v}_t$ and $\mathbf{w}_t$ are zero-mean white Gaussian noises with covariances matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$

We can approximate the filtering distribution using EKF

EKF-SLAM

- State-Space Equations:

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{u}_t, \mathbf{v}_t)$$

$$\mathbf{z}_t = g(\mathbf{s}_t, \theta_{n_t}, \mathbf{w}_t)$$

- $f(\cdot,\cdot,\cdot)$ and $g(\cdot,\cdot,\cdot)$ are non-linear functions
- $\mathbf{v}_t$ and $\mathbf{w}_t$ are zero-mean white Gaussian noises with covariances matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$

We can approximate the filtering distribution using EKF

- State Vector $\mathbf{x}_t = [\mathbf{s}_t \quad \boldsymbol{\theta}]^T = [\mathbf{s}_t \quad \theta_1 \ldots \theta_M]^T$

EKF-SLAM

- State-Space Equations:

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{u}_t, \mathbf{v}_t)$$

$$\mathbf{z}_t = g(\mathbf{s}_t, \theta_{n_t}, \mathbf{w}_t)$$

- $f(\cdot,\cdot,\cdot)$ and $g(\cdot,\cdot,\cdot)$ are non-linear functions
- $\mathbf{v}_t$ and $\mathbf{w}_t$ are zero-mean white Gaussian noises with covariances matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$

We can approximate the filtering distribution using EKF

- State Vector $\mathbf{x}_t = [\mathbf{s}_t \quad \boldsymbol{\theta}]^T = [\mathbf{s}_t \quad \theta_1 \ldots \theta_M]^T$
- Covariance Matrix

$$P = \begin{bmatrix} P_{\mathbf{ss}} & P_{\mathbf{s}\boldsymbol{\theta}} \\ P_{\mathbf{s}\boldsymbol{\theta}}^T & P_{\boldsymbol{\theta\theta}} \end{bmatrix}$$

EKF-SLAM Cont'd.

- Motion and observation models are linearized around the best available estimate:

$$\mathbf{x}_t \approx f(\hat{\mathbf{x}}^+_{t-1}, \mathbf{u}_t, 0) + \nabla \mathbf{f}_\mathbf{x}(\mathbf{x}_{t-1} - \hat{\mathbf{x}}^+_{t-1}) + \nabla \mathbf{f}_\mathbf{v} \mathbf{v}_t$$
$$\mathbf{z}_t \approx g(\hat{\mathbf{x}}^-_t, 0) + \nabla \mathbf{g}_\mathbf{x}(\mathbf{x}_t - \hat{\mathbf{x}}^-_t) + \nabla \mathbf{g}_\mathbf{w} \mathbf{w}_t$$

- Filtering density is approximated by a Gaussian distribution: $p(\mathbf{s_t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \approx \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}^+_t, \mathbf{P}^+_t)$ where:

$$\begin{aligned}
\hat{\mathbf{x}}^-_t &= f(\hat{\mathbf{x}}^+_{t-1}, \mathbf{u}_t, 0) \\
\mathbf{P}^-_t &= \nabla \mathbf{f}_\mathbf{x} \mathbf{P}^+_{t-1} \nabla \mathbf{f}^T_\mathbf{x} + \nabla \mathbf{f}_\mathbf{v} \mathbf{Q}_t \nabla \mathbf{f}^T_\mathbf{v} \\
\hat{\mathbf{x}}^+_t &= \hat{\mathbf{x}}^-_t + \mathbf{K}_t \nu_t \\
\mathbf{P}^+_t &= \mathbf{P}^-_t - \mathbf{K}_t \mathbf{S}_t \mathbf{K}^T_t \\
\nu_t &= \mathbf{z}_t - g(\hat{\mathbf{x}}^-_t, 0) \\
\mathbf{S}_t &= \nabla \mathbf{g}_\mathbf{x} \mathbf{P}^-_t \nabla \mathbf{g}^T_\mathbf{x} + \nabla \mathbf{g}_\mathbf{w} \mathbf{R}_t \nabla \mathbf{g}^T_\mathbf{w} \\
\mathbf{K}_t &= \mathbf{P}^-_t \nabla \mathbf{g}^T_\mathbf{x} \mathbf{S}^{-1}_t
\end{aligned}$$

EKF-SLAM Cont'd.

EKF-SLAM drawbacks

1. Quadratic Computational Complexity in the Number of Features

EKF-SLAM Cont'd.

EKF-SLAM drawbacks

1. Quadratic Computational Complexity in the Number of Features
2. Overconfident and Inconsistent Estimates: Linearization Errors!

EKF-SLAM Cont'd.

EKF-SLAM drawbacks

1. Quadratic Computational Complexity in the Number of Features
2. Overconfident and Inconsistent Estimates: Linearization Errors!

- Submapping approaches partially address these issues: build small local maps with EKF and join them together

EKF-SLAM Cont'd.

EKF-SLAM drawbacks

1. Quadratic Computational Complexity in the Number of Features
2. Overconfident and Inconsistent Estimates: Linearization Errors!

- Submapping approaches partially address these issues: build small local maps with EKF and join them together
- Alternative KF-based solutions: UKF-SLAM, IEKF-SLAM

EKF-SLAM Cont'd.

EKF-SLAM drawbacks

1. Quadratic Computational Complexity in the Number of Features
2. Overconfident and Inconsistent Estimates: Linearization Errors!

- Submapping approaches partially address these issues: build small local maps with EKF and join them together
- Alternative KF-based solutions: UKF-SLAM, IEKF-SLAM
- In practice the application of EKF-SLAM is limited to small environments

Outline

Particle Filter in Robotics

Unlike EKF-SLAM, FastSLAM (Rao-Blackwellized Particle Filter SLAM in general) is basically a Full SLAM solution (*Smoothing* instead of Filtering).

Particle Filter in Robotics

Unlike EKF-SLAM, FastSLAM (Rao-Blackwellized Particle Filter SLAM in general) is basically a Full SLAM solution (*Smoothing* instead of Filtering).

- Particle Filter had been successfully applied to Localization: MCL (Monte Carlo Localization)

Particle Filter in Robotics

Unlike EKF-SLAM, FastSLAM (Rao-Blackwellized Particle Filter SLAM in general) is basically a Full SLAM solution (*Smoothing* instead of Filtering).

- Particle Filter had been successfully applied to Localization: MCL (Monte Carlo Localization)
- SLAM is a very high-dimensional problem (map is growing) while the dimension of state vector in Localization is fixed and small

Particle Filter in Robotics

Unlike EKF-SLAM, FastSLAM (Rao-Blackwellized Particle Filter SLAM in general) is basically a Full SLAM solution (*Smoothing* instead of Filtering).

- Particle Filter had been successfully applied to Localization: MCL (Monte Carlo Localization)
- SLAM is a very high-dimensional problem (map is growing) while the dimension of state vector in Localization is fixed and small

Estimating $p(\mathbf{s}_{0:t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ using a Particle Filter can be very inefficient

Particle Filter in Robotics

Unlike EKF-SLAM, FastSLAM (Rao-Blackwellized Particle Filter SLAM in general) is basically a Full SLAM solution (*Smoothing* instead of Filtering).

- Particle Filter had been successfully applied to Localization: MCL (Monte Carlo Localization)
- SLAM is a very high-dimensional problem (map is growing) while the dimension of state vector in Localization is fixed and small

Estimating $p(\mathbf{s}_{0:t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ using a Particle Filter can be very inefficient

Rao-Blackwellization in Particle Filtering

Factor the posterior distribution and estimate the map analytically using EKF

FastSLAM Factorization

- We can factor the smoothing distribution into two parts as

$$p(\mathbf{s}_{0:t}, \boldsymbol{\theta}|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{s}_{0:t}|\mathbf{z}_{1:t}, \mathbf{u}_{1:t})\, p(\boldsymbol{\theta}|\ \mathbf{s}_{0:t}\ , \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

But . . .

In SLAM, landmark estimates are **conditionally independent** given the **robot trajectory**

- Therefore we have:

$$p(\mathbf{s}_{0:t}, \boldsymbol{\theta}|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{s}_{0:t}|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \prod_{k=1}^{M} p(\theta_k|\ \mathbf{s}_{0:t}\ , \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

1. Estimate robot's trajectory using a Particle Filter (e.g., SIR)
2. For each path particle, estimate each landmark's locations using a low dimensional (e.g., 2 × 2) EKF

FastSLAM Factorization

- We can factor the smoothing distribution into two parts as

$$p(\mathbf{s}_{0:t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{s}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})\, p(\boldsymbol{\theta} | \mathbf{s}_{0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

But . . .

In SLAM, landmark estimates are **conditionally independent** given the **robot trajectory**

- Therefore we have:

$$p(\mathbf{s}_{0:t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{s}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \prod_{k=1}^{M} p(\theta_k | \mathbf{s}_{0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

1. Estimate robot's trajectory using a Particle Filter (e.g., SIR)
2. For each path particle, estimate each landmark's locations using a low dimensional (e.g., 2 × 2) EKF

FastSLAM Factorization

- We can factor the smoothing distribution into two parts as

$$p(\mathbf{s}_{0:t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{s}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})\, p(\boldsymbol{\theta} | \mathbf{s}_{0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

But ...

In SLAM, landmark estimates are **conditionally independent** given the **robot trajectory**

- Therefore we have:

$$p(\mathbf{s}_{0:t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{s}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \prod_{k=1}^{M} p(\theta_k | \mathbf{s}_{0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

1. Estimate robot's trajectory using a Particle Filter (e.g., SIR)
2. For each path particle, estimate each landmark's locations using a low dimensional (e.g., 2×2) EKF

FastSLAM Factorization

- We can factor the smoothing distribution into two parts as

$$p(\mathbf{s}_{0:t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{s}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})\, p(\boldsymbol{\theta} | \mathbf{s}_{0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

But ...

In SLAM, landmark estimates are **conditionally independent** given the **robot trajectory**

- Therefore we have:

$$p(\mathbf{s}_{0:t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{s}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \prod_{k=1}^{M} p(\theta_k | \mathbf{s}_{0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

1. Estimate robot's trajectory using a Particle Filter (e.g., SIR)
2. For each path particle, estimate each landmark's locations using a low dimensional (e.g., 2×2) EKF

FastSLAM Factorization

- We can factor the smoothing distribution into two parts as

$$p(\mathbf{s}_{0:t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{s}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})\, p(\boldsymbol{\theta} | \mathbf{s}_{0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

But ...

In SLAM, landmark estimates are **conditionally independent** given the **robot trajectory**

- Therefore we have:

$$p(\mathbf{s}_{0:t}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{s}_{0:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \prod_{k=1}^{M} p(\theta_k | \mathbf{s}_{0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

1. Estimate robot's trajectory using a Particle Filter (e.g., SIR)
2. For each path particle, estimate each landmark's locations using a low dimensional (e.g., 2 $\times$ 2) EKF

Pros and Cons

Pros

- FastSLAM: Linear/Logarithmic complexity in the number of landmarks

Pros and Cons

Pros

- FastSLAM: Linear/Logarithmic complexity in the number of landmarks
- Multiple-hypothesis data association: Perform data association per-particle

Pros and Cons

Pros

- FastSLAM: Linear/Logarithmic complexity in the number of landmarks
- Multiple-hypothesis data association: Perform data association per-particle

Cons

Like any other sequential Monte Carlo method, FastSLAM suffers from:

Pros and Cons

Pros

- FastSLAM: Linear/Logarithmic complexity in the number of landmarks
- Multiple-hypothesis data association: Perform data association per-particle

Cons

Like any other sequential Monte Carlo method, FastSLAM suffers from:

- Degeneracy

Pros and Cons

Pros

- FastSLAM: Linear/Logarithmic complexity in the number of landmarks
- Multiple-hypothesis data association: Perform data association per-particle

Cons

Like any other sequential Monte Carlo method, FastSLAM suffers from:

- Degeneracy
- Sample Impoverishment (A.K.A. particle depletion, common history, etc)

Pros and Cons

Pros

- FastSLAM: Linear/Logarithmic complexity in the number of landmarks
- Multiple-hypothesis data association: Perform data association per-particle

Cons

Like any other sequential Monte Carlo method, FastSLAM suffers from:

- Degeneracy
- Sample Impoverishment (A.K.A. particle depletion, common history, etc)

Partial Solutions

- Sample from the "optimal proposal distribution" instead of motion model: slower rate of degeneracy (FastSLAM 2.0)

Pros and Cons

Pros

- FastSLAM: Linear/Logarithmic complexity in the number of landmarks
- Multiple-hypothesis data association: Perform data association per-particle

Cons

Like any other sequential Monte Carlo method, FastSLAM suffers from:

- Degeneracy
- Sample Impoverishment (A.K.A. particle depletion, common history, etc)

Partial Solutions

- Sample from the "optimal proposal distribution" instead of motion model: slower rate of degeneracy (FastSLAM 2.0)
- Resample according to the Effective Sample Size (ESS) to avoid unnecessary Resampling steps: slower rate of depletion (GridSLAM)

Outline

Extended Information Filter

- Information Filter (IF) is based on the canonical representation of a Gaussian distribution $\mathcal{N}(\mu, \Omega)$
- Information (Precision) Matrix instead of Covariance Matrix: $\Omega = \Sigma^{-1}$
- Information Vector instead of Mean Vector: $\xi = \Omega\mu$
- (E)IF is mathematically identical to (E)KF: updates information matrix and information vector instead of covariance matrix and mean vector

But ...

Complexity of a naive implementation of EIF-SLAM is cubic in the number of features!

Extended Information Filter

- Information Filter (IF) is based on the canonical representation of a Gaussian distribution $\mathcal{N}(\mu, \Omega)$
- Information (Precision) Matrix instead of Covariance Matrix: $\Omega = \Sigma^{-1}$
- Information Vector instead of Mean Vector: $\xi = \Omega\mu$
- (E)IF is mathematically identical to (E)KF: updates information matrix and information vector instead of covariance matrix and mean vector

But . . .

Complexity of a naive implementation of EIF-SLAM is cubic in the number of features!

Extended Information Filter

- Information Filter (IF) is based on the canonical representation of a Gaussian distribution $\mathcal{N}(\mu, \Omega)$
- Information (Precision) Matrix instead of Covariance Matrix: $\Omega = \Sigma^{-1}$
- Information Vector instead of Mean Vector: $\xi = \Omega\mu$
- (E)IF is mathematically identical to (E)KF: updates information matrix and information vector instead of covariance matrix and mean vector

But . . .

Complexity of a naive implementation of EIF-SLAM is cubic in the number of features!

Extended Information Filter

- Information Filter (IF) is based on the canonical representation of a Gaussian distribution $\mathcal{N}(\mu, \Omega)$
- Information (Precision) Matrix instead of Covariance Matrix: $\Omega = \Sigma^{-1}$
- Information Vector instead of Mean Vector: $\xi = \Omega\mu$
- (E)IF is mathematically identical to (E)KF: updates information matrix and information vector instead of covariance matrix and mean vector

But ...

Complexity of a naive implementation of EIF-SLAM is cubic in the number of features!

Extended Information Filter

- Information Filter (IF) is based on the canonical representation of a Gaussian distribution $\mathcal{N}(\mu, \Omega)$
- Information (Precision) Matrix instead of Covariance Matrix: $\Omega = \Sigma^{-1}$
- Information Vector instead of Mean Vector: $\xi = \Omega\mu$
- (E)IF is mathematically identical to (E)KF: updates information matrix and information vector instead of covariance matrix and mean vector

But . . .

Complexity of a naive implementation of EIF-SLAM is cubic in the number of features!

Sparse EIF

Reminder

Fully Correlated Landmarks in EKF-SLAM: covariance matrix in EKF-SLAM becomes dense very soon

Sparse EIF

Reminder

Fully Correlated Landmarks in EKF-SLAM: covariance matrix in EKF-SLAM becomes dense very soon

- It was first observed by Thrun *et al.* (2004) that the *normalized* information matrix in SLAM is almost sparse

Sparse EIF

Reminder

Fully Correlated Landmarks in EKF-SLAM: covariance matrix in EKF-SLAM becomes dense very soon

- It was first observed by Thrun *et al.* (2004) that the *normalized* information matrix in SLAM is almost sparse
- This observation was later theoretically proven by Frese (2005)

Sparse EIF

Reminder

Fully Correlated Landmarks in EKF-SLAM: covariance matrix in EKF-SLAM becomes dense very soon

- It was first observed by Thrun *et al.* (2004) that the *normalized* information matrix in SLAM is almost sparse
- This observation was later theoretically proven by Frese (2005)

SEIF

SEIF utilizes this observation and by introducing a “sparsification” step **approximates** the information matrix of EIF-SLAM by a sparse matrix

Sparse EIF

Reminder

Fully Correlated Landmarks in EKF-SLAM: covariance matrix in EKF-SLAM becomes dense very soon

- It was first observed by Thrun *et al.* (2004) that the *normalized* information matrix in SLAM is almost sparse
- This observation was later theoretically proven by Frese (2005)

SEIF

SEIF utilizes this observation and by introducing a "sparsification" step **approximates** the information matrix of EIF-SLAM by a sparse matrix

- Sparse Information Matrix $\Rightarrow$ constant-time algorithm (instead of cubic!)

Sparse EIF

Reminder

Fully Correlated Landmarks in EKF-SLAM: covariance matrix in EKF-SLAM becomes dense very soon

- It was first observed by Thrun *et al.* (2004) that the *normalized* information matrix in SLAM is almost sparse
- This observation was later theoretically proven by Frese (2005)

SEIF

SEIF utilizes this observation and by introducing a "sparsification" step **approximates** the information matrix of EIF-SLAM by a sparse matrix

- Sparse Information Matrix $\Rightarrow$ **constant-time** algorithm (instead of cubic!)

Sparse EIF

Reminder

Fully Correlated Landmarks in EKF-SLAM: covariance matrix in EKF-SLAM becomes dense very soon

- It was first observed by Thrun *et al.* (2004) that the *normalized* information matrix in SLAM is almost sparse
- This observation was later theoretically proven by Frese (2005)

SEIF

SEIF utilizes this observation and by introducing a "sparsification" step **approximates** the information matrix of EIF-SLAM by a sparse matrix

- Sparse Information Matrix ⇒ **constant-time** algorithm (instead of cubic!)

Trade-off

We are simply loosing accuracy!

Important Remarks

It was first realized by Thrun *et al.* in SEIF ...

- The canonical formulation of SLAM is equivalent to a Gaussian Markov Random Field (GMRF) (each robot pose/landmark position corresponds to a node, and two nodes are connected iff the corresponding term in the information matrix is non-zero)

Important Remarks

It was first realized by Thrun *et al.* in SEIF . . .

- The canonical formulation of SLAM is equivalent to a Gaussian Markov Random Field (GMRF) (each robot pose/landmark position corresponds to a node, and two nodes are connected iff the corresponding term in the information matrix is non-zero)
- It is the marginalization of previous robot pose in motion update step that makes the information matrix dense

Important Remarks

It was first realized by Thrun *et al.* in SEIF . . .

- The canonical formulation of SLAM is equivalent to a Gaussian Markov Random Field (GMRF) (each robot pose/landmark position corresponds to a node, and two nodes are connected iff the corresponding term in the information matrix is non-zero)
- It is the marginalization of previous robot pose in motion update step that makes the information matrix dense $\Rightarrow$ If we keep the robot trajectory in the state vector we will end up with an **exactly** sparse information matrix

Important Remarks

It was first realized by Thrun *et al.* in SEIF . . .

- The canonical formulation of SLAM is equivalent to a Gaussian Markov Random Field (GMRF) (each robot pose/landmark position corresponds to a node, and two nodes are connected iff the corresponding term in the information matrix is non-zero)
- It is the marginalization of previous robot pose in motion update step that makes the information matrix dense $\Rightarrow$ If we keep the robot trajectory in the state vector we will end up with an **exactly** sparse information matrix

- It can be shown that under the Gaussian assumption, information matrix describes the conditional independence (CI) between the random variables:

Important Remarks

It was first realized by Thrun *et al.* in SEIF ...

- The canonical formulation of SLAM is equivalent to a Gaussian Markov Random Field (GMRF) (each robot pose/landmark position corresponds to a node, and two nodes are connected iff the corresponding term in the information matrix is non-zero)
- It is the marginalization of previous robot pose in motion update step that makes the information matrix dense $\Rightarrow$ If we keep the robot trajectory in the state vector we will end up with an **exactly** sparse information matrix

- It can be shown that under the Gaussian assumption, information matrix describes the conditional independence (CI) between the random variables: an off-diagonal term in the information matrix is zero iff the corresponding random variables are conditionally independent given other random variables

Important Remarks

It was first realized by Thrun *et al.* in SEIF . . .

- The canonical formulation of SLAM is equivalent to a Gaussian Markov Random Field (GMRF) (each robot pose/landmark position corresponds to a node, and two nodes are connected iff the corresponding term in the information matrix is non-zero)
- It is the marginalization of previous robot pose in motion update step that makes the information matrix dense $\Rightarrow$ If we keep the robot trajectory in the state vector we will end up with an **exactly** sparse information matrix

- It can be shown that under the Gaussian assumption, information matrix describes the conditional independence (CI) between the random variables: an off-diagonal term in the information matrix is zero iff the corresponding random variables are conditionally independent given other random variables
- This reminds us of the CI property of Full SLAM (remember FastSLAM?)

Outline

Optimization approach to Full SLAM

Consequently Full SLAM came under the spotlight once again

- Counterintuitive conclusion: Solving Full SLAM could be easier that Online SLAM!
- As it was predicted in SEIF, the relation between GMRF and information matrix attracted the attention of SLAM community toward probabilistic graphical models
- Graphical models provide a natural representation of SLAM due to their natural ability in describing **conditional independence**

Optimization approach to Full SLAM

Consequently Full SLAM came under the spotlight once again

- Counterintuitive conclusion: Solving Full SLAM could be easier that Online SLAM!
- As it was predicted in SEIF, the relation between GMRF and information matrix attracted the attention of SLAM community toward probabilistic graphical models
- Graphical models provide a natural representation of SLAM due to their natural ability in describing **conditional independence**

Optimization approach to Full SLAM

Consequently Full SLAM came under the spotlight once again

- Counterintuitive conclusion: Solving Full SLAM could be easier that Online SLAM!
- As it was predicted in SEIF, the relation between GMRF and information matrix attracted the attention of SLAM community toward probabilistic graphical models
- Graphical models provide a natural representation of SLAM due to their natural ability in describing **conditional independence**

Optimization approach to Full SLAM

Consequently Full SLAM came under the spotlight once again

- Counterintuitive conclusion: Solving Full SLAM could be easier that Online SLAM!
- As it was predicted in SEIF, the relation between GMRF and information matrix attracted the attention of SLAM community toward probabilistic graphical models
- Graphical models provide a natural representation of SLAM due to their natural ability in describing **conditional independence**

Non-linear Least Squares

- The smoothing distribution can be factored into:

$$p(\mathbf{s_{0:t}}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \propto p(\mathbf{s}_0) \prod_i p(\mathbf{s}_i | \mathbf{s}_{i-1}, \mathbf{u}_i) p(\mathbf{z}_i | \mathbf{s}_i, \theta_{n_i})$$

- Maximizing the posterior is equivalent to minimizing its negative Log
- For Gaussian motion and observation models (which implies linear/linearized motion and observation equations with respect to the noise variable), maximum a posteriori (MAP) estimate is obtained by minimizing:

$$\mathbf{s}_0^T P_0^{-1} \mathbf{s}_0 + \sum_i (\mathbf{s}_i - f(\mathbf{s}_{i-1}, \mathbf{u}_i))^T \mathbf{Q}_i^{-1} (\mathbf{s}_i - f(\mathbf{s}_{i-1}, \mathbf{u}_i))$$
$$+ \sum_i (\mathbf{z}_i - g(\mathbf{s}_i, \theta_{n_i}))^T \mathbf{R}_i^{-1} (\mathbf{z}_i - g(\mathbf{s}_i, \theta_{n_i}))$$

- This is equivalent to the maximum likelihood estimate if we are treating the state variables as nonrandom/fixed unknown parameters.

Non-linear Least Squares

- The smoothing distribution can be factored into:

$$p(\mathbf{s_{0:t}}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \propto p(\mathbf{s}_0) \prod_i p(\mathbf{s}_i | \mathbf{s}_{i-1}, \mathbf{u}_i) p(\mathbf{z}_i | \mathbf{s}_i, \theta_{n_i})$$

- Maximizing the posterior is equivalent to minimizing its negative Log
- For Gaussian motion and observation models (which implies linear/linearized motion and observation equations with respect to the noise variable), maximum a posteriori (MAP) estimate is obtained by minimizing:

$$\mathbf{s}_0^T P_0^{-1} \mathbf{s}_0 + \sum_i (\mathbf{s}_i - f(\mathbf{s}_{i-1}, \mathbf{u}_i))^T \mathbf{Q}_i^{-1} (\mathbf{s}_i - f(\mathbf{s}_{i-1}, \mathbf{u}_i))$$
$$+ \sum_i (\mathbf{z}_i - g(\mathbf{s}_i, \theta_{n_i}))^T \mathbf{R}_i^{-1} (\mathbf{z}_i - g(\mathbf{s}_i, \theta_{n_i}))$$

- This is equivalent to the maximum likelihood estimate if we are treating the state variables as nonrandom/fixed unknown parameters.

Non-linear Least Squares

- The smoothing distribution can be factored into:

$$p(\mathbf{s_{0:t}}, \boldsymbol{\theta} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \propto p(\mathbf{s}_0) \prod_i p(\mathbf{s}_i | \mathbf{s}_{i-1}, \mathbf{u}_i) p(\mathbf{z}_i | \mathbf{s}_i, \theta_{n_i})$$

- Maximizing the posterior is equivalent to minimizing its negative Log
- For Gaussian motion and observation models (which implies linear/linearized motion and observation equations with respect to the noise variable), maximum a posteriori (MAP) estimate is obtained by minimizing:

$$\begin{aligned} \mathbf{s}_0^T P_0^{-1} \mathbf{s}_0 + & \sum_i (\mathbf{s}_i - f(\mathbf{s}_{i-1}, \mathbf{u}_i))^T \mathbf{Q}_i^{-1} (\mathbf{s}_i - f(\mathbf{s}_{i-1}, \mathbf{u}_i)) \\ & + \sum_i (\mathbf{z}_i - g(\mathbf{s}_i, \theta_{n_i}))^T \mathbf{R}_i^{-1} (\mathbf{z}_i - g(\mathbf{s}_i, \theta_{n_i})) \end{aligned}$$

- This is equivalent to the maximum likelihood estimate if we are treating the state variables as nonrandom/fixed unknown parameters.

State Of The Art

- Various techniques have been applied to this non-linear least squares form of SLAM: Gauss-Newton, Levenberg-Marquardt, Gradient Descent, Stochastic Gradient Descent, Multi-Level Relaxation, Convex Relaxation, Powell's dogleg (Trust Region), . . .
- Very efficient and accurate Full SLAM solutions
- Full SLAM is no longer an offline/batch approach (e.g., see iSAM by Kaess *et al.*)

Pose-graphs

- Landmarks are omitted from the state vector and instead, additional constraints ("observations") are added between different robot poses (e.g., using scan matching)
- See Tree-based network optimizer (TORO) by Grisetti *et al.*
- More popular than feature-based SLAM!

State Of The Art

- Various techniques have been applied to this non-linear least squares form of SLAM: Gauss-Newton, Levenberg-Marquardt, Gradient Descent, Stochastic Gradient Descent, Multi-Level Relaxation, Convex Relaxation, Powell's dogleg (Trust Region), ...
- Very efficient and accurate Full SLAM solutions
- Full SLAM is no longer an offline/batch approach (e.g., see iSAM by Kaess *et al.*)

Pose-graphs

- Landmarks are omitted from the state vector and instead, additional constraints ("observations") are added between different robot poses (e.g., using scan matching)
- See Tree-based network optimizer (TORO) by Grisetti *et al.*
- More popular than feature-based SLAM!

State Of The Art

- Various techniques have been applied to this non-linear least squares form of SLAM: Gauss-Newton, Levenberg-Marquardt, Gradient Descent, Stochastic Gradient Descent, Multi-Level Relaxation, Convex Relaxation, Powell's dogleg (Trust Region), ...
- Very efficient and accurate Full SLAM solutions
- Full SLAM is no longer an offline/batch approach (e.g., see iSAM by Kaess *et al.*)

Pose-graphs

- Landmarks are omitted from the state vector and instead, additional constraints ("observations") are added between different robot poses (e.g., using scan matching)
- See Tree-based network optimizer (TORO) by Grisetti *et al.*
- More popular than feature-based SLAM!

State Of The Art

- Various techniques have been applied to this non-linear least squares form of SLAM: Gauss-Newton, Levenberg-Marquardt, Gradient Descent, Stochastic Gradient Descent, Multi-Level Relaxation, Convex Relaxation, Powell's dogleg (Trust Region), ...
- Very efficient and accurate Full SLAM solutions
- Full SLAM is no longer an offline/batch approach (e.g., see iSAM by Kaess *et al.*)

Pose-graphs

- Landmarks are omitted from the state vector and instead, additional constraints ("observations") are added between different robot poses (e.g., using scan matching)
- See Tree-based network optimizer (TORO) by Grisetti *et al.*
- More popular than feature-based SLAM!

State Of The Art

- Various techniques have been applied to this non-linear least squares form of SLAM: Gauss-Newton, Levenberg-Marquardt, Gradient Descent, Stochastic Gradient Descent, Multi-Level Relaxation, Convex Relaxation, Powell's dogleg (Trust Region), . . .
- Very efficient and accurate Full SLAM solutions
- Full SLAM is no longer an offline/batch approach (e.g., see iSAM by Kaess *et al.*)

Pose-graphs

- Landmarks are omitted from the state vector and instead, additional constraints ("observations") are added between different robot poses (e.g., using scan matching)
- See Tree-based network optimizer (TORO) by Grisetti *et al.*
- More popular than feature-based SLAM!

State Of The Art

- Various techniques have been applied to this non-linear least squares form of SLAM: Gauss-Newton, Levenberg-Marquardt, Gradient Descent, Stochastic Gradient Descent, Multi-Level Relaxation, Convex Relaxation, Powell's dogleg (Trust Region), ...
- Very efficient and accurate Full SLAM solutions
- Full SLAM is no longer an offline/batch approach (e.g., see iSAM by Kaess *et al.*)

Pose-graphs

- Landmarks are omitted from the state vector and instead, additional constraints ("observations") are added between different robot poses (e.g., using scan matching)
- See Tree-based network optimizer (TORO) by Grisetti *et al.*
- More popular than feature-based SLAM!

Outline

1. Introduction
   - SLAM Problem
   - Probabilistic Methods
   - Bayesian Filtering
2. SLAM: Past and Present
   - History
   - EKF-SLAM
   - RBPF-SLAM (FastSLAM, . . .)
   - SEIF
   - Optimization Approach (GraphSLAM, SAM, . . .)
   - Softwares and Datasets
   - Final Remarks

Softwares and Datasets

Open Source Software Packages

- g2o: A General Framework for Graph Optimization (github.com/RainerKuemmerle/g2o)
- iSAM (people.csail.mit.edu/kaess/isam)
- GTSAM (borg.cc.gatech.edu/download)
- MRPT (mrpt.org)
- Check out OpenSlam.org and ROS.org for more free/open source software packages

Dataset Repositories

- MRPT (mrpt.org/robotic_datasets)
- Radish (radish.sf.net)
- Check out available SLAM software packages for preprocessed datasets

Outline

1. Introduction
   - SLAM Problem
   - Probabilistic Methods
   - Bayesian Filtering
2. SLAM: Past and Present
   - History
   - EKF-SLAM
   - RBPF-SLAM (FastSLAM, . . .)
   - SEIF
   - Optimization Approach (GraphSLAM, SAM, . . .)
   - Softwares and Datasets
   - Final Remarks

Final Remarks

- After about 25 years, SLAM is still an *active* research area in Autonomous Mobile Robotics (ICRA 2012 had two SLAM sessions)
- We have now accurate and efficient SLAM solutions for large-scale environments
- Data Association was not addressed in this presentation: still a big challenge!

Ongoing Research Topics

- 3D SLAM
- Cooperative SLAM
- Life-long SLAM
- Robustness against wrong data association
- Local/Global convergence properties
- ...

Final Remarks

- After about 25 years, SLAM is still an *active* research area in Autonomous Mobile Robotics (ICRA 2012 had two SLAM sessions)
- We have now accurate and efficient SLAM solutions for large-scale environments
- Data Association was not addressed in this presentation: still a big challenge!

Ongoing Research Topics

- 3D SLAM
- Cooperative SLAM
- Life-long SLAM
- Robustness against wrong data association
- Local/Global convergence properties
- ...

Final Remarks

- After about 25 years, SLAM is still an *active* research area in Autonomous Mobile Robotics (ICRA 2012 had two SLAM sessions)
- We have now accurate and efficient SLAM solutions for large-scale environments
- Data Association was not addressed in this presentation: still a big challenge!

Ongoing Research Topics

- 3D SLAM
- Cooperative SLAM
- Life-long SLAM
- Robustness against wrong data association
- Local/Global convergence properties
- ...

Final Remarks

- After about 25 years, SLAM is still an *active* research area in Autonomous Mobile Robotics (ICRA 2012 had two SLAM sessions)
- We have now accurate and efficient SLAM solutions for large-scale environments
- Data Association was not addressed in this presentation: still a big challenge!

Ongoing Research Topics

- 3D SLAM
- Cooperative SLAM
- Life-long SLAM
- Robustness against wrong data association
- Local/Global convergence properties
- ...

Final Remarks

- After about 25 years, SLAM is still an *active* research area in Autonomous Mobile Robotics (ICRA 2012 had two SLAM sessions)
- We have now accurate and efficient SLAM solutions for large-scale environments
- Data Association was not addressed in this presentation: still a big challenge!

Ongoing Research Topics

- 3D SLAM
- Cooperative SLAM
- Life-long SLAM
- Robustness against wrong data association
- Local/Global convergence properties
- ...

Final Remarks

- After about 25 years, SLAM is still an *active* research area in Autonomous Mobile Robotics (ICRA 2012 had two SLAM sessions)
- We have now accurate and efficient SLAM solutions for large-scale environments
- Data Association was not addressed in this presentation: still a big challenge!

Ongoing Research Topics

- 3D SLAM
- Cooperative SLAM
- Life-long SLAM
- Robustness against wrong data association
- Local/Global convergence properties
- ...

Final Remarks

- After about 25 years, SLAM is still an *active* research area in Autonomous Mobile Robotics (ICRA 2012 had two SLAM sessions)
- We have now accurate and efficient SLAM solutions for large-scale environments
- Data Association was not addressed in this presentation: still a big challenge!

Ongoing Research Topics

- 3D SLAM
- Cooperative SLAM
- Life-long SLAM
- Robustness against wrong data association
- Local/Global convergence properties
- ...

Final Remarks

- After about 25 years, SLAM is still an *active* research area in Autonomous Mobile Robotics (ICRA 2012 had two SLAM sessions)
- We have now accurate and efficient SLAM solutions for large-scale environments
- Data Association was not addressed in this presentation: still a big challenge!

Ongoing Research Topics

- 3D SLAM
- Cooperative SLAM
- Life-long SLAM
- Robustness against wrong data association
- Local/Global convergence properties
- ...

Final Remarks

- After about 25 years, SLAM is still an *active* research area in Autonomous Mobile Robotics (ICRA 2012 had two SLAM sessions)
- We have now accurate and efficient SLAM solutions for large-scale environments
- Data Association was not addressed in this presentation: still a big challenge!

Ongoing Research Topics

- 3D SLAM
- Cooperative SLAM
- Life-long SLAM
- Robustness against wrong data association
- Local/Global convergence properties
- ...

Thank you for your attention