3D Camera Localization with Extended Kalman Filter
Probabilistic Robotics Course

Norman Di Palo
MSc. in Artificial Intelligence and Robotics

February 11, 2018

1 Introduction

In this report, I will discuss the development of the 3D Camera Localization project, briefly showing the steps needed to develop the Extended Kalman Filter applied to the case of a 3D moving robot with a camera, a known map of landmarks but unknown associations.

2 Dynamical Equations and Jacobians

The robot can move in the 3D space, and the inputs, given in the .g2o file, are increments espressed in the relative frame of the robot, that can be directly added to the position of the robot. The angles increments are given as quaternions, thus I used an algorithm to translate those into Euler angles. Hence, the state space of the robot is $SE(3)$.
The dynamical equation describing the position evolutions is then:

$$f_{1,3} = x_{t+1} = x_t + R(\theta)R(\phi)R(\psi)u_t$$

that transforms the input vector from robot frame to world frame. The angles increments are simply:

$$f_{3,6} = \theta_{t+1} = \theta_t + u_{\theta,t}$$

In order to use the Extended Kalman Filter, we need to find the A and B Jacobians, respectively $\frac{\partial f}{\partial x_i}$ and $\frac{\partial f}{\partial \theta_i}$.
The A matrix is 6x6, being it *state* x *state*. The B matrix is 6x6 aswell, being it *state* x *input*.

$$A = \begin{bmatrix} I_{3x3} & R'(\theta)R(\phi)R(\psi)u_{1,3} & R(\theta)R'(\phi)R(\psi)u_{1,3} & R(\theta)R(\phi)R'(\psi)u_{1,3} \\ 0_{3x3} & & I_{3x3} & \end{bmatrix}$$

where $R'(\theta)R(\phi)R(\psi)u_{1,3}$ are column vectors.

$$B = \begin{bmatrix} R_{\theta,\phi,\psi} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix}$$

where ${}^{world}R_{(\theta,\phi,\psi),robot}$ is the rotation matrix from robot frame to world frame.

3 Observation Equations and Jacobians

The observations of the robot are camera projections of the 3D landmarks present in the map. The camera has a relative displacement (0.3 along the z robot direction) and orientation w.r.t. the robot frame. It also has a K matrix based on its focal lenght and principal points. Finally, there is the projective equation, considering that the camera can also perceive depth:

$$proj = \begin{bmatrix} \frac{x_{cam}}{z_{cam}} \\ \frac{y_{cam}}{z_{cam}} \\ z_{cam} \end{bmatrix}$$

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

The complete obesrvation equation is

$$h(p_w) = proj(K^{cam}R_{rob}({}^{rob}R_{world}(p_w - x_{rob}) - (0,0,0.3))$$

To evaluate its Jacobian, C, we can use the chain rule.

$$\frac{\partial h}{\partial x} = \frac{\partial proj(p_{cam})}{\partial p_{cam}} \frac{\partial p_{cam}}{\partial x}$$

.

$$\frac{\partial proj(p_{cam})}{\partial p_{cam}} = \begin{bmatrix} -1/z & 0 & \frac{x}{z^2} \\ 0 & -1/z & \frac{y}{z^2} \\ 0 & 0 & -1 \end{bmatrix}$$

We know that

$$p_{cam} = K^{cam}R_{rob}({}^{rob}R_{world}(p_w - x_{rob}) - (0,0,0.3))$$

so the second partial Jacobian is

$$\frac{\partial p_{cam}}{\partial x} = \begin{bmatrix} \frac{\partial p_{cam}}{\partial x_{1,3}} & \frac{\partial p_{cam}}{\partial \theta} & \frac{\partial p_{cam}}{\partial \phi} & \frac{\partial p_{cam}}{\partial \psi} \end{bmatrix}$$

$$\frac{\partial p_{cam}}{\partial x_{1,3}} = -K^{cam}R_{rob}({}^{rob}R_{world})$$

the former is a $3x3$ matrix. The latters are $3x1$ vectors.

$$\frac{\partial p_{cam}}{\partial \theta} = K^{cam} R_{rob}(R'(\theta)R(\phi)R(\psi)(p_w - x_{rob})$$

$$\frac{\partial p_{cam}}{\partial \phi} = K^{cam} R_{rob}(R(\theta)R'(\phi)R(\psi)(p_w - x_{rob})$$

$$\frac{\partial p_{cam}}{\partial \psi} = K^{cam} R_{rob}(R(\theta)R(\phi)R'(\psi)(p_w - x_{rob})$$

The total C Jacobian is thus $3x6$, $observationsxstate$.

4 Data Association

We have a map of unique landmarks, but we do not have an associaton observation-landmark during the localization phase. Hence, we need to adopt some heuristics to define which landmark has most probably caused which observation. I first tried to use an association based on Euclidean distance alone between $h(x)_i$ and z_i. To leverage the number of wrong association that can dramatically break the Kalman filter, I also implemented the Best Friends heuristic: after building a cost association matrix between landmarks and association, a landmark is coupled with an observation only if the couple is a minimum both of the column and of the row. Intuitively, that means that the landmark is the best prediction for a particular observation, and that observation is the best prediction for that landmark too. This resulted in a robust association algorithm that allowed the filter to smoothly work.

5 Localization with EKF

To actually perform the localization of the robot, we follow the usual steps of the Extended Kalman Filter. First, we update the state, adding to the current state mean the increments given by the dynamical equations. Then we update the covariance of the state.

$$\mu_{t,t+1} = \mu_t + f(x_t, u_t)$$
$$\Sigma_{t,t+1} = A\Sigma_{x,t,t}A^T + B\Sigma_{u,t,t}B^T$$

After this update, we compute the expected observations, $h(\mu_x)$ and evaluate the associations between observations and landmarks. Then we compute the C Jacobian and then the Kalman gain.

$$K_t = \Sigma_{x,t+1,t}C^T(\Sigma_z + C\Sigma_{t+1,t}C^T)^{-1}$$

$$\mu_{t+1,t+1} = \mu_{t+1,t} + K(z - h(\mu_x))$$
$$\Sigma_{x,t+1,t+1} = (I - KC)\Sigma_{x,t+1,t}$$

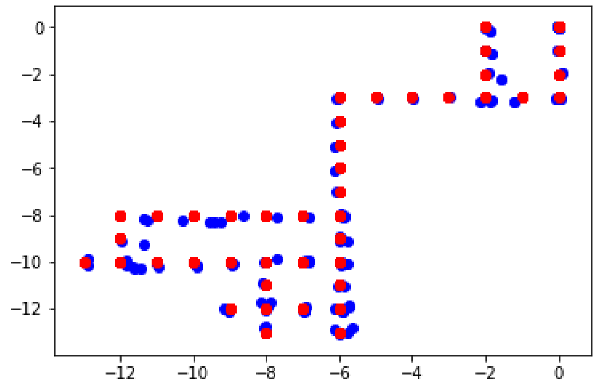

Figure 1: Localization with EKF (blue) and ground truths (red).

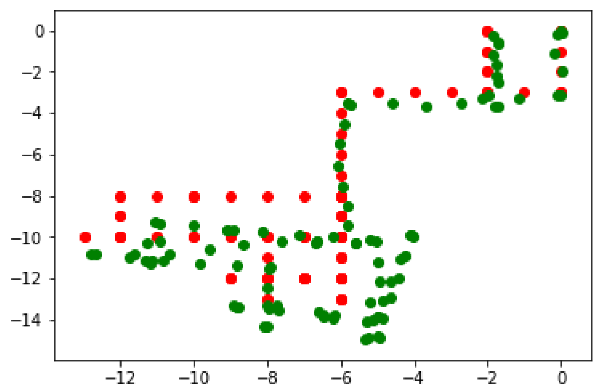

Figure 2: Localization with encoder ticks (green) and ground truths (red).

The EKF greatly improves the accuracy of the trajectory tracking. Here are some figures showing the localization done by the EFK and by the encoder ticks alone.