

IVCalib: A Toolbox to Calibrate Between a Camera and an IMU

16-831 HW3: Fall 2011

Andrew Chambers, Natasha Kholgade, and Marynel Vázquez

1 Introduction

In this project, we estimate the rotation and translation between a rigidly connected camera and IMU using an unscented Kalman filter. Calibrating a camera-IMU setup is the first step in sensor fusion research. We are interested in creating an open-source calibration toolbox that can be used as a starting point by sensor fusion enthusiasts. The unscented Kalman filter helps to propagate the state of our system through process and measurement models that incorporate rotation, which is non-linear.

The state vector in our system consists of the camera-IMU rotation and translation, as well as the orientation and position of the IMU in the world coordinate system, the velocity in world space, the gravity vector, and the biases of the IMU's accelerometer and the gyroscope. The IMU-camera setup moves in the world space, and the camera images a set of world points with known locations in the world space. The imaged points form the measurement in our system. We predict the next state of the system from the current state by propagating a set of sigma points through the process model. We correct the state whenever a measurement is obtained from the camera.

We have performed an initial set of experiments where we assume we know everything about the system except the translation between the camera and the IMU. We estimate the translation using a modified process model for translation alone, and we analyze our performance in the face of noise in the measurement of the imaged points and the velocity.

2 UKF Formulation

In general, we follow the unscented Kalman filter formulation by Kelly and Sukhatme [1]. This section summarizes important aspects of the filtering process and some implementation details.

2.1 System State

In our system, we will ultimately estimate the world space position and orientation of the IMU, $\mathbf{p}_I^W(t)$ and $\bar{q}_I^W(t)$, the velocity of the IMU-camera setup in world space, \mathbf{v}^W , the biases for the gyroscope and accelerometer, $\mathbf{b}_g(t)$ and $\mathbf{b}_a(t)$, the gravity vector, $\mathbf{g}^W(t)$, and, of course, the position and orientation of the camera with respect to the IMU, \mathbf{p}_C^I and \bar{q}_C^I (i.e. the translation and rotation between the devices). The latter two are the calibration parameters we intend to estimate, and they do not change over time. The state vector, $\mathbf{x}_s(t)$ has 26×1 variables.

$$\mathbf{x}_t = [(\mathbf{p}_I^W(t))^T \quad (\bar{q}_I^W(t))^T \quad (\mathbf{v}^W)^T \quad (\mathbf{b}_g(t))^T \quad (\mathbf{b}_a(t))^T \quad (\mathbf{g}^W(t))^T \quad (\mathbf{p}_C^I)^T \quad (\bar{q}_C^I)^T] \quad (1)$$

The process model uses IMU measurements as control inputs along the vein of Kelly and Sukhatme [1]. We model the accelerometer and gyroscope biases as Gaussian random walk processes driven by zero mean noise \mathbf{n}_{aw} and \mathbf{n}_{gw} , with covariances \mathbf{Q}_{aw} and \mathbf{Q}_{gw} , and we model their additive noise by zero mean noise

\mathbf{n}_g and \mathbf{n}_a , with covariances \mathbf{Q}_a and \mathbf{Q}_g . The differential equations describing the state evolution are:

$$\begin{aligned}
\dot{\mathbf{p}}_I^W &= \mathbf{v}^W \\
\dot{\bar{q}}_I^W &= \frac{1}{2} \begin{bmatrix} 0 & -(\omega^I)^T \\ \omega^I & -[\omega^I]_{\times} \end{bmatrix} \bar{q}_I^W \\
\dot{\mathbf{v}}^W &= \mathbf{a}^W \\
\dot{\mathbf{g}}^W &= \mathbf{0}_{3 \times 1} \\
\dot{\mathbf{b}}_g &= \mathbf{n}_{gw} \\
\dot{\mathbf{b}}_a &= \mathbf{n}_{aw} \\
\dot{\mathbf{p}}_C^I &= \mathbf{0}_{3 \times 1} \\
\dot{\bar{q}}_C^I &= \mathbf{0}_{4 \times 1}
\end{aligned} \tag{2}$$

The measured acceleration and gravity are given as

$$\begin{aligned}
\omega_m &= \omega^I + \mathbf{b}_g + \mathbf{n}_g \\
\mathbf{a}_m &= \mathbf{C}^T (\bar{q}_I^W) (\mathbf{a}^W - \mathbf{g}^W) + \mathbf{b}_a + \mathbf{n}_a
\end{aligned} \tag{3}$$

The entire process noise mean and covariance are

$$\mathbf{n} = \begin{bmatrix} \mathbf{n}_{gw} \\ \mathbf{n}_{aw} \\ \mathbf{n}_g \\ \mathbf{n}_a \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{gw} & & & \\ & \mathbf{Q}_{aw} & & \\ & & \mathbf{Q}_g & \\ & & & \mathbf{Q}_a \end{bmatrix} \tag{4}$$

During measurement, the camera images the world points, $\mathbf{p}_{l_i}^W$ (which in our current system, we assume are known). To take the world points from the world coordinate system to the camera's coordinate system $\mathbf{p}_{l_i}^C$, we apply the rotation and orientation of the IMU in the world coordinate system, and those of the camera with respect to the IMU:

$$\mathbf{p}_{l_i}^C = (\mathbf{C}(\bar{q}_C^I))^T ((\mathbf{C}(\bar{q}_I^W))^T (\mathbf{p}_{l_i}^W - \mathbf{p}_I^W) - \mathbf{p}_C^I) \tag{5}$$

We model the projected points u_i and v_i in the camera's coordinate system with additive measurement noise.

$$\begin{aligned}
\mathbf{z}_i &= \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} x'_i \\ y'_i \end{bmatrix} + \eta_i \\
\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} &\equiv K \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}
\end{aligned} \tag{6}$$

The noise η is modeled as zero mean with a covariance matrix \mathbf{R} .

2.2 Unscented Kalman Filter equations

In the unscented Kalman filter, at every time step, a set of sigma points is estimated by jittering the state at that time step strategically along significant directions of the augmented covariance matrix. These significant directions correspond to eigenvectors of the augmented covariance matrix. The *a priori* predicted state is a weighted average of the sigma points. When a measurement comes in, the sigma points are fed to the measurement model to get an estimate of the measurement, and the estimate of the state and the covariance matrix is corrected by computing a Kalman gain matrix.

The augmented state and covariance updated from the time instant t_{k-1} is generated by tacking on the noise mean and covariance to the original state and covariance matrix:

$$\hat{\mathbf{x}}_a^+(t_{k-1}) = \begin{bmatrix} \hat{\mathbf{x}}^+(t_{k-1}) \\ \mathbf{0}_{12 \times 1} \end{bmatrix}, \mathbf{P}_a^+(t_{k-1}) = \begin{bmatrix} \mathbf{P}^+(t_{k-1}) & \\ & \mathbf{Q} \end{bmatrix} \tag{7}$$

The N sigma points consist of the augmented state at t_{k-1} and the state jittered positively and negatively along the columns of the matrix \mathbf{S} found by the Cholesky decomposition of the augmented covariance matrix.

$$\begin{aligned}\mathbf{S} &= \sqrt{(\lambda + N)\mathbf{P}_a^+(t_{k-1})} \\ {}^0\chi_a(t_{k-1}) &= \hat{\mathbf{x}}_a^+(t_{k-1}) \\ {}^l\chi_a(t_{k-1}) &= \hat{\mathbf{x}}_a^+(t_{k-1}) + {}^l\mathbf{S}(t_{k-1}), l = 1, \dots, N \\ {}^{2N+l}\chi_a(t_{k-1}) &= \hat{\mathbf{x}}_a^+(t_{k-1}) - {}^l\mathbf{S}(t_{k-1}), l = 1, \dots, N\end{aligned}\tag{8}$$

Here λ is a constant specified as $\lambda = \alpha^2(N + \beta) - N$, α is a small number (we use $\alpha = .1$), and $\beta = 2$. Weights are computed for the sigma points as:

$$\begin{aligned}{}^0W_m &= \frac{\lambda}{\lambda + N} \\ {}^0W_c &= \frac{\lambda}{\lambda + N} + (1 - \alpha^2 + \beta) \\ {}^jW_m = {}^jW_c &= \frac{1}{2(\lambda + N)}\end{aligned}\tag{9}$$

The sigma points are propagated through the process model, and the *a priori* estimate of the state and covariance matrix is calculated using the propagated sigma points and the weights.

$$\begin{aligned}{}^i\chi_a(t_k) &= \mathbf{f}({}^i\chi_a(t_{k-1})), i = 0, \dots, 2N \\ \hat{\mathbf{x}}^-(t_k) &= \sum_{i=0}^{2N} {}^iW_m {}^i\chi(t_k) \\ \mathbf{P}^-(t_k) &= \sum_{i=0}^{2N} {}^iW_c ({}^i\chi(t_k) - \hat{\mathbf{x}}^-(t_k)) ({}^i\chi(t_k) - \hat{\mathbf{x}}^-(t_k))^T\end{aligned}\tag{10}$$

When measurement arrives from the camera, the sigma points are propagated through the measurement model to get an estimate of the measurement $\hat{\mathbf{z}}(t_k)$

$$\begin{aligned}{}^i\gamma(t_k) &= \mathbf{h}({}^i\chi(t_k)), i = 0, \dots, 2N \\ \hat{\mathbf{z}}(t_k) &= \sum_{i=0}^{2N} {}^iW_m {}^i\gamma(t_k)\end{aligned}\tag{11}$$

The Kalman gain matrix is computed and used to compute the *a posteriori* state and covariance matrix.

$$\begin{aligned}\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{z}}}(t_k) &= \sum_{i=0}^{2N} {}^iW_c ({}^i\chi(t_k) - \hat{\mathbf{x}}^-(t_k)) ({}^i\gamma(t_k) - \hat{\mathbf{z}}(t_k))^T \\ \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k) &= \sum_{i=0}^{2N} {}^iW_c ({}^i\gamma(t_k) - \hat{\mathbf{z}}(t_k)) ({}^i\gamma(t_k) - \hat{\mathbf{z}}(t_k))^T \\ \mathbf{K}(t_k) &= \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{z}}}(t_k) (\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k) + \mathbf{R})^{-1} \\ \hat{\mathbf{x}}^+(t_k) &= \hat{\mathbf{x}}^-(t_k) + \mathbf{K}(t_k) (\mathbf{z}(t_k) - \hat{\mathbf{z}}(t_k)) \\ \mathbf{P}^+(t_k) &= \mathbf{P}^-(t_k) - \mathbf{K}(t_k) \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k) (\mathbf{K}(t_k))^T\end{aligned}\tag{12}$$

Note: As described in Kelly and Sukhatme [1], the unscented Kalman filter updates the state by computing the barycentric mean of the sigma points. In general the barycentric mean of unit quaternions \bar{q}_I^W and \bar{q}_C^I will not lie on the unit sphere, and we require a different parametrization for their update. We will use the modified Rodrigues parameters (MRPs) of the error quaternions used in [1]. The modified Rodrigues parameter $\delta\mathbf{e}$ is given as:

$$\delta\mathbf{e} = \frac{\delta\mathbf{q}}{1 + \delta q_0},\tag{13}$$

where $\delta \mathbf{q}$ and δq_0 are each the vector and scalar portion of the error quaternion. At time step t_k , we start off with a mean error quaternion of ${}^0\delta \hat{\mathbf{e}}_I^W(t_{k-1}) = \mathbf{0}_{3 \times 1}$ and ${}^0\delta \hat{\mathbf{e}}_C^I = \mathbf{0}_{3 \times 1}$. The MRP part of the sigma points is computed ${}^j\hat{\mathbf{e}}_I^W(t_{k-1})$ in terms of these mean MRPs using 9, and corresponding quaternions are propagated as

$${}^j\hat{q}_I^W(t_{k-1}) = {}^j\delta \hat{q}_I^W(t_{k-1}) \otimes \hat{q}^{W+}(t_{k-1}) \quad (14)$$

The sigma points are passed through the process model, and before computing the barycentric mean, the mean quaternion is divided out to get the error quaternion at t_k .

$${}^j\delta \hat{q}_I^W(t_k) = {}^j\hat{q}_I^W(t_k) \otimes \left({}^0\hat{q}_I^W(t_k) \right)^{-1} \quad (15)$$

The process of multiplying error vectors to get unit quaternions is repeated before passing the sigma points to the measurement model, and they are re-computed before obtaining the Kalman gain matrix. For the IMU-camera orientation \bar{q}_C^I , the error quaternions are used only when a measurement comes in.

3 Tests

We decided to implement the filter progressively given the complexity of the state space. We considered a reduced number of unknowns and a very simple model at the beginning. The following subsections detail our progress towards a complete toolbox for sensor-to-sensor calibration.

3.1 Estimating rigid body translation

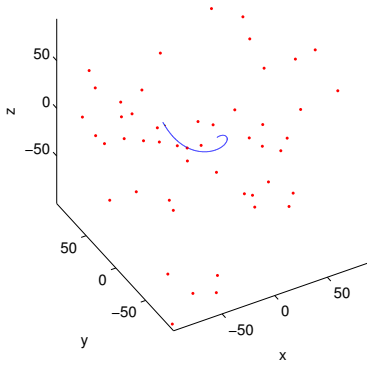


Figure 1: Automatically generated landmarks (red points) in our simulated world. The blue line depicts the uniform velocity path used for the rigid body translation test.

Our first filter implementation consists of estimating the position in 3D space of the rigid body sensor suite. We assume that we have a "world velocity sensor," which provided noisy estimates of the system's velocity in the world frame. No bias or gravity terms needed to be considered in this model.

We generated a ground truth path for the sensor suite using a simple motion simulator which generates motions from a vector of constant angular velocity and constant acceleration. These motions determine the ground truth for world velocity, position, and orientation of the sensors at every point in time. We corrupt the ground truth velocity and provide it to the filter as a sensor measurement at the correction step. Our state vector has a dimensionality of 3 in this test, instead of 26 as in (1). The process noise covariance is a 3×3 diagonal matrix.

$$\mathbf{x}_t = [p_{I_x}^W(t) \quad p_{I_y}^W(t) \quad p_{I_z}^W(t)] \quad Q = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \quad (16)$$

We automatically generate landmarks by uniformly scattering 3D points in our simulated world, and projected them into our simulated camera sensor. Figure 1 depicts a possible distribution of landmarks in our simulated world. When landmarks are projected into the camera frame and measured as pixel coordinates, we add independent Gaussian noise to the pixel coordinates to simulate the real-world behavior of cameras.

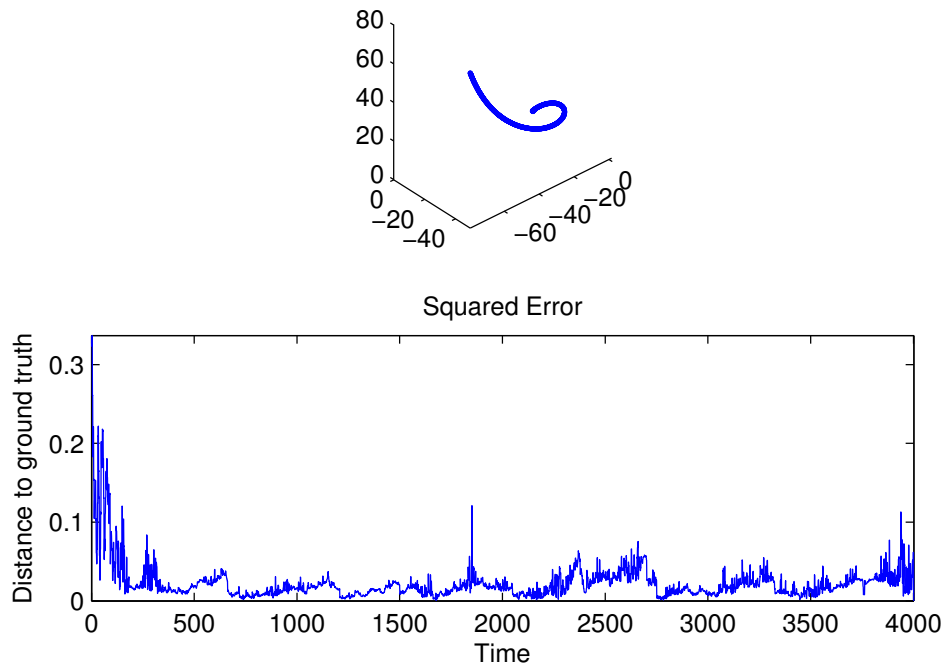


Figure 2: Estimated 3D world position of the camera and IMU from the UKF and squared error distance between estimated position and ground truth

Fig. 2 shows the 3D position estimate for the IMU in the world frame from the unscented Kalman filter. We also plot the squared distance between the ground truth and the estimated position at each step in time. There is a large initial error because we initialize the UKF with an error of 0.5 meters in each axis. Even with this large initial error, the filter is able to converge and track the true position.

4 Future work

The next step in the project will be to expand our filter to estimate the full state vector. The most difficult part of this implementation will be the quaternion prediction and correction steps within the Kalman filter. We are also developing a more complicated motion simulator to simulate the motion and field of view for a realistic camera-IMU system.

References

- [1] Jonathan Kelly and Gaurav Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *Int. J. Rob. Res.*, 30:56–79, January 2011.