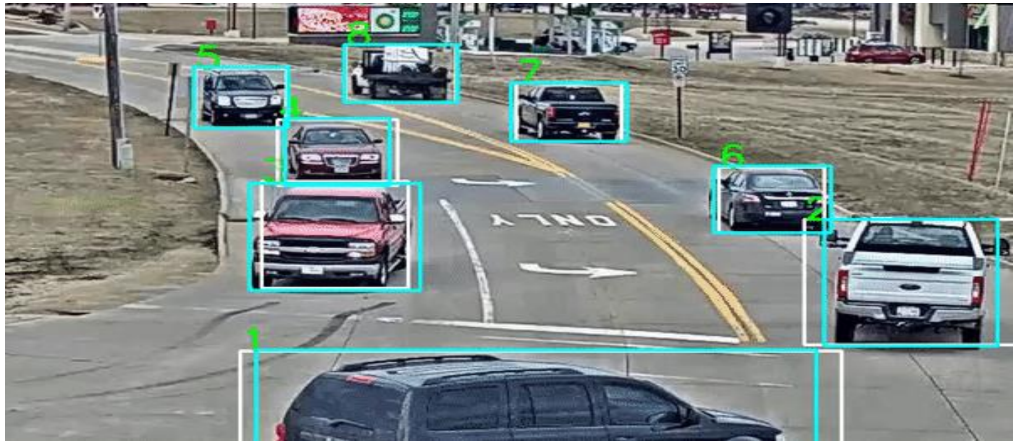
Tracking

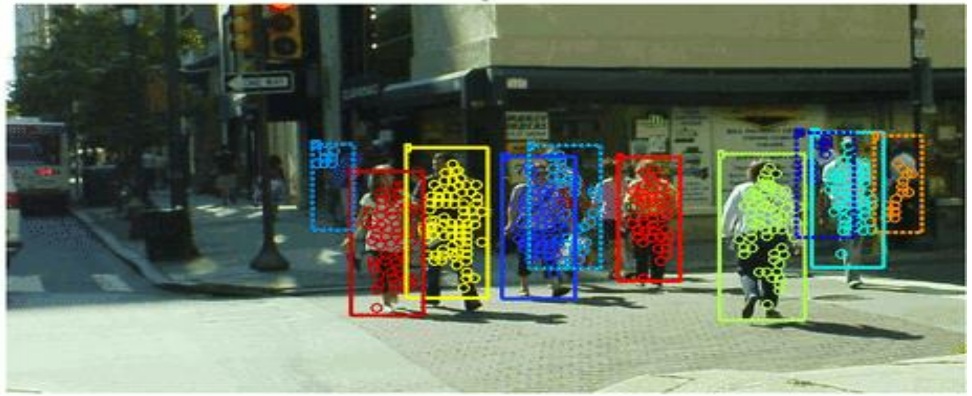
Jayanta Mukhopadhyay
Dept. of Computer Science and Engg.

Object tracking

- Estimating an object trajectory in the image plane as it moves.
  - Temporal sequence of images or A video
- Different contexts
  - Single / multi-object tracking
  - 2D / 3D trajectory
  - Point / Mass tracking
  - Rigid / deformable body tracking
  - Offline / Online tracking

A few examples





Optical flow

- The distribution of apparent velocities of movement of brightness pattern in an image
- Computed by assuming the brightness remains constant due to apparent motion of image point for a moving object.

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) + \underbrace{(dI/dx)\,\Delta x + (dI/dy)\,\Delta y + (dI/dt)\,\Delta t + \text{higher orders}}_{0}$$

$$(dI/dx)\,\Delta x + (dI/dy)\,\Delta y + (dI/dt)\,\Delta t = 0$$

$$(dI/dx)\,(\Delta x/\Delta t) + (dI/dy)\,(\Delta y/\Delta t) + (dI/dt) = 0$$

$$I_x v_x + I_y v_y + I_t = 0$$

$$\nabla I.\ v = -I_t$$

Assuming the velocity profile smooth, minimize also

$$|\nabla^2 v_x| + |\nabla^2 v_y| \text{ or } ||\nabla v_x||^2 + ||\nabla v_y||^2$$

Optical flow: Optimization problem (Horn and Shunk, AI,1980)

To minimize:

$E_1=\Sigma(\ (I_t+I_xv_x+I_yv_y)^2+k.(|\nabla^2 v_x|+|\nabla^2 v_y|))$

$\nabla^2 v=d^2v/dx^2+\ d^2v/dy^2$

Another function:

$E_2=\Sigma((I_t+I_xv_x+I_yv_y)^2+\ k.(||\nabla v_x||^{2}+||\nabla v_y||^2))$

$||\nabla v||^2=(dv/dx)^2+\ (dv/dy)^2$

A constant

$dE/dv_x=0$ and $dE/dv_y=0$

From E_2: $I_x^2v_x+I_xI_yv_y=k\nabla^2v_x-I_xI_t$

$I_xI_yv_x + I_y^2v_y=k\nabla^2v_y-I_yI_t$

$I_x^2 v_x + I_x I_y v_y = k\nabla^2 v_x - I_x I_t$

$I_x I_y v_x + I_y^2 v_y = k\nabla^2 v_y - I_y I_t$

Optical flow solution:

$\nabla^2 v \sim c.(v_m - v)$ $\quad\quad V_m = avg(v)$

A constant, usually taken as 3 in 2D.

Solving two equations,

$$(k+I_x^2+I_y^2)v_x = (k+I_y^2)v_{x,m} - I_x I_y v_{y,m} - I_x I_t$$

$$(k+I_x^2+I_y^2)v_y = -I_x I_y v_{x,m} + (k+I_x^2)\, v_{y,m} - I_y I_t$$

Need to solve a set of 2N simultaneous equations where N is the number of points in the image.

$$\boxed{v_x^{(n+1)} = v_{x,m}^{(n)} - I_x(I_x.\ v_{x,m}^{(n)} + I_y v_{y,m}^{(n)} + I_t)/\ (k+I_x^2+I_y^2)}$$

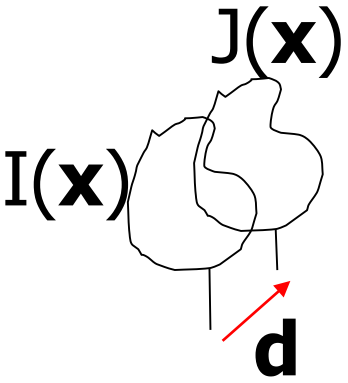
$$\boxed{v_y^{(n+1)} = v_{y,m}^{(n)} - I_y(I_x.\ v_{x,m}^{(n)} + I_y v_{y,m}^{(n)} + I_t)/\ (k+I_x^2+I_y^2)}$$

Optical flow: Pros and Cons

- Pros
  - No assumption on shape and motion
    - Applicable for rigid body and deformable motion
- Cons
  - Computes for every pixel, but does not model whole body motion.
  - Aggregation to be done as a post-processing.
    - non-trivial
  - Computationally intensive
  - Susceptible to noise

Kanade-Lucas-Tomasi (KLT) Tracker (Lucas & Kanade, IJCAI, 1981; Tomasi and Kanade, TR (CMU), 1991)

- Similar principle of optical flow.
  - Intensity remains constant in the direction of movement.



$$J(\mathbf{x})=I(\mathbf{x}-\mathbf{d}) = \underbrace{I(\mathbf{x}) - \nabla I.\mathbf{d} + \text{higher terms}}_{\text{Taylor series}}$$

$$I(\mathbf{x}) - J(\mathbf{x}) = h = \nabla I.\mathbf{d}$$

To minimize: $E=\sum_{\mathbf{x}\,\epsilon\, N_{\mathbf{x}}} w(\mathbf{x})(I(\mathbf{x})-J(\mathbf{x})- \nabla I.\mathbf{d})^2 \,/\, \sum w(\mathbf{x})$

$$\partial E/ \partial \mathbf{d} = 0 \;\Rightarrow\; \sum_{\mathbf{x}\,\epsilon\, N_{\mathbf{x}}} w'(h-\nabla I.\mathbf{d})\, \nabla I = 0$$

w′: Normalized

$w' = w / \sum w$

KLT Tracking solution

$$\sum_{\mathbf{x}\,\epsilon\, N_{\mathbf{x}}} w'(h-\nabla I.\mathbf{d})\,\nabla I = 0 \quad\Rightarrow\quad \sum_{\mathbf{x}\,\epsilon\, N_{\mathbf{x}}} w'(h\,\nabla I - \nabla I \nabla I^{T}\,\mathbf{d}) = 0$$

$\nabla I^{T}\,\mathbf{d}$

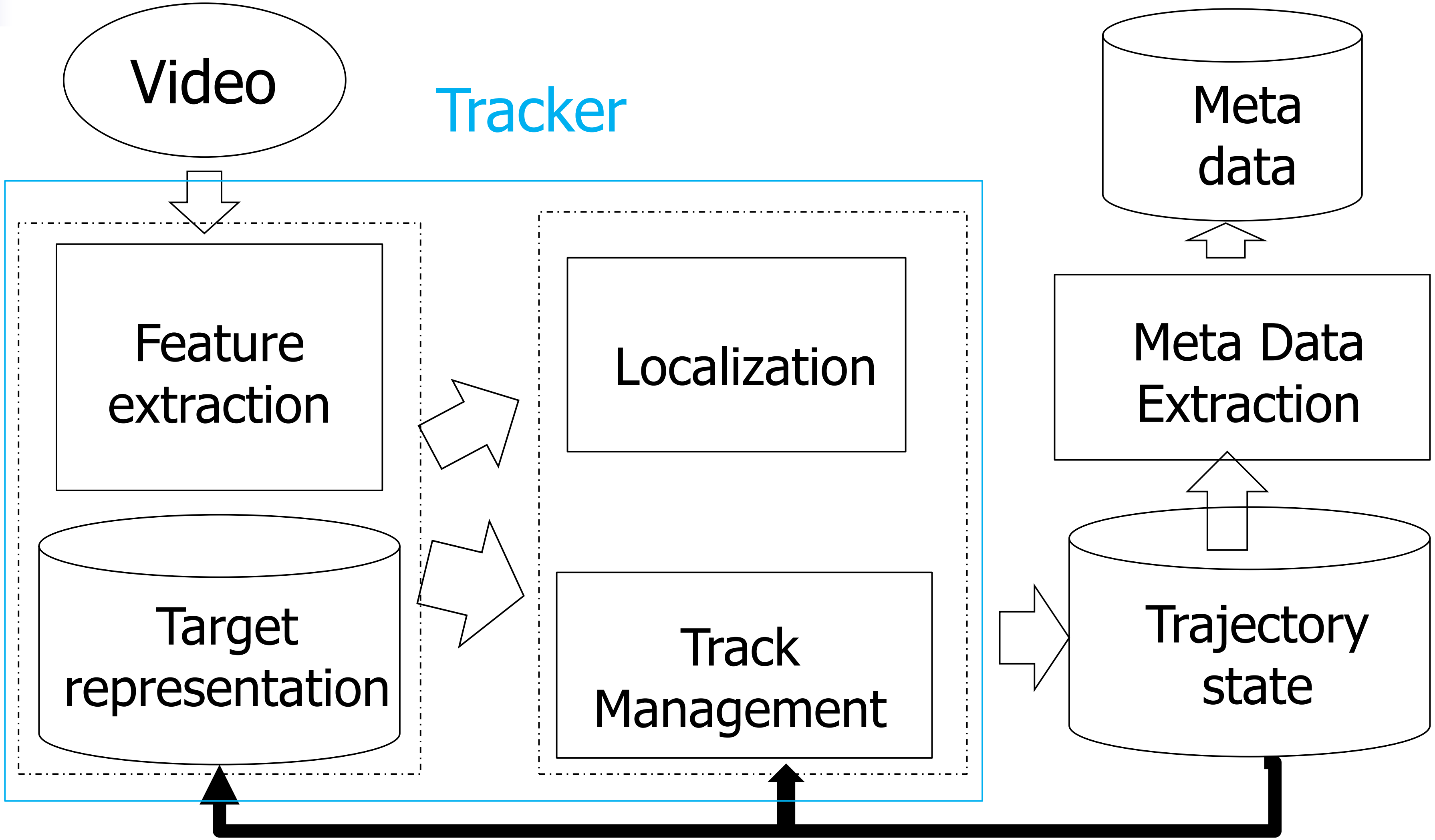
Column vector

$$\mathbf{d} = \Big(\sum_{\mathbf{x}\,\epsilon\, N_{\mathbf{x}}} w' \underbrace{\nabla I \nabla I^{T}}_{\text{2x2 matrix}}\Big)^{-1} \Big(\sum_{\mathbf{x}\,\epsilon\, N_{\mathbf{x}}} w'\, h\, \nabla I\Big)$$

KLT Tracker: Selection of feature points

- Corners: Changes perceptible in any direction in the neighborhood
  - Similar principle of Harris Operator.
  - Eigen values of $\sum w' \nabla I \nabla I^T$ at x should be high.
  - Minimum Eigen value to be greater than a threshold
- Compute **d** at those points and aggregate them appropriately

Object Tracking: Pipeline of Processing



Tracking approaches

- Generative Tracker
  - to establish the appearance model of the target
  - to search the region most similar to the target object in a continuous sequence
    - learning appearance model critical
      - subspace learning, sparse representation, spatio-temporal motion energy, boolean map…
- Discriminative Tracker
  - to formulate as a binary classification problem
    - a classifier trained to distinguish the target from the background
      - correlation filter based tracker

Chen et al, Visual object tracking: A survey, Computer Vision and Image Understanding, 222, (2022), 103508

Tracking approaches (Contd.)

- Collaborative Tracker
  - take advantage of both generative and discriminative approaches
- Deep learning based trackers
  - combines deep features with traditional tracking algorithms
  - End to end learning of a deep neural network for tracking.

Chen et al, Visual object tracking: A survey, Computer Vision and Image Understanding, 222, (2022), 103508

Generative tracker

- Given localized object develop appearance model:
  - Points, Kernels, Silhouette, Moment invariants, ..
- Distribution based descriptor
  - Histogram of pixel intensities, HOG, SIFT, GLOH
- Differential descriptor
  - Steerable filters, Gaussian derivatives
- Binary Descriptor
  - BRIEF, ORB, BRISK, FREAK
- Spatial-Frequency based descriptor
  - Gabor / Haar -wavelet responses, SURF

Generative tracker: Search object in consecutive frames

- Use of similarity measures in searching the object location in the next frame.
  - Normalized correlation coefficient (ncc).

$$ncc=(\Sigma I_t(\mathbf{x}).I_{t+1}(\mathbf{x+u}))^{1/2} \;/\; ((\Sigma I_t(\mathbf{x})^2)\;(\Sigma I_{t+1}(\mathbf{x+u})^2)^{1/2})$$

velocity

Can be defined in a feature space also.

Bayesian Tracking

- x_k : State of the object at time instance k
  - State: <location, velocity, acceleration …>
  - $x_k = f_k(x_{k-1}, v_{k-1})$; $v_{k-1} \sim$ i.i.d noise for all instances
- z_k : Measurements at time instance k
  - $z_k = h_k(x_k, u_k)$; $u_k \sim$ i.i.d noise for all instances
- To compute $p(x_k | z_{1:k})$
  - Assume for initial state x_0, $p(x_0)$ given (no measurement initially)
- Recursively compute from $p(x_{k-1} | z_{1:k-1})$
  - Apply Bayes' rule

Recursive Bayesian Tracking

- Chapman-Kolmogorov Equation
  - $p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1}, z_{1:k-1})\, p(x_{k-1}|z_{1:k-1})\, dx_{k-1}$
  - $p(x_k|x_{k-1}, z_{1:k-1}) = p(x_k|x_{k-1})$
    - First order Markovian system as $x_k = f_k(x_{k-1}, v_{k-1})$
  - Hence, $p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})\, p(x_{k-1}|z_{1:k-1})\, dx_{k-1}$

Prediction from $p(x_{k-1}|z_{1:k-1})$

- Apply Bayesian Rule for computing $p(x_k|z_{1:k})$
  - $p(x_k|z_{1:k}) = p(z_k|\, x_k)\, p(x_k|\, z_k, z_{1:k-1}) \,/\, p(z_k|z_{1:k-1})$
    - where $p(z_k|z_{1:k-1}) = \int p(z_k|x_{k-1})\, p(x_{k-1}|z_{1:k-1})\, dx_{k-1}$

Update

x_k conditionally independent of $z_{1:k}$

M. S. Arulampalam, et al, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, Feb. 2002,

Recursive Bayesian Tracking :Kalman filtering

- With posterior density at every time step Gaussian.
  - If $p(x_{k-1}|z_{1:k-1})$ is Gaussian, $p(x_k|z_{1:k-1})$ is also Gaussian for linear functional relationship of state and measurement with Gaussian noises.
    - $x_k = F_k x_{k-1} + v_{k-1};\ v_{k-1} \sim \mathcal{N}(0, Q_{k-1})$
    - $z_k = H_k x_k + u_k;\ u_k \sim \mathcal{N}(0, R_k)$
- Posterior densities:
  - $p(x_{k-1}|z_{1:k-1}) \sim \mathcal{N}(m_{k-1|k-1}\ ,\ P_{k-1|k-1})$
  - $p(x_k|z_{1:k-1}) \sim \mathcal{N}(m_{k|k-1}\ ,\ P_{k|k-1})$
  - $p(x_k|z_{1:k}) \sim \mathcal{N}(m_{k|k}\ ,\ P_{k|k})$

Note state and measurement equations are time varying.

Kalman Filtering: Recursive Solution

- Given the functional relationships, density functions and the measurement $z_k^{(o)}$
  - $x_k = F_k x_{k-1} + v_{k-1};\ v_{k-1} \sim \mathcal{N}(0, Q_{k-1})$
  - $z_k = H_k x_k + u_k;\ u_k \sim \mathcal{N}(0, R_k)$
  - $p(x_{k-1} | z_{1:k-1}) \sim \mathcal{N}(m_{k-1|k-1}, P_{k-1|k-1})$
- Compute $p(x_k | z_{1:k})$
  - Estimate parameters of density functions

Kalman Filtering: Recursive Solution

$x_k = F_k x_{k-1} + v_{k-1};\ v_{k-1} \sim \mathcal{N}(0, Q_{k-1})$
$z_k = H_k x_k + u_k;\ u_k \sim \mathcal{N}(0, R_k)$
$p(x_{k-1} | z_{1:k-1}) \sim \mathcal{N}(m_{k-1|k-1}, P_{k-1|k-1})$
Measurement: $z_k^{(o)}$

- Predict
  - $m_{k|k-1} = F_k m_{k-1}$
  - $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_{k-1}$
- Update
  - $m_{k|k} = m_{k|k-1} + \underbrace{K_k}_{\text{Kalman gain}} (Z_k^{(o)} - H_k\, m_{k|k-1})$
  - $P_{k|k} = P_{k-1|k-1} - K_k H_k P_{k-1|k-1}$

$$\boxed{\begin{aligned} S_k &= H_k P_{k-1|k-1} H_k^T + R_k \\ K_k &= P_{k|k-1} H_k^T S_k^{-1} \end{aligned}}$$

Kalman filtering: Another perspective

- Decision fusion handling uncertainties.
- Multiple pairwise uncorrelated noisy measurements / estimates
- A linear combination of estimates minimizing its variance
  - Gaussian pdf
  - Handling prior in tracking: A special case
    - Two different observations:
      - Prediction of states of motion (location, velocities, ..) from previous frame
      - Measurement in the current frame
        - Correction through decision fusion

Noises are not shown for convenience.

Coefficients are inversely proportional to variances of estimates

$$\underbrace{x_k = a.f(x_{k-1}) + b.\ h(z_k)}$$

$$\left.\begin{array}{l} x_k^{(1)} = f(x_{k-1}) \\ x_k^{(2)} = h(z_k) \end{array}\right\}$$

Kalman filtering in 1-D

- Let $x_1 \sim p_1(\mu_1, \sigma^2_1)$, ..., $x_n \sim p_n(\mu_n, \sigma^2_n)$ be a set of pairwise uncorrelated random variables.
- Let $y = \sum \alpha_i x_i$
  - mean and variance :
    - $\mu_y = \sum \alpha_i \mu_i$
    - $\sigma^2_y = \sum \alpha^2_i \sigma^2_i$
  - If a random variable z is pairwise uncorrelated with x_1, .., x_n, it is uncorrelated with y.

y: a random variable and a linear combination of the x_i 's.

Pei et al, An Elementary Introduction to Kalman Filtering, arxiv, June, 2019

Optimal linear combination

- Optimal coefficients with minimum variance of y
  - $\sum a_i = 1$
  - $a_i = (1/\sigma^2_i\ ) \ /\ \sum (1/\sigma^2_j)$
- Optimal estimate:
  - $y^* = \sum a_i x_i$

Kalman Filter: Extension to multidimension

- Let $\mathbf{x}_1 \sim p_1(\boldsymbol{\mu}_1, \Sigma_1)$, ..., $\mathbf{x}_n \sim p_1(\boldsymbol{\mu}_n, \Sigma_n)$ be a set of pairwise uncorrelated random variables.
- Let $\mathbf{y} = \sum A_i \mathbf{x}_i$ be a random variable
  - The mean and variance of y are:
    - $\boldsymbol{\mu}_\mathbf{y} = \sum A_i \boldsymbol{\mu}_i$
    - $\Sigma_\mathbf{y} = \sum A_i \Sigma_i A_i^T$
  - If a random variable **z** is pairwise uncorrelated with $\mathbf{x}_1, .., \mathbf{x}_n$, it is uncorrelated with **y**.

Kalman Filter: Optimal linear combination in multidimension

- To minimize MSE of **y** : $E((\mathbf{y}-\boldsymbol{\mu}_\mathbf{y})^T(y-\boldsymbol{\mu}_\mathbf{y}))$
  - $\sum A_i = I$
  - $A_i = (\Sigma_i)^{-1} (\sum (\Sigma_j)^{-1})^{-1}$
- Optimal estimate: $\sum A_i \mathbf{x}_i$

Linear combination rewritten for Kalman Gain.

For two variables:

- Prediction: $\mathbf{x}_1$ with F
- Measurement: $\mathbf{z} \rightarrow \mathbf{x}_2 = H\mathbf{z}$
- Correction: $\mathbf{y} = \mathbf{x}_1 + K(\mathbf{x}_2 - \mathbf{x}_1)$

  where $K = \Sigma_1 (\Sigma_1 + \Sigma_2)^{-1}$

Kalman Gain

Uncertainty in **y**:

$\Sigma_\mathbf{y} = (I-K)\,\Sigma_1$

Graph based method

Construction of a Directed Weighted Graph

Objects in a frame form nodes.

Between two correlated objects in two different frames an arc (edge) is formed.

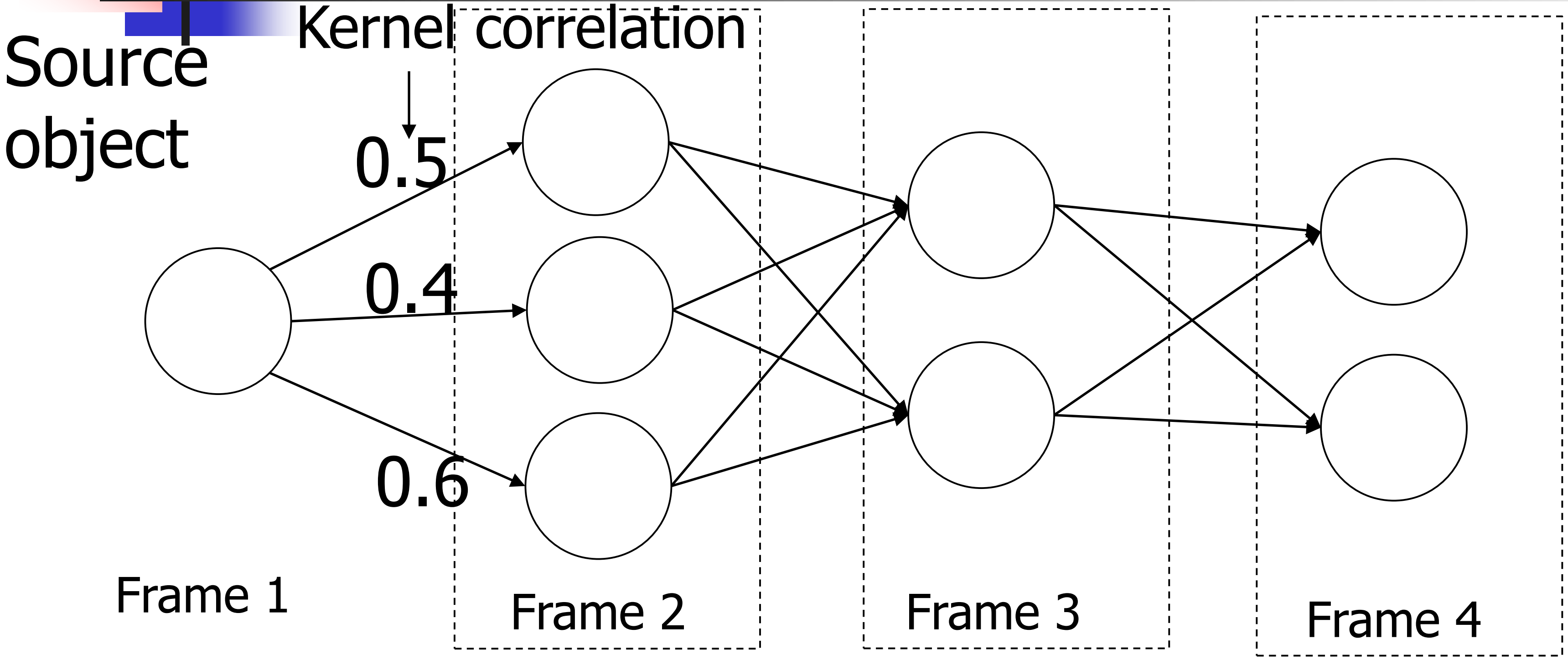
The measure of correlation provides the weight.

Temporal direction provides the direction of the edge.

Trajectory as the longest path in a graph

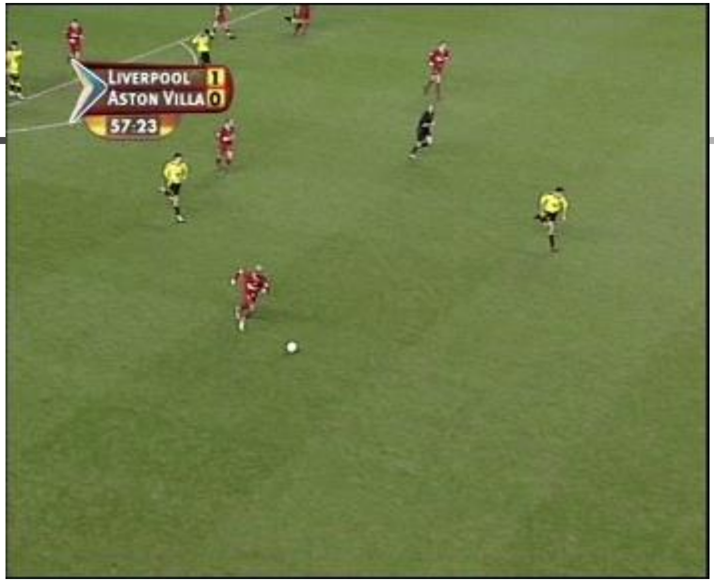
Pallavi et al, Graph Based Multi-Player Detection and Tracking in Broadcast Soccer Videos, IEEE trans. on Multimedia, vol. 10, no. 5, Aug., 794-805, 2008

Object Trajectory Detection



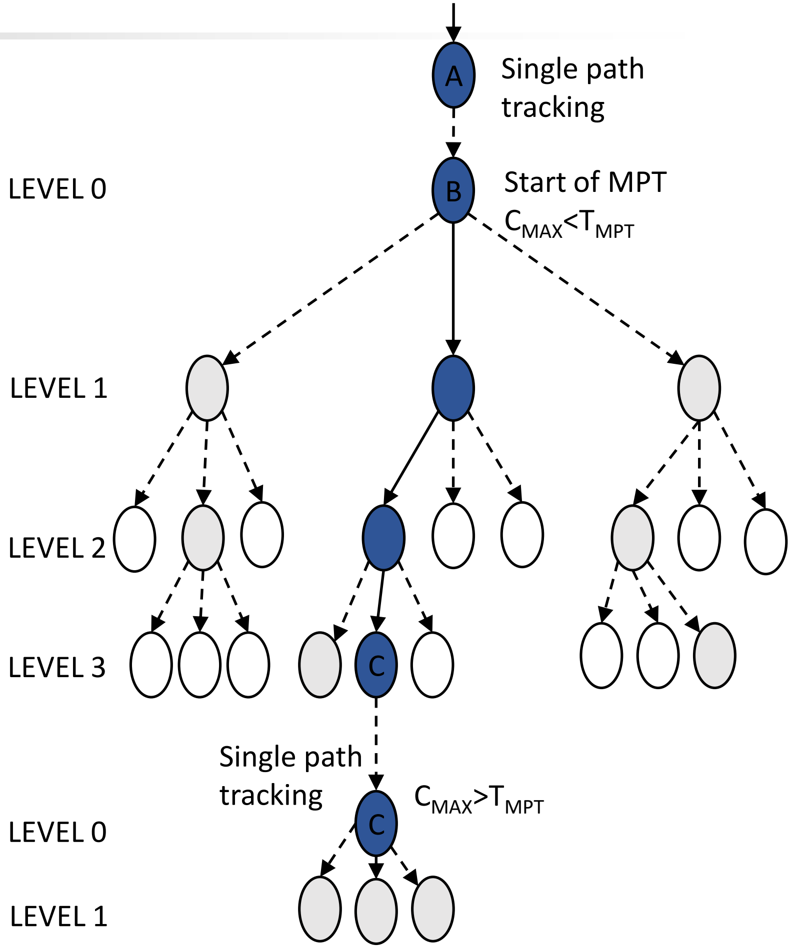
Given a source node, longest path of the graph obtained by dynamic programming gives the path of the object.

Ball detection in long shots



Multi-path tracking (MPT)

- A greedy on the fly approach
- Use of maximum kernel correlation score (C_{MAX}) to find object location in the next frame.
- If C_{MAX} of a block > threshold value (T_{MPT}),
  - Start / Continue MPT with top K candidates
  - else
  - Start / Continue SPT.



D. P. Dogra *et al.*, “Video analysis of Hammersmith lateral tilting examination using Kalman filter guided multi-path tracking,” *Med. Biol. Eng. Comput.*, vol. 52, no. 9, pp. 759–772, Sep. 2014.

Discriminative Tracker

Tracking as detection

- Classifying background / foreground
  - Given pairs of training image and its output, (f_i, g_i), design a classifier.
    - Output in the form of 0 /1 or -1 /+1 (background / foreground)
    - Choice of features and classifiers?
      - Texture features, …

Learning correlation filter

- Correlation of $f(x)$ with $h(.)$

$$\boxed{g(x) = f \otimes h(x) = \sum_{i=-K}^{K} f(x+i)h(i)}$$

- Correlation of an image

$$\boxed{g(m,n) = \sum_{k=-K}^{K} \sum_{l=-K}^{K} f(m+k, n+l)h(k,l)}$$

- Fourier Transforms

$$\boxed{G = F \odot H^*}$$

$$\boxed{G = F \odot H^*}$$

Complex conjugate

Given labelled data of tracking (Object locations in consecutive / a set of frames) learn H.

Optimization function

$$\sum_i |F_i \odot H^* - G_i|^2$$

The method of linear regression

- Problem statement:
  - Given $(\mathbf{x}_i, y_i)$, i=1,2,..,N, where $\mathbf{x}_i$ is a vector and y_i is a scalar value, get the filter coefficients **w** (in the form of a vector) so that $\mathbf{w}^T\mathbf{x}_i = y_i$, for all i.
  - Let $X=[\mathbf{x}_1\ \mathbf{x}_2\ \dots\ \mathbf{x}_N]$ and $y=[y_1\ y_2\ \dots\ y_N]^T$
- Ridge regression: $\boxed{\min_{\mathbf{w}} \|\mathbf{w}^T X - y\|^2 + \lambda\|\mathbf{w}\|^2}$

$$\partial E / \partial \mathbf{w} = 0 \quad \Rightarrow \quad 2(\mathbf{w}^T X - y) X^T + 2\lambda \mathbf{w}^T = 0$$

$$\Downarrow$$

$$\mathbf{w}^T X X^T - y X^T + \lambda \mathbf{w}^T = 0$$

$$\boxed{\mathbf{w} = (X^T X + \lambda I)^{-1} X^T \mathbf{y}}$$

Kernelized Correlation Filters (KCF)

- Core component of the tracker is a Discriminative classifier.

- **Discriminative classifier** : Distinguishes between target and back ground.

- Classifier is trained with scaled and translated image patches to cope with natural scene changes

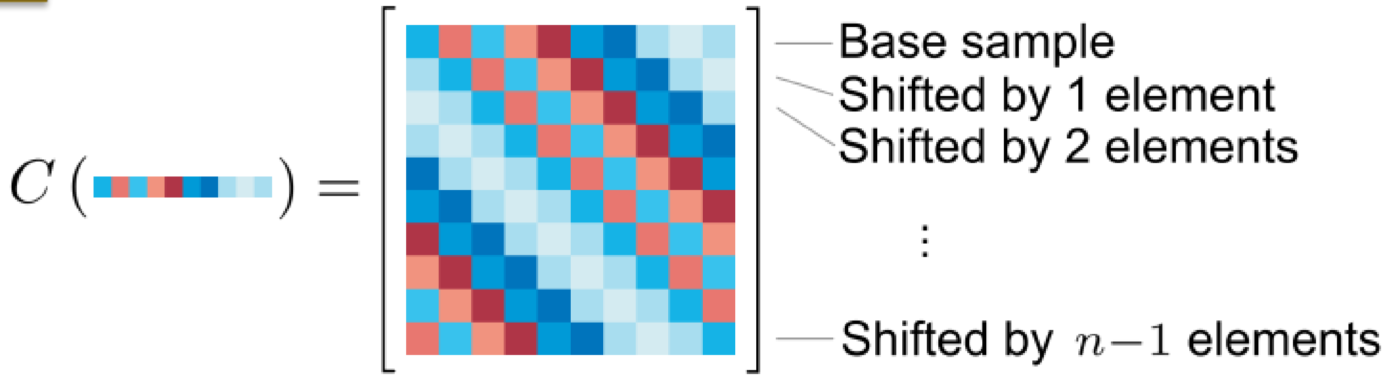
- Circulant data matrix is constructed based in thousands of translated patches.

Henriques João F. et al. "High-speed tracking with kernelized correlation filters." *IEEE PAMI,2015*

Circulant Matrix

➢Circulant matrix:

$$X = C(\mathbf{x}) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}$$



➢Relationship with Fourier transform:

$$\boxed{X = F^H \, diag(\hat{\mathbf{x}}) \, F}$$

IDFT

$\hat{\mathbf{x}}$: Fourier transform of vector $\mathbf{x} = (x_1, \ldots, x_n)$

F: the DFT matrix ➔$\hat{\mathbf{x}} = \mathcal{F}(\mathbf{x}) = \sqrt{n}\, F\, \mathbf{x}$

H: Hermitian transpose

Henriques João F. et al. "High-speed tracking with kernelized correlation filters." *IEEE PAMI,2015*

Ridge regression for KCF

$$\boxed{\min_{\mathbf{w}} \sum_i (f(\mathbf{x}_i) - y_i)^2 + \lambda ||\boldsymbol{w}||^2}$$

λ is a regularization parameter that controls overfitting

y_i is the regression target

$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$

$\mathbf{w}$ is the weight vector

Henriques João F. et al. "High-speed tracking with kernelized correlation filters." *IEEE PAMI,2015*

KCF: Solution of Loss function for Linear Regression

$$\mathbf{w} = (X^T X + \lambda I)^{-1} X^T \mathbf{y} \quad \Leftrightarrow \quad \mathbf{w} = (X^H X + \lambda I)^{-1} X^H \mathbf{y}$$

$$X^H X = F^H \, diag(\hat{\mathbf{x}}^*) F F^H diag(\hat{\mathbf{x}}) F = F^H \, diag(\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}}) \, F$$

$$X^H X + \lambda I = F^H \, diag(\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda) \, F$$

$$X^H \mathbf{y} = F^H \, diag(\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}) \, F$$

$\mathbf{w}$

IDFT

$$\hat{\mathbf{w}} = \left(\frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda} \right)$$

The fraction denotes element-wise division.

KCF: Non-linear Regression

Mapping to higher dimension

$\mathbf{x}_i \rightarrow \varphi(\mathbf{x}_i)$

$$\mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}_i)$$

$\mathbf{w}$ as a linear combination of the samples

$$f(\mathbf{z}) = \mathbf{w}^T \varphi(\mathbf{z}) \Rightarrow f(\mathbf{z}) = \sum_i \alpha_i \underbrace{\varphi^T(\mathbf{x}_i)\,\varphi(\mathbf{z})}$$

Not explicitly computed

Kernel function

$$f(\mathbf{z}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{z})$$

Kernel Matrix: $K = [K_{ij}]$

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \varphi^T(\mathbf{x}_i)\varphi(\mathbf{x}_j)$$

KCF: Kernel Regression

$$\mathbf{x}_i \rightarrow \varphi(\mathbf{x}_i) \qquad \mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}_i) \qquad f(\mathbf{z}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{z})$$

➢Optimization Problem:

$$\boxed{\min_{\mathbf{w}} \sum_j \left(f(\mathbf{x}_j) - y_j\right)^2 + \lambda \|\boldsymbol{w}\|^2}$$

Kernel Matrix: $K = [K_{ij}]$

dual space variable

$$\boxed{\min_{\boldsymbol{\alpha}} \sum_j \left(\sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_j) - y_j\right)^2 + \lambda \|\boldsymbol{w}\|^2}$$

$$K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$$

If the kernel functional value is invariant to ordering of dimensions, the kernel derived from the columns of circulant matrix is also circulant.

⇨ $K = C(\mathbf{k}^{\mathbf{xx}})$

$\mathbf{k}^{\mathbf{xx}}$ is the first row of kernel matrix

KCF: Solution of Non-linear Regression

$\mathbf{x}_i \rightarrow \varphi(\mathbf{x}_i)$

$$\mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}_i)$$

$f(\mathbf{z}) = \mathbf{w}^T \varphi(\mathbf{z}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{z})$

$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \varphi^T(\mathbf{x}_i)\varphi(\mathbf{x}_j)$

$$\boxed{\boldsymbol{\alpha} = \frac{\mathbf{y}}{K + \lambda I}}$$

$$\boxed{\hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\mathbf{xx}} + \lambda}}$$

$\mathbf{k}^{\mathbf{xx}}$ is the first row of kernel matrix

- Different kernels:

Gaussian kernel ⇨ $\boxed{k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{\sigma^2}(\|\mathbf{x}_i - \mathbf{x}_j\|^2)\right)}$

Linear kernel ⇨ $\boxed{k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i{}^{\boldsymbol{T}}\mathbf{x}_j}$

Summary

- Object Tracking
  - Estimating an object trajectory in the image plane as it moves
- Optical flow
  - The distribution of apparent velocities of movement of brightness pattern in an image
  - Governing equation:
    - $\nabla I . v = -I_t$
- KLT Tracker

$$\sum_{\mathbf{x} \in N_{\mathbf{x}}} w'(h - \nabla I.\mathbf{d})\, \nabla I = 0$$

- Recursive Bayesian Tracking
  - To compute $p(x_k|z_{1:k})$ recursively from $p(x_{k-1}|z_{1:k-1})$
  - Kalman Filtering
- Graph based method
  - Longest path with accumulated evidence from source to destination in a directed acyclic graph.
  - On the fly greedy approach
    - Multipath tracking
- Kernelized Correlation Filter

Thank you!