G2O/GTSAM Tutorial

Pose Graph Optimisation and Bundle Adjustment

Pose Graph Optimisation

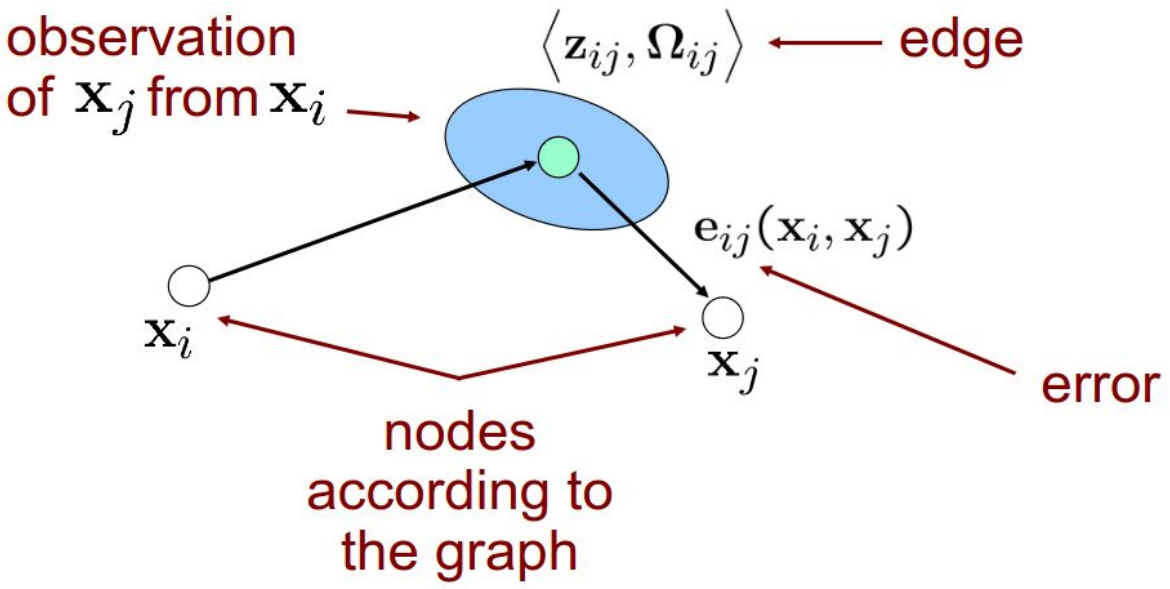
Graph-based optimization is a SLAM optimization problem represented by a graph that is composed of vertexes and edges. In the graph, the poses are represented by vertices and the observation equations are represented by edges. Therefore, the objective function can be written as:

$$x^\star = \min_x F(x)$$

$$F(x) = \sum_{k=1}^{n} e_k(x_k, z_k)^T \Omega_k e_k(x_k, z_k)$$

where Ω is the inverse of the covariance matrix of poses, e is the error function representing difference between real and estimated measurement. By minimizing F(x), we can find a set of poses x that best describes measurements z

Pose Graph Optimisation



- **Goal:** $\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{ij} \mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}$

Pose Graph Optimisation

G2O file format:

- 2D case
  - Robot's Pose: VERTEX_SE2 i x y theta
  - Landmark: VERTEX_XY i x y
  - Pose-Pose: EDGE_SE2 i j x y theta info(x, y, theta)

  Where z_ij = (x, y, theta)^T is the measurement moving from x_i to x_j, i.e. x_j = x_i + z_ij

Pose Graph Optimisation

G2O file format:

- 3D case
    - Robot's Pose: VERTEX_SE3:QUAT i x y z qx qy qz qw
    - Landmark: VERTEX_TRACKXYZ i x y z
    - Pose-Pose: EDGE_SE3:QUAT i j x y z qx qy qz qw info
    - Pose-Point: EDGE_SE3_TRACKXYZ i j PO_id x y z info
    - PO is not a variable of the optimization problem but a fixed offset between the pose of the vertex and the camera observing the points. It allows to directly add measurements without projecting measurement along a kinematic chain e.g. PARAMS_SE3OFFSET 0 0 0 0 0 0 0 1

Pose Graph Optimisation

Code Examples on PGO…

Pose Graph Optimisation

Couple of tasks to observe difference in optimisation:

1. Edit the ground truth trajectory
2. Experiment with 2-3 different values of noise
3. Experiment with 2-3 different values of information matrix
4. Observe the difference that number of loop pairs have on optimisation.

Bundle Adjustment

- Similar to pose-graph optimization but it also optimizes 3D point. So, we'll need to handle landmarks.

- BA is more precise than pose-graph optimization because it adds additional constraints (landmark constraints)

- Typically used for SFM task.Consider a situation in which a set of 3D points is viewed by a set of cameras with matrices . Denote by the coordinates of the j-th point as seen by the i-th camera. We wish to solve the following reconstruction problem: given the set of image coordinates xi_j find the set of camera matrices, Pi, and the points such that Pi . Xj = xi_j .

Bundle Adjustment

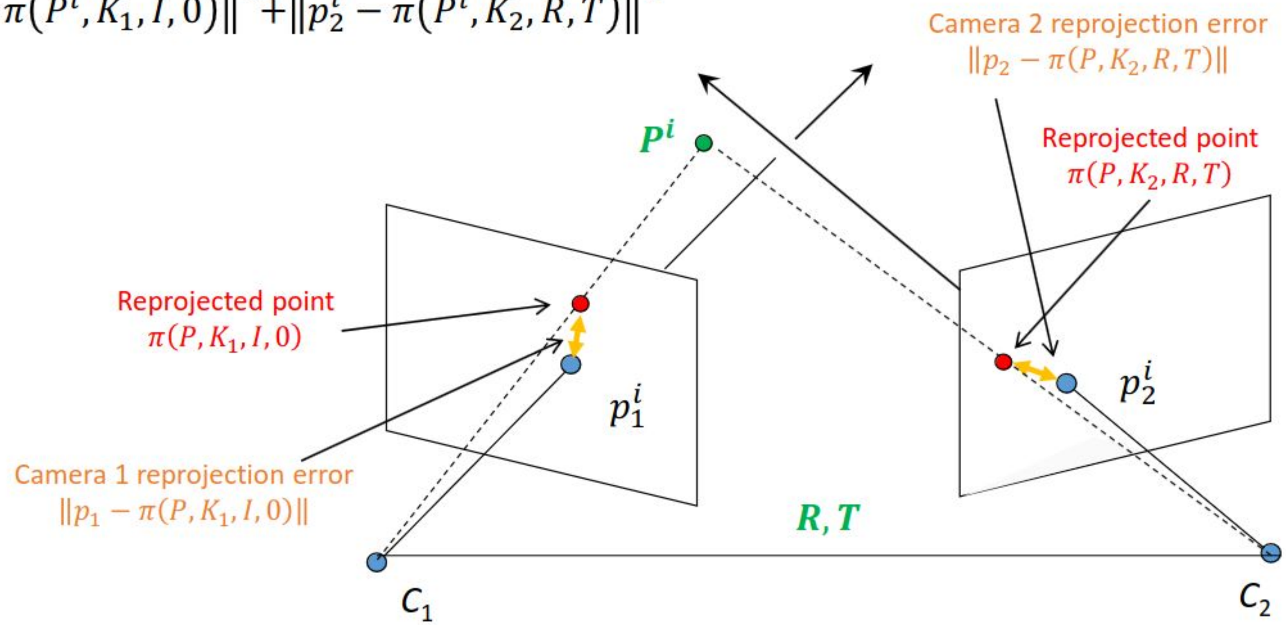
Non-linear, joint optimization of structure, $\boldsymbol{P_i}$, and camera poses $C_1 = [I, 0]$, ... , C_k = $[R_k, T_k]$. Minimizes the Sum of Squared Reprojection Errors across all views:

$$P^i, C_2, \ldots, C_k = argmin_{P^i, C_2, \ldots, C_k} \sum_{k=1}^{n} \sum_{i=1}^{N} \left\| p_k^i - \pi(P^i, K_k, C_k) \right\|^2$$

Bundle Adjustment

For example if K=2:-

$$P^i, R, T = argmin_{P^i,R,T} \sum_{i=1}^{N} \left\|p_1^i - \pi\left(P^i, K_1, I, 0\right)\right\|^2 + \left\|p_2^i - \pi\left(P^i, K_2, R, T\right)\right\|^2$$



Bundle Adjustment

Code Examples on BA…

Bundle Adjustment

Code Examples on BA…

$$
\begin{aligned}
\mathbf{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \mathbf{C} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\
&= \mathbf{C}\,[\mathbf{R}|\ \mathbf{T}] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}
\end{aligned}
$$

which defines the 3×4 projection matrix from Euclidean 3-space to an image:

$$
\mathbf{x} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \qquad \mathbf{P} = \mathbf{C}\,[\mathbf{R}|\ \mathbf{T}]
$$

Resources

- Tutorial code links
- https://rpg.ifi.uzh.ch/docs/teaching/2020/10_multiple_view_geometry_4.pdf
- https://cmsc426.github.io/gtsam/
- https://www.andrew.cmu.edu/user/tianxian/files/Analysis_for_Graph_Based_SLAM_Algorithms_under_g2o_Framework.pdf
- http://mrsl.grasp.upenn.edu/loiannog/tutorial_ICRA2016/VO_Tutorial.pdf