UNIVERSITY OF LINCOLN

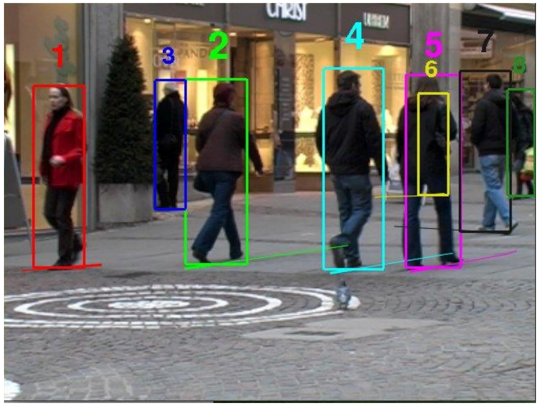
CMP9135M - Computer Vision

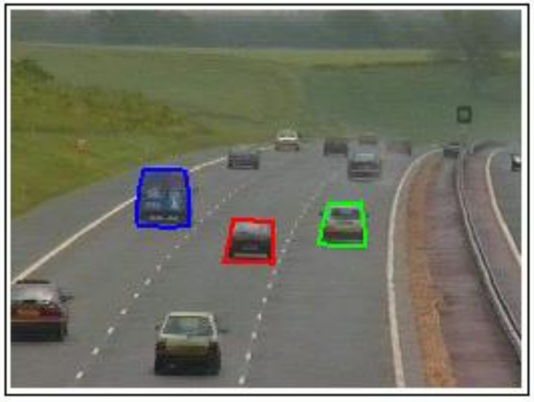
Visual Tracking I

Dr Nicola Bellotto

Overview of Target Tracking

- Motion models
- Observation models
- Recursive Bayesian Estimation





Target Tracking

UNIVERSITY OF LINCOLN

- Tracking is the problem of estimating the position a target moving in space

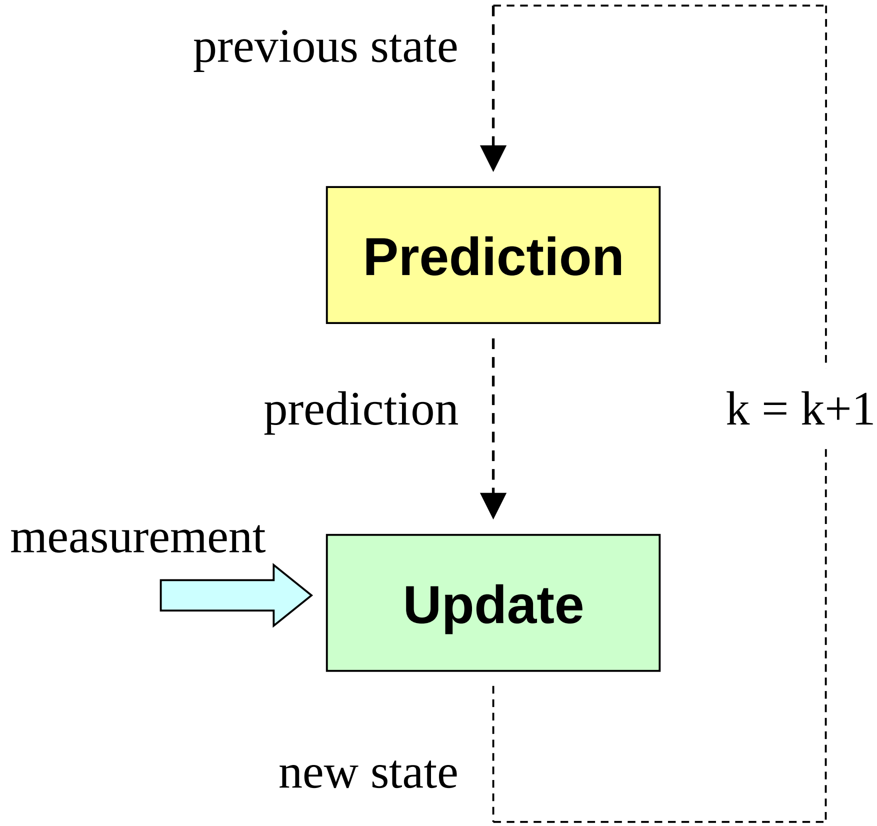
Bayesian Target Tracking

UNIVERSITY OF LINCOLN

- Bayesian Target Tracking systems are probabilistic solutions that take into account variations of the target's motion and errors of the observer
- Important elements are motion models, observation models and Bayesian filters

Bayesian Target Tracking

- Recursive estimation of target state:
  - Prediction – based on known **motion model**
  - Update – based on actual and expected **measurement** of **observation model**



Motion models: assumptions

- We consider targets moving in a 2D space (i.e. on a horizontal plane)
- Also, we use a Cartesian representation, i.e. based on (x,y) coordinates
- We are interested in discrete time systems (instead of continuous time), where quantities are expressed at regular time intervals t_k , t_{k+1} , t_{k+2} , …

Brownian Motion

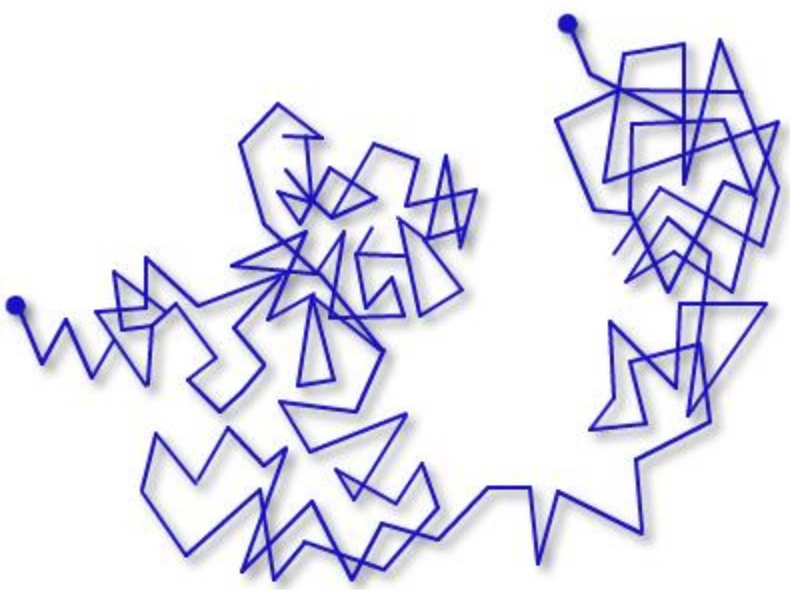
- One of the simplest models for target tracking (also called discrete Wiener process)

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

- Where *n* are zero-mean Gaussian noises

Brownian Motion

- Example of trajectory for a target moving according to Brownian motion



Brownian MM: pros & cons

- This model does not consider velocity and orientation of the target
- In many applications, it is a good approximation of human walking
- However, it requires very frequent and relatively precise observations
- Not good in case of occlusions (i.e. target temporarily behind another object)

Constant Velocity

- This model is derived from a continuous curvilinear-motion model

$$\begin{cases} \dot{x}(t) &= v(t)\cos\phi(t) \\ \dot{y}(t) &= v(t)\sin\phi(t) \\ \dot{v}(t) &= a_t(t) \\ \dot{\phi}(t) &= \dfrac{a_n(t)}{v(t)} \end{cases}$$

Constant Velocity

- Assuming all the accelerations are just "noise", the equivalent discrete-time version of the previous model is as follows

$$\begin{cases} x_k &= x_{k-1} + v_{k-1}\Delta t_k \cos\phi_{k-1} + n^x_{k-1} \\ y_k &= y_{k-1} + v_{k-1}\Delta t_k \sin\phi_{k-1} + n^y_{k-1} \\ v_k &= v_{k-1} + n^v_{k-1} \\ \phi_k &= \phi_{k-1} + n^\phi_{k-1} \end{cases}$$

where $\Delta t_k = t_k - t_{k-1}$

Constant Velocity

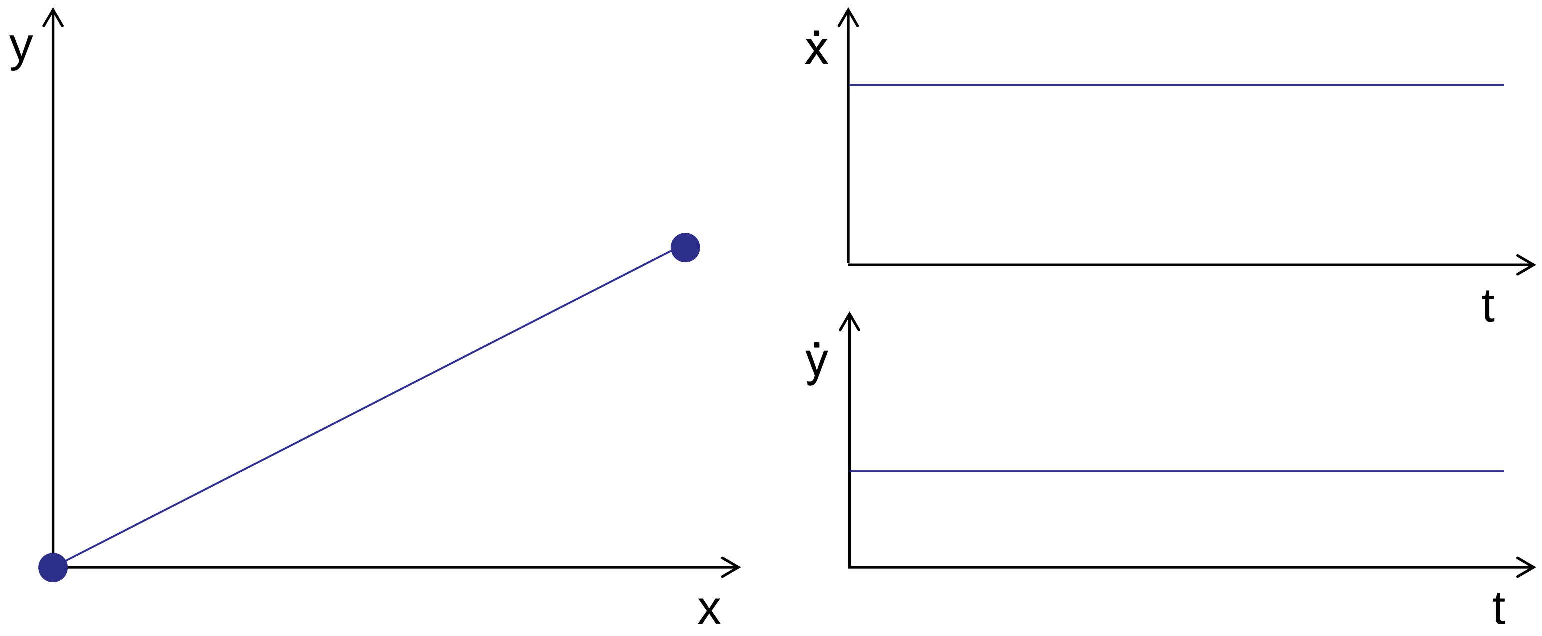
UNIVERSITY OF LINCOLN

- More common and simpler representation of the CV model used for 2D tracking:

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

Constant Velocity

- Example of trajectory for a target moving according to CV model (without noise)



CV MM: pros & cons

UNIVERSITY OF LINCOLN

- This is a popular model for nearly-constant (piecewise) rectilinear motions
- Thanks to the velocity components, it can deal with small occlusions
- It is used also for non-rectilinear trajectories, as long as the latter can be locally approximated to rectilinear ones

Constant Acceleration & Constant Turn-rate Models

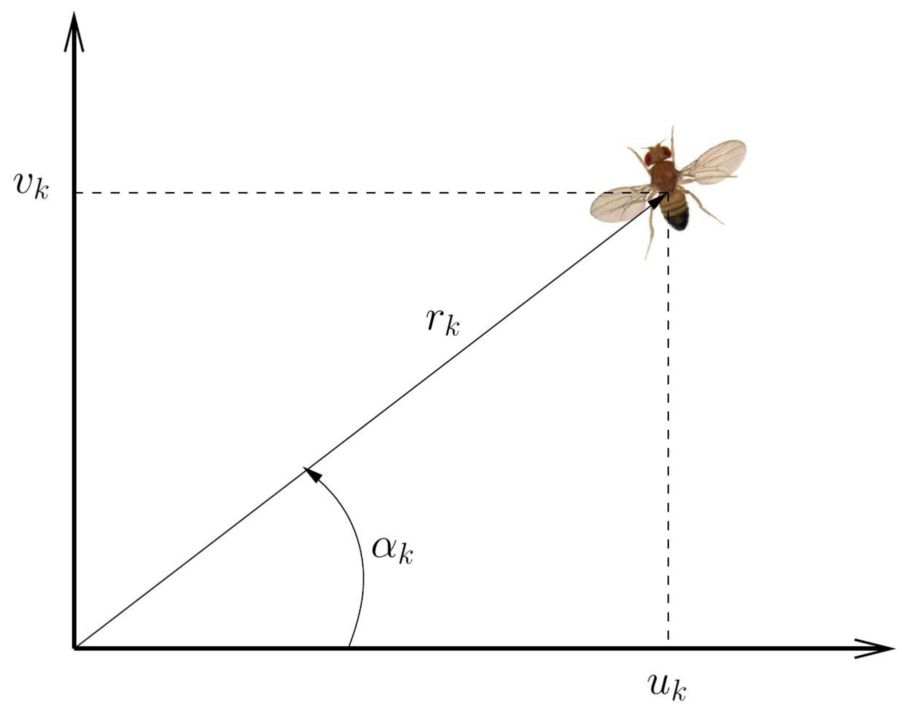
- In case of significant accelerations, the motion could be better described by a Constant Acceleration (CA) or a Constant Turn-rate (CT) model
- They can deal with complex curvilinear trajectories, but they are more complicated
- Examples of applications include aircraft tracking, military applications, etc.

Observation Models

- To be tracked, targets are observed by sensors affected by noise
- Sensing algorithms (e.g. image segmentation) also introduce errors
- A good observation model takes into account these problems
- While motion models depends on targets, observation models depends on sensors

Observation Models

- Example of coordinates in a Cartesian and Polar observation model



Observation Models

- In 2D, the simplest observation model is a Cartesian one where the (*x*,*y*) coordinates are directly measured, up to a scale factor

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

- Here (*u*,*v*) are the measured quantities (e.g. image pixels) and *s* is the scale factor

Observation Models

UNIVERSITY OF LINCOLN

- Other sensors are better modelled using Polar rather Cartesian coordinates

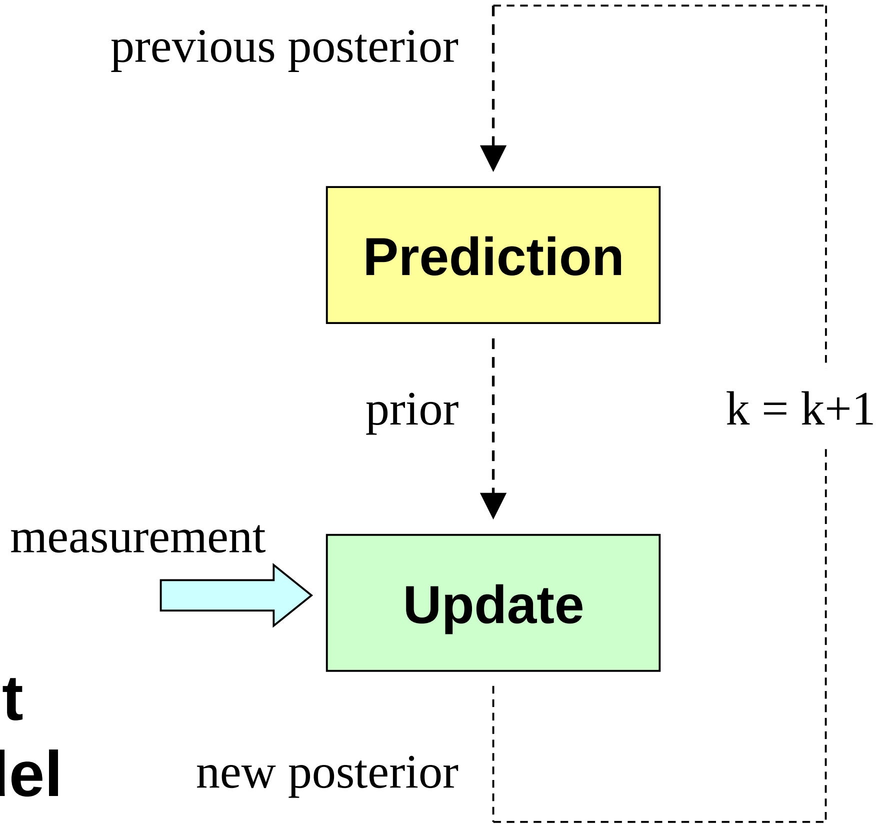
$$\begin{cases} r_k = \sqrt{x_k^2 + y_k^2} + n_k^r \\ \alpha_k = \tan^{-1}\left(\dfrac{y_k}{x_k}\right) + n_k^\alpha \end{cases}$$

Recursive Bayesian Estimation

- Recursive Bayesian estimators are the most popular techniques used for target tracking (not only in computer vision)
- The majority of them belong to two classes: **Kalman filters** and **Particle filters**
- The core principle is the same:
  - first, given the target's position at time t_{k-1}, "predict" where it should be at time t_k
  - then, "update" (or "correct") the prediction based on actual observations at time t_k

Bayesian Estimation

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



Assumptions

UNIVERSITY OF LINCOLN

- Given the target state $\boldsymbol{x_k}$ and the set of observations $\boldsymbol{Z_k = \{z_1, \ldots, z_k\}}$, the probability distribution of $\boldsymbol{x_k}$ given $\boldsymbol{Z_k}$ is called ***posterior*** and written as $\boldsymbol{p(x_k|Z_k)}$
- The posterior can be computed using the Bayes rule and the following two assumptions:

  - The current target state $\boldsymbol{x_k}$ depends only on the previous $\boldsymbol{x_{k-1}}$ (*1st order Markov assumption*)
  - The observation $\boldsymbol{z_k}$ depends only on the current state $\boldsymbol{x_k}$ (*sensor Markov assumption*)

Bayesian Estimation

- Prediction: compute the prior
- Update: using the measurement $\boldsymbol{z_k}$
  - compute the likelihood
  - normalize
  - obtain the posterior

$$p(x_k \mid Z_k) = \alpha \; p(z_k \mid x_k) \, p(x_k \mid Z_{k-1})$$

Bayesian Filters

- In case of linear models (e.g. Brownian and CV) where all the noises are independent and (zero-mean) normally distributed (Gaussian noise), the Kalman Filter can be used for Bayesian estimation
- If the models are not linear (e.g. Polar observation) an Extended Kalman Filter or an Unscented Kalman Filter are usually more suitable
- If the noises are not normally distributed, a Particle Filter is often the best choice

Bayesian Filters

- Comparison of filter choices

Model / Noise	Linear	Non-linear
Gaussian	**Kalman filter**	**EKF** or **UKF**
Non-Gaussian	**Particle filter**	**Particle Filter**

Conclusions

UNIVERSITY OF LINCOLN

- Suggested reading
  - Online material
- Next week
  - Visual Tracking II (Kalman filter-based)
- Any question?