Human-Oriented Robotics

Temporal Reasoning

Part 3/3

Kai Arras

Social Robotics Lab, University of Freiburg

Contents

Introduction

- **Detection** is knowing the **presence** of an object, possibly with some attribute information
- **Tracking** is estimating the **state of a moving object** over time based on remote measurements
- **Tracking** also involves maintaining the **identity of an object** over time despite detection errors (FN, FP) and the presence of other objects
- **Tracking** may involve estimating the state of **several objects** at a time. This gives rise to **origin uncertainty**, that is, uncertainty about which object generated which observation
- **Data association** addresses the origin uncertainty problem. It's the process of associating uncertain measurements to known tracks
- **Data association** may involve **interpreting** measurements as new tracks, false alarms or misdetections and tracks as occluded or terminated

Introduction

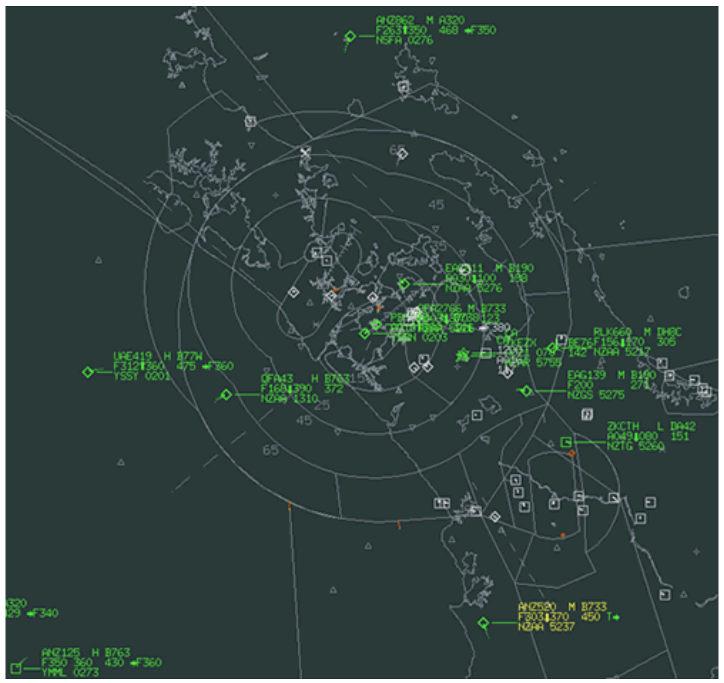
- Imagine watching a **rare exotic bird flying through dense jungle foliage**

- You can only glimpse **brief**, intermittent **flashes of motion**
- **Occlusion** from foliage and trees makes it hard to guess where the bird is and where it will appear next
- There are **many birds**, they may even look alike
- It is hard to differentiate between **bird and background**

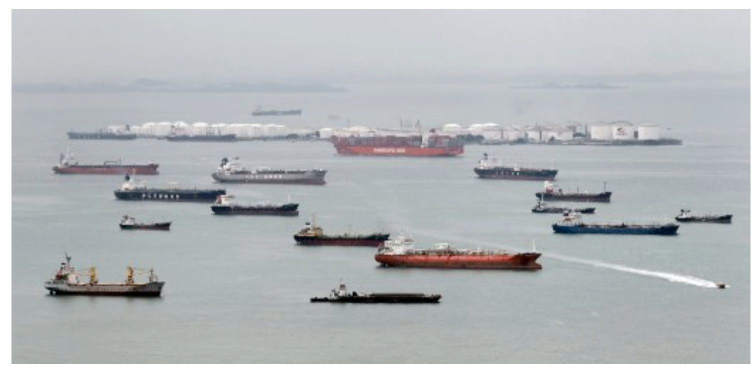
Example from [2]

Introduction: Applications

fleet management

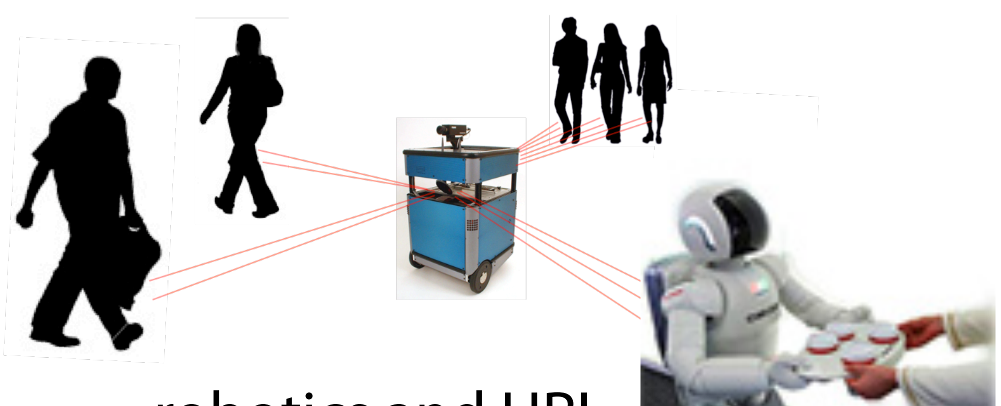


air traffic control

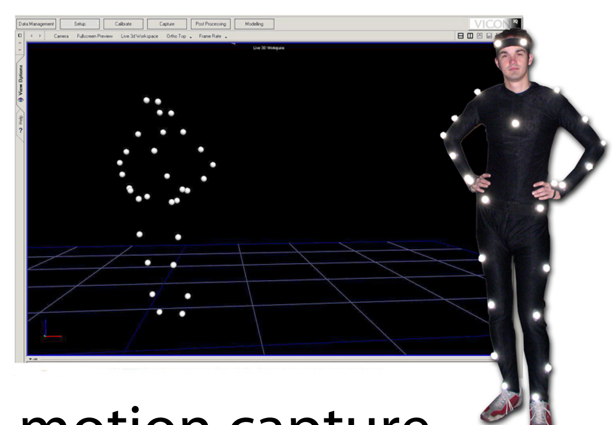


maritime surveillance
and port traffic control

surveillance



robotics and HRI

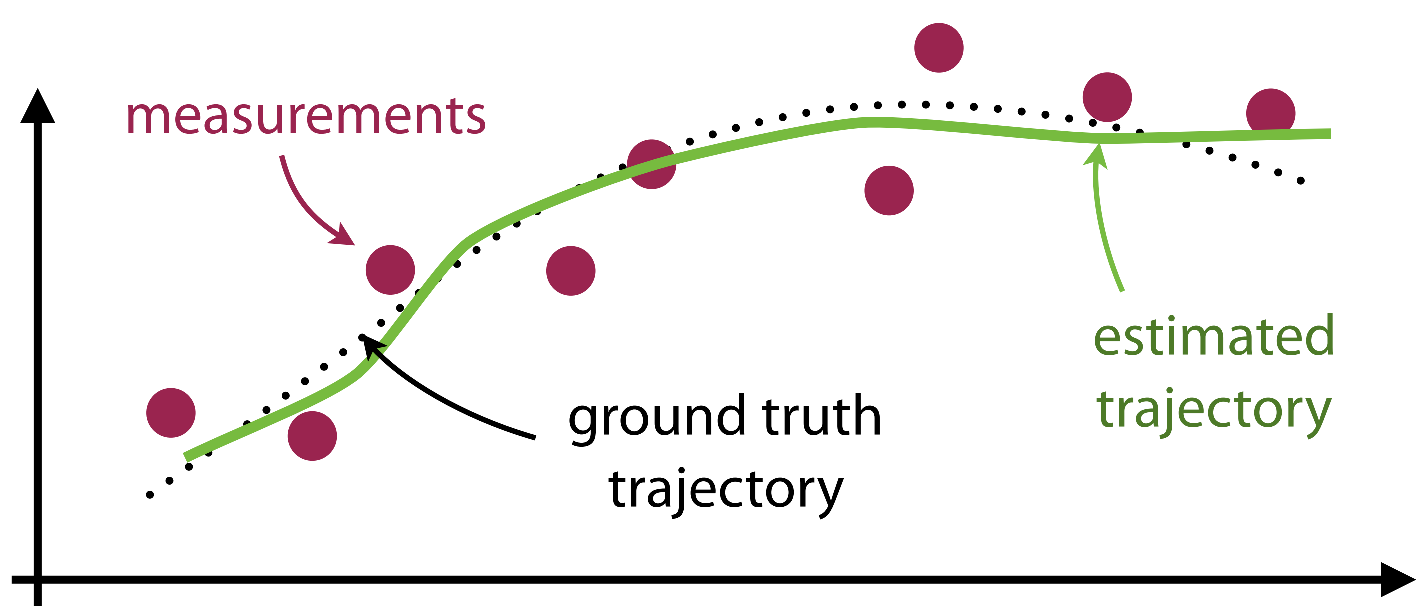


motion capture

military
applications

Introduction

- **Problem Statement of Tracking:**
  - Given an LDS model with parameters transition model, observation model and prior, we want to compute state estimates in a way that their **accuracy is higher** than the raw measurements and that they contain **information not available** in the measurements (e.g. identity, velocity, or accelerations)



Introduction

- **Error Types**
  - Uncertainty in the **values of measurements** ("noise").
    Solution: **filtering**
  - Uncertainty in the **origin of measurements** due to false alarms, multiple targets, or decoys and countermeasures.
    Solution: **data association**

Tracking = Data association + Filtering

Introduction: Problem Types

- **Track stage** (track "life cycle")
  - Track formation (initialization)
  - Track maintenance (continuation)
  - Track termination (deletion)
- **Number of sensors**
  - Single sensor
  - Multiple sensors
- **Sensor characteristics**
  - Detection probability P_D (true positive rate)
  - False alarm rate P_F (false positive rate)
- **Target behavior**
  - Non-maneuvering (straight or quasi-straight motion)
  - Maneuvering (makes turns, stops, etc.)
- **Number of targets**
  - Single target
  - Multiple targets
- **Target size**
  - Point-like target
  - Extended target
  - Groups of targets

Introduction: Track Stage

Formation

- When to create a new track?
- What is the initial state?

- **Greedy** initialization heuristics
  - Every observation that cannot be associated is a new track
  - Initialize position from observation, heuristics for derivatives e.g. velocity
- **Lazy** initialization
  - Wait and look for sequences of unassociated observations
  - Initialize position and higher order derivatives from sequence

Occlusion vs. Deletion

- When to delete a track?
- Or is it just occluded?

- **Greedy** deletion heuristics
  - Delete track as soon as no observation can be associated to it
  - No occlusion handling
- **Lazy** deletion
  - Delete if no observation can be associated for several time steps
  - Implicit occlusion handling

Introduction: Tracking Algorithms

- **Single non-maneuvering target, no origin uncertainty**
  - Kalman filter (KF) or extended Kalman filter (EKF)
- **Single maneuvering target, no origin uncertainty**
  - KF/EKF with variable process noise
  - Multiple model approaches (MM)
- **Single non-maneuvering target, origin uncertainty**
  - KF/EKF with nearest/strongest neighbor data association
  - Probabilistic data association filter (PDAF)
- **Single maneuvering target, origin uncertainty**
  - Multiple model-PDAF (MM-PDAF)

Introduction: Tracking Algorithms

- **Multiple non-maneuvering targets**
  - Joint probabilistic data association filter (JPDAF)
  - Multiple hypothesis tracker (MHT)
  - Markov chain Monte Carlo data association (MCMCDA)
- **Multiple maneuvering targets**
  - MM-variants of MHT (e.g. IMMMHT)
  - MM-variants of other data association techniques

- Other Bayesian filtering schemes such as **particle filters** have also been successfully applied to the tracking problem. They are currently not covered here. See references.

Validation Gate

- We have already seen the **statistical compatibility test** in the Kalman filter cycle:

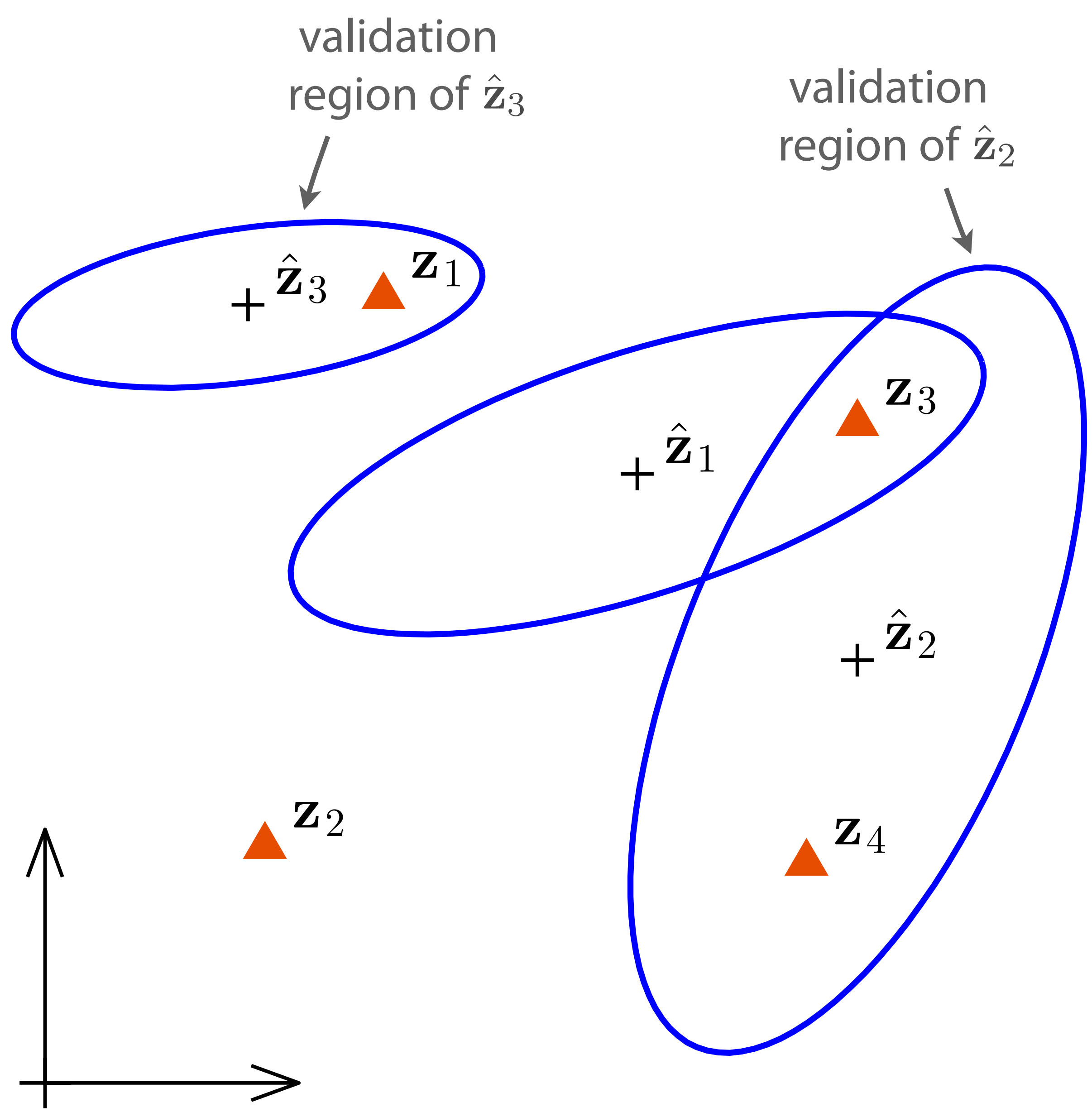
  1. Predict measurement based on the predicted track state.
     This gives an area in sensor coordinates where to expect the next observation.
  2. Make observations.
     Observations may be raw sensory data or the output of a target detector
  3. Check if the actual measurement lies close to the predicted measurement in terms of the squared Mahalanobis distance. If the distance is smaller than a threshold from a cumulative χ^2 distribution, then they form a **pairing** or **match**

- The area around the predicted measurement in which pairings are accepted is called **validation gate** or **validation region**
- This procedure is also called validation gaiting or simply **gaiting**
- Let us take a closer look at the validation gate

Validation Gate

What makes this a difficult problem:

- **Multiple targets**.
  May lead to **association ambiguity** when several measurements are in the gate
- **False alarms** (false positives)
- **Detection uncertainty**, occlusions, misdetections (false negatives)



Validation Gate

- The validation test implies that measurements $\mathbf{z}(k+1)$ are distributed according to a Gaussian distribution, centered at the measurement prediction $\hat{\mathbf{z}}(k+1)$ with covariance $S(k+1)$. Skipping time indices,

$$p(\mathbf{z}) = \mathcal{N}_{\mathbf{z}}(\hat{\mathbf{z}}, S)$$

- This assumption is called **measurement likelihood model**
- Then, with $d^2 = \nu^T S^{-1} \nu$ being the squared Mahalanobis distance of a pairing, measurements will be in the area

$$\mathcal{V}(\gamma) = \{\mathbf{z} : d^2 \leq \gamma\}$$

  with a probability defined by the gate threshold γ
- This area is the **validation gate**

Validation Gate

- The **shape** of the validation gate is a **hyperellipsoid**
- This follows from the measurement likelihood model set to

$$c = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}\nu^T S^{-1} \nu\right\}$$

leading to $c' = \nu^T S^{-1} \nu$ which describes a **conic section** in matrix form

$$\mathbf{x}^T \mathbf{Q}\, \mathbf{x} = 0 \qquad \mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \mathbf{Q} = \begin{pmatrix} A & B & D \\ B & C & E \\ D & E & F \end{pmatrix}$$

- The validation gate is an **iso-probability contour** obtained when intersecting a Gaussian with a hyperplane

Validation Gate

- Why a χ^2 **distribution**?
- We remember that if several x's form a set of k i.i.d. standard normally distributed random variables

$$x_i \sim \mathcal{N}_{x_i}(0,1)$$

- Then, variable q with

$$q = \sum_{i=1}^{k} x_i^2$$

  follows a χ^2 distribution with k degrees of freedom
- We will now show that the **Mahalanobis distance** is a sum of squared standard normally distributed random variables

Validation Gate

- Assume **1-dimensional observations** and $\mu = \hat{z}(k+1), \ \sigma^2 = S(k+1)$
- The 1-dimensional Mahalanobis distance is then

$$d^2 = (z-\mu)^T \ \left(\sigma^2\right)^{-1} \ (z-\mu) = \frac{(z-\mu)^2}{\sigma^2}$$

- By changing variables $y = \dfrac{(z-\mu)}{\sigma}$, we have

$$y \sim \mathcal{N}_y(0,1)$$

- Thus, $d^2 = y^2$ and is χ^2 **distributed** with 1 degree of freedom

Validation Gate

- Assume n-**dimensional observations** and $\boldsymbol{\mu} = \hat{\mathbf{z}}(k+1), \ \Sigma = S(k+1)$
- The n-dimensional Mahalanobis distance is then

$$d^2 = (\mathbf{z} - \boldsymbol{\mu})^T \, \Sigma^{-1} \, (\mathbf{z} - \boldsymbol{\mu})$$

- By changing variables $\mathbf{y} = C^{-1} \, (\mathbf{z} - \boldsymbol{\mu})$ with $\Sigma = C \cdot C^T$, we have $\mathbf{y} \sim \mathcal{N}_{\mathbf{y}}(0, \mathbf{I})$ and therefore

$$d^2 = \mathbf{y}^T \, \mathbf{I} \, \mathbf{y} \quad \Rightarrow \quad d^2 = \sum_{i=1}^{k} y_i^2$$

  which is χ^2 **distributed** with k degrees of freedom
- C is obtained from a Cholesky decomposition

Validation Gate

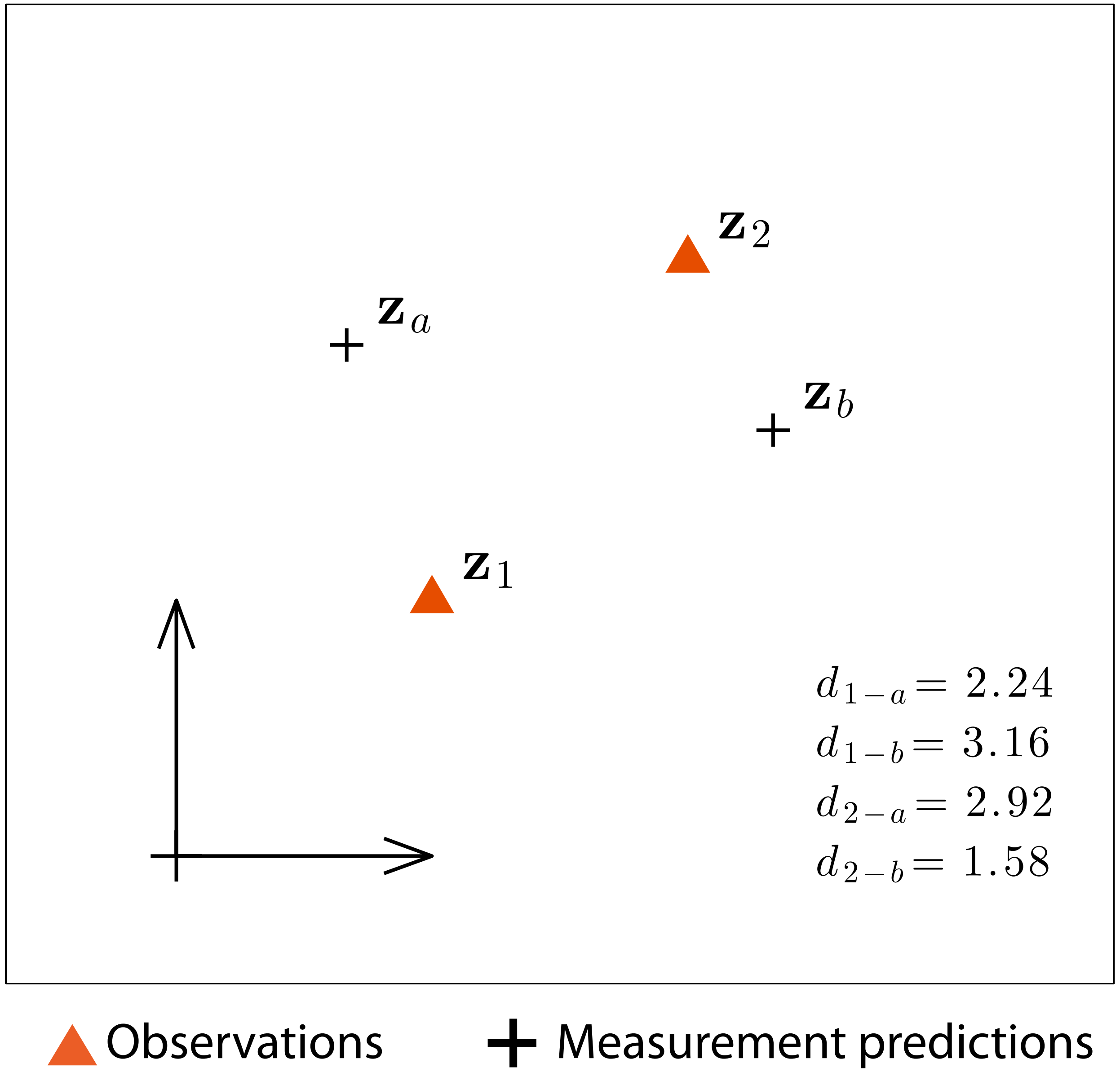
- Where does the **threshold** γ come from?
- γ is typically written as $\chi^2_{k,\alpha}$. The value is taken from the inverse χ^2 cumulative distribution at a level α and k degrees of freedom
- The values are typically given in tables, e.g. in statistics textbooks or by the Matlab function `chi2inv`
- Given the level α, we can now understand the interpretation of the validation gate

> The validation gate is a **region of acceptance** such that $100(1-\alpha)\%$ of **true measurements** are **rejected**

- Typical values for α are 0.95 or 0.99

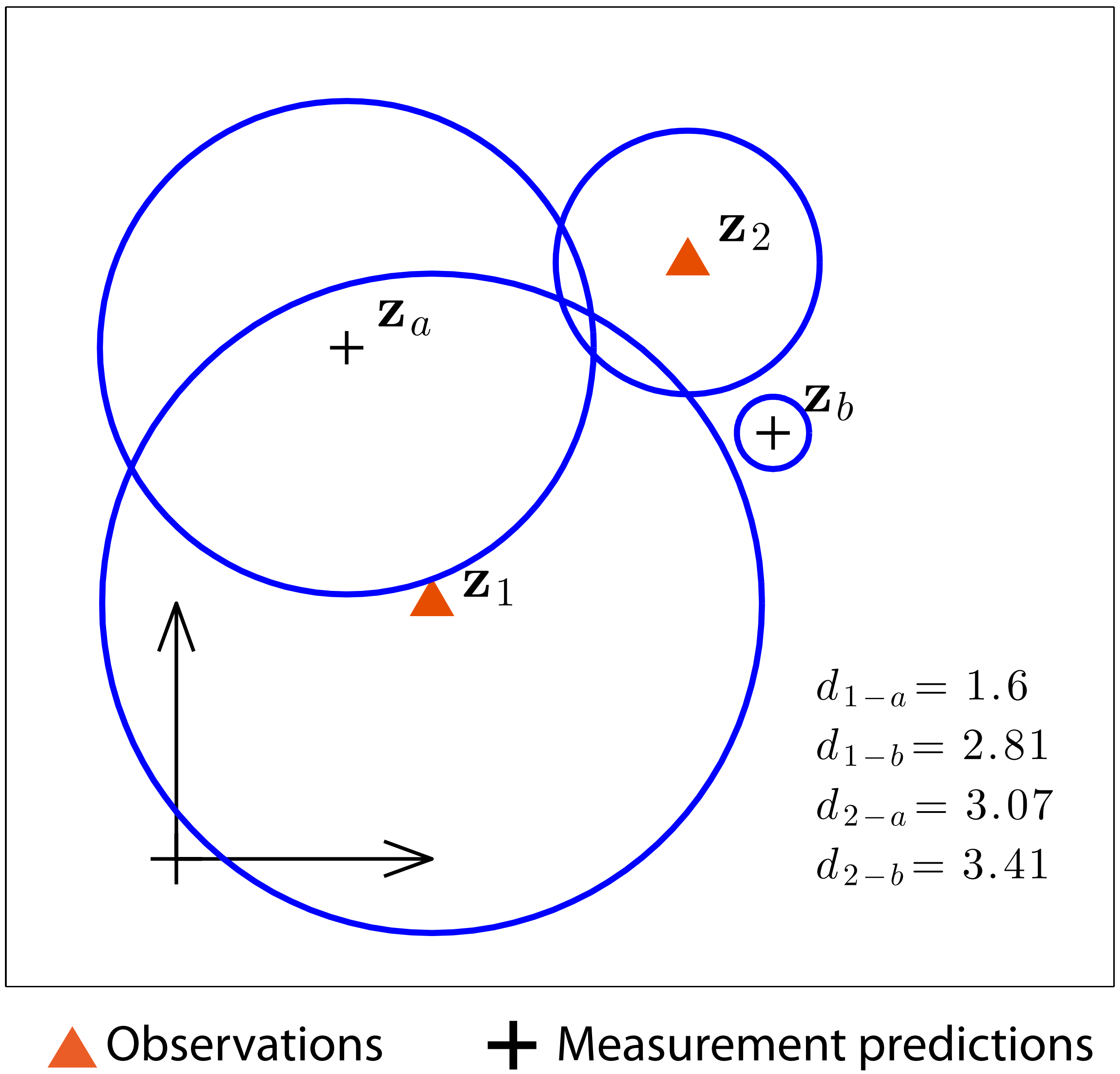
Validation Gate

- How does the Mahalanobis distance look **geometrically**?
- **Euclidian distance** accounts for
  - ☑ Position
  - ☐ Uncertainty
  - ☐ Correlations
- It seems that 1–a and 2–b belong together



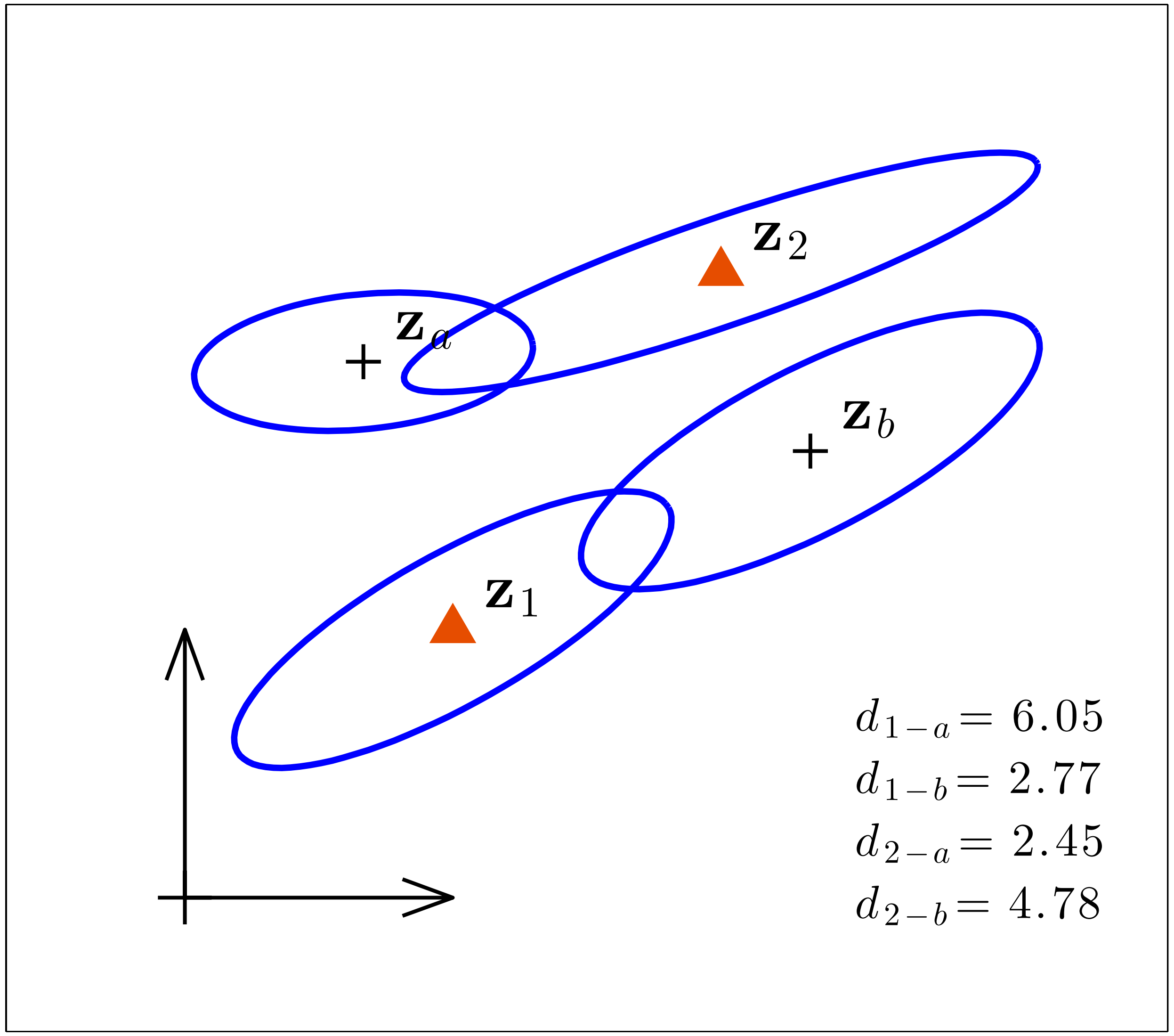
Validation Gate

- How does the Mahalanobis distance look **geometrically**?
- **Mahalanobis distance** with **spherical covariance matrices** accounts for
  - ☑ Position
  - ☑ Uncertainty
  - ☐ Correlations
- Now 2–b is furthest away. It seems that 1–a belong together, situation for 2 and b unclear



Validation Gate

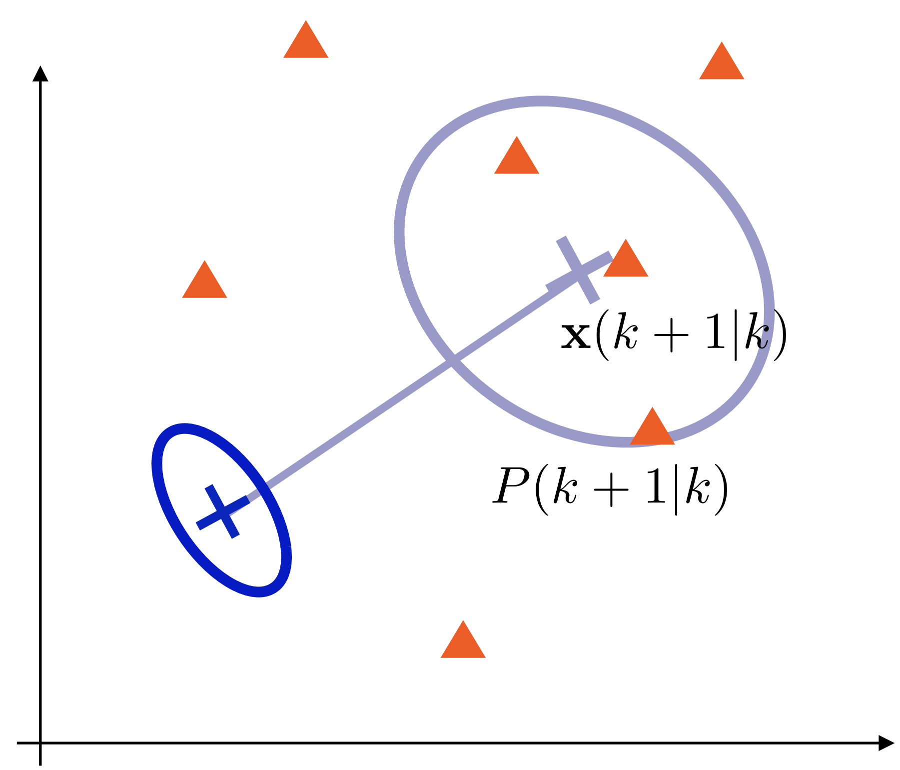
- How does the Mahalanobis distance look **geometrically**?
- **Mahalanobis distance** accounts for
  - ☑ Position
  - ☑ Uncertainty
  - ☑ Correlations
- It's actually 2–a and 1–b that belong together!
- Mahalanobis distance can be seen as a **generalization** of the Euclidian distance



▲ Observations   + Measurement predictions

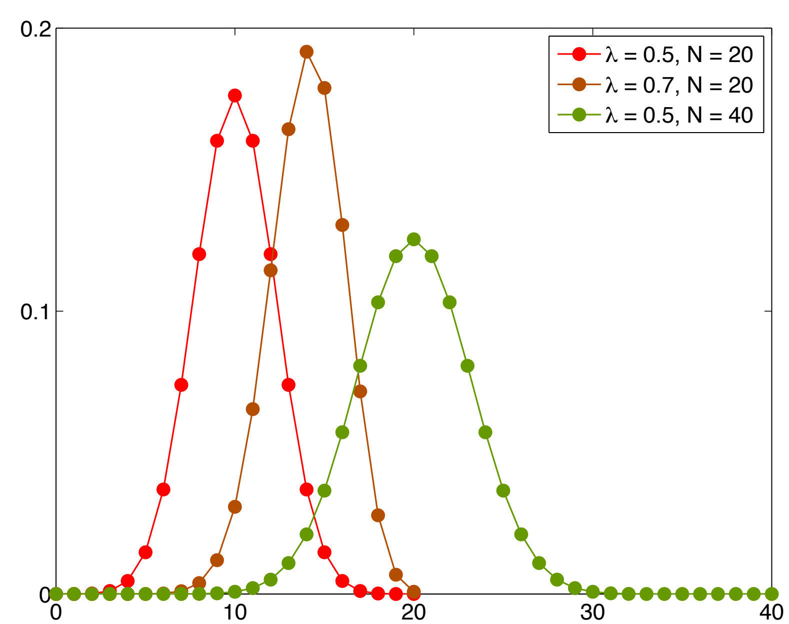
False Alarm Model

- False alarms (a.k.a. false positives, false detections) may come from sensor imperfections, detector failures, or clutter
- Clutter is unwanted echoes such as atmospheric turbulences. Originates from the "classical" radar tracking domain
- So, what's **inside the gate**
  - A measurement from the tracked object?
  - A false alarm?
- How to **model false alarms**?
  - Uniform over the sensor field of view
  - Independent across time



False Alarm Model

- Assume (temporarily) that the **sensor field of view** V is discretized into N discrete cells c_i , i = 1,...,N (like pixels)
- In each cell, false alarms occur with probability P_F
- Assume independence of false alarm events across cells
- The occurrence of false alarms is a Bernoulli process with probability of success P_F (flipping an unfair coin)
- Then, the **number of false alarms** m_F per time step follows a **binomial distribution**



$$p(m_F) = \binom{N}{m_F} P_F^{m_F} (1 - P_F)^{N - m_F}$$

with expected value $\mathrm{E}[m_F] = N \cdot P_F$

False Alarm Model

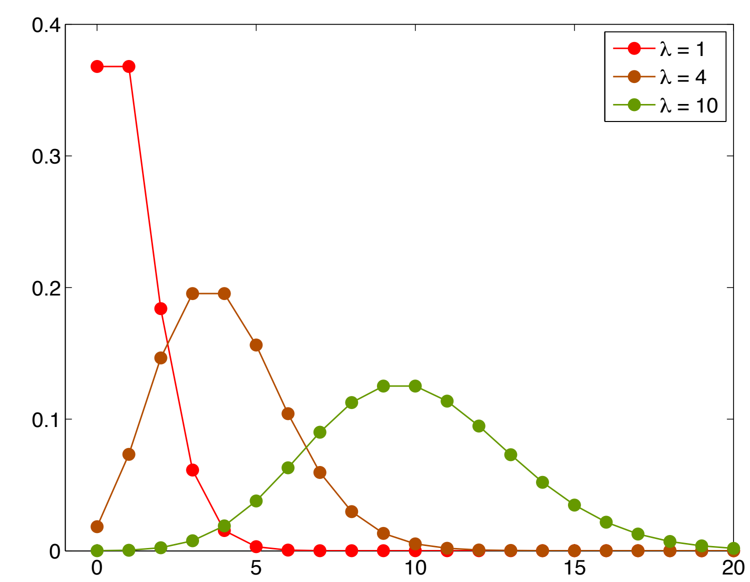
- Let the **spatial density** λ be the number of false alarms over space

$$\lambda = \frac{\mathrm{E}[m_F]}{V} = \frac{N \cdot P_F}{V} \qquad \text{[occurences per m}^2\text{]}$$

- If $P_F \ll 1$ and $N \rightarrow \infty$, that is, we reduce the cell size and approach the continuous case, then the above Binomial becomes a **Poisson distribution**

$$\mu(m_F) = e^{-\lambda V} \frac{(\lambda V)^{m_F}}{m_F!}$$

- This is the probability mass function of the **number of false alarms** in the volume V in terms of the spatial density λ



False Alarm Model

- The **spatial distribution** of false alarm is, based on the same assumptions, uniform over the sensor field of view V
- Thus, the density of the location of of a false alarm is

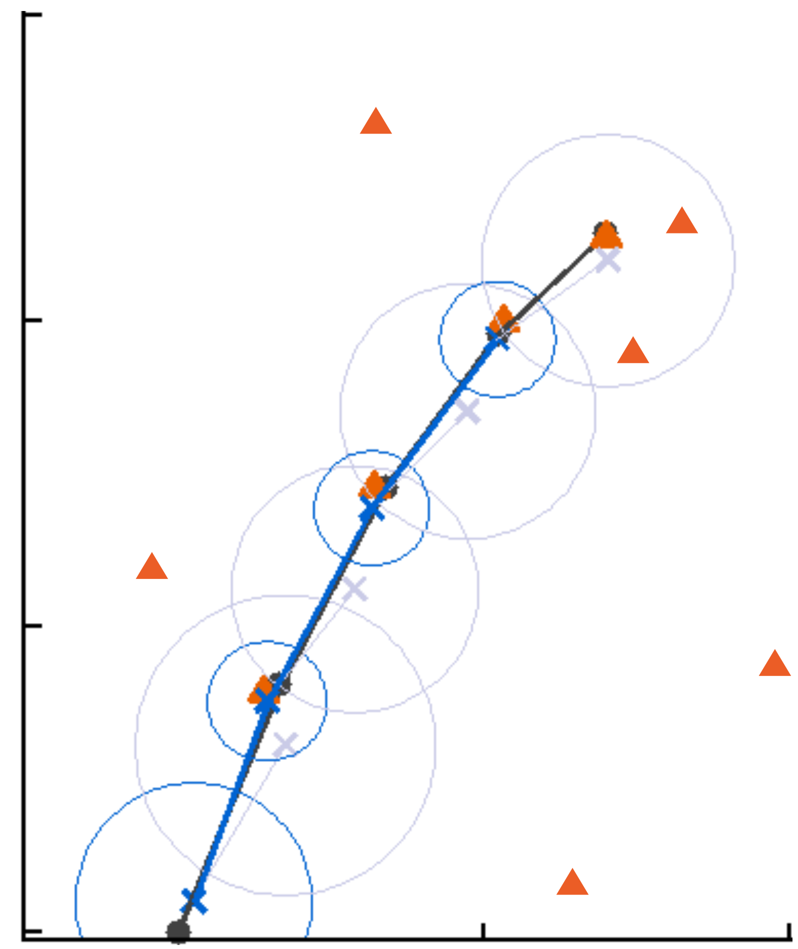
$$p(z \mid z \text{ is a false alarm}) = \frac{1}{V}$$

- In practice, this distribution may be **non-uniform** when P_F, and consequently λ, vary over space (e.g. detector performance varies in front of different backgrounds)
- **Persistent sources** of false alarms or clutter may also exist (e.g. from reflections, emitters, or background objects with target-like appearance)
- One approach is to **learn a background model**

Single-Target Data Association

Assumptions

- A **single** target to track
- Track already initialized
- Detection probability $P_D < 1$
- False alarm probability $P_F > 0$



Two groups of approaches

- **Non-Bayesian**: no association probabilities
  - Nearest neighbor standard filter (NNSF)
  - Strongest neighbor standard filter (SNSF)
  - Track splitting filter
- **Bayesian**: computes association probabilities
  - Probabilistic data association filter (PDAF)

Nearest Neighbor Standard Filter (NNSF)

- In each step

1. Compute Mahalanobis distance to all measurement
2. Validate the measurements by gaiting
3. Accept the **closest** validated measurement
4. Update the track as if it were the correct one

- With some probability the selected measurement is **not the correct one**
- **Incorrect associations** can lead to
  - overconfident covariances (covariances collapse in any case)
  - filter divergence and track loss

Strongest Neighbor Standard Filter (SNSF)

- In each step

1. Compute Mahalanobis distance to all measurement
2. Validate the measurements by gaiting
3. Accept the **strongest** validated measurement
4. Update the track as if it were the correct one

- This technique makes sense if there is a **confidence measures** or **signal strength** associated with each measurement
- A **conservative variant** of NNSF and SNSF is to not associate in case of ambiguities (waiting for better weather)

Track Splitting Filter

- In each step

1. When there is more than one measurement in the validation gate, **split the track**
2. Update each split track with the standard Kalman filter equation
3. Compute the **likelihood** of each track
4. Take a **keep/discard decision** by thresholding the likelihood

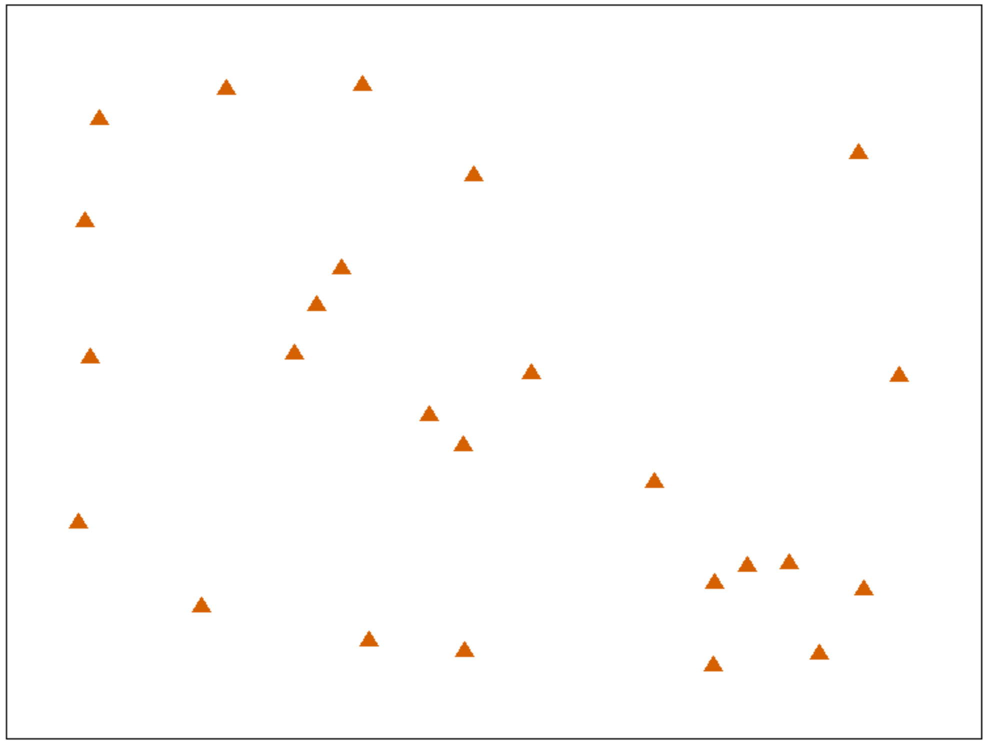
- Exponential growth of number of tracks, unlikely tracks are discarded
- The track likelihood describes the **goodness of fit** of the observations to the assumed target model
- There is **no competition** between the tracks because their likelihoods are computed **separately** and not jointly/globally

Single-Target Data Association

- The previous three approaches achieve decent performance in **well-behaved conditions** (detection probability P_D close to 1, false alarm probability P_F close to zero)
- What if conditions are more challenging in terms of origin uncertainty?
- This may occur when measurements that originate from target are **weak** with respect to background signals and sensor noise
- Integrating false measurements in a tracking filter leads to **divergence** and **track loss**
- Let us thus consider a **more robust** method

Single-Target Data Association

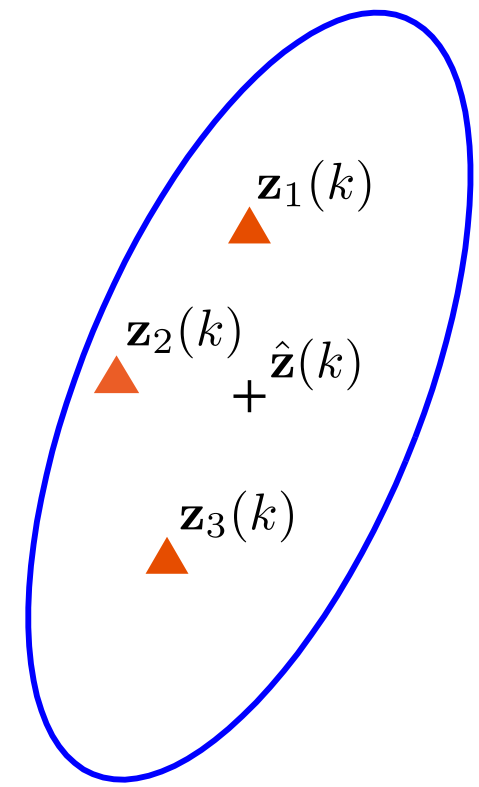
- The previous three approaches achieve decent performance in **well-behaved conditions** (detection probability P_D close to 1, false alarm probability P_F close to zero)
- What if conditions are more challenging in terms of origin uncertainty?
- This may occur when measurements that originate from target are **weak** with respect to background signals and sensor noise
- Integrating false measurements in a tracking filter leads to **divergence** and **track loss**
- Let us thus consider a **more robust** method



Human-Oriented Robotics
Prof. Kai Arras
Social Robotics Lab

Probabilistic Data Association Filter

- Unlike the previous three approaches, the **probabilistic data association filter** (PDAF) is a Bayesian approach that computes the probability of track-to-measurement associations
- **Idea:** Instead of taking a hard decision, the PDAF updates the track with a **weighted average** of **all validated measurements**
- The weights are the individual association probabilities
- Let us define the **association events** $\theta_i(k)$

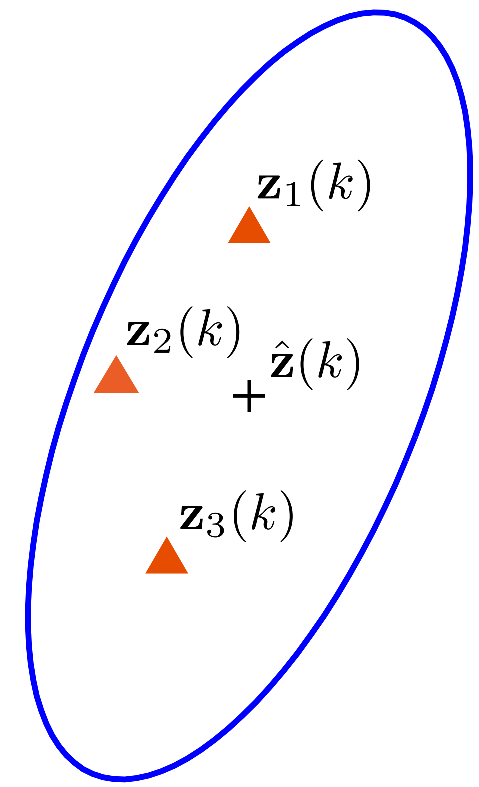


$$\theta_i(k) = \begin{cases} \mathbf{z}_i(k) \text{ is the correct measurement} & i = 1, \ldots, m(k) \\ \text{no correct measurement in gate} & i = 0 \end{cases}$$

where $m(k)$ is the number of validated measurements at time index k

Probabilistic Data Association Filter

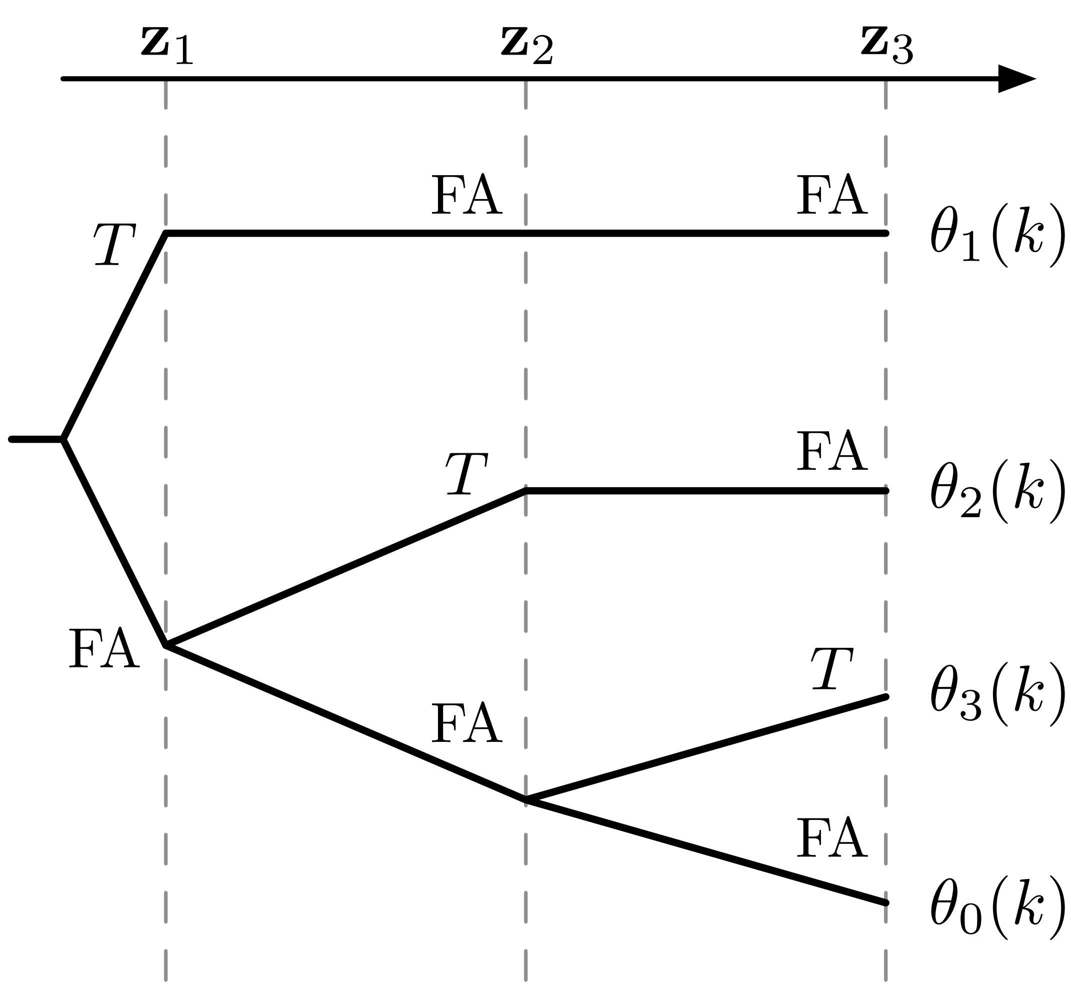
- The PDAF makes the following assumptions:
- Among the $m(k)$ validated measurements, **at most** (i.e. maximal) **one of the validated observations** is target-originated – provided the target (the tracked object) was detected and its observation fell into the validation gate
- The remaining measurements are due to **false alarms** and are modeled with **uniform spatial distribution** and the **number of false alarms** obeys a **Poisson** distribution (the previously considered false alarm model)
- Targets are detected with known probability P_D
- Later, we will also consider track-specific probabilites P_D^t

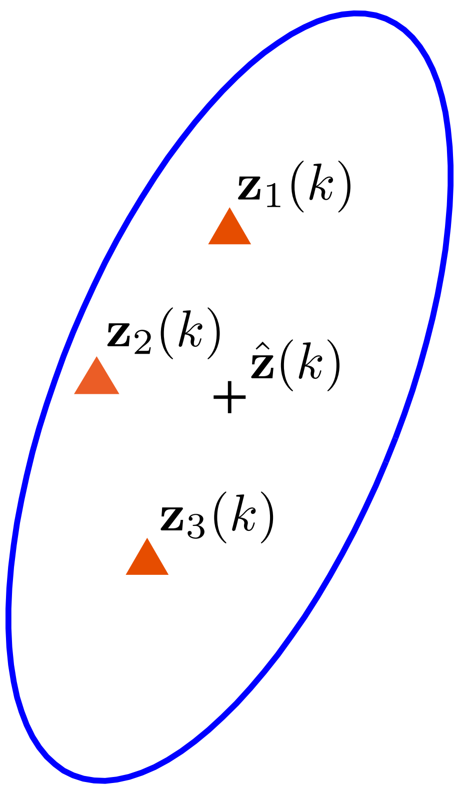


$m(k) = 3$

Probabilistic Data Association Filter

- We can visualize association events in a **tree**





- Each root-to-leaf branch can be seen as an **association hypothesis**

Probabilistic Data Association Filter

- For each association event $\theta_i(k)$, we define the **association probability** $\beta_i = p(\theta_i \,|\, Z^k)$ conditioned on Z^k, the observation history until time k
- It can be shown that this becomes (derivation skipped)

$$p(\theta_i \,|\, Z^k) = \beta_i(k) = \begin{cases} \dfrac{\mathcal{L}_i(k)}{1 - P_D P_G + \sum_{j=1}^{m(k)} \mathcal{L}_i(k)} & i = 1, \ldots, m(k) \\[2ex] \dfrac{1 - P_D P_G}{1 - P_D P_G + \sum_{j=1}^{m(k)} \mathcal{L}_i(k)} & i = 0 \end{cases}$$

where

$$\mathcal{L}_i(k) = \frac{P_D \, \mathcal{N}_{\mathbf{z}_i(k)}(\hat{\mathbf{z}}(k), S(k))}{\lambda}$$

is the **likelihood ratio** of validated measurement $\mathbf{z}_i(k)$ originating from the tracked object rather than being a false alarm

Probabilistic Data Association Filter

- For an interpretation of this result, let us ignore the normalizing denominators and consider the event that **none** of the validated measurements is the correct one, that is $i = 0$

$$\beta_0(k) = \frac{1 - P_D P_G}{1 - P_D P_G + \sum_{j=1}^{m(k)} \mathcal{L}_i(k)}$$

- Parameter P_G is the **gate probability**, the probability that the gate contains the true measurement if detected. Corresponds to threshold γ
- $P_D P_G$ is the probability that the target has been detected **and** its measurement has fallen into the validation gate
- Thus, 1–$P_D P_G$ is the probability that the target has not been detected **or** its measurement has not fallen into the validation gate

Probabilistic Data Association Filter

- In the case that validated measurement $\mathbf{z}_i(k)$ with i = {1, ... , $m(k)$} is the **correct one**, the likelihood ratio

$$\mathcal{L}_i(k) = \frac{P_D \, \mathcal{N}_{\mathbf{z}_i(k)}(\hat{\mathbf{z}}(k), S(k))}{\lambda}$$

trades off the probability that the measurement is target-originated with Gaussian density scaled by P_D versus the spatial uniform Poisson density for false alarms

- The **discrimination capability** of the PDAF relies on the difference between the **Gaussian** and **uniform** densities
- The association probabilities sum up to one, $\sum_{i=1}^{m(k)} \beta_i(k) = 1$, because the association events are **mutually exclusive** and **exhaustive**

Probabilistic Data Association Filter

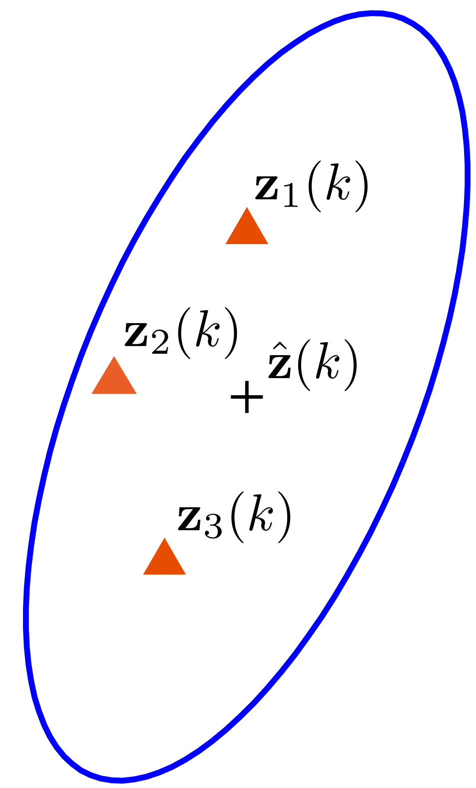
- We will now consider the state and the covariance update of the PDAF
- The **state update** equation of the PDAF is the same as in the Kalman filter

$$\mathbf{x}(k|k) = \mathbf{x}(k|k-1) + K(k)\,\nu(k)$$

but uses a **combined innovation**

$$\nu(k) = \sum_{i=1}^{m(k)} \beta_i(k)\,\nu_i(k)$$

that sums over **all** $m(k)$ **association events** incorporating all validated measurements



$$\nu_1(k) = \mathbf{z}_1(k) - \hat{\mathbf{z}}(k)$$
$$\nu_2(k) = \mathbf{z}_2(k) - \hat{\mathbf{z}}(k)$$
$$\nu_3(k) = \mathbf{z}_3(k) - \hat{\mathbf{z}}(k)$$

- The combined innovation is a **Gaussian mixture**

Probabilistic Data Association Filter

- With the combined innovation, the **covariance update** of the PDAF is

$$P(k|k) = \beta_0 \, P(k|k-1) + (1-\beta_0) \, P(k|k) + \tilde{P}(k)$$

- It contains **three terms** (derivation skipped)
  - With probability β_0 **none** of the measurements is correct, the predicted covariance appears with this weighting ("no update")
  - With probability $(1-\beta_0)$ the **correct** measurement is available and the posterior covariance appears with this weighting
  - Since it is unknown **which** if the $m(k)$ validated measurements is correct, the term $\tilde{P}$ increases the covariance of the updated state. **This increase is the effect of the measurement origin uncertainty**
- Covariance $\tilde{P}$ is the called **spread of innovations**

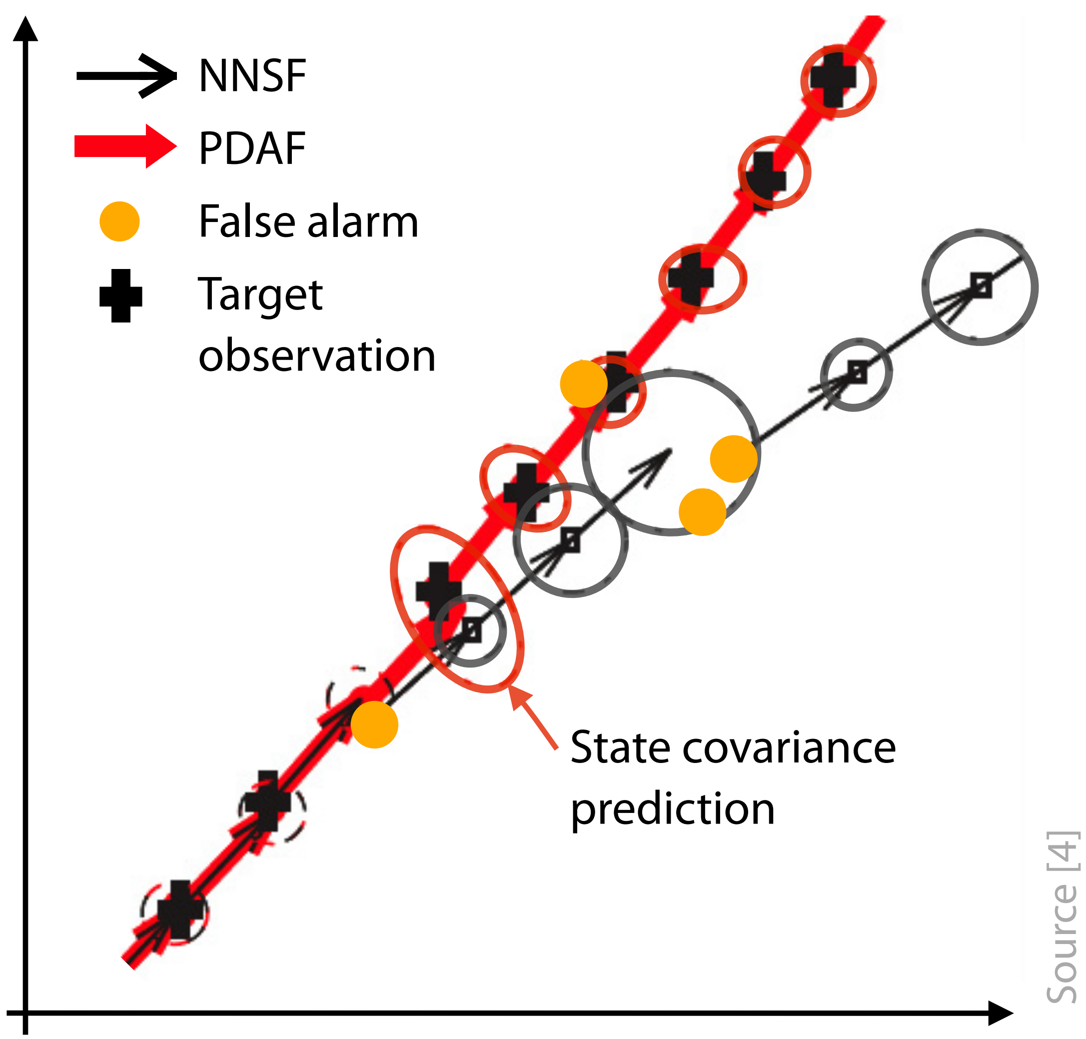
Probabilistic Data Association Filter

- All other calculations in the PDAF
  - state prediction
  - state covariance prediction
  - innovation covariance
  - Kalman gain

  are the same as in the standard Kalman filter
- The only difference is in the use of the **combined innovation** in the state update and the **increased covariance** of the updated state
- Comparing to the nearest neighbor standard filter, the PDAF can be seen as an "**all neighbors**" **filter**
- The **computational requirements** of the PDAF are modest, about double compared to the Kalman filter

Probabilistic Data Association Filter

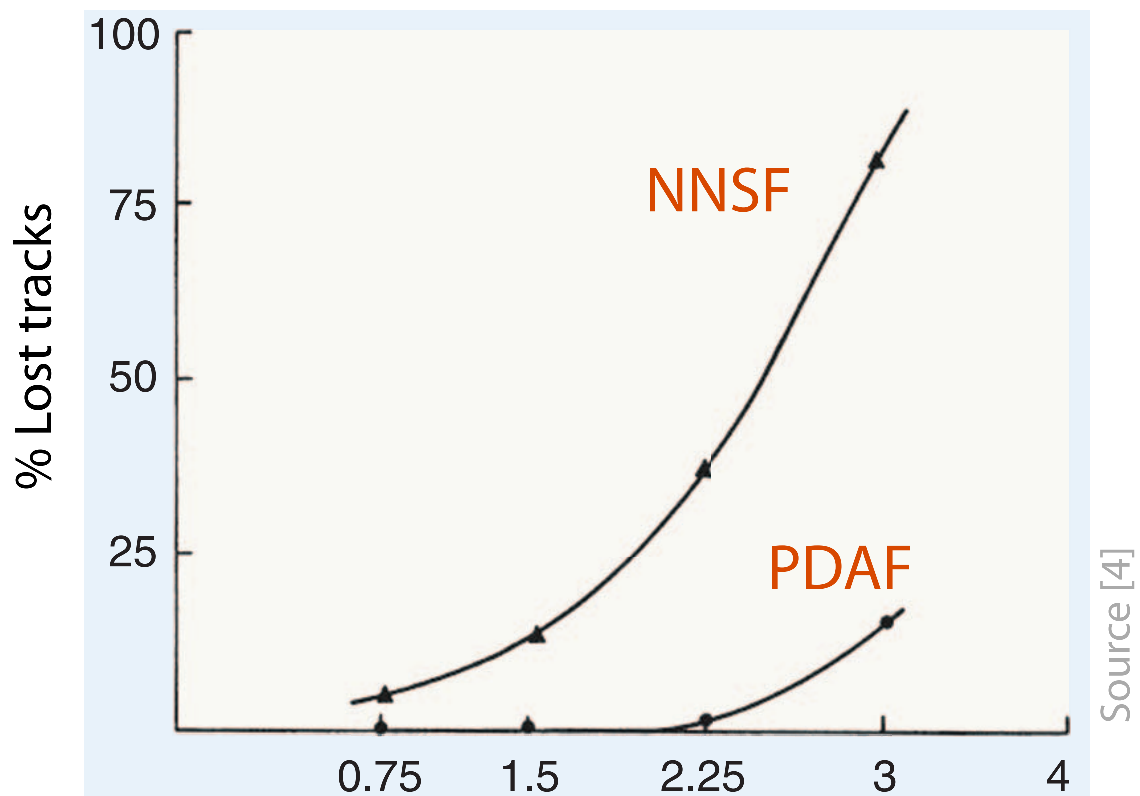
- **Example results**



- Tracking in the presence of false alarms and misdetections
- At $k = 3$ there is no target detection but a false alarm
- The **PDAF**, accounting for the origin uncertainty, attaches a low probability that this measurement is target originated and, consequently, its prediction covariance in the next step is **very large**
- The **NNSF** tracker uses the false measurement as if it were true one and **loses the target**

Probabilistic Data Association Filter

- **Example results**

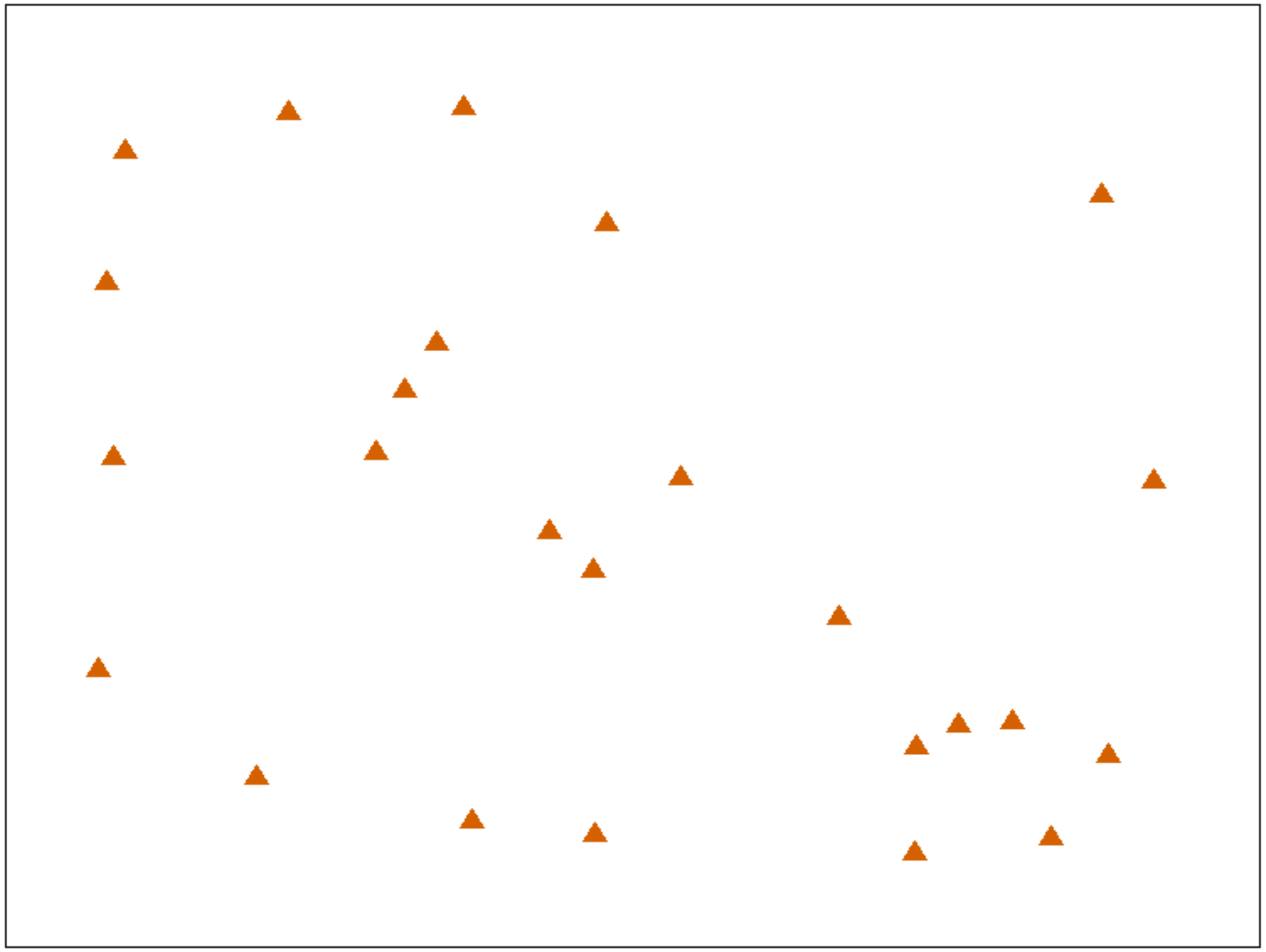


Number of false measurements in NNSF validation gate

- Tracking in the presence of false alarms and misdetections
- The PDAF allows **reliable tracking** up to a clutter level of two false alarms in the validation gate, at which level the NNSF tracker has a track loss probability of about 30%

Probabilistic Data Association Filter

- **Example results**



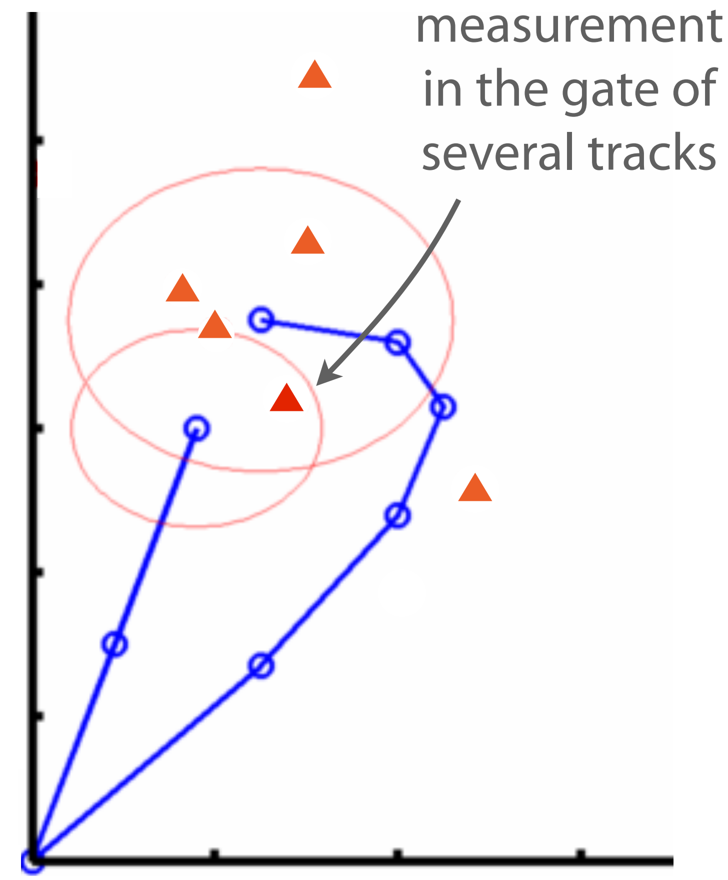
Single-Target Data Association: Wrap Up

- NNSF
  - The **NNSF** takes **hard association decisions**. These hard decisions are sometimes correct and sometimes wrong
  - NNSF is simple to implement and works well in **well-behaved** conditions
- Track splitting filter
  - Instead of taking association decisions, the track splitting filter **grows a tree** of tracks from association ambiguities and relies on the track likelihood as a goodness of fit measure for pruning. Rarely used in practice
- PDAF
  - The **PDAF** makes **soft decisions**, it averages over all validated association possibilities. This soft decision is never totally correct but never totally wrong. This is why the PDAF is a **suboptimal strategy**
  - Compared to the NNSF, the PDAF can significantly improve tracking in regions of high false alarm densities

Multi-Target Data Association

Assumptions

- **Multiple** targets to track
- Tracks already initialized
- Detection probability $P_D < 1$
- False alarm probability $P_F > 0$



Two groups of approaches

- **Non-Bayesian**: no association probabilities
  - Nearest neighbor standard filter (NNSF)
  - Global nearest neighbor standard filter (GNN)
- **Bayesian**: computes association probabilities
  - Joint probabilistic data association filter (JPDAF)
  - Multiple hypothesis tracking (MHT)
  - Markov chain Monte Carlo data association (MCMCDA)

Human-Oriented Robotics
Prof. Kai Arras
Social Robotics Lab

Nearest Neighbor Standard Filter

- Let us **revisit the NNSF** for **multiple targets**
- We introduce the assignment matrix

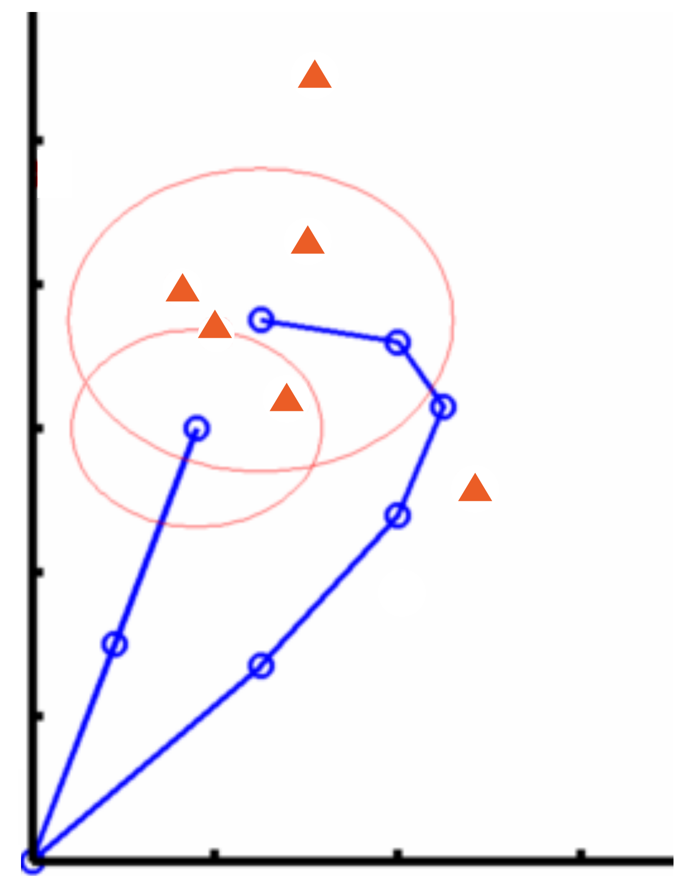
$$A = [d_{ij}^2]$$

with

$$d_{ij}^2 = \nu_{ij}(k)^T \, S_{ij}(k)^{-1} \, \nu_{ij}(k)$$

- For the shown example

$$A = \begin{array}{cc} & \text{observations} \\ \text{tracks} & \begin{pmatrix} d_{11}^2 & d_{12}^2 & d_{13}^2 & d_{14}^2 & d_{15}^2 & d_{16}^2 \\ d_{21}^2 & d_{22}^2 & d_{23}^2 & d_{24}^2 & d_{25}^2 & d_{26}^2 \end{pmatrix} \end{array}$$



2 tracks
6 observations

Nearest Neighbor Standard Filter

- In each step

1. Build the assignment matrix $A = [d_{ij}^2]$
2. Iterate as long as closest pairing passes gaiting test
   - Find the **closest** pairing in A
   - **Remove** the row and column of that pairing
3. Update all tracks as if the associations were the correct ones
4. Unassociated tracks can be used for track deletion, unassociated observations can be used for track initialization

- **Problem**: generally does **not** find the **global** minimum (greedy algorithm)
- **Conservative** variant: no association in case of ambiguities

Global Nearest Neighbor Standard Filter (GNN)

- In each step

1. Build the assignment matrix $A = [d_{ij}^2]$
2. Solve the **linear assignment problem**

$$min \sum d_{ij}^2 \cdot x_{ij} \qquad \text{with} \qquad x_{ij} \in \{0, 1\}$$

$$\sum_i x_{ij} = 1 \qquad \sum_j x_{ij} = 1$$

   - **Hungarian method** for square matrices
   - **Munkres algorithm** for rectangular matrices

3. Check if assignments are in the validation gate and, if yes, update

- Performs data associations **jointly**, finds **global optimum**

Global Nearest Neighbor Standard Filter (GNN)

Linear assignment problem

- The linear assignment problem is a standard problem in **linear programming** and combinatorial optimization
- Used to find, for example, the best assignment of n differently qualified workers to n jobs
- Also called "the personnel assignment problem", first solutions in the 1940s
- By today, many efficient solution methods exist. The **Hungarian method** – while not the most efficient one – is still popular
- The problem can also be solved for non-square matrices by **Munkres' algorithm**

Global Nearest Neighbor Standard Filter (GNN)

Linear assignment problem

- **Problem statement:** We are given an n x n cost matrix $C = [c_{ij}]$, and we want to select n elements of C, so that there is exactly one element in each row and one in each column

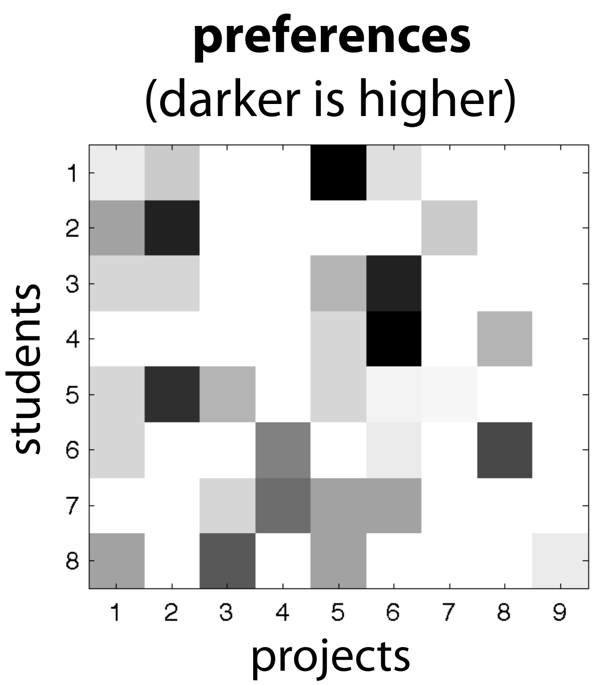
$$\sum_i x_{ij} = 1 \qquad \sum_j x_{ij} = 1$$

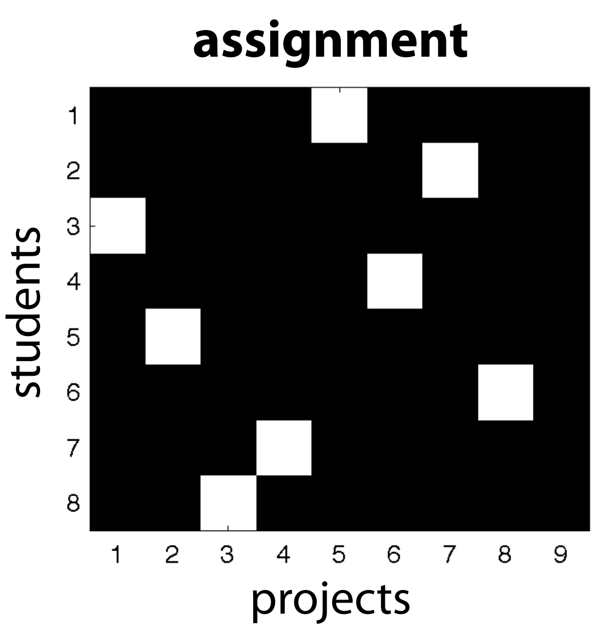
and the sum of the corresponding costs

$$min \sum d_{ij}^2 \cdot x_{ij} \qquad \text{with} \qquad x_{ij} \in \{0,1\}$$

is a **minimum**

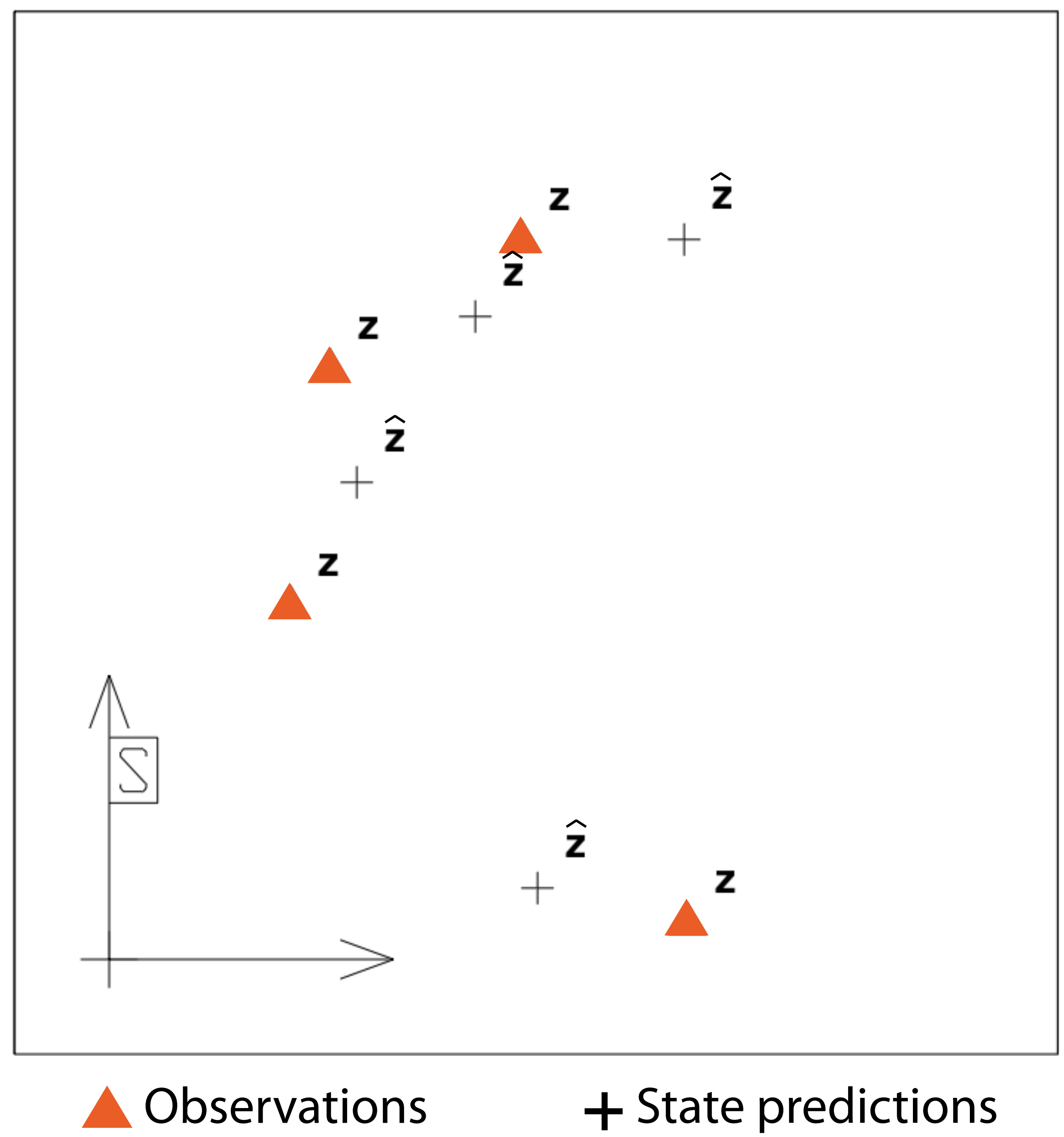
- **Example:** assigning students to class projects





Global Nearest Neighbor Standard Filter (GNN)

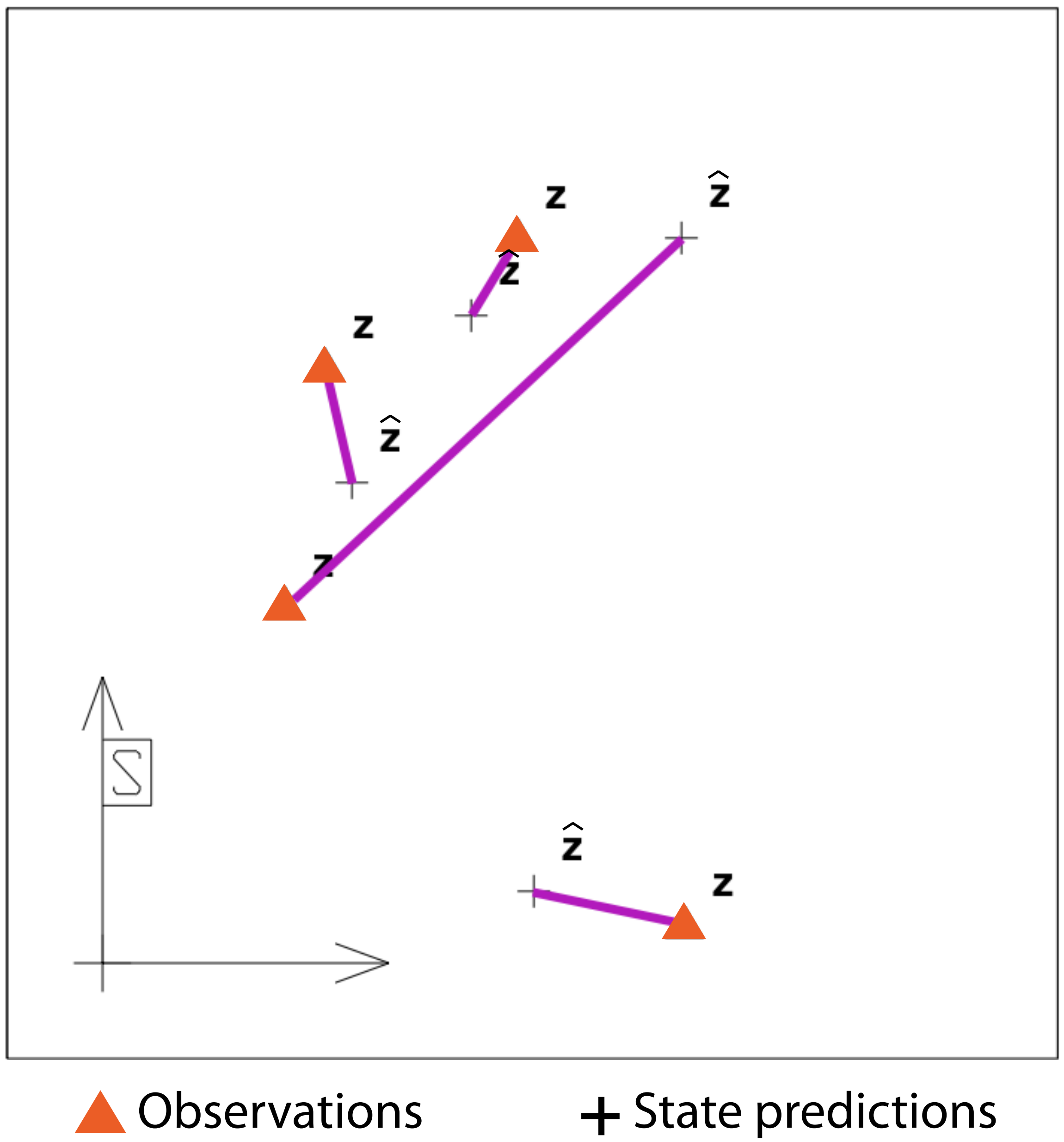
- **NNSF versus GNN example**



What is the globally **best assignment**?

Global Nearest Neighbor Standard Filter (GNN)

- **NNSF versus GNN example**

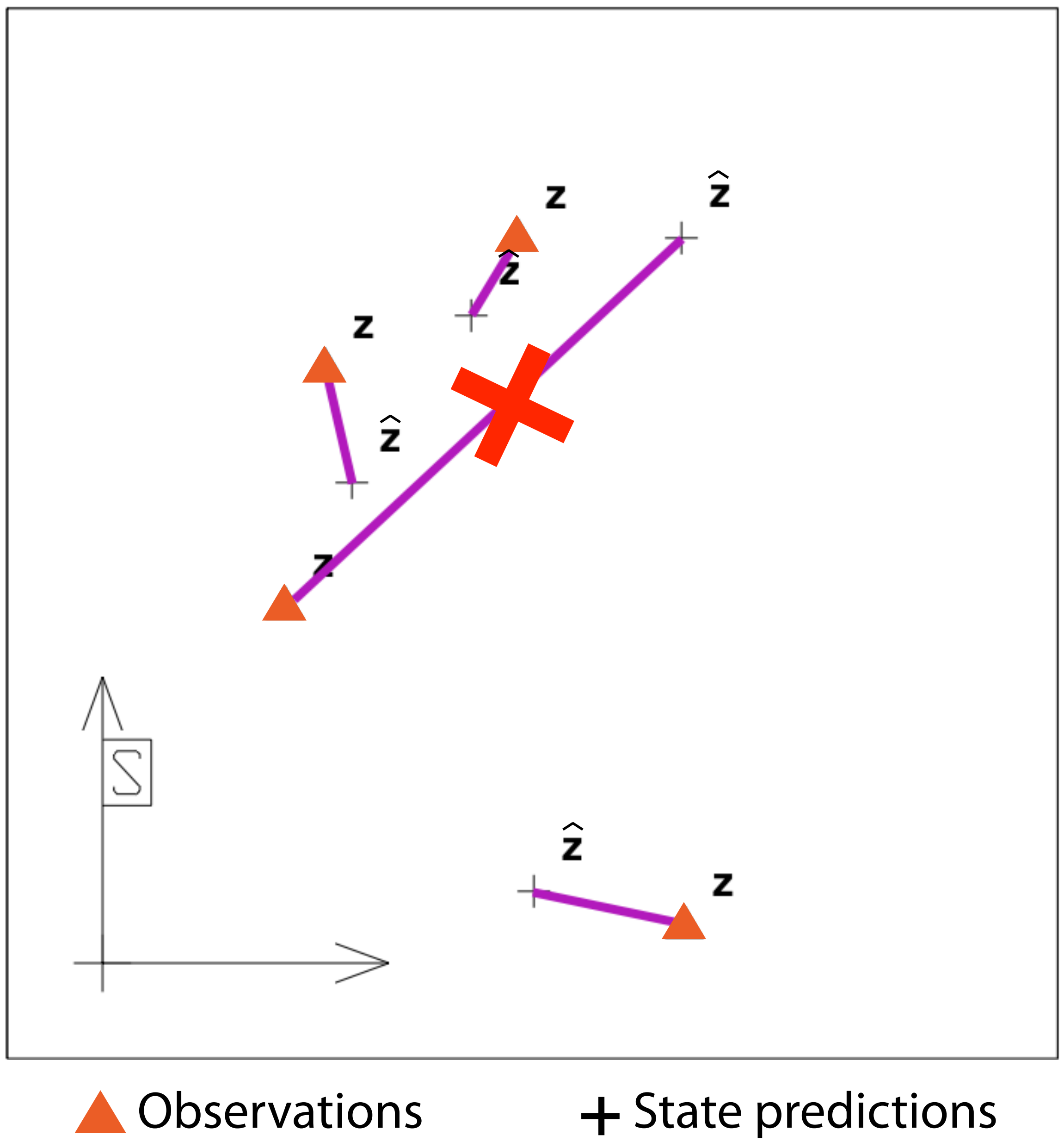


NNSF:
greedy

Human-Oriented Robotics
Prof. Kai Arras
Social Robotics Lab

Global Nearest Neighbor Standard Filter (GNN)

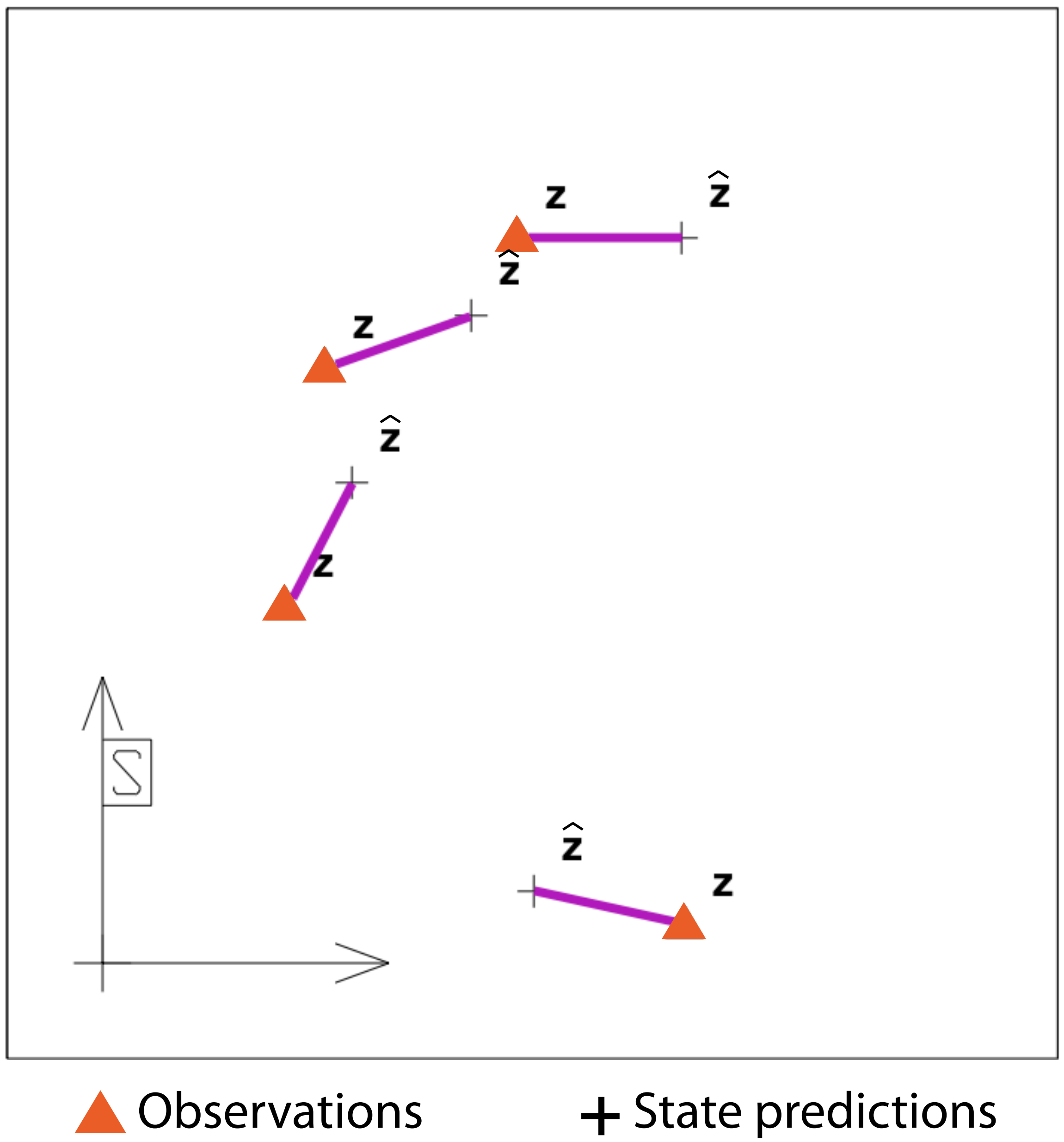
- **NNSF versus GNN example**



Gaiting will reject this assignment

Global Nearest Neighbor Standard Filter (GNN)

- **NNSF versus GNN example**



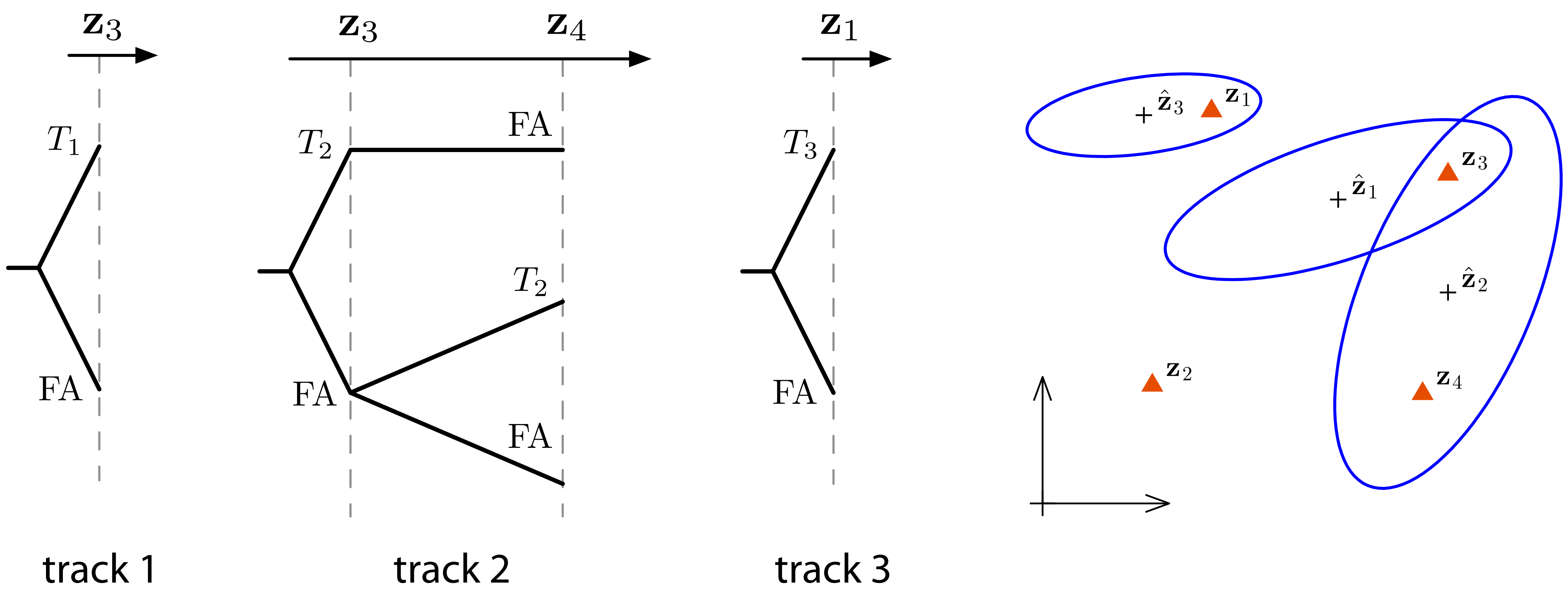
Global NNSF:
Jointly optimal

Joint Probabilistic Data Association Filter (JPDAF)

- Despite its joint optimization, the GNN makes **hard decisions**. Its performance is likely to degrade under more challenging conditions
- Looking for a way to make soft decisions, the **joint probabilistic data association filter** (JPDAF) is a natural multi-target extension of the PDAF
- The **difference** between PDAF and JPDAF lies in the definition of the association events and their probability: the JPDAF considers **joint association events**
- It has the **same state update expressions** as the PDAF
- Given probabilities of joint association events as weights, the JPDAF updates – like the PDAF – the track states with the **combined innovation** over all validated measurements and the track **covariances** with the **spread of innovations term** that accounts for the measurement origin uncertainty

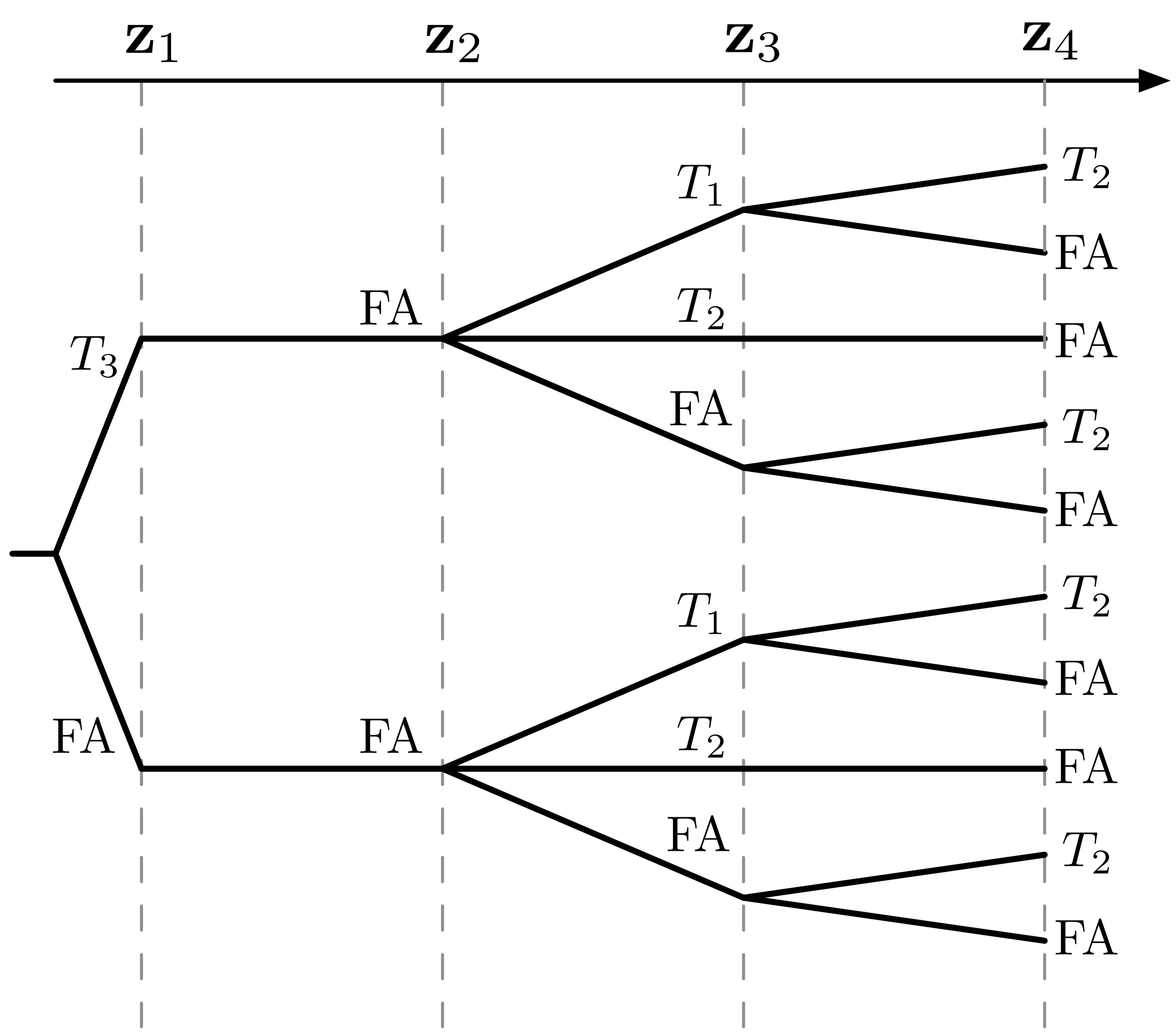
Joint Probabilistic Data Association Filter (JPDAF)

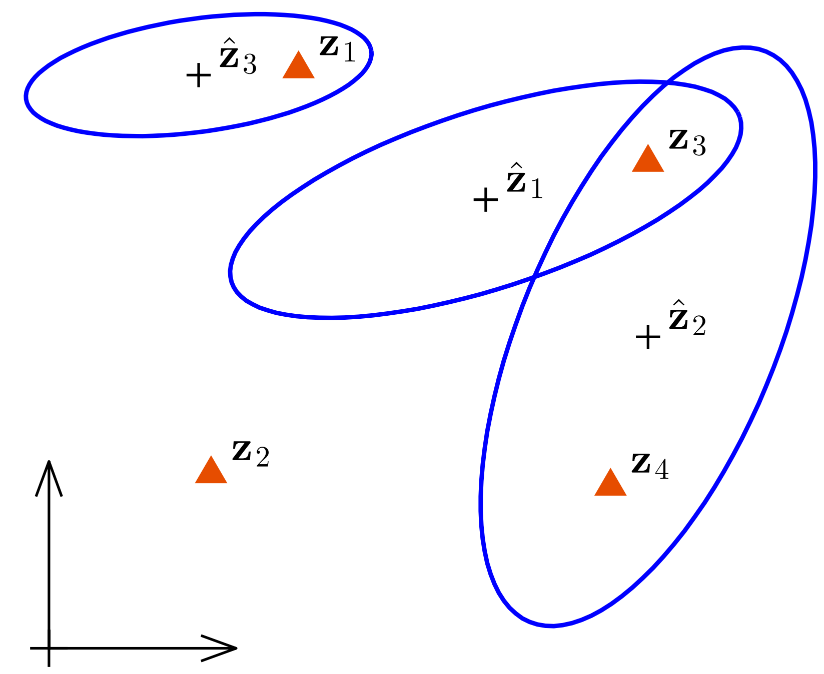
- In this example, the PDAF would define **three disjoint trees** of data association events, one for each track



Joint Probabilistic Data Association Filter (JPDAF)

- The JPDAF defines a single tree of **joint association events**





Joint Probabilistic Data Association Filter (JPDAF)

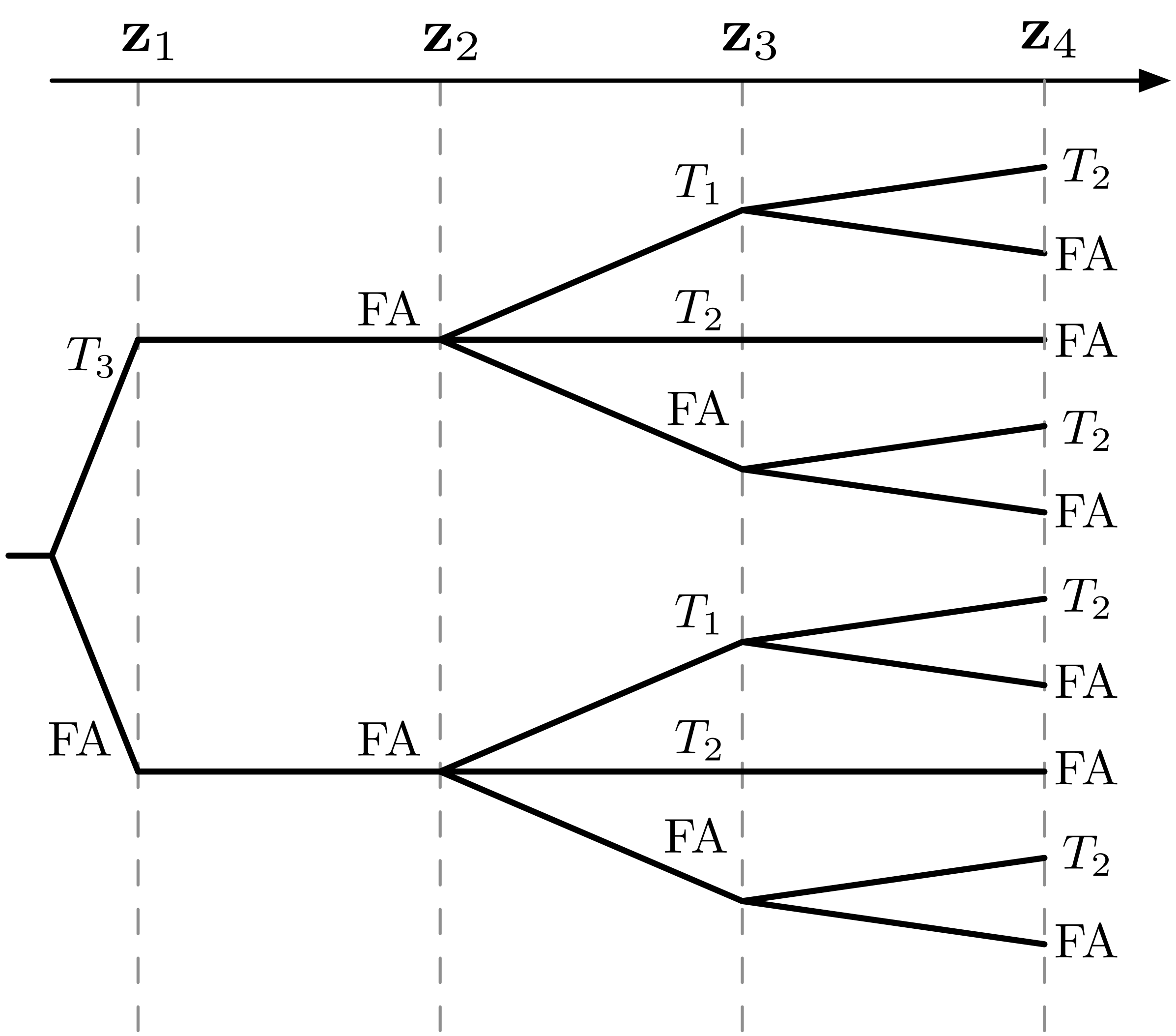
- It can be shown that the **probability** of a **joint association event** θ is

$$p(\theta(k) \,|\, Z^k) = \eta \cdot \underbrace{\lambda^{m_{FA}(k)}}_{\text{all false alarms in } \theta} \underbrace{\prod_{t \in \mathcal{T}_D} P_D^t \, \mathcal{N}_{\mathbf{z}_{j_t}(k)}(\hat{\mathbf{z}}_t(k), S_t(k))}_{\text{all associated targets in } \theta} \underbrace{\prod_{t \in \mathcal{T}_{ND}} (1 - P_D^t P_G^t)}_{\text{all non-associated targets in } \theta}$$

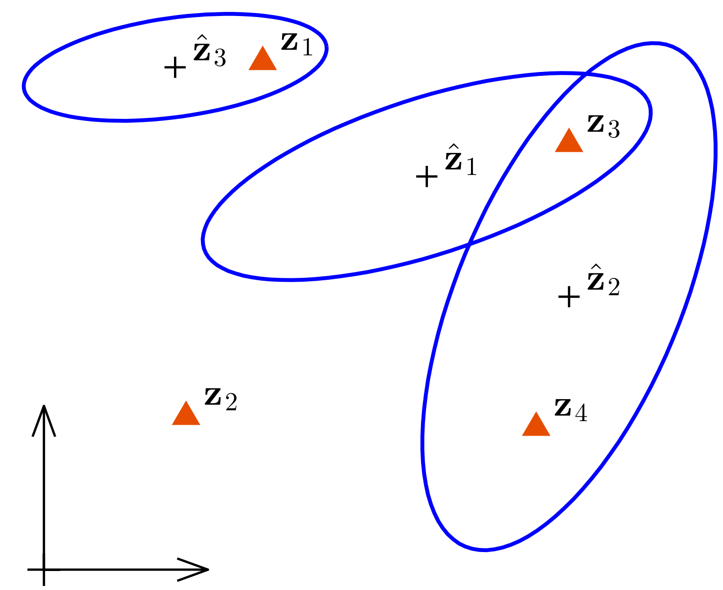
- λ is the Poisson density of false alarms
- P_D^t is the **track-specific** detection probability of track t
- P_G^t is the known gate probability of track t
- $\mathcal{N}_{\mathbf{z}_{j_t}(k)}(\hat{\mathbf{z}}_t(k), S_t(k))$ is the measurement likelihood of observation j_t given track t

Joint Probabilistic Data Association Filter (JPDAF)

- The JPDAF defines a single tree of **joint association events**



$$p_1 \propto \lambda\, l_{13}\, l_{31}\, l_{42}$$
$$p_2 \propto \lambda^2\, l_{13}\, l_{31}\, \bar{l}_2$$
$$p_3 \propto \lambda^2\, l_{13}\, l_{32}\, \bar{l}_1$$
$$p_4 \propto \lambda^2\, l_{13}\, l_{42}\, \bar{l}_1$$
$$p_5 \propto \lambda^3\, l_{13}\, \bar{l}_1\, \bar{l}_2$$
$$p_6 \propto \lambda^2\, l_{31}\, l_{42}\, \bar{l}_3$$
$$p_7 \propto \lambda^3\, l_{31}\, \bar{l}_2\, \bar{l}_3$$
$$p_8 \propto \lambda^3\, l_{32}\, \bar{l}_1\, \bar{l}_3$$
$$p_9 \propto \lambda^3\, l_{42}\, \bar{l}_1\, \bar{l}_3$$
$$p_{10} \propto \lambda^4\, \bar{l}_1\, \bar{l}_2\, \bar{l}_3$$



where $l_{ij} = P_D^j \mathcal{N}_{\mathbf{z}_i}(\hat{\mathbf{z}}_j, S_j)$ and $\bar{l}_j = (1 - P_D^j P_G^j)$

Joint Probabilistic Data Association Filter (JPDAF)

- For the state update of track t we require the **marginal association probability** $\beta_{jt} = p(\theta_{jt} | Z^k)$ of the event that observation j originates from track t, obtained by marginalization of the joint probability
- The marginal association probability β_{jt} are then the weights in the combined innovation for state and state covariance updates

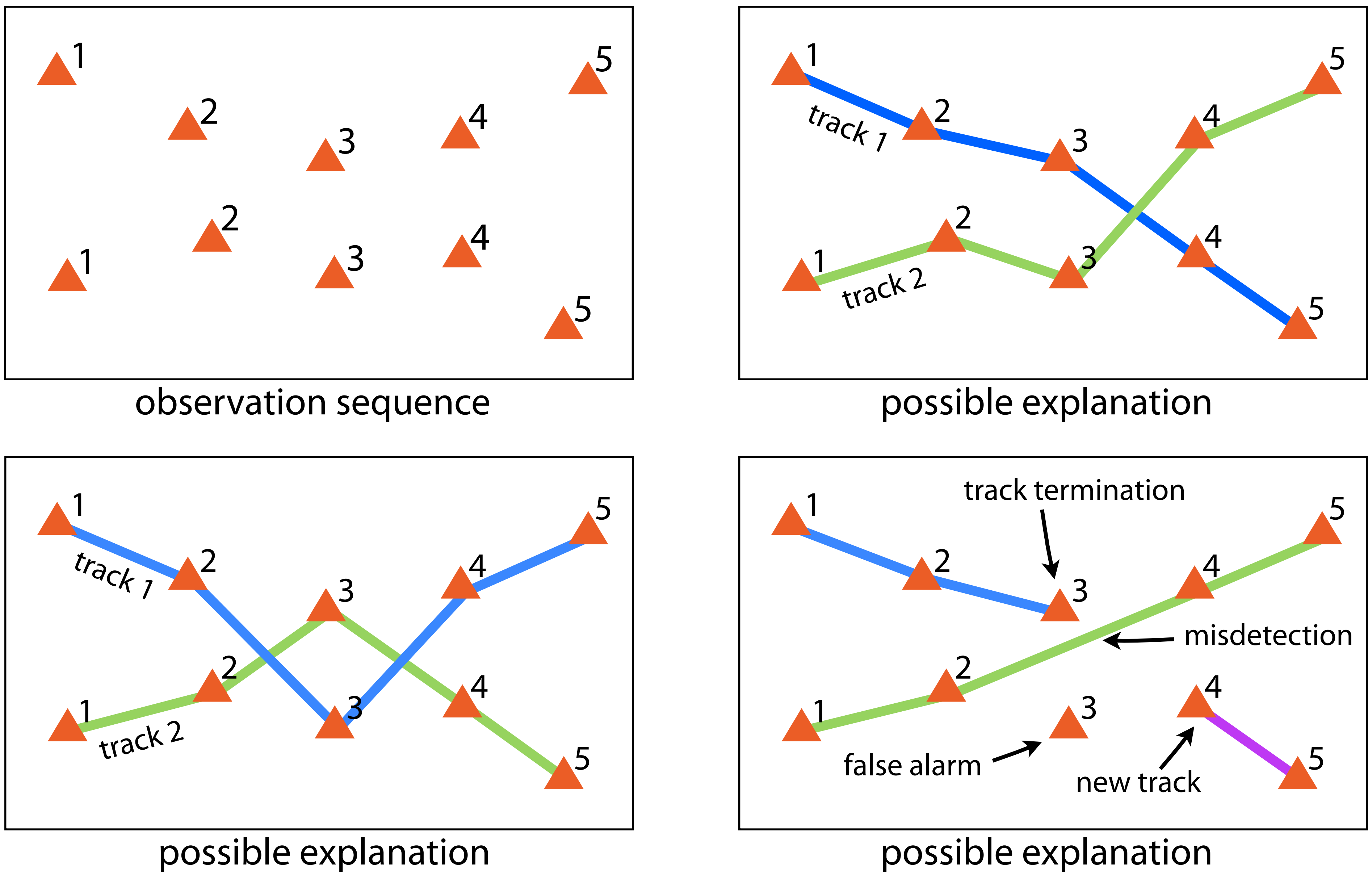
- The filter assumes the number of tracks to be known. Thus, a **separate track initiation logic** must run along to create new tracks
- JPDAF is the **soft decision equivalent of the GNN** in the same way that the PDAF is a soft version of the NNSF
- JPDAF **collapses** the hypothesis trees after each step

Multi-Target Data Association

- All data association methods considered so far are **single-frame** or **single-scan.** Decisions – hard or soft – are taken after each step
- This is a rather **myopic strategy** and likely to fail in challenging conditions where, for example, misdetections have to be distinguished from occlusion events in the presence of both false alarms and target maneuvers
- Thus, we want to **delay decisions** and accumulate information to the point where we can take more informed decisions ("integrating information over time")
- This implies the maintenance of multiple histories/sequences of hypothetical data association decisions
- The **multiple hypothesis tracking** (MHT) approach implements this idea in a general way

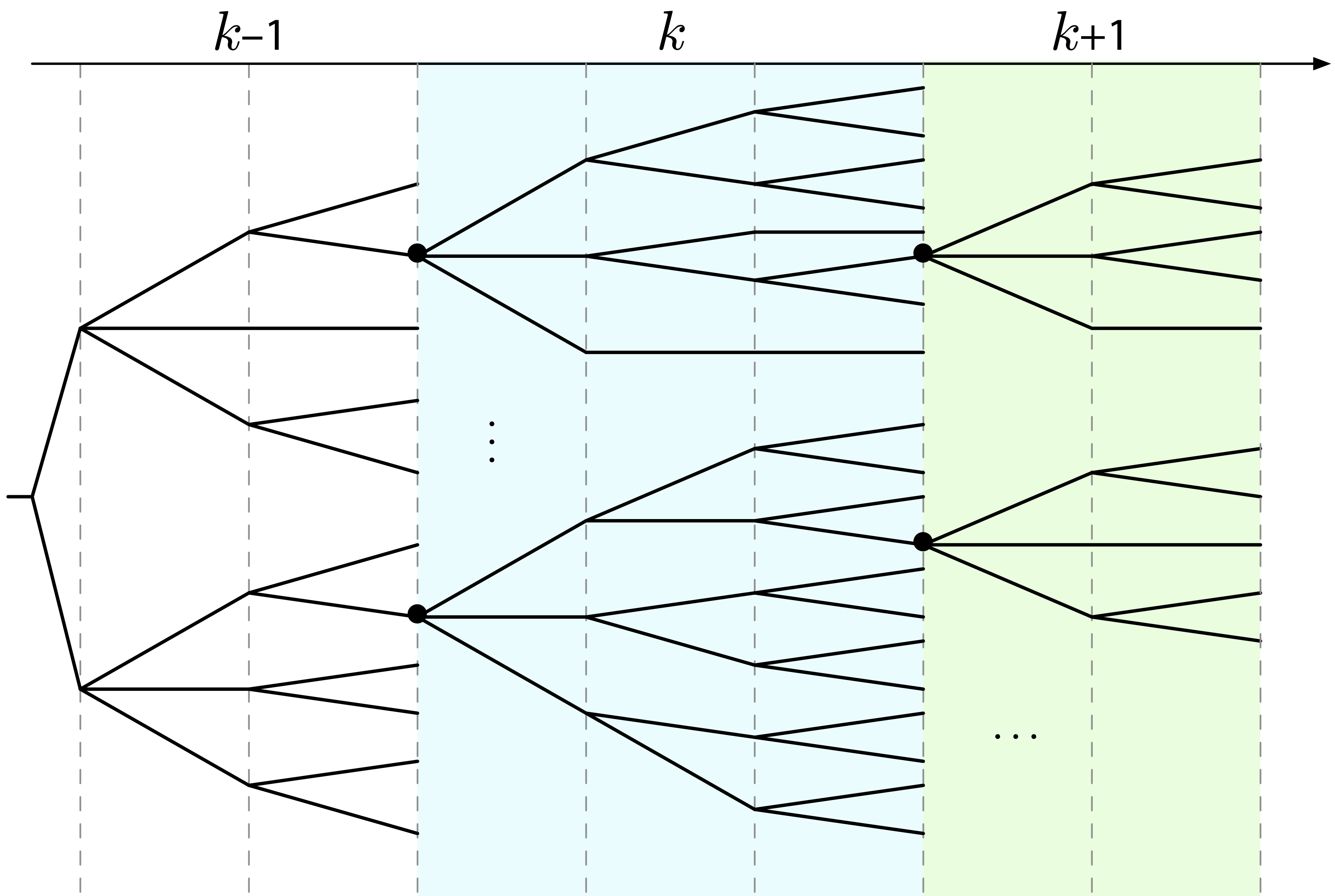
Multiple Hypothesis Tracking (MHT)

- The MHT considers the association of **sequences of observations**



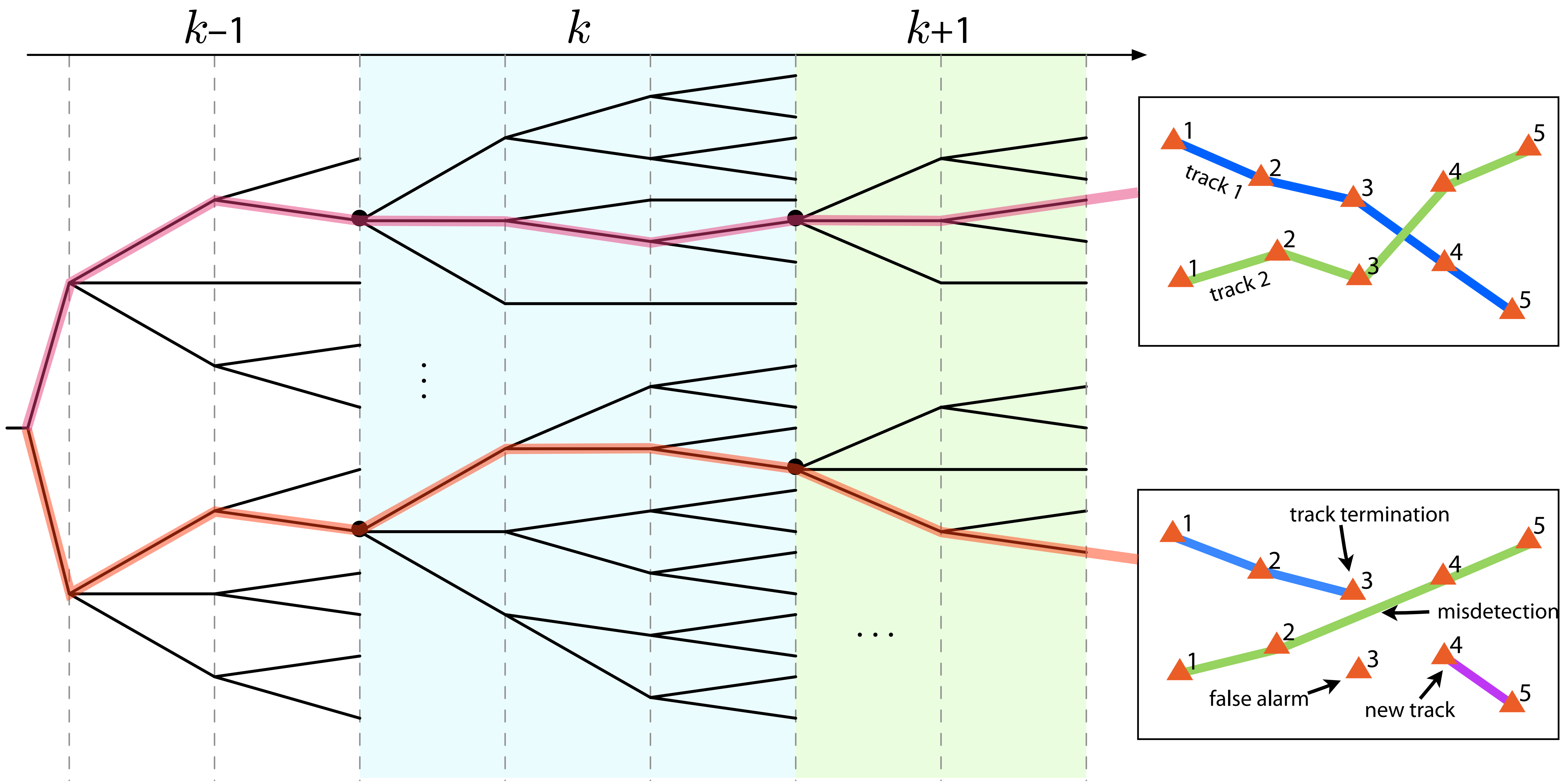
Multiple Hypothesis Tracking (MHT)

- The MHT **concatenates** the trees of each step to one big **hypothesis tree**



Multiple Hypothesis Tracking (MHT)

- The MHT **concatenates** the trees of each step to one big **hypothesis tree**



Multiple Hypothesis Tracking (MHT)

- The number of association histories increases **exponentially** and results in an ever-growing **hypothesis tree**
- For practical implementations, **pruning** strategies are mandatory
- Without pruning, the MHT approach is the **optimal** Bayesian data association solution (no simplifications or approximations)
- In addition to the measurement-to-track associations, the MHT can reason about **track interpretations** as
  - occluded (label O)
  - deleted (label T)

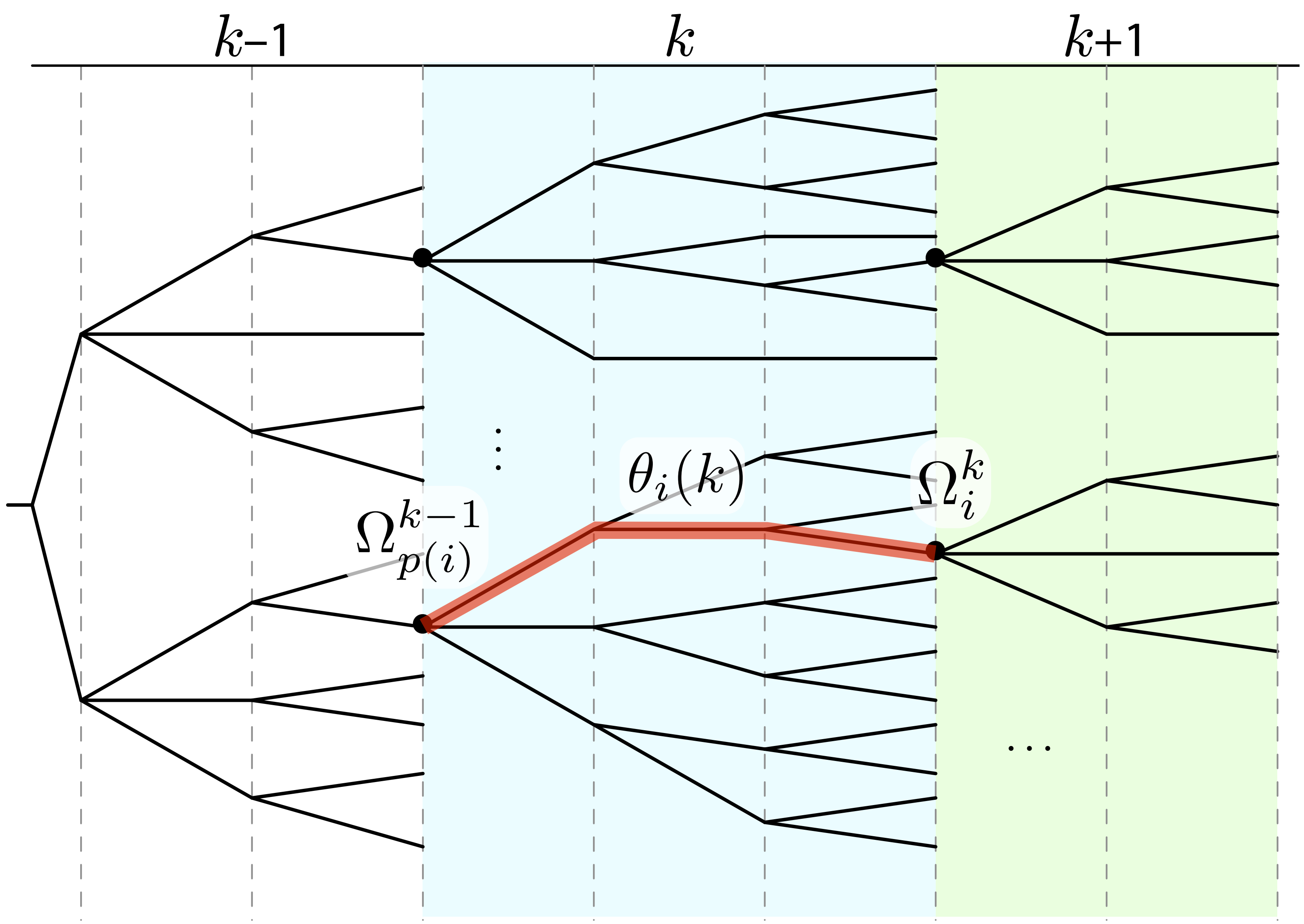
  and **measurement interpretations** as
  - false alarms (label FA or F)
  - new tracks (label N)
- **New tracks** are also modeled as spatially uniform, Poisson in the number

Multiple Hypothesis Tracking (MHT)

- In this way, the MHT can deal with the **entire life cycle of tracks** initialization–confirmation–occlusions–deletion in a probabilistically consistent way
- No need for an additional track management logic (for initialization or deletion)
- Let an **association hypothesis** or simply **hypothesis** Ω^k be a root-to-leaf path through the entire tree until time k
- What is then the **best hypothesis**?
- To answer this query, we compute probabilities for hypotheses leading to a discrete probability distribution over hypotheses
- Then, we search through all hypotheses and find the best one as the one with the **highest probability**

Multiple Hypothesis Tracking (MHT)

- Let us consider the **probability of a hypothesis**. We define



- **Parent** hypothesis $p(i)$
  $\Omega_{p(i)}^{k-1}$
- **Association** or assignment set
  $\theta_i(k)$
- **Child** hypothesis i
  Ω_i^k
- Their **relation**
  $\Omega_i^k = \{\Omega_{p(i)}^{k-1}, \theta_i(k)\}$

Multiple Hypothesis Tracking (MHT)

- Let further $Z(k) = \{\mathbf{z}_i(k)\}_{i=1}^{m(k)}$ be the set of current observations and Z^k the observation history up to time k
- Then the **probability** of Ω_i^k at time k is

$$
\begin{aligned}
p(\Omega_i^k | Z^k) &= p(\theta_i(k), \Omega_{p(i)}^{k-1} | Z^k) \\
&= p(\theta_i(k), \Omega_{p(i)}^{k-1} | Z(k), Z^{k-1}) && \text{dividing up the evidence} \\
&= p(\Omega_{p(i)}^{k-1} | Z(k), Z^{k-1})\; p(\theta_i(k) | \Omega_{p(i)}^{k-1}, Z(k), Z^{k-1}) && \text{chain rule} \\
&= p(\theta_i(k) | Z(k), Z^{k-1}, \Omega_{p(i)}^{k-1})\; p(\Omega_{p(i)}^{k-1} | Z^{k-1}) && \text{conditional indep.} \\
&= \eta\, p(Z(k) | \theta_i(k), Z^{k-1}, \Omega_{p(i)}^{k-1})\; p(\theta_i(k) | Z^{k-1}, \Omega_{p(i)}^{k-1})\; p(\Omega_{p(i)}^{k-1} | Z^{k-1}) && \text{Bayes} \\
&= \eta\, p(Z(k) | \theta_i(k), \Omega_{p(i)}^{k-1})\; p(\theta_i(k) | \Omega_{p(i)}^{k-1})\; p(\Omega_{p(i)}^{k-1} | Z^{k-1}) && \text{conditional indep.}
\end{aligned}
$$

Multiple Hypothesis Tracking (MHT)

- Let further $Z(k) = \{\mathbf{z}_i(k)\}_{i=1}^{m(k)}$ be the set of current observations and Z^k the observation history up to time k
- Then the **probability** of Ω_i^k at time k is

$$
\begin{aligned}
p(\Omega_i^k|Z^k) &= p(\theta_i(k), \Omega_{p(i)}^{k-1}|Z^k) \\
&= p(\theta_i(k), \Omega_{p(i)}^{k-1}|Z(k), Z^{k-1}) \\
&= p(\Omega_{p(i)}^{k-1}|Z(k), Z^{k-1})\; p(\theta_i(k)|\Omega_{p(i)}^{k-1}, Z(k), Z^{k-1}) \\
&= p(\theta_i(k)|Z(k), Z^{k-1}, \Omega_{p(i)}^{k-1})\; p(\Omega_{p(i)}^{k-1}|Z^{k-1}) \\
&= \eta\; p(Z(k)|\theta_i(k), Z^{k-1}, \Omega_{p(i)}^{k-1})\; p(\theta_i(k)|Z^{k-1}, \Omega_{p(i)}^{k-1})\; p(\Omega_{p(i)}^{k-1}|Z^{k-1}) \\
&= \eta\; \underbrace{p(Z(k)|\theta_i(k), \Omega_{p(i)}^{k-1})}_{\text{measurement likelihood}}\; \underbrace{p(\theta_i(k)|\Omega_{p(i)}^{k-1})}_{\text{association probability}}\; \underbrace{p(\Omega_{p(i)}^{k-1}|Z^{k-1})}_{\text{recursive term}}
\end{aligned}
$$

Multiple Hypothesis Tracking (MHT)

- The **measurement likelihood** (derivation skipped)

$$\underbrace{p(Z(k)|\theta_i(k), \Omega_{p(i)}^{k-1})}_{\text{measurement likelihood}} = \frac{1}{V^{m_F(k)+m_N(k)}} \prod_{i=1}^{m(k)} \mathcal{N}_{\mathbf{z}_i(k)}(\hat{\mathbf{z}}_{t_i}(k), S_{t_i}(k))^{\tau_i}$$

  - If observation $\mathbf{z}_i(k) \in Z(k)$ is **in the gate** of track t_i

$$p(\mathbf{z}_i(k)|\theta_i(k), \Omega_{p(i)}^{k-1}) = \mathcal{N}_{\mathbf{z}_i(k)}(\hat{\mathbf{z}}_{t_i}(k), S_{t_i}(k))$$

  - If observation $\mathbf{z}_i(k)$ is a **false alarm**

$$p(\mathbf{z}_i(k)|\theta_i(k), \Omega_{p(i)}^{k-1}) = V^{-1}$$

  - If observation $\mathbf{z}_i(k)$ is a **new track**

$$p(\mathbf{z}_i(k)|\theta_i(k), \Omega_{p(i)}^{k-1}) = V^{-1}$$

Multiple Hypothesis Tracking (MHT)

- The **association probability** (derivation skipped)

$$\underbrace{p(\theta_i(k)|\Omega^{k-1}_{p(i)})}_{\text{association probability}} = \frac{m_F(k)!m_N(k)!}{m(k)!}\,\mu_F(m_F(k))\,\mu_N(m_N(k))\prod_{t=1}^{n_T(k)}(P_D^t)^{\delta_t}\,(1-P_D^t)^{1-\delta_t}$$

  - Computes the **prior probability** of association $\theta_i(k)$ based on known parameters such as probability of detection and Poisson densities for false alarms/new tracks

- The **final expression**

$$\boxed{\begin{aligned} p(\Omega_i^k|Z^k) = \eta\cdot\lambda_F^{m_F(k)}\lambda_N^{m_N(k)}\prod_{t\,\in\,\mathcal{T}_D} P_D^t\,\mathcal{N}_{\mathbf{z}_{j_t(k)}}(\hat{\mathbf{z}}_t(k),S_t(k))\prod_{t\,\in\,\mathcal{T}_{ND}}(1-P_D^t) \\ \cdot\, p(\Omega^{k-1}_{p(i)}|Z^{k-1}) \end{aligned}}$$

Multiple Hypothesis Tracking (MHT)

- Note the **similarity** of the association probability in the MHT and JPDAF. The differences come from the ability of the MHT to interpret observations also as **new tracks** and the **recursiveness** of the computation
- The JPDAF creates only single-step trees and collapses them after each step by incorporating all validated observations over a combined innovation approach
- In the same situation, the MHT solves the data association ambiguity by **splitting** the track and creating **offsprings**
- But unlike the track splitting filter, the different offsprings compete with each other in a fully **Bayesian framework**
- MHT maintains a **standard** KF or EKF for each hypothesized track
- MHT takes **hard**, **multiple** and **delayed** decisions

Multiple Hypothesis Tracking (MHT)

There are several **pruning** techniques to limit the number of hypotheses

- **Clustering** spatially disjoint hypothesis trees
  - Tracks are partitioned into clusters along "**uncoupled**" **observations**
  - A separate tree is grown for each cluster
- **Merging hypotheses**
  - Combine hypotheses with **similar effect**, typically with a **common recent history**
  - For example, the same number of targets but with slightly different track states
- **Eliminate low probability hypotheses**
  - A variant thereof if **ratio pruning** that considers the probability ratio with the best hypothesis. Unlikely hypothesis below a ratio threshold are discarded
  - Caution! Branches that turn out to hold the true hypothesis at a later point may start with a very unlikely ancestor hypothesis

Multiple Hypothesis Tracking (MHT)

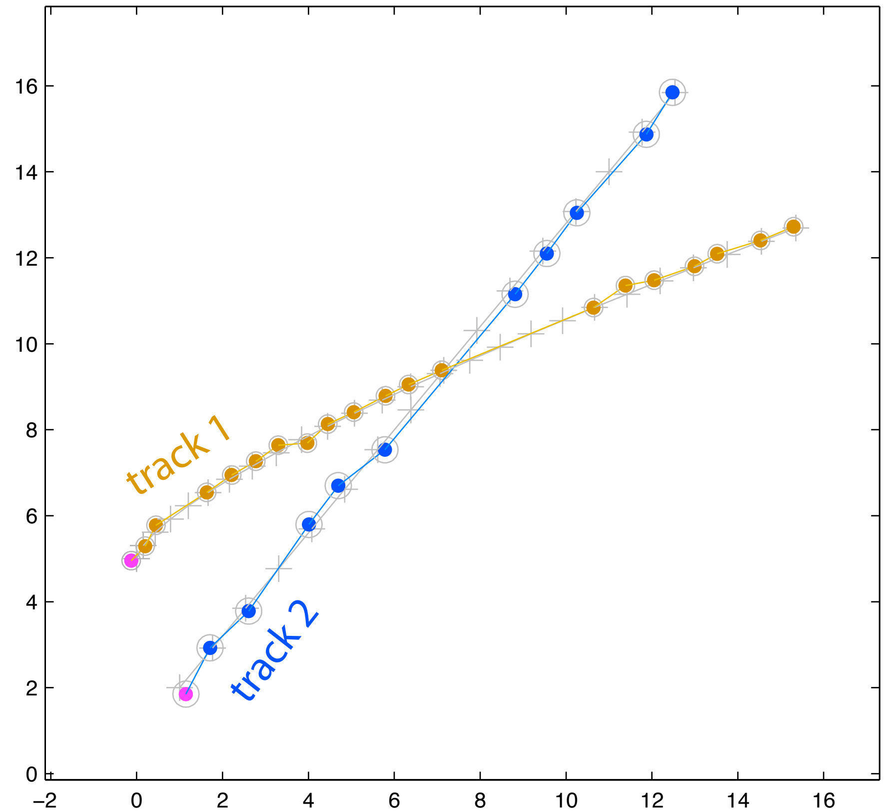
There are several **pruning** techniques to limit the number of hypotheses

- K**-best branching**
    - Directly generate the k-best hypotheses
    - **Murty's algorithm** incorporates the generation and evaluation of hypotheses in a single algorithm with **polynomial time** complexity
    - Implements a generate-while-prune versus a generate-then-prune strategy
- N-**scan back** pruning
    - Ambiguities are assumed to be resolved after N steps
    - Ancestor hypotheses at time k–N receive probability mass of their descendants at k
    - Keep only subtree of the most probably ancestor hypothesis

Human-Oriented Robotics
Prof. Kai Arras
Social Robotics Lab

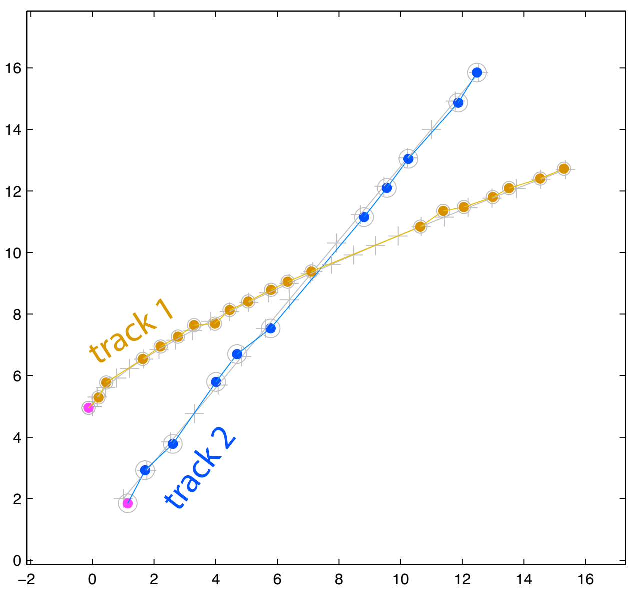
Multiple Hypothesis Tracking (MHT)

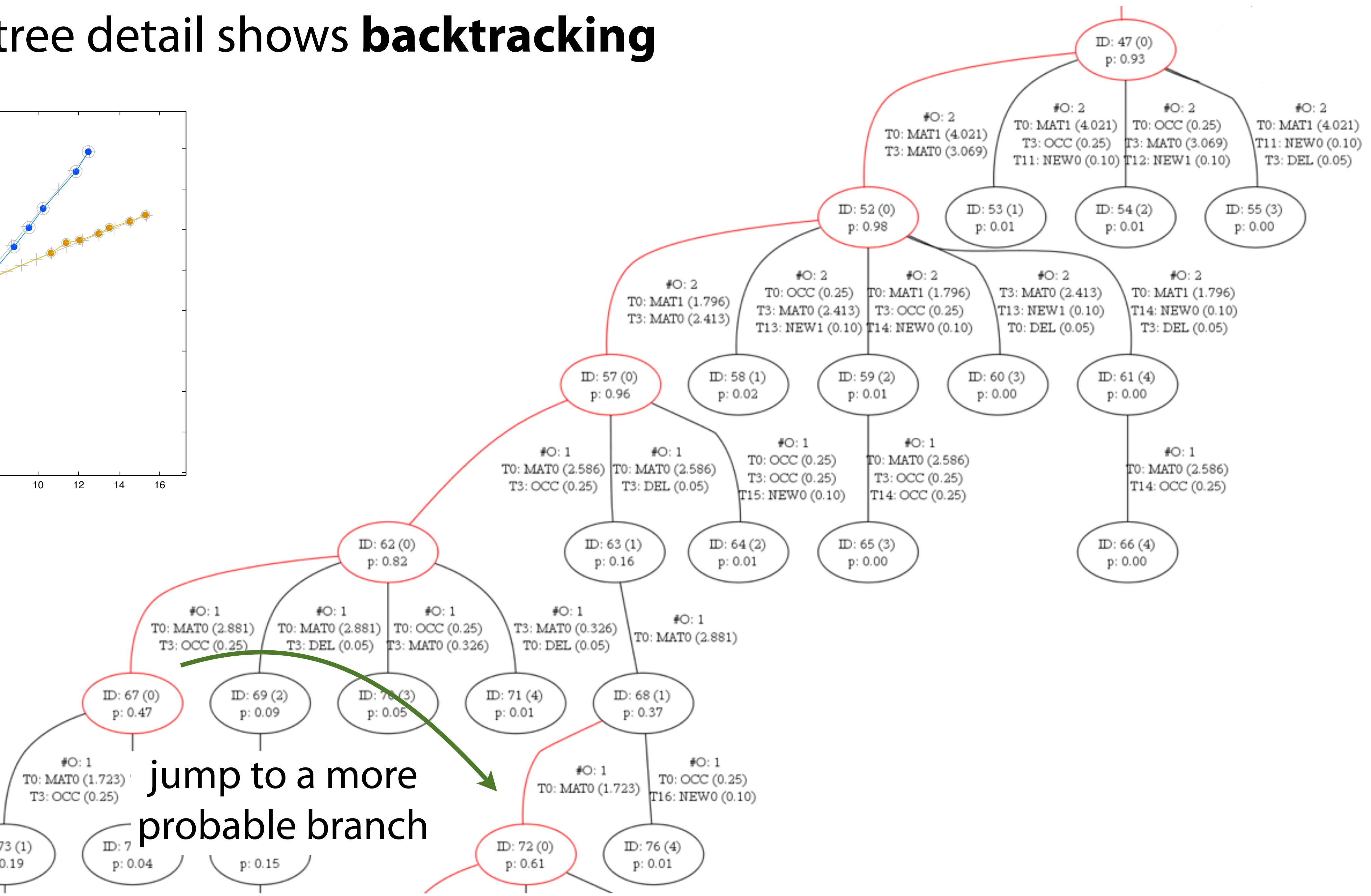
- Example, showing only the 5-best hypotheses



Multiple Hypothesis Tracking (MHT)

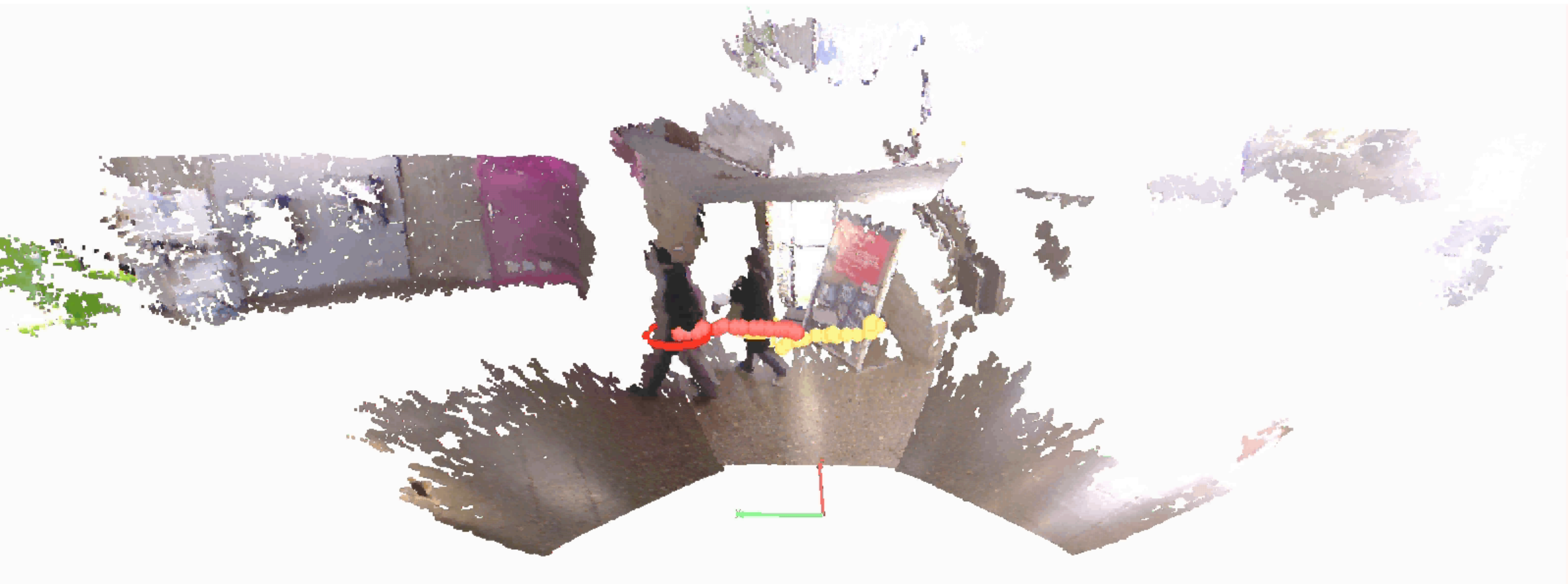
- Example, tree detail shows **backtracking**





Multiple Hypothesis Tracking (MHT)

- Example: Tracking people in RGB-D data



Multiple Hypothesis Tracking (MHT)

- Example: Tracking pedestrians in 3D point clouds

Multiple Hypothesis Tracking (MHT)

- Example: Tracking pedestrian in Freiburg city center in 2D laser data
- Difficult scenario with track identifier switches – mainly due to little information from sensor, frequent and long occlusion events

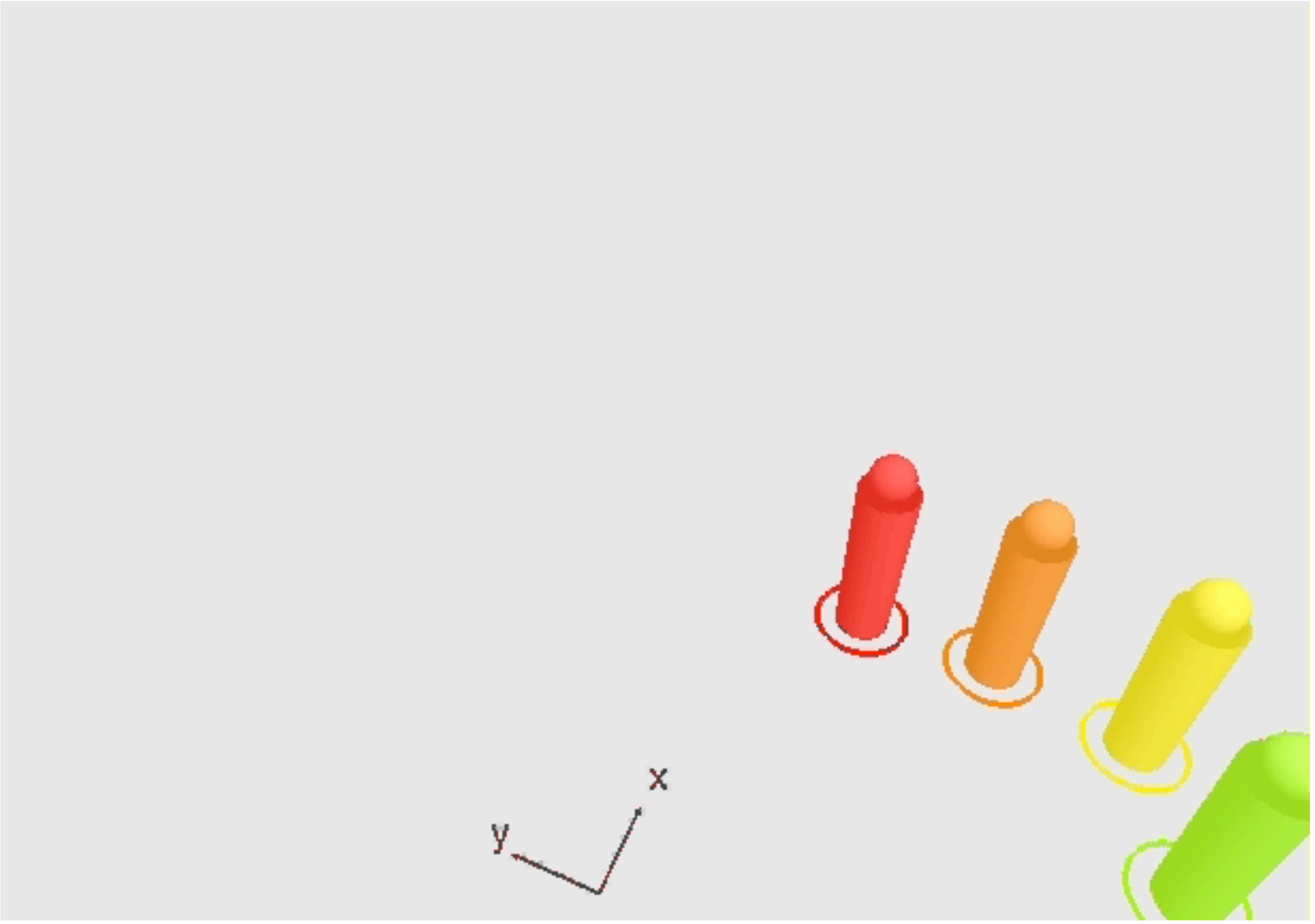
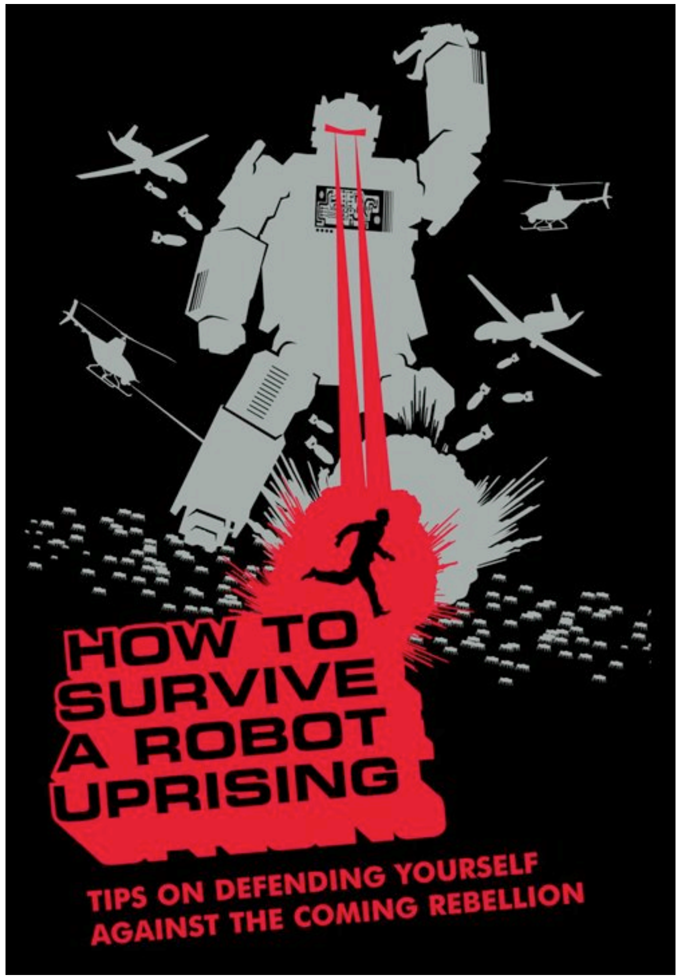


image data (not used for tracking)

Why We Teach This...

How to Escape a Rebellious Humanoid Robot?

- Run toward the light
- Find clutter to hide
- Hug a comrade, then dive into random direction
- Wear similar clothing
- Don't run in a predictable line, zigzag erratically
- Try to mix with the crowd
- Wear trenchcoat or long skirt to mask your movements
- Hop, skip or jump occasionally
- Vary rhythm and length of your stride



"How to Survive a Robot Uprising: Tips on Defending Yourself Against the Coming Rebellion,"

by Daniel H. Wilson, Bloomsbury 2005

Ask yourself which parts of the robot's tracking system is fooled by those actions

Summary

- **Tracking** is maintaining the **state** and **identity** of a **moving object** over time based on remote measurements
- An key issue in tracking is **data association**: the problem of associating measurements to tracks under significant levels of **origin uncertainty**. Data association also deals with the **interpretation** of measurements and tracks as false alarms/new tracks, or occluded/terminated
- The simplest form of data association (which can also be seen as a preprocessing step) is **gaiting**: the validation gate is a **region of acceptance** such that $100(1-\alpha)\%$ of **true measurements** are **rejected**
- **False alarms** (as well as new tracks) are modeled as uniform over space and Poisson distribution in their number per step
- The **NNSF** makes greedy associations based on smallest Mahalanobis distances. These hard decisions are sometimes **correct**, sometimes **wrong**

Summary

- The **PDAF** makes soft decisions by integrating all validated measurements in the gate. Decisions are never totally correct but never totally wrong
- For multi-target data association, the **GNN** makes hard but **jointly optimal** decisions by solving a linear assignment problem
- The **JPDAF** is the soft version of the GNN in the way that the PDAF is a soft version of the NNSF
- The JPDAF considers **joint association events** and computes their probability. State update is like in the PDAF using a combined innovation
- The **MHT**, the optimal data association algorithm without pruning, maintains a growing tree of association hypotheses. It makes **hard** but **multiple** decisions and **delays** them until more evidence has arrived
- Data association is a **hard** problem and currently an active area of research. A promising approach not covered here is MCMC data association.

Sources and Further Reading

These slides partly follow the books of Bar-Shalom et al. [1] and Blackman [6]. The JPDAF trees are inspired by the nice lecture notes of Orguner [3]. A brief, AI/machine learning view on data association is given by Russell and Norvig [2] (chapter 15.6). More on PDAF and JPDAF can be found in [4], in particular also application examples. A comprehensive treatment of particle filter-based tracking techniques is given in Ristic et al. [5].

[1] Y. Bar-Shalom, X. Rong Li, T. Kirubarajan, “Estimation with Applications to Tracking and Navigation”, Wiley, 2001

[2] S. Russell, P. Norvig, “Artificial Intelligence: A Modern Approach”, 3rd edition, Prentice Hall, 2009. See http://aima.cs.berkeley.edu

[3] U. Orguner, “Target Tracking”, Lecture notes, Linköpings University, 2010. See https://www.control.isy.liu.se/student/graduate/TargetTracking

[4] Y. Bar-Shalom, F. Daum, and J. Huang, “The probabilistic data association filter,” IEEE control system magazine, 29(6), Dec. 2009.

[5] B. Ristic, S. Arulampalam, N. Gordon, “Beyond the Kalman Filter: Particle Filters for Tracking Applications” Artech House, 2004

[6] S.S. Blackman, R. Popoli, “Design and Analysis of Modern Tracking Systems”, Artech House, 1999