Homework 2

AUE 8240: Autonomous Driving Technologies

Student:

Adithya Suresh

C18590622

Problem 1:

Question 1: Write the complete camera model, i.e., transformation from world frame coordinates to pixel frame coordinates. What are the meanings of all intrinsic and extrinsic parameters respectively?

Answer: [1]

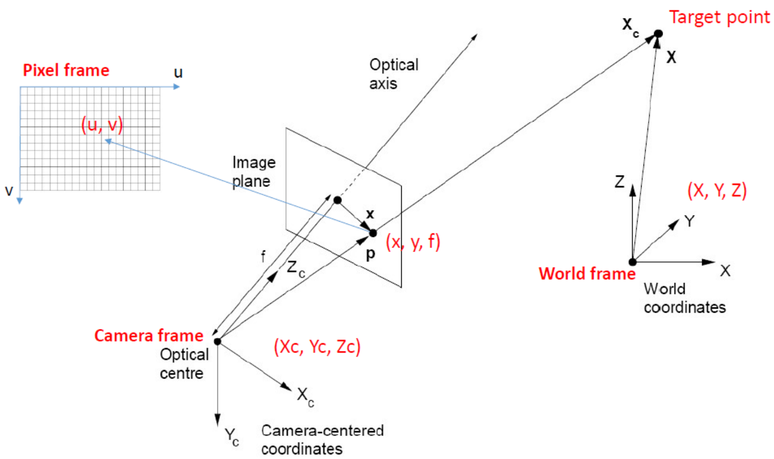

Figure 1 Camera Model

Here the parameters include the World frame coordinates – (X ,Y ,Z)
Camera frame coordinates – (X_c, Y_c, Z_c)
Image plane coordinates – (x, y, f)
Pixel frame coordinates – (u, v)

The transformation from World frame coordinates to pixel frame coordinates will happen in three steps, which are:

1. World frame coordinates to Camera frame coordinates.
2. Camera frame coordinates to image plane coordinates.
3. Image plane coordinates to pixel frame coordinates.

Step 1: World frame coordinates to Camera frame coordinates transformation:

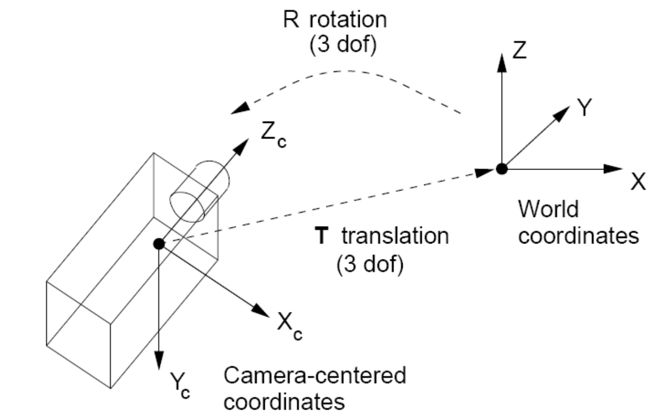

Figure 2 World to Camera frame transformation

Transformation in coordinate points will be from (X, Y, Z) to (X_c, Y_c, Z_c).

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

$$\mathbf{X}_c = \mathrm{R}\mathbf{X} + \mathbf{T}$$

This can be written as,

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Here,

$$\tilde{\mathbf{X}}_c = \mathrm{P}_r \tilde{\mathbf{X}}\ , \text{ where } \mathrm{P}_r = \left[\begin{array}{ccc|c} & \mathrm{R} & & \mathbf{T} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

But the rotation matrix, R in 3D can be written as follows,

$$R = R_z(\alpha)\, R_y(\beta)\, R_x(\gamma) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$

This implies,

$$R = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}$$

The Rotation matrix represents the rotation of camera along α, β and γ angles about the axis.
The translation matrix represents,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

<u>Step 2: Camera frame coordinates to image plane coordinates transformation:</u>

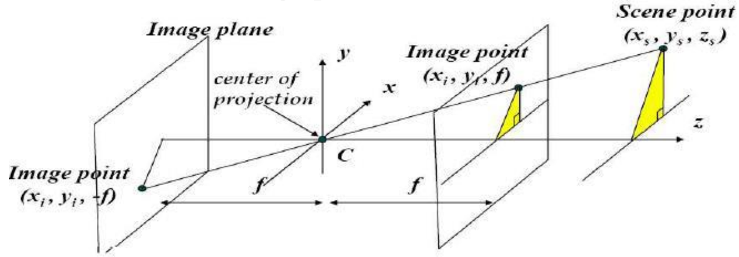

Figure 3 Pin hole Camera model

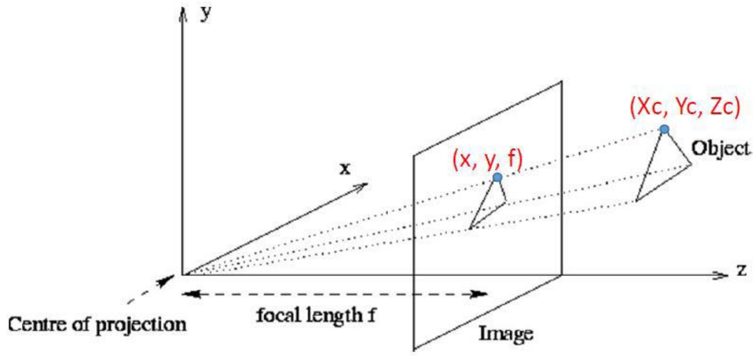

Figure 4 Image plane with respect to object and projection point

Transformation in coordinate points will be from (X_c, Y_c, Z_c) to (x, y).
Using the Triangular similarity concept,

$$\frac{x}{X_c} = \frac{y}{Y_c} = \frac{z}{Z_c}$$

Then the values for x and y can be calculated as,

$$x = f * \frac{X_c}{Z_c} \text{ and } y = f * \frac{Y_c}{Z_c}$$

Here f is the focal length between image plane and the center of projection.
The matrix can be substituted as,

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \end{bmatrix}$$

<u>Step 3: Image plane coordinates to pixel frame coordinates transformation:</u>

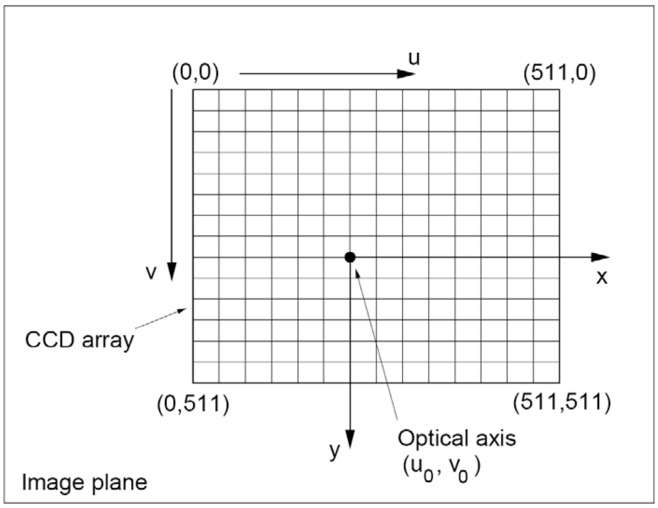

Figure 5 Pixel Frame

Transformation in coordinate points will be from (x, y) to (u, v).
The equation in pixel coordinates can be represented by,

$$u = u_0 + k_u x \text{ and } v = v_0 + k_v y$$

Here, k_u is the number of one unit length of u axis,
k_v is the number of one unit length of v axis.

The matrix representation is as follows,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transformation from World Coordinates to Pixel coordinates in matrix representation is as:

Step 1

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Step 2

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \end{bmatrix}$$

Step 3

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} su \\ sv \\ s1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Intrinsic parameters (5) Extrinsic parameters (6)

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

The rotation matrix encircled in green marker above consists of 9 parameters which are represented by three variables α, β and γ as shown in Step 1.

<u>Extrinsic parameters definition:</u>

Here, T_x, T_y, T_z, α, β and γ are the 6 extrinsic parameters.
T_x : Translation in x axis
T_y : Translation in x axis
T_z : Translation in x axis
α : Euler angle about x axis of world frame coordinate system
β : Euler angle about y axis of world frame coordinate system
γ: Euler angle about z axis of world frame coordinate system

Intrinsic parameters definition:

k_u , k_v, u_0, v_0, f are the 5 intrinsic parameters.
k_u : No. of pixels on one unit length of x axis
k_v : No. of pixels on one unit length of y axis
(u_0 , v_0): Optical axis center in pixel frame
f : focal length of camera lens

Question 2: A camera is mounted in the front of a vehicle and all its intrinsic parameters are assumed to be known. Assume there is a front vehicle and its measured vehicle width and same-depth lane width in the camera's pixel frame are d and l respectively. Assume the actual lane width is L. Calculate the distance between the camera and front vehicle.

Answer: [2]

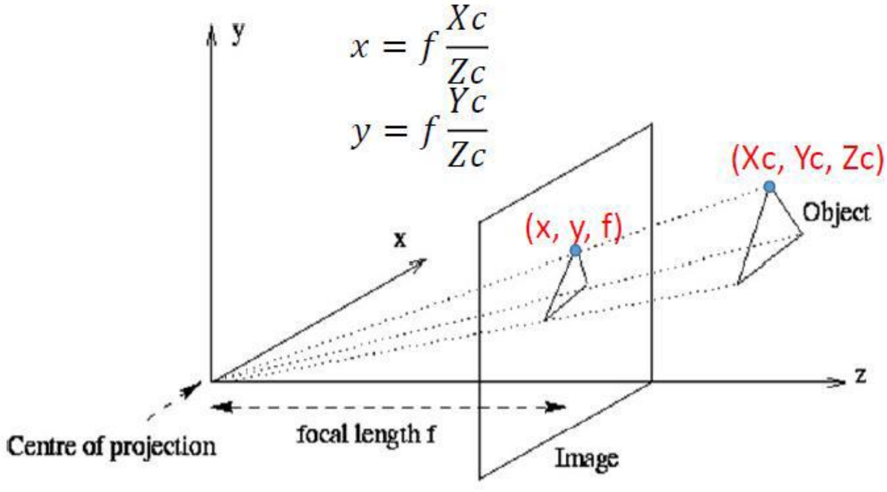

Figure 6 Image plane with respect to object

From Single image to coordinates we can calculate the distance between camera and front vehicle.

Given data:

Vehicle width = d (Pixel frame)
Lane width = l (Pixel frame)

From this, we can calculate the Actual lane width,

$$\frac{x}{X_c} = \frac{f}{Z_c}$$

But x is the distance along x axis in pixel frame.
X_c is the distance along x axis in World frame.
Z_c is the depth from the object to camera.

f is the focal length of camera.

$$\frac{\frac{l}{k_u}}{L} = \frac{f}{Z_c}$$

Where,
K_u is the number of pixels on one unit length along x axis.

Rearranging the above equation,

$$Z_c = \frac{f * L}{\frac{l}{k_u}}$$

The above Z_c will give the distance between camera and front vehicle.

Question 3: Describe the process of KF, EKF and PF and their applicable conditions respectively.

Answer:

Kalman Filter (KF): [3]

Kalman Filter is an algorithm that uses a series of sensing signals observed over time, containing statistical noises and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone. It is also a common algorithm used for sensor fusion and data fusion.

Process Overview:

Make prediction based on previous data : $\hat{y}^-$, σ^-

↓

Take measurement – z_k, σ_z

↓

Optimal estimate ($\hat{y}$) = Prediction + (Kalman Gain) * (Measurement - Prediction)

Variance of estimate = Variance of prediction * (1 – Kalman Gain)

Process Model:

$$y_k = Ay_{k-1} + Bu_k + w_k$$

$$z_k = Hy_k + v_k$$

Here,

k is the discrete time.
y is the system state to be estimated.
u is the system input.
A, B, H are the constant matrices.
z is the system measurement.
w is the process noise, w ~ N (0, Q).
v is the measurement noise v ~ N (0, R).

Filter process:

Prediction step:

$\hat{y}_k{}^-$ is predicted based on measurements at previous time-step.

Predicted state estimate (to predict the state ahead): $\hat{y}_k^- = Ay_{k-1} + Bu_k$
Predicted error covariance: $P_k^- = AP_{k-1}A^T + Q$

Here, u_k is the control signal input.
z_k is the input measurement.

Correction step:

$\hat{y}_k$ (output) is estimated by correcting $\hat{y}_k^-$ based on new measurement.

Finding the Kalman Gain: $K = P_k^- H^T (HP_k^- H^T + R)^{-1}$

Updating the state estimate with z_k: $\hat{y}_k = \hat{y}_k{}^- + K(z_k - H\hat{y}_k{}^-)$

Updating the error covariance: $P_k = (I - KH)P_k^-$

Applicable conditions:

Kalman filter can be applied when the system is linear, and noise is gaussian.

Extended Kalman Filter (EKF):

The Extended Kalman Filter (EKF) is the nonlinear version of the Kalman filter which linearizes about an estimate of the current mean and covariance. In the Extended Kalman Filter, the state transition and observation models do not need to be linear functions of the state but may instead be differentiable functions.

Process Model:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \boldsymbol{w}_k$$
$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k$$

Filter process:

Prediction step:
Predicting the state ahead using predicted state estimate,

$$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_k)$$

Predicting the error covariance ahead,

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_k^\top + \boldsymbol{Q}_k$$

w_k is process noise and v_k is observation noise. Both assumed to be Gaussian noises with covariance Q_k and R_k respectively.
The function f can be used to compute the predicted state from the previous estimate and similarly the function h can be used to compute the predicted measurement from the predicted state. However, f and h cannot be applied to the covariance directly rather a matrix of partial derivatives (the Jacobian) is computed.

$$\boldsymbol{F}_k = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_k}$$

$$\boldsymbol{H}_k = \left.\frac{\partial h}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k|k-1}}$$

At each time step, the Jacobian is evaluated with current predicted states. These matrices can be used in the Kalman filter equations. This process linearizes the non-linear function around the current estimate.

Correction step:

Innovation or measurement residual,

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_{k|k-1})$$

Innovation (or residual) covariance,

$$\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^\top + \boldsymbol{R}_k$$

Near-optimal Kalman Gain,

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^\top \boldsymbol{S}_k^{-1}$$

Updated state estimate,

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \tilde{\boldsymbol{y}}_k$$

Updated covariance estimate,

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1}$$

Applicable conditions:

Extended Kalman filter can be applied when the system is non-linear. However, the noise must be gaussian. Extended Kalman Filter will not work with highly non-linear relationships and non-Gaussian noise.

Particle Filter (PF):

Particle filtering is a Bayesian based filter that uses a set of particles (also called samples) to represent the posterior distribution of some stochastic process given noisy and/or partial observations. The state-space model can be non-linear, and the initial state and noise distributions can take any form required. Particle filter techniques provide a well-established methodology for generating samples from the required distribution without requiring assumptions about the state-space model or the state distributions. However, these methods do not perform well when applied to very high-dimensional systems.

Algorithm:

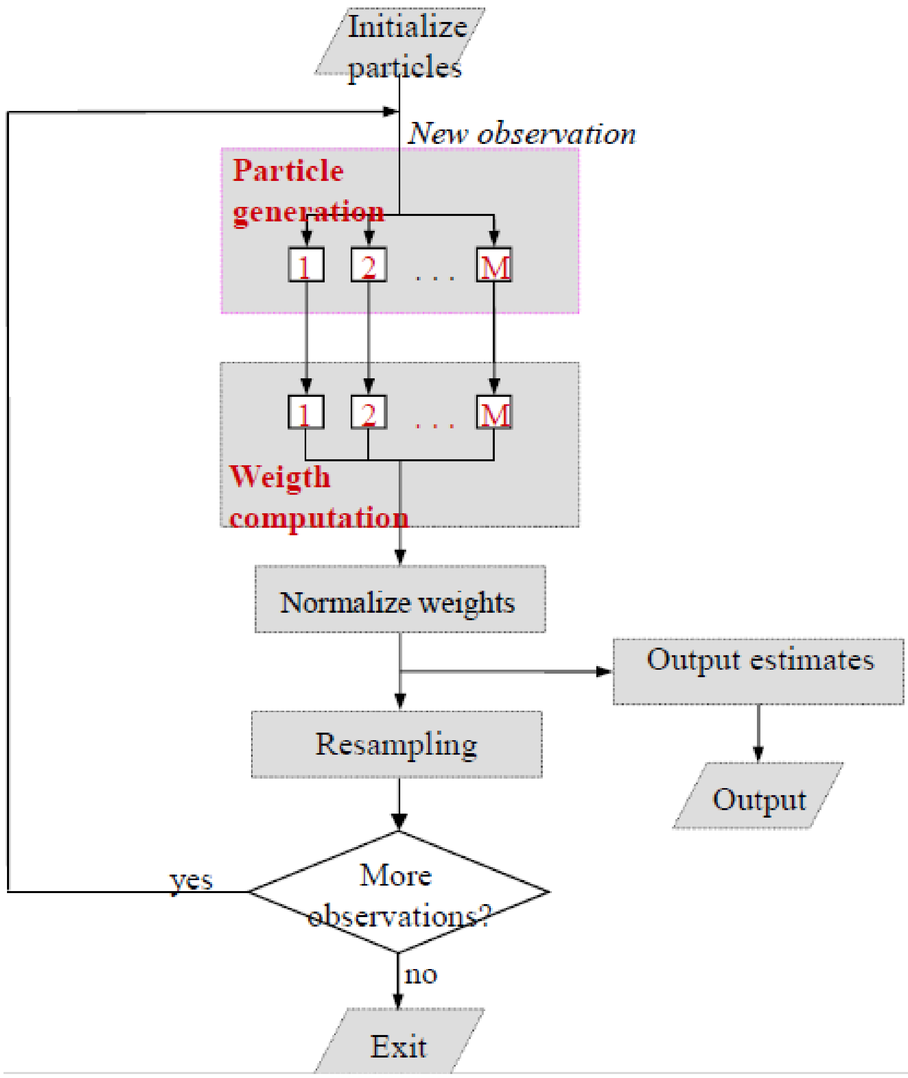

Figure 7 Particle Filter Algorithm

Process Model:

States: x_k

Observations: z_k
State equation: $x_k=f(x_k-1, u_k)$ or $p(x_k / x_k-1)$
Observation equation: $z_k=g(x_k, v_k)$ or $p(z_k / x_k)$

Filter process:

Prediction step:

Step 1: Particle generation:

$$x_k^{(m)} \sim p(x_k \mid x_{k-1})$$

Step 2: Weight computation:

$$w_k^{*(m)} = w_{k-1}^{*(m)} p(z_k \mid x_k^{(m)})$$

Step 3: Weight normalization:

$$w_k^{(m)} = \frac{w_k^{*(m)}}{\sum_{m=1}^{M} w_k^{*(m)}}$$

Step 4: Estimate computation:

$$E(g(x_k \mid z_{1:k})) = \sum_{m=1}^{M} g(x_k^{(m)}) w_k^{(m)}$$

Step 5: Resampling:

$$\left\{ \tilde{\mathbf{x}}_k^{(m)}, \frac{1}{M} \right\}_{m=1}^{M} \sim \left\{ \mathbf{x}_k^{(m)}, w_k^{(m)} \right\}_{m=1}^{M}$$

Applicable conditions:
Particle filter can be applied when the system is non-linear, and the noise is non-gaussian.

Problem 2: An inertial measurement unit (IMU) can report angular angles (roll ϕ, pitch θ, yaw ψ) using a combination of accelerometer, magnetometer and gyroscope. The accelerometer can measure roll ϕ and pitch θ with noises based on acceleration direction, the magnetometer can measure yaw ψ with noise, and the gyroscope can measure angular rates ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$) with noises and potential drifts. Design a multi-sensor fusion approach to fuse these three noisy sensors to calculate accurate angular angles: roll ϕ, pitch θ, yaw ψ.

Question 1: What sensor(s) should be used for prediction and what sensor(s) should be used for correction? What sensor fusion approach should be applied?
Answer:

Gyroscope can be used for prediction of roll pitch and yaw.
Accelerometer can be used for correction of roll and pitch.
Magnetometer can be used for correction of yaw.
Kalman filter (parallel approach) can be used to apply sensor fusion of the three sensors.

Question 2: Define the state to be estimate as a vector y=[ϕ,θ,ψ]T, and write the prediction process in matrix form.

Answer:

Given data:

u_k is given by gyroscope, $v_k = 0$ (measurement noise) and $w_k = 0$ (mean gaussian noise).
Kalman filter process model can be represented as,

$$y_k = Ay_{k-1} + Bu_k + w_k$$

$$z_k = Hy_k + v_k$$

Given that, the Roll is ϕ, Pitch is θ, Yaw is ψ

We can represent in process model form, $\phi_k = \phi_{k-1} + T\phi$

The accelerometer can measure the Roll ϕ, Pitch θ. This can be used for correction step in filtering. The magnetometer can measure yaw ψ. Also, the gyroscope can measure angular rates ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$). This can be used for prediction step in filtering.

Expanding the process model in matrix representation,

$$\begin{bmatrix} \phi_k \\ \theta_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi_{k-1} \\ \theta_{k-1} \\ \psi_{k-1} \end{bmatrix} + \begin{bmatrix} T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} \begin{bmatrix} \dot{\phi}_k \\ \dot{\theta}_k \\ \dot{\psi}_k \end{bmatrix} + w_k$$

$$y_k = Ay_{k-1} + Bu_k + w_k$$

$$\begin{bmatrix} \phi_k \\ \theta_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi_k \\ \theta_k \\ \psi_k \end{bmatrix} + v_k$$

$$z_k = Hy_k + v_k$$

The A, B and H matrices can be found from the above equations.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

<u>Kalman Filter Prediction:</u>

Predicted state estimate (to predict the state ahead):

$$\hat{y}_k^- = Ay_{k-1} + Bu_k$$

Here, $\hat{y}_k{}^-$ is predicted based on measurements at previous time-step.

$$\begin{bmatrix} \widehat{\phi}_k^- \\ \hat{\theta}_k^- \\ \widehat{\psi}_k^- \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi_{k-1} \\ \theta_{k-1} \\ \psi_{k-1} \end{bmatrix} + \begin{bmatrix} T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} \begin{bmatrix} \dot{\phi}_k \\ \dot{\theta}_k \\ \dot{\psi}_k \end{bmatrix}$$

Predicted error covariance: $P_k^- = AP_{k-1}A^T + Q$

$$\begin{bmatrix} P_{k\phi}^- & 0 & 0 \\ 0 & P_{k\theta}^- & 0 \\ 0 & 0 & P_{k\psi}^- \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} Q_{\dot{\phi}} & 0 & 0 \\ 0 & Q_{\dot{\theta}} & 0 \\ 0 & 0 & Q_{\dot{\psi}} \end{bmatrix}$$

Here the P_{11}, P_{22}, P_{33} are the variance of roll, pitch and yaw respectively and $Q_{\dot{\phi}}, Q_{\dot{\theta}}, Q_{\dot{\psi}}$ are the process noise.

<u>Question3:</u> Write the correction process in matrix form.

<u>Answer:</u>

$\hat{y}_k$ (output) is estimated by correcting $\hat{y}_k^-$ based on new measurement.

Finding the Kalman Gain: $K = P_k^- H^T (HP_k^- H^T + R)^{-1}$

The matrix representation of Kalman Gain is,

$$\begin{bmatrix} K_{11} & 0 & 0 \\ 0 & K_{22} & 0 \\ 0 & 0 & K_{33} \end{bmatrix} = \begin{bmatrix} P_{k\phi}^- & 0 & 0 \\ 0 & P_{k\theta}^- & 0 \\ 0 & 0 & P_{k\psi}^- \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{k\phi}^- & 0 & 0 \\ 0 & P_{k\theta}^- & 0 \\ 0 & 0 & P_{k\psi}^- \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} R_{\phi} & 0 & 0 \\ 0 & R_{\theta} & 0 \\ 0 & 0 & R_{\psi} \end{bmatrix} \right\}^{-1}$$

Here the K_{11}, K_{22}, K_{33} are the Kalman Gain of roll, pitch and yaw respectively.

Updating the state estimate: $\hat{y}_k = \hat{y}_k{}^{-} + K(z_k - H\hat{y}_k{}^{-})$

$$\begin{bmatrix}\hat{\Phi}_k\\ \hat{\theta}_k\\ \hat{\Psi}_k\end{bmatrix} = \begin{bmatrix}\hat{\Phi}_k^{-}\\ \hat{\theta}_k^{-}\\ \hat{\Psi}_k^{-}\end{bmatrix} + \begin{bmatrix}K_{11} & 0 & 0\\ 0 & K_{22} & 0\\ 0 & 0 & K_{33}\end{bmatrix}\left\{\begin{bmatrix}\Phi_k\\ \theta_k\\ \Psi_k\end{bmatrix} - \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\hat{\Phi}_k^{-}\\ \hat{\theta}_k^{-}\\ \hat{\Psi}_k^{-}\end{bmatrix}\right\}$$

Updating the error covariance: $P_k = (I - KH)P_k^{-}$

$$\begin{bmatrix}P_{k\Phi} & 0 & 0\\ 0 & P_{k\theta} & 0\\ 0 & 0 & P_{k\Psi}\end{bmatrix} = \left[1 - \begin{bmatrix}K_{11} & 0 & 0\\ 0 & K_{22} & 0\\ 0 & 0 & K_{33}\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\right]\begin{bmatrix}P_{k\Phi}^{-} & 0 & 0\\ 0 & P_{k\theta}^{-} & 0\\ 0 & 0 & P_{k\Psi}^{-}\end{bmatrix}$$

Problem 3:

Question 1: Use least square approach to find the 11 parameters for the left camera:

Question 2: Use least square approach to find the 11 parameters for the right camera:

Solution (Derivations and calculations): [4]

By Least square approach, the image frame to world frame coordinates conversion is performed.

$$\begin{bmatrix}su\\ sv\\ s1\end{bmatrix} = \begin{bmatrix}k_u & 0 & u_0\\ 0 & k_v & v_0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}f & 0 & 0\\ 0 & f & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & T_x\\ r_{21} & r_{22} & r_{23} & T_y\\ r_{31} & r_{32} & r_{33} & T_z\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}X\\ Y\\ Z\\ 1\end{bmatrix}$$

Intrinsic parameters (5) Extrinsic parameters (6)

The parameters which needs to be found, the intrinsic and extrinsic parameters are found together using the least square approach.

$$\begin{bmatrix}su\\ sv\\ s1\end{bmatrix} = \begin{bmatrix}c_{11} & c_{12} & c_{13} & c_{14}\\ c_{21} & c_{22} & c_{23} & c_{24}\\ c_{31} & c_{32} & c_{33} & 1\end{bmatrix}\begin{bmatrix}X\\ Y\\ Z\\ 1\end{bmatrix}$$

This general matrix represents the image pixel frame coordinates which can in turn link to the global or world coordinates. Representing the above matrix in the perspective of left and right cameras,

Left camera:

$$\begin{bmatrix} su_l \\ sv_l \\ s1 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Right camera:

$$\begin{bmatrix} su_r \\ sv_r \\ s1 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

From the above matrices, the 11 unknown parameters are left to be found for both left and right cameras. These 11 parameters can be found solving for the above matrix,

$$\begin{aligned} su_l &= b_{11}X + b_{12}Y + b_{13}Z + b_{14} \\ sv_l &= b_{21}X + b_{22}Y + b_{23}Z + b_{24} \\ s &= b_{31}X + b_{32}Y + b_{33}Z + 1 \end{aligned}$$

The third equation, s can be substituted in the first two equations and the result will be,

$$\begin{aligned} u_l &= (b_{11} - b_{31}u_l)X + (b_{12} - b_{32}u_l)Y + (b_{13} - b_{33}u_l)Z + b_{14} \\ v_l &= (b_{21} - b_{31}v_l)X + (b_{22} - b_{32}v_l)Y + (b_{23} - b_{33}v_l)Z + b_{24} \end{aligned}$$

Solving for the above equations and expanding,

$$\begin{aligned} u_l &= b_{11}X + b_{12}Y + b_{13}Z + b_{14} + b_{21}0 + b_{22}0 + b_{23}0 + b_{24}0 - b_{31}u_lX - b_{32}u_lY - b_{33}u_lZ \\ v_l &= b_{11}0 + b_{12}0 + b_{13}0 + b_{14}0 + b_{21}X + b_{22}Y + b_{23}Z + b_{24} - b_{31}v_lX - b_{32}v_lY - b_{33}v_lZ \end{aligned}$$

Representing the above equations in matrix format,

$$\underbrace{\begin{bmatrix} X, Y, Z, 1, 0, 0, 0, 0, -u_lX, -u_lY, -u_lZ \\ 0, 0, 0, 0, X, Y, Z, 1, -v_lX, -v_lY, -v_lZ \\ \vdots \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{14} \\ b_{21} \\ b_{22} \\ b_{23} \\ b_{24} \\ b_{31} \\ b_{32} \\ b_{33} \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} u_l \\ v_l \\ \vdots \end{bmatrix}}_{B}$$

The unknowns are the X matrix values for both left and right cameras, which can be found using the MATLAB command,

X = A\B or X = pinv(A) * B [General formula]

Answer 1:

11 parameters of the left camera are:

Table 1 Parameters 11 for left camera

Left camera parameters	Parameter values
b1	11.7261
b2	-1.2806
b3	-2.9809
b4	2396.1661
b5	-0.0853
b6	-0.0337
b7	-11.1668
b8	2.2259
b9	0.0017
b10	0.0023
b11	-0.0013

Answer 2:

11 parameters of the right camera are:

Table 2 Parameters 11 for right camera

Right camera parameters	Parameter values
c1	3.9421
c2	7.6183
c3	-2.3144
c4	1084.6857
c5	-0.0186
c6	-0.7463
c7	-7.8804
c8	1652.4684
c9	-0.0013
c10	0.0019
c11	-0.0010

Question 3: Calculate the coordinates (X, Y, Z) of other marked corners of the mold based on the two images:

Solution:

For solving this, the pixel frame coordinate values for both left and right cameras equations are used.

$$u_l = (b_{11} - b_{31}u_l)X + (b_{12} - b_{32}u_l)Y + (b_{13} - b_{33}u_l)Z + b_{14}$$
$$v_l = (b_{21} - b_{31}v_l)X + (b_{22} - b_{32}v_l)Y + (b_{23} - b_{33}v_l)Z + b_{24}$$

$$u_r = (c_{11} - c_{31}u_r)X + (c_{12} - c_{32}u_r)Y + (c_{13} - c_{33}u_r)Z + c_{14}$$
$$v_r = (c_{21} - c_{31}v_r)X + (c_{22} - c_{32}v_r)Y + (c_{23} - c_{33}v_r)Z + c_{24}$$

Matrix representation of the above equations will look like,

$$\underbrace{\begin{bmatrix} b_{11} - b_{31}u_l & b_{12} - b_{32}u_l & b_{13} - b_{33}u_l \\ b_{21} - b_{31}v_l & b_{22} - b_{32}v_l & b_{23} - b_{33}v_l \\ c_{11} - c_{31}u_r & c_{12} - c_{32}u_r & c_{13} - c_{33}u_r \\ c_{21} - c_{31}v_r & c_{22} - c_{32}v_r & c_{23} - c_{33}v_r \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} u_l - b_{14} \\ v_l - b_{24} \\ u_r - c_{14} \\ v_r - c_{24} \end{bmatrix}}_{B}$$

General equation can be written as,

X = pinv(A) * B

Answer:

Table 3 Coordinates of Unmarked points

Unmarked Coordinates	X coordinate (mm)	Y coordinate (mm)	Z coordinate (mm)
I	-0.2232	-0.1947	-10.1931
J	100.1302	-0.1336	-9.6743
K	19.9319	0.0722	-0.5652
L	20.2899	35.4880	-0.1869
M	79.9223	35.5724	-0.1638
N	80.3575	0.08089	-0.3152
O	75.2565	0.3209	-20.7574
P	24.8809	0.3786	-20.8883
Q	14.2292	0.0941	-10.2557
R	85.7376	0.2235	-9.8417

Question 4: Calculate the dimensions of the bar (length, width, height) beside the mold:

The global coordinates of the bar are found using the similar process used in question 3. The distance between coordinates are found using the formula,

$$d=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$$

Answer:

Table 4 Dimensions of the bar

Dimension Parameters	Dimensions of the bar (mm)
Length (L)	142.0709
Width (W)	20.1387
Height (H)	19.8134

Question 5: Estimate the volume of the bottle:

The global coordinates required to measure the dimensions of bottle are found using the similar process used in part 3. The distance between coordinates are found using the formula,

$$d=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$$

Part 1: Cylindrical part

Total volume of bottle = Volume of lower cylinder + Volume of upper frustum
Volume of lower cylinder = $\pi R^2 H$
Lower radius of the bottle (R) = 30.3436 mm
Height of the bottle until frustum (H) = 136.2655 mm
Volume of lower cylinder = 3941585 mm^3 = 394.1585 ml

Part 2: Frustum (top) part

Volume of upper frustum = $\pi * h^3 * (R^2 + r^2 + Rr)$
Radius at top frustum (r) = 16.0157 mm
Length (l) = 70.7347 mm
Frustum height (h) = $l^2 - r^2$
h = 63.8957 mm
Volume of upper frustum = 111288.17 mm^3 = 111.2881 ml

Total volume of bottle = Volume of lower cylinder + Volume of upper frustum = 394.1585 ml + 111.23 ml
Total volume of the bottle = 505.4467 ml

References

[1] D. Y. Jia. [Online]. Available: https://clemson.instructure.com/courses/128604/files/folder/Lecture%20Slides?preview=9155960.

[2] D. Jia. [Online]. Available: https://clemson.instructure.com/courses/128604/files/folder/Lecture%20Slides?preview=9155962.

[3] "Imtool MATLAB," [Online]. Available: https://www.mathworks.com/help/images/ref/imtool.html.

[4] D. J. S. fusion. [Online]. Available: https://clemson.instructure.com/courses/128604/files/folder/Lecture%20Slides?preview=9244265.