Accurate Tracking with Fusion of Video and Radio Signals

Yuning Liu, Weiwei Li, Chi Chen, and Yuan Gao

University of New South Wales

March 1, 2019

Abstract

SLAM (Simultaneous Localization And Mapping) is a rising-discussion topic today. There are several excellent open-source software, for instances, PTAM, LSD-SLAM and ORB-SLAM, etc. After installing several softwares and digging out the limitation of different systems, We choose ORB-SLAM as our research topic. We also did some experienments trying to figured out the backbone of ORB-SLAM. Our objective is to optimzed the algorithm to enhance the system performance and mapping accuracy.

1 Introduction

The core of non-filtered based SLAM is basically to minimize the error between the data from predictions, the geometric relation between feature and camera position, and measurements, that is actual graphical input, using Least Square method (Equation 1).

$$x = \underset{x}{\operatorname{argmin}} \sum_{i \in \mathcal{X}} \|Ax - b\|^2 \tag{1}$$

where b is the measurements and A is the predictions using the variable we want to obtain. We wants to find out a optimal x which can minimize the distance between our measurements and predictions. The solutions can be solved by Newton method, gradient descent, etc.

In ORB-SLAM case, the LS problem, also known as Bundle Adjustment, is specified as minimizing the reprojection error which is shown as below:

$$\{R, t\} = \underset{R,t}{\operatorname{argmin}} \sum_{i \in \mathcal{X}} \rho(\left\| x^i_{(\cdot)} - \pi_m(RX^i + t) \right\|^2_\Sigma) \tag{2}$$

where ρ is the robust Huber cost function and Σ is the covariance matrix associated to the scale of the keypoint. The projection function π_m for monocular camer is definded as follow:

$$\pi_m = \left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \right) = \begin{bmatrix} f_x \frac{X}{Z} + c_x \\ f_y \frac{Y}{Z} + c_y \end{bmatrix} \tag{3}$$

Bundle adjustment optimizes camera pose R and translation t that minimize the reprojection error. In every keyframe, camera pose R and translation t are store in a 4-by-4 matrix. The fourth rows is $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$ which is for homogeneous coordinates. The complete 4-by-4 matrix is shown below:

$$\begin{bmatrix} R_{3\times3} & t_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}$$

The detailed mathmatical explanation will be discussed in the following section.

2 Related Mathmatical Work

2.1 Reprojection Error

As shown in Figure 1, P is our feature point which can be seen in two keyframes. P'_1 and P'_2 are point P captured on two keyframe and P_1 and P_2 are the intersection points of equalvalent camera plane and the lines ($O'P$ and $O''P$) between camera center (O' and O'') and feature point P. The reprojection error is the distance between P_1 and P'_1 (as shown in the green). The reprojection error can be optimized by adjusting the camera pose R and translation t, which is discussed above.

2.2 Epipolar Geometry

It is mentioned that the reprojection error can be optimized by camera pose and translation matrix in the above seesion, and in this part, the Geometrical relation will be analysed by using mathematical method. the equation relation is shown as below:

$$p_2^T K^{-T} t^\wedge R K^{-1} p_1 = 0 \tag{4}$$

where K, R, t represent camera internal metrix, camera pose and translation.

2.3 Rigid transformation

In mathematics, the rigid transformation, also called Euclidean transformation, is a geometric transformation in Euclidean space, and it do not change the Euclidean distance between two points. The camera can be seen as rigid body when it moving, which means the length and angle always keep stable at any frame of reference, just like a Rigid transformation.

Considering the vector a at world coordinate system, after once rotation R and translation t, Euclidean transformation include two parts, rotation and translation. Firstly consider rotation. Set the vector $\vec{a}$, the coordinate points are

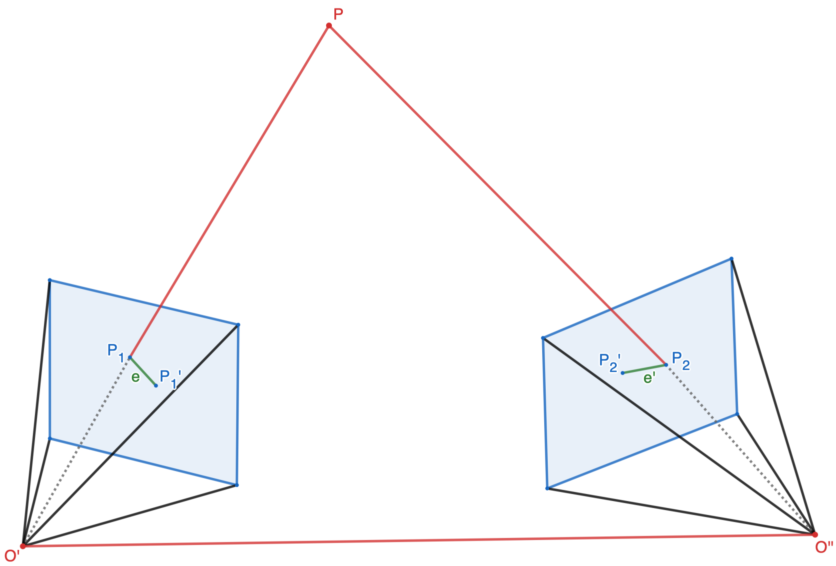

Figure 1: Epipolar Geometry and Reprojection Error

$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}$ and $\begin{bmatrix} a_1' & a_2' & a_3' \end{bmatrix}^T$ at two different frame of reference. One of orthogonal basis is $\begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}$, after the rotation, it becomes $\begin{bmatrix} e_1' & e_2' & e_3' \end{bmatrix}$.

$$\begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} e_1' & e_2' & e_3' \end{bmatrix} \begin{bmatrix} a_1' \\ a_2' \\ a_3' \end{bmatrix}$$

Then do left multiplication at the same time by $\begin{bmatrix} e_1^T & e_2^T & e_3^T \end{bmatrix}^T$, and the coefficient at left side become identity matrix $\boldsymbol{I}$. We can get the following equation:

$$\begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix} = \begin{bmatrix} e_1^T e_1' & e_1^T e_2' & e_1^T e_3' \\ e_2^T e_1' & e_2^T e_2' & e_2^T e_3' \\ e_3^T e_1' & e_3^T e_2' & e_3^T e_3' \end{bmatrix} \begin{bmatrix} a_1' \\ a_2' \\ a_3' \end{bmatrix} \triangleq \boldsymbol{R}\boldsymbol{a}'$$

The matrix at middle define as rotation matrix $\boldsymbol{R}$, which is composed by two set of orthogonal basis $\begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}$ and $\begin{bmatrix} e_1' & e_2' & e_3' \end{bmatrix}$. The matrix $\boldsymbol{R}$ describes rotation itself, so it also called rotation matrix. Meanwhile, the rotation matrix is an orthogonal matrix with determinant equal to 1. So the set of rotation matrices is defined as follows:

$$SO(n) = \{\boldsymbol{R} \in \mathbb{R}^{n \times n} | \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}, det(\boldsymbol{R}) = 1\} \tag{5}$$

where $SO(n)$ is the meaning of special orthogonal group. In particular, $SO(n)$ is the rotation of three-dimensional space. In the Euclidean transformation, there is a translation in addition to the rotation. Consider the vector $\boldsymbol{a}$ at world coordinate system, after once rotation (present by $\mathbf{R}$) and once translation $\boldsymbol{t}$, getting $\boldsymbol{a}'$. Then put the rotation and translation together we have the following equation.

$$\boldsymbol{a}' = \boldsymbol{R}\boldsymbol{a} + \boldsymbol{t} \tag{6}$$

The $\mathbf{t}$ is defined as translation vector. Compared to rotation, translation is just adding together. By the above equation, we use a rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$ to completely describe the coordinate transformation relationship of an Euclidean space.

2.4 Homogeneous coordinates

At the above, we use a rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$ to completely describe the coordinate transformation relationship of an Euclidean space. But there is a problem. When do several times transformation,the equation will become complex and not linear relation. So here introduce the Homogeneous coordinates and transformation matrix rewriting.

$$\begin{bmatrix} \boldsymbol{a}' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ 1 \end{bmatrix} \triangleq \boldsymbol{T} \begin{bmatrix} \boldsymbol{a} \\ 1 \end{bmatrix} \tag{7}$$

Thus homogeneous coordinate is adding 1 to the end of a 3-dimensional vector then it become a 4-dimensional vector. As for the 4-dimensional vector, we put the rotation and transformation into one matrix. At the equation, the matrix $\boldsymbol{T}$ defined as transform matrix, and using $\tilde{\boldsymbol{a}}$ to present homogeneous coordinates of $\boldsymbol{a}$.

2.5 Transformation matrix

For the transformation matrix, it has special structure: top left corner is rotation matrix, top right corner is translation vector, bottom left corner is 0 vector, bottom right corner is 1. Such kind of matrix also called as Special Euclidean Group and written as follow.

$$SE(3) = \{\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} | \boldsymbol{R} \in SO(3), \boldsymbol{t} \in \mathbb{R}^3\} \tag{8}$$

2.6 Review

We introduce the transformation between different frame of reference is described by Euclidean transformation, which is composed by rotation and translation. Rotation can be described by matrix $SO(3)$, translation is described by $\mathbb{R}^3$ vector. If putting the rotation and translation into one matrix, it formed transformation matrix $SE(3)$.

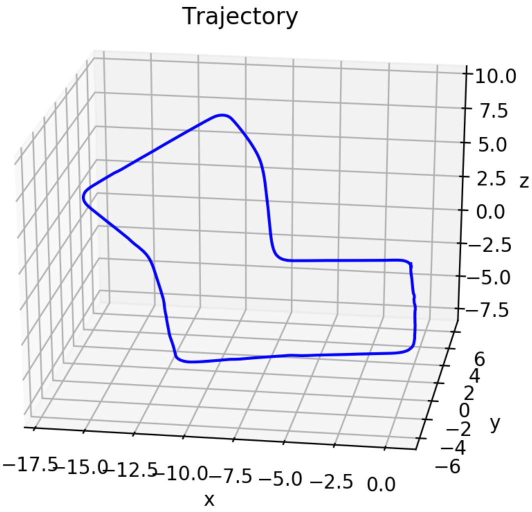

Figure 2: Trajectory from Dataset

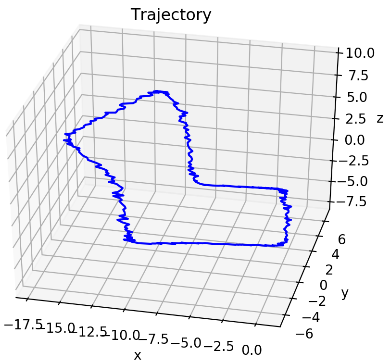

Figure 3

3 Experienment

Currently, we don't have Wi-Fi localization data for experienment. Instead, we decided to use offline dataset from *** and output the map data which can be regarded as the wifi data. To be noticed that the offline dataset should have closed loop otherwise it would be affected by the camera drift. Global bundle adjustment is engaged after the loop closure which means the drift is reduced or, in some extent, elminated. For simplication, we choose dataset 07 from *** and the map is shown at the Figure 2.

First, since all the map data are stored in the keyframes which is consist of camera pose R and translation t, we modified the original software and output the keyframes' data and timestamps. After that, we use Python script to process the data based on the math we discussed above and simulate the noise from the real Wi-Fi localization device. The processed map data is shown at the Figure 3. The script is attached in the appendix.

4 Objective

Our objective is to fuse localization data given by Wi-Fi anchor to enhance our system performance and reduce trajectory drift using monocular cameara. The mathmatical representation is given by Equation

$$\{R, t\} = \underset{R,t}{\operatorname{argmin}} \sum_{i \in \chi} \rho(\left\| x^i_{(\cdot)} - \pi_m(RX^i + t) \right\|^2_\Sigma + \alpha \left\| \vec{x} - \vec{x}_{wifi} \right\|^2) \tag{9}$$

5 Others