SLAM With Underwater Camera And Sonar

Kevin Eckenhoff, Yulin Yang
Dept. of Mechanical Engineering
University of Delaware

1 Target and Equipment Setup

This project aims to use the Outland 1000 ROV (shown in Figure 1) in Robotics Discovery Lab (RDL) to do 3D reconstruction and camera motion estimation at the same time in an underwater situation. The sensor setup is listed below:

Figure 1: Outland 1000 ROV

(a) Gemini 720i Multibeam Imaging Sonar (FWR)
(b) UWC-360 Fixed Focus Color Camera (FWR)
(c) UWC-325/p Fixed Focus Color Camera (REAR)

2 Related Literature

Some works have explored different ways to recover 3D geometry with sonar in the underwater situation. Huang et al. [1] inspired by the idea of structure from Motion (SFM), proposes acoustic structure from motion (ASFM). ASFM uses multiple imaging sonar views of the same scene and odometry information to recover the 3D positions of point features. Assalih [2] exploits stereo system and uses two imaging sonar systems placed one on the top of the other, with fixed geometrical connection. Besides, Babaee and Negahdaripour [3] also uses a stereo imaging system, but instead consists of one sonar and one optic camera. The trajectory of the system can be obtained by using optic-acoustic bundle adjustment. However, the stereo system requires that the centers for both cameras' coordinate system align, which limits its application.

ASFM and the work of Assalih [2] are not suitable for our term project due to lack of IMU in Outland 1000. However, the Outland 1000 has one FWR optic-camera and a FWR sonar camera. The sonar images can provide the azimuth angle and range information for the features, which can be used for the optic images which lack the depth information for the features. Therefore, our proposed method is to use both the sonar and optic camera for the measurement. This will yield 3D position estimation as well as the motion of the camera through a batch-up optimization of all the feature measurements. Unlike Babaee and Negahdaripour [3], the proposed method will not have the special requirement of the alignment of both centers of the two cameras, and try to handle the general case of combining one sonar and one optic camera.

3 Measuring Process

The measuring process is shown in Figure 2. Yellow nodes and green nodes represent camera poses and sonar poses at different time steps respectively. The pentagon nodes represent the lankmarks measured by the sonar or camera. The links between these nodes represent different mathematical connections (measurement models). For example, the yellow links between camera nodes and pentagon nodes represent the camera measurement of the landmarks. Similarly, the green links represent the sonar measurement of the landmarks. Because the camera and the sonar are both fixed to the ROV, the red link represents the rigid frame transformation between camera and soanr. The doted links between consecutive camera poses represent the trajectory of camera, which will be also regarded as the trajectory of the ROV.

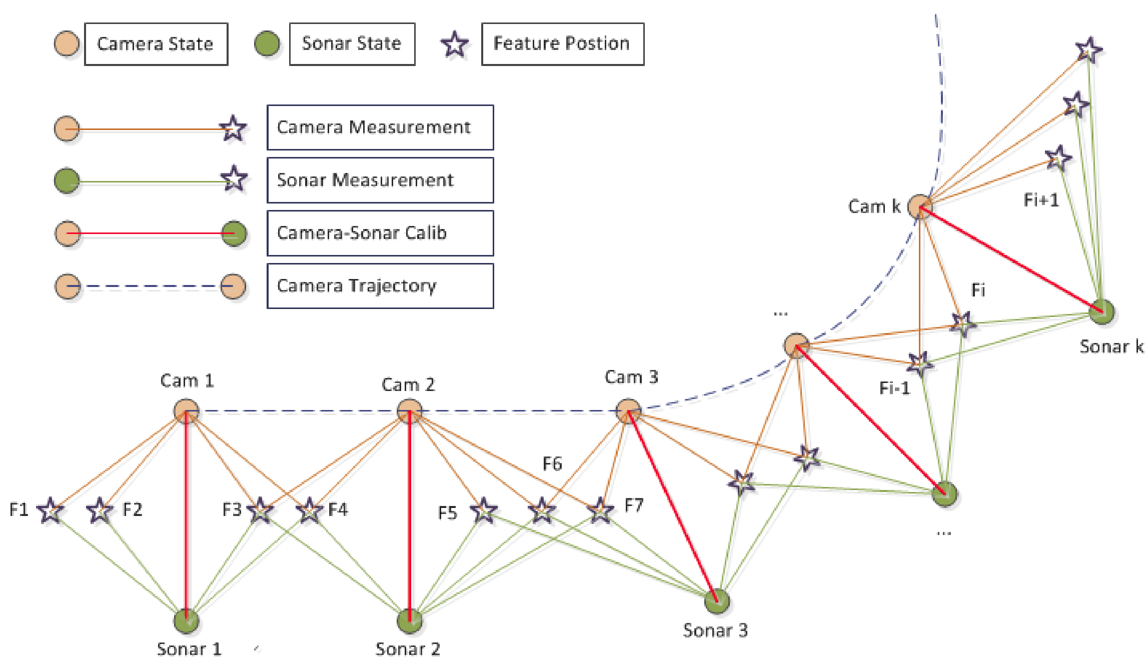

Figure 2: Measuring Process

When the ROV begins to explore the underwater environment, the camera and sonar will measure the landmarks within the environment. The camera will provide the bearing measurements for the landamarks, while the sonar will provide the range and azimuth angle measurements. For each landmark, we can recover its 3D postion with the corresponding camera and sonar measurents. For the camera, we can also recover its poses at different time steps by finding out the landmarks that it has measured. In this way, we can recover the landmarks' postions and the trajectory of the camera. Given the fixed pose transformation between camera and sonar, we can get the trajectory of sonar once the camera trajectory is obtained.

4 Solution Process

The solution process for this problem can be shown in Figure 3. We first get the raw camera images and raw sonar images from two channels of the ROV. Then these raw images will be enhanced and filtered. After that, we will extract features from those images.

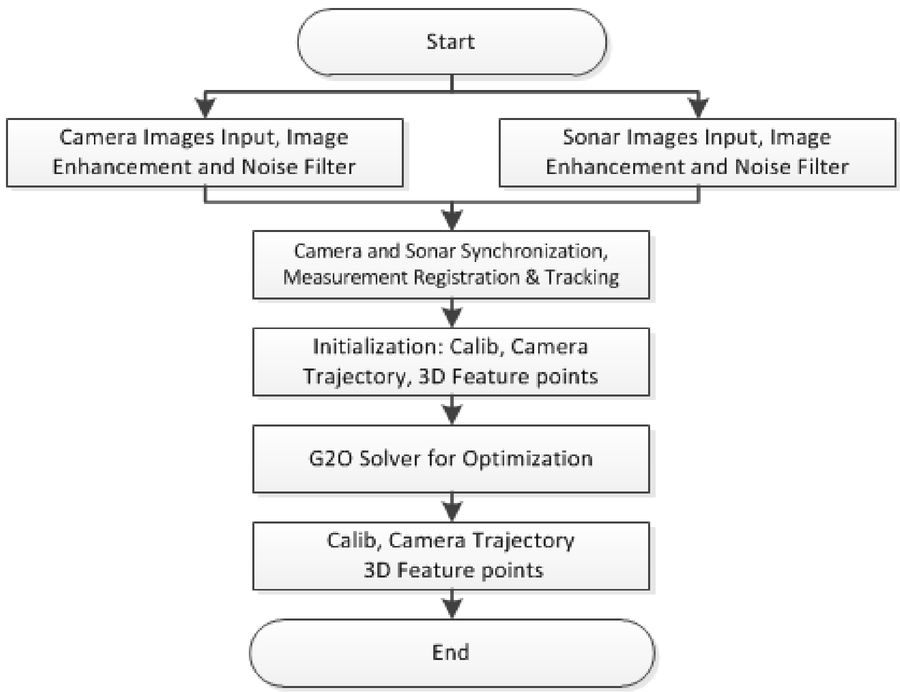

Figure 3: Meausring Process

We need to synchronize the camera and sonar measurements to find the image pairs that are taken at the same time step. This is important becuase we need the image pairs to find out the corresponding features to initialize the landmarks. With camera measurements, we also need to search out the corresponding features in consecutive images, which can be used for the intialization of camera trajectory. During the measuring process, the pose transformation between the camera and sonar is fixed, so we need to calibrate the pose transofromation before we solve the problem. Since it is difficult to get an accurate calibration for this transformation, the calibration information will also be put into the optimization solver for refinement.

After initialization of the camera trajectory, landmark positions and the calibration of camera and sonar, we can feed these information into a batch-up optimization solver: General Graph Optimizatoin (g2o). G2o (Kummerle [4]) is a kind of generalized non-linear least squares problem solver, which is realiable and easy to use. In order to use g2o, we need to transfer the problem into g2o formulation. The following sections will show the detail of derivation.

With the solver, we can get the optimal camera trajectory, 3D landmark positons and the calibration information between camera and sonar.

5 Problem Math Formulation

This section will introduce about the math formation for this SLAM problem. we first introduce the camera and sonar measurement models. Then, the cost function will be formulated based on the measurement model.

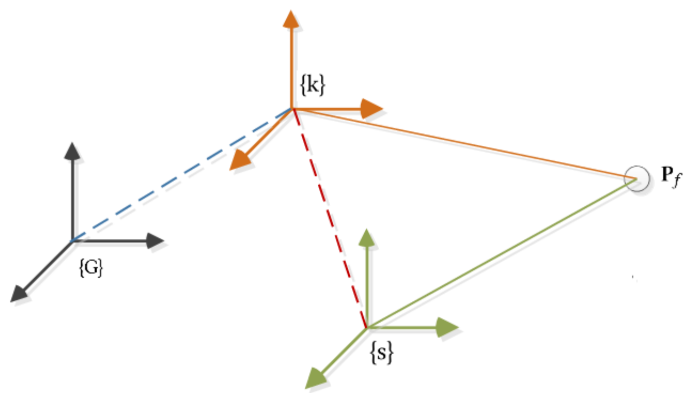

Figure 4: Frame setup for the measuring system

We setup the above 3 frames to better illustrate the problem. {**G**} represents the global frame. All the landmark postions and camera trajectory will be described in this frame. {**k**} represents the camera frame and will be used to describe the measurements from camera. {S} represents the sonar frame, and will be used for the description of the sonar measurements.

5.1 Camera Measurement Model

The camera projects the 3D landmark postions into 2D image Plane. We first consider camera k (camera state at time step k) measuring a feature $\mathbf{p}_f$ (shown in Figure 5).

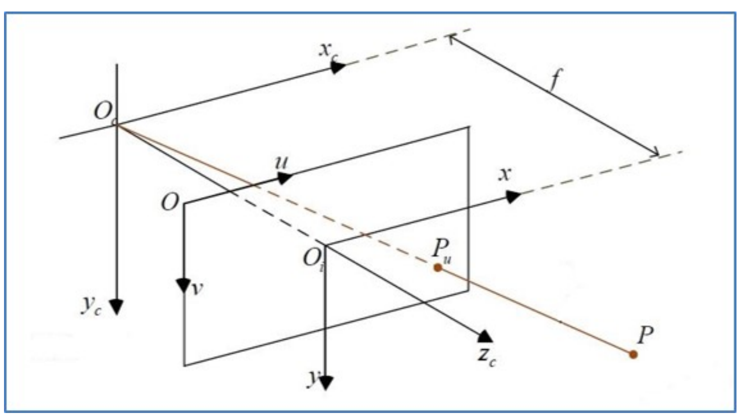

Figure 5: Camera Model

The measurement equation is:

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

where $\mathbf{z}_{ck}$ represents the normalized image pixel measurements from the camera images. ${}^{k}\mathbf{p}_f$ represents the landmark $\mathbf{p}_f$ in the camera frame k. The ${}^{k}\mathbf{p}_f$ can be obtained from the rigid transformation between global frame $\{\mathbf{G}\}$ and camera frame $\{\mathbf{k}\}$ (shown in Figure 6).

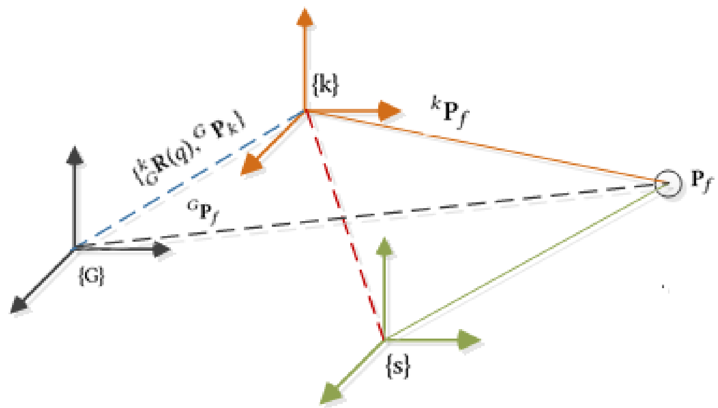

Figure 6: Camera Frame

${}^{k}\mathbf{p}_f$ can be represented as:

$$ {}^{k}\mathbf{p}_f = {}^{k}_{G}\mathbf{R}(q)\left({}^{G}\mathbf{p}_f - {}^{G}\mathbf{p}_k\right) \tag{2}$$

where ${}^{k}_{G}\mathbf{R}(q)$ is the rotation matrix from the global frame to camera frame k regarding quaternion ${}^{k}_{G}q$. Therefore, the overall camera measurement model can be written as:

$$ \mathbf{z}_{ck} = \mathbf{h}_c(\mathbf{x}) + \omega_c \tag{3}$$

where $\mathbf{x}$ represents the state vector, contains the feature position in camera frame ${}^{k}\mathbf{p}_f$, the feature position in global frame ${}^{G}\mathbf{p}_f$, the rotation quaternion from global frame to camera frame ${}^{k}_{G}q$, camera center in the global frame ${}^{G}\mathbf{p}_k$. ω_c represents white noises, with covariance $\mathbf{w}_c$.

5.2 Sonar Measurement Model

The sonar will project the landmark point into the sonar image plane, losing the elevation angle information (shown in Figure 7).

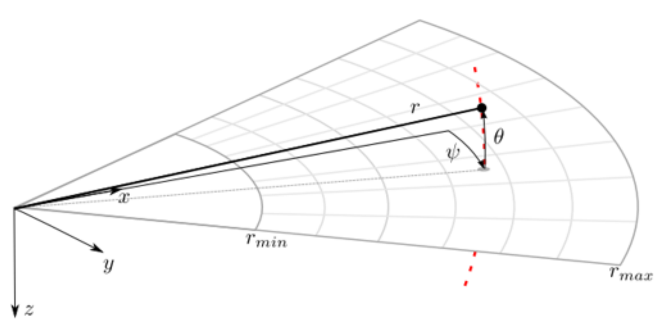

Figure 7: Sonar model

If the sonar s is measuring a landmark $\mathbf{p}_f$, then:

$$\mathbf{z}_{sk} = \begin{bmatrix} \sqrt{({}^s\mathbf{p}_f(1))^2 + ({}^s\mathbf{p}_f(2))^2 + ({}^s\mathbf{p}_f(3))^2} \\ \arctan\left(\frac{{}^s\mathbf{p}_f(2)}{{}^s\mathbf{p}_f(1)}\right) \end{bmatrix} \tag{4}$$

where $\mathbf{z}_{sk}$ represents the range measurement r and the azimuth measurement ψ from the sonar images. ${}^s\mathbf{p}_f$ represents the landmark $\mathbf{p}_f$ represented in the sonar frame s. The ${}^s\mathbf{p}_f$ can be obtained from the rigid transformation between global frame {G} and sonar frame {s} (shown in Figure 8).

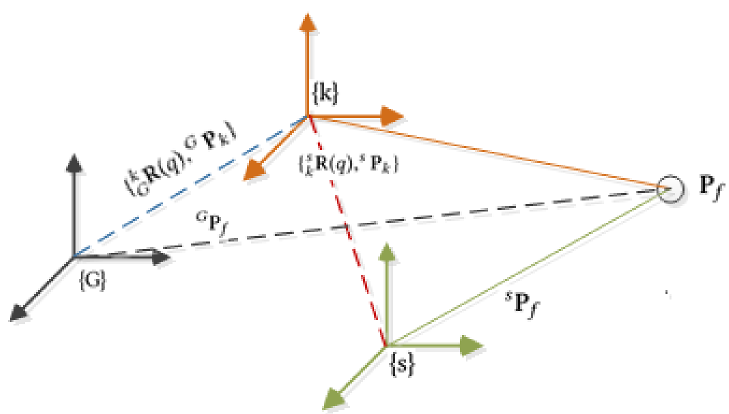

Figure 8: Sonar Frame

${}^s\mathbf{p}_f$ can be represented as:

$${}^s\mathbf{p}_f =^s_k \mathbf{R}^k_G\mathbf{R}(q)\left({}^G\mathbf{p}_f -^G \mathbf{p}_k\right) +^s \mathbf{p}_k \tag{5}$$

Therefore, the overall sonar measurement model can be represented as:

$$\mathbf{z}_{sk} = \mathbf{h}_s(\mathbf{x}) + \omega_s \tag{6}$$

where $\mathbf{x}$ represents the state vector, contains the feature position in sonar frame ${}^s\mathbf{p}_f$, feature position in global frame ${}^G\mathbf{p}_f$, camera center in global frame ${}^G\mathbf{p}_k$, the rotation quaternion from global frame to camera frame ${}^k_Gq$, camera center in sonar frame ${}^s\mathbf{p}_k$, the rotation quaternion from camera frame to sonar frame ${}^k_sq$. ω_s represents white noises, with covariance $\mathbf{w}_s$.

5.3 Cost Function

From the above models, the measurements are corrupted by noises. Thus, we can formulate the batch optimizaton by minimizing the camera measurement residues and the sonar measurement reisdues:

$$\min_{\mathbf{x}} \sum_i ||\mathbf{z}_{ci} - \mathbf{h}_{ci}(\mathbf{x})||^2_{\mathbf{W}_i} + \sum_j \left\|\mathbf{z}_{sj} - \mathbf{h}_{sj}(\mathbf{x})\right\|^2_{\mathbf{W}_j} \tag{7}$$

where $\mathbf{x}$ is the state vector, which contains camera poses at different time instants, the landmarks that are being estimated and the pose transformation bewteen camera and sonar. Thus the state vecor $\mathbf{x}$ can be represented as:

$$\mathbf{x} = \begin{bmatrix} {}^sq_c & {}^s\mathbf{p}_c & q_0 & {}^G\mathbf{p}_0 & \cdots & q_n & {}^G\mathbf{p}_n & {}^G\mathbf{p}_{f0} & \cdots & {}^G\mathbf{p}_{fm} \end{bmatrix}^T \tag{8}$$

6 Problem Solution With g2o

6.1 G2o Representation

From the above analysis, each sonar pose is corresponding to a camera pose and they are related by the pose tranformation calibration information. We can cancel the sonar nodes (shown in Figure 2) and replace them with only one calibration node, which represents the pose tranformation between camera and sonar. Therefore, we can fuse sonar information with camera information. The problem can be formulated as Figure 9.

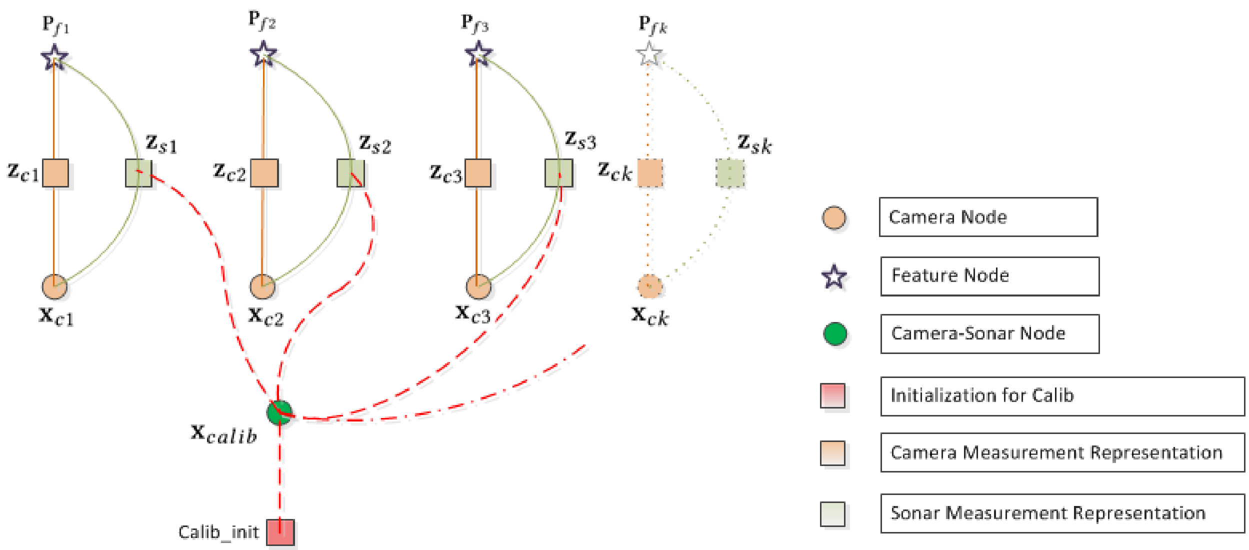

Figure 9: g2o graph representation

In Figure 9, the camera measurents $\mathbf{z}_{c1}$ are still related to yellow (camera) nodes and pentagon (feature) nodes, while the sonar measurements are fused by yellow camera nodes, green calib nodes and the pentagon (feature) nodes. These nodes are corresponding to different components of the state vector $\mathbf{x}$. Therefore, the nonlinear least squares problem can be transferred as minimizing the cost function by getting the best estimate of all the nodes shown in Figure 9.

In the meantime, in order to solve the optimization problem, we perturb our estimates by some error $\tilde{\mathbf{x}}$, then the nonlinear least squares problem can be written as:

$$\min_{\tilde{\mathbf{x}}} \sum_i ||\mathbf{z}_{ci} - \mathbf{h}_{ci}(\hat{\mathbf{x}} \boxplus \tilde{\mathbf{x}})||^2_{\mathbf{W}_i} + \sum_j \left\|\mathbf{z}_{sj} - \mathbf{h}_{sj}(\hat{\mathbf{x}} \boxplus \tilde{\mathbf{x}})\right\|^2_{\mathbf{W}_j} \tag{9}$$

where we have the state vector defined as $\mathbf{x} = \hat{\mathbf{x}} \boxplus \tilde{\mathbf{x}}$:

$$\mathbf{x} \boxplus \tilde{\mathbf{x}} = \begin{bmatrix} \delta q_0 \otimes \hat{q}_0 \\ {}^G\hat{\mathbf{p}}_0 +^G \tilde{\mathbf{p}}_0 \\ \vdots \\ \delta q_n \otimes \hat{q}_n \\ {}^G\hat{\mathbf{p}}_n +^G \tilde{\mathbf{p}}_n \\ {}^G\hat{\mathbf{p}}_{f0} +^G \tilde{\mathbf{p}}_{f0} \\ \vdots \\ {}^G\hat{\mathbf{p}}_{fm} +^G \tilde{\mathbf{p}}_{fm} \end{bmatrix} \tag{10}$$

Here, we use the typical error paramterization of quaternions:

$$\delta q_0 \otimes \hat{q}_0 = \begin{bmatrix} \frac{1}{2}\delta\theta \\ 1 \end{bmatrix} \otimes \hat{q}_0 \tag{11}$$

6.2 Optimizaton Process

With the g2o formulation, we can setup the g2o optimization process (shown in Figure 10).

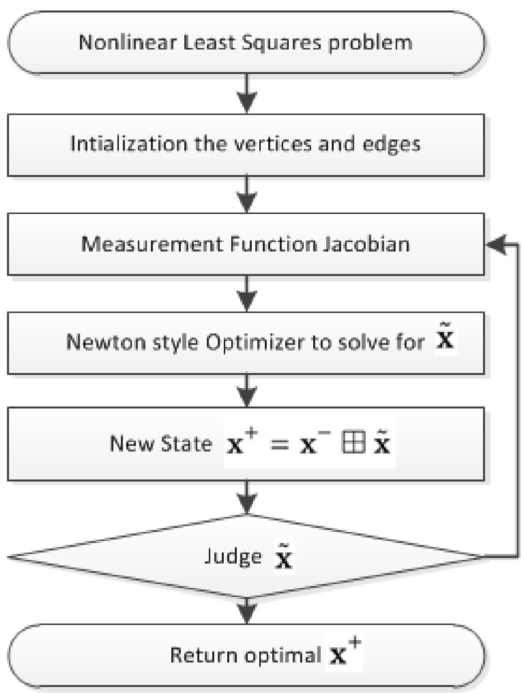

Figure 10: Optimization Process

G2o essentially will solve the nonlinear least squares problems using a Newton type optimizer in a recursive way. We first need to feed in an initial estimate of the state vector $\mathbf{x}^-$, then linearize the nonlinear problem at the initial estimate and get the jacobians for the error state vector $\tilde{\mathbf{x}}$. Then g2o will solve the error state vector $\tilde{\mathbf{x}}$ and we can get the new state vector through:

$$\mathbf{x}^+ = \mathbf{x}^- \boxplus \tilde{\mathbf{x}} \tag{12}$$

This is a recursive process until we get a small enough $\tilde{\mathbf{x}}$ or the iteration times reach a predefined limit. Then the new state will serve as the optimal state vector for the nonlinear least squares problem. The intialization is a key step for the optimization, for a good intilization will improve the accuracy and shorten the time needed for iterations. We need also to linearize the problem by taking the first order derivatives of the measurement equations with regarding the state vector, These derivatives are also known as measurement jacobians.

6.3 Landmark Initializtion

For the landmarks with both camera measurements and sonar measurements, we can use triangulate method to intialize their postions in local camera frame (Figure 11).

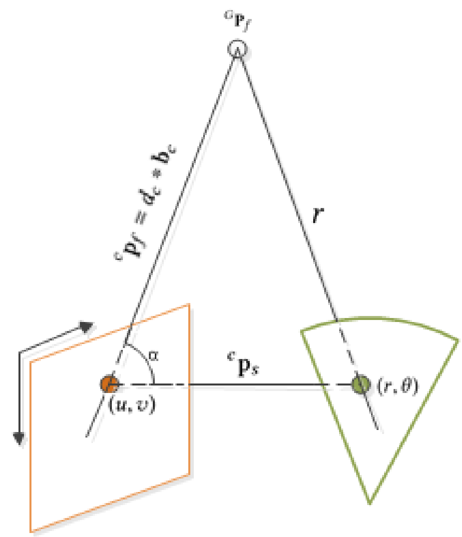

Figure 11: Local 3D features reconstruction

From the camera, we can get the feature pixel measurements (u, v), through which we can recover the bearing b_c. From the sonar, we can get the range and azimuth angle (r, ψ) measurements. Combined the two kind of measurements, we can recover the depth information for the camera measurement and intialize the landmarks. The sonar measurement can be represented as:

$$ {}^{s}\mathbf{p}_f = {}^{s}_{c}\mathrm{R}\, {}^{c}\mathbf{p}_f + {}^{s}\mathbf{p}_c \tag{13} $$

For the triangular, using the cosine theorem, we can get:

$$ \mathbf{r}^2 = \mathbf{d}_c^2 + ||{}^{c}\mathbf{p}_s|| - 2\mathbf{d}_c\, ||{}^{c}\mathbf{p}_c|| \cos\alpha \tag{14} $$

where α represents the angle between the feature bearing b_c and ${}^{c}\mathbf{p}_s$:

$$ \cos\alpha = \frac{\mathbf{b}_c \cdot {}^{c}\mathbf{p}_s}{||{}^{c}\mathbf{p}_s||} \tag{15} $$

By solving the Equation (14) we can get the depth information d_c, then recover the landmarks.

6.4 Camera Trajectoy Initialization

With the corresponding landmark measurements of the camera at two different poses, we can recover the camera trajectory (shown in Figure 12).

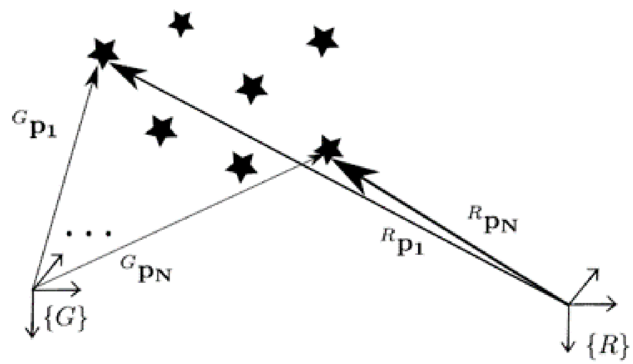

Figure 12: Camera trajectory recovery with n points

The corresponding landmark measurements can be represented as:

$$ {}^{G}\mathbf{p}_i = {}^{G}_{k}\mathbf{R}(q)\,{}^{k}\mathbf{p}_i + {}^{G}\mathbf{p}_k, \quad i = 1 \dots N \tag{16} $$

Then, we can define the centroid of these points as:

$$ {}^{G}\mathbf{p}_{\odot} = \frac{1}{N}\sum_{i=1}^{N} {}^{G}\mathbf{p}_i \tag{17} $$

$$ {}^{k}\mathbf{p}_{\odot} = \frac{1}{N}\sum_{i=1}^{N} {}^{k}\mathbf{p}_i \tag{18} $$

We can seperate the ${}^{G}\mathbf{p}_k$ and formulate:

$$ {}^{G}\mathbf{p}_i - {}^{G}\mathbf{p}_{\odot} = {}^{G}_{k}\mathbf{R}(q)({}^{k}\mathbf{p}_i - {}^{k}\mathbf{p}_{\odot}), \quad i = 1 \dots N \tag{19} $$

We can define the differences as $\mathbf{u}_i, \mathbf{v}_i$:

$$ \mathbf{u}_i := {}^{G}\mathbf{p}_i - {}^{G}\mathbf{p}_{\odot} \tag{20} $$

$$ \mathbf{v}_i := {}^{k}\mathbf{p}_i - {}^{k}\mathbf{p}_{\odot} \tag{21} $$

Then, we can formulate a linear least squares problem:

$$ C = \sum_{i=1}^{N}(\mathbf{u}_i^T\mathbf{u}_i - 2\mathbf{u}_i^T\,{}^{G}_{k}\mathbf{R}(q)\mathbf{v}_i - \mathbf{v}_i^T\mathbf{v}_i) \tag{22} $$

Solve Equation (22) and we can get the ${}^{G}_{k}\mathbf{R}(q)$. Plug ${}^{G}_{k}\mathbf{R}(q)$ back into Equation (16), and we can get ${}^{G}\mathbf{p}_k$.

6.5 Measurement Jacobian

After initialization, we need to linearize the problem and calculate the Jacobians. For the perturbed camera measurement, we can get:

$$ {}^{k}\mathbf{p}_f = \mathbf{R}(\delta q)^{k}_{G}\mathbf{R}(q)\left({}^{G}\hat{\mathbf{p}}_f +^{G}\tilde{\mathbf{p}}_f -^{G}\hat{\mathbf{p}}_k -^{G}\tilde{\mathbf{p}}_k\right) \tag{23} $$

$$ {}^{k}\mathbf{p}_f \approx (\mathbf{I} - \lfloor\delta\theta\times\rfloor)^{k}_{G}\mathbf{R}(\hat{q})\left({}^{G}\hat{\mathbf{p}}_f +^{G}\tilde{\mathbf{p}}_f -^{G}\hat{\mathbf{p}}_k -^{G}\tilde{\mathbf{p}}_k\right) \tag{24} $$

where $\lfloor\delta\theta\times\rfloor$ represents the skew matrix from $\delta\theta$:

$$ \lfloor\delta\theta\times\rfloor = \begin{bmatrix} 0 & -\delta\theta(3) & \delta\theta(2) \\ \delta\theta(3) & 0 & -\delta\theta(1) \\ -\delta\theta(2) & \delta\theta(1) & 0 \end{bmatrix} \tag{25} $$

The Jacobians for this measurement can be found as:

$$ \mathbf{J}_{ck} = \frac{\partial\mathbf{z}_{ck}}{\partial^{k}\mathbf{p}_f}\frac{\partial^{k}\mathbf{p}_f}{\partial\tilde{\mathbf{x}}} \tag{26} $$

Where:

$$ \frac{\partial\mathbf{z}_{ck}}{\partial^{k}\mathbf{p}_f} = \begin{bmatrix} \frac{1}{^{k}\mathbf{p}_f(3)} & 0 & \frac{-^{k}\mathbf{p}_f(1)}{(^{k}\mathbf{p}_f(3))^2} \\ 0 & \frac{1}{^{k}\mathbf{p}_f(3)} & \frac{-^{k}\mathbf{p}_f(2)}{(^{k}\mathbf{p}_f(3))^2} \end{bmatrix} \tag{27} $$

$$ \frac{\partial^{k}\mathbf{p}_f}{\partial\delta\theta} = \lfloor^{k}_{G}\mathbf{R}(\hat{q})\left({}^{G}\hat{\mathbf{p}}_f -^{G}\hat{\mathbf{p}}_k\right)\times\rfloor \tag{28} $$

$$ \frac{\partial^{k}\mathbf{p}_f}{\partial^{G}\tilde{\mathbf{p}}_k} = -^{k}_{G}\mathbf{R}(\hat{q}) \tag{29} $$

$$ \frac{\partial^{k}\mathbf{p}_f}{\partial^{G}\tilde{\mathbf{p}}_k} = {}^{k}_{G}\mathbf{R}(\hat{q}) \tag{30} $$

Next we derive the Jacobians associated with the sonar measurements. These can be written with the perturbed measurement being:

$$ {}^{s}\mathbf{p}_f =^{s}_{k}\mathbf{R}\mathbf{R}(\delta q)^{k}_{G}\mathbf{R}(q)\left({}^{G}\hat{\mathbf{p}}_f +^{G}\tilde{\mathbf{p}}_f -^{G}\hat{\mathbf{p}}_k -^{G}\tilde{\mathbf{p}}_k)\right) +^{s}\mathbf{p}_k \tag{31} $$

$$ {}^{s}\mathbf{p}_f \approx^{s}_{k}\mathbf{R}\,(\mathbf{I} - \lfloor\delta\theta\times\rfloor)^{k}_{G}\mathbf{R}(q)\left({}^{G}\hat{\mathbf{p}}_f +^{G}\tilde{\mathbf{p}}_f -^{G}\hat{\mathbf{p}}_k -^{G}\tilde{\mathbf{p}}_k)\right) +^{s}\mathbf{p}_k \tag{32} $$

The Jacobians of the sonar measurements can be found as:

$$ \mathbf{J}_{sk} = \frac{\partial\mathbf{z}_{sk}}{\partial^{s}\mathbf{p}_f}\frac{\partial^{s}\mathbf{p}_f}{\partial\tilde{\mathbf{x}}} \tag{33} $$

$$ \frac{\partial\mathbf{z}_{sk}}{\partial^{s}\mathbf{p}_f} = \begin{bmatrix} \frac{^{s}\mathbf{p}_f(1)}{||^{s}\mathbf{p}_f||_2} & \frac{^{s}\mathbf{p}_f(2)}{||^{s}\mathbf{p}_f||_2} & \frac{^{s}\mathbf{p}_f(3)}{||^{s}\mathbf{p}_f||_2} \\ \frac{\frac{-^{s}\mathbf{p}_f(2)}{(^{s}\mathbf{p}_f(1))^2}}{1+\left(\frac{^{s}\mathbf{p}_f(2)}{^{s}\mathbf{p}_f(1)}\right)^2} & \frac{\frac{1}{^{s}\mathbf{p}_f(1)}}{1+\left(\frac{^{s}\mathbf{p}_f(2)}{^{s}\mathbf{p}_f(1)}\right)^2} & 0 \end{bmatrix} \tag{34} $$

$$\frac{\partial^s \mathbf{p}_f}{\partial \delta\theta} =^s_k \mathbf{R} \lfloor {}^k_G\mathbf{R}(\hat{q}) \left({}^G\hat{\mathbf{p}}_f -^G \hat{\mathbf{p}}_k \right) \times \rfloor \tag{35}$$

$$\frac{\partial^s \mathbf{p}_f}{\partial^G \tilde{\mathbf{p}}_k} = -^s_k\mathbf{R}^k_G\mathbf{R}(\hat{q}) \tag{36}$$

$$\frac{\partial^s \mathbf{p}_f}{\partial^G \tilde{\mathbf{p}}_f} = {}^s_k\mathbf{R}^k_G\mathbf{R}(\hat{q}) \tag{37}$$

In addition, we need to estimate the the calibration parameters ${}^s_c q$, and ${}^s\mathbf{p}_c$, and simply optimize them in our state vector. The portions of the Jacobians associated with these can be found, as shown before, by perturbing these variables.

$$\frac{\partial^s \mathbf{p}_f}{\partial^s \tilde{\mathbf{p}}_c} = \mathbf{I} \tag{38}$$

$$\frac{\partial^s \mathbf{p}_f}{\partial \delta\theta_{calib}} = \lfloor {}^s_k\mathbf{R}^k_G\mathbf{R}(\hat{q}) \left({}^G\hat{\mathbf{p}}_f -^G \hat{\mathbf{p}}_k \right) \times \rfloor \tag{39}$$

7 Simulation

We use a simulation experiment to verify the proposed solution. We design a simulator which generates 600 random landmark points and 20 robot poses. In order to test the robustness of the proposed method, we add random noises to these simulated data. For camera, we add $[-5, 5]$ pixels errors to camera measurements; for sonar, we add $[-5, 5]$ degrees errors to θ and $[-5, 5]$cm errors to the range measurements.

Figure 13 shows the generated raw landmark points (blue) and the initialized landmark points (red). It is obvious that the intialized points have a relatively large deviations with the raw points. Figure 14 shows another set of generated raw landmark points (blue) and the estimated points (red) after the g2o optimization. It is clear that the optimized points are quite close to raw points. This proves that the proposed soltion can have a robust estimation for simulated landmark data.

Figure 15 and Figure 16 shows the simulated results related to the camera trajectory estimation. Figure 15 shows that the raw trajectory (blue) and initialized trajectory (red) have a larger deviations than the optimal trajectory using g2o optimiztion shown in Figure 16.

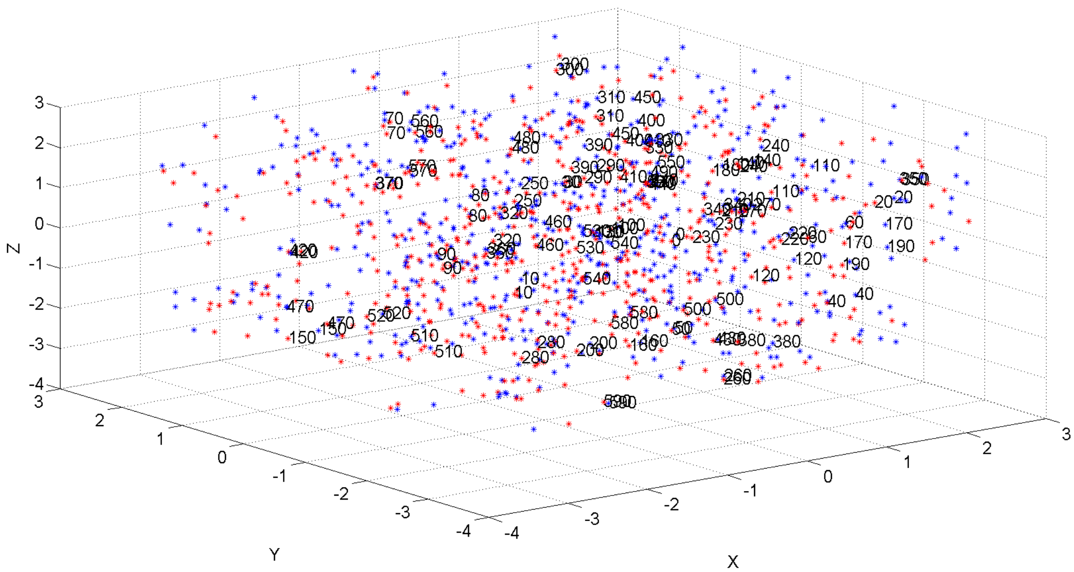

Figure 13: Feature points initializatoin

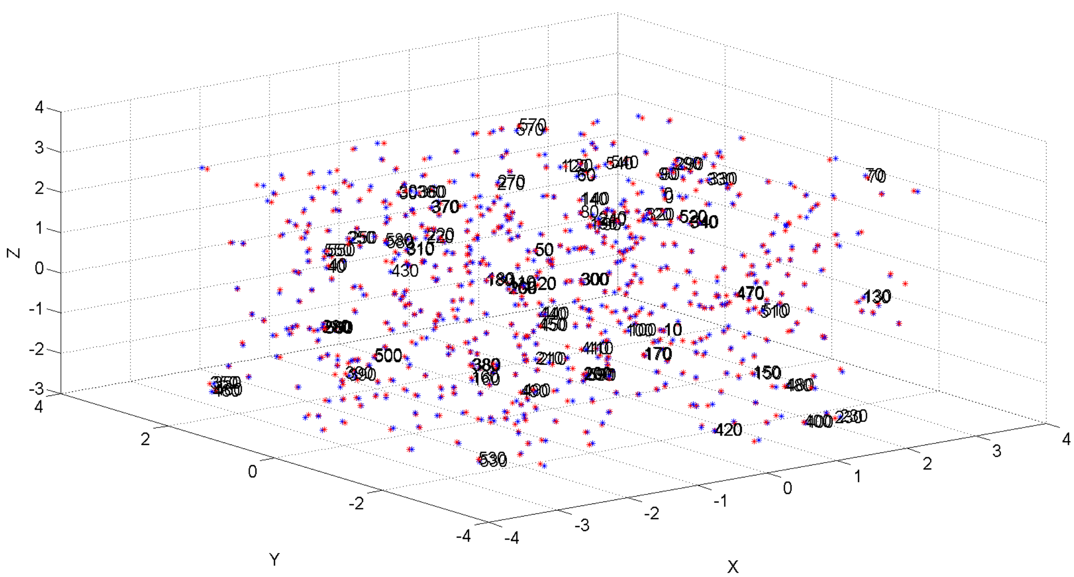

Figure 14: Feature points recovery after g2o

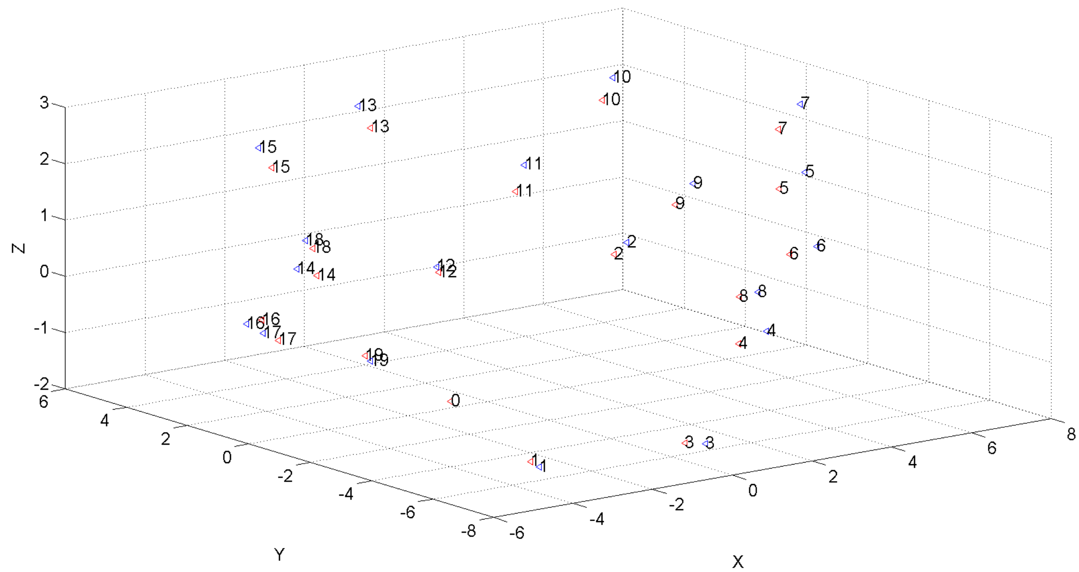

Figure 15: Camera trajectory initialization

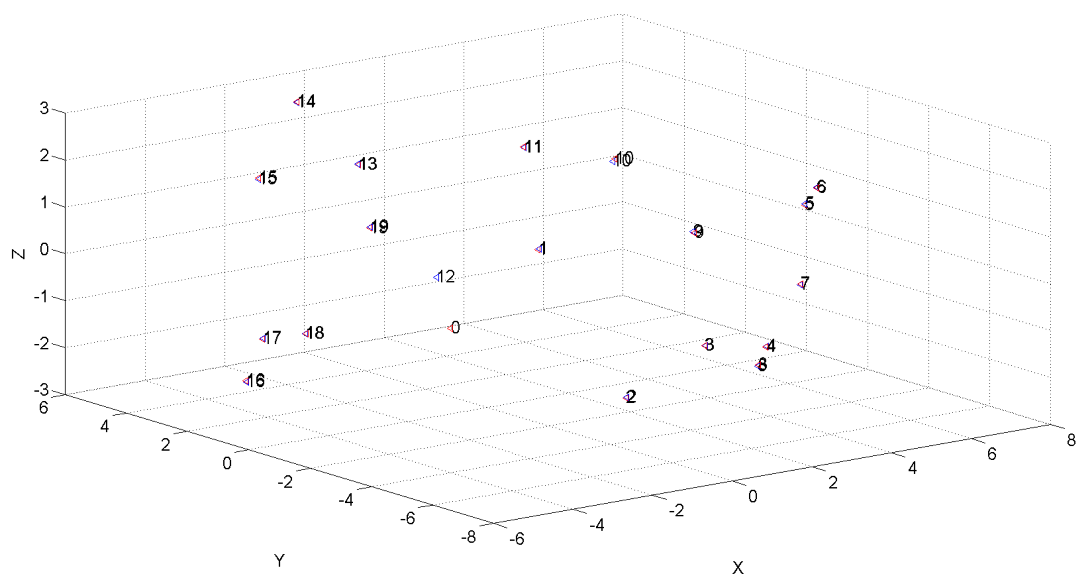

Figure 16: Camera trajectory recovery after g2o

8 Conclusion and Future Work

We propse to fuse the measurements of sonar and underwater camera and use g2o solver to solve the underwater SLAM problem. We show how to initialize the state vector with the measurements and drive the measurement jacobians to linearize the problems. Finally we use a simulation to prove that the proposed method is robust and can fully solve the SLAM problem.

In the meantime, there are still some work needing to be continued. 1) Real measurements directly from the sensor are too noisy and it is a challenge to correspond feature points from camera and sonar measurements. We need to collect real sonar and camera datasets and find a solution. 2) We also need to implement a visulization component in the code to show the real SLAM results, such as the landmark points and the camera trajectory.

References

[1] T. Huang and M. Kaess, "Towards acoustic structure from motion for imaging sonar," in *Intelligent Robots and Systems (IROS)*, pp. 758–765, Sept 2015.

[2] H. Assalih, *3D reconstruction and motion estimation using forward looking sonar.* Thesis, Heriot-Watt University, Oct. 2013.

[3] M. Babaee and S. Negahdaripour, "3-d object modeling from occluding contours in opti-acoustic stereo images," in *Oceans - San Diego, 2013*, pp. 1–8, Sept 2013.

[4] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE International Conference on Robotics and Automation*, (Shanghai, China), pp. 3607–3613, May 9–13, 2011.