Kalman Filter

Main Points

- Very useful tool.
- It produces an optimal estimate of the state vector based on the noisy measurements (observations).
- For the state vector it also provides confidence (certainty) measure in terms of a covariance matrix .
- It integrates estimate of state over time.
- It is a sequential state estimator.

State-Space Model

State model error
With covariance
Q(k)

State-transition equation

$$\mathbf{z}(k) = \Phi(k, k-1)\mathbf{z}(k-1) + \mathbf{w}(k)$$

State Vector

Measurement (observation) equation

$$\mathbf{y}(k) = \mathbf{H}(k)\mathbf{z}(k) + \mathbf{v}(k)$$

Observation
Noise with covariance
R(k)

Measurement Vector

Kalman Filter Equations

State Prediction $$\hat{\mathbf{z}}_b(k) = \Phi(k, k-1)\hat{\mathbf{z}}_a(k-1)$$

Covariance Prediction $$\mathbf{P}_b(k) = \Phi(k, k-1)\mathbf{P}_a(k-1)\Phi^T(k, k-1) + \mathbf{Q}(k)$$

Kalman Gain $$\mathbf{K}(k) = \mathbf{P}_b(k)\mathbf{H}^T(k)(\mathbf{H}(k)\mathbf{P}_b(k)\mathbf{H}^T(k) + \mathbf{R}(k))^{-1}$$

State-update $$\hat{\mathbf{z}}_a(k) = \hat{\mathbf{z}}_b(k) + \mathbf{K}(k)[\mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{z}}_b(k)]$$

Covariance-update $$\mathbf{P}_a(k) = \mathbf{P}_b(k) - \mathbf{K}(k)\mathbf{H}(k)\mathbf{P}_b(k)$$

Two Special Cases

- Steady State

$$\Phi(k,k-1)=\Phi$$
$$\mathbf{Q}(k)=\mathbf{Q}$$
$$\mathbf{H}(k)=\mathbf{H}$$
$$\mathbf{R}(k)=\mathbf{R}$$

- Recursive least squares

$$\Phi(k,k-1)=\mathbf{I}$$
$$\mathbf{Q}(k)=0$$

Comments

- In some cases, state transition equation and the observation equation both may be non-linear.
- We need to linearize these equation using Taylor series.

Extended Kalman Filter

$$\mathbf{z}(k) = \mathbf{f}(\mathbf{z}(k-1)) + \mathbf{w}(k)$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{z}(k)) + \mathbf{v}(k)$$

Taylor series

$$\mathbf{f}(\mathbf{z}(k-1)) \approx \mathbf{f}(\hat{\mathbf{z}}_a(k-1)) + \frac{\partial \mathbf{f}(\mathbf{z}(k-1))}{\partial \mathbf{z}(k-1)}(\mathbf{z}(\mathrm{k}-1) - \hat{\mathbf{z}}_a(k-1))$$

$$\mathbf{h}(\mathbf{z}(k)) \approx \mathbf{h}(\hat{\mathbf{z}}_b(k)) + \frac{\partial \mathbf{h}(\mathbf{z}(k))}{\partial \mathbf{z}(k)}(\mathbf{z}(k) - \hat{\mathbf{z}}_b(k-1))$$

Extended Kalman Filter

$$\mathbf{z}(k) = \mathbf{f}(\mathbf{z}(k-1)) + \mathbf{w}(k)$$

$$\mathbf{z}(k) = \mathbf{f}(\hat{\mathbf{z}}_a(k-1)) + \frac{\partial \mathbf{f}(\mathbf{z}(k-1))}{\partial \mathbf{z}(k-1)}(\mathbf{z}(\mathrm{k}-1) - \hat{\mathbf{z}}_a(k-1)) + \mathbf{w}(k)$$

$$\mathbf{z}(k) \approx \Phi(k, k-1)\mathbf{z}(k-1) + \mathbf{u}(k) + \mathbf{w}(k)$$

$$\mathbf{u}(k) = \mathbf{f}(\hat{\mathbf{z}}_a(k-1)) - \Phi(k, k-1)\hat{\mathbf{z}}_a(k-1)$$

$$\Phi(k, k-1) = \frac{\partial \mathbf{f}(\mathbf{z}(k-1))}{\partial \mathbf{z}(k-1)}$$

Extended Kalman Filter

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{z}(k)) + \mathbf{v}(k)$$

$$\mathbf{y}(k) = \mathbf{h}(\hat{\mathbf{z}}_b(k)) + \frac{\partial \mathbf{h}(\mathbf{z}(k))}{\partial \mathbf{z}(k)}(\mathbf{z}(k) - \hat{\mathbf{z}}_b(k-1)) + \mathbf{v}(k)$$

$$\tilde{\mathbf{y}}(k) \approx \mathbf{H}(k)\mathbf{z}(k) + \mathbf{v}(k)$$

$$\tilde{\mathbf{y}}(k) = \mathbf{y}(k) - \mathbf{h}(\hat{\mathbf{z}}_b(k)) + \mathbf{H}(k)\hat{\mathbf{z}}_b(k)$$

$$\mathbf{H}(k) = \frac{\partial \mathbf{h}(\mathbf{z}(k))}{\partial \mathbf{z}(k)}$$

Multi-Frame Feature Tracking

Application of Kalman Filter

- Assume feature points have been detected in each frame.
- We want to track features in multiple frames.
- Kalman filter can estimate the position and uncertainty of feature in the next frame.
  - – Where to look for a feature
  - – how large a region should be searched

$$\mathbf{p}_k = [x_k, y_k]^T$$ Location

$$\mathbf{v}_k = [u_k, v_k]^T$$ Velocity

$$\mathbf{Z} = [x_k, y_k, u_k, v_k]^T$$ State Vector

System Model

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{v}_{k-1} + x_{k-1}$$

noise

$$\mathbf{v}_k = \mathbf{v}_{k-1} + h_{k-1}$$

$$\mathbf{Z}_k = \Phi_{k-1}\mathbf{Z}_{k-1} + \mathbf{w}_{k-1}$$

$$\Phi_{k-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{w}_{k-1} = \begin{bmatrix} x_{k-1} \\ h_{k-1} \end{bmatrix}$$

Measurement Model

$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{bmatrix} + m_k$$

$$\mathbf{y}_k = \mathbf{H} \begin{bmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{bmatrix} + m_k$$

Measurement matrix

Kalman Filter Equations

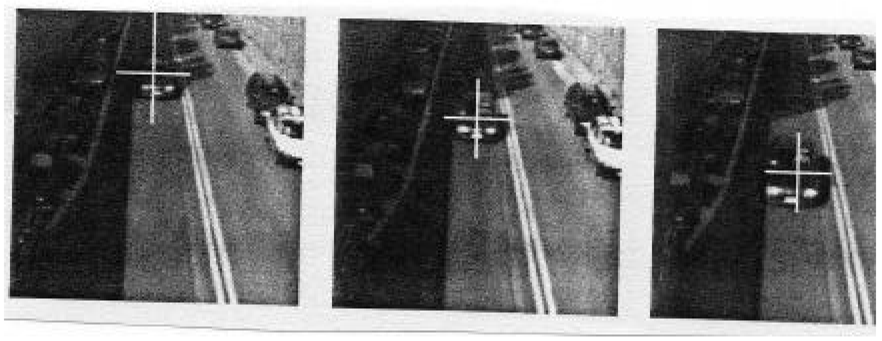
State Prediction $$\hat{\mathbf{z}}_b(k) = \Phi(k, k-1)\hat{\mathbf{z}}_a(k-1)$$

Covariance Prediction $$\mathbf{P}_b(k) = \Phi(k, k-1)\mathbf{P}_a(k-1)\Phi^T(k, k-1) + \mathbf{Q}(k)$$

Kalman Gain $$\mathbf{K}(k) = \mathbf{P}_b(k)\mathbf{H}^T(k)(\mathbf{H}(k)\mathbf{P}_b(k)\mathbf{H}^T(k) + \mathbf{R}(k))^{-1}$$

State-update $$\hat{\mathbf{z}}_a(k) = \hat{\mathbf{z}}_b(k) + \mathbf{K}(k)[\mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{z}}_b(k)]$$

Covariance-update $$\mathbf{P}_a(k) = \mathbf{P}_b(k) - \mathbf{K}(k)\mathbf{H}(k)\mathbf{P}_b(k)$$



Kalman Filter: Relation to Least Squares

$$f_i(\mathbf{Z}, \mathbf{y}_i) = 0$$

$$\Downarrow \quad \text{Taylor series}$$

$$f_i(\mathbf{Z}, \mathbf{y}_i) = 0 \approx f_i(\hat{\mathbf{Z}}_{i-1}, \hat{\mathbf{y}}_i) + \frac{\partial f_i}{\partial \mathbf{y}}(\mathbf{y} - \hat{\mathbf{y}}_i) + \frac{\partial f_i}{\partial \mathbf{z}}(\mathbf{z} - \hat{\mathbf{z}}_{\mathbf{i}})$$

$$\mathbf{Y}_i = H_i \mathbf{Z} + w_i$$

$$\mathbf{Y}_i = -f_i(\hat{\mathbf{Z}}_{i-1}, \hat{\mathbf{y}}_i) + \frac{\partial f_i}{\partial \mathbf{z}} \hat{\mathbf{z}}_{i-1}, H_i = \frac{\partial f_i}{\partial \mathbf{z}}$$

$$w_i = \frac{\partial f_i}{\partial \mathbf{y}}(\mathbf{y} - \hat{\mathbf{y}}_i)$$

Kalman Filter: Relation to Least Squares

$$C = (\hat{\mathbf{Z}}_0 - \mathbf{Z})^T P_0^{-1} (\hat{\mathbf{Z}}_0 - \mathbf{Z}) + \sum_{i=1}^{k} (\mathbf{Y}_i - H_i \mathbf{Z})^T W^{-1}{}_i (\mathbf{Y}_i - H_i \mathbf{Z})$$

$$\Downarrow \quad \text{minimize}$$

$$\hat{\mathbf{Z}} = [P_0^{-1} + \sum_{i=1}^{k} H_i{}^T W_i^{-1} H_i]^{-1} [P_0^{-1} \hat{\mathbf{Z}}_0 + \sum_{i=1}^{k} H_i{}^T W_i^{-1} \mathbf{Y}_i]$$

Batch Mode

Kalman Filter: Relation to Least Squares

$$\hat{\mathbf{Z}}_k = [P_0^{-1} + \sum_{i=1}^{k} H_i^T W_i^{-1} H_i]^{-1} [P_0^{-1}\hat{\mathbf{Z}}_0 + \sum_{i=1}^{k} H_i^T W_i^{-1} \mathbf{Y}_i]$$

$$\hat{\mathbf{Z}}_{k-1} = [P_0^{-1} + \sum_{i=1}^{k-1} H_i^T W_i^{-1} H_i]^{-1} [P_0^{-1}\hat{\mathbf{Z}}_0 + \sum_{i=1}^{k-1} H_i^T W_i^{-1} \mathbf{Y}_i]$$

Recursive Mode

Kalman Filter: Relation to Least Squares

$$\mathbf{Z}_k = \mathbf{Z}_{k-1} + K_k (Y_k - H_k \mathbf{Z}_{k-1})$$

$$K_k = P_{k-1} H^T{}_k \ (W^T + H^T P_{k-1} H_k{}^T)^{-1}$$

$$P_k = (I - K_k H_k) P_{k-1}$$

$$Y_k = -f^T(\mathbf{Z}_{k-1}, \mathbf{y}) + \frac{\partial f}{\partial \mathbf{Z}} \mathbf{Z}_{k-1}$$

$$H_k = \frac{\partial f}{\partial \mathbf{Z}}$$

$$W = \frac{\partial f}{\partial \mathbf{y}} \mathrm{A}^T \frac{\partial f}{\partial \mathbf{y}}^T$$

$$\Phi(k, k-1) = \mathbf{I}$$

$$\mathbf{Q}(k) = 0$$

Kalman Filter (Least Squares)

State Prediction $\hat{\mathbf{z}}_b(k) = \Phi(k,k-1)\hat{\mathbf{z}}_a(k-1)$

$$\hat{\mathbf{z}}_b(k) = \hat{\mathbf{z}}_a(k-1)$$

Covariance Prediction $\mathbf{P}_b(k) = \Phi(k,k-1)\mathbf{P}_a(k-1)\Phi^T(k,k-1) + \mathbf{Q}(k)$

$$\mathbf{P}_b(k) = \mathbf{P}_a(k-1)$$

Kalman Gain $\mathbf{K}(k) = \mathbf{P}_b(k)\mathbf{H}^T(k)(\mathbf{H}(k)\mathbf{P}_b(k)\mathbf{H}^T(k) + \mathbf{R}(k))^{-1}$

$$\mathbf{K}(k) = \mathbf{P}_b(k)\mathbf{H}^T(k)(\mathbf{H}(k)\mathbf{P}_b(k)\mathbf{H}^T(k) + \mathbf{W}(k))^{-1}$$

Kalman Filter (Least Squares)

State-update $\hat{\mathbf{z}}_a(k) = \hat{\mathbf{z}}_b(k) + \mathbf{K}(k)[\mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{z}}_b(k)]$

$$\hat{\mathbf{z}}\ (k) = \hat{\mathbf{z}}\ (k-1) + \mathbf{K}(k)[\mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{z}}\ (k-1)]$$

Covariance-update

$$\mathbf{P}_a(k) = \mathbf{P}_b(k) - \mathbf{K}(k)\mathbf{H}(k)\mathbf{P}_b(k)$$

$$\mathbf{P}\ (k) = \mathbf{P}\ (k-1) - \mathbf{K}(k)\mathbf{H}(k)\mathbf{P}\ (k-1)$$

Computing Motion Trajectories

Algorithm For Computing Motion Trajectories

- Compute tokens using Moravec's interest operator (intensity constraint).
- Remove tokens which are not interesting with respect to motion (optical flow constraint).
  - Optical flow of a token should differ from the mean optical flow around a small neighborhood.

Algorithm For Computing Motion Trajectories

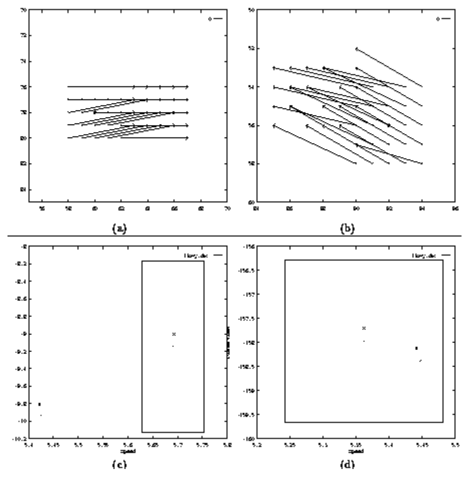
- Link optical flows of a token in different frames to obtain motion trajectories.
  - Use optical flow at a token to predict its location in the next frame.
  - Search in a small neighborhood around the predicted location in the next frame for a token.
- Smooth motion trajectories using Kalman filter.

Kalman Filter (Ballistic Model)

$$x(t) = .5a_x t^2 + v_x t + x_0 \qquad \mathbf{Z} = (a_x, a_y, v_x, v_y)$$

$$y(t) = .5a_y t^2 + v_y t + y_0 \qquad \mathbf{y} = (x(t), y(t))$$

$$f(\mathbf{Z}, \mathbf{y}) = (x(t) - .5a_x t^2 - v_x t - x_0, y(t) - .5a_y t^2 - v_y t - y_0)$$



Kalman Filter (Ballistic Model)

$$\mathbf{Z}(k) = \mathbf{Z}(k-1) + K(k)(Y(k) - H(k)\mathbf{Z}(k-1))$$

$$K(k) = P(k-1)H^T(k)\ \ (W^T + H^T P(k-1)H^T(k))^{-1}$$

$$P(k) = (I - K(k)H(k))P(k-1)$$

$$Y(k) = -f^T(\mathbf{Z}(\mathbf{k}-\mathbf{1}), \mathbf{y}) + \frac{\partial f}{\partial \mathbf{Z}}\mathbf{Z}(k-1)$$

$$H(k) = \frac{\partial f}{\partial \mathbf{Z}}$$

$$W = \frac{\partial f}{\partial \mathbf{y}} \mathrm{A}^T \frac{\partial f}{\partial \mathbf{y}}^T$$