Camera EKF Documentation

Benjamin Wong[*]

November 2023

1 Camera Projection

We follow the pinhole camera model for projecting an point in 3D space to the image plane. In this section, the 3D coordinate of the point is defined in the camera's coordinate frame, where z-axis is forward from the camera, and y-axis is pointing downward from the camera.

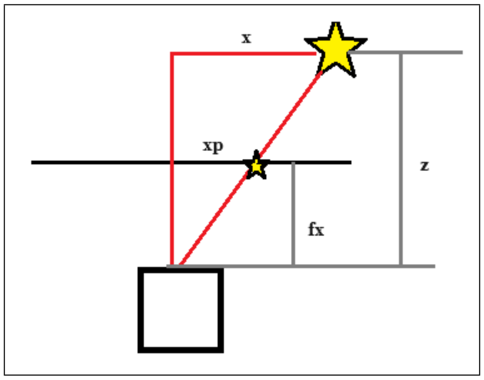

Figure 1: similar triangle of camera projection

The equation can be derived by similar triangle shown in figure 1, which can be put into matrix form as follow

$$\begin{bmatrix} x_{\text{pixel}} \\ y_{\text{pixel}} \\ 1 \end{bmatrix} = \frac{1}{z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{p/c} \\ y_{p/c} \\ z_{p/c} \end{bmatrix} \tag{1}$$

Where f_x, f_y are the focal lengths in the x and y direction correspondingly in unit of pixel; c_x, c_y are the offset from the center of the pinhole to the top left corner of the image in unit of pixel. The matrix is called the camera matrix,

[*] bycw@uw.edu

or intrinsic matrix, which can be obtained by camera calibration. To keep it generic, we use the notation K instead.

$$\begin{bmatrix} x_{\text{pixel}} \\ 1 \end{bmatrix} = \frac{1}{z} K x_{p/c} \tag{2}$$

2 Global Frame

To project from any arbitrary global coordinate frame, we need to introduce the extrinsic matrix. The extrinsic matrix describes the pose of the camera in the global frame in the form of coordinate transformation. Which is represented by the rigid transformation matrix

$$M = \begin{bmatrix} R_{3\times 3} & T_{3\times 1} \\ 0 & 1 \end{bmatrix} \tag{3}$$

where R is a rotation matrix, T is a translation vector. A point in the global coordinate can be transformed to the camera coordinate as such

$$\begin{bmatrix} x_{p/c} \\ y_{p/c} \\ z_{p/c} \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3\times 3} & T_{3\times 1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{p/w} \\ y_{p/w} \\ z_{p/w} \\ 1 \end{bmatrix} \tag{4}$$

To get $x_{/c}$, we need to remove the 1, which can be done by removing the last row of the rigid transformation

$$\begin{bmatrix} x_{p/c} \\ y_{p/c} \\ z_{p/c} \end{bmatrix} = \begin{bmatrix} R_{3\times 3} & T_{3\times 1} \end{bmatrix} \begin{bmatrix} x_{p/w} \\ y_{p/w} \\ z_{p/w} \\ 1 \end{bmatrix} \tag{5}$$

Note that the rigid transformation matrix is a closed operator, where it follows the form

$$\begin{bmatrix} R_3 & T_3 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_1 & T_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_2 & T_2 \\ 0 & 1 \end{bmatrix} \tag{6}$$

For SLAM purposes, we separate the coordinate transform into global to robot $M_{w\to r}$ and robot to camera $M_{r\to c}$. for a camera fixed on a robot, $M_{r\to c}$ is a fixed value, which can be obtained from the URDF. For $M_{w\to r}$, we first focus on the inverse $M_{w\to r}^{-1} = M_{r\to w}$. This is parameterized by the pose of the robot in the global frame. We assume a ground robot running on a flat ground $z_r = 0$, which the transformation can be represented by

$$M_{r\to w} = \begin{bmatrix} R_{r\to w} & T_{r\to w} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_r & -\sin\theta_r & 0 & x_r \\ \sin\theta_r & \cos\theta_r & 0 & y_r \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where $\begin{bmatrix} x_r & y_r & \theta_r \end{bmatrix}$ is the robot's pose. The inverse is

$$M_{w\to r} = M_{r\to w}^{-1} = \begin{bmatrix} R_{r\to w}^{-1} & R_{r\to w}^{-1} T_{r\to w} \\ 0 & 1 \end{bmatrix} \tag{8}$$

The full transformation function is

$$x_{p/c} = \begin{bmatrix} R_{r\to c} & T_{r\to c} \end{bmatrix} M_{w\to r} \begin{bmatrix} x_{p/w} \\ 1 \end{bmatrix} \tag{9}$$

3 Observation Function

The observation vector we can obtain from the camera are the pixel location of the feature and the depth from the camera to the feature.

$$z = \begin{bmatrix} x_{\text{pixel}} \\ y_{\text{pixel}} \\ z_{p/c} \end{bmatrix} \tag{10}$$

Using the results from the previous two sections, we can derive the observation function, $h(x)$, with slight modification to the projection function

$$\begin{bmatrix} x_{\text{pixel}} \\ y_{\text{pixel}} \\ z_{p/c} \end{bmatrix} = h\left(\begin{bmatrix} x_{p/w} \\ y_{p/w} \\ z_{p/w} \\ x_r \\ y_r \\ \theta_r \end{bmatrix}\right) = \begin{bmatrix} 1/z_{p/c} & 0 & 0 \\ 0 & 1/z_{p/c} & 0 \\ 0 & 0 & 1 \end{bmatrix} K \begin{bmatrix} R_{r\to c} & T_{r\to c} \end{bmatrix} M_{w\to r} \begin{bmatrix} x_{p/w} \\ 1 \end{bmatrix} \tag{11}$$

4 Mapping Function

While the Kalman gain takes care of mapping the observation error to the global coordinate, the mapping function is necessary for initializing the location of a newly observed feature. To get the location in the camera reference frame, take the inverse

$$\begin{bmatrix} x_{p/c} \\ y_{p/c} \\ z_{p/c} \end{bmatrix} = K^{-1} \begin{bmatrix} x_{\text{pixel}} \\ y_{\text{pixel}} \\ 1 \end{bmatrix} z_{p/c} \tag{12}$$

$$\begin{bmatrix} x_{p/w} \\ y_{p/w} \\ z_{p/w} \end{bmatrix} = \begin{bmatrix} R_{r\to w} & T_{r\to w} \end{bmatrix} M_{c\to r} \begin{bmatrix} x_{p/c} \\ y_{p/c} \\ z_{p/c} \\ 1 \end{bmatrix} \tag{13}$$

5 Extended Kalman filter

For EKF, the Kalman gain is calculated as such

$$K = \bar{\Sigma} H^T (H \Sigma H^T + R)^{-1} \tag{14}$$

Where H is the jacobian of the observation function. R is the observation uncertainty. Assume an independent Gaussian noise on the pixel space and depth measurement

$$R = \mathrm{Var}\left\{ \begin{bmatrix} x_{\text{pixel}} \\ y_{\text{pixel}} \\ z_{p/c} \end{bmatrix} \right\} = \begin{bmatrix} \sigma^2_{x_{\text{pixel}}} & 0 & 0 \\ 0 & \sigma^2_{y_{\text{pixel}}} & 0 \\ 0 & 0 & \sigma^2_{z_{p/c}} \end{bmatrix} \tag{15}$$

we wish to find the linearized mapping from global frame to the expected observation

$$\bar{z} = \begin{bmatrix} \bar{x}_{\text{pixel}} \\ \bar{y}_{\text{pixel}} \\ \bar{z}_{p/c} \end{bmatrix} = h(\mu) \approx h(\mu_0) + H(\mu - \mu_0) \tag{16}$$

Where μ is the current estimated location of the robot and the feature.

$$\mu = \begin{bmatrix} \bar{x}_r \\ \bar{y}_r \\ \bar{\theta}_r \\ \bar{x}_{p/w} \\ \bar{y}_{p/w} \\ \bar{z}_{p/w} \end{bmatrix} \tag{17}$$

To avoid complications of writing all matrix component explicitly, we will find the Jacobian with chain rule.

The first Jacobian we need is the transformation from world frame to robot frame

$$J_{w \to r} = \bigtriangledown_\mu \bar{x}_{/r} = \begin{bmatrix} -c & -s & c(\bar{y}_{p/w} - \bar{y}_r) + s(\bar{x}_r - \bar{x}_{p/w}) & c & s & 0 \\ s & -c & c(\bar{x}_r - \bar{x}_{p/w}) + s(\bar{y}_r - \bar{y}_{p/w}) & -s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

translation from robot frame to camera frame is affine, the Jacobian is simply the rotation part of the transformation matrix

$$J_{r \to c} = R_{r \to c} \tag{19}$$

The final Jacobian is the modified camera projection

$$J_c = \begin{bmatrix} 1/z_{p/c} & 0 & -x_{p/c}/z^2_{p/c} \\ 0 & 1/z_{p/c} & -y_{p/c}/z^2_{p/c} \\ 0 & 0 & 1 \end{bmatrix} K \tag{20}$$

According to chain rule, the overall Jacobian of the observation function is the product of all the step wise jacobians

$$H = J_c J_{r \to c} J_{w \to r} \tag{21}$$

6 Handling Orientation in 1 d.o.f. SO(3)

Assume that the apriltags are perfectly vertical, with θ being the orientation of the normal vector of the tag in the world frame

$$\tau_{\text{tag}} = \begin{bmatrix} 0 \\ 0 \\ \theta_{\text{tag}} \end{bmatrix} \in \mathfrak{so}(3) \tag{22}$$

and the robot orientation on ground

$$\tau_r = \begin{bmatrix} 0 \\ 0 \\ \theta_r \end{bmatrix} \in \mathfrak{so}(3) \tag{23}$$

The *Exp* mapping for the tag and robot are

$$\bar{R}_{\text{tag}} = \text{Exp}(\tau_{\text{tag}}) = \begin{bmatrix} \cos\theta_{\text{tag}} & -\sin\theta_{\text{tag}} & 0 \\ \sin\theta_{\text{tag}} & \cos\theta_{\text{tag}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{24}$$

$$\bar{R}_r = \text{Exp}(\tau_{\text{tag}}) = \begin{bmatrix} \cos\theta_r & -\sin\theta_r & 0 \\ \sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{25}$$

The transformation from the camera to the robot is available from the URDF

$$R_{c\to r} \in SO(3) \tag{26}$$

The homography returns a rotation matrix from the camera to the tag

$$R_{\text{tag}/c} \in SO(3) \tag{27}$$

by convecnsion the normal vector of the tag is the z axis.we will transform the frame such that x-axis is the normal.

$$R_{\text{tag}/c} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} R_{\text{tag}/c} \tag{28}$$

for simplicity, we find the error on the camera's tangent space

$$e = \tau_{\text{tag}/c} - \bar{\tau}_{\text{tag}/c} \tag{29}$$

such that the jacobian of the observation function is the negative jacobian of the error function so we can use the standard EKF formulation.

The observation function is

$$\begin{aligned} h(\theta_r, \theta_{\text{tag}}) &= \bar{\tau}_{\text{tag}/c} \\ &= \text{Log}(\bar{R}_{\text{tag}/c}) \\ &= \text{Log}(R_{c\to r}^T \bar{R}_r^T \bar{R}_{\text{tag}}) \\ &= (\bar{R}_{\text{tag}}) \ominus (\bar{R}_r R_{c\to r}) \end{aligned} \tag{30}$$

We want to get the jacobian with respect to θ_r and θ_{tag}, using the jacobians from [1]

$$\begin{aligned} J_{\theta_r} &= \nabla_{\theta_r} h \\ &= J^{h}_{\bar{R}_r R_{c\to r}} \cdot J^{\bar{R}_r R_{c\to r}}_{\bar{R}_r} \cdot J^{\bar{R}_r}_{\theta_r} \\ &= -J_l^{-1}(\bar{\tau}_{\text{tag/c}}) \cdot R_{c\to r}^T \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned} \tag{31}$$

$$\begin{aligned} J_{\theta_{\text{tag}}} &= \nabla_{\theta_{\text{tag}}} h \\ &= J^{h}_{\bar{R}_{\text{tag}}} \cdot J^{\bar{R}_{\text{tag}}}_{\theta_{\text{tag}}} \\ &= J_r^{-1}(\bar{\tau}_{\text{tag/c}}) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned} \tag{32}$$

These Jacobians can be used in the EKF as such

$$H = \begin{bmatrix} J_r & J_{\text{tag}} \end{bmatrix} \tag{33}$$

$$d\tau_c = \tau_{\text{tag}/c} - \bar{\tau}_{\text{tag}/c} \tag{34}$$

$$K = \Sigma H^T (H \Sigma H^T + R) \tag{35}$$

$$\tau = \tau + K d\tau_c \tag{36}$$

$$\Sigma = (I - KH)\Sigma \tag{37}$$

In this formulation, the covariance Σ is defined on the Lie algebra. Since we assumed planar rotation on the tags and the robot, all tangent spaces coincides on the same plane and which space we choose to represent the covariance is irrelevant.

7 Handling Orientation in SO(3)

For full d.o.f., we use the SO(3) representation through out the estimation

$$\begin{aligned} J_{\bar{R}_r} &= \frac{D}{D\bar{R}_r} h \\ &= J^{h}_{\bar{R}_r R_{c\to r}} \cdot J^{\bar{R}_r R_{c\to r}}_{\bar{R}_r} \\ &= -J_l^{-1}(\bar{\tau}_{\text{tag/c}}) \cdot R_{c\to r}^T \end{aligned} \tag{38}$$

$$\begin{aligned} J_{\bar{R}_{\text{tag}}} &= \nabla_{\theta_{\text{tag}}} h \\ &= J^{h}_{\bar{R}_{\text{tag}}} \\ &= J_r^{-1}(\bar{\tau}_{\text{tag/c}}) \end{aligned} \tag{39}$$

$$H = \begin{bmatrix} J_r & J_{\text{tag}} \end{bmatrix} \tag{40}$$

$$d\tau_c = \tau_{\text{tag}/c} - \bar{\tau}_{\text{tag}/c} \tag{41}$$

$$K = \Sigma H^T (H \Sigma H^T + R) \tag{42}$$

$$R = R \oplus K d\tau_c \tag{43}$$

$$\Sigma = (I - KH)\Sigma \tag{44}$$

This formulation, the covariance is defined on the local tangent space of each pose.

References

[1] J. Solà, J. Deray, D. Atchuthan, A micro lie theory for state estimation in robotics (2021). arXiv:1812.01537.