LS Optimization in Vision and Robotics: A Generic Formulation

Comments	Recommended to read before reading ICP/pose- graph SLAM
Dates Taught	
Lecture No.	
Links of Videos	Supplementary: Recommended to read before reading ICP/pose- graph SLAM
Module	Basics
Related to All Questions (Property)	

> The objective of this page is to understand the optimization functions in Vision and Robotics as a generic optimization function from the perspective of state and measurements. And differences between these optimization functions, for example, we discuss what is the difference between "Graph SLAM with landmarks" and "Bundle Adjustment"?

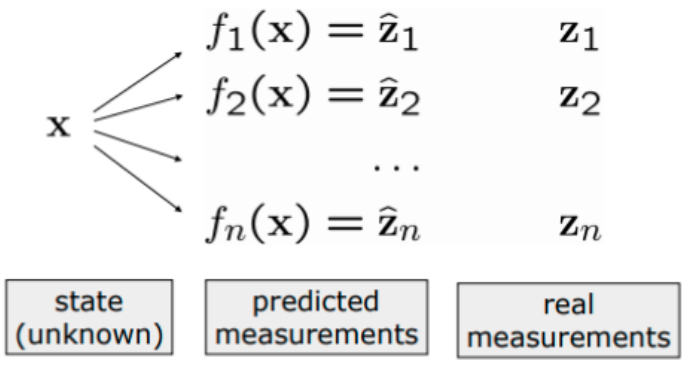
Introduction

In any generic optimization problem, be it in Robotics or Computer Vision, we have sensor(s) which give us some information about the environment or robot or both in the form of measurements (for example, how far is a landmark from our robot's current position; or how much the robot has moved compared to its previous pose). Now, our goal is to find a set of states $\mathbf{x}$ (for example, poses) which best explain these measurements $\mathbf{z}_i$. Also, the system is described by a set of n observation functions $\{f_i(\mathbf{x})\}_{i=1:n}$ which maps our state $\mathbf{x}$ to a predicted measurement $\hat{\mathbf{z}}_i$. Since our goal is to find state $\mathbf{x}$ which explains measurements $\mathbf{z}_i$, we want our "predicted" measurements $\hat{\mathbf{z}}_i$ to be as close as possible to "real" measurements $\mathbf{z}_i$.

$\mathbf{x}$: the state vector.

$\mathbf{z}_i$: a "real" measurement of the state $\mathbf{x}$.

$\hat{\mathbf{z}}_i = f_i(\mathbf{x})$: observation function which maps $\mathbf{x}$ to a "predicted" measurement $\hat{\mathbf{z}}_i$.



- Say we have n "noisy" measurements $\mathbf{z}_{1:n}$ about the state $\mathbf{x}$.

> **Objective:** Estimate the state $\mathbf{x}$ which best explains the measurements $\mathbf{z}_{1:n}$.

Error function

Following the terminology of our least squares optimization page, "residual vector" is the difference between actual and predicted measurement:

$$\mathbf{e}_i(\mathbf{x}) = \mathbf{z}_i - f_i(\mathbf{x}) = \mathbf{z}_i - \hat{\mathbf{z}}_i$$

Assume it is Gaussian error having zero mean with information matrix Ω_i.

Our final squared error thus is a scalar:

$$e_i(\mathbf{x}) = \mathbf{e}_i(\mathbf{x})^T \mathbf{\Omega}_i \mathbf{e}_i(\mathbf{x})$$

This is a Non-Linear Weighted Least Squares problem as we have seen in our least squares optimization page. We have seen how we can solve such optimization problems in the same page. In this page, we discuss optimization functions in Vision and Robotics.

1. Graph SLAM

Optimizing variables: poses here.
Our measurements are relative transforms between poses, either odometry (relative transforms between consecutive notes) or loop closure (relative transforms between arbitrary nodes).

$$\mathbf{e}_i(\mathbf{x}) = \mathbf{z}_i - f_i(\mathbf{x}) = \mathbf{z}_i - \hat{\mathbf{z}}_i$$

In SLAM, it is desirable to rewrite the above error term explicitly as follows to denote that these terms correspond to relative transformation between two nodes:

$$\mathbf{e}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{u}_{ij} - f(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{u}_{ij} - \hat{\mathbf{u}}_{ij}$$

where $\mathbf{u}_{ij}$'s are control inputs, either odometry (relative transforms between consecutive notes) or loop closure (relative transforms between arbitrary nodes). And $f(\mathbf{x}_i, \mathbf{x}_j)$ is our "predicted" measurement $\hat{\mathbf{u}}_i$.

1D SLAM

> Let us first take a look at the residual for 1D SLAM to build up the intuition:

Say in the question: A robot navigates in a 1D environment and closes the loop by returning to the starting point. Our measurements or observed values are given by:

For full solution of this problem, please visit this page. HOWEVER, please note while the convention used there is **not** incorrect ($f\left(\mathbf{x}_i, \mathbf{u}_i\right) = \mathbf{x}_i + \mathbf{u}_i$) and mathematically same as convention used in this page ($f\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{x}_j - \mathbf{x}_i$), ***it is different from the standard convention*** we use in this page. **Convention used in this page is standard one used in SLAM papers usually**. So you will have to tune your understanding accordingly.

Quantity	Observed / Measured
$\mathbf{u}_0$	1.1
$\mathbf{u}_1$	1.0
$\mathbf{u}_2$	1.1
$\mathbf{u}_3$	-2.7
$\mathbf{u}_{0,4}$	0.0

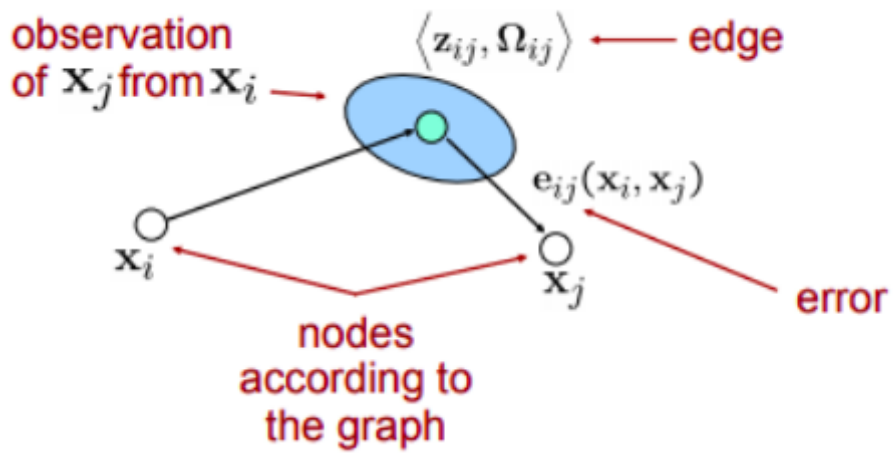
Then our residual for a simple 1D model given by $\hat{\mathbf{u}}_{ij} = f\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{x}_j - \mathbf{x}_i$ will look like $\mathbf{e}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{u}_{ij} - \left(\mathbf{x}_j - \mathbf{x}_i\right)$.
The overall residual vector will be:

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \mathbf{u}_0 - f\left(\mathbf{x}_0, \mathbf{x}_1\right) \\ \mathbf{u}_1 - f\left(\mathbf{x}_1, \mathbf{x}_2\right) \\ \mathbf{u}_2 - f\left(\mathbf{x}_2, \mathbf{x}_3\right) \\ \mathbf{u}_3 - f\left(\mathbf{x}_3, \mathbf{x}_4\right) \\ \mathbf{u}_{0,4} - f\left(\mathbf{x}_0, \mathbf{x}_4\right) \\ \mathbf{x}_0 - \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_0 + \mathbf{x}_0 - \mathbf{x}_1 \\ \mathbf{u}_1 + \mathbf{x}_1 - \mathbf{x}_2 \\ \mathbf{u}_2 + \mathbf{x}_2 - \mathbf{x}_3 \\ \mathbf{u}_3 + \mathbf{x}_3 - \mathbf{x}_4 \\ \mathbf{u}_{0,4} + \mathbf{x}_0 - \mathbf{x}_4 \\ \mathbf{x}_0 - 0 \end{pmatrix}$$

The last line is a "prior" constraint that anchors the first pose $\mathbf{x}_0$ at the origin. We typically initialize our system with odometry constraints, so our residual will look like $\mathbf{f}(\mathbf{x_{init}}) = (0 \quad 0 \quad 0 \quad 0 \quad -0.5 \quad 0)^T$.

Our final goal is:

$$\begin{aligned} \mathrm{x}^* &= \underset{\mathrm{x}}{\operatorname{argmin}} \sum_{ij} \mathrm{e}_{ij}^T \Omega_{ij} \mathrm{e}_{ij} \\ &= \underset{\mathrm{x}}{\operatorname{argmin}} \mathbf{f}(\mathbf{x})^\top \boldsymbol{\Omega} \mathbf{f}(\mathbf{x}) \end{aligned}$$



For full solution of this problem, please visit this page. HOWEVER, please note while the convention used there is **not** incorrect ($f\left(\mathbf{x}_i, \mathbf{u}_i\right) = \mathbf{x}_i + \mathbf{u}_i$) and mathematically same as convention used in this page ($f\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{x}_j - \mathbf{x}_i$), ***it is different from the standard convention*** we use in this page. **Convention used in this page is standard one used in SLAM papers usually**. So you will have to tune your understanding accordingly.

Now it is straightforward for 1D case as you've seen above. However, in 2D/3D SLAM, it is more involved than this as follows:

2D/3D SLAM

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

If you want to understand the above term in detail, visit this page. A simplistic explanation is as follows:

Following our above terminology, $\mathbf{X}_i^{-1}\mathbf{X}_j$ is our "predicted measurement" whereas $\mathbf{Z}_{ij}$ is our actual measurement. However, it doesn't make sense to subtract these directly because they are no longer in Euclidean space (subtracting 2 transformation matrices need not be a transformation matrix); rather we pre-multiply our "predicted measurement" by $\mathbf{Z}_{ij}^{-1}$ and finally instead of (*ideally*) obtaining a 0 vector, we must obtain **identity matrix**. We then do $\mathrm{t2v}$ (transformation to vector) and now it should *ideally* be a 0 vector. For more details, visit this page.

> **That's it, that's our error function for Graph SLAM when we are optimizing only for poses.**

2. Graph SLAM with landmarks (3D points)

Optimizing variables: poses and landmarks/3D points here
Our measurements are typically local 3D/2D measurements of a landmark from current pose of the robot. It could even be 1D measurement as described here (but the point here is, it is measurements of a landmark from the current pose of the robot).

> This is what was taught as "ICP-SLAM" in class as explained below.

Optimizing for 3D points (2D points in the 2D world) or landmarks is actually simpler than optimizing for poses because 3D points belong to Euclidean space (so you can directly subtract) while poses are Lie Groups.

Let us talk about 2D world. If $\mathbf{x}_i$ is robot pose $(\mathbf{t}_i, \theta) = (x, y, \theta)$ and $\mathbf{x}_j$ **is landmark** (x, y) in **global** coordinate system; and $\mathbf{z}_{ij}$ is relative observation from my current **local** robot location $(\delta x, \delta y)$

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

▼ Also do note that this error term is not the only possible formulation:

For example in case of bearing only observations, $\hat{\mathbf{z}}_{ij}$ will be:

$$\hat{\mathbf{z}}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \operatorname{atan}\frac{\left(\mathbf{x}_j - \mathbf{t}_i\right)\cdot y}{\left(\mathbf{x}_j - \mathbf{t}_i\right)\cdot x} - \theta_i$$

If you went through the above link, the above term must be clear to you that it is simply the translation component of the predicted measurement. If not, see this.

> There you have it. The above term is the residual for landmarks, whereas the one in previous section is for poses. **If you are optimizing over both, your final error is simply a summation over both kinds of residuals.**

$$\mathrm{x}^* = \underset{\mathrm{x}}{\operatorname{argmin}} \sum_{ij} \mathrm{e}_{ij}^T \Omega_{ij} \mathrm{e}_{ij}$$

ICP-SLAM for Mobile Robotics 2020 Students

This is the link to ICP-SLAM notes. Now, you can see that ICP-SLAM taught to you in MR 2020 is simply an unnecessarily complicated way of writing the same "Graph SLAM with landmarks" optimization function:

The important term in our 1.1 was: (Here, 0 is assumed to be the global frame)

$$\vec{X}_{ij} - \widehat{\mathbf{T}}_{i0}\vec{X}_{0j}$$

Rewriting it in standard form (Rewriting it in global frame terms and changing X_j to $\mathbf{x}_j$ and X_{ij} to $\mathbf{z}_{ij}$ (They mean **exactly the same)**:

$$\begin{aligned}\mathbf{e}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) &= \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij} \\ &= \vec{X}_{ij} - \widehat{\mathbf{T}}_{i0}\vec{X}_{0j} \\ &= \vec{X}_{ij} - \widehat{\mathbf{T}}_{0i}^{-1}\vec{X}_{0j} \\ &= \mathbf{z}_{ij} - \widehat{\mathbf{T}}_i^{-1}\mathbf{x}_j\end{aligned}$$

By now, we already know that is same as:

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

> Voila! It is exactly the same formulation! It is just that the way it was introduced was a bit confusing. Calling it "ICP-SLAM" isn't entirely correct. It is simply instead an optimization problem where we are optimizing both poses and landmarks/3D points. ICP is just one of the ways through which we can get the initialization for the robot poses, the other way, for example, could be through odometry.

3. Bundle Adjustment / Structure from Motion

We have seen in these notes that our "reprojection error" is

Optimizing variables: poses and landmarks/3D points here.
Our measurements are 2D pixels on images.

$$\arg\min_{\vec{X}j,P_i} \sum_{i=1}^{M}\sum_{j=1}^{N} \left\| P_i \vec{X}_j - \vec{x}_{ij} \right\|^2$$

$$\arg\min_{Xj,P_i} \sum_{i=1}^{M}\sum_{j=1}^{N} \left\| \widehat{\vec{x}}_{ij} - \vec{x}_{ij} \right\|^2$$

Reprojection Error

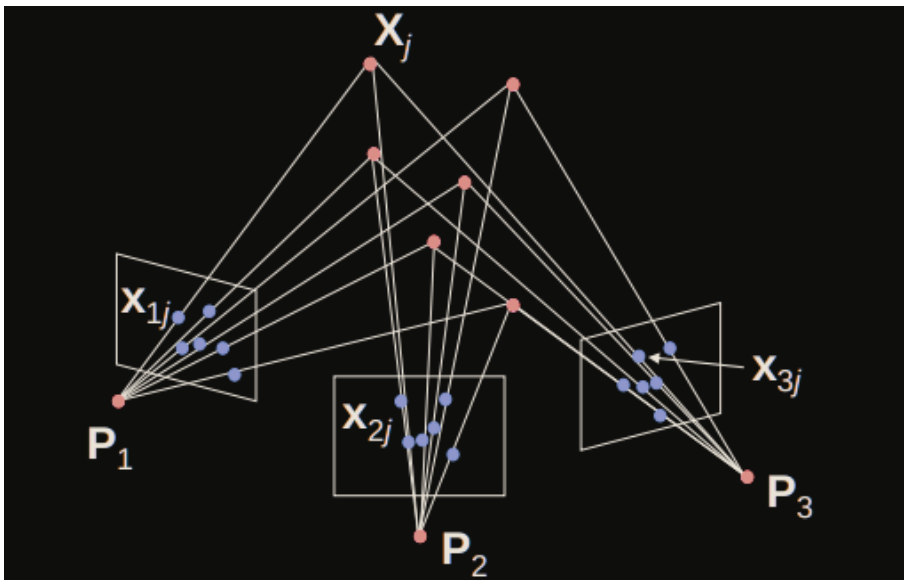
P_i is projection matrix of the i^{th} view, $\vec{X}_j$ is the j^{th} $3D$ point.

$\vec{x}_{ij}$ — **the observation**: the image (pixel locations) of the j^{th} pixel in the i^{th} image. *(using say SIFT matcher)*

$\widehat{\vec{x}}_{ij}$ — **the predicted projection**: of initial reconstruction of the j^{th} $3D$ point to the i^{th} view.

So the difference between "Graph SLAM with landmarks" and "Bundle Adjustment":

- Our measurements here are different: They are 2D pixels on images as represented by blue dots on the image.



- Our model is the camera projection model here,

> **mapping** 3D points **to** 2D pixels **through** the camera projection matrix.

Whereas **in landmark SLAM**, our model was

> **mapping *global*** 3D points **to *local*** measurements from the current pose of the robot **through** a ***global*** robot pose.

(local measurements could be 2D/3D and could even be 1D bearing measurements in case of 2D SLAM)

References

- References same as mentioned here.