Robot Mapping

Graph-Based SLAM with Landmarks

Cyrill Stachniss
Graph-Based SLAM (Chap. 15)

- Use a **graph** to represent the problem
- Every **node** in the graph corresponds to a pose of the robot during mapping
- Every **edge** between two nodes corresponds to a spatial constraint between them
- **Graph-Based SLAM:** Build the graph and find a node configuration that minimize the error introduced by the constraints
The Graph

So far:
- Vertices for robot poses \((x, y, \theta)\)
- Edges for virtual observations (transformations) between robot poses

Topic today:
- How to deal with landmarks
Landmark-Based SLAM
Real Landmark Map Example

Image courtesy: E. Nebot
The Graph with Landmarks

- Feature
- Pose
- Constraint
The Graph with Landmarks

- **Nodes** can represent:
  - Robot poses
  - Landmark locations

- **Edges** can represent:
  - Landmark observations
  - Odometry measurements

- The minimization optimizes the landmark locations and robot poses
2D Landmarks

- Landmark is a \((x, y)\)-point in the world
- Relative observation in \((x, y)\)
Landmarks Observation

- Expected observation (x-y sensor)

\[ \hat{z}_{ij}(x_i, x_j) = R_i^T(x_j - t_i) \]

robot \quad landmark \quad robot translation
Landmarks Observation

- Expected observation (x-y sensor)

\[
\hat{z}_{ij}(x_i, x_j) = R_i^T(x_j - t_i)
\]

- Error function

\[
e_{ij}(x_i, x_j) = \hat{z}_{ij} - z_{ij} = R_i^T(x_j - t_i) - z_{ij}
\]
Bearing Only Observations

- A landmark is still a 2D point
- The robot observe only the bearing (orientation towards the landmark)
- Observation function

\[
\hat{z}_{ij}(x_i, x_j) = \arctan \left( \frac{(x_j - t_i)_y}{(x_j - t_i)_x} \right) - \theta_i
\]
Bearing Only Observations

- Observation function

\[ \hat{z}_{ij}(x_i, x_j) = \arctan\left(\frac{(x_j - t_i) \cdot y}{(x_j - t_i) \cdot x}\right) - \theta_i \]

- Error function

\[ e_{ij}(x_i, x_j) = \arctan\left(\frac{(x_j - t_i) \cdot y}{(x_j - t_i) \cdot x}\right) - \theta_i - z_j \]
The Rank of the Matrix $H$

- What is the rank of $H_{ij}$ for a 2D landmark-pose constraint?
The Rank of the Matrix $H$

- What is the rank of $H_{ij}$ for a 2D landmark-pose constraint?
  - The blocks of $J_{ij}$ are a 2x3 matrices
  - $H_{ij}$ cannot have more than rank 2

$$\text{rank}(A^TA) = \text{rank}(A^T) = \text{rank}(A)$$
The Rank of the Matrix H

- What is the rank of $H_{ij}$ for a 2D landmark-pose constraint?
  - The blocks of $J_{ij}$ are a 2x3 matrices
  - $H_{ij}$ cannot have more than rank 2
    \[ \text{rank}(A^T A) = \text{rank}(A^T) = \text{rank}(A) \]

- What is the rank of $H_{ij}$ for a bearing-only constraint?
The Rank of the Matrix $H$

- What is the rank of $H_{ij}$ for a 2D landmark-pose constraint?
  - The blocks of $J_{ij}$ are a 2x3 matrices
  - $H_{ij}$ cannot have more than rank 2
    $$\text{rank}(A^TA) = \text{rank}(A^T) = \text{rank}(A)$$

- What is the rank of $H_{ij}$ for a bearing-only constraint?
  - The blocks of $J_{ij}$ are a 1x3 matrices
  - $H_{ij}$ has rank 1
Where is the Robot?

- Robot observes one landmark \((x, y)\)
- Where can the robot be?

The robot can be somewhere on a circle around the landmark

It is a 1D solution space (constrained by the distance and the robot’s orientation)
Where is the Robot?

- Robot observes one landmark (bearing-only)
- Where can the robot be?

The robot can be anywhere in the x-y plane

It is a 2D solution space (constrained by the robot’s orientation)
Rank

- In landmark-based SLAM, the system can be under-determined
- The rank of $\mathbf{H}$ is less or equal to the sum of the ranks of the constraints
- To determine a unique solution, the system must have full rank
Questions

- The rank of $H$ is \textbf{less or equal} to the sum of the ranks of the constraints
- To determine a \textbf{unique solution}, the system must have \textbf{full rank}

Questions:
- How many 2D landmark observations are needed to resolve for a robot pose?
- How many bearing-only observations are needed to resolve for a robot pose?
Under-Determined Systems

- No guarantee for a full rank system
  - Landmarks may be observed only once
  - Robot might have no odometry
- We can still deal with these situations by adding a "damping" factor to $H$
- Instead of solving $H \Delta x = -b$, we solve
  $$(H + \lambda I) \Delta x = -b$$

What is the effect of that?
\[(H + \lambda I) \Delta x = -b\]

- Damping factor for \(H\)
- \((H + \lambda I)\Delta x = -b\)
- The damping factor \(\lambda I\) makes the system positive definite
- It adds an additional constraints that “drag” the increments towards 0
- What happens when \(\lambda \gg \text{det}(H)\) ?
**Simplified Levenberg Marquardt**

- Damping to regulate the convergence using backup/restore actions

  - \( x \): the initial guess
  - while (! converged)
    - \( \lambda = \lambda_{\text{init}} \)
    - \( <H, b> = \text{buildLinearSystem}(x) \)
    - \( E = \text{error}(x) \)
    - \( x_{\text{old}} = x \)
    - \( \Delta x = \text{solveSparse}((H + \lambda I) \Delta x = -b) \)
    - \( x += \Delta x; \)
    - If \( (E < \text{error}(x)) \) \{ \n      - \( x = x_{\text{old}}; \)
      - \( \lambda *= 2; \)
    \} else \{ \lambda /= 2; \}
Summary

- Graph-Based SLAM for landmarks
- The rank of $H$ matters
- Levenberg Marquardt for optimization