Chapter Contents

- Overview
- Validation Gate
- False Alarm Model
- Single-Target DA
  - NNSF
  - PDAF
- Multi-Target DA
  - NNSF
  - Global NNSF
  - Multiple Hypothesis Tracking (MHT)
- Summary
Data Association

“Data association is the process of associating uncertain measurements to known tracks.”

**Problem types**
- Track creation, maintenance, and deletion
- Single or multiple targets and sensors
- Imperfect target detection
- False alarms
- Target occlusions

**Approaches**
- **Bayesian:** compute a full (or approx.) distribution in DA space from priors, posterior beliefs, and observations
- **Non-Bayesian:** compute a maximum likelihood estimate from the possible set of DA solutions
Data Association

Overall procedure:

- **Make observations** (= measurements).
  Measurements can be raw data (e.g. processed radar signals) or the output of some target detector (e.g. people detector)

- **Predict the measurements** from the predicted tracks. This yields an area in sensor space where to expect an observation. The area is called **validation gate** and is used to narrow the search

- **Check if a measurement lies in the gate.**
  If yes, then it is a valid candidate for a pairing/match
Data Association

What makes this a difficult problem

- **Multiple targets**
- **False alarms**
- **Detection uncertainty** (occlusions, sensor failures, ...)
- **Ambiguities** (several measurements in the gate)
Measurement Prediction

- Measurement and measurement cov. prediction
  - This is typically a frame transformation into sensor space

\[ \hat{z}(k) = H(k)\hat{x}(k|k-1) \]
\[ \hat{R}(k) = H(k)\hat{P}(k|k-1)H^T(k) \]

- If only the position of the target is observed (typical case), the measurement matrix is

\[ z = [x \ y]^T \quad H = \begin{bmatrix} 1 & 0 & 0 & \cdots \end{bmatrix} \]

- Note: One can also observe
  - Velocity (Doppler radar)
  - Acceleration (accelerometers)
Validation Gate

- Assume that measurements are distributed according to a Gaussian, centered at the measurement prediction $\tilde{z}(k)$ with covariance $\tilde{S}(k)$

$$p(z(k)) = \mathcal{N}(z(k); \tilde{z}(k), \tilde{S}(k))$$

This is the measurement likelihood model

- Let further

$$d = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}$$

be the Mahalanobis distance between $x$ and $\mu$
Validation Gate

- Then, the measurements will be in the area
  \[
  \mathcal{V}(k, \gamma) = \{ z : (z - \bar{z})^T \hat{S}^{-1} (z - \bar{z}) \leq \gamma \}
  = \{ z : d^2 \leq \gamma \}
  \]
  with a probability defined by the gate threshold \( \gamma \) (omitting indices \( k \))

- This area is called validation gate

- The threshold is obtained from the inverse \( \chi^2 \) cumulative distribution at a significance level \( \alpha \)

- \( \chi^2 = \text{“chi square”} \)
Validation Gate

- The **shape** of the validation gate is a hyper-ellipsoid.
- This follows from setting

  \[ c = \frac{1}{(2\pi)^{k/2}|S|^{1/2}} \exp \left( -\frac{1}{2} (z - \bar{z})^T S^{-1} (z - \bar{z}) \right) \]

  leading to

  \[ c' = (z - \bar{z})^T S^{-1} (z - \bar{z}) \]

  which describes a **conic section** in matrix form

  \[ x^T Q x = 0 \]

  \[ x = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad x^T = [x, y, 1] \quad Q = \begin{bmatrix} A & B & D \\ B & C & E \\ D & E & F \end{bmatrix} \]

- The gate is a **iso-probability contour** obtained when intersecting a Gaussian with a hyper-plane.
Validation Gate

Why a $\chi^2$ distribution?

- Let $X_i$ be a set of $k$ i.i.d. standard normally distributed random variables, $X_i \sim \mathcal{N}(x; 0, 1)$. Then, the variable $Q$
  $$Q = \sum_{i=1}^{k} X_i^2$$
  follows a $\chi^2$ distribution with $k$ “degrees of freedom”

- We will now show that the Mahalanobis distance is a sum of squared standard normally distributed RVs.
Validation Gate in 1D

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

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

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

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

- Thus, $d^2 = y^2$ and is $\chi^2$ distributed with 1 degree of freedom
Validation Gate in ND

- Assume ND measurements and \( \mu = \hat{z}(k), \Sigma = \hat{S}(k) \)
- The Mahalanobis distance is then
  \[
  d^2 = (z - \mu)^T \Sigma^{-1} (z - \mu)
  \]
- By changing variables, \( y = C^{-1}(z - \mu), \Sigma = CC^T \) we have \( y \sim \mathcal{N}(0, I) \) and therefore
  \[
  d^2 = y^T I^{-1} y \quad \Rightarrow \quad d^2 = \sum_{i=1}^{k} y_i^2
  \]
  which is \( \chi^2 \) distributed with \( k \) degrees of freedom.
- \( C \) is obtained from a Cholesky decomposition
Validation Gate

Where does the threshold $\gamma$ come from?

- $\gamma$, often denoted $\chi^2_{k,\alpha}$, is taken from the inverse $\chi^2$ cumulative distribution at a level $\alpha$ and $k$ d.o.f.s.
- The values are typically given in tables, e.g. in most statistics books (or by the Matlab function `chi2inv`)
- Given the level $\alpha$, 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 $\alpha$ are 0.95 or 0.99
Validation Gate

Euclidian distance

Takes into account:
✔ Position
✖ Uncertainty
✖ Correlations

→ It seems that i-a and j-b belong together

d_{i-a} = 2.24

\[d_{i-b} = 3.54\]
\[d_{j-a} = 3.61\]
\[d_{j-b} = 1.58\]
Validation Gate

Mahalanobis distance with diagonal covariance matrices

Takes into account:

✓ Position
✓ Uncertainty
✗ Correlations

→ Now, i-b is “closer” than j-b
Validation Gate

Mahalanobis distance

Takes into account:
✓ Position
✓ Uncertainty
✓ Correlations

→ It’s actually i-b and j-a that belong together!

\[ d_{i-a} = 6.05 \]
\[ d_{i-b} = 2.77 \]
\[ d_{j-a} = 2.45 \]
\[ d_{j-b} = 4.78 \]
False Alarms

- False alarms are **false positives**
- They can come from sensor imperfections, detector failures, or clutter
- **Clutter** is “unwanted echoes”, e.g. atmospheric turbulences
- Thus, the questions:

  What’s inside the **gate**?
  - A measurement or
  - A false alarm?

How to **model false alarms**?
- Uniform over sensor space
- Independent across time
**False Alarm Model**

- Assume (temporarily) that the sensor field of view $V$ is discretized into $N$ discrete cells, $c_i, \ i = 1, \ldots, N$
- In each cell, false alarms occur with probability $P_F$
- Assume independence across cells
- The occurrence of false alarms is a Bernoulli process (flipping an unfair coin) with probability $p = P_F$
- Then, the number of false alarms $m_F$ follows a **Binomial distribution**

\[
P(K = m_F) = \binom{N}{m_F} p^{m_F} (1 - p)^{N-m_F}
\]

with expected value $Np$
False Alarm Model

- Let the spatial density $\lambda$ be the number of false alarms over space
  \[ \lambda = \frac{Np}{V} \]  
  \([\text{occurrences per m}^2]\)

- Let now $N \to \infty$, that is, we reduce the cell size until the continuous case. Then the Binomial becomes a Poisson distribution with
  \[ \mu_F(m_F) = e^{-\lambda V} \frac{(\lambda V)^m_F}{m_F!} \]

- The **measurement likelihood** of false alarms is assumed to be uniform,
  \[ p(z|z \text{ is a false alarm}) = \frac{1}{V} \]
Single Target Data Association

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

Data Association approaches

*Non-Bayesian*: no prior association probabilities
- Nearest neighbor Standard filter (NNSF)
- Track splitting filter

*Bayesian*: computes association probabilities
- Probabilistic Data Association Filter (PDAF)
Single Target DA: NNSF

Nearest Neighbor Standard Filter (NNSF)

1. Compute Mahalanobis distance to all measurements
2. Accept the closest measurement
3. Update the track as if it were the correct one

Problem: with some probability the selected measurement is not the correct one. An incorrect association can lead to overconfident covariances, filter divergence and track loss. Note: covariances will collapse in any case.

- Conservative NNSF variant:
  Do not associate in case of ambiguities
- Other variant: Strongest Neighbor Standard filter:
  Used, e.g., with sonar sensors
Probabilistic Data Association filter (PDAF)

- Computes the **probability** of track-to-measurement associations, thus a **Bayesian** data association technique
- Opposed e.g. to the NNSF that uses a **ML** criterion based on the minimum Mahalanobis distance

**Idea:** Instead of taking a hard decision, update the track with a **weighted average** of all validated measurements
- The weights being the **individual association probabilities**
Single Target DA: PDAF

Probabilistic Data Association filter

- Integrates all measurements in the validation gate
  - Conditioning the update on the association events

\[ \theta_i(k) = \begin{cases} 
  z_i(k) \text{ is the correct measurement} & i = 1, \ldots, m(k) \\
  \text{no correct measurement is present} & i = 0 
\end{cases} \]

- \( \beta_i \triangleq P(\theta_i|Z^k) \) is the association probability

- Assumption: At most one of the validated measurements comes from the target. All others are independently and uniformly distributed
Single Target DA: PDAF

- **Association probability** $\beta_i \triangleq P(\theta_i|Z^k)$ for a Poisson false alarm model is

$$
\beta_i(k) = \begin{cases} 
\frac{e_i}{b + \sum_{j=1}^{\mu_F}} & i = 1, \ldots, m(k) \\
\frac{b}{b + \sum_{j=1}^{\mu_F}} & i = 0 
\end{cases}
$$

- $e_i = \mu_F (m(k) - 1) \cdot P_D P_G \cdot P_G^{-1} \mathcal{N}(\nu_i(k); 0, \hat{S}(k))$
- $b = \mu_F (m(k))(1 - P_D P_G)$

- **Intuition:** depends on the number of validated measurements $m(k)$ versus the false alarms rate, the detection probability $P_D$ of the target, the probability $P_G$ that the target detection falls into the gate, and the individual innovations

(Derivation skipped)
Mixture Distributions

To understand the PDAF state update expressions, we recall some basics:

- **Mixture distributions**
  - A mixture pdf is a weighted sum of pdfs with the weights summing up to 1.
  - Consider a Gaussian mixture
    \[
    p(x) = \sum_{i=1}^{n} p_i \mathcal{N}(x; \bar{x}_i, P_i)
    \]
    with events \( A_i = \{ x \sim \mathcal{N}(\bar{x}_i, P_i) \} \). Then
    \[
    p(x) = \sum_{i=1}^{n} P\{A_i\} p(x|A_i) = \sum_{i=1}^{n} p_i \ p(x|A_i)
    \]
    with the events being mutually exclusive and exhaustive.
Mixture Distributions

- **Conditional expectation**
  \[ E[x | A_i] = \bar{x}_i \]
  \[ E[(x - \bar{x}_i)(x - \bar{x}_i)^T | A_i] = P_i \]

- **Moments of a mixture**
  - **Mean**
    \[ \bar{x} = \sum_{i=1}^{n} p_i \bar{x}_i \]
  - **Covariance**
    \[ E[(x - \bar{x})(x - \bar{x})^T] = \sum_{i=1}^{n} E[(x - \bar{x})(x - \bar{x})^T | A_i] p_i \]
Mixture Distributions

- **Moments of a mixture**: Covariance

\[
E[(x - \bar{x})(x - \bar{x})^T] = \sum_{i=1}^{\infty} E[(x - \bar{x})(x - \bar{x})^T|A_i] \ p_i
\]

\[
= \sum E[(x - \bar{x}_i + \bar{x}_i - \bar{x})(x - \bar{x}_i + \bar{x}_i - \bar{x})^T|A_i] \ p_i
\]
Mixture Distributions

- **Moments of a mixture**: Covariance

\[
E[(x - \bar{x})(x - \bar{x})^T] = \sum_{i=1}^{\infty} E[(x - \bar{x})(x - \bar{x})^T | A_i] \ p_i
\]

\[
= \sum E[(x - \bar{x}_i + \bar{x}_i - \bar{x})(x - \bar{x}_i + \bar{x}_i - \bar{x})^T | A_i] \ p_i
\]

\[
= \sum E[((x - \bar{x}_i) + (\bar{x}_i - \bar{x}))(x - \bar{x}_i + (\bar{x}_i - \bar{x}))^T | A_i] \ p_i
\]
Mixture Distributions

- Moments of a mixture: Covariance

\[
E[(x - \bar{x})(x - \bar{x})^T] = \sum_{i=1}^{n} E[(x - \bar{x})(x - \bar{x})^T | A_i] p_i \\
= \sum E[(x - \bar{x}_i + \bar{x}_i - \bar{x})(x - \bar{x}_i + \bar{x}_i - \bar{x})^T | A_i] p_i \\
= \sum E[((x - \bar{x}_i) + (\bar{x}_i - \bar{x}))(\bar{x}_i - \bar{x})^T | A_i] p_i \\
= \sum E[(x - \bar{x}_i)(x - \bar{x}_i)^T | A_i] p_i \\
+ E[(x - \bar{x}_i)(\bar{x}_i - \bar{x})^T | A_i] p_i + E[(\bar{x}_i - \bar{x})(x - \bar{x}_i)^T | A_i] p_i \\
+ E[(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T | A_i] p_i
\]
Mixture Distributions

- **Moments of a mixture:** Covariance

\[
E[(x - \bar{x})(x - \bar{x})^T] = \sum_{i=1}^{\pi} E[(x - \bar{x}_i)(x - \bar{x})^T | A_i] p_i \\
= \sum E[(x - \bar{x}_i + \bar{x}_i - \bar{x})(x - \bar{x}_i + \bar{x}_i - \bar{x})^T | A_i] p_i \\
= \sum E[((x - \bar{x}_i) + (\bar{x}_i - \bar{x}))(x - \bar{x}_i) + (\bar{x}_i - \bar{x}))^T | A_i] p_i \\
= \sum E[(x - \bar{x}_i)(x - \bar{x}_i)^T | A_i] p_i \\
+ E[(x - \bar{x}_i)(\bar{x}_i - \bar{x})^T | A_i] p_i + E[(\bar{x}_i - \bar{x})(x - \bar{x}_i)^T | A_i] p_i \\
+ E[(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T | A_i] p_i \\
= \sum p_i P_i \\
+ (E[x | A_i] - \bar{x}_i)(\bar{x}_i - \bar{x})^T p_i + (\bar{x}_i - \bar{x})(E[x | A_i] - \bar{x}_i)^T p_i \\
+ (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T p_i
\]
Mixture Distributions

- **Moments of a mixture:** Covariance

\[
E[(x - \bar{x})(x - \bar{x})^T] = \sum_{i=1}^{\infty} E[(x - \bar{x})(x - \bar{x})^T | A_i] p_i
\]

\[
= \sum E[(x - \bar{x}_i + \bar{x}_i - \bar{x})(x - \bar{x}_i + \bar{x}_i - \bar{x})^T | A_i] p_i
\]

\[
= \sum E[((x - \bar{x}_i) + (\bar{x}_i - \bar{x}))( (x - \bar{x}_i) + (\bar{x}_i - \bar{x}))^T | A_i] p_i
\]

\[
= \sum E[(x - \bar{x}_i)(x - \bar{x}_i)^T | A_i] p_i
+ E[(x - \bar{x}_i)(\bar{x}_i - \bar{x})^T | A_i] p_i + E[(\bar{x}_i - \bar{x})(x - \bar{x}_i)^T | A_i] p_i
+ E[(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T | A_i] p_i
\]

\[
= \sum p_i P_i
+ (E[x | A_i] - \bar{x}_i)(\bar{x}_i - \bar{x})^T p_i + (\bar{x}_i - \bar{x})(E[x | A_i] - \bar{x}_i)^T p_i
+ (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T p_i
\]

\[
= \sum p_i P_i + \sum (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T p_i
\]
Mixture Distributions

- **Moments of a mixture:** Covariance

\[
E[(x - \bar{x})(x - \bar{x})^T] = \sum_{i=1}^{n} E[(x - \bar{x}_i)(x - \bar{x}_i)^T | A_i] p_i \\
= \sum_{i=1}^{n} E[(x - \bar{x}_i + \bar{x}_i - \bar{x})(x - \bar{x}_i + \bar{x}_i - \bar{x})^T | A_i] p_i \\
= \sum_{i=1}^{n} E[((x - \bar{x}_i) + (\bar{x}_i - \bar{x}))(x - \bar{x}_i) + (\bar{x}_i - \bar{x})^T | A_i] p_i \\
= \sum_{i=1}^{n} E[(x - \bar{x}_i)(x - \bar{x}_i)^T | A_i] p_i \\
+ E[(x - \bar{x}_i)(\bar{x}_i - \bar{x})^T | A_i] p_i + E[(\bar{x}_i - \bar{x})(x - \bar{x}_i)^T | A_i] p_i \\
+ E[(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T | A_i] p_i \\
= \sum_{i=1}^{n} p_i P_i \\
+ (E[x | A_i] - \bar{x}_i)(\bar{x}_i - \bar{x})^T p_i + (\bar{x}_i - \bar{x})(E[x | A_i] - \bar{x}_i)^T p_i \\
+ (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T p_i \\
= \sum_{i=1}^{n} p_i P_i + \sum_{i=1}^{n} (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T p_i \\
= \sum_{i=1}^{n} p_i P_i + \bar{P}
Mixture Distributions

- **Moments of a mixture:** Spread of the means

\[
\tilde{P} = \sum_{i=1}^{n} (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T p_i
\]

Note resemblance to the **sample covariance matrix**

- **Alternative expression**

\[
\tilde{P} = \sum (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T p_i
\]

\[
= \sum (\bar{x}_i \bar{x}_i^T - \bar{x}_i \bar{x}^T - \bar{x} \bar{x}_i^T + \bar{x} \bar{x}^T) p_i
\]

\[
= \sum \bar{x}_i \bar{x}_i^T p_i - \sum \bar{x}_i \bar{x}^T p_i - \sum \bar{x} \bar{x}_i^T p_i + \sum \bar{x} \bar{x}^T p_i
\]

\[
= \sum \bar{x}_i \bar{x}_i^T p_i - \sum p_i \bar{x}_i \bar{x}^T - \bar{x} \sum p_i \bar{x}_i^T + \bar{x} \bar{x}^T \sum p_i
\]

\[
= \sum \bar{x}_i \bar{x}_i^T p_i - \bar{x} \bar{x}^T - \bar{x} \bar{x}^T + \bar{x} \bar{x}^T
\]

\[
= \sum p_i \bar{x}_i \bar{x}_i^T - \bar{x} \bar{x}^T
\]
State update

\[
\hat{x}(k|k) = \hat{x}(k|k - 1) + K(k) \nu(k)
\]

- With the combined innovation

\[
\nu(k) = \sum_{i=1}^{m} \beta_i(k) \nu_i(k)
\]

summed over all \(m\) association events \(\theta_i(k)\).

The events \(\theta_i(k)\) are assumed to be exhaustive (their probabilities sums up to one) and mutually exclusive (they cannot occur at the same time)
Single Target DA: PDAF

Covariance update

\[ P(k|k) = E[(x - \hat{x}(k|k))(x - \hat{x}(k|k))^T] \]

\[ = \sum_{i=0}^{m} E[(x - \hat{x}(k|k))(x - \hat{x}(k|k))^T | \theta_i] \beta_i \]

\[ = \sum_{i=0}^{m} \beta_i P_i(k|k) + \tilde{P} \]

For \( i = 0 \) (no correct measurement), we have

\[ P_0(k|k) = P(k|k - 1) \]

while for \( i \neq 0 \) (one of the \( z_i \)'s is the correct measurement)

\[ P_i(k|k) = P(k|k) = (I - K(k)H(k)) P(k|k - 1) \]
Single Target DA: PDAF

Covariance update

Therefore, with $\beta_0(k) = 1 - \sum_{i=1}^{m} \beta_i(k)$ we get

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

The last term is obtained as follows. Starting from

$$\tilde{P} = \sum_{i=0}^{m} \beta_i \hat{x}_i(k|k) \hat{x}_i(k|k)^T - \hat{x}(k|k) \hat{x}(k|k)^T$$

we substitute

$$\hat{x}_i(k|k) = \hat{x}(k|k - 1) + K(k) \nu_i(k)$$

$$\hat{x}(k|k) = \hat{x}(k|k - 1) + K(k) \nu(k)$$
Covariance update

... and, over some intermediate steps, arrive at

\[
\tilde{P} = K(k) \left[ \sum_{i=0}^{m} \beta_i \nu_i(k) \nu_i(k)^T - \nu(k) \nu(k)^T \right] K(k)^T
\]

- This is the (weighted) spread of the innovations term
- Error propagation from the measurement space into the state space across the Kalman gain
- It is positive semidefinite (a sum of dyads $a \cdot a^T$ with positive weighting)
Single Target DA: PDAF

Covariance update

\[ P(k|k) = \beta_0(k)P(k|k - 1) + (1 - \beta_0(k))P(k|k) + \tilde{P}(k) \]

- With probability \( \beta_0(k) \) none of the measurements is correct, the predicted covariance appears with this weighting ("no update")

- With probability \( (1 - \beta_0(k)) \) the correct measurement is available and the posterior covariance appears with this weighting

- Since it is unknown which if the measurements is correct, the term \( \tilde{P} \) increases the covariance to account for the origin uncertainty
Single Target DA: PDAF

- All **other** calculations in the PDAF
  - State prediction
  - Covariance prediction
  - Innovation covariance
  - Kalman gain

are done 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
Single Target DA: PDAF

- Example

- Tracking in presence of false alarms and mis-detections ($P_D < 1$)
- At $k = 7$ there is no target detection but a false alarm
- The PDAF, accounting for the origin uncertainty, has a large validation gate
- The NNSF-tracker loses the target
Single Target DA: Wrap Up

- The NNSF takes a **hard** association decision
  - This hard decision is sometimes correct and sometimes wrong

- The PDAF relies on a **soft** decision since it averages over all the association possibilities
  - This soft decision is never totally correct but never totally wrong

- This is why **the PDAF is a suboptimal** strategy
  - To be precise: the PDAF is suboptimal since it approximates the conditional pdf of the target’s state at every stage as a Gaussian with moments matched to the mixture

\[
p(x \mid Z) = \sum_{A_i \in \mathcal{A}} p(x \mid A_i, Z) \beta_i
\]
Multi-Target Data Association

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

Data Association approaches
- **Non Bayesian**: ML criteria
  - NNSF, Global NNSF
- **Bayesian**: compute association probabilities
  - JPDAF, MHT, MCMC
Multi-Target DA: NNSF

Nearest Neighbor Standard Filter (NNSF)

1. Build the assignment matrix $A = \begin{bmatrix} d_{ij}^2 \end{bmatrix}$ with

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

- Rectangular

$$A = \begin{bmatrix} d_{11}^2 & d_{12}^2 & d_{13}^2 & d_{14}^2 & d_{15}^2 & d_{16}^2 & d_{17}^2 & d_{18}^2 \\ d_{21}^2 & d_{22}^2 & d_{23}^2 & d_{24}^2 & d_{25}^2 & d_{26}^2 & d_{27}^2 & d_{28}^2 \end{bmatrix}$$

- Square

Multi-Target DA: NNSF

Nearest Neighbor Standard Filter (NNSF)

1. Build the assignment matrix $A = [d_{ij}^2]$ with
   $$d_{ij}^2 = v_{ij}(k)^T S_j^{-1}(k) v_{ij}(k)$$

2. Iterate
   - Find the minimum cost assignment in $A$
   - Remove the row and column of that assignment

3. Check if assignment is in the validation regions
   - Unassociated tracks can be used for track deletion
   - Unassociated meas. can be used for track creation

- Problem: Does generally not find the global minimum
- Conservative variant: no association in case of ambiguities
Multi-Target DA: Global NNSF

1. Build the assignment matrix \( A = [d_{ij}^2] \) with
   \[
d_{ij}^2 = \nu_{ij}(k)^T S_j^{-1}(k) \nu_{ij}(k)
   \]

2. Solve the **linear assignment problem**
   \[
   \min \sum d_{ij}^2 \cdot x_{ij} \quad x_{ij} \in \{0, 1\}
   \]
   \[
   \sum_i x_{ij} = 1 \quad \sum_j x_{ij} = 1
   \]
   - **Hungarian** method for square matrices
   - **Munkres** algorithm for rectangular matrices

3. Check if assignments are in the validation gate

Performs DA jointly, finds **global optimum**.
Multi-Target DA: Global NNSF

Linear assignment problem

- Is one of the most famous problems in linear programming and in combinatorial optimization
- Used to find 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 methods exist. The **Hungarian method**, while not the most efficient one, is still a popular algorithm
- Can also be solved for non-square problems by **Munkres' algorithm**
Problem statement:

We are given an \( n \times n \) 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, and the sum of the corresponding costs

\[
\sum_i x_{ij} = 1 \quad \sum_j x_{ij} = 1
\]

and the sum of the corresponding costs

\[
\min \sum d_{ij}^2 \cdot x_{ij} \quad x_{ij} \in \{0, 1\}
\]

is a minimum.
Multi-Target DA: Global NNSF

Example: NNSF versus Global NNSF

Which is the best assignment?

△ Observations  ⊕ Predictions
Multi-Target DA: Global NNSF

Example: NNSF versus Global NNSF

△ Observations  ⊕ Predictions

NNSF:
Greedy
Multi-Target DA: Global NNSF

Example: NNSF versus Global NNSF

NNSF:
Greedy

△ Observations  ⊕ Predictions
Multi-Target DA: Global NNSF

Example: NNSF versus Global NNSF

Global NNSF:
Globally optimal

\[ \triangle \text{ Observations} \quad \color{purple} \pm \text{ Predictions} \]
Multi-Target DA: MHT

- All DA methods considered so far are **single-frame**
- Hard or soft **decisions** are taken after **each step**
- In the presence of false alarms, misdetections, maneuvers and lengthy occlusion events, this is an **error-prone** strategy

- We want to **delay decisions** until sufficient information has arrived
- This implies the maintenance of multiple histories of **hypothetical** data association decisions in **parallel**

- **Multiple Hypothesis Tracking (MHT)**
Multi-Target DA: MHT

Multiple Hypothesis Tracking

- The number of association histories grows **exponentially**
- Growth yields a **hypothesis tree**
- **Pruning** strategies are mandatory in practice
- **Optimal Bayesian solution** (without pruning)

- In addition to the measurement-to-track associations, the MHT can also reason about **track interpretations** as
  - Occluded (label O)
  - Deleted (label D)

and **measurement interpretations** as
  - False alarms (label F)
  - New tracks (label N)

- Interpretations are like associations to fixed labels
Multi-Target DA: MHT

$\Theta^k_i$: Hypothesis

$\Theta^{k-1}_{p(i)}$: Parent Hypothesis

$\theta_{c(i)}(k)$: Assignment Set

$\theta_{c(i)}(k)$:

\[
\begin{array}{c|c|c|c|c|c}
   & t_1 & t_2 & t_3 & F & N \\
\hline
z_1 & 0 & 0 & 0 & 0 & 1 \\
\hline
z_2 & 1 & 0 & 0 & 0 & 0 \\
z_3 & 0 & 0 & 1 & 0 & 0 \\
\end{array}
\]

$\{\{z_1, N\}, \{z_2, t_1\}, \{z_3, t_3\}\}$
Multi-Target DA: 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 additional track management system is needed

- Which is then the **best hypothesis**?
  - Compute **probabilities for hypotheses**
  - This is done in a **recursive Bayesian** fashion
  - **Best** hypothesis is, for instance, the one with the **highest** probability

- Yields a **probability distribution over hypotheses**
MHT Example
MHT Example (Detail)
Multi-Target DA: MHT

- The **probability of an hypothesis** \( \Theta_i^k = \{ \Theta_{p(i)}^{k-1}, \theta_c(i)(k) \} \) can be calculated using Bayes rules.

\[
P(\Theta_i^k | Z^k) = P(\Theta_{p(i)}^{k-1}, \theta_c(i)(k) | Z^k) =
\]
\[
\frac{1}{\eta} \cdot p(Z(k) | \Theta_{p(i)}^{k-1}, \theta_c(i)(k), Z^{k-1}) \cdot P(\theta_c(i)(k) | \Theta_{p(i)}^{k-1}, Z^k) \cdot P(\Theta_{p(i)}^{k-1} | Z^{k-1})
\]

- **Likelihood**
- **Assignment probability**
- **Prior**
Multi-Target DA: MHT

- **Measurement likelihood**

\[
p(Z(k)|\Theta^{k-1}, \theta(k), Z^{k-1}) = \prod_{l=1}^{m(k)} p(z_l(k)|\Theta^{k-1}, \theta(k), Z^{k-1})
\]

- **Case 1: associated with track t**

\[
p(z_l(k)|\Theta^{k-1}, \theta(k), Z^{k-1}) = \mathcal{N}(z_l(k); \hat{z}_t(k|k-1), S_t(k))
\]

- **Case 2: false alarm**

\[
p(z_l(k)|\Theta^{k-1}, \theta(k), Z^{k-1}) = V^{-1}
\]

- **Case 3: new track**

\[
p(z_l(k)|\Theta^{k-1}, \theta(k), Z^{k-1}) = V^{-1}
\]
Multi-Target DA: MHT

- Assignment probability

\[ P(\theta(k)|\Theta^{k-1}, Z^k) = P(\theta(k)|N_M, N_O, N_D, N_N, N_F) \cdot P(N_M, N_O, N_D, N_N, N_F|\theta(k), \Theta^{k-1}) \]

- \( P(N_M, N_O, N_D, N_N, N_F|\theta(k), \Theta^{k-1}) \) is the probability of having \( N_M \) matched tracks, \( N_O \) occluded tracks, \( N_D \) deleted tracks, \( N_N \) false alarm and \( N_F \) new tracks

- \( P(\theta(k)|N_M, N_O, N_D, N_N, N_F) \) is the probability of a possible configuration \( \theta(k) \) given the number of events defined before
Multi-Target DA: MHT

- Assignment probability 1: \( P(N_M, N_O, N_D, N_N, N_F | \theta(k), \Theta^{k-1}) \)
  - Assuming a multinomial distribution for track labels
    \[
    P(N_M, N_O, N_D | \theta(k), \Theta^{k-1}) = \frac{N_T!}{N_M! N_O! N_D!} p_M^{N_M} p_O^{N_O} p_D^{N_D}
    \]
  - Assuming a Poisson distribution for new tracks
    \[
    P(N_N | \theta(k), \Theta^{k-1}) = \frac{(V \lambda_N)^{N_N} e^{-V \lambda_N}}{N_N!}
    \]
  - Assuming a Poisson distribution for false alarm
    \[
    P(N_F | \theta(k), \Theta^{k-1}) = \frac{(V \lambda_F)^{N_F} e^{-V \lambda_F}}{N_F!}
    \]
  - We obtain
    \[
    P(\cdot) = \frac{N_T! (e^{-V \lambda_N})(e^{-V \lambda_F})}{N_N! N_F! N_M! N_O! N_D!} (V \lambda_N)^{N_N} (V \lambda_F)^{N_F} p_M^{N_M} p_O^{N_O} p_D^{N_D}
    \]
Multi-Target DA: MHT

- **Assignment probability 2:** $P(\theta(k)|N_M, N_O, N_D, N_N, N_F)$
  - The possible choices of $m(k)$ taken as matched tracks:
    $$\binom{m(k)}{N_M} \cdot \text{Perm}(N_M, N_T) = \frac{m(k)!}{N_M!(m(k) - N_M)! (N_T - N_M)!}$$
  - The combinations of $m(k) - N_M$ taken as new tracks or false alarms:
    $$\binom{m(k) - N_M}{N_N} \cdot \binom{m(k) - N_M}{N_F} = \frac{(m(k) - N_M)!}{N_N!(m(k) - N_M - N_N)!}$$
  - The combinations of $N_T - N_M$ taken as occluded or deleted:
    $$\binom{N_T - N_M}{N_O} \cdot \binom{N_T - N_M}{N_D} = \frac{(N_T - N_M)!}{N_O!(N_T - N_M - N_O)!}$$
  - The probability is 1 over all the possible choices:
    $$\left[ \frac{m(k)!}{N_M!(m(k) - N_M)! (N_T - N_M)!} \cdot \frac{N_T!}{N_N!(m(k) - N_M - N_N)!} \cdot \frac{(m(k) - N_M)!}{N_O!(N_T - N_M - N_O)!} \right]^{-1}$$
Multi-Target DA: MHT

- **Assignment probability 2:** $P(\theta(k)|N_M, N_O, N_D, N_N, N_F)$
  - The possible choices of $N_M$ taken as matched tracks
    \[
    \binom{m(k)}{N_M} \text{Perm}(N_M, N_T) = \frac{m(k)!}{N_M!(m(k) - N_M)! (N_T - N_M)!}
    \]
  - The combinations of $N_N$ taken as new tracks or false alarms
    \[
    \binom{m(k) - N_M}{N_N} \binom{m(k) - N_M - N_N}{N_F} = \frac{(m(k) - N_M)!}{N_N!(m(k) - N_M - N_N)!}
    \]
  - The combinations of $N_O$ taken as occluded or deleted
    \[
    \binom{N_T - N_M}{N_O} \binom{N_T - N_M}{N_D} = \frac{(N_T - N_M)!}{N_O!(N_T - N_M - N_O)!}
    \]
  - The probability is 1 over all the possible choices

\[
\begin{bmatrix}
\frac{m(k)!}{N_M!(m(k) - N_M)!} & \frac{N_T!}{(N_T - N_M)!} & \frac{(m(k) - N_M)!}{N_N!(m(k) - N_M - N_N)!} & \frac{(N_T - N_M)!}{N_O!(N_T - N_M - N_O)!}
\end{bmatrix}^{-1}
\]
Multi-Target DA: MHT

- **Assignment probability 2:** \( P(\theta(k)|N_M, N_O, N_D, N_N, N_F) \)
  - The possible choices of \( m(k) \) taken as matched tracks
    \[
    \binom{m(k)}{N_M} \text{Perm}(N_M, N_T) = \frac{m(k)!}{N_M!(m(k) - N_M)! (N_T - N_M)!} \frac{N_T!}{N_T}
    \]
  - The combinations of \( m(k) - N_M \) \( N_N \) taken as new tracks or false alarms
    \[
    \binom{m(k) - N_M}{N_N} \binom{m(k) - N_M - N_N}{N_F} = \frac{(m(k) - N_M)!}{N_N!(m(k) - N_M - N_N)!}
    \]
  - The combinations of \( N_T - N_M \) \( N_O \) taken as occluded or deleted
    \[
    \binom{N_T - N_M}{N_O} \binom{N_T - N_M - N_O}{N_D} = \frac{(N_T - N_M)!}{N_O!(N_T - N_M - N_O)!}
    \]
  - The probability is 1 over all the possible choices
    \[
    P(\theta(k)|N_M, N_O, N_D, N_N, N_F) = \frac{N_M!N_N!N_F!N_O!N_D!}{m(k)!N_T!}
    \]
Multi-Target DA: MHT

- **Assignment probability**

\[
P(\theta(k)|\Theta^{k-1}, Z^k) = P(\theta(k)|N_M, N_O, N_D, N_N, N_F) \cdot P(N_M, N_O, N_D, N_N, N_F|\theta(k), \Theta^{k-1})
\]

- **Putting everything together:**

\[
P(\theta(k)|\Theta^{k-1}, Z^k) = \frac{N_T!(e^{-V\lambda_N})(e^{-V\lambda_F})}{N_N!N_F!N_M!N_O!N_D!}(V\lambda_N)^{N_N}(V\lambda_F)^{N_F}p_M^{N_M}p_O^{N_O}p_D^{N_D} \frac{N_M!N_N!N_F!N_O!N_D!}{m(k)!N_T!}
\]
Multi-Target DA: MHT

- **Assignment probability**

\[
P(\theta(k)|\Theta^{k-1}, Z^k) = P(\theta(k)|N_M, N_O, N_D, N_N, N_F) \cdot P(N_M, N_O, N_D, N_N, N_F|\theta(k), \Theta^{k-1})
\]

- **Putting everything together:**

\[
P(\theta(k)|\Theta^{k-1}, Z^k) = \frac{N_T!(e^{-V\lambda_N})(e^{-V\lambda_F})}{N_N!N_T!N_M!N_O!N_D!} (V\lambda_N)^{N_N} (V\lambda_F)^{N_F} p_M^{N_M} p_O^{N_O} p_D^{N_D} \frac{N_M!N_N!N_F!N_O!N_D!}{m(k)!N_T!}
\]

- **Simplifying the expression we obtain**

\[
P(\theta(k)|\Theta^{k-1}, Z^k) = \frac{(e^{-V\lambda_N})(e^{-V\lambda_F})}{m(k)!} (V\lambda_N)^{N_N} (V\lambda_F)^{N_F} p_M^{N_M} p_O^{N_O} p_D^{N_D}
\]
Multi-Target DA: MHT Pruning

- **Clustering** spatially disjoint hypothesis trees
  - Tracks are partitioned into clusters based on gating
  - A separate tree is grown for each cluster

- **K-best** hypothesis tree
  - Directly generate the $k$-best hypothesis
  - Generation and evaluation in a single step by Murty's algorithm and a linear assignment solver
  - Implements a *generate-while-prune* versus a *generate-then-prune* strategy

- **N-Scan back** pruning
  - Ambiguities are assumed to be resolved after $N$ steps
  - Children at step $k+N$ give the prob. of parents at step $k$
  - Keep only the most probable branch
Multi-Target DA: MHT Example

- People tracking in RGB-D data (three MS Kinect)
Multi-Target DA: MHT Example

- People tracking in 3D range data (Velodyne scanner)
Summary

- The validation gate is a **region of acceptance** such that $100(1 - \alpha)\%$ of **true measurements** are **rejected**
- False alarms are assumed to occur according to a Poisson distribution with rate lambda and uniformly in space

- The NNSF is **simple** to implement but **greedy** and takes **hard** decisions. Good only if DA ambiguity is low
- The PDAF is a Bayesian DA method that takes **soft** decisions by incorporating all validated measurements into a **mixture** distribution
Summary

- The NNSF works also for multiple targets. Same advantages and drawbacks.
- The global NN is formulated as a linear assignment problem, solved using e.g. the Hungarian method.
- The GNN finds the jointly optimal assignment in a multi-target setting.
- The MHT is a multi-frame DA method with delayed decision making.
- Maintains multiple histories of association decisions (hypotheses), computes a probabilities for them.
- Optimal Bayesian method (up to pruning).
- Implementations of PDAF and MHT are used in many real-world applications (e.g. air traffic control).
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," Daniel H. Wilson, Bloomsbury, 2005