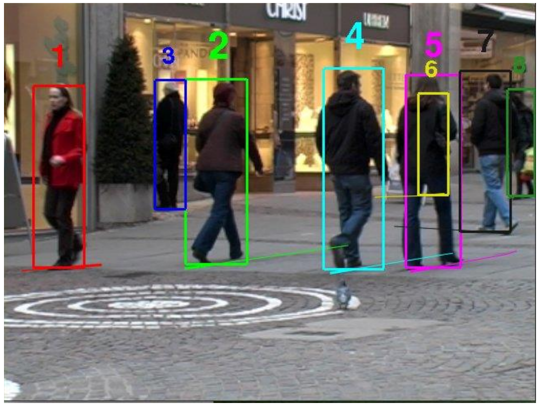
CMP9135M - Computer Vision

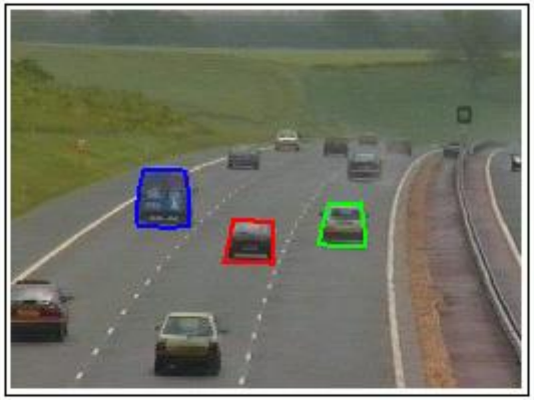
Visual Tracking II

Dr Nicola Bellotto

Recap of last week

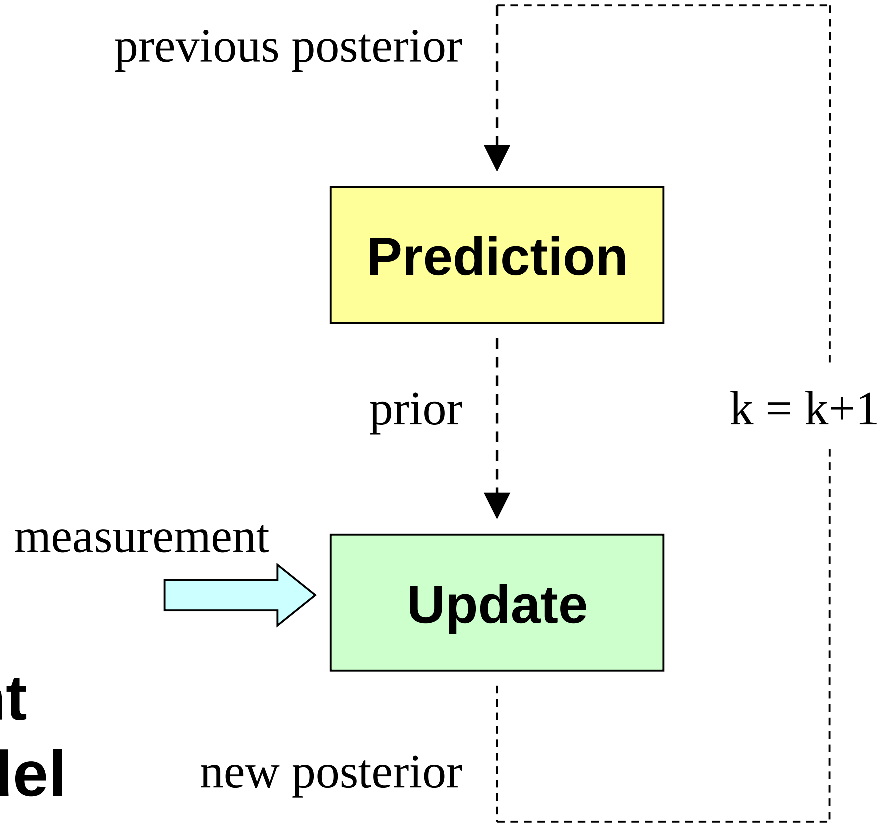
- Motion models
- Observation models
- Recursive Bayesian Estimation





Bayesian Estimation

- Recursive estimation with two main steps:
  - Prediction – to compute a prior probability with the **motion model**
  - Update – to compute a posterior probability using the **current measurement** and the **observation model**



Kalman Filter

UNIVERSITY OF LINCOLN

- In this case, the prediction-update cycle of the Kalman filter becomes a simple sequence of matrix operations as follows

Prediction

$$\hat{\mathbf{x}}_k = \mathbf{F}\,\mathbf{x}_{k-1}$$

$$\hat{\mathbf{P}}_k = \mathbf{F}\,\mathbf{P}_{k-1}\,\mathbf{F}^T + \mathbf{Q}$$

Update

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k)$$

$$\mathbf{P}_k = \hat{\mathbf{P}}_k - \mathbf{K}_k\,\mathbf{S}_k\,\mathbf{K}_k^T$$

- where

$$\hat{\mathbf{z}}_k = \mathbf{H}\hat{\mathbf{x}}_k \quad \text{and} \quad \mathbf{S}_k = \mathbf{H}\,\hat{\mathbf{P}}_k\,\mathbf{H}^T + \mathbf{R} \quad \text{and} \quad \mathbf{K}_k = \hat{\mathbf{P}}_k\,\mathbf{H}^T\,\mathbf{S}_k^{-1}$$

KF Prediction

UNIVERSITY OF LINCOLN

Prediction

$$\hat{\mathbf{x}}_k = \mathbf{F}\,\mathbf{x}_{k-1}$$

$$\hat{\mathbf{P}}_k = \mathbf{F}\,\mathbf{P}_{k-1}\,\mathbf{F}^T + \mathbf{Q}$$

- **x** = state vector (**^** means "predicted")
- **F** = matrix of the motion model
- **Q** = matrix of the motion noise
- **P** = covariance matrix of **x**

KF Prediction

- Example: Brownian motion

$$\begin{cases} x_k & = & x_{k-1} + n_{k-1}^x \\ y_k & = & y_{k-1} + n_{k-1}^y \end{cases}$$

- in this case

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \qquad \mathbf{F} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} \sigma_{nx}^2 & 0 \\ 0 & \sigma_{ny}^2 \end{bmatrix}$$

- **P** is computed from **F** and **Q**

KF Prediction

UNIVERSITY OF LINCOLN

- Example: Constant Velocity model

$$\begin{cases} x_k &= x_{k-1} + \dot{x}_{k-1}\Delta t_k + n^x_{k-1} \\ \dot{x}_k &= \dot{x}_{k-1} + n^{\dot{x}}_{k-1} \\ y_k &= y_{k-1} + \dot{y}_{k-1}\Delta t_k + n^y_{k-1} \\ \dot{y}_k &= \dot{y}_{k-1} + n^{\dot{y}}_{k-1} \end{cases}$$

- in this case $\mathbf{x}_k = ? \quad \mathbf{F} = ? \quad \mathbf{Q} = ?$

KF Update

UNIVERSITY OF LINCOLN

Update

$$\boxed{\begin{aligned}\mathbf{x}_k &= \hat{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k) \\ \mathbf{P}_k &= \hat{\mathbf{P}}_k - \mathbf{K}_k\,\mathbf{S}_k\,\mathbf{K}_k^T\end{aligned}}$$

$$\hat{\mathbf{z}}_k = \mathbf{H}\hat{\mathbf{x}}_k \quad \text{and} \quad \mathbf{S}_k = \mathbf{H}\,\hat{\mathbf{P}}_k\,\mathbf{H}^T + \mathbf{R} \quad \text{and} \quad \mathbf{K}_k = \hat{\mathbf{P}}_k\,\mathbf{H}^T\,\mathbf{S}_k^{-1}$$

- **x** = (updated) state vector
- **z** = observation vector ($\hat{\mathbf{z}}$ is the predicted one)
- **H** = matrix of the observation model
- **R** = matrix of the observation noise
- **S** = innovation matrix
- **K** = Kalman gain
- **P** = covariance matrix of **x**

KF Update

UNIVERSITY OF LINCOLN

- Example: Cartesian observations

$$\begin{cases} u_k = s \cdot x_k + n_k^u \\ v_k = s \cdot y_k + n_k^v \end{cases}$$

- where

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \quad \mathbf{z}_k = \begin{bmatrix} u_k \\ v_k \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} \sigma_{nu}^2 & 0 \\ 0 & \sigma_{nv}^2 \end{bmatrix}$$

- and **S**, **K**, **P** are computed as specified

KF Initialization and Iteration

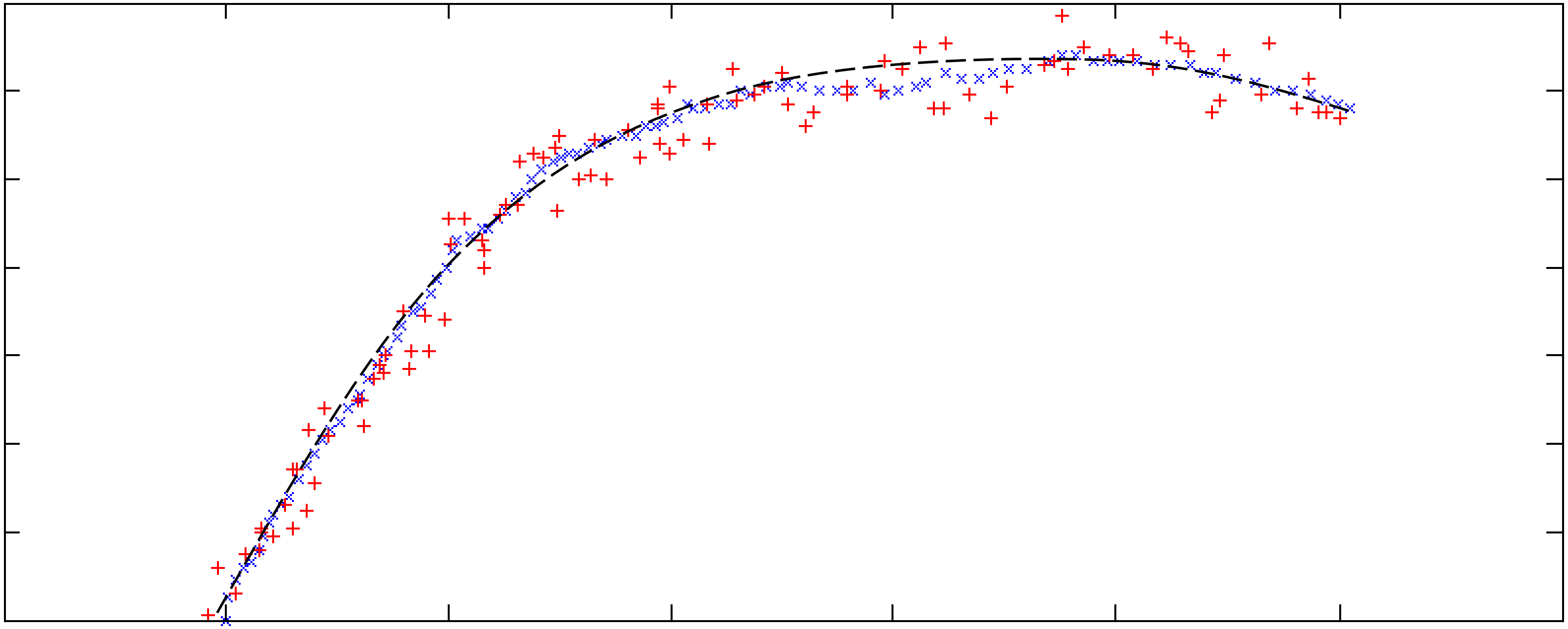
- Initial values of the state vector $\mathbf{x}_0$ and covariance matrix $\mathbf{P}_0$ must be known (e.g. $\mathbf{x}_0 = [0\ 0]^T$ and $\mathbf{P}_0 = \mathbf{Q}$)

1) Prediction: compute (predicted) $\mathbf{x}_1$ and $\mathbf{P}_1$

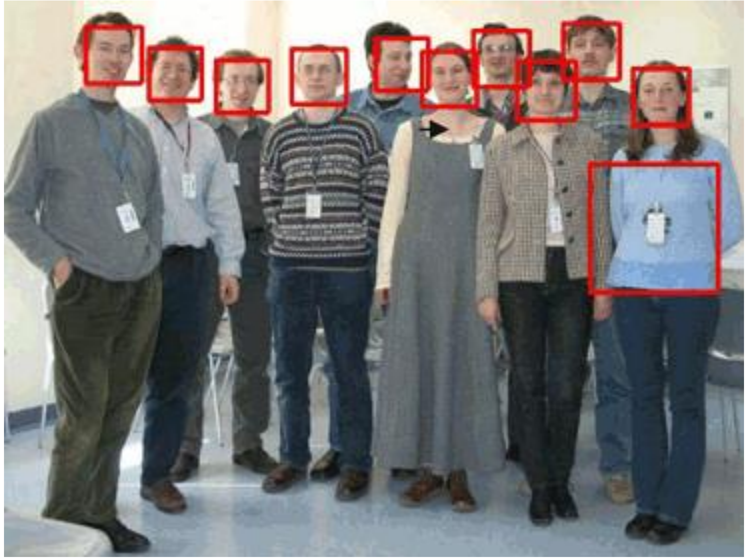
2) Update:
   - compute innovation **S** and gain **K**
   - update $\mathbf{x}_1$ and $\mathbf{P}_1$ using current $\mathbf{z}_1$

- Repeat from point 1)

E.g.: Trajectory Tracking



Multi-target Tracking



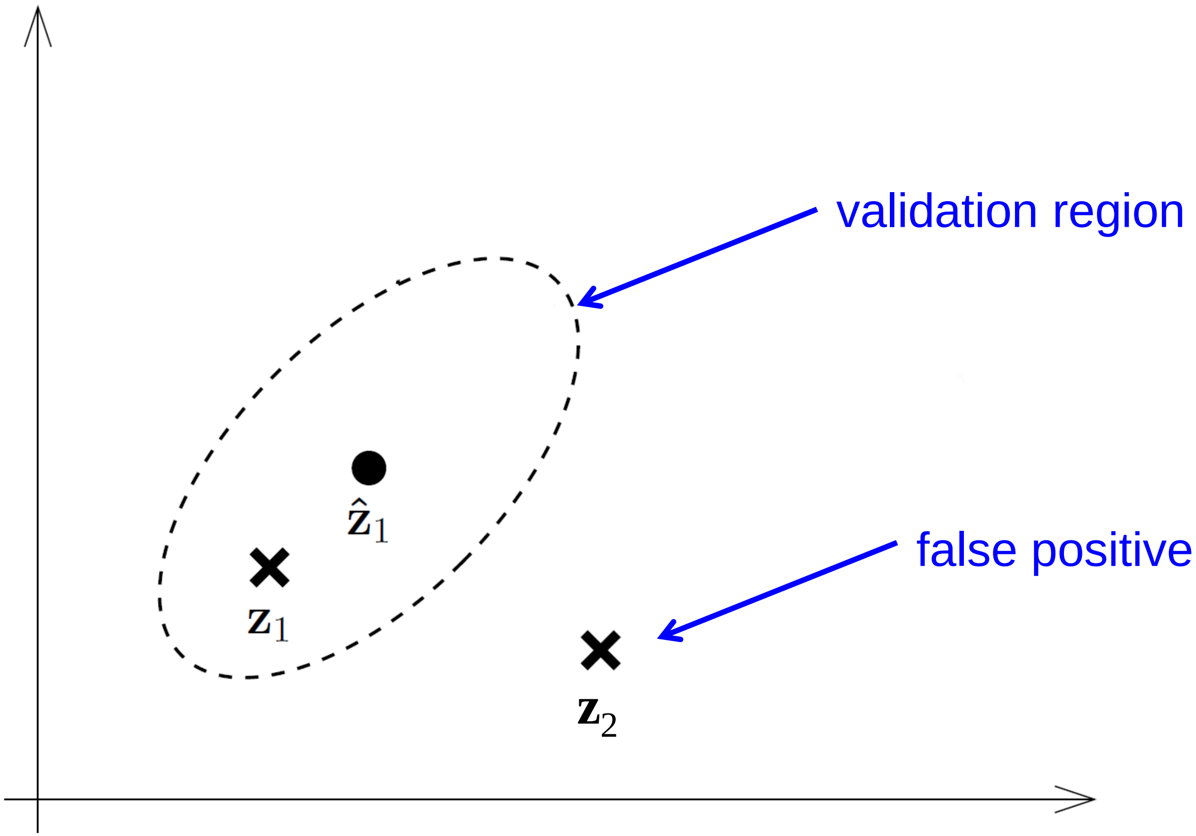
https://www.youtube.com/watch?v=3gJZr-DPzcI

Gating and Data Association

- What happen if there are observations which are completely wrong (i.e. false positives), maybe generated by a different target?
- Before updating the KF, it might be necessary to perform the following:
  - **Validation gating** – get rid of observations that could not have been generated by the target
  - **Data association** – if more than one target, associate the right observation to the right target

Example: Validation Gating

- The dot is the predicted observation, the cross is the real one from the sensor



Validation Gating

UNIVERSITY OF LINCOLN

- A simple way to determine the validation region is to consider the maximum distance that the target could have travelled, since the previous time step, if moving at maximum speed

- A better solution is to compute the (squared) **Mahalanobis distance** for observation i and target j, and check if it is less than a threshold:

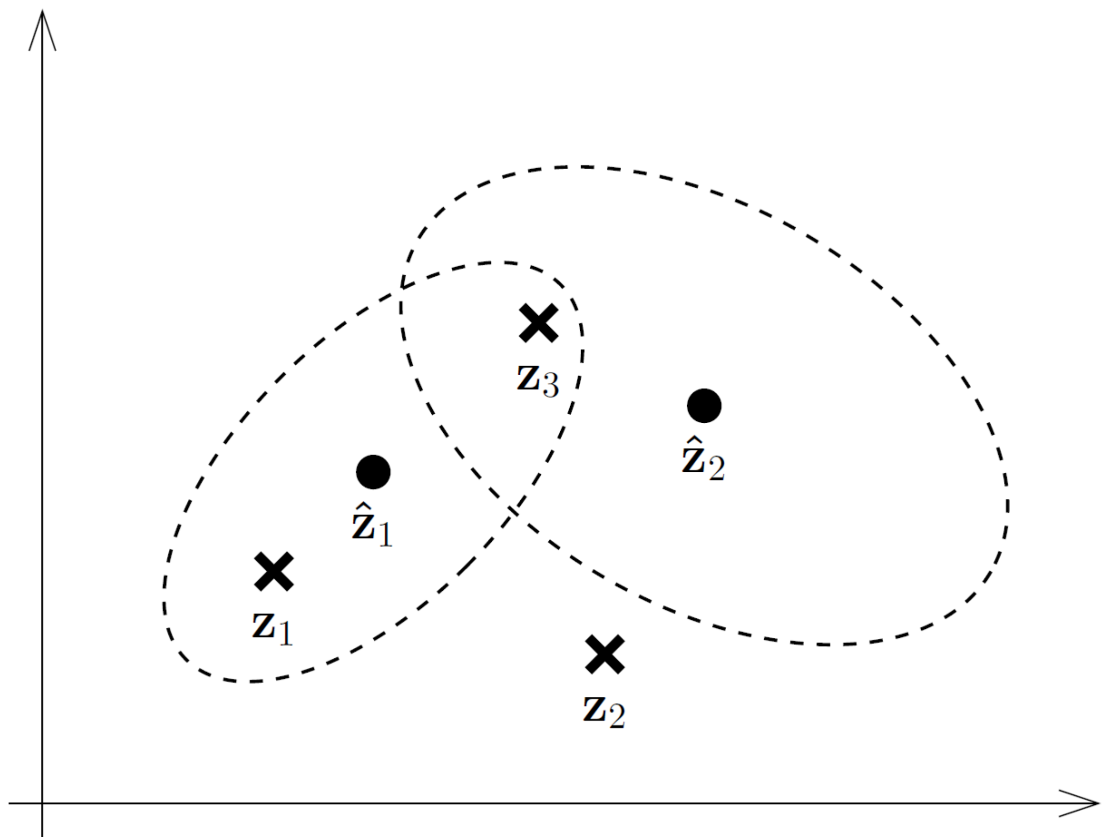
$$\boxed{(\mathbf{z}_i - \hat{\mathbf{z}}_j)^T \, \mathbf{S}_{ij}^{-1} \, (\mathbf{z}_i - \hat{\mathbf{z}}_j) \leqslant \lambda}$$

(λ = 9.21 for 2D observations)

- If the inequality does not hold, then the observation is considered a false positive and not used to update the KF

Example: Data Association

- The dots are the predicted observations for 2 separate targets (i.e. 2 separate KFs)



NN Data Association

- Nearest Neighbour (NN) data association is one of the simplest methods to associate multiple observations to multiple targets

- The squared Mahalanobis distance $\left(\mathbf{z}_i - \hat{\mathbf{z}}_j\right)^T \mathbf{S}_{ij}^{-1} \left(\mathbf{z}_i - \hat{\mathbf{z}}_j\right)$ is computed for all the possible combinations of observation i and target j

- The combination (i, j) for which the Mahalanobis distance is minimum will be the chosen association (i.e. filter *i* will be updated using observation j)

Conclusions

- Suggested reading
  - Online material

- Next lecture
  - Please check email and BB for updates

- Any question?
  - nbellotto@lincoln.ac.uk