Robot Mapping

Least Squares

Gian Diego Tipaldi, Luciano Spinello, Wolfram Burgard
Three Main SLAM Paradigms

Kalman filter

Particle filter

Graph-based

least squares approach to SLAM
Least Squares in General

- Approach for computing a solution for an **overdetermined system**
- “More equations than unknowns”
- Minimizes the **sum of the squared errors** in the equations
- Standard approach to a large set of problems
Least Squares History

- Method developed by Carl Friedrich Gauss in 1795 (he was 18 years old)
- First showcase: predicting the future location of the asteroid Ceres in 1801

Courtesy: Astronomische Nachrichten, 1828
Problem

- Given a system described by a set of \( n \) observation functions \( \{ f_i(x) \}_{i=1:n} \)

- Let
  - \( x \) be the state vector
  - \( z_i \) be a measurement of the state \( x \)
  - \( \hat{z}_i = f_i(x) \) be a function which maps \( x \) to a predicted measurement \( \hat{z}_i \)

- Given \( n \) noisy measurements \( z_{1:n} \) about the state \( x \)

**Goal:** Estimate the state \( x \) which bests explains the measurements \( z_{1:n} \)
Graphical Explanation

\[ f_1(x) = \hat{z}_1 \quad z_1 \]
\[ f_2(x) = \hat{z}_2 \quad z_2 \]
\[ \ldots \]
\[ f_n(x) = \hat{z}_n \quad z_n \]

- state (unknown)
- predicted measurements
- real measurements
Example

- $x$ position of 3D features
- $z_i$ coordinates of the 3D features projected on camera images
- Estimate the most likely 3D position of the features based on the image projections (given the camera poses)
Error Function

- Error \( e_i \) is typically the **difference** between the predicted and actual measurement

\[
e_i(x) = z_i - f_i(x)
\]

- We assume that the error has **zero mean** and is **normally distributed**
- Gaussian error with information matrix \( \Omega_i \)
- The squared error of a measurement depends only on the state and is a scalar

\[
e_i(x) = e_i(x)^T \Omega_i e_i(x)
\]
Goal: Find the Minimum

- Find the state $x^*$ which minimizes the error given all measurements

\[
x^* = \arg\min_x F(x)
\]

\[
= \arg\min_x \sum_i e_i(x)
\]

\[
= \arg\min_x \sum_i e_i^T(x)\Omega_i e_i(x)
\]

- Global error (scalar)
- Squared error terms (scalar)
- Error terms (vector)
Goal: Find the Minimum

- Find the state $x^*$ which minimizes the error given all measurements

$$x^* = \arg\min_x \sum_{i} e_i^T(x) \Omega_i e_i(x)$$

- A general solution is to derive the global error function and find its nulls

- In general complex and no closed form solution

→ Numerical approaches
Assumption

- A “good” initial guess is available
- The error functions are “smooth” in the neighborhood of the (hopefully global) minima

- Then, we can solve the problem by iterative local linearizations
Solve Via Iterative Local Linearizations

- Linearize the error terms around the current solution/initial guess
- Compute the first derivative of the squared error function
- Set it to zero and solve linear system
- Obtain the new state (that is hopefully closer to the minimum)
- Iterate
Linearizing the Error Function

- Approximate the error functions around an initial guess \( \hat{x} \) via Taylor expansion

\[
e_i(\hat{x} + \Delta x) \approx e_i(\hat{x}) + J_i(\hat{x}) \Delta x
\]

- Reminder: Jacobian

\[
J_f(x) = \begin{pmatrix}
\frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \ldots & \frac{\partial f_1(x)}{\partial x_n} \\
\frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \ldots & \frac{\partial f_2(x)}{\partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_m(x)}{\partial x_1} & \frac{\partial f_m(x)}{\partial x_2} & \ldots & \frac{\partial f_m(x)}{\partial x_n}
\end{pmatrix}
\]
Squared Error

- With the previous linearization, we can fix $x$ and carry out the minimization in the increments $\Delta x$.
- We replace the Taylor expansion in the squared error terms:

$$ e_i(\hat{x} + \Delta x) = \ldots $$
Squared Error

- With the previous linearization, we can fix $\mathbf{x}$ and carry out the minimization in the increments $\Delta \mathbf{x}$

- We replace the Taylor expansion in the squared error terms:

$$
e_i(\hat{\mathbf{x}} + \Delta \mathbf{x}) = e_i^T(\hat{\mathbf{x}} + \Delta \mathbf{x})\Omega_i e_i(\hat{\mathbf{x}} + \Delta \mathbf{x})
\simeq (e_i + J_i \Delta \mathbf{x})^T \Omega_i (e_i + J_i \Delta \mathbf{x})
= e_i^T \Omega_i e_i + e_i^T \Omega_i J_i \Delta \mathbf{x} + \Delta \mathbf{x}^T J_i^T \Omega_i e_i + \Delta \mathbf{x}^T J_i^T \Omega_i J_i \Delta \mathbf{x}$$
Squared Error (cont.)

- All summands are scalar so the transposition has no effect
- By grouping similar terms, we obtain:

\[ e_i(\hat{x} + \Delta x) \]
\[ \simeq e_i^T \Omega_i e_i + e_i^T \Omega_i J_i \Delta x + \Delta x^T J_i^T \Omega_i e_i + \]
\[ \Delta x^T J_i^T \Omega_i J_i \Delta x \]
\[ = e_i^T \Omega_i e_i + 2 e_i^T \Omega_i J_i \Delta x + \Delta x^T J_i^T \Omega_i J_i \Delta x \]
\[ = c_i + 2b_i^T \Delta x + \Delta x^T H_i \Delta x \]
Global Error

- The global error is the sum of the squared errors terms corresponding to the individual measurements.
- Form a new expression which approximates the global error in the neighborhood of the current solution $\mathbf{x}$.

\[
F(\hat{\mathbf{x}} + \Delta \mathbf{x}) \approx \sum_i \left(c_i + b_i^T \Delta \mathbf{x} + \Delta \mathbf{x}^T H_i \Delta \mathbf{x}\right) \\
= \sum_i c_i + 2(\sum_i b_i^T) \Delta \mathbf{x} + \Delta \mathbf{x}^T (\sum_i H_i) \Delta \mathbf{x}
\]
Global Error (cont.)

\[ F(\hat{x} + \Delta x) \approx \sum_i \left( c_i + b_i^T \Delta x + \Delta x^T H_i \Delta x \right) \]
\[ = \sum_i c_i + 2 \left( \sum_i b_i^T \right) \Delta x + \Delta x^T \left( \sum_i H_i \right) \Delta x \]
\[ = c + 2b^T \Delta x + \Delta x^T H \Delta x \]

with

\[ b^T = \sum_i e_i^T \Omega_i J_i \]
\[ H = \sum_i J_i^T \Omega J_i \]
Quadratic Form

- We can write the global error terms as a quadratic form in $\Delta x$

$$F(\hat{x} + \Delta x) \simeq c + 2b^T\Delta x + \Delta x^T H \Delta x$$

- We need to compute the derivative of $F(\hat{x} + \Delta x)$ w.r.t. $\Delta x$ (given $\hat{x}$)
Deriving a Quadratic Form

- Assume a quadratic form

\[ f(x) = x^T H x + b^T x \]

- The first derivative is

\[ \frac{\partial f}{\partial x} = (H + H^T)x + b \]

See: The Matrix Cookbook, Section 2.2.4
Quadratic Form

- We can write the global error terms as a quadratic form in $\Delta x$

$$F(\hat{x} + \Delta x) \approx c + 2b^T \Delta x + \Delta x^T H \Delta x$$

- The derivative of the approximated $F(\hat{x} + \Delta x)$ w.r.t. $\Delta x$ is then:

$$\frac{\partial F(\hat{x} + \Delta x)}{\partial \Delta x} \approx 2b + 2H \Delta x$$
Minimizing the Quadratic Form

- Derivative of \( F(\hat{x} + \Delta x) \)
  \[
  \frac{\partial F(\hat{x} + \Delta x)}{\partial \Delta x} \approx 2b + 2H\Delta x
  \]

- Setting it to zero leads to
  \[
  0 = 2b + 2H\Delta x
  \]

- Which leads to the linear system
  \[
  H\Delta x = -b
  \]

- The solution for the increment \( \Delta x^* \) is
  \[
  \Delta x^* = -H^{-1}b
  \]
Gauss-Newton Solution

Iterate the following steps:

- Linearize around $\mathbf{x}$ and compute for each measurement

$$e_i(\hat{x} + \Delta x) \simeq e_i(\hat{x}) + J_i \Delta x$$

- Compute the terms for the linear system

$$b^T = \sum_i e_i^T \Omega_i J_i$$

$$H = \sum_i J_i^T \Omega_i J_i$$

- Solve the linear system

$$\Delta x^* = -H^{-1} b$$

- Updating state

$$\hat{x} \leftarrow \hat{x} + \Delta x^*$$
Example: Odometry Calibration

- Odometry measurements $u_i$
- Eliminate systematic error through calibration
- Assumption: Ground truth odometry $u_i^*$ is available
- Ground truth by motion capture, scan-matching, or a SLAM system
Example: Odometry Calibration

- There is a function $f_i(x)$ which, given some bias parameters $x$, returns an unbiased (corrected) odometry for the reading $u_i'$ as follows

$$u_i' = f_i(x) = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} u_i$$

- To obtain the correction function $f(x)$, we need to find the parameters $x$
The state vector is
\[ x = \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{21} & x_{22} & x_{23} & x_{31} & x_{32} & x_{33} \end{pmatrix}^T \]

The error function is
\[ e_i(x) = u_i^* - \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} u_i \]

Its derivative is:
\[ J_i = \frac{\partial e_i(x)}{\partial x} = -\begin{pmatrix} u_{i,x} & u_{i,y} & u_{i,\theta} \\ u_{i,x} & u_{i,y} & u_{i,\theta} \\ u_{i,x} & u_{i,y} & u_{i,\theta} \end{pmatrix} \]

Does not depend on \( x \), why? What are the consequences?

\( e \) is linear, no need to iterate!
Questions

- How do the parameters look like if the odometry is perfect?
- How many measurements (at least) are needed to find a solution for the calibration problem?
- $H$ is symmetric. Why?
- How does the structure of the measurement function affects the structure of $H$?
How to Efficiently Solve the Linear System?

- Linear system $H \Delta x = -b$
- Can be solved by matrix inversion (in theory)
- In practice:
  - Cholesky factorization
  - QR decomposition
  - Iterative methods such as conjugate gradients (for large systems)
Cholesky Decomposition for Solving a Linear System

- A symmetric and positive definite
- System to solve $Ax = b$
- Cholesky leads to $A = LL^T$ with $L$ being a lower triangular matrix
- Solve first
  \[ Ly = b \]
- and then
  \[ L^Tx = y \]
Gauss-Newton Summary

Method to minimize a squared error:

- Start with an initial guess
- Linearize the individual error functions
- This leads to a quadratic form
- One obtains a linear system by setting its derivative to zero
- Solving the linear systems leads to a state update
- Iterate
Relation to Probabilistic State Estimation

- So far, we minimized an error function.
- How does this relate to state estimation in the probabilistic sense?
### General State Estimation

- Bayes rule, independence and Markov assumptions allow us to write

\[
p(x_{0:t} \mid z_{1:t}, u_{1:t}) = \eta \cdot p(x_0) \prod_t \left[p(x_t \mid x_{t-1}, u_t) \cdot p(z_t \mid x_t)\right]
\]
Log Likelihood

- Written as the log likelihood, leads to

\[
\log p(x_{0:t} \mid z_{1:t}, u_{1:t}) = \text{const.} + \log p(x_0) + \sum_{t} \left[ \log p(x_t \mid x_{t-1}, u_t) + \log p(z_t \mid x_t) \right]
\]
Gaussian Assumption

- Assuming Gaussian distributions

\[
\log p(x_{0:t} \ | \ z_{1:t}, u_{1:t}) = \text{const.} + \log p(x_0) \\
\sum_t \left[ \log p(x_t \ | \ x_{t-1}, u_t) + \log p(z_t \ | \ x_t) \right]
\]
Log of a Gaussian

- Log likelihood of a Gaussian

\[
\log \mathcal{N}(x, \mu, \Sigma) = \text{const.} - \frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)
\]
Error Function as Exponent

- Log likelihood of a Gaussian

\[
\log \mathcal{N}(x, \mu, \Sigma) = \text{const.} - \frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)
\]

- is up to a constant equivalent to the error functions used before
Log Likelihood with Error Terms

- Assuming Gaussian distributions

\[
\log p(x_{0:t} \mid z_{1:t}, u_{1:t}) = \text{const.} - \frac{1}{2} e_p(x) - \frac{1}{2} \sum_t [e_{u_t}(x) + e_{z_t}(x)]
\]
Maximizing the Log Likelihood

- Assuming Gaussian distributions

\[
\log p(x_{0:t} \mid z_{1:t}, u_{1:t}) \\
= \text{const.} - \frac{1}{2} e_p(x) - \frac{1}{2} \sum_t [e_{u_t}(x) + e_{z_t}(x)]
\]

- Maximizing the log likelihood leads to

\[
\arg\max \log p(x_{0:t} \mid z_{1:t}, u_{1:t}) \\
= \arg\min e_p(x) + \sum_t [e_{u_t}(x) + e_{z_t}(x)]
\]
Minimizing the Squared Error is Equivalent to Maximizing the Log Likelihood of Independent Gaussian Distributions

with individual error terms for the motions, measurements, and prior:

\[
\arg\max \log p(x_{0:t} \mid z_{1:t}, u_{1:t}) = \arg\min e_p(x) + \sum_t \left[ e_{u_t}(x) + e_{z_t}(x) \right]
\]
Summary

- Technique to minimize squared error functions
- Gauss-Newton is an iterative approach for non-linear problems
- Uses linearization (approximation!)
- Equivalent to maximizing the log likelihood of independent Gaussians
- Popular method in a lot of disciplines
Literature

Least Squares and Gauss-Newton

- Basically every textbook on numeric calculus or optimization
- Wikipedia (for a brief summary)

Relation to Probability Theory

- Thrun et al.: “Probabilistic Robotics”, Chapter 11.4
Slide Information

- These slides have been created by Cyrill Stachniss as part of the robot mapping course taught in 2012/13 and 2013/14. I created this set of slides partially extending existing material of Edwin Olson, Pratik Agarwal, and myself.

- I tried to acknowledge all people that contributed image or video material. In case I missed something, please let me know. If you adapt this course material, please make sure you keep the acknowledgements.

- Feel free to use and change the slides. If you use them, I would appreciate an acknowledgement as well. To satisfy my own curiosity, I appreciate a short email notice in case you use the material in your course.

- My video recordings are available through YouTube: http://www.youtube.com/playlist?list=PLgnQpQtFTOGQrZ4O5Qzb1Hgl3b1JHimN&_feature=g-list

Cyrill Stachniss, 2014
cyrill.stachniss@igg.uni-bonn.de