SLAM Using Sonar and Underwater Camera

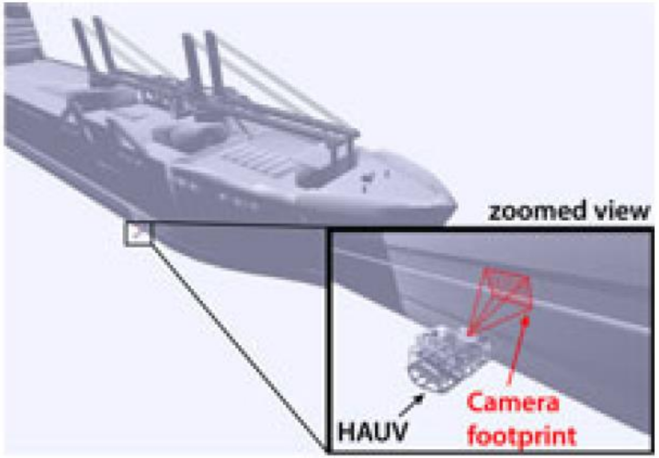
University of Delaware

Outline

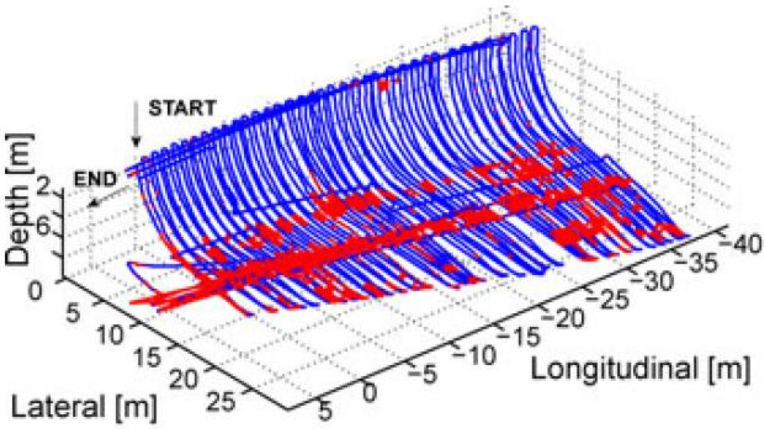
✓System Setup
✓Measurement Process
✓G2o formulation and solver
✓Simulation

Background for Underwater 3D SLAM

- Security inspection for dams, ship hulls, harbors and pipelines;
- Growing scientific requirement of a regular monitoring of the underwater ecosystems. Tracking and modeling the changes in marine environment.

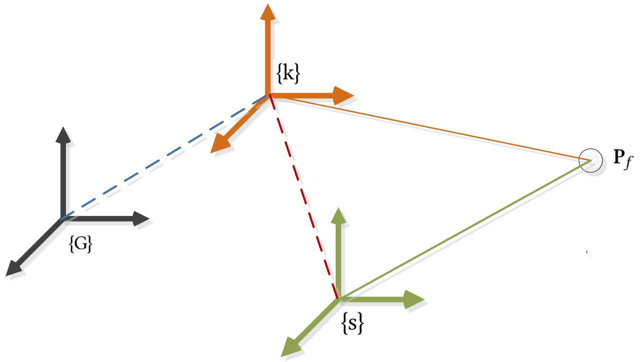


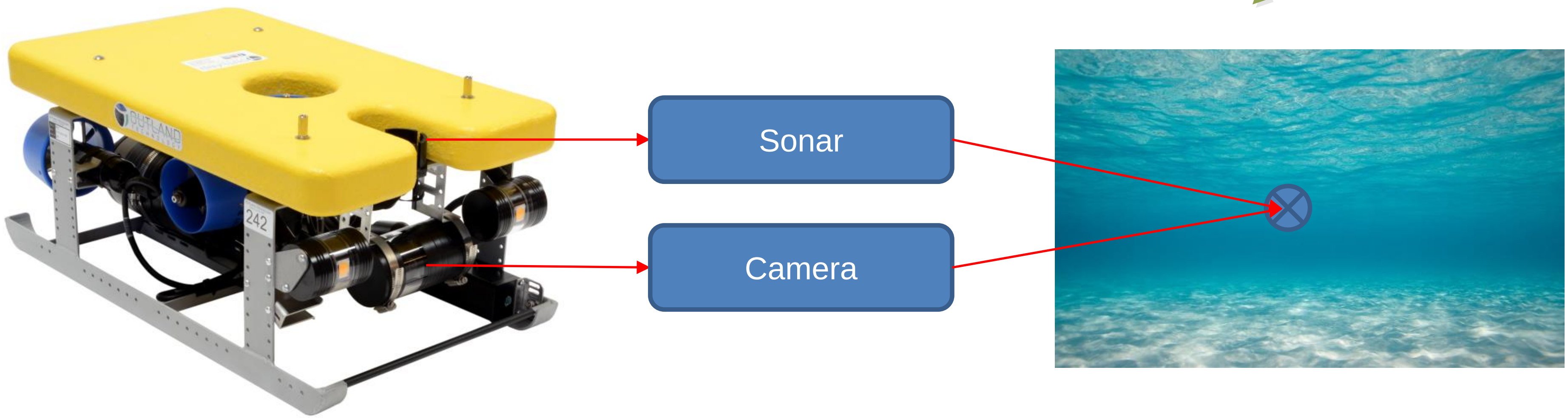
From Ayound Kim, 2013



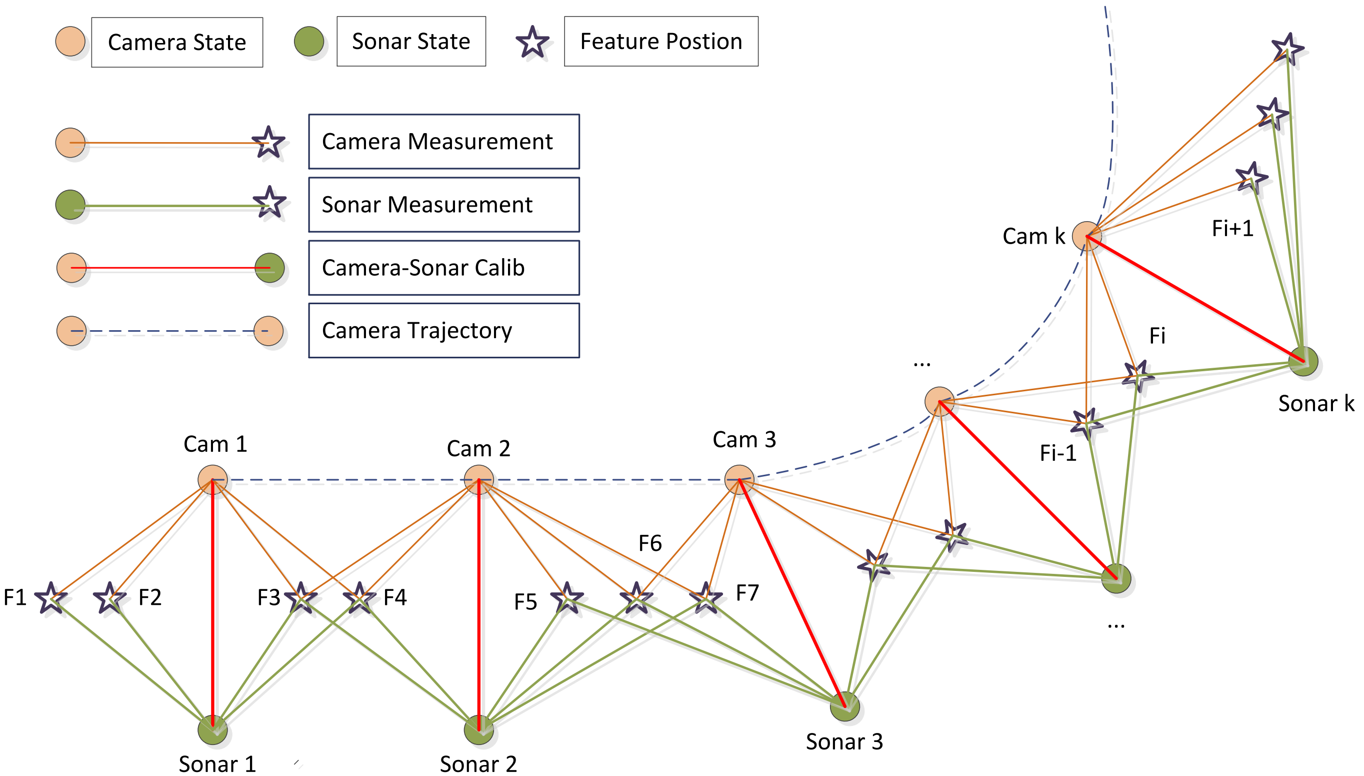
System Setup

- ➢ Recover 3D Feature points and sensor trajectory
- ➢ ROV 1000 Outland, Underwater Camera, Sonar
- ➢ Initialization with Sonar and Camera Measurement
- ➢ G2O (general graph optimization) to get the final value

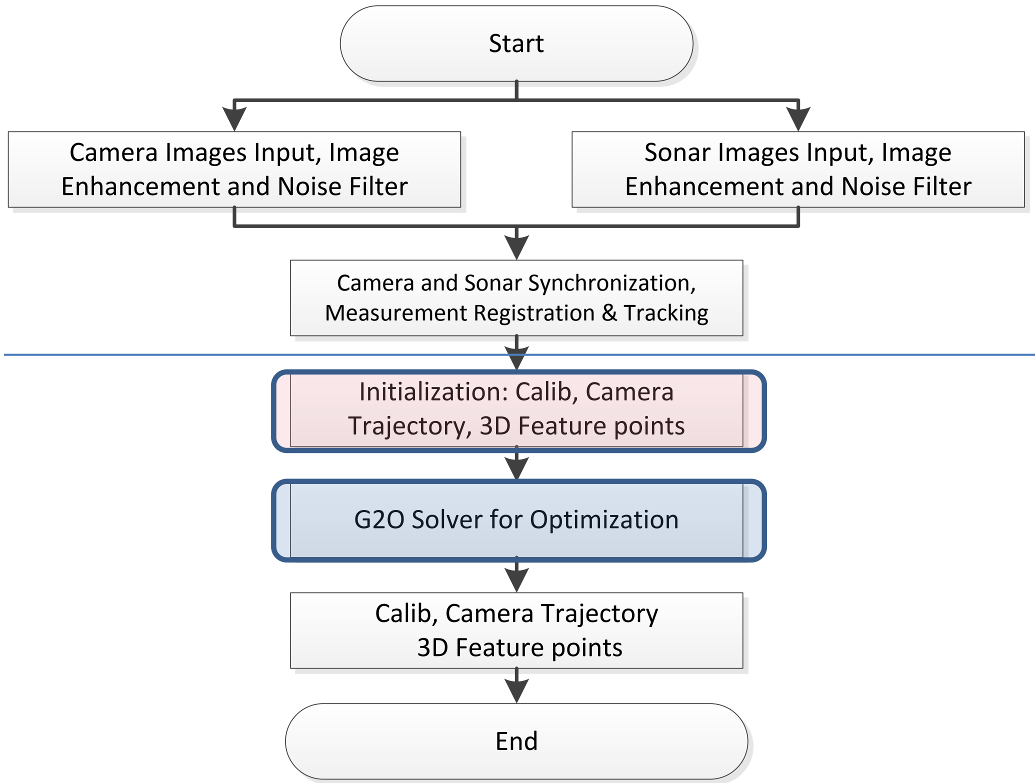




Measuring Process



Solution Process



Data Generation:

- Simulate the Camera and Sonar dataset
- Synchronization, Registration & tracking are easier to realized with simulated dataset

Solution:

- Camera and Sonar meas pair for local 3D feature initialization
- Use G2O solver for optimization to get the final value
- G2O is a popular nonlinear least squares solver

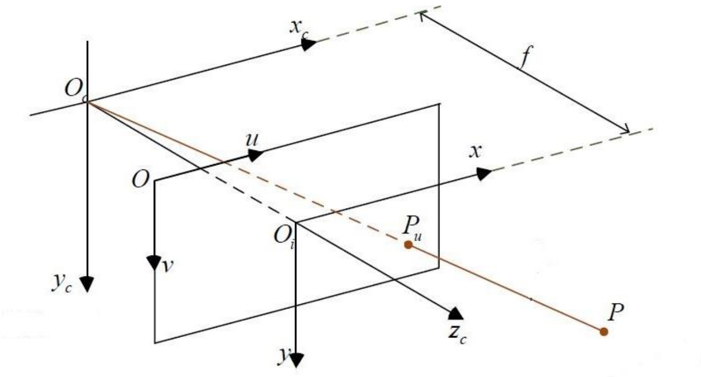
Measurement Model for Camera

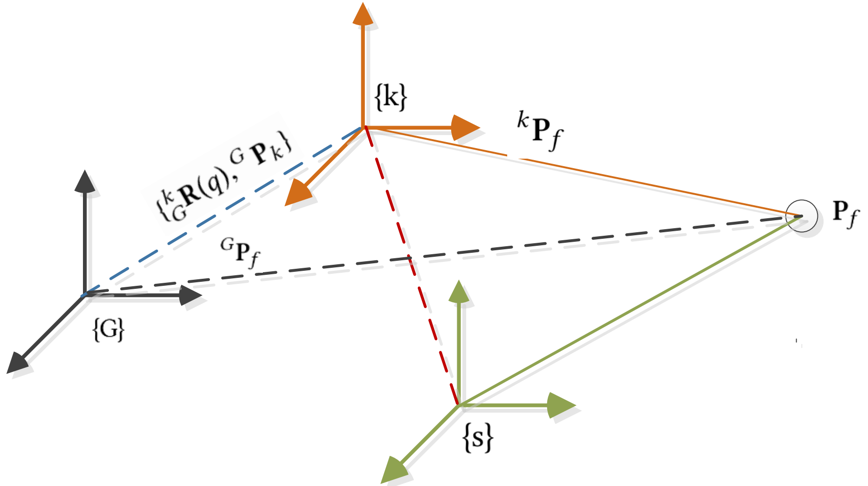
Camera Measuring Model: Camera k measuring a feature f:

$$\mathbf{z}_{ck} = \mathbf{h}_c(\mathbf{x}) + \omega_c$$

$$\mathbf{z}_{ck} = \begin{bmatrix} \frac{{}^k\mathbf{p}_f(1)}{{}^k\mathbf{p}_f(3)} \\ \frac{{}^k\mathbf{p}_f(2)}{{}^k\mathbf{p}_f(3)} \end{bmatrix}$$

$${}^k\mathbf{p}_f = {}^k_G\mathbf{R}(q)\left({}^G\mathbf{p}_f - {}^G\mathbf{p}_k\right)$$



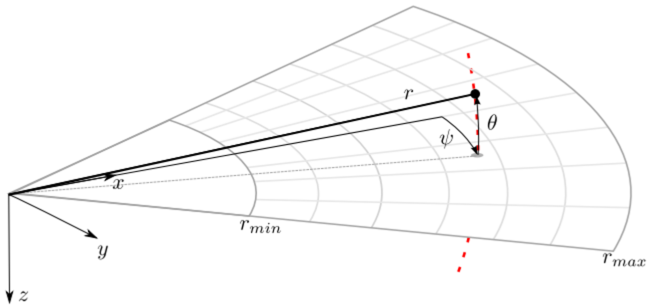


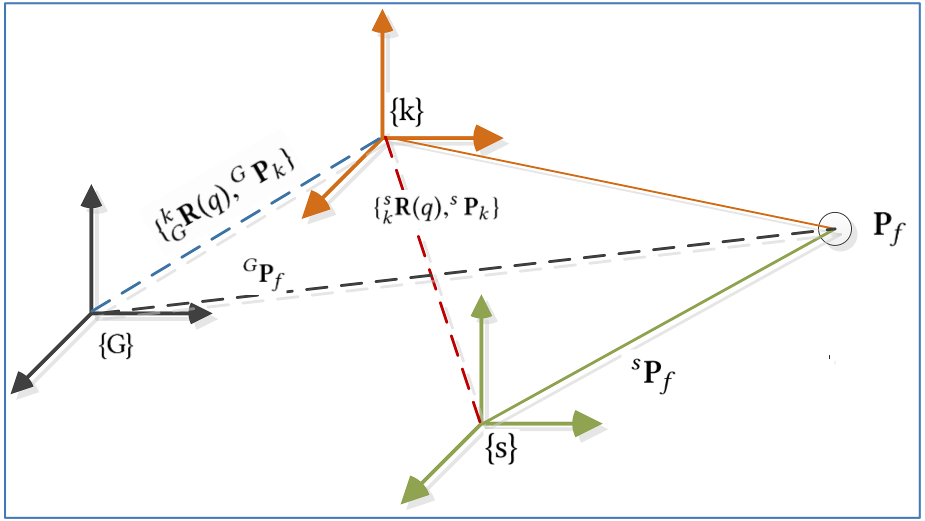
Measurement Model for Sonar

Sonar Measuring Model: Sonar k measuring a feature f: $\mathbf{z}_{sk} = \mathbf{h}_s(\mathbf{x}) + \omega_s$

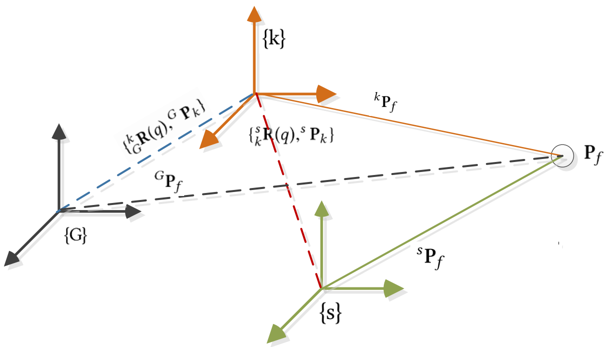
$$\mathbf{z}_{sk} = \begin{bmatrix} \sqrt{({}^s\mathbf{p}_f(1))^2 + ({}^s\mathbf{p}_f(2))^2 + ({}^s\mathbf{p}_f(3))^2} \\ \arctan\left(\frac{{}^s\mathbf{p}_f(2)}{{}^s\mathbf{p}_f(1)}\right) \end{bmatrix}$$

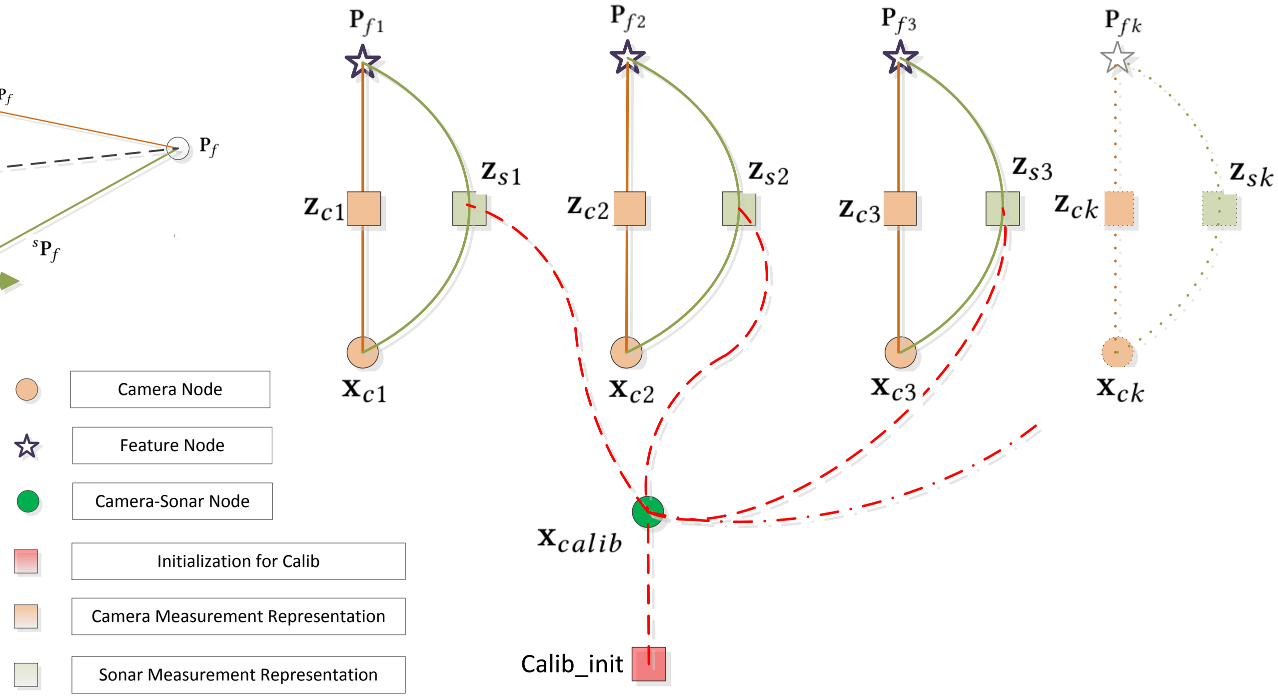
$${}^s\mathbf{p}_f = {}^s_k\mathbf{R}\,{}^k_G\mathbf{R}(q)\left({}^G\mathbf{p}_f - {}^G\mathbf{p}_k\right) + {}^s\mathbf{p}_k$$





Graph Representation





Math formulation

- Target: get the best state vector x. includes:

$$\mathbf{x} = \Big[\underbrace{{}^{S}q_c \quad {}^{S}p_c}_{\text{Xcalib}} \quad \underbrace{q_0 \quad {}^{G}\mathbf{p}_0 \quad \cdots \quad q_n \quad {}^{G}\mathbf{p}_n}_{\text{Xc}} \quad \underbrace{{}^{G}\mathbf{p}_{f0} \quad \cdots \quad {}^{G}\mathbf{p}_{fm}}_{\text{Xf}}\Big]$$

Camera Sonar Calib

Camera State(0 … k)

Feature Position(0 … m)

- Method: Minimize the following least squares with batch up optimization

$$\min_{\mathbf{x}} \sum_i ||\mathbf{z}_{ci} - \mathbf{h}_{ci}(\mathbf{x})||^2_{\mathbf{W}_i} + \sum_j \left\|\mathbf{z}_{sj} - \mathbf{h}_{sj}(\mathbf{x})\right\|^2_{\mathbf{W}_j}$$

Cam Meas residue

Sonar Meas residue

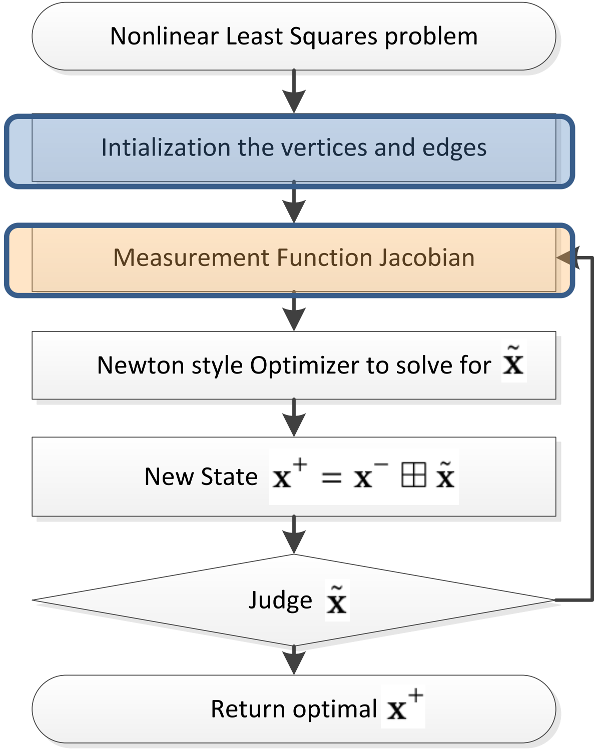
G2O Optimization Process

- Use g2o for solving the nonlinear least squares

$$\min_{\tilde{\mathbf{x}}} \sum_i \|\mathbf{z}_{ci} - \mathbf{h}_{ci}(\hat{\mathbf{x}} \boxplus \tilde{\mathbf{x}})\|^2_{\mathbf{W}_i} + \sum_j \left\|\mathbf{z}_{sj} - \mathbf{h}_{sj}(\hat{\mathbf{x}} \boxplus \tilde{\mathbf{x}})\right\|^2_{\mathbf{W}_j}$$

- Initial guess for the vertices and edges
- Using the newton type optimization method
- First order Taylor expansion for linearization
- Try to find the optimal $\tilde{\mathbf{x}}$ best fit the problem
- Get the optimal estimate by :

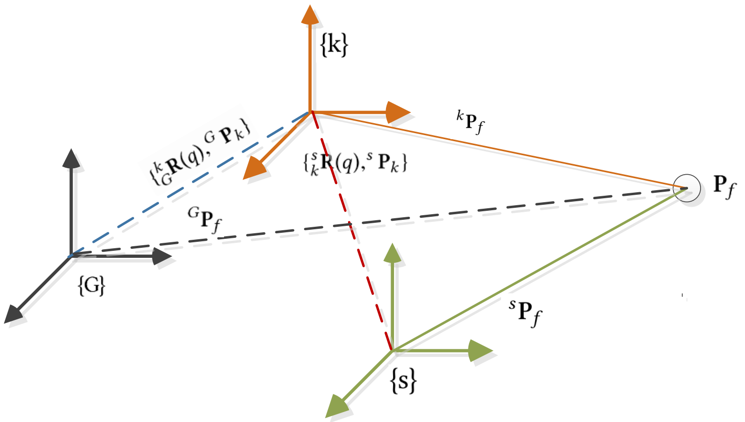
$$\mathbf{x}^{+} = \mathbf{x}^{-} \boxplus \tilde{\mathbf{x}}$$



Camera Sonar Feature Points Initialization

Frame representation:

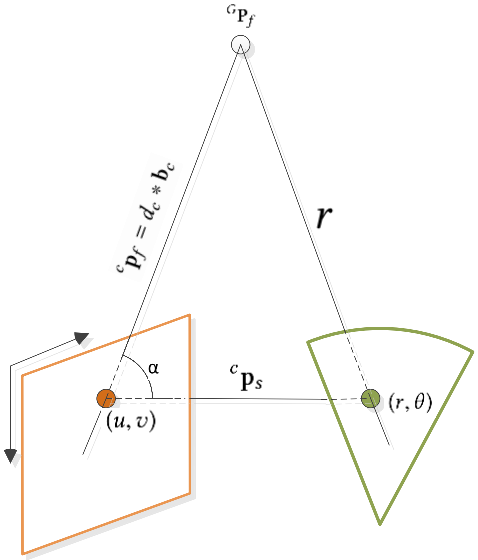
$$ {}^{s}\mathbf{p}_f = {}^{s}_{c}\mathbf{R}\,{}^{c}\mathbf{p}_f + {}^{s}\mathbf{p}_c $$



Triangular Method:

$$ \mathbf{r}^2 = \mathbf{d}_c^2 + \|{}^{c}\mathbf{p}_s\| - 2\mathbf{d}_c\,\|{}^{c}\mathbf{p}_c\| \cos\alpha $$

$$ \cos\alpha = \frac{\mathbf{b}_c \cdot {}^{c}\mathbf{p}_s}{\|{}^{c}\mathbf{p}_s\|} $$

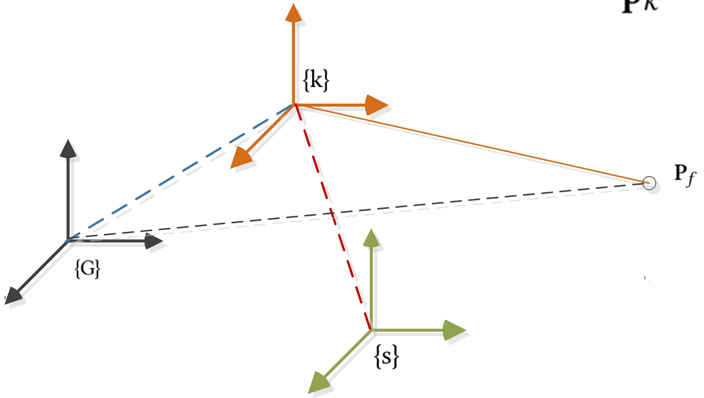


Init for CamTrajectory with 3 points

Frame representation:

$$ {}^{G}\mathbf{p}_{f\,i} = {}^{G}\mathbf{p}_{k} + {}^{G}_{k}\mathbf{R}\,{}^{k}\mathbf{p}_{f\,i} $$

$$ \Delta_{ij} = \mathbf{p}_{f\,i} - \mathbf{p}_{f\,j} $$

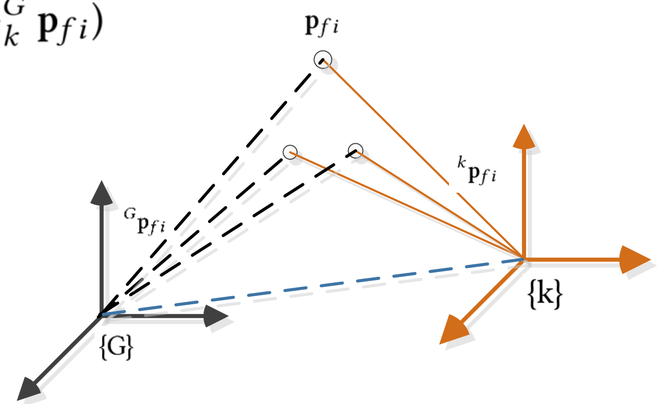


3 corresponding points are needed for the trajectory recovery

$$ \begin{bmatrix} {}^{G}\Delta_{12} & {}^{G}\Delta_{13} & {}^{G}\Delta_{12} \times {}^{G}\Delta_{13} \end{bmatrix} = {}^{G}_{k}\mathbf{R} \begin{bmatrix} {}^{k}\Delta_{12} & {}^{k}\Delta_{13} & {}^{k}\Delta_{12} \times {}^{k}\Delta_{13} \end{bmatrix} $$

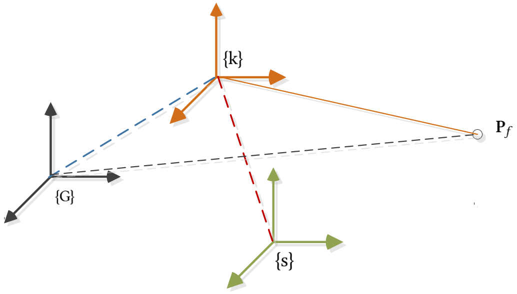
$$ {}^{G}_{k}\mathbf{R} = \begin{bmatrix} {}^{G}\Delta_{12} & {}^{G}\Delta_{13} & {}^{G}\Delta_{12} \times {}^{G}\Delta_{13} \end{bmatrix} \begin{bmatrix} {}^{G}\Delta_{12} & {}^{G}\Delta_{13} & {}^{G}\Delta_{12} \times {}^{G}\Delta_{13} \end{bmatrix}^{-1} $$

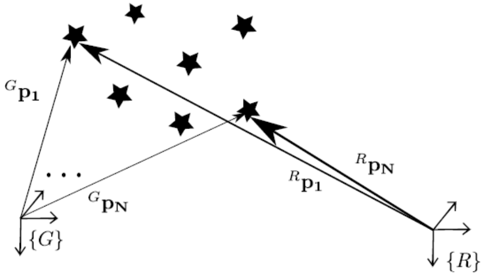
$$ {}^{G}\mathbf{p}_{k} = \frac{1}{3} \sum_{i=1} \left({}^{G}\mathbf{p}_{f\,i} - {}^{G}_{k}\,\mathbf{p}_{f\,i} \right) $$



What if more points are available?

If more than corresponding points are available…





$${}^G\mathbf{p}_i = {}^G_R\mathbf{C}\,{}^R\mathbf{p}_i + {}^G\mathbf{p}_R\,, \quad i = 1 \ldots N$$

Define centroids of points:

$${}^G\mathbf{p}_\Theta = \frac{1}{N}\sum_{i=1}^{N} {}^G\mathbf{p}_i$$

$${}^R\mathbf{p}_\Theta = \frac{1}{N}\sum_{i=1}^{N} {}^R\mathbf{p}_i$$

Separate ${}^G\mathbf{p}_R$ and formulate:

$${}^G\mathbf{p}_i - {}^G\mathbf{p}_\Theta = {}^G_R\mathbf{C}({}^R\mathbf{p}_i - {}^R\mathbf{p}_\Theta),\ i = 1 \ldots N.$$

Define the differences:

$$\mathbf{u}_i := {}^G\mathbf{p}_i - {}^G\mathbf{p}_\Theta$$

$$\mathbf{v}_i := {}^R\mathbf{p}_i - {}^R\mathbf{p}_\Theta$$

Formulate a least squares:

$$\begin{aligned} C &= \sum_{i=1}^{N} (\mathbf{u}_i - {}^G_R\mathbf{C}\mathbf{v}_i)^T(\mathbf{u}_i - {}^G_R\mathbf{C}\mathbf{v}_i) \\ &= \sum_{i=1}^{N} (\mathbf{u}_i^T\mathbf{u}_i - \mathbf{u}_i^T{}^G_R\mathbf{C}\mathbf{v}_i - \mathbf{v}_i^T{}^G_R\mathbf{C}^T\mathbf{u}_i + \mathbf{v}_i^T{}^G_R\mathbf{C}^T{}^G_R\mathbf{C}\mathbf{v}_i) \\ &= \sum_{i=1}^{N} (\mathbf{u}_i^T\mathbf{u}_i - 2\mathbf{u}_i^T{}^G_R\mathbf{C}\mathbf{v}_i - \mathbf{v}_i^T{}^G_R\mathbf{C}^T{}^G_R\mathbf{C}\mathbf{v}_i) \\ &= \sum_{i=1}^{N} (\mathbf{u}_i^T\mathbf{u}_i - 2\mathbf{u}_i^T\boxed{{}^G_R\mathbf{C}}\mathbf{v}_i - \mathbf{v}_i^T\mathbf{v}_i) \end{aligned}$$

${}^G\mathbf{p}_R$

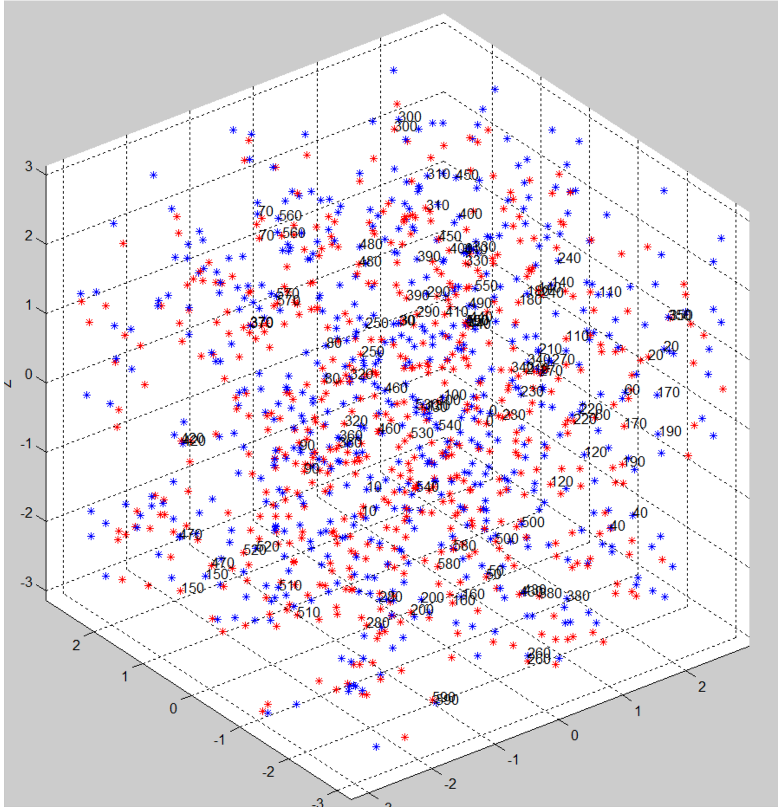
Simulation

- ➢ Simulate the 600 feature points (point cloud)
- ➢ 20 robot poses
- ➢ Add noises:

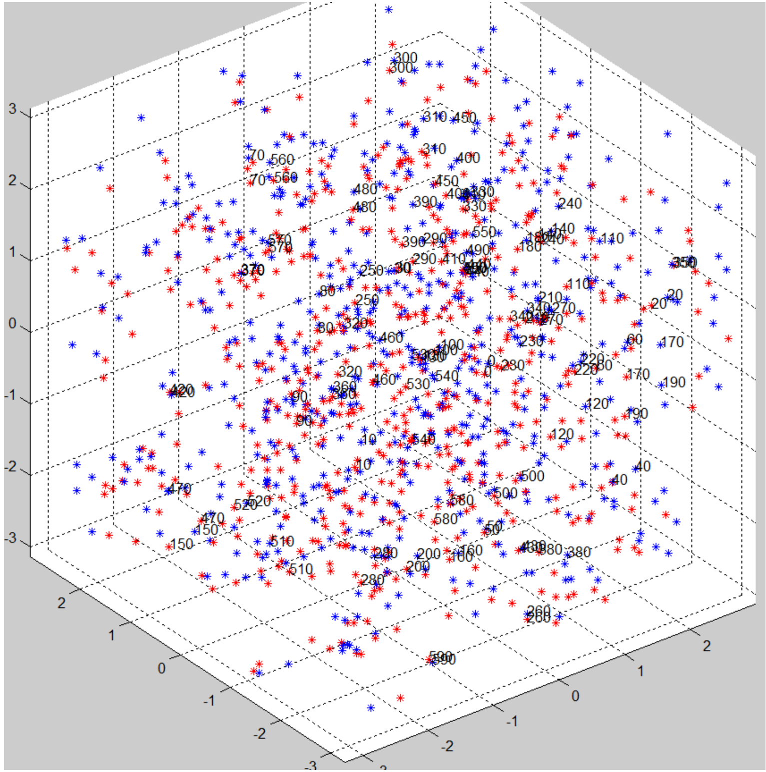
for camera: [-5,5] pixels to camera measurement

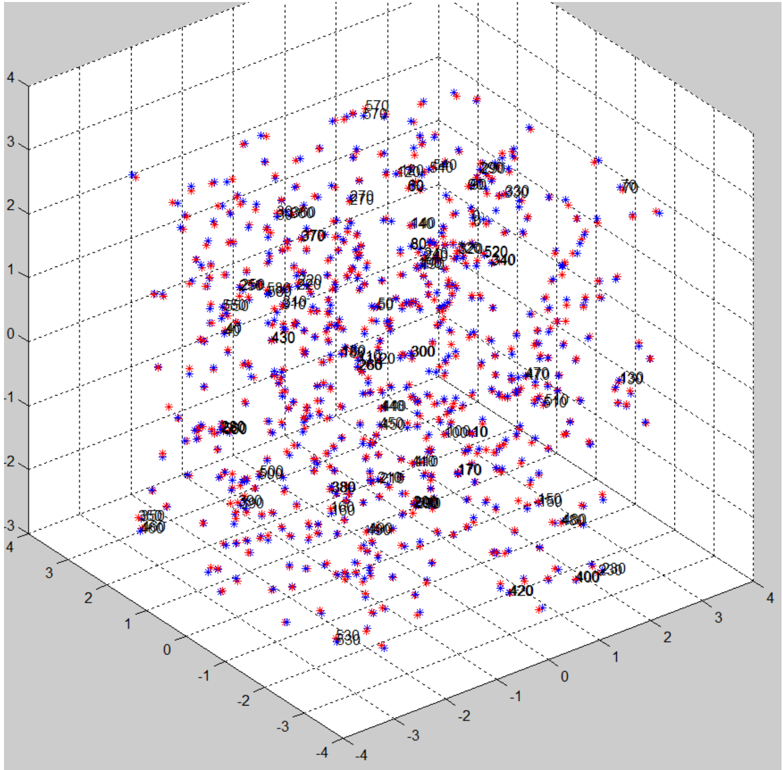
for sonar:[-5,5]degrees to Ө and [-5,5]cm to radius

Raw value with initialization:



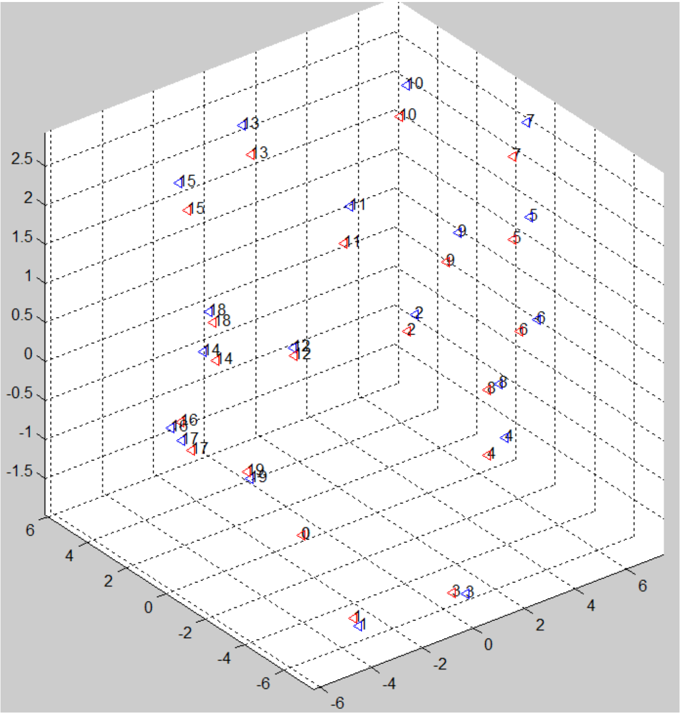
Initialization VS g2o Optimization



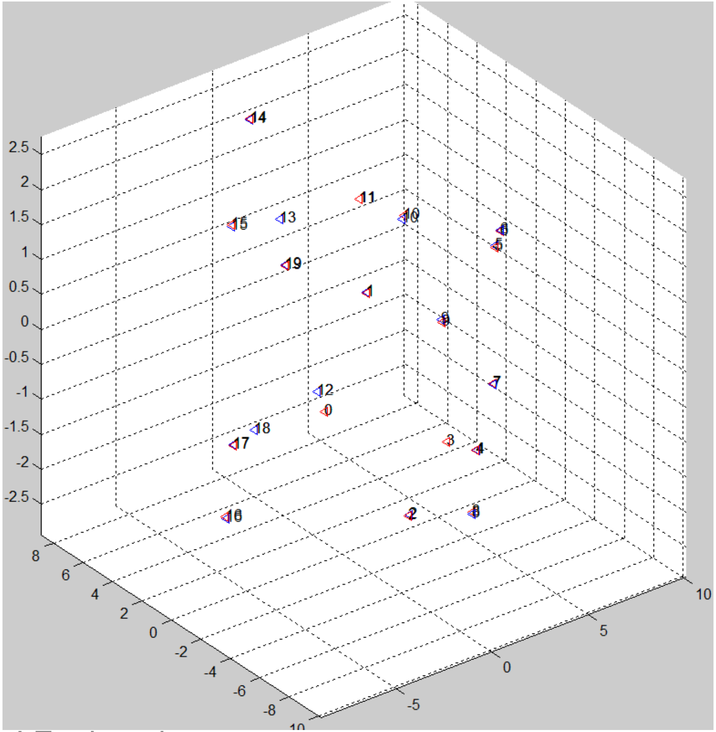


Simulation

- Camera Pose Initialization



- Camera Pose g2o Optimization



Thanks a lot!

Q&A