Report: Camera-Radar fusion using Kalman Filter

Anirudh R Iyer

January 2024

1 Problem Statement

Combining information from RADAR and Camera to gain rich information for perception and tracking semantic objects in an autonomous driving scenario.

2 Introduction

The perception step of autonomous driving systems requires rich information, which in most cases is not attainable through a single source sensor. Hence, multiple sensors like: RADAR, LiDAR and Camera are used to get different information. Each of these sensors have their advantages and disadvantages over each other: Camera is information rich for semantic segmentation but doesn't have depth while RADAR and LiDAR are sparse in nature hence detection is difficult. Though using multiple sensors gives us important information, in order to use the data we need to combine them together.

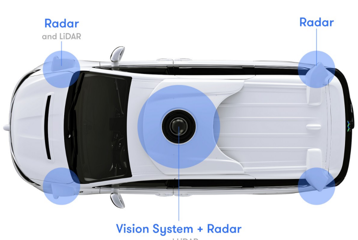


Figure 1: Sensor setup of Waymo with multiple sensors

Sensor fusion is essential to combine information gained from different modalities to form information rich data. In this case, our goal is to combine the coordinates from Camera and RADAR. Data from RGB-Camera is typically 2-dimensional, it provides width and height coordinates while RADAR provides width and depth coordinates. Effectively combining data from these sensors would result in information rich data with: width, height and depth.

In order for fusing data effectively, we propose using Kalman Filter: a popular method used for sensor fusion. Kalman Filter[3] makes use of intuitive physical model formulae and statistics in order to predict the next state using information about the previous state and updates the prediction using actual measurements to fine-tune the error covariance.

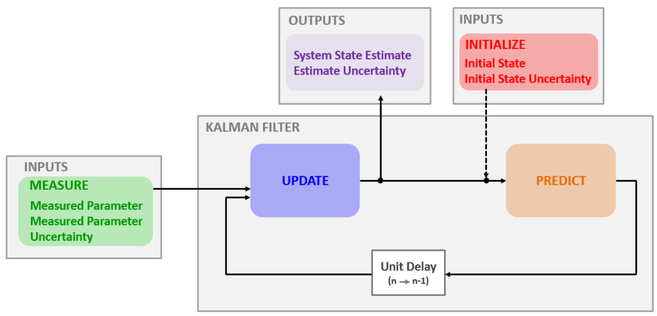


Figure 2: Kalman Filter schematic diagram

3 Mathematical Formulation

We are assuming near constant velocity Kalman Filter (NCVKF), for which we estimate the process noise covariance to give more accurate state estimates.[1]

3.1 Notation

$$\begin{bmatrix} P_x \\ P'_y \\ P_z \end{bmatrix} = R^{-1}K^{-1} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}$$

P_x: Real world x-coordinate width direction, P'_y: Scaled y-coordinate depth direction, P_z: Real world z-coordinate height direction, R: extrinsic camera calibration matrix, K: intrinsic camera calibration matrix

$$x_n = \begin{bmatrix} P_x \\ P_y \\ P_z \\ V_x \\ V_y \end{bmatrix}$$

x_n: Kalman state vector at timestep-n, $x_{n,n-1}$: Kalman prediction for state at time n given state at timestep n-1 P_y: Real world y-coordinate, V_x: velocity of obejct x-coordinate, V_y: velocity of object y-coordinate

Note: *Abuse of notation, Kalman Filter conventional variable for state estimate error covariance is $P_{n,n-1}$, not to be confused with position variables P_x, P_y, P_z*

3.2 Prediction step

Near constant velocity Kalman filter model, prediction step predicts next state given: current state, state transition matrix (F) and process noise covariance (Q) which accounts for minor acceleration/deceleration by the object. F is based on kinematics model:

$$s = u + v * \delta t$$

$$x_{n,n-1} = Fx_{n-1}$$

$$P_{n,n-1} = F.P_{n-1}.F^T + Q$$

$$F = \begin{bmatrix} 1 & 0 & 0 & \delta t & 0 \\ 0 & 1 & 0 & 0 & \delta t \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The following steps are used to estimate process noise matrix (Q) using kinematics model.

Noise can be represented as vector:

$$\gamma_{noise} = \begin{bmatrix} \gamma_{P_x} \\ \gamma_{P_y} \\ \gamma_{P_z} \\ \gamma_{V_x} \\ \gamma_{V_y} \end{bmatrix}$$

The prediction step can be broken down as follows using the kinematics model for displacement using velocity:

$$P'_x = P_x + V_x * \delta t + \gamma_{P_x}$$

$$P'_y = P_y + V_y * \delta t + \gamma_{P_y}$$

$$P'_z = P_z + \gamma_{P_z}$$

$$V'_x = V_x + \gamma_{V_x}$$

$$V'_y = V_y + \gamma_{V_y}$$

Noise variables can be estimated using kinematics rules wrt acceleration, for which we are assuming acceleration is a random variable with normal distribution:

$$a_x = N(\mu = 0, \sigma = 1)$$

$$a_y = N(\mu = 0, \sigma = 1)$$

$$a_y = N(\mu = 0, \sigma = 1)$$

$$\gamma_{P_x} = 1/2 * a_x \delta t^2$$

$$\gamma_{P_y} = 1/2 * a_y \delta t^2$$

$$\gamma_{P_z} = 1/2 * a_z \delta t^2$$

$$\gamma_{V_x} = a_x \delta t$$

$$\gamma_{V_y} = a_x \delta t$$

Noise vector γ_{noise} can be represented as matrix multiplication of acceleration as:

$$\gamma_{noise} = G.a$$

$$\gamma_{noise} = \begin{bmatrix} 1/2 * \delta t^2 & 0 & 0 \\ 0 & 1/2 * \delta t^2 & 0 \\ 0 & 0 & 1/2 * \delta t^2 \\ \delta t & 0 & 0 \\ 0 & \delta t & 0 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

Q is the covariance of the noise vector, hence Q can be represented as:

$$Q = E[\gamma_{noise} \gamma_{noise}^T]$$

$$Q = E[G.a(G.a)^T]$$

$$Q = E[G.a.a^T.G^T]$$

$$Q = G.E[a.a^T].G^T$$

Assuming acceleration random variables a_x, a_y, a_z are independent of each other.

$$E[a.a^T] = \begin{bmatrix} \sigma_{a_x}^2 & 0 & 0 \\ 0 & \sigma_{a_y}^2 & 0 \\ 0 & 0 & \sigma_{a_z}^2 \end{bmatrix}$$

$$Q = G.E[a.a^T].G^T = \begin{bmatrix} \sigma_{a_x}^2 \delta t^4/4 & 0 & 0 & \sigma_{a_x}^2 \delta t^3/2 & 0 \\ 0 & \sigma_{a_y}^2 \delta t^4/4 & 0 & 0 & \sigma_{a_y}^2 \delta t^3/2 \\ 0 & 0 & \sigma_{a_z}^2 \delta t^4/4 & 0 & 0 \\ \sigma_{a_x}^2 \delta t^3/2 & 0 & 0 & \sigma_{a_x}^2 \delta t^2 & 0 \\ 0 & \sigma_{a_y}^2 \delta t^3/2 & 0 & 0 & \sigma_{a_y}^2 \delta t^2 \end{bmatrix}$$

For now we are assuming that $\sigma_{a_x} = \sigma_{a_y} = \sigma_{a_z} = \sigma_a$ and we are fine-tuning σ_a using minimization algorithm wrt average MSE error of distances + MSE error of velocity predictions.

3.3 Measurement Step

Kalman Filter has 3 steps, the measurement step is used to compare the predicted and measured values which will be later used in the Update Step. The measurement prediction vector $z_{n,n-1}$ is the entire state vector(x) in our use case is identity matrix of size (5,5).

$$z_{n,n-1} = H.x_{n,n-1}$$

Innovation (Measurement Residual) is the difference between actual measurement z_k and $z_{k,k-1}$.

$$y_n = z_n - z_{n,n-1}$$

Innovation covariance, where R_n is the measurement error covariance, which in the case of using ground truth points is taken to be a zero matrix.

$$S_n = H.P_{n,n-1}.H^T + R_n$$

3.4 Update Step

Kalman Filter's final step which updates the predicted state estimation and error covariance in order to get better estimates for the next time steps.

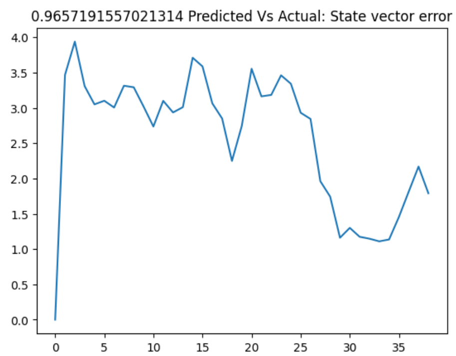
Kalman gain: $K_n = P_{n,n-1}.H^T.S_n^{-1}$

State update: $x_{n,n} = x_{n,n-1} + K_n.y_n$

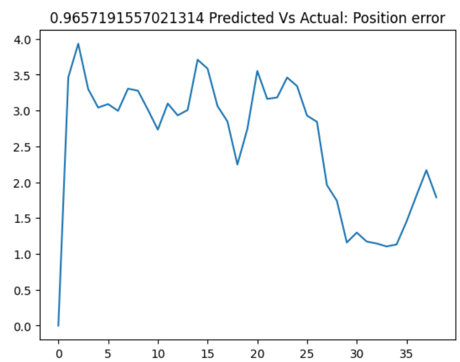
Covariance update: $P_{n,n} = (I - K_n.H).P_{n,n-1}$

4 Results

The results below are from Nuscenes[2] mini dataset using [x,y,z,vx,vy] from ground truth wrt ego coordinate system, scene 0. We have used minimization over area under velocity + sum of position error across timesteps to optimize acceleration variance a (0,1.5]. The following graphs show average MSE of state vector, position and velocity of all objects at individual timesteps for acceleration variance a = 0.9657.



(a) Average MSE state vector



(b) Average MSE position

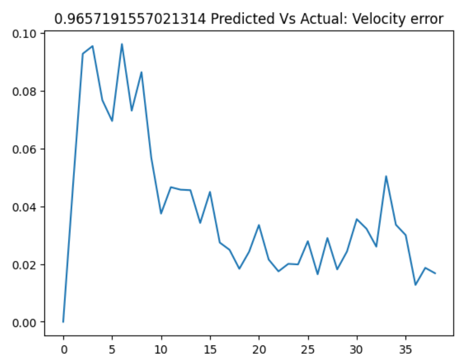


Figure 4: Average MSE velocity

It can be seen that after 40 time-steps, the error for velocity estimates is significantly low as we have implemented a near constant velocity model, it can also be observed that the average position error goes up to 4m.

5 Conclusion and Future work

Our goal is to combine information from Camera[x,z] and Radar[x,y,vx,vy] and track semantic objects. We applied Kalman Filter using nuScenes ground truth annotation for tracking semantic objects in a scene by getting [x,y,z,vx,vy] from ground truth. Through the results it can be seen that the Kalman Filter with near constant velocity assumption does reduce the average MSE error over time.

As a part for future work, more data could give us a better understanding of the results from Kalman Filter. We could try fusing RADAR and Camera data from CARLA as a starter for the same. Once we get an understanding of the use of Kalman Filter for object tracking through fused data, we could try using Extended or Unscented Kalman Filters as well. Once the proof of concept works, we could even try fusing multiple RADAR and Cameras together to get information wrt one singular coordinate frame for better perception.

References

[1] Basso, G., and Amorim, T. Kalman filter with dynamical setting of optimal process noise. *IEEE Access* (2017).

[2] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027* (2019).

[3] Kalman, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering* (1960).