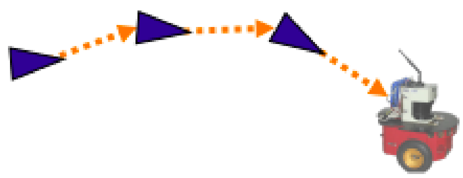
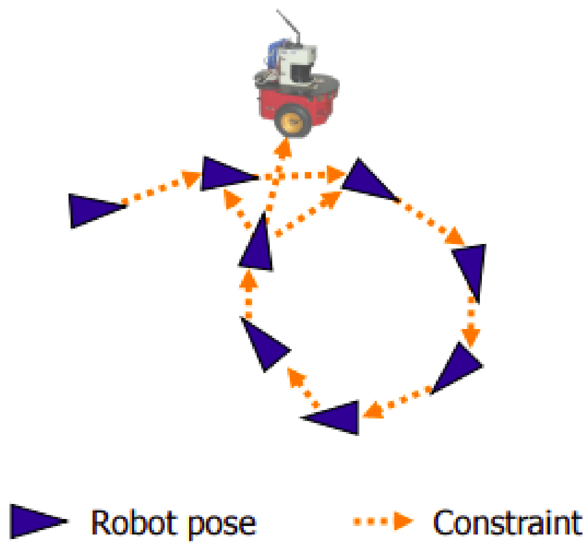
Pose Graph Optimization

Introduction

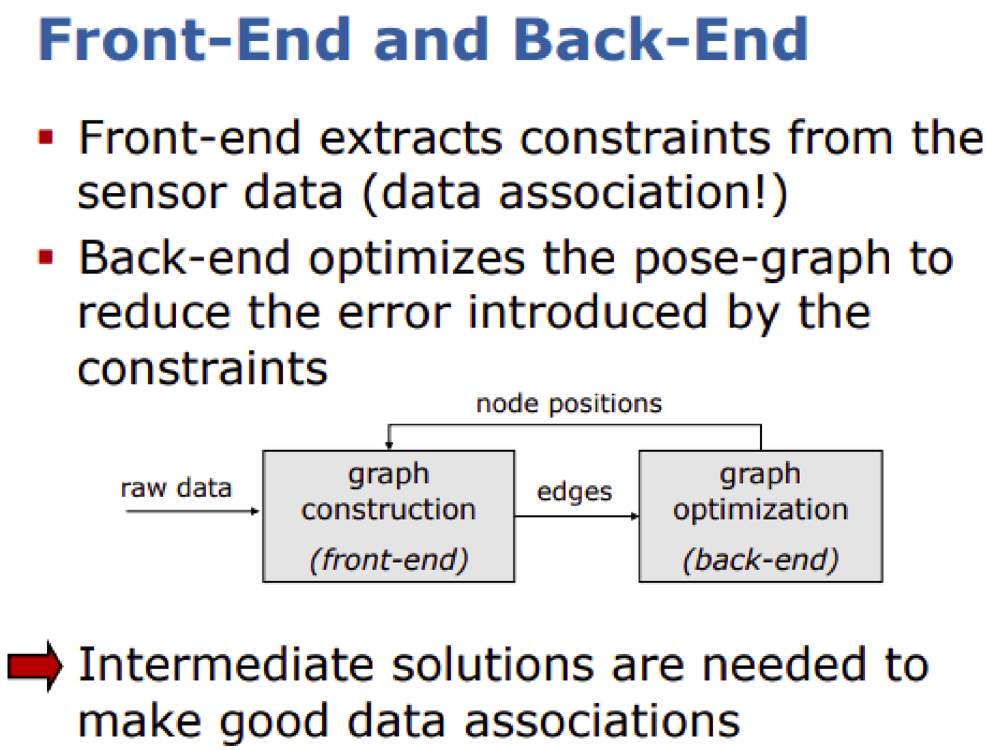
- Constraints connect the poses of the robot while it is moving
- Constraints are inherently uncertain



- Observing previously seen areas generates constraints between non-successive poses



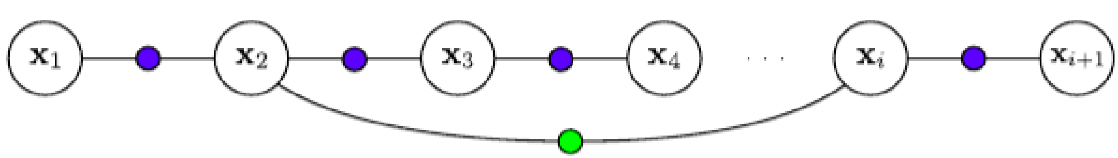
- Use a **graph** to represent the problem
- Every **node** in the graph corresponds to a pose of the robot during mapping
- Every **edge** between two nodes corresponds to a spatial constraint between them
- **Graph-Based SLAM:** Build the graph and find a node configuration that minimize the error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



Problem Formulation

A Pose Graph would have two types of edges

- Odometric Constrains/Edges that connect two successive states $\mathbf{x}_i$ *and* $\mathbf{x}_{i+1}$ via a motion model.
- Loop Closure-based edges These constraints connect two not necessarily successive poses $\mathbf{x}_i$ *and* $\mathbf{x}_j$.



Odometry constraints/edges u_i **between** x_i **and** x_{i+1}***:*** **As denoted by blue 🔵 circle**

$$x_{i+1} \sim \mathcal{N}(f(x_i, u_i), \Sigma_i)$$

Loop closure constraints/edges u_{ij} **between** x_i **and** x_j ***:*** **As denoted by green 🎾 circle**

$$x_j \sim \mathcal{N}(f(x_i, u_{ij}), \Lambda_{ij})$$

The *Optimal Set* of Robot Poses

The MAP estimate of the random variable X, given that we have observed $Y = y$, is given by the value of x that maximizes $P(X \mid Y = y)$ i.e.

> In our case,

$$X^* = \underset{X}{argmax} \; P(X|U)$$

We can factor the joint probability distribution as

$$P(X \mid U) \propto \prod_i \underbrace{P\left(\mathbf{x}_{i+1} \mid \mathbf{x}_i, \mathbf{u}_i\right)}_{\text{Odometry Constraints}} \cdot \underbrace{\prod_{ij} P\left(\mathbf{x}_j \mid \mathbf{x}_i, \mathbf{u}_{ij}\right)}_{\text{Loop Closure Constraints}}$$

💡 This is true for non-gaussian distributions as well, but in robotics, Gaussians are considered for simplification.

$$\underset{X}{argmin} \underbrace{\sum_i \|f(x_i, u_i) - x_{i+1}\|^2_{\sum_i}}_{Odometry\ Constraints} + \underbrace{\sum_{ij} \|f(x_i, u_{ij}) - x_j\|^2_{\Lambda_{ij}}}_{Loop\ Closure\ Constraints}$$

Here, error function is $e_{ij}(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i, \mathbf{u}_{ij}) - \mathbf{x}_j$.

> Therefore, we ended up with a ***weighted* non-linear least-squares** optimization problem.

Using Levenberg-Marquard's algorithm:

$$(\mathbf{J}^\top \mathbf{\Omega} \mathbf{J} \ + \lambda \mathbf{I})\Delta \mathbf{x} = -\mathbf{J}^\top \mathbf{\Omega}^\top \mathbf{f}(\mathbf{x})$$

💡 For 2D/3D SLAM, the error function is

$$\mathbf{e}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathrm{t2v}\left(\mathbf{U}_{ij}^{-1}\left(\mathbf{X}_i^{-1}\mathbf{X}_j\right)\right)$$

where

- X, U are transformation matrices
- $\left(\mathrm{X}_i^{-1}\mathrm{X}_j\right)$ is $\mathrm{X}_j \ w.r.t. \ \mathrm{X}_i$
- U_{ij} is $\mathrm{U}_j \ w.r.t. \ \mathrm{U}_i$.
- ▼ More details about X and t2v: transformation to vector —
    - In 2D case, X would look like $\begin{bmatrix} c\theta & -s\theta & x \\ s\theta & c\theta & y \\ 0 & 0 & 1 \end{bmatrix}$
    - In 2D case,

        > transform matrix (3×3) expressed in homogenous coordinates $\rightarrow$ vector (x, y, θ)

    - $t2v$ will give 0,0,0 when matrix is Identity.

> Final objective to be minimized is:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_k \mathbf{e}_k^T(\mathrm{x})\Sigma_k^{-1}\mathbf{e}_k(\mathrm{x})$$

Optimization In Detail

$$P(X \mid U) \propto \prod_i \underbrace{P\left(\mathbf{x}_{i+1} \mid \mathbf{x}_i, \mathbf{u}_i\right)}_{\text{Odometry Constraints}} \cdot \underbrace{\prod_{ij} P\left(\mathbf{x}_j \mid \mathbf{x}_i, \mathbf{u}_{ij}\right)}_{\text{Loop Closure Constraints}}$$

$$x_{i+1} \sim \mathcal{N}(f(x_i, u_i), \Sigma_i)$$

$$P\left(\mathbf{x}_{i+1}|\mathbf{x}_i, \mathbf{u}_i\right) = \frac{1}{\sqrt{2\pi\left|\mathbf{\Sigma}_i\right|}} \exp\left(-\frac{1}{2}\left(f\left(\mathbf{x}_i, \mathbf{u}_i\right) - \mathbf{x}_{i+1}\right)^\top \mathbf{\Sigma}_i^{-1}\left(f\left(\mathbf{x}_i, \mathbf{u}_i\right) - \mathbf{x}_{i+1}\right)\right)$$

$$P\left(\mathbf{x}_{i+1}|\mathbf{x}_i, \mathbf{u}_i\right) = \eta \exp -\frac{1}{2}\left\|f\left(\mathbf{x}_i, \mathbf{u}_i\right) - \mathbf{x}_{i+1}\right\|^2_{\mathbf{\Sigma}_i}$$

$$\left\|\mathbf{x}_i - \mathbf{y}_i\right\|^2_{\mathbf{\Sigma}_i} = \left(\mathbf{x}_i - \mathbf{y}_i\right)^\top \mathbf{\Sigma}_i^{-1}\left(\mathbf{x}_i - \mathbf{y}_i\right)$$

$$P(X|U) \propto \prod_i \exp -\frac{1}{2}\left\|f\left(\mathbf{x}_i, \mathbf{u}_i\right) - \mathbf{x}_{i+1}\right\|^2_{\Sigma_i} \cdot \prod_{ij} \exp -\frac{1}{2}\left\|f\left(\mathbf{x}_i, \mathbf{u}_{ij}\right) - \mathbf{x}_j\right\|^2_{\mathbf{\Lambda}_{ij}}$$

Taking negative $\log$ on both the sides,

$$-\ \log\ P(X|U) \propto \sum_i \|f(x_i, u_i) - x_{i+1}\|^2_{\sum_i} + \sum_{ij} \|f(x_i, u_{ij}) - x_j\|^2_{\Lambda_{ij}}$$

$$X^* \quad = \quad \underset{X}{argmax}\ P(X|U)$$

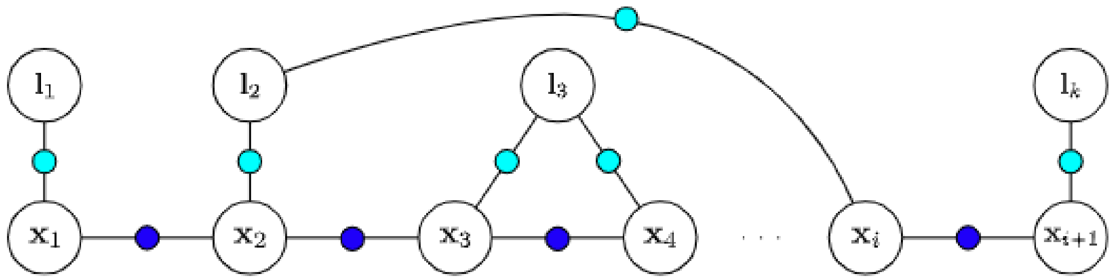
$$= \underset{X}{argmin}\ -log\ P(X|U)$$

$$= \underset{X}{argmin} \underbrace{\sum_i \|f(x_i, u_i) - x_{i+1}\|^2_{\sum_i}}_{Odometry\ Constraints} + \underbrace{\sum_{ij} \|f(x_i, u_{ij}) - x_j\|^2_{\Lambda_{ij}}}_{Loop\ Closure\ Constraints}$$

The Complete SLAM Problem

$$P(X_t, M|U_{t-1}, Z_t) \qquad \begin{array}{l} X_t \rightarrow \text{Poses}; M \rightarrow \text{Map}; \\ U_{t-1} \rightarrow \text{Control Inputs}; Z_t \rightarrow \text{Observations} \end{array}$$

Given the robot's control inputs and observations, we want to estimate the *probability distribution* of the robot's path and the map.



Over here, dark blue dots refer to the control inputs and light blue dots refer to the observations. $\mathbf{l}$ stands for landmarks and $\mathbf{x}$ stands for poses.

Solved Examples

💡 **Let's first go through the 1D solved example !**

- G2o
- Ceres Solver
- RPVIO Demonstration

Acknowledgments

This presentation has been compiled from the contents taken from MR2020 course pages primarily authored by Shubodh Sai. I have also referred to the Mobile Robotics Course by Prof. Cyrill Stachniss at the University of Bonn.