Good Features to Track

Good Features to Track

- Corners
- Moravec's Interest Operator

Corners

$$C = \begin{bmatrix} \sum_Q f_x^2 & \sum_Q f_x f_y \\ \sum_Q f_x f_y & \sum_Q f_y^2 \end{bmatrix}$$

$$\Downarrow$$

$$C = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Corners

- For perfectly uniform region $\lambda_1 = \lambda_2 = 0$
- If Q contains an ideal step ed

$$\lambda_2 = 0, \lambda_1 > 0$$

- if Q contains a corner of black square on white background $\lambda_1 \geq \lambda_2 \succ 0$

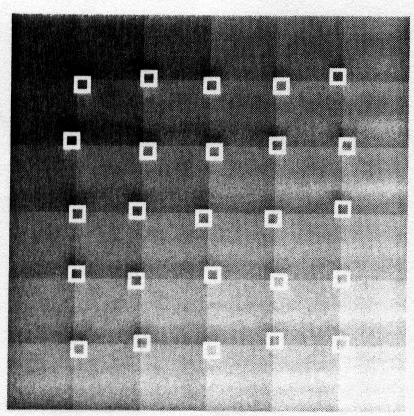
Algorithm Corners

- Compute the image gradient over entire image f.
- For each image point p:
  - form the matrix C over (2N+1)X(2N+1) neighborhood Q of p;
  - compute the smallest eigenvalue of C;
  - if eigenvalue is above some threshold, save the coordinates of p into a list L.

Algorithm Corners

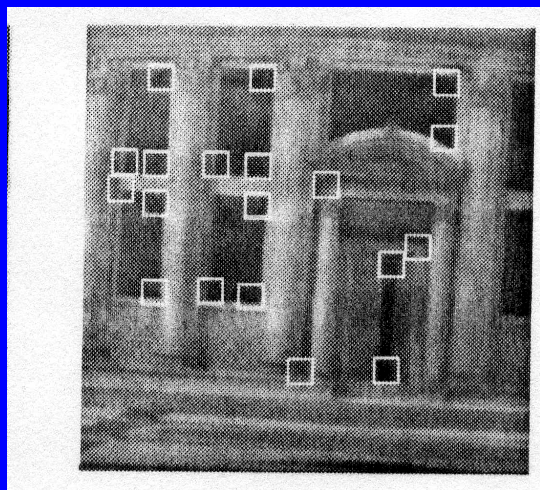
- Sort L in decreasing order of eigenvalues.
- Scanning the sorted list top to bottom: for each current point, p, delete all other points on the list which belong to the neighborhood of p.

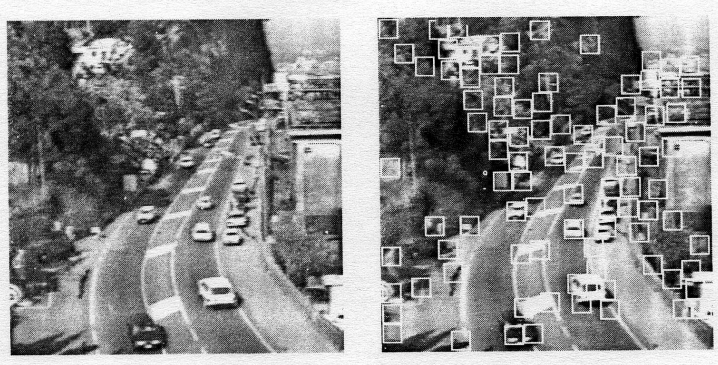
Results



(a)

Results



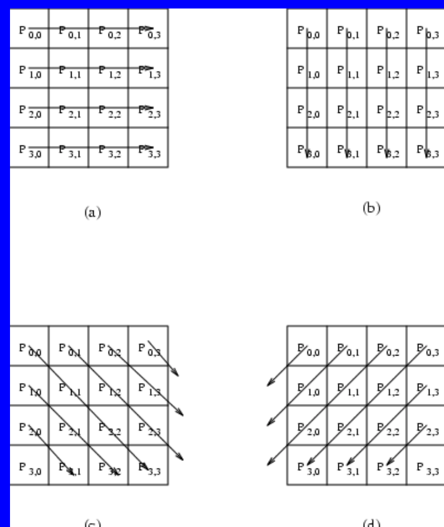


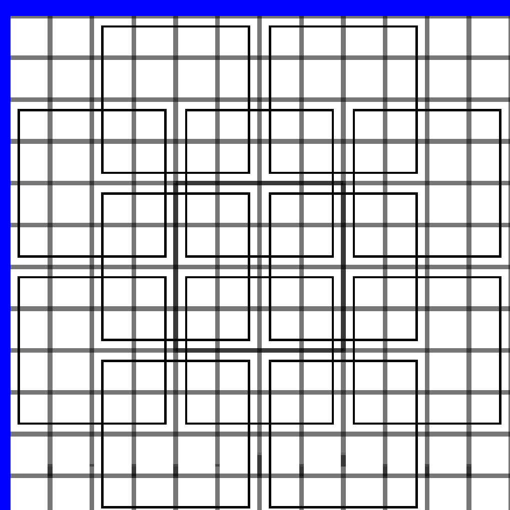
(a) (b)

Moravec’s Interest Operator

Algorithm

- Compute four directional variances in horizontal, vertical, diagonal and anti-diagonal directions for each 4 by 4 window.
- If the minimum of four directional variances is a local maximum in a 12 by 12 overlapping neighborhood, then that widow (point) is interesting.


P 0,0 P 0,1 P 0,2 P 0,3
P 1,0 P 1,1 P 1,2 P 1,3
P 2,0 P 2,1 P 2,2 P 2,3
P 3,0 P 3,1 P 3,2 P 3,3
(a)
(b)
(c)
(d)



$$V_h = \sum_{j=0}^{3}\sum_{i=0}^{2}(P(x+i,y+j)-P(x+i+1,y+j))^2$$

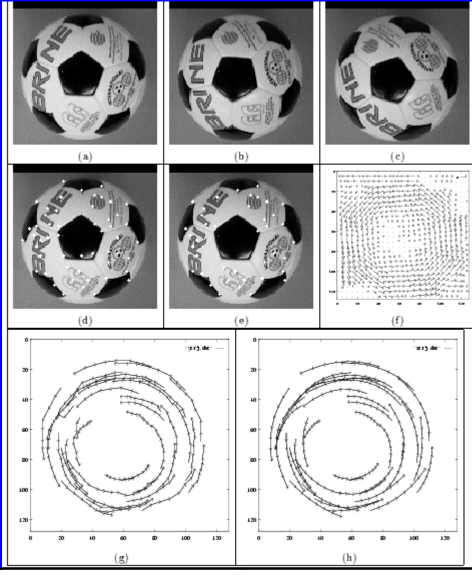
$$V_v = \sum_{j=0}^{2}\sum_{i=0}^{3}(P(x+i,y+j)-P(x+i,y+j+1))^2$$

$$V_d = \sum_{j=0}^{2}\sum_{i=0}^{2}(P(x+i,y+j)-P(x+i+1,y+j+1))^2$$

$$V_a = \sum_{j=0}^{2}\sum_{i=1}^{3}(P(x+i,y+j)-P(x+i-1,y+j+1))^2$$

$$V(x,y) = \min(V_h(x,y),V_v(x,y),V_d(x,y),V_a(x,y))$$

$$I(x,y) = \begin{cases} 1 & if V(x,y) local \max \\ 0 & 0therwise \end{cases}$$



Feature-based Matching

Feature-based Matching

- The input is formed by f1 and f2, two frames of an image sequence, and a set of corresponding feature points in two frames.
- Let Q1, Q2 and Q' be three NXN image regions.
- Let "d" be the unknown displacement vector between f1 and f2 of a feature point "p", on which Q1 is centered.

Algorithm

- (1) Set d=0, center Q1 on p1.
- (2) Estimate the displacement "d0" of "p", center of "Q1", using Lucas and Kanade method. Let d=d+d0.
- (3) Let Q' bet the patch obtained by warping Q1 according to "d0".
  - Compute Sum of Square (SSD) difference between new patch Q' and corresponding patch Q2 in frame f2.
- (4) If SSD more than threshold, set Q1=Q' and go to step 2, otherwise exit.

Lucas & Kanade (Least Squares)

- Optical flow eq

$$f_x u + f_y v = -f_t$$

- Consider 3 by 3 window

$$f_{x1} u + f_{y1} v = -f_{t1}$$

$$\vdots$$

$$f_{x9} u + f_{y9} v = -f_{t9}$$

$$\begin{bmatrix} f_{x1} & f_{y1} \\ \vdots & \vdots \\ & \\ f_{x9} & f_{y9} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_{t1} \\ \vdots \\ \\ -f_{t9} \end{bmatrix}$$

$$\mathbf{Au} = \mathbf{f_t}$$

Lucas & Kanade

$$\mathbf{Au} = \mathbf{f_t}$$

$$\mathbf{A^T A u} = \mathbf{A^T f_t}$$

$$\mathbf{u} = (\mathbf{A^T A})^{-1} \mathbf{A^T f_t}$$

$$\min \sum_{i=-2}^{2} \sum_{j=-2}^{2} (f_{xi} u + f_{yi} v + f_{ti})^2$$

Lucas & Kanade

$$\min \sum_{i=-2}^{2} \sum_{j=-2}^{2} (f_{xi}u + f_{yi}v + f_{ti})^2$$

$$\Downarrow$$

$$\sum (f_{xi}u + f_{yi}v + f_{ti}) f_{xi} = 0$$

$$\sum (f_{xi}u + f_{yi}v + f_{ti}) f_{yi} = 0$$

Lucas & Kanade

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum f_{xi}^2 & \sum f_{xi} f_{yi} \\ \sum f_{xi} f_{yi} & \sum f_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum f_{xi} f_{ti} \\ -\sum f_{yi} f_{ti} \end{bmatrix}$$

Lucas & Kanade

$$\min \sum_{i=-2}^{2} \sum_{j=-2}^{2} w_i (f_{xi} u + f_{yi} v + f_{ti})^2$$

$$\Downarrow$$

$$\mathbf{WAu} = \mathbf{Wf_t}$$

$$\mathbf{A^T WAu} = \mathbf{A^T Wf_t}$$

$$\mathbf{u} = (\mathbf{A^T WA})^{-1} \mathbf{A^T Wf_t}$$

Sum of Squares Differences

$$SSD = \sum_{j=-\frac{n}{2}}^{n} \sum_{i=-\frac{n}{2}}^{n} (Q'(x+i, y+j) - Q_2(x+i, y+j))^2$$

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 Vec

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

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

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

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

Taylor series

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

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

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

noise

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

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

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

Measurement Model

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

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

Measurement matrix

Kalman Filter Equations

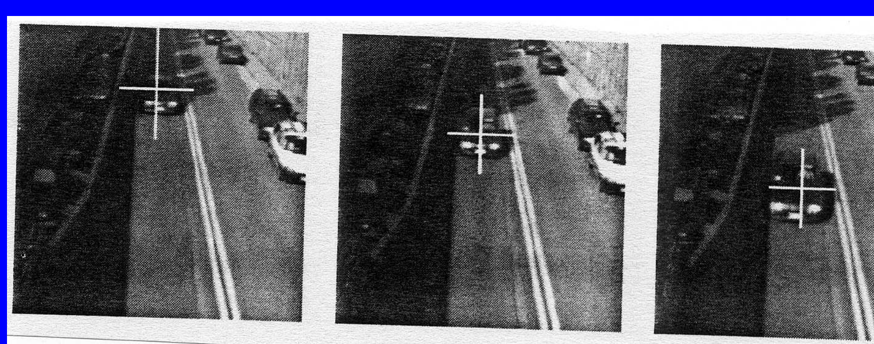
State Prediction $\quad \hat{\mathbf{z}}_b(k) = \Phi(k,k-1)\hat{\mathbf{z}}_a(k-1)$

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

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

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

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



Kalman Filter: Relation to Least Squares

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

$\hat{y}_i = y_i + l_i$

$E[l_i l_i^t] = \mathrm{A}_i$

Taylor series

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

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

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

New measurement

where

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

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

$W_i = \frac{\partial f_i}{\partial y}\mathrm{A}_i\frac{\partial f_i}{\partial y}^t$, covariance matrix of new measurement

Kalman Filter: Relation to Least Squares

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

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

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

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

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

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

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

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

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

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

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

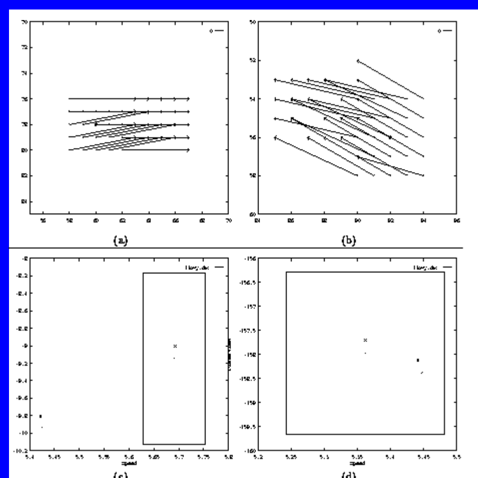
- Link optical flows of a token in different frames to obtain motion trajectories.
  - Use optical flow at a token to predict its location in the next frame.
  - Search in a small neighborhood around the predicted location in the next frame for a token.
- Smooth motion trajectories using Kalman filter.

Kalman Filter (Ballistic Model)

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

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

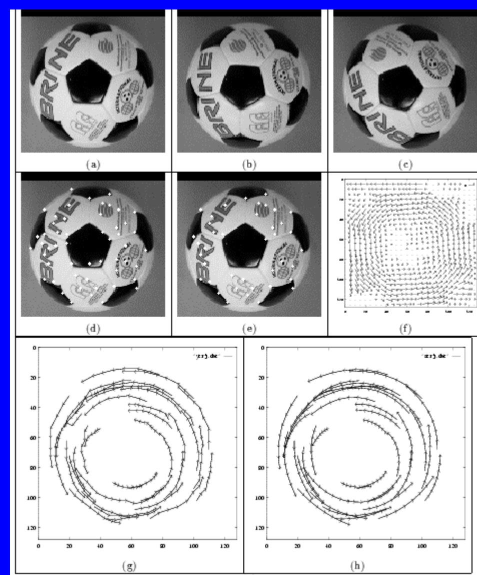
$$f(\mathbf{Z}, \mathbf{y}) = (x(t) - .5a_x t^2 - v_x t - x_0, y(t) - .5a_y t^2 - v_y t - y_0)$$

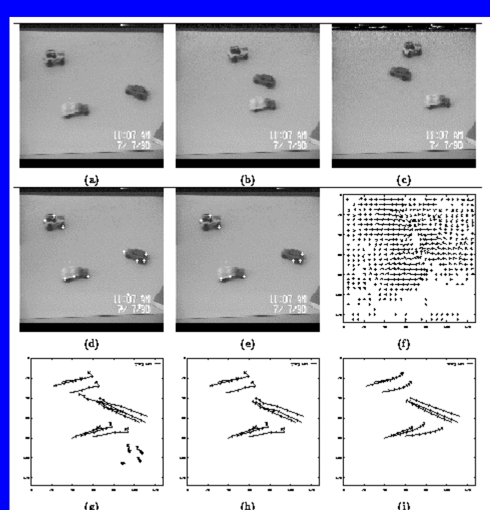


Kalman Filter (Ballistic Model)

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

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


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)