# Robot Mapping

## Least Squares

Cyrill Stachniss



## 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 $\{f_i(\mathbf{x})\}_{i=1:n}$
- Let
  - $\mathbf{x}$ be the state vector
  - $\mathbf{z}_i$ be a measurement of the state $\mathbf{x}$
  - $\hat{\mathbf{z}}_i = f_i(\mathbf{x})$ be a function which maps $\mathbf{x}$ to a predicted measurement $\hat{\mathbf{z}}_i$
- Given n noisy measurements $\mathbf{z}_{1:n}$ about the state $\mathbf{x}$
- **Goal:** Estimate the state $\mathbf{x}$ which bests explains the measurements $\mathbf{z}_{1:n}$



## Graphical Explanation

$$\mathbf{x} \quad \begin{array}{ll} f_1(\mathbf{x}) = \hat{\mathbf{z}}_1 & \mathbf{z}_1 \\ f_2(\mathbf{x}) = \hat{\mathbf{z}}_2 & \mathbf{z}_2 \\ \quad\ldots & \\ f_n(\mathbf{x}) = \hat{\mathbf{z}}_n & \mathbf{z}_n \end{array}$$

state (unknown) | predicted measurements | real measurements



## Example

$$\mathbf{x} \quad \begin{array}{ll} f_1(\mathbf{x}) = \hat{\mathbf{z}}_1 & \mathbf{z}_1 \\ f_2(\mathbf{x}) = \hat{\mathbf{z}}_2 & \mathbf{z}_2 \\ \quad\ldots & \\ f_n(\mathbf{x}) = \hat{\mathbf{z}}_n & \mathbf{z}_n \end{array}$$

- $\mathbf{x}$ position of 3D features
- $\mathbf{z}_i$ 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 $\mathbf{e}_i$ is typically the **difference** between the **predicted and actual** measurement

$$\mathbf{e}_i(\mathbf{x}) \quad = \quad \mathbf{z}_i - f_i(\mathbf{x})$$

- We assume that the error has **zero mean** and is **normally distributed**
- Gaussian error with information matrix $\mathbf{\Omega}_i$
- The squared error of a measurement depends only on the state and is a scalar

$$e_i(\mathbf{x}) \quad = \quad \mathbf{e}_i(\mathbf{x})^T \mathbf{\Omega}_i \mathbf{e}_i(\mathbf{x})$$

## Goal: Find the Minimum

- Find the state $\mathbf{x}^*$ which minimizes the error given all measurements

$$\begin{aligned}
\mathbf{x}^* &= \underset{\mathbf{x}}{\operatorname{argmin}}\, F(\mathbf{x}) \longleftarrow \boxed{\text{global error (scalar)}} \\
&= \underset{\mathbf{x}}{\operatorname{argmin}} \sum_i e_i(\mathbf{x}) \longleftarrow \boxed{\text{squared error terms (scalar)}} \\
&= \underset{\mathbf{x}}{\operatorname{argmin}} \sum_i \mathbf{e}_i^T(\mathbf{x}) \boldsymbol{\Omega}_i \mathbf{e}_i(\mathbf{x})
\end{aligned}$$

error terms (vector)



## Goal: Find the Minimum

- Find the state $\mathbf{x}^*$ which minimizes the error given all measurements

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_i \mathbf{e}_i^T(\mathbf{x}) \boldsymbol{\Omega}_i \mathbf{e}_i(\mathbf{x})$$

- 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 $\mathbf{x}$ via Taylor expansion

$$\mathbf{e}_i(\mathbf{x} + \Delta\mathbf{x}) \simeq \underbrace{\mathbf{e}_i(x)}_{\mathbf{e}_i} + \mathbf{J}_i(\mathbf{x})\Delta\mathbf{x}$$

- Reminder: Jacobian

$$\mathbf{J}_f(x) = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \cdots & \frac{\partial f_2(x)}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_m(x)}{\partial x_1} & \frac{\partial f_m(x)}{\partial x_2} & \cdots & \frac{\partial f_m(x)}{\partial x_n} \end{pmatrix}$$



## 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:

$$e_i(\mathbf{x} + \Delta\mathbf{x}) = \ldots$$



## 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:

$$\begin{aligned} e_i(\mathbf{x} + \Delta\mathbf{x}) &= \mathbf{e}_i^T(\mathbf{x} + \Delta\mathbf{x})\boldsymbol{\Omega}_i\mathbf{e}_i(\mathbf{x} + \Delta\mathbf{x}) \\ &\simeq (\mathbf{e}_i + \mathbf{J}_i\Delta\mathbf{x})^T\boldsymbol{\Omega}_i(\mathbf{e}_i + \mathbf{J}_i\Delta\mathbf{x}) \\ &= \mathbf{e}_i^T\boldsymbol{\Omega}_i\mathbf{e}_i + \\ &\quad\ \mathbf{e}_i^T\boldsymbol{\Omega}_i\mathbf{J}_i\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{J}_i^T\boldsymbol{\Omega}_i\mathbf{e}_i + \\ &\quad\ \Delta\mathbf{x}^T\mathbf{J}_i^T\boldsymbol{\Omega}_i\mathbf{J}_i\Delta\mathbf{x} \end{aligned}$$



## Squared Error (cont.)

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

$$\begin{aligned} & e_i(\mathbf{x} + \Delta\mathbf{x}) \\ &\quad \simeq \mathbf{e}_i^T\boldsymbol{\Omega}_i\mathbf{e}_i + \\ &\quad\quad\ \mathbf{e}_i^T\boldsymbol{\Omega}_i\mathbf{J}_i\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{J}_i^T\boldsymbol{\Omega}_i\mathbf{e}_i + \\ &\quad\quad\ \Delta\mathbf{x}^T\mathbf{J}_i^T\boldsymbol{\Omega}_i\mathbf{J}_i\Delta\mathbf{x} \\ &\quad = \underbrace{\mathbf{e}_i^T\boldsymbol{\Omega}_i\mathbf{e}_i}_{c_i} + 2\underbrace{\mathbf{e}_i^T\boldsymbol{\Omega}_i\mathbf{J}_i}_{\mathbf{b}_i^T}\Delta\mathbf{x} + \Delta\mathbf{x}^T\underbrace{\mathbf{J}_i^T\boldsymbol{\Omega}_i\mathbf{J}_i}_{\mathbf{H}_i}\Delta\mathbf{x} \\ &\quad = c_i + 2\mathbf{b}_i^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}_i\Delta\mathbf{x} \end{aligned}$$

## 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 $\mathbf{x}$

$$
\begin{aligned}
F(\mathbf{x}+\Delta\mathbf{x}) &\simeq \sum_i \left(c_i + \mathbf{b}_i^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}_i\Delta\mathbf{x}\right) \\
&= \sum_i c_i + 2(\sum_i \mathbf{b}_i^T)\Delta\mathbf{x} + \Delta\mathbf{x}^T(\sum_i \mathbf{H}_i)\Delta\mathbf{x}
\end{aligned}
$$



## Global Error (cont.)

$$
\begin{aligned}
F(\mathbf{x}+\Delta\mathbf{x}) &\simeq \sum_i \left(c_i + \mathbf{b}_i^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}_i\Delta\mathbf{x}\right) \\
&= \underbrace{\sum_i c_i}_{c} + 2\,\underbrace{(\sum_i \mathbf{b}_i^T)}_{\mathbf{b}^T}\,\Delta\mathbf{x} + \Delta\mathbf{x}^T\,\underbrace{(\sum_i \mathbf{H}_i)}_{\mathbf{H}}\,\Delta\mathbf{x} \\
&= c + 2\mathbf{b}^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}\Delta\mathbf{x}
\end{aligned}
$$

with

$$
\begin{aligned}
\mathbf{b}^T &= \sum_i \mathbf{e}_i^T\Omega_i\mathbf{J}_i \\
\mathbf{H} &= \sum_i \mathbf{J}_i^T\Omega\mathbf{J}_i
\end{aligned}
$$



## Quadratic Form

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

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

- We need to compute the derivative of $F(\mathbf{x}+\Delta\mathbf{x})$ w.r.t. $\Delta\mathbf{x}$ (given $\mathbf{x}$)



## Deriving a Quadratic Form

- Assume a quadratic form

$$
f(\mathbf{x}) = \mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{b}^T\mathbf{x}
$$

- The first derivative is

$$
\frac{\partial f}{\partial \mathbf{x}} = (\mathbf{H}+\mathbf{H}^T)\mathbf{x} + \mathbf{b}
$$

See: The Matrix Cookbook, Section 2.2.4

## Quadratic Form

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

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

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

$$\frac{\partial F(\mathbf{x}+\Delta\mathbf{x})}{\partial \Delta\mathbf{x}} \simeq 2\mathbf{b} + 2\mathbf{H}\Delta\mathbf{x}$$



## Minimizing the Quadratic Form

- Derivative of $F(\mathbf{x}+\Delta\mathbf{x})$

$$\frac{\partial F(\mathbf{x}+\Delta\mathbf{x})}{\partial \Delta\mathbf{x}} \simeq 2\mathbf{b} + 2\mathbf{H}\Delta\mathbf{x}$$

- Setting it to zero leads to

$$0 = 2\mathbf{b} + 2\mathbf{H}\Delta\mathbf{x}$$

- Which leads to the linear system

$$\mathbf{H}\Delta\mathbf{x} = -\mathbf{b}$$

- The solution for the increment $\Delta\mathbf{x}^*$ is

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



## Gauss-Newton Solution

**Iterate the following steps:**

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

$$\mathbf{e}_i(\mathbf{x}+\Delta\mathbf{x}) \simeq \mathbf{e}_i(\mathbf{x}) + \mathbf{J}_i\Delta\mathbf{x}$$

- Compute the terms for the linear system $\mathbf{b}^T = \sum_i \mathbf{e}_i^T\Omega_i\mathbf{J}_i \qquad \mathbf{H} = \sum_i \mathbf{J}_i^T\Omega_i\mathbf{J}_i$
- Solve the linear system

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

- Updating state $\mathbf{x} \leftarrow \mathbf{x} + \Delta\mathbf{x}^*$



## Example: Odometry Calibration

- Odometry measurements $\mathbf{u}_i$
- Eliminate systematic error through calibration
- Assumption: Ground truth odometry $\mathbf{u}_i^*$ is available
- Ground truth by motion capture, scan-matching, or a SLAM system

## Example: Odometry Calibration

- There is a function $f_i(\mathbf{x})$ which, given some bias parameters $\mathbf{x}$, returns a an unbiased (corrected) odometry for the reading $\mathbf{u}'_i$ as follows

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

- To obtain the correction function $f(\mathbf{x})$, we need to find the parameters $\mathbf{x}$



## Odometry Calibration (cont.)

- The state vector is

$$\mathbf{x} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{21} & x_{22} & x_{23} & x_{31} & x_{32} & x_{33} \end{pmatrix}^T$$

- The error function is

$$\mathbf{e}_i(\mathbf{x}) = \mathbf{u}^*_i - \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \mathbf{u}_i$$

- Its derivative is:

$$\mathbf{J}_i = \frac{\partial \mathbf{e}_i(\mathbf{x})}{\partial \mathbf{x}} = - \begin{pmatrix} u_{i,x} & u_{i,y} & u_{i,\theta} & & & & & & \\ & & & u_{i,x} & u_{i,y} & u_{i,\theta} & & & \\ & & & & & & u_{i,x} & u_{i,y} & u_{i,\theta} \end{pmatrix}$$

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?
- $\mathbf{H}$ is symmetric. Why?
- How does the structure of the measurement function affects the structure of $\mathbf{H}$?



## How to Efficiently Solve the Linear System?

- Linear system $\mathbf{H}\Delta\mathbf{x} = -\mathbf{b}$
- 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

- $\mathbf{A}$ symmetric and positive definite
- System to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$
- Cholesky leads to $\mathbf{A} = \mathbf{L}\mathbf{L}^T$ with $\mathbf{L}$ being a lower triangular matrix
- Solve first

$$\mathbf{L}\mathbf{y} = \mathbf{b}$$

- an then

$$\mathbf{L}^T\mathbf{x} = \mathbf{y}$$



## 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 settings 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

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

## Log Likelihood

- Written as the log likelihood, leads to

$$\begin{aligned}\log p(x_{0:t} \mid z_{1:t}, u_{1:t}) \\ &= \text{const.} + \log p(x_0) \\ &+ \sum_t \left[\log p(x_t \mid x_{t-1}, u_t) + \log p(z_t \mid x_t)\right]\end{aligned}$$



## Gaussian Assumption

- Assuming Gaussian distributions

$$\begin{aligned}\log p(x_{0:t} \mid z_{1:t}, u_{1:t}) \\ &= \text{const.} + \log \underbrace{p(x_0)}_{\mathcal{N}} \\ &+ \sum_t \left[\log \underbrace{p(x_t \mid x_{t-1}, u_t)}_{\mathcal{N}} + \log \underbrace{p(z_t \mid x_t)}_{\mathcal{N}}\right]\end{aligned}$$



## Log of a Gaussian

- Log likelihood of a Gaussian

$$\begin{aligned}\log \mathcal{N}(x, \mu, \Sigma) \\ &= \text{const.} - \frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\end{aligned}$$



## Error Function as Exponent

- Log likelihood of a Gaussian

$$\begin{aligned}\log \mathcal{N}(x, \mu, \Sigma) \\ &= \text{const.} - \frac{1}{2} \underbrace{\underbrace{(x-\mu)^T}_{\mathbf{e}^T(x)} \underbrace{\Sigma^{-1}}_{\Omega} \underbrace{(x-\mu)}_{\mathbf{e}(x)}}_{e(x)}\end{aligned}$$

- is up to a constant equivalent to the error functions used before

## Log Likelihood with Error Terms

- Assuming Gaussian distributions

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



## Maximizing the Log Likelihood

- Assuming Gaussian distributions

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

- Maximizing the log likelihood leads to

$$\begin{aligned}\operatorname{argmax} \log p(x_{0:t} \mid z_{1:t}, u_{1:t}) \\ = \quad \operatorname{argmin} e_p(x) + \sum_t \left[ e_{u_t}(x) + e_{z_t}(x) \right]\end{aligned}$$



## 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:

$$\begin{aligned}\operatorname{argmax} \log p(x_{0:t} \mid z_{1:t}, u_{1:t}) \\ = \quad \operatorname{argmin} e_p(x) + \sum_t \left[ e_{u_t}(x) + e_{z_t}(x) \right]\end{aligned}$$



## 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