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

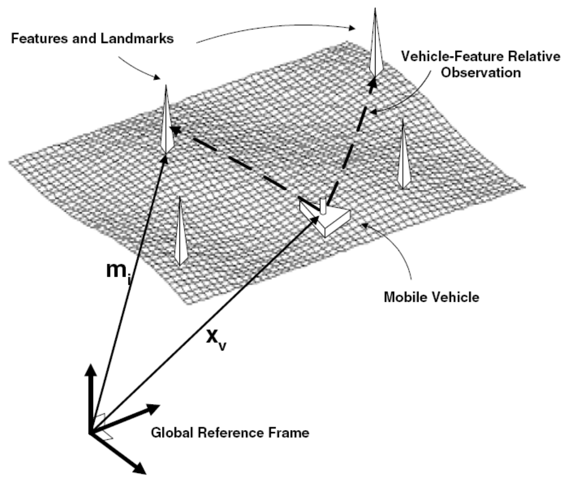
So far:

- Vertices for robot poses (x, y, θ)
- Edges for virtual observations (transformations) between robot poses

Topic today:

- How to deal with landmarks

Landmark-Based SLAM



Real Landmark Map Example

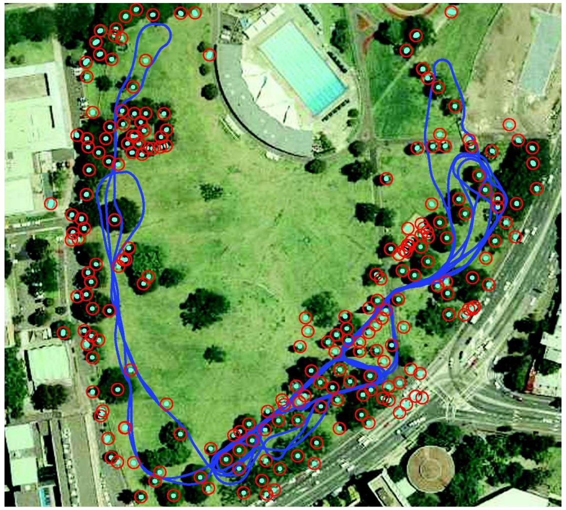
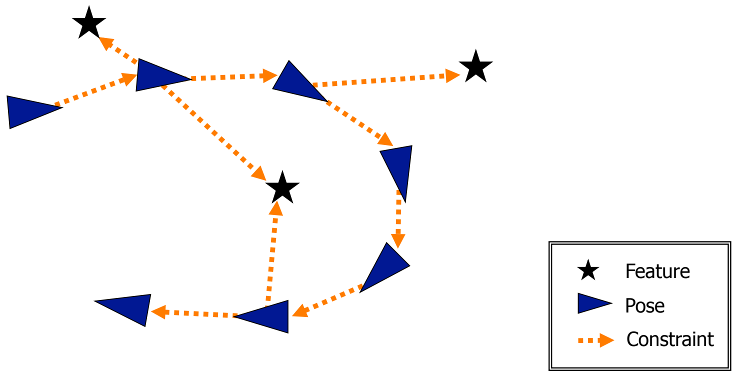


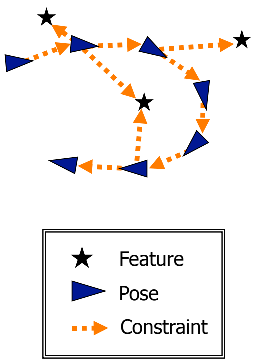
Image courtesy: E. Nebot

The Graph with Landmarks



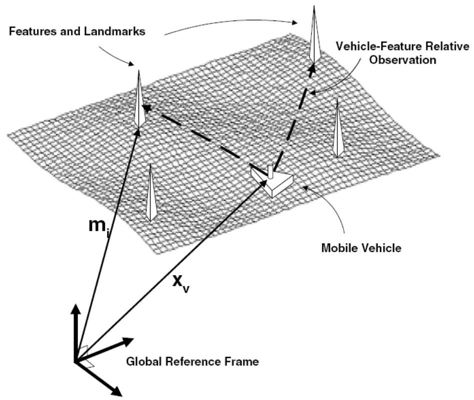
The Graph with Landmarks

- **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)

$$\hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{R}_i^T(\mathbf{x}_j - \mathbf{t}_i)$$

robot — landmark — robot translation

Landmarks Observation

- Expected observation (x-y sensor)

$$\hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{R}_i^T(\mathbf{x}_j - \mathbf{t}_i)$$

robot — landmark — robot translation

- Error function

$$\begin{aligned} \mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) &= \hat{\mathbf{z}}_{ij} - \mathbf{z}_{ij} \\ &= \mathbf{R}_i^T(\mathbf{x}_j - \mathbf{t}_i) - \mathbf{z}_{ij} \end{aligned}$$

Bearing Only Observations

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

$$\hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \text{atan}\frac{(\mathbf{x}_j - \mathbf{t}_i).y}{(\mathbf{x}_j - \mathbf{t}_i).x} - \theta_i$$

robot — landmark — robot-landmark angle — robot orientation

Bearing Only Observations

- Observation function

$$\hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \text{atan}\frac{(\mathbf{x}_j - \mathbf{t}_i).y}{(\mathbf{x}_j - \mathbf{t}_i).x} - \theta_i$$

robot — landmark — robot-landmark angle — robot orientation

- Error function

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \text{atan}\frac{(\mathbf{x}_j - \mathbf{t}_i).y}{(\mathbf{x}_j - \mathbf{t}_i).x} - \theta_i - \mathbf{z}_j$$

The Rank of the Matrix H

- What is the rank of $\mathbf{H}_{ij}$ for a 2D landmark-pose constraint?

The Rank of the Matrix H

- What is the rank of $\mathbf{H}_{ij}$ for a 2D landmark-pose constraint?
  - The blocks of $\mathbf{J}_{ij}$ are a 2x3 matrices
  - $\mathbf{H}_{ij}$ cannot have more than rank 2

    $\mathrm{rank}(A^T A) = \mathrm{rank}(A^T) = \mathrm{rank}(A)$

The Rank of the Matrix H

- What is the rank of $\mathbf{H}_{ij}$ for a 2D landmark-pose constraint?
  - The blocks of $\mathbf{J}_{ij}$ are a 2x3 matrices
  - $\mathbf{H}_{ij}$ cannot have more than rank 2

    $\mathrm{rank}(A^T A) = \mathrm{rank}(A^T) = \mathrm{rank}(A)$

- What is the rank of $\mathbf{H}_{ij}$ for a bearing-only constraint?

The Rank of the Matrix H

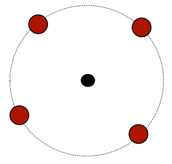
- What is the rank of $\mathbf{H}_{ij}$ for a 2D landmark-pose constraint?
  - The blocks of $\mathbf{J}_{ij}$ are a 2x3 matrices
  - $\mathbf{H}_{ij}$ cannot have more than rank 2

    $\mathrm{rank}(A^T A) = \mathrm{rank}(A^T) = \mathrm{rank}(A)$

- What is the rank of $\mathbf{H}_{ij}$ for a bearing-only constraint?
  - The blocks of $\mathbf{J}_{ij}$ are a 1x3 matrices
  - $\mathbf{H}_{ij}$ has rank 1

Where is the Robot?

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

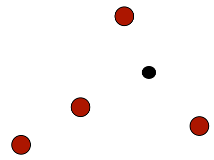


The robot can be somewhere on a circle around the landmark

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 $\mathbf{H}$ 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:**
  - 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 $\mathbf{H}$
- Instead of solving $\mathbf{H}\Delta\mathbf{x} = -\mathbf{b}$, we solve

$$(\mathbf{H} + \lambda\mathbf{I})\Delta\mathbf{x} = -\mathbf{b}$$

What is the effect of that?

(H + λ I) Δx = -b

- Damping factor for $\mathbf{H}$
- $(\mathbf{H} + \lambda\mathbf{I})\Delta\mathbf{x} = -\mathbf{b}$
- The damping factor $\lambda\mathbf{I}$ makes the system positive definite
- It adds an additional constraints that "drag" the increments towards 0
- What happens when $\lambda >> \det(\mathbf{H})$?

Simplified Levenberg Marquardt

- Damping to regulate the convergence using backup/restore actions

```
x: the initial guess
while (! converged)
  λ = λ_init
  <H,b> = buildLinearSystem(x);
  E = error(x)
  x_old = x;
  Δx  = solveSparse( (H + λ I) Δx = -b);
  x += Δx;
  If (E < error(x)){
    x  = x_old;
    λ *= 2;
} else { λ /= 2; }
```


Summary

- Graph-Based SLAM for landmarks
- The rank of $\mathbf{H}$ matters
- Levenberg Marquardt for optimization