Department of Computer Engineering
University of California at Santa Cruz

Object Tracking and Kalman Filtering

CMPE 264: Image Analysis and Computer Vision

Hai Tao

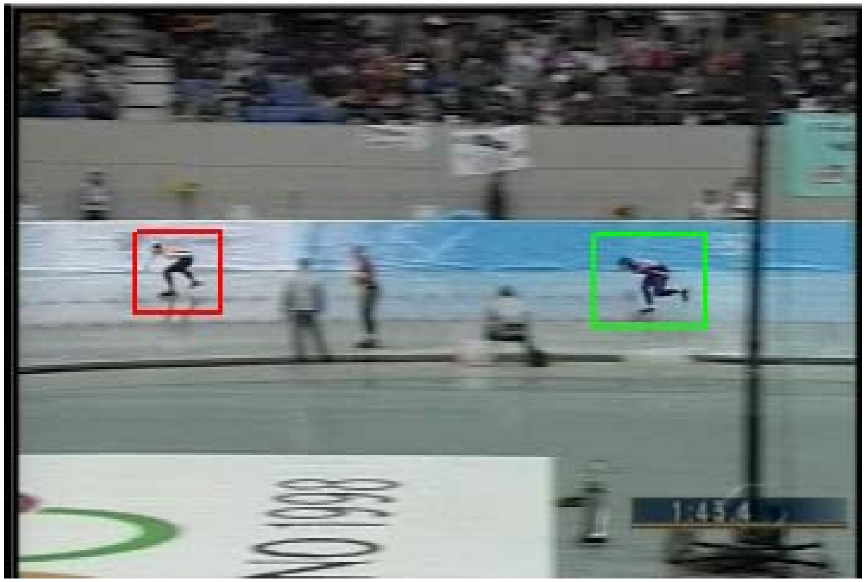
Object tracking

- Object tracking is the problem of estimating the positions and other relevant information of moving objects in image sequences
  - Two-frame tracking can be accomplished using correlation-based matching methods, optical flow techniques, or change-based moving object detection methods
  - The main difficulties in reliable tracking of moving objects include
    - Rapid appearance changes caused by image noise, illumination changes, nonrigid motion, and varying poses
    - Occlusion
    - Cluttered Background
    - Interaction between multiple multiple objects
  - In a long image sequence, if the dynamics of the moving object is known, prediction can be made about the positions of the objects in the current image. This information can be combined with the actual image observation to achieve more robust results

Department of Computer Engineering
University of California at Santa Cruz

Correlation-based tracking

- For a given region in one frame, find the corresponding region in the next frame by finding the maximum correlation score in a search region



Change-based tracking

- Algorithm
  - Align the background using a parametric motion model, e.g. a homography
  - Image subtraction to detection motion blobs
    - Compute the difference image between two frames
    - Thresholding to find the blobs
    - Locate the blob center as the position of the object

Change-based tracking

- Example

Tracking formulated using the Hidden Markov model

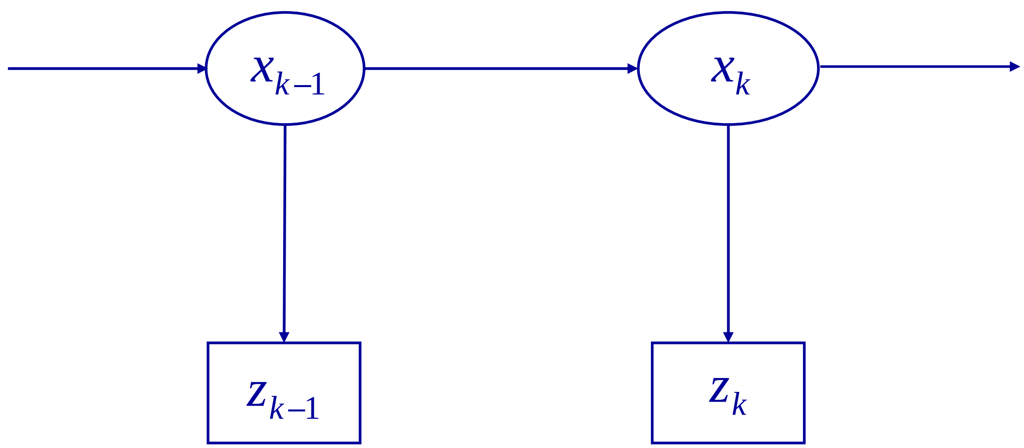
- Hidden Markov model (HMM)
  - x_k is called the hidden state and z_k is called the observation
  - Markov chain

$$p(x_k \mid x_1, \ldots, x_{k-1}) = p(x_k \mid x_{k-1})$$

$$p(z_k \mid x_1, \ldots, x_k) = p(z_k \mid x_k)$$

  - As the result

$$p(x_1, \ldots, x_k, z_1, \ldots, z_k) = p(x_1)\, p(z_1 \mid x_1) \prod_{i=2}^{k} [\, p(x_i \mid x_{i-1})\, p(z_i \mid x_i)]$$



Tracking formulated using the Hidden Markov model

- The state x_k usually consists of the position, the velocity, the acceleration, and other features of the object
- The observation z_k is usually the video frame at current time instance.
- The transition probability is modeled using dynamic model such as constant velocity model, or constant acceleration model
- Example: A constant velocity dynamic model

A constant velocity dynamic system

- The dynamic system and the state

$$\mathbf{x}_k = \Phi \mathbf{x}_{k-1} + \xi$$
$$\mathbf{z}_k = H \mathbf{x}_k + \mu$$

where $\mathbf{x}_k$ is the a state vector at time instant k, e.g. $\mathbf{x}_k = [x_k, y_k, v_{x_k}, v_{y_k}]$,

$\mathbf{z}_k$ is the a observation vector at time instant k, e.g. $\mathbf{z}_k = [z_x, z_y]$,

Φ is the state transition matrix, e.g. $\Phi = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

H is the measurement matrix, e.g. $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

ξ is a random vector modelling the uncertainty of the model, assumed to be $N(0, Q)$

μ is a random vector modelling teh additive noise in the observation, assumed to be $N(0, R)$

Forward algorithm

- To compute the *a posterior* probability $p(x_k \mid z_1,\ldots,z_k)$, we can use the forward algorithm

$$p(x_k \mid z_k,\ldots,z_1) \propto p(z_k \mid x_k, z_{k-1},\ldots,z_1)\, p(x_k \mid z_{k-1},\ldots,z_1)$$
$$= p(z_k \mid x_k) \int_{x_{k-1}} p(x_k, x_{k-1} \mid z_{k-1},\ldots,z_1) dx_{k-1}$$
$$= p(z_k \mid x_k) \int_{x_{k-1}} p(x_k \mid x_{k-1}, z_{k-1},\ldots,z_1)\, p(x_{k-1} \mid z_{k-1},\ldots,z_1) dx_{k-1}$$
$$= p(z_k \mid x_k) \int_{x_{k-1}} p(x_k \mid x_{k-1})\, p(x_{k-1} \mid z_{k-1},\ldots,z_1) dx_{k-1}$$

- Notice that Bayes rules is used for computing the *a posterior* probability

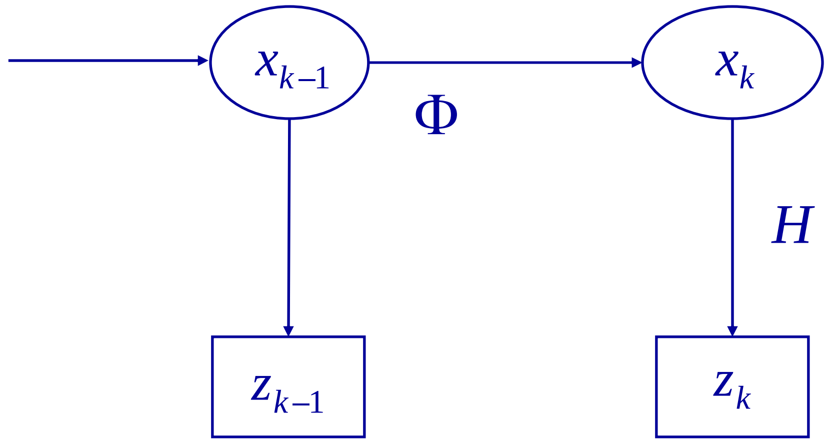
$$p(x \mid z) = \frac{p(z \mid x)\, p(x)}{p(z)}$$

Department of Computer Engineering
University of California at Santa Cruz

Probabilistic interpretation of the Kalman filter

$N(\hat{x}_{k-1} : x_{k-1}, P_{k-1})$

$p(x_k \mid x_{k-1}) = N(x_k^- : \Phi \hat{x}_{k-1}, \Phi P_{k-1} \Phi^T + Q)$



$p(z_k \mid x_k) = N(Hx_k : z_k, R)$

$$p(x_k \mid z_k, z_{k-1}, \dots, z_1) \propto p(z_k \mid x_k, z_{k-1}, \dots, z_1)\, p(x_k \mid z_{k-1}, \dots, z_1)$$

$$= p(z_k \mid x_k)\, p(x_k \mid z_{k-1}, \dots, z_1) = p(z_k \mid x_k) \int_{x_{k-1}} p(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{k-1}, \dots, z_1) dx_{k-1}$$

$$= N(Hx_k : z_k, R) \int_{x_{k-1}} N(x_k : \Phi x_{k-1}, \Phi P_{k-1} \Phi^T + Q) dx_{k-1}$$

(HW derivation of the Kalman filter)

The best linear unbiased estimator (BLUE)

- An optimal estimation algorithm – produce optimal estimates of the state of a dynamic system, on the basis of noisy measurements and an uncertain model of the system's dynamics
- To derive the Kalman filter, we will first introduce the best linear unbiased estimator (BLUE)

$$\mathbf{y} = H\mathbf{x} + \mathbf{n}$$

$\mathbf{y}$: observation, $\mathbf{x}$: state, H : measurement matrix, $\mathbf{n}$: noise $N(0,R)$

Given $\mathbf{y}, R, H$, estimate $\mathbf{x}$

- Linear filter: $\hat{\mathbf{x}} = L\mathbf{y}$
- Unbiased: $E(\hat{\mathbf{x}} - \mathbf{x}) = 0$
- Best: Covariance $E[(\hat{\mathbf{x}} - \mathbf{x})^T (\hat{\mathbf{x}} - \mathbf{x})]$ is minimum
- Lemma: The linear estimator is unbiased iff $I = LH$

  Proof:

$$E[\mathbf{x} - \hat{\mathbf{x}}] = E[\mathbf{x} - L\mathbf{y}] = E[\mathbf{x} - L(H\mathbf{x} + \mathbf{n})]$$
$$= E[(I - LH)\mathbf{x} - L\mathbf{n}] = E[(I - LH)\mathbf{x}]$$
$$E[(I - LH)\mathbf{x}] = 0 \text{ iff } LH = I$$

The best linear unbiased estimator (BLUE)

- Theorem: The best linear unbiased estimator $\hat{\mathbf{x}} = L\mathbf{y}$ for the measurement model $\mathbf{y} = H\mathbf{x} + \mathbf{n}$ is $L = (H^T R^{-1} H)^{-1} H^T R^{-1}$ and the covariance matrix of the estimate is $P = (H^T R^{-1} H)^{-1}$

  Proof: This is equivalent to $\min_L \| E[(L\mathbf{y} - \mathbf{x})(L\mathbf{y} - \mathbf{x})^T] \|$

  subject to $LH = I$

Observe that $E[(L\mathbf{y} - \mathbf{x})(L\mathbf{y} - \mathbf{x})^T] = E[L\mathbf{n}\mathbf{n}^T L^T] = LRL^T$

Suppose a solution can be written as $L = L_0 + (L - L_0)$

then it is obvious that $(L - L_0)H = LH - L_0 H = I - I = 0.$

The covariance can be written as

$P = LRL^T = (L_0 + (L - L_0))R(L_0 + (L - L_0))^T$

$= L_0 R L_0^T + (L - L_0) R L_0^T + L_0 R (L - L_0)^T + (L - L_0) R (L - L_0)^T$

Since $RL_0^T = R[(H^T R^{-1} H)^{-1} H^T R^{-1}]^T = H(H^T R^{-1} H)^{-1}$

therefore $(L - L_0) R L_0^T = (L - L_0) H (H^T R^{-1} H)^{-1} = 0,$ similarly $L_0 R (L - L_0)^T = [(L - L_0) R L_0^T]^T = 0$

As the result $P = L_0 R L_0^T + (L - L_0) R (L - L_0)^T$

Since both terms are positive definite or semidefinite matrices, it is minimum when $L = L_0$

Kalman filter – derivation

Suppose the prediction is $\hat{\mathbf{x}}_k^-$, with covariance matrix P_k^-. They are based on observations before k. As the result, if the true state is $\mathbf{x}_k$

$$\hat{\mathbf{x}}_k^- = \mathbf{x}_k + e_k : N(0, P_k^-)$$

Another piece of information is

$$\mathbf{z}_k = H\mathbf{x}_k + \mu : N(0, R)$$

This can be written as

$$\begin{bmatrix} \hat{\mathbf{x}}_k^- \\ \mathbf{z}_k \end{bmatrix} = \begin{bmatrix} I \\ H \end{bmatrix} \mathbf{x}_k + \mathbf{n} : N\left(0, \begin{bmatrix} P_k^- & 0 \\ 0 & R \end{bmatrix}\right)$$

The BLUE estimator of this system is

$$\hat{\mathbf{x}}_k = P_k [I, H^T] \begin{bmatrix} (P_k^-)^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_k^- \\ \mathbf{z}_k \end{bmatrix} = P_k [(P_k^-)^{-1} \hat{\mathbf{x}}_k^- + H^T R^{-1} \mathbf{z}_k]$$

$$P_k^{-1} = \left([I, H^T] \begin{bmatrix} (P_k^-)^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix} \begin{bmatrix} I \\ H_k \end{bmatrix} \right) = (P_k^-)^{-1} + H^T R^{-1} H$$

Therefore

$$\hat{\mathbf{x}}_k = P_k [(P_k^-)^{-1} \hat{\mathbf{x}}_k^- + H^T R^{-1} \mathbf{z}_k] = P_k [(P_k^{-1} - H^T R^{-1} H) \hat{\mathbf{x}}_k^- + H^T R^{-1} \mathbf{z}_k]$$

$$= \hat{\mathbf{x}}_k^- + P_k H^T R^{-1} (\mathbf{z}_k - H \hat{\mathbf{x}}_k^-)$$

This is the update stage of the Kalman filter

Department of Computer Engineering
University of California at Santa Cruz

Kalman filtering

Propogation stage:

$$\hat{\mathbf{x}}_{k+1}^- = \Phi \hat{\mathbf{x}}_k$$

$$P_{k+1}^- = E[(\hat{\mathbf{x}}_{k+1}^- - \mathbf{x}_{k+1})(\hat{\mathbf{x}}_{k+1}^- - \mathbf{x}_{k+1})^T]$$

$$= E[(\Phi\hat{\mathbf{x}}_k - \Phi\mathbf{x}_k - \xi)(\Phi\hat{\mathbf{x}}_k - \Phi\mathbf{x}_k - \xi)^T]$$

$$= \Phi P_k \Phi^T + Q$$

Update equations

$$\boxed{\begin{aligned} P_k &= ((P_k^-)^{-1} + H^T R^{-1} H)^{-1} \\ K_k &= P_k H^T R^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - H\hat{\mathbf{x}}_k^-) \end{aligned}}$$

or

$$\boxed{\begin{aligned} K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - H\hat{\mathbf{x}}_k^-) \\ P_k &= (I - K_k H) P_k^- (I - K_k H)^T + K_k R K_k^T \\ &= (I - K_k H) P_k^- \end{aligned}}$$

Propagation equations

$$\boxed{\begin{aligned} \hat{\mathbf{x}}_{k+1}^- &= \Phi \hat{\mathbf{x}}_k \\ P_{k+1}^- &= \Phi P_k \Phi^T + Q \end{aligned}}$$

Kalman filtering - algorithm

- Kalman_filtering_algorithm
  - Initialize $Q, R, \Phi, H, \mathbf{x}_0 = \mathbf{z}_0$, and P_0 as a large covariance matrix
  - For each time instant k=1,…
    - Predict $P_k^-, \mathbf{x}_k^-$ using the propogation equations
    - Compute the gain K_k and update $P_k, \hat{\mathbf{x}}_k$ using the update equations
    - Output the estimate $\hat{\mathbf{x}}_k$ and optionally the covariance matrix P_k

Homework

- In the Kalman filtering equations, does P_k depend on the observation data ? Does K_k depend on observation data ?
- Optional: Using the probabilistic interpretation to derive the Kalman equations