Graph-Based Multiobject Tracking

Florian Meyer
joint work with Jason Williams, Paolo Braca, Peter Willett, and Franz Hlawatsch

Scripps Institution of Oceanography
Electrical and Computer Engineering Department
University of California San Diego

Graph-Based Multiobject Tracking

Part 1: Probabilistic Data Association

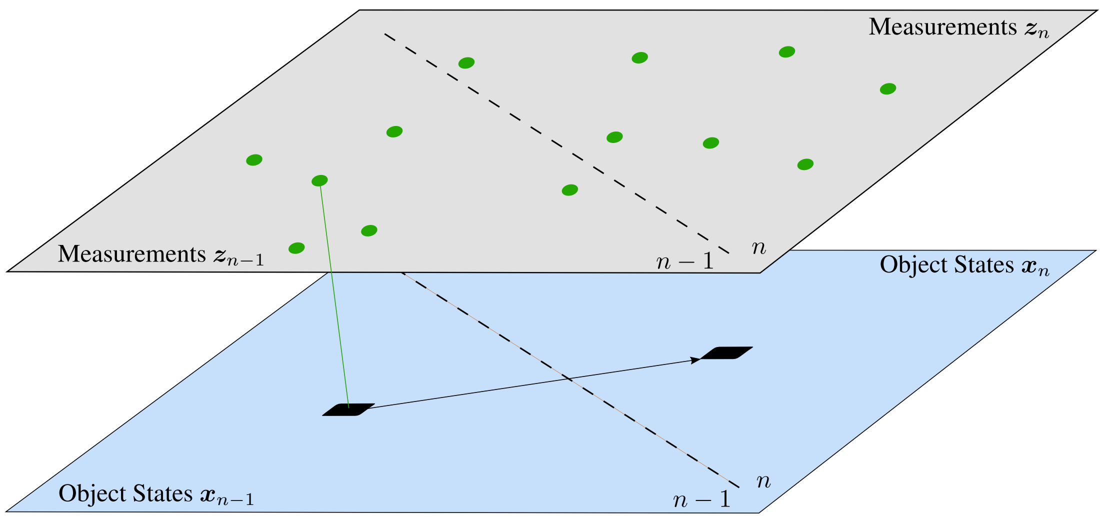
Florian Meyer
joint work with Jason Williams, Paolo Braca, Peter Willett, and Franz Hlawatsch

Scripps Institution of Oceanography
Electrical and Computer Engineering Department
University of California San Diego

UC San Diego
JACOBS SCHOOL OF ENGINEERING

Basic Setting

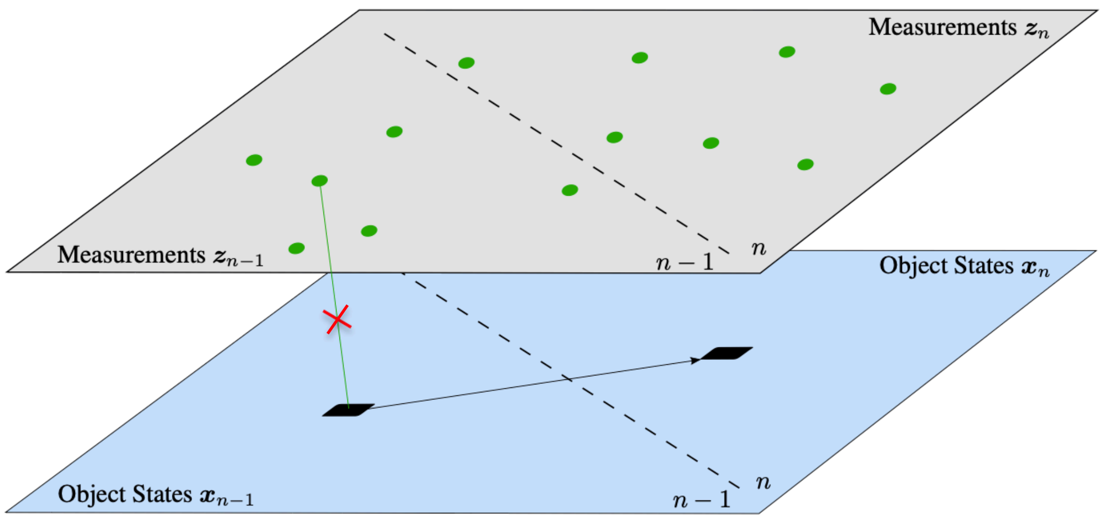
- "Single object tracking in clutter" problem



Basic Setting

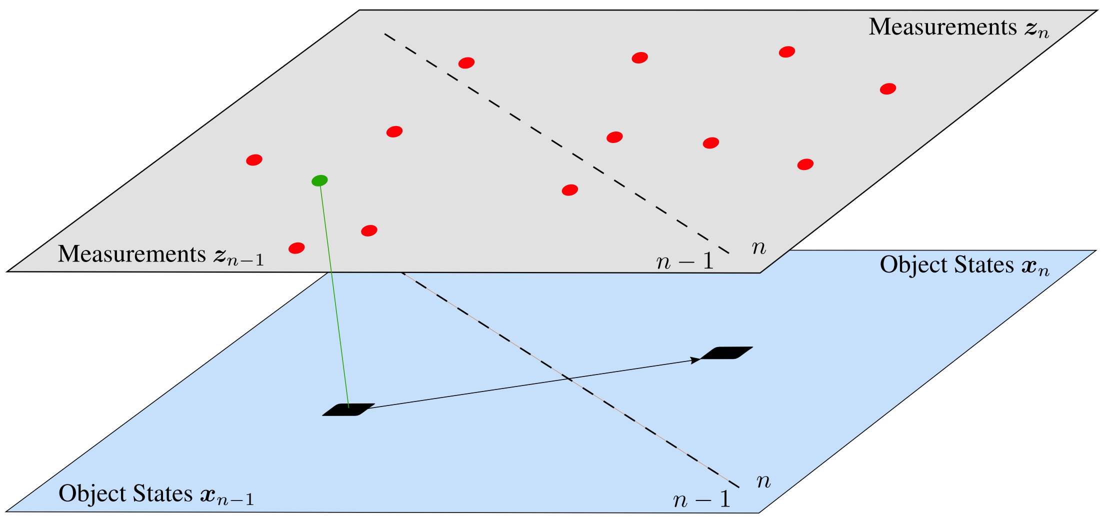
- “Single object tracking in clutter” problem

- Measurement-origin uncertainty (MOU), false clutter measurements and missed detections



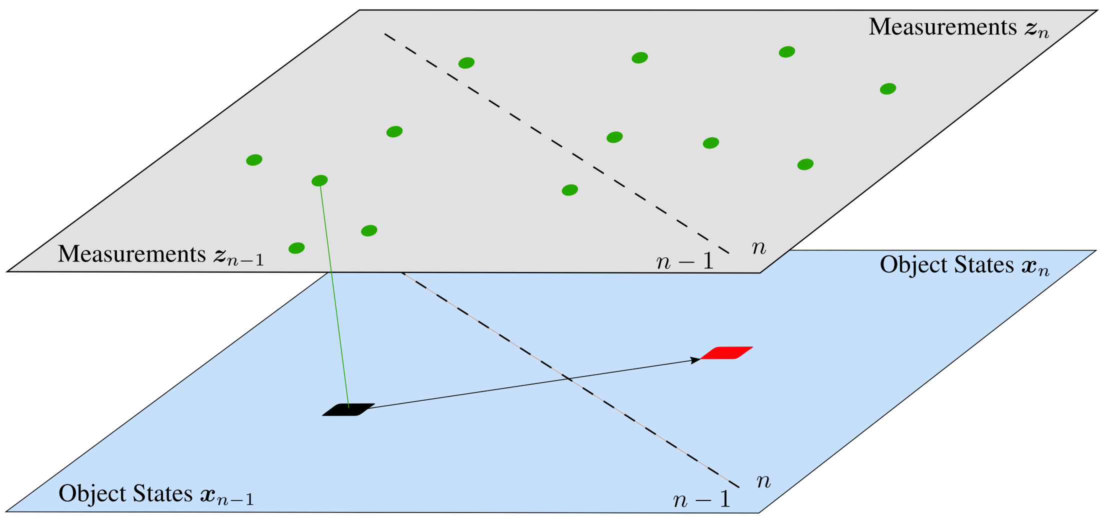
Basic Setting

- “Single object tracking in clutter” problem
- Measurement-origin uncertainty (MOU), false clutter measurements and missed detections



Basic Setting

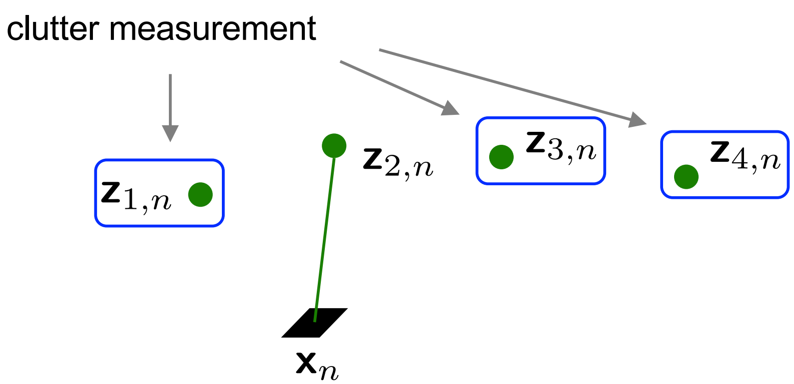
- “Single object tracking in clutter” problem
- Measurement-origin uncertainty (MOU), false clutter measurements and missed detections



The Association Variable

- Object-oriented association variable $a_n \in \{0, 1, \ldots, \boxed{M_n}\}$ number of measurements
  - $a_n = m > 0$: at time n, the object generates the measurement with index m

- Example 1:

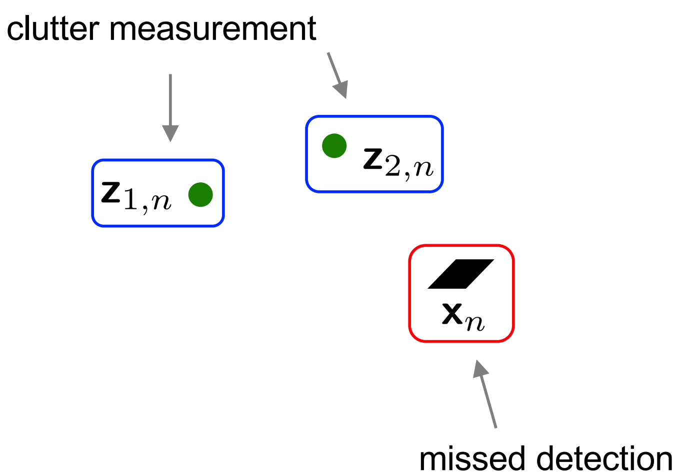


$M_n = 4$ $\qquad a_n = 2$

The Association Variable

- Object-oriented association variable $a_n \in \{0, 1, \ldots, M_n\}$
  - $a_n = m > 0$: at time n, the object generates the measurement with index m
  - $a_n = 0$: at time n, the object did not generate a measurement

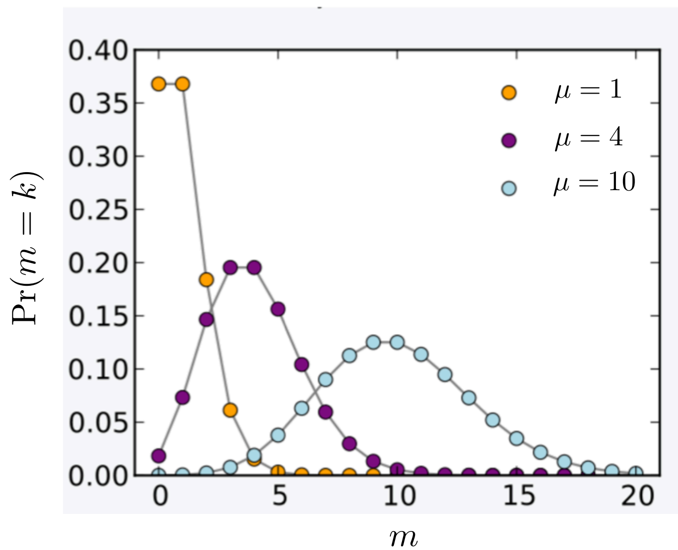
- Example 2:



$M_n = 2$ $\qquad$ $a_n = 0$

The Poisson Distribution

- A discrete random variable m is said to have a Poisson distribution with parameter $\mu > 0$, if for $m = 0, 1, 2, \ldots$ the probability mass function is given by



$$p(m) = \frac{\mu^m \, \mathrm{e}^{-\mu}}{m!}$$

The parameter μ is the mean as well as the variance

Single Object Tracking in Clutter

- The state of the object is denoted $\boldsymbol{x}_n \in \mathbb{R}^w$ and the joint measurement is given by $\boldsymbol{z}_n \triangleq [\boldsymbol{z}_{1,n}^{\mathrm{T}}, \boldsymbol{z}_{2,n}^{\mathrm{T}}, \ldots, \boldsymbol{z}_{M_n,n}^{\mathrm{T}}]^{\mathrm{T}}$ with entries $\boldsymbol{z}_{m,n} \in \mathbb{R}^d$

- The association variable a_n is given by
  - $m \in \{1, 2, \ldots, M_n\}$, if measurement $\boldsymbol{z}_{m,n}$ was generated by the object
  - 0, if no measurement was generated by the object

- Association variable a_n and the number of measurements M_n are random variables

Prior Distribution

- Key Assumptions I:
  - Object detection $\theta_n \in \{0,1\}$ is a Bernoulli trial with success probability $0 < p_\mathrm{d} \leqslant 1$
  - The number of clutter measurements L_n is Poisson distributed with mean μ_c
  - At most one measurement $\boldsymbol{z}_{m,n}$ is generated by the object

- Joint prior probability mass function (pmf):

$$p(a_n, \theta_n, L_n) = p(a_n|\theta_n, L_n)\, p(\theta_n)\, p(L_n)$$

$$p(\theta_n) = \begin{cases} p_\mathrm{d} & \theta_n = 1 \\ 1 - p_\mathrm{d} & \theta_n = 0 \end{cases}$$

$$p(a_n|\theta_n = 1, L_n) = \begin{cases} \frac{1}{L_n+1} & a_n \in \{1, \dots, M_n\} \\ 0 & a_n = 0 \end{cases}$$

$$p(L_n) = \frac{\mu_\mathrm{c}^{L_n}}{L_n!} e^{-\mu_\mathrm{c}}$$

$$p(a_n|\theta_n = 0, L_n) = \begin{cases} 0 & a_n \in \{1, \dots, M_n\} \\ 1 & a_n = 0 \end{cases}$$

Prior Distribution

- After variable transform $L_n + \theta_n \rightarrow M_n$ we obtain

$$p(a_n, M_n) = \begin{cases} p_{\mathrm{d}} \frac{\mu_{\mathrm{c}}^{M_n - 1}}{M_n!} e^{-\mu_{\mathrm{c}}} & a_n \in \{1, \ldots, M_n\} \\ (1 - p_{\mathrm{d}}) \frac{\mu_{\mathrm{c}}^{M_n}}{M_n!} e^{-\mu_{\mathrm{c}}} & a_n = 0 \end{cases}$$

- Properties:
  - For all arbitrary $L_n + \theta_n = M_n$ we have $p(a_n, \theta_n, L_n) = p(a_n, M_n)$
  - $p(a_n, M_n)$ is a valid pmf in the sense that $\sum_{M_n=0}^{\infty} \sum_{a_n=0}^{M_n} p(a_n, M_n) = 1$

Likelihood Function

- Key Assumption II:
  - Clutter measurements are independent and identically distributed (iid) according to $f_{\mathrm{c}}(\boldsymbol{z}_{m,n})$
  - Condition on $\boldsymbol{x}_n$, the object-generated measurement $\boldsymbol{z}_{a_n,n}$ is conditionally independent of all the other measurements

- Likelihood function:
  - for $\boldsymbol{z}_n \in \mathbb{R}^{M_n d}$,

measurement model $\boldsymbol{z}_{a_n,n} = h_n(\boldsymbol{x}_n, \boldsymbol{v}_n)$ with noise $\boldsymbol{v}_n$

$$f(\boldsymbol{z}_n|\boldsymbol{x}_n, a_n, M_n) = \begin{cases} \prod_{m=1}^{M_n} f_{\mathrm{c}}(\boldsymbol{z}_{m,n}) & a_n = 0 \\ \frac{\boxed{f(\boldsymbol{z}_{a_n,n}|\boldsymbol{x}_n)}}{f_{\mathrm{c}}(\boldsymbol{z}_{a_n,n})} \prod_{m=1}^{M_n} f_{\mathrm{c}}(\boldsymbol{z}_{m,n}) & a_n \in \{1, \dots, M_n\} \end{cases}$$

  - For $\boldsymbol{z}_n \notin \mathbb{R}^{M_n d}$,

$$f(\boldsymbol{z}_n|\boldsymbol{x}_n, a_n, M_n) = 0$$

Joint Distributions

- Joint prior for $\boldsymbol{x}_{0:n}$

State transition model $\boldsymbol{x}_n = g_n(\boldsymbol{x}_{n-1}, \boldsymbol{u}_n)$ with noise $\boldsymbol{u}_n$

$$f(\boldsymbol{x}_{0:n}) = f(\boldsymbol{x}_0) \prod_{n'=1}^{n} \boxed{f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1})}$$

Driving noise independent across time n and independent of $\boldsymbol{x}_0$

- Joint prior for $\boldsymbol{a}_{1:n}$ and $\boldsymbol{M}_{1:n}$

$$p(\boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n}) = \prod_{n'=1}^{n} p(a_{n'}, M_{n'})$$

Measurement generation independent across time n

- Joint likelihood function

$$f(\boldsymbol{z}_{1:n} | \boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n}) = \prod_{n'=1}^{n} f(\boldsymbol{z}_{n'} | \boldsymbol{x}_{n'}, a_{n'}, M_{n'})$$

Measurement noise and clutter independent across time n

The Joint Posterior Distribution

- The joint posterior distribution ($M_{1:n}$ and $\boldsymbol{z}_{1:n}$ are observed and thus fixed)

$$f(\boldsymbol{x}_{0:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) = f(\boldsymbol{x}_{0:n}, \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n} | \boldsymbol{z}_{1:n}) \longleftarrow M_{1:n} \text{ fixed}$$

Bayes rule $\longrightarrow$

$$\propto f(\boldsymbol{z}_{1:n} | \boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n}) f(\boldsymbol{x}_{0:n}, \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n})$$

$\boldsymbol{x}_{0:n} \perp\!\!\!\perp \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n}$ $\longrightarrow$

$$= f(\boldsymbol{z}_{1:n} | \boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n}) f(\boldsymbol{x}_{0:n}) p(\boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n})$$

Expressions for joint distributions $\longrightarrow$

$$= f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) f(\boldsymbol{z}_{n'} | \boldsymbol{x}_{n'}, \boldsymbol{a}_{n'}, M_{n'}) p(\boldsymbol{a}_{n'}, M_{n'})$$

Problem Formulation

- Input at time n:
  - All observations up to time $\boldsymbol{z}_{1:n}$
  - ``Markovian'' statistical model

$$f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n}|\boldsymbol{z}_{1:n}) \propto f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'}|\boldsymbol{x}_{n'-1}) f(\boldsymbol{z}_{n'}|\boldsymbol{x}_{n'}, \boldsymbol{a}_{n'}, M_{n'}) p(\boldsymbol{a}_{n'}, M_{n'})$$

- Output at time n:
  - Estimate of $\hat{\boldsymbol{x}}_n$

- Calculation of an estimate $\hat{\boldsymbol{x}}_n$ is based on the marginal posterior pdf $f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n})$

The Factor Graph

- Recall factorization of the joint posterior distribution:

$$f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n} | \boxed{\boldsymbol{z}_{1:n}}) \propto f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) f(\boxed{\boldsymbol{z}_{n'}} | \boldsymbol{x}_{n'}, a_{n'}, \boxed{M_{n'}}) p(a_{n'}, \boxed{M_{n'}})$$

observed and fixed

The Factor Graph

- Recall factorization of the joint posterior distribution:

$$f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) \propto f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) f(\boldsymbol{z}_{n'} | \boldsymbol{x}_{n'}, a_{n'}, M_{n'}) p(a_{n'}, M_{n'})$$

$$\propto f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) g_1(\boldsymbol{x}_{n'}, a_{n'}) g_2(a_{n'})$$

$$f(\boldsymbol{z}_n | \boldsymbol{x}_n, a_n, M_n) = \begin{cases} \frac{f(\boldsymbol{z}_{a_n,n} | \boldsymbol{x}_n)}{f_{\mathrm{c}}(\boldsymbol{z}_{a_n,n})} \prod_{m=1}^{M} f_{\mathrm{c}}(\boldsymbol{z}_{m,n}) & a_n \in \{1, \dots, M_n\} \\ \underbrace{\boxed{\prod_{m=1}^{M} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})}}_{\text{constant}} & a_n = 0 \end{cases}$$

$$g_1(\boldsymbol{x}_n, a_n) = \begin{cases} \frac{f(\boldsymbol{z}_{a_n,n} | \boldsymbol{x}_n)}{f_{\mathrm{c}}(\boldsymbol{z}_{a_n,n})} & a_n \in \{1, \dots, M_n\} \\ 1 & a_n = 0 \end{cases}$$

The Factor Graph

- **Recall factorization of the joint posterior distribution:**

$$f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) \propto f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) f(\boldsymbol{z}_{n'} | \boldsymbol{x}_{n'}, a_{n'}, M_{n'}) p(a_{n'}, M_{n'})$$

$$\propto f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) g_1(\boldsymbol{x}_{n'}, a_{n'}) g_2(a_{n'})$$

$$p(a_n, M_n) = \begin{cases} p_{\mathrm{d}} \frac{\mu_{\mathrm{c}}^{M_n - 1}}{M_n!} e^{-\mu_{\mathrm{c}}} & a_n = \{1, \ldots, M_n\} \\ (1 - p_{\mathrm{d}}) \underbrace{\boxed{\frac{\mu_{\mathrm{c}}^{M_n}}{M_n!} e^{-\mu_{\mathrm{c}}}}}_{\text{constant}} & a_n = 0 \end{cases}$$

$$g_2(a_n) = \begin{cases} \frac{p_{\mathrm{d}}}{\mu_{\mathrm{c}}} & a_n \in \{1, \ldots, M_n\} \\ (1 - p_{\mathrm{d}}) & a_n = 0 \end{cases}$$

The Factor Graph

- **Recall factorization of the joint posterior distribution:**

$$
\begin{aligned}
f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) &\propto f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) f(\boldsymbol{z}_{n'} | \boldsymbol{x}_{n'}, a_{n'}, M_{n'}) p(a_{n'}, M_{n'}) \\
&\propto f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) g_1(\boldsymbol{x}_{n'}, a_{n'}) g_2(a_{n'}) \\
&= f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) g_{\boldsymbol{z}_n}(\boldsymbol{x}_{n'}, a_{n'})
\end{aligned}
$$

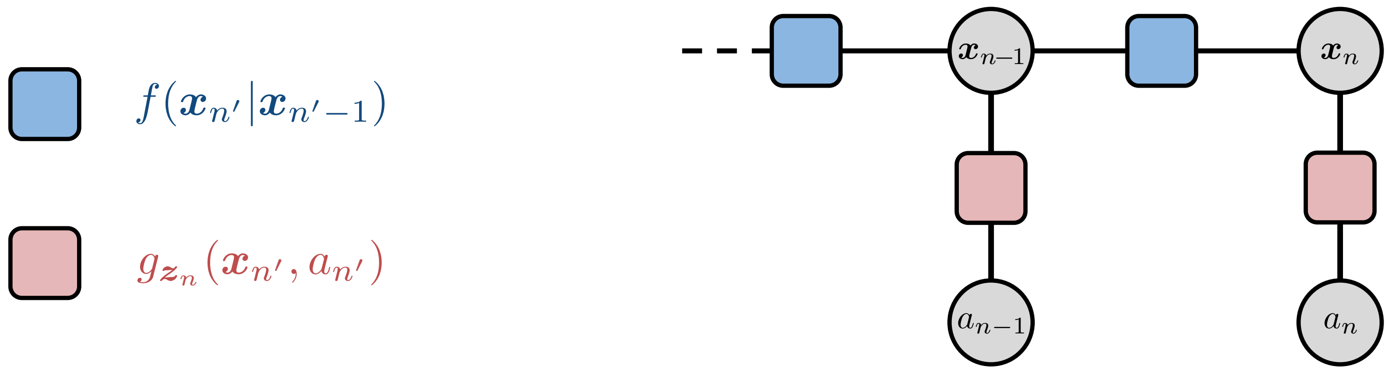
$$
g_{\boldsymbol{z}_n}(\boldsymbol{x}_n, a_n) = g_1(\boldsymbol{x}_n, a_n) g_2(a_n) = \begin{cases} \frac{p_{\mathrm{d}} f(\boldsymbol{z}_{a_n,n} | \boldsymbol{x}_n)}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{a_n,n})} & a_n \in \{1, \ldots, M_n\} \\ (1 - p_{\mathrm{d}}) & a_n = 0 \end{cases}
$$

The Factor Graph

- Recall factorization of the joint posterior distribution:

$$f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) \propto f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) \, g_{\boldsymbol{z}_n}(\boldsymbol{x}_{n'}, a_{n'})$$

- Factor graph for two time steps $n' \in \{n-1, n\}$



- The factor graph is cycle free $\Rightarrow$ message passing can provide exact marginals

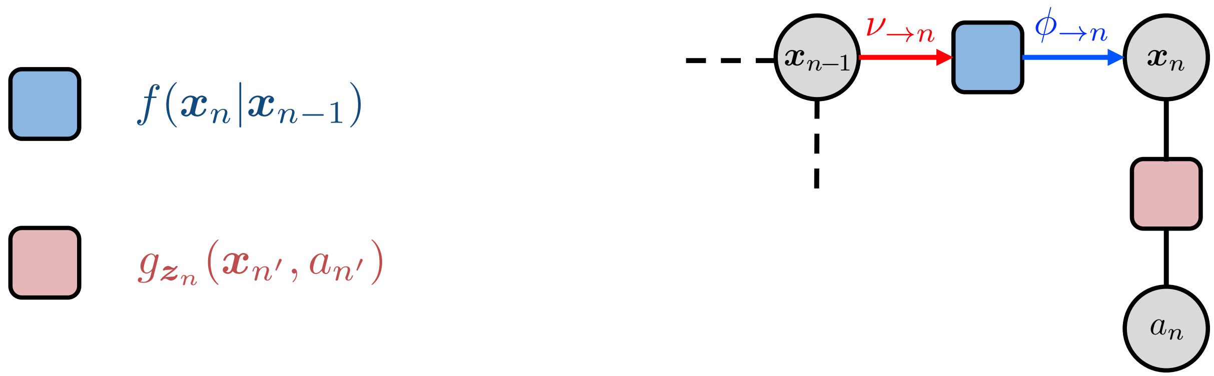
Prediction

- Prediction step:

$$f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1}) = \int f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) f(\boldsymbol{x}_{n-1}|\boldsymbol{y}_{1:n-1}) \mathrm{d}\boldsymbol{x}_{n-1}$$

$$\phi_{\rightarrow n}(\boldsymbol{x}_n) = \int f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})\, \nu_{\rightarrow n}(\boldsymbol{x}_{n-1})\, \mathrm{d}\boldsymbol{x}_{n-1}$$

- Factor graph for two time steps $n' \in \{n-1, n\}$



Data Association

- Data association step:

$$\phi_n(\boldsymbol{x}_n) = \sum_{m=0}^{M_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_n, a_n = m)\, \nu_n(a_n = m)$$

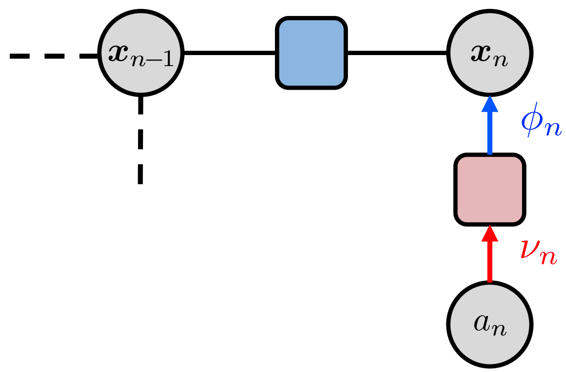
$$\nu_n(a_n) = 1$$

(no other neighbors)

- Factor graph for two time steps $n' \in \{n-1, n\}$

$f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{n'}, a_{n'})$



Update

- Update step:

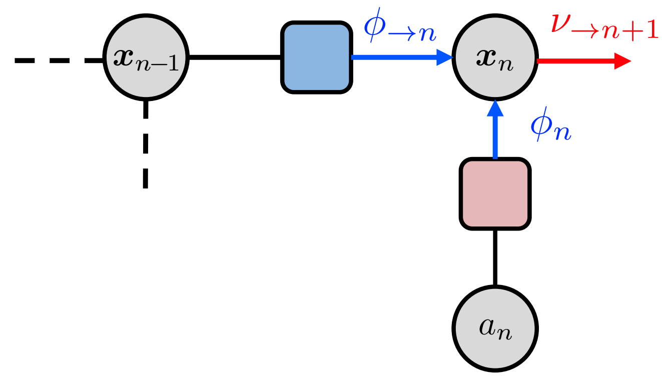
$$f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n}) \propto \phi_n(\boldsymbol{x}_n)\, \phi_{\rightarrow n}(\boldsymbol{x}_n) \qquad f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n}) \propto \phi_n(\boldsymbol{x}_n)\, f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1})$$

$$\nu_{\rightarrow n+1}(\boldsymbol{x}_n) = \phi_n(\boldsymbol{x}_n)\, \phi_{\rightarrow n}(\boldsymbol{x}_n)$$

- Factor graph for two time steps $n' \in \{n-1, n\}$

$f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{n'}, a_{n'})$



Particle-Based Update Step (cf. Class 4)

- **Given:** Particles $\{(\boldsymbol{x}_n^{(j)})\}_{j=1}^{J} \simeq f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1})$ representing the predicted posterior PDF

- **Wanted:** Particles $\{(\overline{\boldsymbol{x}}_n^{(j)})\}_{j=1}^{J} \simeq f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})$ representing the posterior PDF

- Perform importance sampling with proposal distribution $f_{\mathrm{p}}(\boldsymbol{x}_n) = f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1})$ and target distribution $f_{\mathrm{t}}(\boldsymbol{x}_n) \propto \phi_n(\boldsymbol{x}_n) f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1})$
  - calculate unnormalized weights $\tilde{w}_n^{(j)} = \boxed{\sum_{m=0}^{M_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_n^{(j)}, a_n = m)} \propto f_{\mathrm{t}}(\boldsymbol{x}_n^{(j)}) / f_{\mathrm{p}}(\boldsymbol{x}_n^{(j)})$
  - normalize weights $w_n^{(j)} = \tilde{w}_n^{(j)} / \sum_{j'=1}^{J} \tilde{w}_n^{(j')}, \quad j = 1, \ldots, J$

- Perform resampling to get $\{(\overline{\boldsymbol{x}}_n^{(j)})\}_{j=1}^{J} \simeq f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})$ from $\{(\boldsymbol{x}_n^{(j)}, w_n^{(j)})\}_{j=1}^{J} \simeq f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})$

Summary

- Single object tracking in clutter
  - possible association events are modelled by discrete random variable
  - data association is performed by summing over all possible association events
  - the sequential estimation problem that can be represented by a cycle free factor graph
  - a particle-based implementation can provide exact estimation results as the number of particles goes to infinity

Graph-Based Multiobject Tracking

Part 2: The Probabilistic Data Association Filter

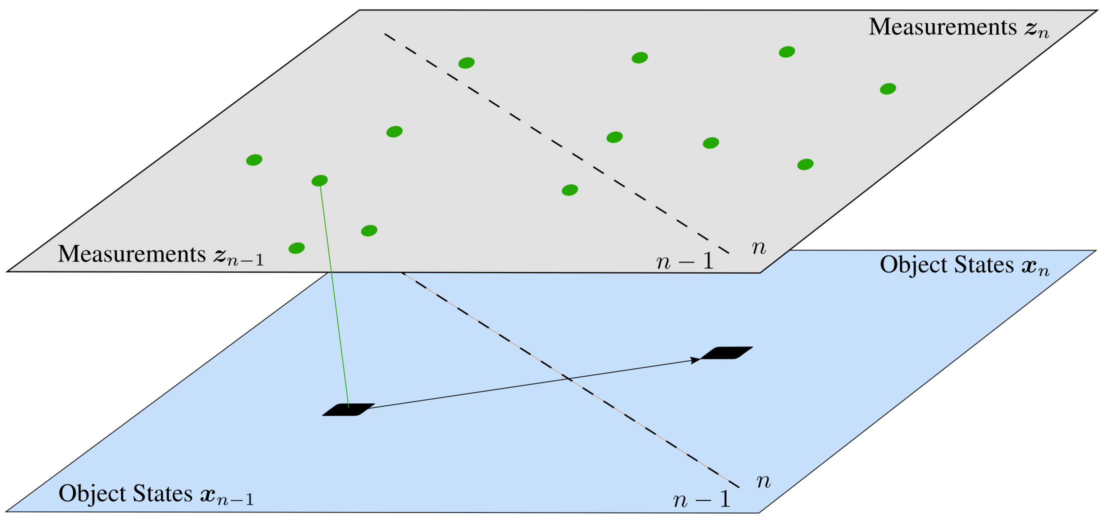
Florian Meyer
joint work with Jason Williams, Paolo Braca, Peter Willett, and Franz Hlawatsch

Scripps Institution of Oceanography
Electrical and Computer Engineering Department
University of California San Diego

UC San Diego
JACOBS SCHOOL OF ENGINEERING

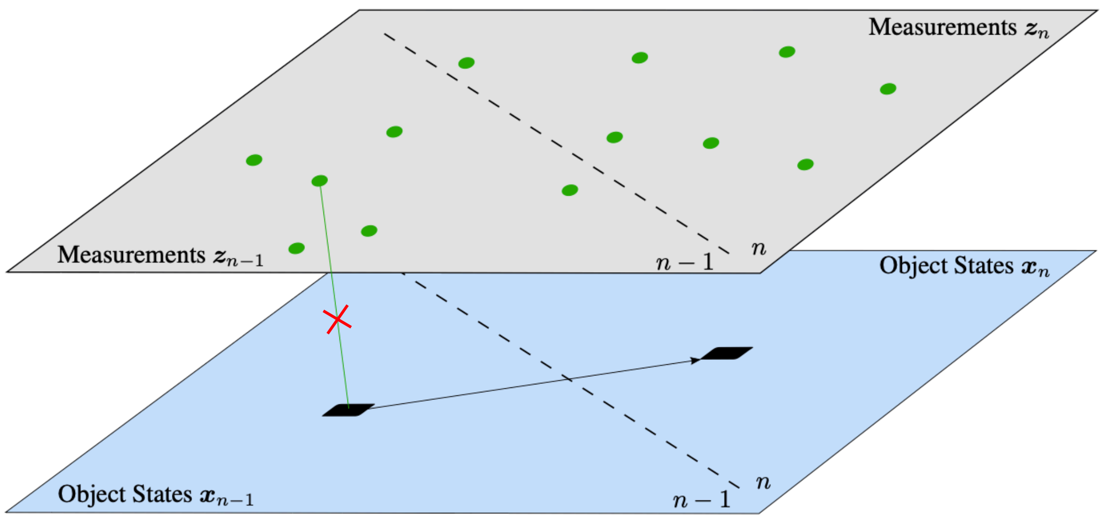
Basic Setting

- "Single object tracking in clutter" problem



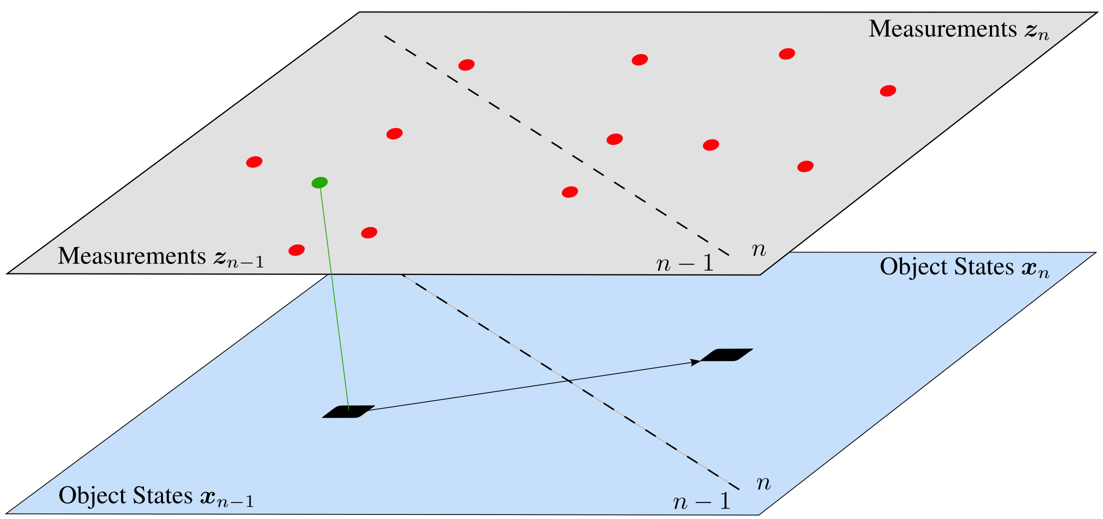
Basic Setting

- “Single object tracking in clutter” problem
- Measurement-origin uncertainty (MOU), false clutter measurements and missed detections



Basic Setting

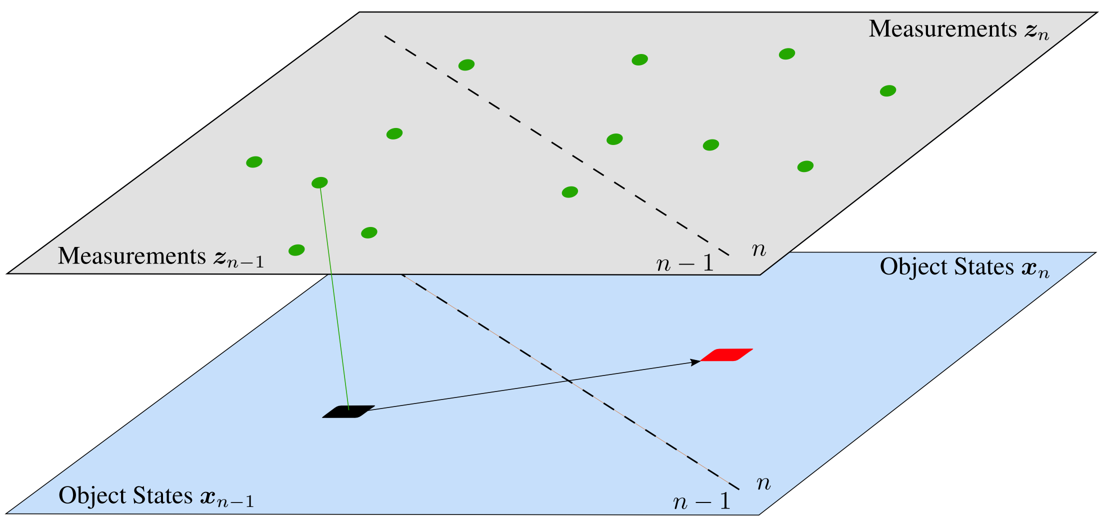
- "Single object tracking in clutter" problem
- Measurement-origin uncertainty (MOU), false clutter measurements and missed detections



Basic Setting

- "Single object tracking in clutter" problem

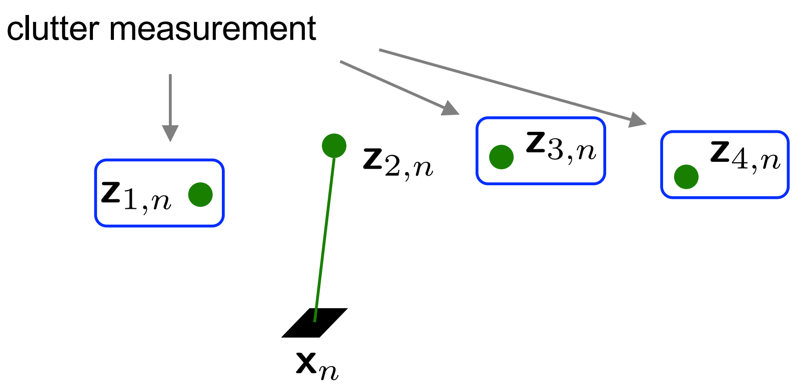
- Measurement-origin uncertainty (MOU), false clutter measurements and missed detections



The Association Variable

- Object-oriented association variable $a_n \in \{0, 1, \ldots, \boxed{M_n}\}$ number of measurements
  - $a_n = m > 0$: at time n, the object generates the measurement with index m
  - $a_n = 0$: at time n, the object did not generate a measurement

- Example:



$M_n = 4$ $a_n = 2$

Probabilistic Data Association Filter

- **Prediction Step**

$$\underbrace{f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1})}_{\text{Predicted posterior pdf}} = \int \underbrace{f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})}_{\text{State-transition pdf}} \underbrace{f(\boldsymbol{x}_{n-1}|\boldsymbol{y}_{1:n-1})}_{\text{Previous posterior pdf}} \mathrm{d}\boldsymbol{x}_{n-1}$$

- **Updated Step**

$$\underbrace{f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n})}_{\text{Posterior pdf}} \propto \underbrace{f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1})}_{\text{Predicted posterior pdf}} \sum_{m=0}^{M_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_n, a_n = m)$$

$$= f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1}) \left((1 - p_{\mathrm{d}}) + \frac{p_{\mathrm{d}} f(\boldsymbol{z}_{m,1}|\boldsymbol{x}_n)}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,1})} + \cdots + \frac{p_{\mathrm{d}} f(\boldsymbol{z}_{m,M_n}|\boldsymbol{x}_n)}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,M_n})} \right)$$

Key Parameters

- Posterior Distribution

$$f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n}) \propto f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1}) \left((1 - p_{\mathrm{d}}) + \frac{p_{\mathrm{d}} f(\boldsymbol{z}_{1,n}|\boldsymbol{x}_n)}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{1,n})} + \cdots + \frac{p_{\mathrm{d}} f(\boldsymbol{z}_{M_n,n}|\boldsymbol{x}_n)}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{M_n,n})} \right)$$

- Probability that a measurements is generated by the object $0 < p_{\mathrm{d}} \leqslant 1$ (probability of detection)

Key Parameters

- Posterior Distribution

$$f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n}) \propto f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1}) \left((1-p_\mathrm{d}) + \frac{p_\mathrm{d} f(\boldsymbol{z}_{1,n}|\boldsymbol{x}_n)}{\mu_\mathrm{c} f_\mathrm{c}(\boldsymbol{z}_{1,n})} + \cdots + \frac{p_\mathrm{d} f(\boldsymbol{z}_{M_n,n}|\boldsymbol{x}_n)}{\mu_\mathrm{c} f_\mathrm{c}(\boldsymbol{z}_{M_n,n})} \right)$$

- Probability that a measurements is generated by the object $0 < p_\mathrm{d} \leqslant 1$ (probability of detection)

- Mean number of clutter measurements $0 < \mu_\mathrm{c}$

Key Parameters

- Posterior Distribution

$$f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n}) \propto f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1}) \left((1 - p_{\mathrm{d}}) + \frac{p_{\mathrm{d}} f(\boldsymbol{z}_{1,n}|\boldsymbol{x}_n)}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{1,n})} + \cdots + \frac{p_{\mathrm{d}} f(\boldsymbol{z}_{M_n,n}|\boldsymbol{x}_n)}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{M_n,n})} \right)$$

- Probability that a measurements is generated by the object $0 < p_{\mathrm{d}} \leqslant 1$ (probability of detection)

- Mean number of clutter measurements $0 < \mu_{\mathrm{c}}$

- Clutter pdf $0 < f_{\mathrm{c}}(\boldsymbol{z}_m)$

Linear-Gaussian State-Space Model

- Consider a sequence of states $\boldsymbol{x}_n$ and a sequence of measurements $\boldsymbol{y}_n$

State-Transition Model:

State $\boldsymbol{x}_n$ evolves according to

$$\boldsymbol{x}_n = \boldsymbol{G}_n \boldsymbol{x}_{n-1} + \underbrace{\boldsymbol{u}_n}_{\text{driving noise (white)}}$$

$\Rightarrow f(\boldsymbol{x}_n | \boldsymbol{x}_{n-1})$

with Gaussian driving noise

$$\boldsymbol{u}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{u}_n})$$

Model for Object Generated Meas.:

Measurement $\boldsymbol{y}_n$ is generated as

$$\boldsymbol{y}_{n,m} = \boldsymbol{H}_n \boldsymbol{x}_n + \underbrace{\boldsymbol{v}_n}_{\text{measurement noise (white)}}$$

$\Rightarrow f(\boldsymbol{y}_{n,m} | \boldsymbol{x}_n)$

with Gaussian measurement noise

$$\boldsymbol{v}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{v}_n})$$

- Prior PDF at $n = 0$, $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}_0}, \boldsymbol{\Sigma}_{\boldsymbol{x}_0})$

Prob. Data Association with Linear-Gaussian Model

- Let us assume $f(\boldsymbol{x}_{n-1}|\boldsymbol{y}_{1:n-1})$ is Gaussian with mean $\boldsymbol{\mu}_{\boldsymbol{x}_{n-1}}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_{n-1}}$

- The Prediction step can be performed in closed form (as in the Kalman filter), i.e., $f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1})$ is Gaussian with mean $\boldsymbol{\mu}^-_{\boldsymbol{x}_n}$ and covariance $\boldsymbol{\Sigma}^-_{\boldsymbol{x}_n}$ given as

$$\boldsymbol{\mu}^-_{\boldsymbol{x}_n} = G_n \boldsymbol{\mu}_{\boldsymbol{x}_{n-1}} \qquad\qquad \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n} = \boldsymbol{G}_n \boldsymbol{\Sigma}_{\boldsymbol{x}_{n-1}} \boldsymbol{G}_n^{\mathrm{T}} + \boldsymbol{\Sigma}_{\boldsymbol{u}_n}$$

- **Goal:** Closed-form solution for the update step such that $f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})$ can be represented be mean $\boldsymbol{\mu}_{\boldsymbol{x}_n}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_n}$

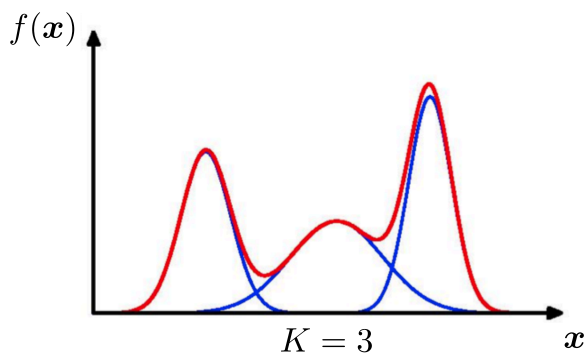
The Gaussian Mixture Distribution

- A continuous random variable $\boldsymbol{x}$ is said to have a Gaussian mixture distribution with K components and parameters w_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k, k = 1, \ldots, K$ if its probability density function is given by

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \, f_{\mathrm{g}}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- The $f_{\mathrm{g}}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ are Gaussian distributions with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$

- The weights $w_k > 0$ normalize to one, i.e., $\sum_{k=1}^{K} w_k = 1$



Mean and Covariance of Gaussian Mixture Distribution

- Let $f(\boldsymbol{x})$ be a Gaussian mixture distributions with parameters w_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, $k = 1, \dots, K$

- The mean of $f(\boldsymbol{x})$ is given by

$$\boldsymbol{\mu}_{\boldsymbol{x}} = w_1 \boldsymbol{\mu}_1 + w_2 \boldsymbol{\mu}_2 + \cdots + w_K \boldsymbol{\mu}_K$$

- The covariance of $f(\boldsymbol{x})$ is given by

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \sum_{k=1}^{K} w_k \boldsymbol{\Sigma}_k + \sum_{k=1}^{K} w_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathrm{T}} - \boldsymbol{\mu}_{\boldsymbol{x}} \boldsymbol{\mu}_{\boldsymbol{x}}^{\mathrm{T}}$$

Closed-Form Update Step

- Recall posterior distribution ($\boldsymbol{z}_{1:n}$ is observed and thus fixed)

$$f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n}) \propto f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1}) \bigg((1-p_\mathrm{d}) + \frac{p_\mathrm{d} f(\boldsymbol{z}_{1,n}|\boldsymbol{x}_n)}{\mu_\mathrm{c} f_\mathrm{c}(\boldsymbol{z}_{1,n})} + \cdots + \frac{p_\mathrm{d} f(\boldsymbol{z}_{M_n,n}|\boldsymbol{x}_n)}{\mu_\mathrm{c} f_\mathrm{c}(\boldsymbol{z}_{M_n,n})} \bigg)$$

- **Theorem:** If the predicted posterior $f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1})$ is Gaussian, with mean $\boldsymbol{\mu}^-_{\boldsymbol{x}_n}$ and covariance $\boldsymbol{\Sigma}^-_{\boldsymbol{x}_n}$ and the model for the object-generated measurement is linear-Gaussian, then $f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n})$ is a Gaussian mixture distribution with $M_n + 1$ components and parameters

$$w_m \propto \frac{p_\mathrm{d} f(\boldsymbol{z}_{m,n}|\boldsymbol{z}_{1:n})}{\mu_\mathrm{c} f_\mathrm{c}(\boldsymbol{z}_{m,n})} \qquad m \in \{1, \ldots, M_n\} \qquad w_{M_n+1} \propto (1-p_\mathrm{d})$$

$$\boldsymbol{\mu}_m = \boldsymbol{\mu}^-_{\boldsymbol{x}_n} + \boxed{\boldsymbol{K}_n}(\boldsymbol{z}_{m,n} - \boldsymbol{H}_n \boldsymbol{\mu}^-_{\boldsymbol{x}_n}) \qquad \boldsymbol{\mu}_{M_n+1} = \boldsymbol{\mu}^-_{\boldsymbol{x}_n}$$

Kalman gain

$$\boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n} - \boxed{\boldsymbol{K}_n}\boldsymbol{H}_n \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n} \qquad \boldsymbol{\Sigma}_{M_n+1} = \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n}$$

Closed-Form Update Step - Sketch of Proof

- Let's take a look at the single component $m \in \{1, \ldots, M_n\}$

$$\frac{p_{\mathrm{d}} f(\boldsymbol{z}_{m,n}|\boldsymbol{x}_n) f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1})}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})} = \frac{p_{\mathrm{d}} f(\boldsymbol{z}_{m,n}|\boldsymbol{x}_n, \boldsymbol{z}_{1:n-1}) f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1})}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})}$$

Statistical Independent Meas. & Driving Noise

$$= \frac{p_{\mathrm{d}} f(\boldsymbol{z}_{m,n}, \boldsymbol{x}_n|\boldsymbol{z}_{1:n-1})}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})}$$

$$= \frac{p_{\mathrm{d}} f(\boldsymbol{z}_{m,n}|\boldsymbol{z}_{1:n-1})}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})} f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1}, \boldsymbol{z}_{m,n})$$

Kalman Update Step

$$= \frac{p_{\mathrm{d}} f(\boldsymbol{z}_{m,n}|\boldsymbol{z}_{1:n-1})}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})} f_{\mathrm{g}}(\boldsymbol{x}_n; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

$$\boldsymbol{\mu}_m = \boldsymbol{\mu}^-_{\boldsymbol{x}_n} + \boldsymbol{K}_n(\boldsymbol{z}_{m,n} - \boldsymbol{H}_n \boldsymbol{\mu}^-_{\boldsymbol{x}_n})$$

$$\boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n} - \boldsymbol{K}_n \boldsymbol{H}_n \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n}$$

$$\boldsymbol{K}_n = \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n} \boldsymbol{H}_n^{\mathrm{T}} (\boldsymbol{H}_n \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n} \boldsymbol{H}_n^{\mathrm{T}} + \boldsymbol{\Sigma}_{\boldsymbol{v}_n})^{-1}$$

- The conditional evidence $f(z_{m,n}|z_{1:n-1})$ is given by

$$f(\boldsymbol{z}_{m,n}|\boldsymbol{z}_{1:n-1}) = f_{\mathrm{g}}(\boldsymbol{z}_{m,n}; \boldsymbol{H}_n \boldsymbol{\mu}^-_{\boldsymbol{x}_n}, \boldsymbol{H}_n \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n} \boldsymbol{H}_n^{\mathrm{T}} + \boldsymbol{\Sigma}_{\boldsymbol{v}_n})$$

Closed-Form Update Step - Discussion

- Mean and covariances $\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m, m = 1, \ldots, M_n + 1$ of the Gaussian mixture distribution are obtained by
  - performing the Kalman update step for each ``measurement component'' $m = 1, \ldots, M_n$
  - keeping predicted mean and covariance for the ``missed-detection component'' $m = M_n + 1$

- Weights $\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m, m = 1, \ldots, M_n + 1$ are given as

$$w_m \propto p_{\mathrm{d}} f(\boldsymbol{z}_{m,n}|\boldsymbol{z}_{1:n}) / \mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n}), \qquad m = 1, \ldots, M_n \qquad\qquad w_{M_n+1} \propto (1 - p_{\mathrm{d}})$$

  - The probability of detection p_{d} determines the ratio between measurement component weights and missed-detection component weight
  - Large conditional evidence $f(\boldsymbol{z}_{m,n}|\boldsymbol{z}_{1:n})$ means that measurement $\boldsymbol{z}_{m,n}$ is likely to be object generated
  - Large $\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})$ means that the measurement is likely to be clutter

Example

Posterior at time $n-1$

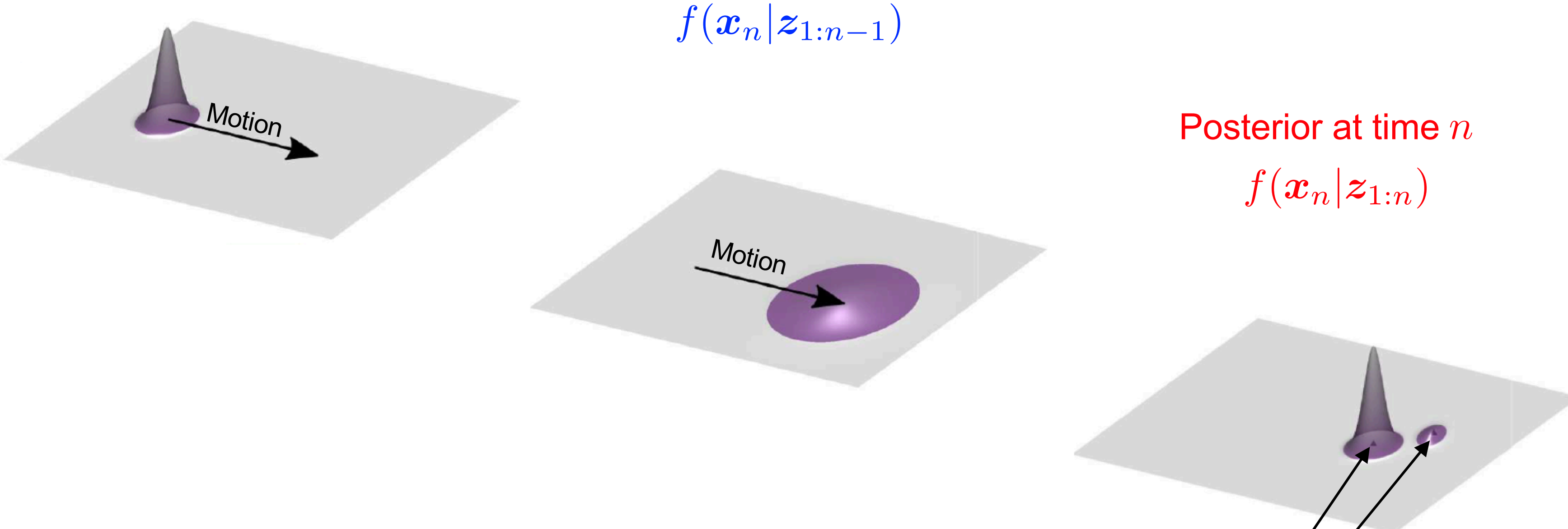
$f(\boldsymbol{x}_{n-1}|\boldsymbol{z}_{1:n-1})$

Predicted Posterior at time n

$f(\boldsymbol{x}_{n}|\boldsymbol{z}_{1:n-1})$

Posterior at time n

$f(\boldsymbol{x}_{n}|\boldsymbol{z}_{1:n})$



Measurements
$(M_n = 2)$

D. Gaglione, G. Soldi, F. Meyer, F. Hlawatsch, P. Braca, A. Farina, and M. Z. Win, *Bayesian information fusion and multitarget tracking for maritime situational awareness*, IET Radar Sonar Navi., Nov. 2020.

Closed-Form Update Step - Approximation

- Let's assume $f(\boldsymbol{x}_{n-1}|\boldsymbol{z}_{1:n-1})$ is a Gaussian mixture distribution with K components

- At time n, we could calculate a predicted posterior $f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1})$ that has a Gaussian mixture distribution from $f(\boldsymbol{x}_{n-1}|\boldsymbol{z}_{1:n-1})$ by performing K prediction steps

- However, after the following update step, we would obtain a Gaussian mixture with $K(M_n + 1)$ components $\Rightarrow$ complexity of the resulting algorithm has a computational complexity that scales exponentially with time n

- Thus, after each update step, we approximate the update posterior $f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n})$ by a single Gaussian with a mean $\boldsymbol{\mu}_{\boldsymbol{x}_n}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_n}$ that are equal to the mean and covariance of its Gaussian mixture distribution (moment matching)

Closed-Form Update Step - Summary

- Step 1: Calculate means and covariances of mixture components:

$$\boldsymbol{\mu}_m = \boldsymbol{\mu}^-_{\boldsymbol{x}_n} + \boldsymbol{K}_n\big(\boldsymbol{z}_{m,n} - \boldsymbol{H}_n\boldsymbol{\mu}^-_{\boldsymbol{x}_n}\big) \qquad m = 1,\dots,M_n$$

$$\boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n} - \boldsymbol{K}_n\boldsymbol{H}_n\boldsymbol{\Sigma}^-_{\boldsymbol{x}_n}$$

$$\boldsymbol{K}_n = \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n}\boldsymbol{H}_n^{\mathrm{T}}\big(\boldsymbol{H}_n\boldsymbol{\Sigma}^-_{\boldsymbol{x}_n}\boldsymbol{H}_n^{\mathrm{T}} + \boldsymbol{\Sigma}_{\boldsymbol{v}_n}\big)^{-1}$$

$$\boldsymbol{\mu}_{M_n+1} = \boldsymbol{\mu}^-_{\boldsymbol{x}_n}$$

$$\boldsymbol{\Sigma}_{M_n+1} = \boldsymbol{\Sigma}^-_{\boldsymbol{x}_n}$$

- Step 2: Calculate unnormalized weights:

$$\tilde{w}_m = \frac{p_{\mathrm{d}} f_{\mathrm{g}}(\boldsymbol{z}_{m,n}; \boldsymbol{H}_n\boldsymbol{\mu}^-_{\boldsymbol{x}_n}, \boldsymbol{H}_n\boldsymbol{\Sigma}^-_{\boldsymbol{x}_n}\boldsymbol{H}_n^{\mathrm{T}} + \boldsymbol{\Sigma}_{\boldsymbol{v}_n})}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})} \qquad m = 1,\dots,M_n \qquad \tilde{w}_{M_n+1} = (1 - p_{\mathrm{d}})$$

- Step 3: Normalize weights: $w_m = \tilde{w}_m / \big(\sum_{m'=1}^{M_n+1} \tilde{w}_{m'}\big)$
- Step 4: Approximate Gaussian mixture by a single Gaussian with same mean and covariance (moment matching):

$$\boldsymbol{\mu}_{\boldsymbol{x}_n} = \sum_{m=1}^{M_n+1} w_m \boldsymbol{\mu}_m \qquad \boldsymbol{\Sigma}_{\boldsymbol{x}_n} = \sum_{m=1}^{M_n+1} w_m \boldsymbol{\Sigma}_m + \sum_{m=1}^{M_n+1} w_m \boldsymbol{\mu}_m \boldsymbol{\mu}_m^{\mathrm{T}} - \boldsymbol{\mu}_{\boldsymbol{x}_n}\boldsymbol{\mu}_{\boldsymbol{x}_n}^{\mathrm{T}}$$

- **Result:** Mean $\boldsymbol{\mu}_{\boldsymbol{x}_n}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_n}$ representing the posterior distribution $f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n})$

Y. Bar-Shalom, F. Daum, and J. Huang, *The Probabilistic Data Association Filter*, IEEE Contr. Syst. Mag., 2009

Conclusion

- Single object tracking in clutter for linear-Gaussian system models
  - prediction and update steps can be performed in closed-form
  - posterior distributions are Gaussian mixture densities with a number of components that scales exponentially with time
  - to limit computational complexity, the posterior distribution is approximate by a single Gaussian after each update step

Graph-Based Multiobject Tracking

Part 3: Graph-Based Processing I

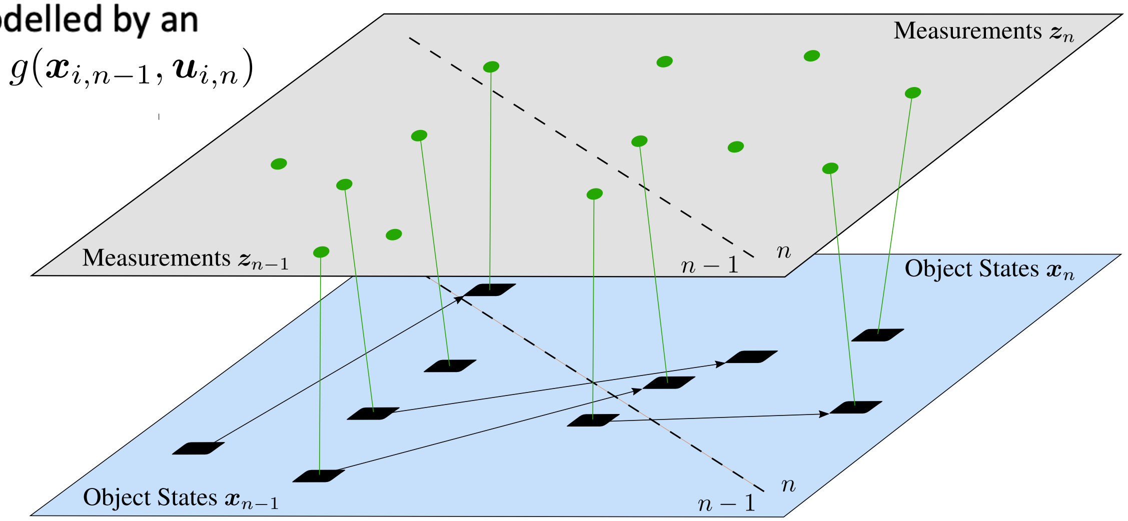
Florian Meyer
joint work with Jason Williams, Paolo Braca, Peter Willett, and Franz Hlawatsch

Scripps Institution of Oceanography
Electrical and Computer Engineering Department
University of California San Diego

The Multiobject Tracking Problem

- At each time n: localize and track multiple objects $\boldsymbol{x}_n = [\boldsymbol{x}_{1,n}^{\mathrm{T}} \ldots \boldsymbol{x}_{I,n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{1,n}^{\mathrm{T}} \ldots \boldsymbol{z}_{M_n,n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin

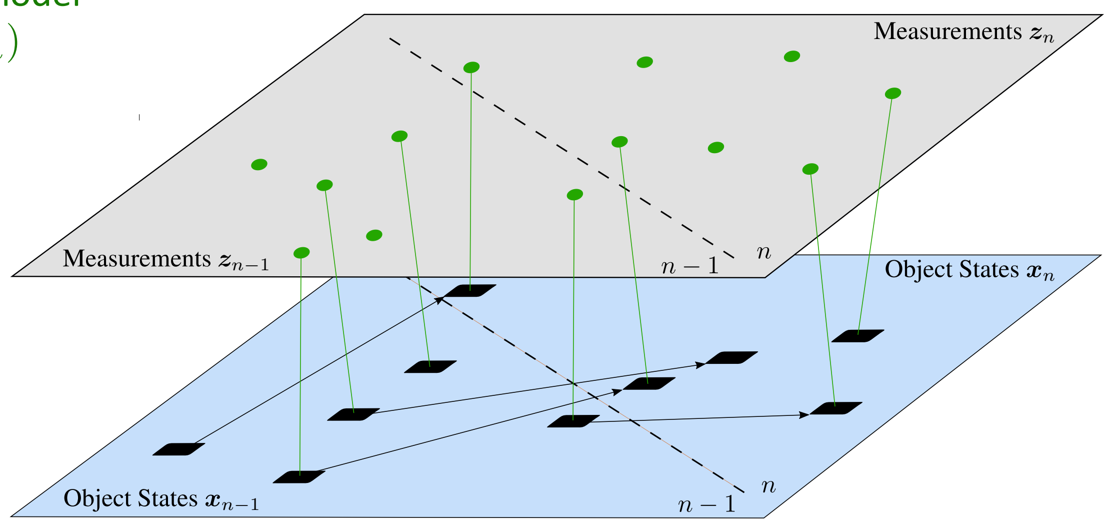
A state $\boldsymbol{x}_{i,n}$ consists of the object's position and further parameters; its evolution is time modelled by an arbitrary model $\boldsymbol{x}_{i,n} = g(\boldsymbol{x}_{i,n-1}, \boldsymbol{u}_{i,n})$ with noise $\boldsymbol{u}_{i,n}$



The Multiobject Tracking Problem

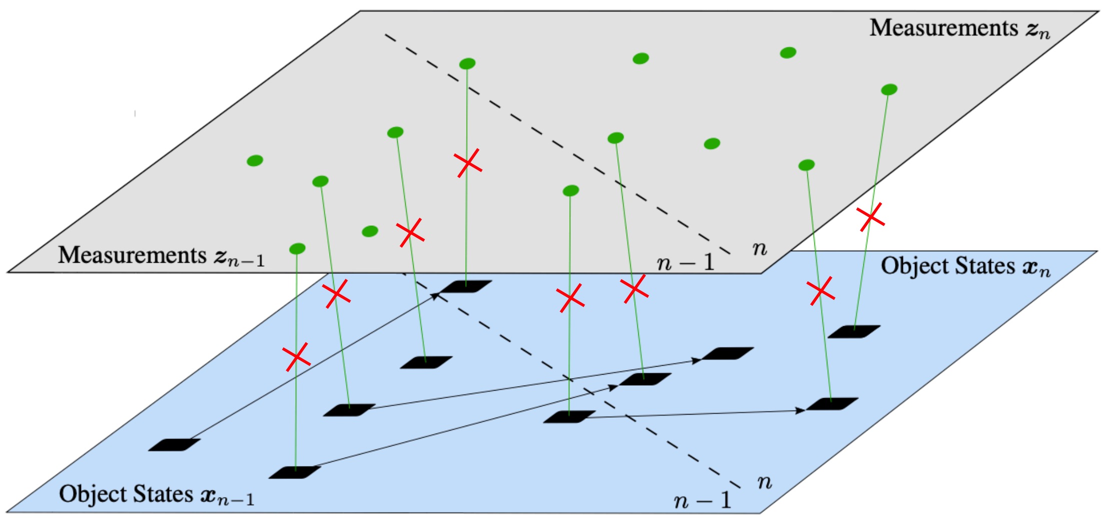
- At each time n: localize and track multiple objects $\boldsymbol{x}_n = [\boldsymbol{x}_{1,n}^{\mathrm{T}} \ldots \boldsymbol{x}_{I,n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{1,n}^{\mathrm{T}} \ldots \boldsymbol{z}_{M_n,n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin

A measurement $\boldsymbol{z}_{m,n}$ is modelled by an arbitrary nonlinear model
$\boldsymbol{z}_{m,n} = h(\boldsymbol{x}_{k,n}, \boldsymbol{v}_{m,n})$
with noise $\boldsymbol{v}_{m,n}$



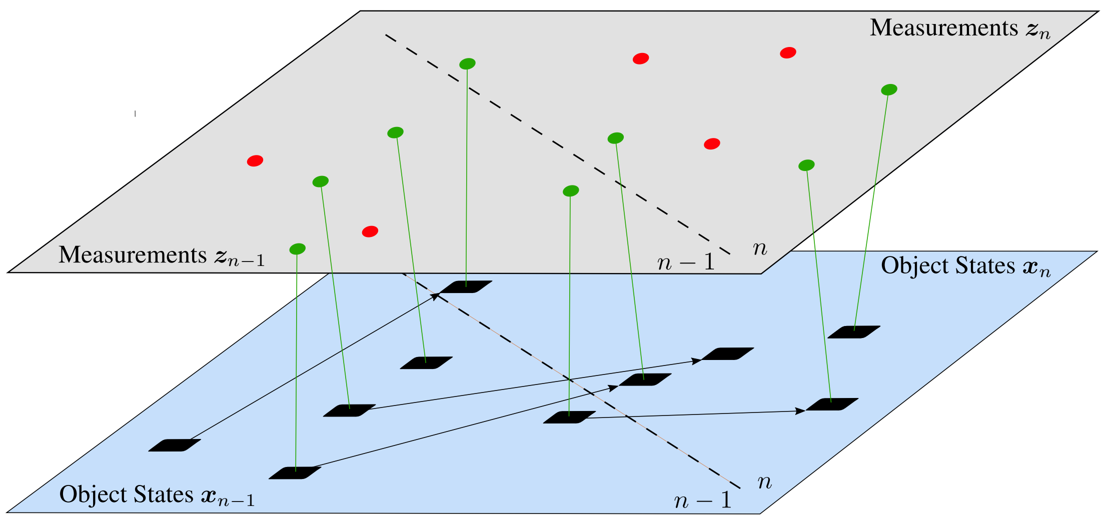
The Multiobject Tracking Problem

- At each time n: localize and track multiple objects $\boldsymbol{x}_n = [\boldsymbol{x}_{1,n}^{\mathrm{T}} \ldots \boldsymbol{x}_{I,n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{1,n}^{\mathrm{T}} \ldots \boldsymbol{z}_{M_n,n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin



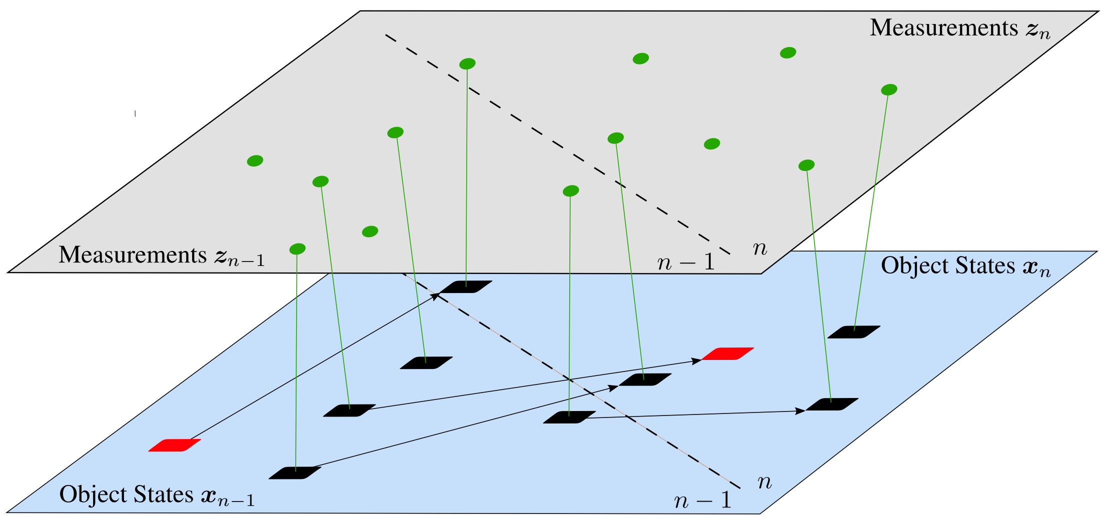
The Multiobject Tracking Problem

- At each time n: localize and track multiple objects $\boldsymbol{x}_n = [\boldsymbol{x}_{1,n}^{\mathrm{T}} \, \ldots \, \boldsymbol{x}_{I,n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{1,n}^{\mathrm{T}} \, \ldots \, \boldsymbol{z}_{M_n,n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin
- Data association is challenging because of false clutter measurements and missing measurements



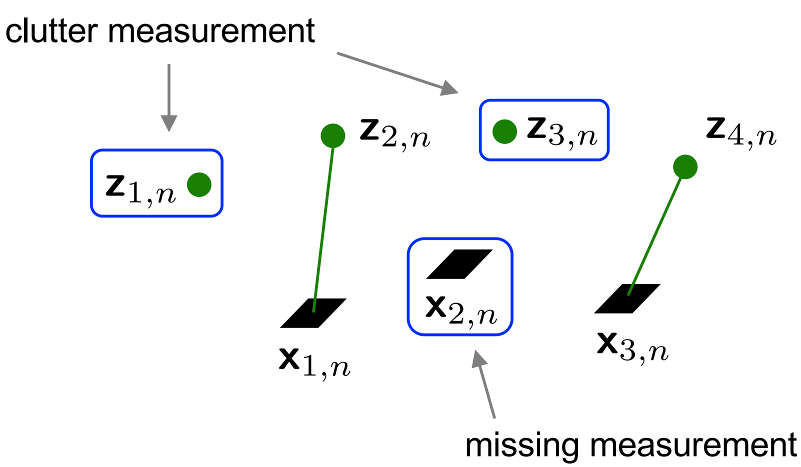
The Multiobject Tracking Problem

- At each time n: localize and track multiple objects $\boldsymbol{x}_n = [\boldsymbol{x}_{1,n}^{\mathrm{T}} \dots \boldsymbol{x}_{I,n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{1,n}^{\mathrm{T}} \dots \boldsymbol{z}_{M_n,n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin
- Data association is challenging because of false clutter measurements and missing measurements



Association Vectors

- Recall measurement vector at time n, $\boldsymbol{z}_n = [\boldsymbol{z}_{1,n}^{\mathrm{T}} \; \boldsymbol{z}_{2,n}^{\mathrm{T}} \; \dots \; \boldsymbol{z}_{M_n,n}^{\mathrm{T}}]^{\mathrm{T}}$
- Object-oriented association vector $\boldsymbol{a}_n = [a_{1,n} \; a_{2,n} \; \dots \; a_{I,n}]^{\mathrm{T}}$
  - $a_{i,n} = m > 0$: at time n object i generates measurement with index m
  - $a_{i,n} = 0$: at time n object i did not generate a measurement



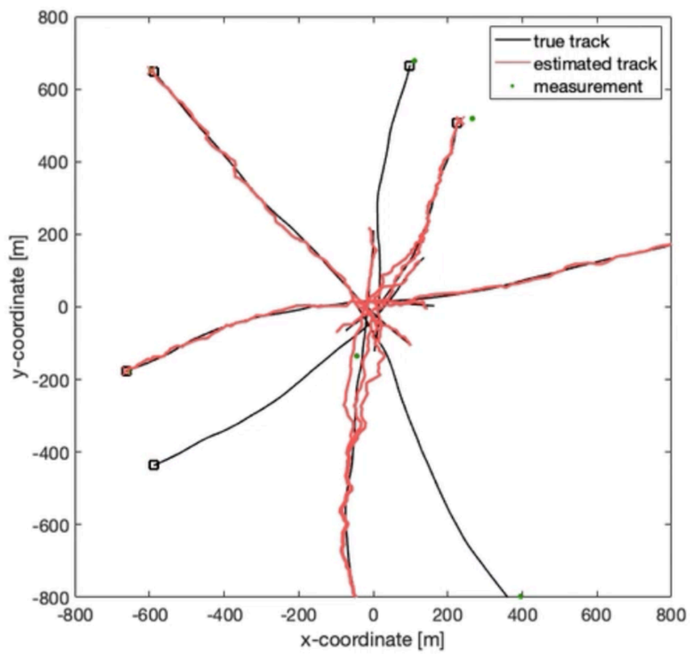
$M_n = 4$ $\quad$ $a_{1,n} = 2$

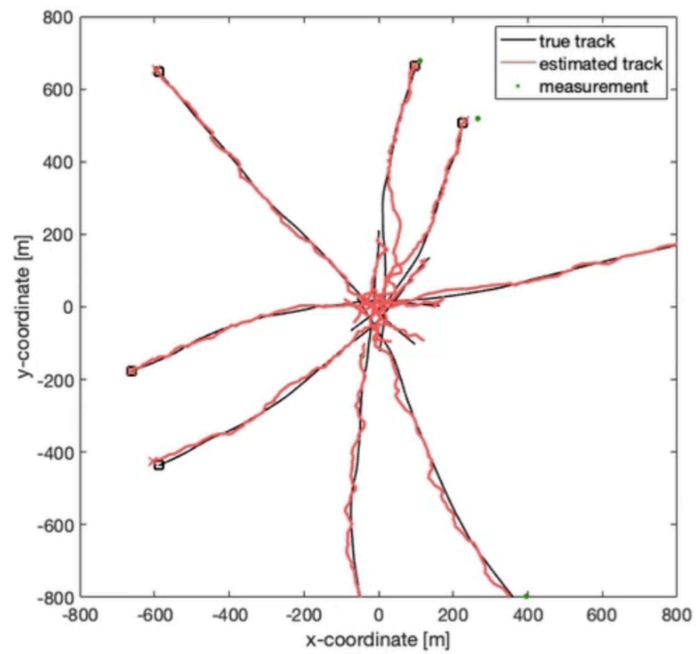
$I = 3$ $\quad$ $a_{2,n} = 0$

$a_{3,n} = 4$

Why Multiobject Tracking?

- Separate single-object tracking (left) vs joint multiobject tracking (right)





- Only a joint multiobject tracking formulation works

Prior Distributions

- Assumptions:
  1. Object detections are independent Bernoulli trials with success probability $0 < p_{\mathrm{d}} \leqslant 1$
  2. The number of clutter measurements is Poisson distributed with mean μ_{c}
  3. At most one measurement is generated by each object
  4. A measurement can be generated from at most one object
- Assumptions 1-3 are parallel to the single object tracking case
- Every association event expressed by a vector $\boldsymbol{a}_n = [a_{1,n} \dots a_{I,n}]^{\mathrm{T}}$ automatically fulfills Assumption 3 (scalar association variable $a_{i,n}$ for each object)
- Assumption 4 can be enforced by the following check function

$$\varphi(\boldsymbol{a}_n) \triangleq \begin{cases} 0, & \exists i, j \in \{1, 2, \dots, I\} \text{ such that } i \neq j \text{ and } a_{i,n} = a_{j,n} \neq 0 \\ 1, & \text{otherwise} \end{cases}$$

Prior Distributions

- Let us denote by $\mathcal{D}_{\boldsymbol{a}_n} = \big\{ i \in \{1, \ldots, I\} \,|\, a_{i,n} > 0 \big\}$ the set of detected object indexes corresponding to vector $\boldsymbol{a}_n$

- The prior pmf $p(\boldsymbol{a}_n, M_n)$ is given by

Check if every measurement is generated by at most one object

$$p(\boldsymbol{a}_n, M_n) = \boxed{\varphi(\boldsymbol{a}_n)} \left(\frac{p_{\mathrm{d}}}{\mu_{\mathrm{c}}(1 - p_{\mathrm{d}})} \right)^{|\mathcal{D}_{\boldsymbol{a}_n}|} \frac{e^{-\mu_{\mathrm{c}}} \mu_{\mathrm{c}}^{M_n}}{M_n!} \big(1 - p_{\mathrm{d}}\big)^I$$

- $p(\boldsymbol{a}_n, M_n)$ is a valid pmf in the sense that it can be normalized as

$$\sum_{M_n=0}^{\infty} \sum_{a_{1,n}=0}^{M_n} \cdots \sum_{a_{I,n}=0}^{M_n} p(\boldsymbol{a}_n, M_n) = 1$$

Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*, YBS, 2011.

Prior Distributions - Examples

- Example 1: No detections, all clutter case

Probability that no object generates a measurement

$$p(\boldsymbol{a}_n, M_n) = \boxed{\frac{e^{-\mu_c}\mu_c^{M_n}}{M_n!}}\boxed{\big(1-p_d\big)^I}$$

Poisson pmf of the number of clutter measurements evaluated at M_n

Prior Distributions - Example

- Example 2: All detections, no clutter case ($\boldsymbol{a}^{\mathrm{d}}$ is any association vector that assigns exactly one measurement to each object, i.e., any permutation of $1, 2, \ldots, I$)

Poisson pmf of the number of clutter measurements evaluated at *0*

$$p(\boldsymbol{a}_n = \boldsymbol{a}_n^{\mathrm{d}}, M_n = I) = \frac{\boxed{e^{-\mu_{\mathrm{c}}}}}{\boxed{I!}} \boxed{p_{\mathrm{d}}^{I}}$$

Probability that every object generates a measurement

There are $I!$ different $\boldsymbol{a}^{\mathrm{d}}$

Prior Distributions

- Joint prior distribution of object states at time $n = 0$

$$f(\boldsymbol{x}_0) = \prod_{i=1}^{I} f(\boldsymbol{x}_{i,0})$$

- Joint state transition function (object states evolve independently)

Driving noise independent across objects

$$f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) = \prod_{i=1}^{I} f(\boldsymbol{x}_{i,n}|\boldsymbol{x}_{i,n-1})$$

Driving noise independent across time n and independent of $\boldsymbol{x}_0$

- Joint prior distribution

$$f(\boldsymbol{x}_{0:n}) = f(\boldsymbol{x}_0) \prod_{n'=1}^{n} f(\boldsymbol{x}_{n'}|\boldsymbol{x}_{n'-1})$$

$$= \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$$

Likelihood Function

- Key Assumption II:
  - Clutter measurements are independent and identically distributed (iid) according to $f_{\mathrm{c}}(\boldsymbol{z}_{m,n})$
  - Condition on $\boldsymbol{x}_{i,n}$, the object-generated measurement $\boldsymbol{z}_{a_{i,n},n}$ is conditionally independent of all the other measurements

- Likelihood function:
  - for $\boldsymbol{z}_n \in \mathbb{R}^{M_n d}$,

measurement model $\boldsymbol{z}_{m,n} = h_n(\boldsymbol{x}_n, \boldsymbol{v}_n)$ with noise $\boldsymbol{v}_n$

$$f(\boldsymbol{z}_n | \boldsymbol{x}_n, a_n, M_n) = \Bigg(\prod_{i \in \mathcal{D}_{\boldsymbol{a}_n}} \frac{\boxed{f(\boldsymbol{z}_{a_{i,n},n} | \boldsymbol{x}_{i,n})}}{f_{\mathrm{c}}(\boldsymbol{z}_{a_{i,n},n})} \Bigg) \prod_{m=1}^{M_n} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})$$

  - For $\boldsymbol{z}_n \notin \mathbb{R}^{M_n d}$,

$$f(\boldsymbol{z}_n | \boldsymbol{x}_n, a_n, M_n) = 0$$

Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*, YBS, 2011.

Joint Distributions

- Joint prior for $\boldsymbol{a}_{1:n}$ and $\boldsymbol{M}_{1:n}$

$$p(\boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n}) = \prod_{n'=1}^{n} p(a_{n'}, M_{n'})$$

Measurement generation independent across time n

- Joint likelihood function

$$f(\boldsymbol{z}_{1:n} | \boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n}) = \prod_{n'=1}^{n} f(\boldsymbol{z}_{n'} | \boldsymbol{x}_{n'}, a_{n'}, M_{n'})$$

Measurement noise and clutter independent across time n

The Joint Posterior Distribution

- The joint posterior distribution ($M_{1:n}$ and $\boldsymbol{z}_{1:n}$ are observed and thus fixed)

$$f(\boldsymbol{x}_{0:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) = f(\boldsymbol{x}_{0:n}, \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n} | \boldsymbol{z}_{1:n}) \longleftarrow \quad \boldsymbol{M}_{1:n} \text{ fixed}$$

$$\text{Bayes rule} \longrightarrow \propto f(\boldsymbol{z}_{1:n} | \boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n}) f(\boldsymbol{x}_{0:n}, \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n})$$

$$\boldsymbol{x}_{0:n} \perp\!\!\!\perp \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n} \longrightarrow = f(\boldsymbol{z}_{1:n} | \boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n}) f(\boldsymbol{x}_{0:n}) p(\boldsymbol{a}_{1:n}, \boldsymbol{M}_{1:n})$$

$$\text{Expressions for joint distributions} \longrightarrow = \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) \right) f(\boldsymbol{z}_{n'} | \boldsymbol{x}_{n'}, \boldsymbol{a}_{n'}, M_{n'}) p(\boldsymbol{a}_{n'}, M_{n'})$$

Problem Formulation

- Input at time n:
  - All observations up to time $\boldsymbol{z}_{1:n}$
  - ``Markovian'' statistical model

$$f(\boldsymbol{x}_{0:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) \propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) \right) f(\boldsymbol{z}_{n'} | \boldsymbol{x}_{n'}, \boldsymbol{a}_{n'}, M_{n'}) p(\boldsymbol{a}_{n'}, M_{n'})$$

- Output at time n:
  - Estimates of all $\hat{\boldsymbol{x}}_{i,n}, i \in \{1, \ldots, I\}$

- Calculation of an estimates $\hat{\boldsymbol{x}}_{i,n}$ is based on the marginal posterior pdfs $f(\boldsymbol{x}_{i,n} | \boldsymbol{z}_{1:n})$

The Factor Graph

- Recall factorization of the joint posterior distribution:

$$f(\boldsymbol{x}_{0:n}, \boldsymbol{a}_{1:n} | \boxed{\boldsymbol{z}_{1:n}}) \propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) \right) f(\boxed{\boldsymbol{z}_{n'}} | \boldsymbol{x}_{n'}, \boldsymbol{a}_{n'}, \boxed{M_{n'}}) \, p(\boldsymbol{a}_{n'}, \boxed{M_{n'}})$$

observed and fixed

The Factor Graph

- Recall factorization of the joint posterior distribution:

$$f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) \propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) \right) f(\boldsymbol{z}_{n'} | \boldsymbol{x}_{n'}, \boldsymbol{a}_{n'}, M_{n'}) p(\boldsymbol{a}_{n'}, M_{n'})$$

$$\propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) g_1(\boldsymbol{x}_{i,n'}, a_{i,n'}) g_2(a_{i,n'}) \right) \varphi(\boldsymbol{a}_{n'})$$

$$f(\boldsymbol{z}_n | \boldsymbol{x}_n, \boldsymbol{a}_n, M_n) = \left(\prod_{i \in \mathcal{D}_{\boldsymbol{a}_n}} \frac{f(\boldsymbol{z}_{a_{i,n},n} | \boldsymbol{x}_{i,n})}{f_{\mathrm{c}}(\boldsymbol{z}_{a_{i,n},n})} \right) \boxed{\prod_{m=1}^{M_n} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})} \longleftarrow \text{constant}$$

$$g_1(\boldsymbol{x}_{i,n}, a_{i,n}) = \begin{cases} \frac{f(\boldsymbol{z}_{a_{i,n},n} | \boldsymbol{x}_n)}{f_{\mathrm{c}}(\boldsymbol{z}_{a_{i,n},n})} & a_{i,n} \in \{1, \ldots, M_n\} \\ 1 & a_{i,n} = 0 \end{cases}$$

The Factor Graph

- Recall factorization of the joint posterior distribution:

$$f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) \propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) \right) f(\boldsymbol{z}_{n'} | \boldsymbol{x}_{n'}, \boldsymbol{a}_{n'}, M_{n'}) \, p(\boldsymbol{a}_{n'}, M_{n'})$$

$$\propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) \, g_1(\boldsymbol{x}_{i,n'}, a_{i,n'}) \, g_2(a_{i,n'}) \right) \varphi(\boldsymbol{a}_{n'})$$

$$p(\boldsymbol{a}_n, M_n) = \varphi(\boldsymbol{a}_n) \boxed{\left(\frac{p_\mathrm{d}}{\mu_\mathrm{c}(1 - p_\mathrm{d})} \right)^{|\mathcal{D}_{\boldsymbol{a}_n}|}} \boxed{\frac{e^{-\mu_\mathrm{c}} \mu_\mathrm{c}^{M_n}}{M_n!} (1 - p_\mathrm{d})^I} \longleftarrow \text{constant}$$

$$g_2(a_{i,n}) = \begin{cases} \frac{p_\mathrm{d}}{\mu_\mathrm{c}(1-p_\mathrm{d})} & a_{i,n} \in \{1, \dots, M_n\} \\ 1 & a_{i,n} = 0 \end{cases}$$

The Factor Graph

- **Recall factorization of the joint posterior distribution:**

$$
\begin{aligned}
f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) &\propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) \right) f(\boldsymbol{z}_{n'} | \boldsymbol{x}_{n'}, \boldsymbol{a}_{n'}, M_{n'}) p(\boldsymbol{a}_{n'}, M_{n'}) \\
&\propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) g_1(\boldsymbol{x}_{i,n'}, a_{i,n'}) g_2(a_{i,n'}) \right) \varphi(\boldsymbol{a}_{n'}) \\
&\propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'}) \right) \boxed{\varphi(\boldsymbol{a}_{n'})}
\end{aligned}
$$

Recall: Check if every measurement is generated by at most one object

$$
g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n}) = g_1(\boldsymbol{x}_{i,n}, a_{i,n}) g_2(a_{i,n}) = \begin{cases} \dfrac{p_{\mathrm{d}} f(\boldsymbol{z}_{a_{i,n},n} | \boldsymbol{x}_{i,n})}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{a_{i,n},n})} & a_{i,n} \in \{1, \dots, M_n\} \\ (1 - p_{\mathrm{d}}) & a_{i,n} = 0 \end{cases}
$$

The Factor Graph

- Recall factorization of the joint posterior distribution:

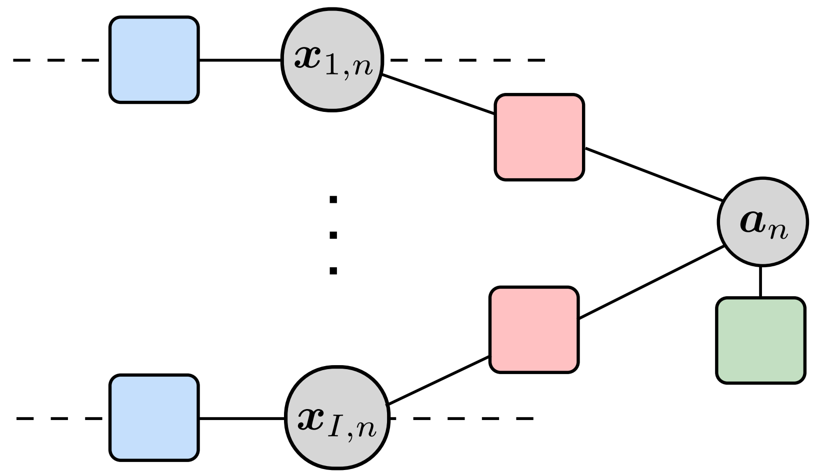
$$f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) \propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) \, g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'}) \right) \varphi(\boldsymbol{a}_{n'})$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Prediction Step

- Prediction step:

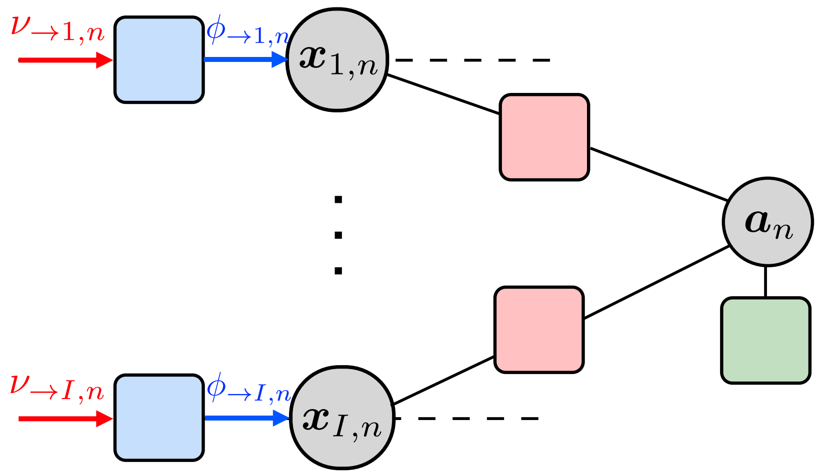
$$\phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) = \int f(\boldsymbol{x}_{i,n}|\boldsymbol{x}_{i,n-1})\, \nu_{\rightarrow i,n}(\boldsymbol{x}_{i,n-1})\, \mathrm{d}\boldsymbol{x}_{i,n-1}$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



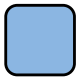
Measurement Evaluation

- Measurement evaluation:

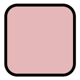
$$\nu_{a_{i,n}}(\boldsymbol{x}_{i,n}) = \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$$

$$\phi_{a_{i,n}}(a_{i,n}) = \int g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \nu_{a_{i,n}}(\boldsymbol{x}_{i,n})\, \mathrm{d}\boldsymbol{x}_{i,n}$$

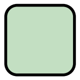
- Factor graph for time step n



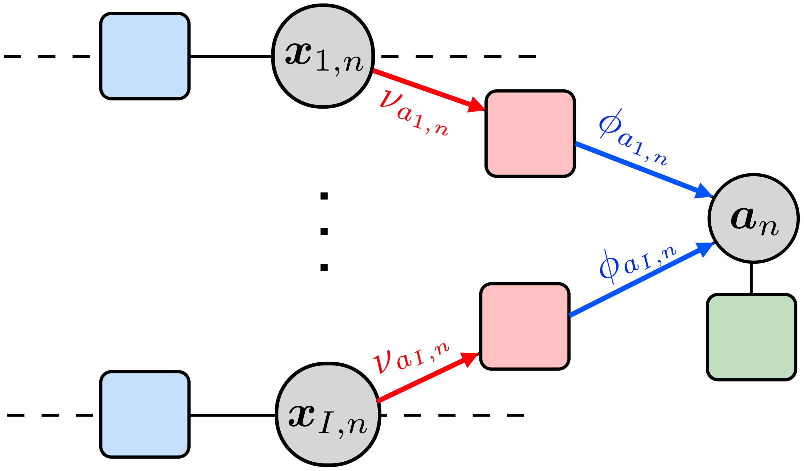
$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$



$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$



$\varphi(\boldsymbol{a}_n)$



Data Association

- Data association:

$$\nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n) = \varphi(\boldsymbol{a}_n) \prod_{\substack{i=1 \\ i=i'}} \phi_{a_{i',n}}(a_{i',n})$$

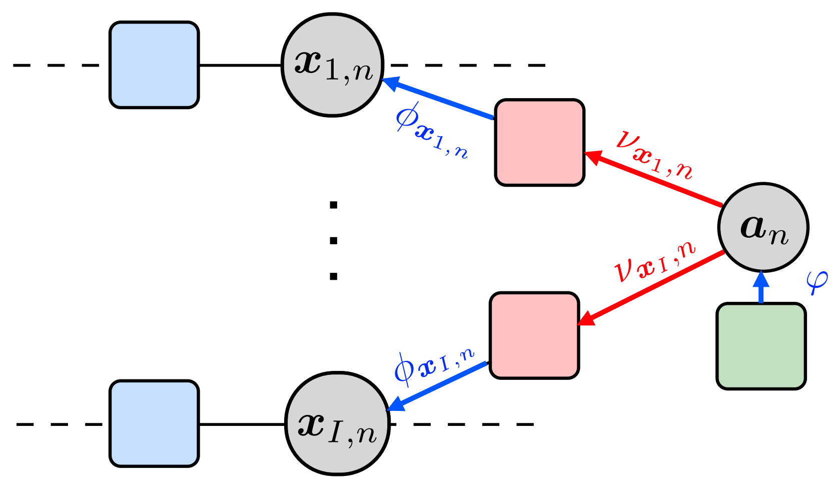
$$\phi_{\boldsymbol{x}_{i,n}}(\boldsymbol{x}_{i,n}) = \sum_{\boldsymbol{a}_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n)$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Update Step

- Update step:

$$\tilde{f}(\boldsymbol{x}_{i,n}) \propto \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})\,\phi_{\boldsymbol{x}_{i,n}}(\boldsymbol{x}_{i,n})$$

$$\nu_{\rightarrow i,n+1}(\boldsymbol{x}_{i,n}) = \phi_{i,n}(\boldsymbol{x}_{i,n})\,\phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$$

$$\boxed{f(\boldsymbol{x}_{i,n}|\boldsymbol{z}_{1:n}) \approx \tilde{f}(\boldsymbol{x}_{i,n})}$$

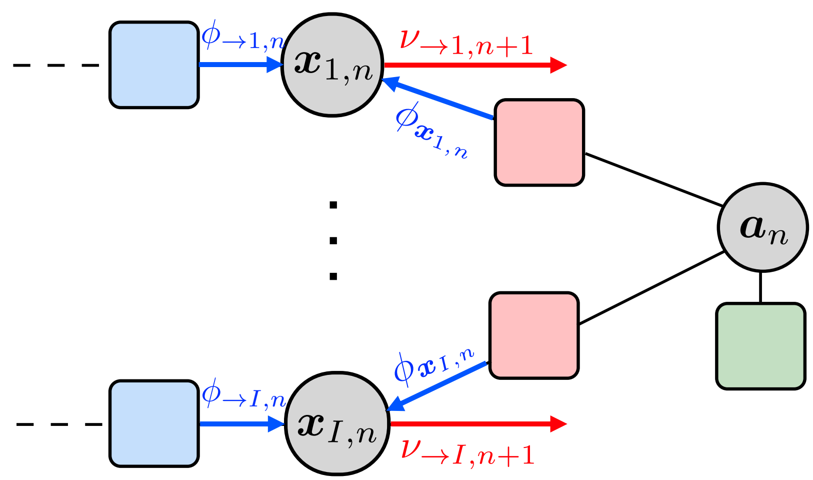
approx. since factor graph is not cycle-free

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Summary

- Multiobject tracking
  - possible association events are modelled by a discrete random vector
  - measurement-origin uncertainty leads to a coupling of sequential estimation problems
  - the joint sequential estimation problem can be represented by a factor graph with cycles
  - approximate marginal posterior distributions can be calculated by passing messages on the factor graph

Graph-Based Multiobject Tracking

Part 4: Graph-Based Processing II

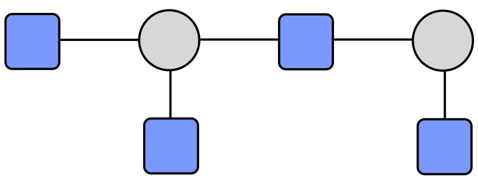
Florian Meyer
joint work with Jason Williams, Paolo Braca, Peter Willett, and Franz Hlawatsch

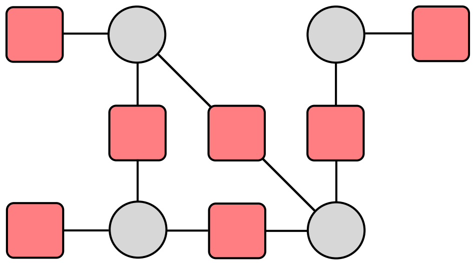
Scripps Institution of Oceanography
Electrical and Computer Engineering Department
University of California San Diego

UC San Diego
JACOBS SCHOOL OF ENGINEERING

Tree vs Cyclic Graph

- A factor graph represents the statistical model of an inference problem
- Message passing (the sum-product algorithm) on a factor graph can strongly reduce the computational complexity of calculating marginal distributions
- Marginalization is exact if the **factor graph is a tree** and approximate if the **factor graph has cycles**

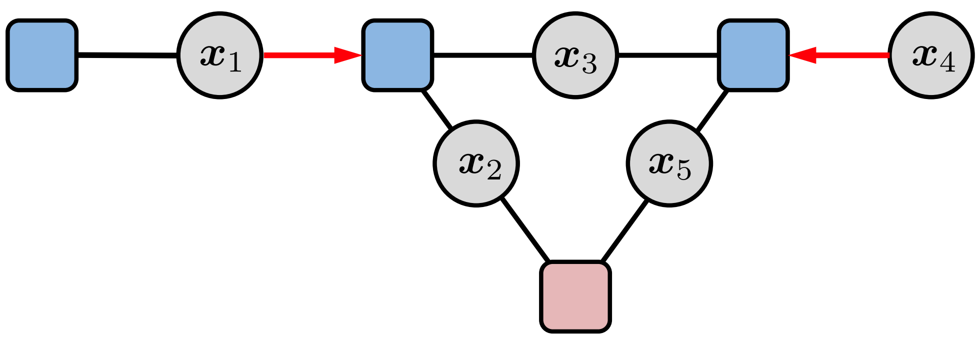




- The factor graph is often not unique. A more ``detailed'' graph
  - has lower-dimensional operations → lower computational complexity
  - may introduce additional cycles → lower inference accuracy

Factor Graphs with Cycles

- Problem: When the factor graphs has cycles, message passing gets stuck
- Solution: Determine message passing order by introducing artificial constant messages
- New Problem 1: Message passing keeps running forever
- Solution: Stop message passing after some time and compute marginals
- New Problem 2: Message passing tend to very large or very small values
  $\longrightarrow$ numerical issues
- Solution: After calculating a message, normalize it

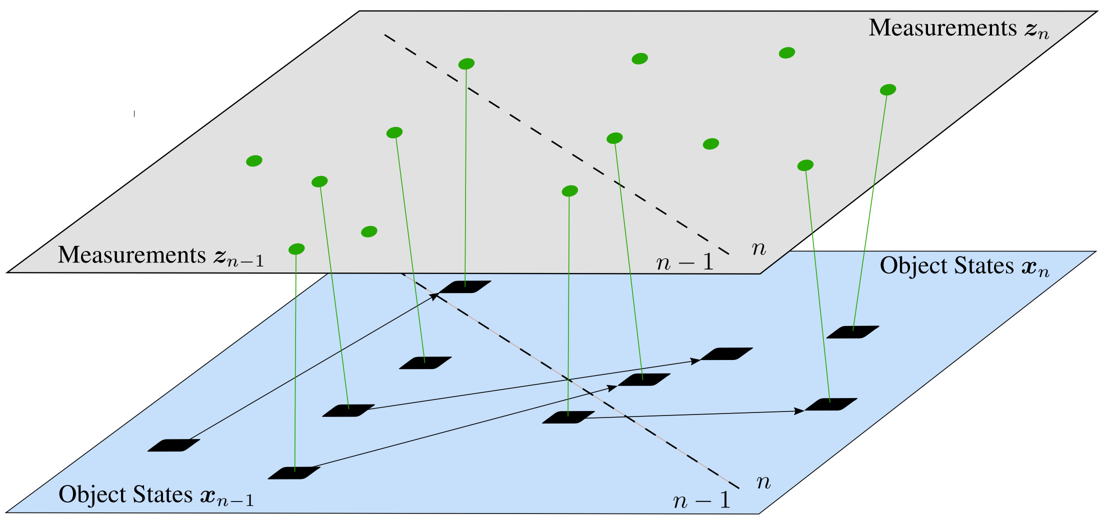


Factor Graphs with Cycles

- It turns out that when we normalize messages
  - the sum-product algorithm (SPA) can still give good results
  - marginal posterior distributions are not exact, but approximations (except means for Gaussian models)
- Approximate marginal distributions are called ``beliefs''
- The SPA on loopy graphs can be derived by approximating the posterior by the ``Bethe free energy''
- Theoretical performance analysis is notoriously difficult in general
- Many practical applications of the SPA involve factor graphs with cycles: turbo codes, LDPB codes, MIMO detection, cooperative localization, data association, …

The Multiobject Tracking Problem

- At each time n: localize and track multiple objects $\boldsymbol{x}_n = [\boldsymbol{x}_{1,n}^{\mathrm{T}} \ldots \boldsymbol{x}_{I,n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{1,n}^{\mathrm{T}} \ldots \boldsymbol{z}_{M_n,n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin
- Data association is challenging because of false clutter measurements and missing measurements



Prior Distributions

- Assumptions:
  1. Object detections are independent Bernoulli trials with success probability $0 < p_{\mathrm{d}} \leqslant 1$
  2. The number of clutter measurements is Poisson distributed with mean μ_{c}
  3. At most one measurement is generated by each object
  4. A measurement can be generated from at most one object
- Assumptions 1-3 are parallel to the single object tracking case
- Every association event expressed by a vector $\boldsymbol{a}_n = [a_{1,n} \dots a_{I,n}]^{\mathrm{T}}$ automatically fulfills Assumption 3 (scalar association variable $a_{i,n}$ for each object)
- Assumption 4 can be enforced by the following check function

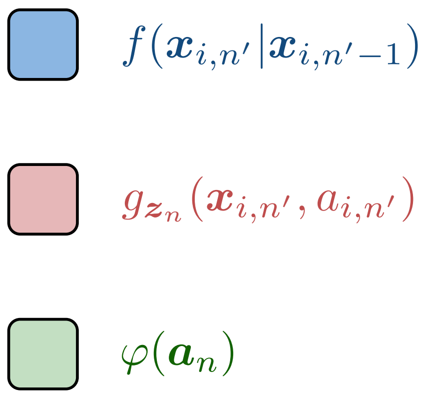
$$\varphi(\boldsymbol{a}_n) \triangleq \begin{cases} 0, & \exists i, j \in \{1, 2, \dots, I\} \text{ such that } i \neq j \text{ and } a_{i,n} = a_{j,n} \neq 0 \\ 1, & \text{otherwise} \end{cases}$$

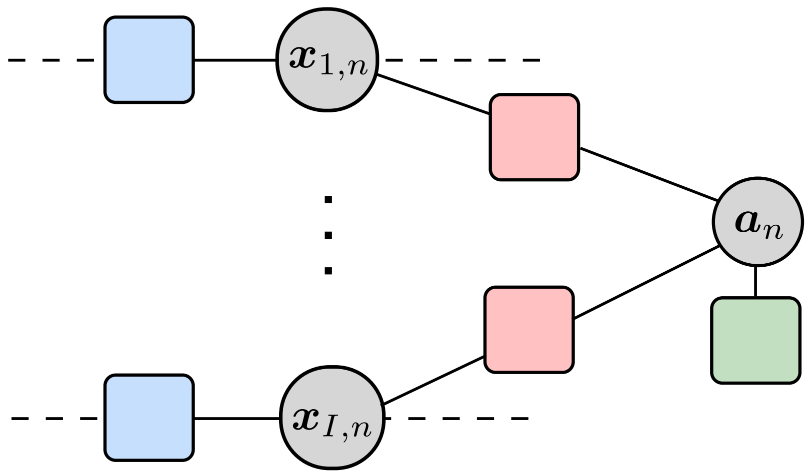
The Factor Graph

- Recall factorization of the joint posterior distribution:

$$f(\boldsymbol{x}_{1:n}, \boldsymbol{a}_{1:n} | \boldsymbol{z}_{1:n}) \propto \left(\prod_{j=1}^{I} f(\boldsymbol{x}_{j,0}) \right) \prod_{n'=1}^{n} \left(\prod_{i=1}^{I} f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1}) \, g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'}) \right) \varphi(\boldsymbol{a}_{n'})$$

- Factor graph for time step n





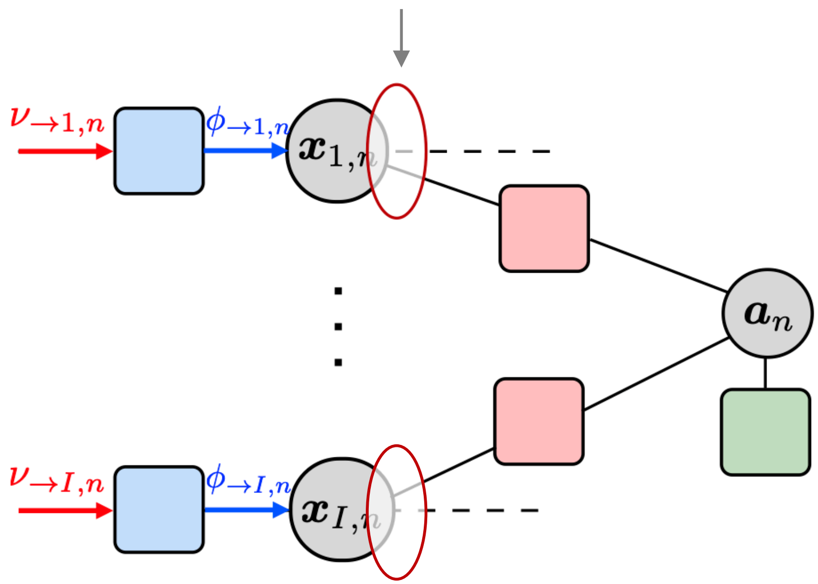
Message Passing Order

- Recall Prediction step:

$$\phi_{\to i,n}(\boldsymbol{x}_{i,n}) = \int f(\boldsymbol{x}_{i,n}|\boldsymbol{x}_{i,n-1})\,\nu_{\to i,n}(\boldsymbol{x}_{i,n-1})\,\mathrm{d}\boldsymbol{x}_{i,n-1}$$

Here we would get stuck since messages from two edges are not available

- Factor graph for time step n



$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$

Message Passing Order

- Measurement evaluation:

$$\nu_{a_{i,n}}(\boldsymbol{x}_{i,n}) = \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$$

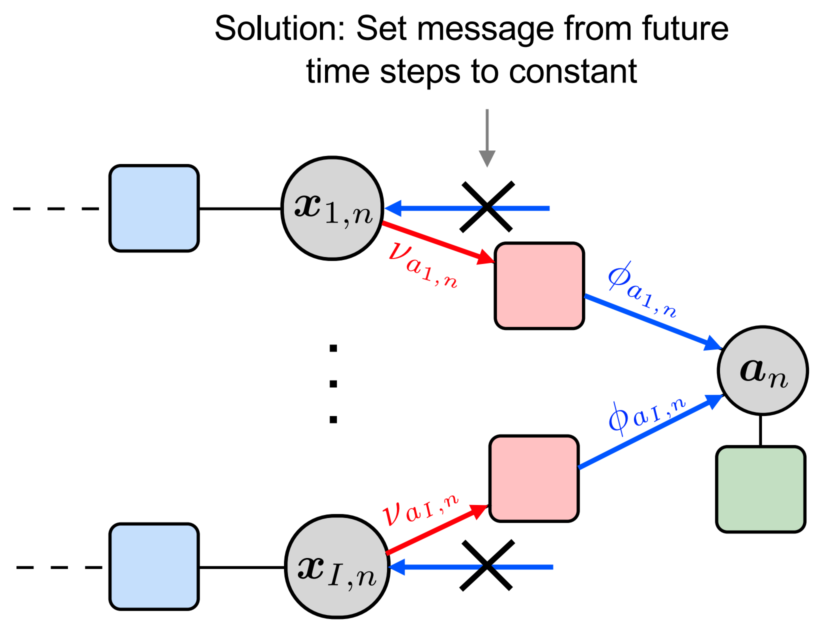
$$\phi_{a_{i,n}}(a_{i,n}) = \int g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \nu_{a_{i,n}}(\boldsymbol{x}_{i,n})\, \mathrm{d}\boldsymbol{x}_{i,n}$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'} | \boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Update Step

- Update step:

$$\tilde{f}(\boldsymbol{x}_{i,n}) \propto \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})\, \phi_{\boldsymbol{x}_{i,n}}(\boldsymbol{x}_{i,n})$$

$$\nu_{\rightarrow i,n+1}(\boldsymbol{x}_{i,n}) = \phi_{i,n}(\boldsymbol{x}_{i,n})\, \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$$

$$\boxed{f(\boldsymbol{x}_{i,n}|\boldsymbol{z}_{1:n}) \approx \tilde{f}(\boldsymbol{x}_{i,n})}$$

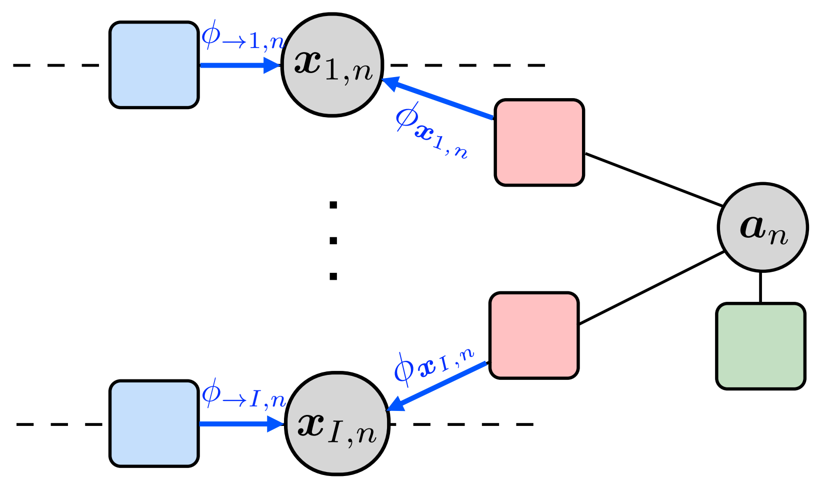
approx. since factor graph is not cycle-free

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Update Step

- Update step revised

$$\tilde{f}(\boldsymbol{x}_{i,n}) \propto \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) \boxed{\phi_{\boldsymbol{x}_{i,n}}(\boldsymbol{x}_{i,n})}$$

Data association message derived in class 10

$$= \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) \sum_{\boldsymbol{a}_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n)$$

$$= \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) \sum_{a_{i,n}=0}^{M_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n}) \underbrace{\sum_{a_{1,n}=0}^{M_n} \cdots \sum_{a_{i-1,n}=0}^{M_n} \sum_{a_{i+1,n}=0}^{M_n} \cdots \sum_{a_{I,n}=0}^{M_n} \nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n)}_{\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})}$$

$$= \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) \sum_{a_{i,n}=0}^{M_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$$

$$= \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) \sum_{a_{i,n}=0}^{M_n} \tilde{g}_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})$$

influence of other objects in the environment

where $\tilde{g}_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n}) = g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n}) \boxed{\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})}$

- The single object updated step of the multiobject tracking solution has the same form as the single object tracking in clutter update step

Multiobject Tracking Filters

- Let's assume at time n, approximate posteriors $\tilde{f}(\boldsymbol{x}_{i,n-1}) \approx f(\boldsymbol{x}_{i,n-1}|\boldsymbol{z}_{1:n-1})$ for all objects $i \in \{1,\ldots,I\}$ are available

- We can develop a multiobject tracking algorithm by performing for each $i \in \{1,\ldots,I\}$
  - the conventional prediction step, i.e., $\phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) = \int f(\boldsymbol{x}_{i,n}|\boldsymbol{x}_{i,n-1})\,\tilde{f}(\boldsymbol{x}_{i,n-1})\mathrm{d}\boldsymbol{x}_{i,n-1}$
  - calculation of $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$
  - the update step of the single object tracking (in clutter) solution where $g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})$ is replaced by $\tilde{g}_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n}) = g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\,\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$

- Multiobject tracking is based on the calculation of $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$

$\longrightarrow$ Joint probabilistic data association

Closed-Form Update Step (cf. Class 9)

- Step 1: Calculate means and covariances of mixture components:

$$\boldsymbol{\mu}_m = \boldsymbol{\mu}^{-}_{\boldsymbol{x}_{i,n}} + \boldsymbol{K}_{i,n}\big(\boldsymbol{z}_{m,n} - \boldsymbol{H}_{i,n}\boldsymbol{\mu}^{-}_{\boldsymbol{x}_{i,n}}\big) \quad m = 1,\dots,M_n$$

$$\boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}} - \boldsymbol{K}_{i,n}\boldsymbol{H}_{i,n}\boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}}$$

$$\boldsymbol{K}_{i,n} = \boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}}\boldsymbol{H}^{\mathrm{T}}_{i,n}\big(\boldsymbol{H}_{i,n}\boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}}\boldsymbol{H}^{\mathrm{T}}_{i,n} + \boldsymbol{\Sigma}_{\boldsymbol{v}_{i,n}}\big)^{-1}$$

$$\boldsymbol{\mu}_{M_n+1} = \boldsymbol{\mu}^{-}_{\boldsymbol{x}_{i,n}}$$

$$\boldsymbol{\Sigma}_{M_n+1} = \boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}}$$

- Step 2: Calculate unnormalized weights:

$$\tilde{w}_m = \frac{p_{\mathrm{d}} f_{\mathrm{g}}(\boldsymbol{z}_{m,n}; \boldsymbol{H}_{i,n}\boldsymbol{\mu}^{-}_{\boldsymbol{x}_{i,n}}, \boldsymbol{H}_{i,n}\boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}}\boldsymbol{H}^{\mathrm{T}}_{i,n} + \boldsymbol{\Sigma}_{\boldsymbol{v}_{i,n}})\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n} = m)}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})} \quad m = 1,\dots,M_n$$

$$\tilde{w}^{(i)}_{M_n+1} = (1 - p_{\mathrm{d}})\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n} = 0)$$

- Step 3: Normalize weights: $w_m = \tilde{w}_m / \big(\sum_{m'=1}^{M_n+1} \tilde{w}_{m'}\big)$

- Step 4: Approximate Gaussian mixture by a single Gaussian with same mean and covariance (moment matching):

$$\boldsymbol{\mu}_{\boldsymbol{x}_{i,n}} = \sum_{m=1}^{M_n+1} w_m \boldsymbol{\mu}_m$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_{i,n}} = \sum_{m=1}^{M_n+1} w_m \boldsymbol{\Sigma}_m + \sum_{m=1}^{M_n+1} w_m \boldsymbol{\mu}_m \boldsymbol{\mu}^{\mathrm{T}}_m - \boldsymbol{\mu}_{\boldsymbol{x}_{i,n}} \boldsymbol{\mu}^{\mathrm{T}}_{\boldsymbol{x}_{i,n}}$$

- **Result:** Mean $\boldsymbol{\mu}_{\boldsymbol{x}_{i,n}}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_{i,n}}$ representing the posterior distribution $\tilde{f}(\boldsymbol{x}_{i,n})$

Y. Bar-Shalom, F. Daum, and J. Huang, *The Probabilistic Data Association Filter*, IEEE Contr. Syst. Mag., 2009

Particle-Based Update Step (cf. Class 4)

- **Given:** Particles $\{(\boldsymbol{x}_{i,n}^{(j)})\}_{j=1}^{J} \simeq \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$ representing the predicted posterior PDF
- **Wanted:** Particles $\{(\overline{\boldsymbol{x}}_{i,n}^{(j)})\}_{j=1}^{J} \simeq \tilde{f}(\boldsymbol{x}_{i,n})$ representing the posterior PDF
- Perform importance sampling with proposal distribution $f_{\mathrm{p}}(\boldsymbol{x}_{i,n}) = \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$ and target distribution $f_{\mathrm{t}}(\boldsymbol{x}_{i,n}) \propto \phi_{\boldsymbol{x}_{i,n}}(\boldsymbol{x}_{i,n})\phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) = \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) \sum_{m=0}^{M_n} \tilde{g}_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n} = m)$
  - calculate unnormalized weights $\tilde{w}_{i,n}^{(j)} = \boxed{\sum_{m=0}^{M_n} \tilde{g}_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}^{(j)}, a_{i,n} = m)} \propto f_{\mathrm{t}}(\boldsymbol{x}_{i,n}^{(j)})/f_{\mathrm{p}}(\boldsymbol{x}_{i,n}^{(j)})$
  - normalize weights $w_{i,n}^{(j)} = \tilde{w}_{i,n}^{(j)} / \sum_{j'=1}^{J} \tilde{w}_{i,n}^{(j')}, \quad j = 1, \dots, J$
- Perform resampling to get $\{(\overline{\boldsymbol{x}}_{i,n}^{(j)})\}_{j=1}^{J} \simeq \tilde{f}(\boldsymbol{x}_{i,n})$ from $\{(\boldsymbol{x}_{i,n}^{(j)}, w_{i,n}^{(j)})\}_{j=1}^{J} \simeq \tilde{f}(\boldsymbol{x}_{i,n})$

Joint Probabilistic Data Association

- Recall measurement evaluation step:

$$\nu_{a_{i,n}}(\boldsymbol{x}_{i,n}) = \phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n})$$

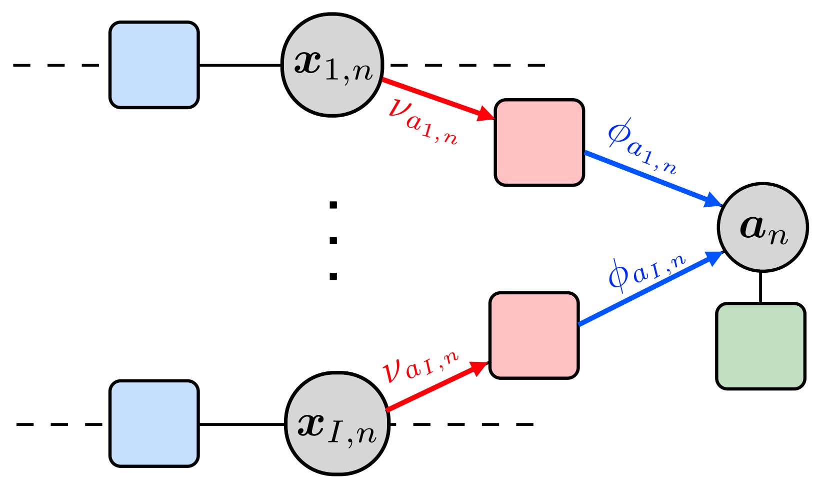
$$\phi_{a_{i,n}}(a_{i,n}) = \int g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \nu_{a_{i,n}}(\boldsymbol{x}_{i,n})\, \mathrm{d}\boldsymbol{x}_{i,n}$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Joint Probabilistic Data Association

- Closed-form measurement evaluation step (for linear-Gaussian meas. models)

$$\phi_{a_{i,n}}(a_{i,n} = 0) = (1 - p_{\mathrm{d}}) \qquad \nu_{a_{i,n}}(\boldsymbol{x}_{i,n}) = f_{\mathrm{g}}(\boldsymbol{x}_{i,n}; \boldsymbol{\mu}^{-}_{\boldsymbol{x}_{i,n}}, \boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}})$$

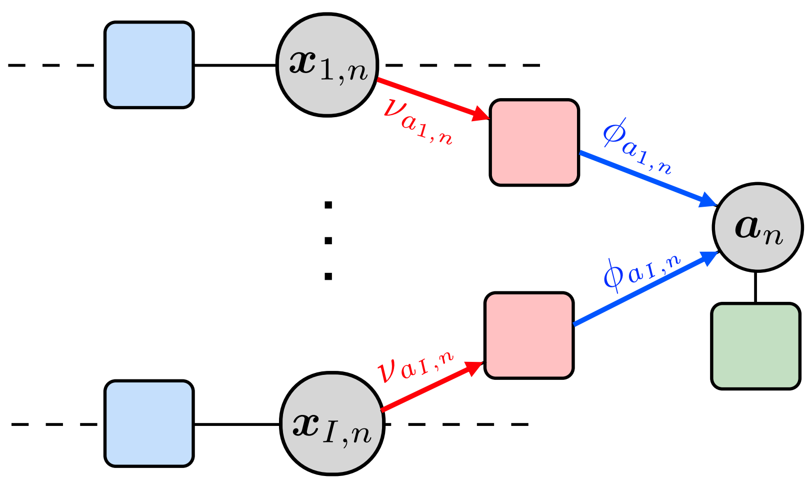
$$\phi_{a_{i,n}}(a_{i,n} = m) = \frac{p_{\mathrm{d}} f_{\mathrm{g}}(\boldsymbol{z}_{m,n}; \boldsymbol{H}_{i,n}\boldsymbol{\mu}^{-}_{\boldsymbol{x}_{i,n}}, \boldsymbol{H}_{i,n}\boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}}\boldsymbol{H}^{\mathrm{T}}_{i,n} + \boldsymbol{\Sigma}_{\boldsymbol{v}_{i,n}})}{\mu_{\mathrm{c}} f_{\mathrm{c}}(\boldsymbol{z}_{m,n})}, \; m \in \{1, \ldots, M_n\}$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Joint Probabilistic Data Association

- Particle-based measurement evaluation step

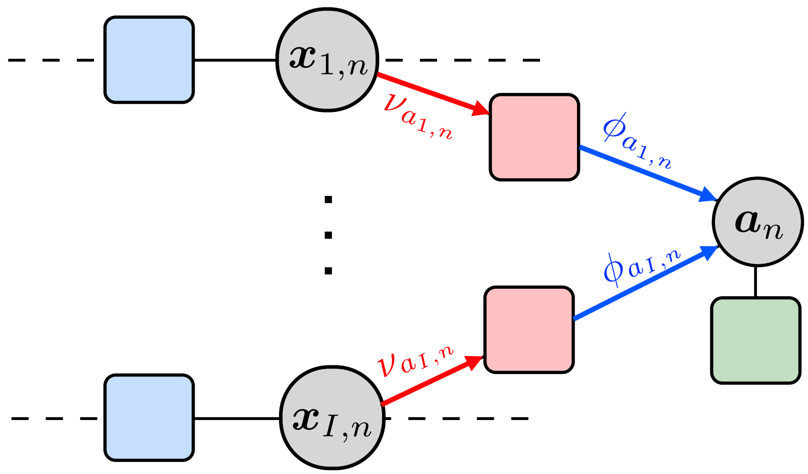
$$\phi_{a_{i,n}}(a_{i,n}) \approx \frac{1}{J}\sum_{j=1}^{J} g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}^{(j)}, a_{i,n}) \qquad \nu_{a_{i,n}}(\boldsymbol{x}_{i,n}) \simeq \{(\boldsymbol{x}_{i,n}^{(j)})\}_{j=1}^{J}$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Joint Probabilistic Data Association

- Recall data association step:

$$\nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n) = \varphi(\boldsymbol{a}_n) \prod_{\substack{i=1 \\ i=i'}} \phi_{a_{i',n}}(a_{i',n})$$

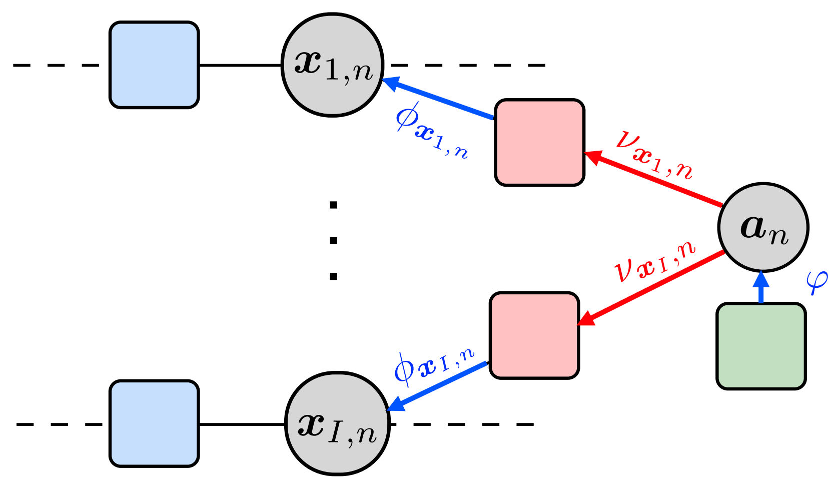
$$\phi_{\boldsymbol{x}_{i,n}}(\boldsymbol{x}_{i,n}) = \sum_{\boldsymbol{a}_n} g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\, \nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n)$$

- Factor graph for time step n

$f(\boldsymbol{x}_{i,n'}|\boldsymbol{x}_{i,n'-1})$

$g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n'}, a_{i,n'})$

$\varphi(\boldsymbol{a}_n)$



Joint Probabilistic Data Association

- Data association:

$$\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n}) = \sum_{a_{1,n}=0}^{M_n} \cdots \sum_{a_{i-1,n}=0}^{M_n} \sum_{a_{i+1,n}=0}^{M_n} \cdots \sum_{a_{I,n}=0}^{M_n} \nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n)$$

$$= \sum_{a_{1,n}=0}^{M_n} \cdots \sum_{a_{i-1,n}=0}^{M_n} \sum_{a_{i+1,n}=0}^{M_n} \cdots \sum_{a_{I,n}=0}^{M_n} \varphi(\boldsymbol{a}_n) \prod_{\substack{i'=1 \\ i' \neq i}}^{I} \phi_{a_{i',n}}(a_{i',n})$$

$$\varphi(\boldsymbol{a}_n) \triangleq \begin{cases} 0, & \exists i, j \in \{1, 2, \ldots, I\} \text{ such that } i \neq j \text{ and } a_{i,n} = a_{j,n} \neq 0 \\ 1, & \text{otherwise} \end{cases}$$

- Computational complexity of calculating $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$ scales as $\mathcal{O}\big((M_n + 1)^I\big)$ and is thus only feasible for small I

$\longrightarrow$ need scalable methods for approximate calculation of $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$

Data Association Representations

- For simplicity we consider a single time step and drop n in the notation
- Recall description of object-measurement associations by object-oriented association vectors $\boldsymbol{a} = [a_1, a_2, \ldots, a_I]^{\mathrm{T}}$

$$a_i \triangleq \begin{cases} m \in \{1, 2, \ldots, M\}, & \text{if object } i \text{ generated measurement } m \\ 0 & \text{if object } i \text{ did not generate a measurement} \end{cases}$$

- Alternative description of object-measurement associations by measurement-oriented association vectors $\boldsymbol{b} = [b_1, b_2, \ldots, b_M]^{\mathrm{T}}$ with entries

$$b_m \triangleq \begin{cases} i \in \{1, 2, \ldots, I\}, & \text{if measurement } m \text{ is generated by object } i \\ 0 & \text{if measurement } m \text{ was not generated by an object} \end{cases}$$

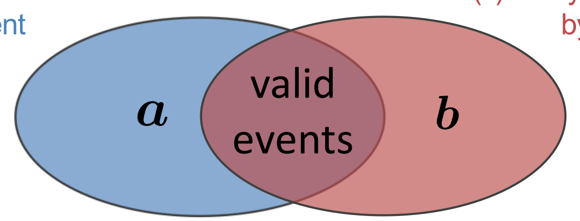
- Recall data association assumptions: An (i) object can generate at most one measurement and a (ii) measurement can be generated by at most one object

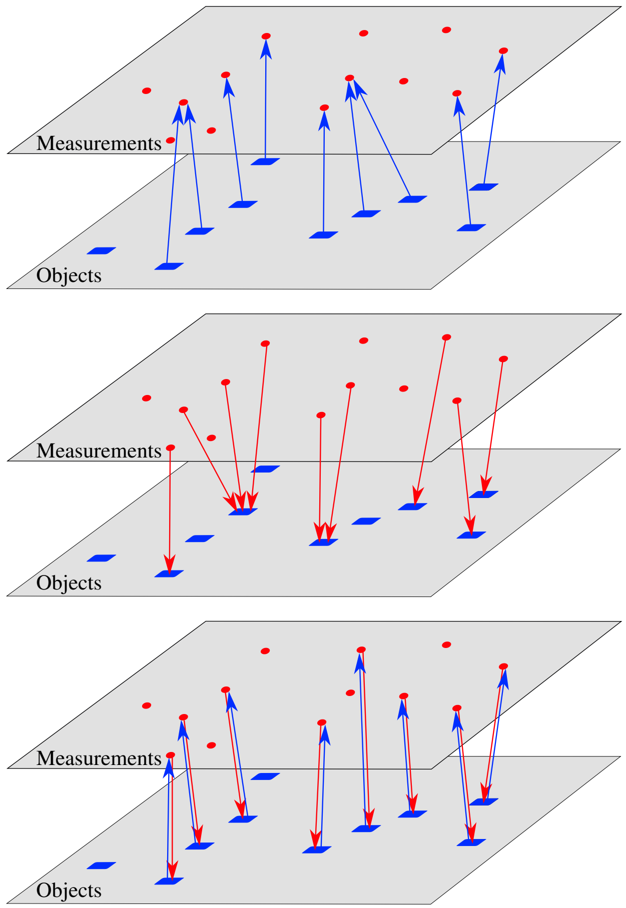
Data Association Representations

- Events represented by object-oriented vector $\boldsymbol{a} = [a_1, a_2, \ldots, a_I]^{\mathrm{T}}$ satisfy property (i)
- Events represented by measurement-oriented vector $\boldsymbol{b} = [b_1, b_2, \ldots, b_M]^{\mathrm{T}}$ satisfy property (ii)
- Events represented by object-oriented $\boldsymbol{a}$ and measurement-oriented $\boldsymbol{b}$ satisfy (i) and (ii)

(i) every object generates at most one measurement

(ii) every measurement is generated by at most one object

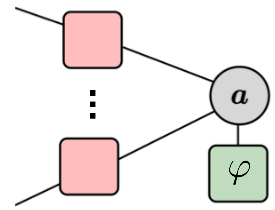




``Stretching'' the Graph

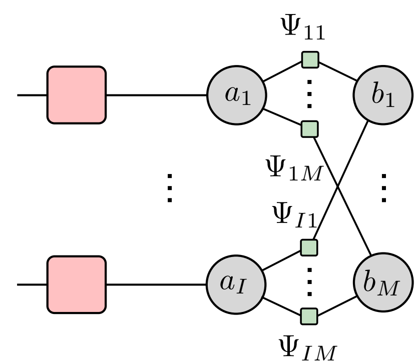
- We use a hybrid description of data association uncertainty to replace $\varphi(\boldsymbol{a})$ by

$$\psi(\boldsymbol{a}, \boldsymbol{b}) \propto \prod_{i=1}^{I} \prod_{m=1}^{M} \Psi_{im}(a_i, b_m)$$



$$\Psi_{im}(a_i, b_m) \triangleq \begin{cases} 0, & a_i = m,\ b_m \neq i \\ & \text{or } b_m = i,\ a_i \neq m \\ 1, & \text{otherwise.} \end{cases}$$

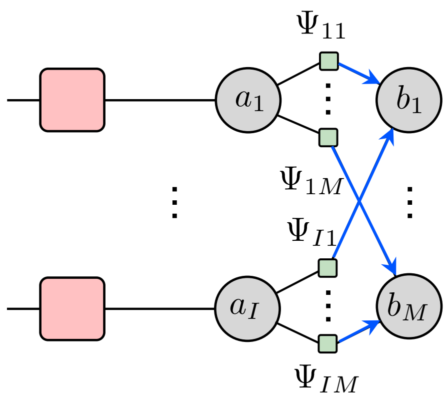
- Properties of $\psi(\boldsymbol{a}, \boldsymbol{b})$:
  - is non-zero only if $\boldsymbol{a}$ and $\boldsymbol{b}$ describe the same event
  - checks consistency by low-dimensional factors $\Psi_{km}(a_k, b_m)$
  - does not alter marginal distributions since there is a deterministic one-to-one mapping from $\boldsymbol{a}$ to $\boldsymbol{b}$ and $\varphi(\boldsymbol{a}) = \sum_{\boldsymbol{b}} \psi(\boldsymbol{a}, \boldsymbol{b})$



Loopy SPA for Joint Probabilistic Data Association

- Stretching the graph enables calculation of approximate $\tilde{\kappa}_{\boldsymbol{x}_{i,n}}(a_i)$ by means of the loopy SPA

- At message passing iteration $\ell \in \{1, \dots, L\}$ we calculate the following SPA messages in parallel



$$\phi^{[\ell]}_{\Psi_{i,m}\to a_i}(a_i) = \sum_{b_m=0}^{I} \Psi_{im}(a_i, b_m) \prod_{\substack{i'=1\\ i'\neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm}\to b_m}(b_m)$$

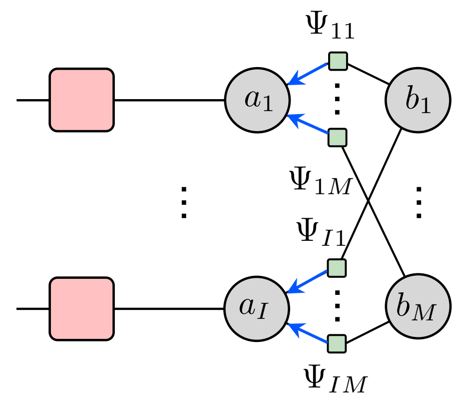
$$\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_{i,n}}(a_i)\, \Psi_{im}(a_i, b_m) \prod_{\substack{m'=1\\ m'\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'}\to a_i}(a_i)$$

- Initialization at $\ell = 0$: $\phi^{[0]}_{\Psi_{im}\to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\, \Psi_{im}(a_i, b_m)$

Loopy SPA for Joint Probabilistic Data Association

- Stretching the graph enables calculation of approximate $\tilde{\kappa}_{\boldsymbol{x}_{i,n}}(a_i)$ by means of the loopy SPA

- At message passing iteration $\ell \in \{1, \ldots, L\}$ we calculate the following SPA messages in parallel



$$\phi^{[\ell]}_{\Psi_{i,m} \to a_i}(a_i) = \sum_{b_m=0}^{I} \Psi_{im}(a_i, b_m) \prod_{\substack{i'=1 \\ i' \neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm} \to b_m}(b_m)$$

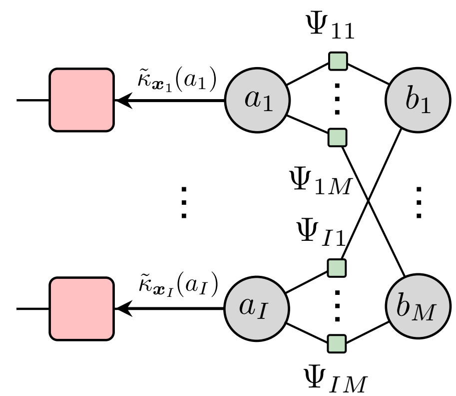
$$\phi^{[\ell]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_{i,n}}(a_i)\, \Psi_{im}(a_i, b_m) \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'} \to a_i}(a_i)$$

- Initialization at $\ell = 0$: $\phi^{[0]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\, \Psi_{im}(a_i, b_m)$

Loopy SPA for Joint Probabilistic Data Association

- Stretching the graph enables calculation of approximate $\tilde{\kappa}_{\boldsymbol{x}_{i,n}}(a_i)$ by means of the loopy SPA

- At message passing iteration $\ell \in \{1,\dots,L\}$ we calculate the following SPA messages in parallel



$$\phi^{[\ell]}_{\Psi_{i,m}\rightarrow a_i}(a_i) = \sum_{b_m=0}^{I} \Psi_{im}(a_i,b_m) \prod_{\substack{i'=1\\ i'\neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm}\rightarrow b_m}(b_m)$$

$$\phi^{[\ell]}_{\Psi_{im}\rightarrow b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_{i,n}}(a_i)\, \Psi_{im}(a_i,b_m) \prod_{\substack{m'=1\\ m'\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'}\rightarrow a_i}(a_i)$$

- Initialization at $\ell=0$: $\phi^{[0]}_{\Psi_{im}\rightarrow b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\, \Psi_{im}(a_i,b_m)$

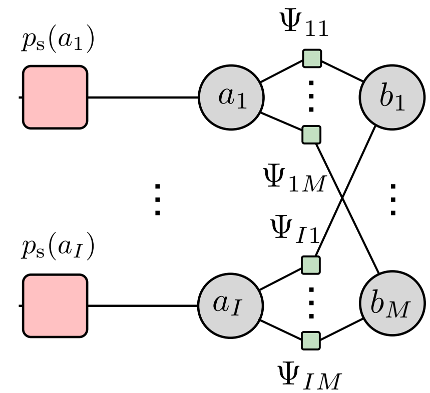
- Result after $\ell = L$ iterations: $\tilde{\kappa}_{\boldsymbol{x}_i}(a_i) = \prod_{m=1}^{M} \phi^{[L]}_{\Psi_{im}\rightarrow a_i}(a_i)$

General Probabilistic Assignment Algorithm

- Calculate joint assignment probabilities $p_{\mathrm{j}}(a_i)$ from single assignment probabilities $p_{\mathrm{s}}(a_i)$
- At message passing iteration $\ell \in \{1, \ldots, L\}$ we calculate the following SPA messages in parallel

$$\phi^{[\ell]}_{\Psi_{i,m} \to a_i}(a_i) = \sum_{b_m=0}^{I} \Psi_{im}(a_i, b_m) \prod_{\substack{i'=1 \\ i' \neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm} \to b_m}(b_m)$$

$$\phi^{[\ell]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} p_{\mathrm{s}}(a_i)\, \Psi_{im}(a_i, b_m) \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'} \to a_i}(a_i)$$



- Initialization at $\ell = 0$: $\phi^{[0]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} p_{\mathrm{s}}(a_i)\, \Psi_{im}(a_i, b_m)$
- Result after $\ell = L$ iterations: $p_{\mathrm{j}}(a_i) = p_{\mathrm{s}}(a_i) \prod_{m=1}^{M} \phi^{[L]}_{\Psi_{im} \to a_i}(a_i)$
- Calculate MAP assignments $\hat{a}_i = \operatorname{argmax} p_{\mathrm{j}}(a_i), \qquad i \in \{1, \ldots, I\}$

Summary

- On factor graphs with cycles, the SPA
  - has to be performed iteratively
  - relies on a predefined message passing order
  - only provides approximate marginal posteriors
- The multiobject tracking problem can be represented by a factor graphs with cycles and solved by means of the SPA (messages are only send forward in time)
- The complexity of joint probabilistic data association for multiobject tracking scales exponentially with the number of objects I
- By making modifications to the graph, the scalability of joint probabilistic data association can be increased

Graph-Based Multiobject Tracking

Part 5: Graph-Based Processing III

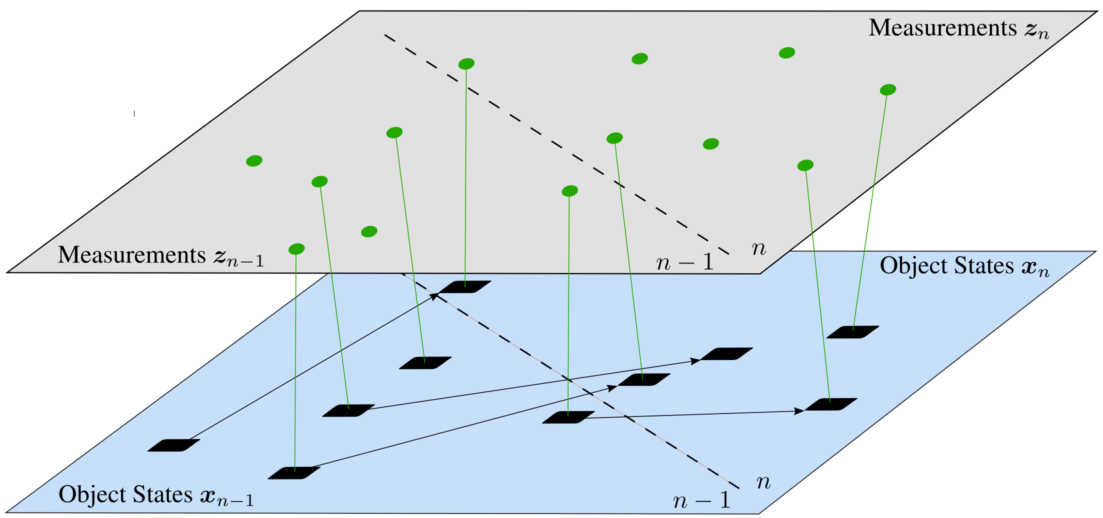
Florian Meyer
joint work with Jason Williams, Paolo Braca, Peter Willett, and Franz Hlawatsch

Scripps Institution of Oceanography
Electrical and Computer Engineering Department
University of California San Diego

The Multiobject Tracking Problem

- At each time n: localize and track multiple objects $\boldsymbol{x}_n = [\boldsymbol{x}_{1,n}^{\mathrm{T}} \ldots \boldsymbol{x}_{I,n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{1,n}^{\mathrm{T}} \ldots \boldsymbol{z}_{M_n,n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin

- Data association is challenging because of false clutter measurements and missing measurements



Multiobject Tracking Filters

- Let's assume at time n, approximate posteriors $\tilde{f}(\boldsymbol{x}_{i,n-1}) \approx f(\boldsymbol{x}_{i,n-1}|\boldsymbol{z}_{1:n-1})$ for all objects $i \in \{1,\ldots,I\}$ are available

- We can develop a multiobject tracking algorithm by performing for each $i \in \{1,\ldots,I\}$
  - the conventional prediction step, i.e., $\phi_{\rightarrow i,n}(\boldsymbol{x}_{i,n}) = \int f(\boldsymbol{x}_{i,n}|\boldsymbol{x}_{i,n-1})\,\tilde{f}(\boldsymbol{x}_{i,n-1})\mathrm{d}\boldsymbol{x}_{i,n-1}$
  - calculation of $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$
  - the update step of the single object tracking (in clutter) solution where $g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})$ is replaced by $\tilde{g}_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n}) = g_{\boldsymbol{z}_n}(\boldsymbol{x}_{i,n}, a_{i,n})\,\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$

- Multiobject tracking is based on the calculation of $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$

$\longrightarrow$ Joint probabilistic data association

Joint Probabilistic Data Association

- Data association:

$$\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n}) = \sum_{a_{1,n}=0}^{M_n} \cdots \sum_{a_{i-1,n}=0}^{M_n} \sum_{a_{i+1,n}=0}^{M_n} \cdots \sum_{a_{I,n}=0}^{M_n} \nu_{\boldsymbol{x}_{i,n}}(\boldsymbol{a}_n)$$

$$= \sum_{a_{1,n}=0}^{M_n} \cdots \sum_{a_{i-1,n}=0}^{M_n} \sum_{a_{i+1,n}=0}^{M_n} \cdots \sum_{a_{I,n}=0}^{M_n} \varphi(\boldsymbol{a}_n) \prod_{\substack{i'=1 \\ i'\neq i}}^{I} \phi_{a_{i',n}}(a_{i',n})$$

$$\varphi(\boldsymbol{a}_n) \triangleq \begin{cases} 0, & \exists i,j \in \{1,2,\ldots,I\} \text{ such that } i \neq j \text{ and } a_{i,n} = a_{j,n} \neq 0 \\ 1, & \text{otherwise} \end{cases}$$

- Computational complexity of calculating $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$ scales as $\mathcal{O}\big((M_n+1)^I\big)$ and is thus only feasible for small I

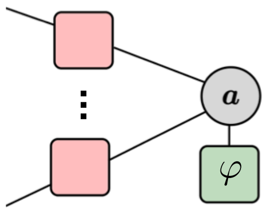
→ need scalable methods for approximate calculation of $\kappa_{\boldsymbol{x}_{i,n}}(a_{i,n})$

``Stretching'' the Graph

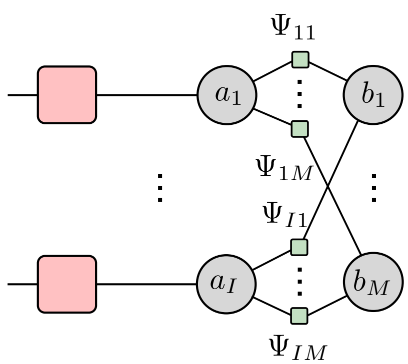
- We use a hybrid description of data association uncertainty to replace $\varphi(\boldsymbol{a})$ by

$$\psi(\boldsymbol{a}, \boldsymbol{b}) \propto \prod_{i=1}^{I} \prod_{m=1}^{M} \Psi_{im}(a_i, b_m)$$

$$\Psi_{im}(a_i, b_m) \triangleq \begin{cases} 0, & a_i = m,\ b_m \neq i \\ & \text{or } b_m = i,\ a_i \neq m \\ 1, & \text{otherwise.} \end{cases}$$



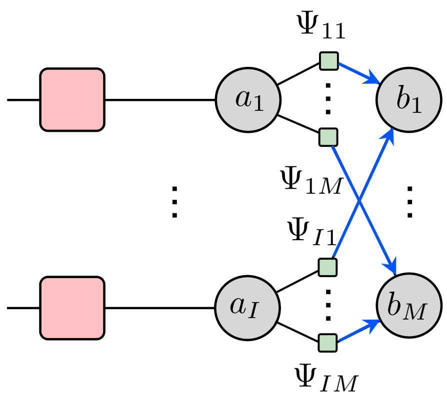
- Properties of $\psi(\boldsymbol{a}, \boldsymbol{b})$:
  - is non-zero only if $\boldsymbol{a}$ and $\boldsymbol{b}$ describe the same event
  - checks consistency by low-dimensional factors $\Psi_{km}(a_k, b_m)$
  - does not alter marginal distributions since there is a deterministic one-to-one mapping from $\boldsymbol{a}$ to $\boldsymbol{b}$ and $\varphi(\boldsymbol{a}) = \sum_{\boldsymbol{b}} \psi(\boldsymbol{a}, \boldsymbol{b})$



Loopy SPA for Joint Probabilistic Data Association

- Stretching the graph enables calculation of approximate $\tilde{\kappa}_{\boldsymbol{x}_{i,n}}(a_i)$ by means of the loopy SPA

- At message passing iteration $\ell \in \{1, \ldots, L\}$ we calculate the following SPA messages in parallel



$$\phi^{[\ell]}_{\Psi_{im} \to a_i}(a_i) = \sum_{b_m=0}^{I} \Psi_{im}(a_i, b_m) \prod_{\substack{i'=1 \\ i' \neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm} \to b_m}(b_m)$$

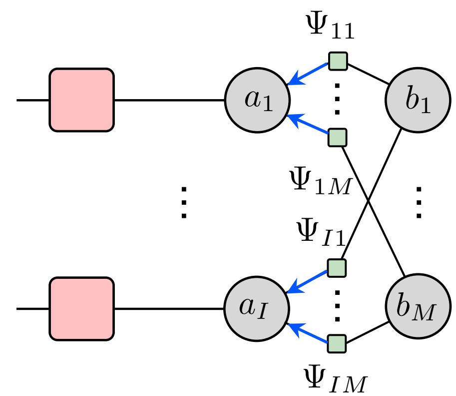
$$\phi^{[\ell]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_{i,n}}(a_i)\, \Psi_{im}(a_i, b_m) \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'} \to a_i}(a_i)$$

- Initialization at $\ell = 0$: $\phi^{[0]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\, \Psi_{im}(a_i, b_m)$

Loopy SPA for Joint Probabilistic Data Association

- Stretching the graph enables calculation of approximate $\tilde{\kappa}_{\boldsymbol{x}_{i,n}}(a_i)$ by means of the loopy SPA

- At message passing iteration $\ell \in \{1,\dots,L\}$ we calculate the following SPA messages in parallel



$$\phi^{[\ell]}_{\Psi_{im}\to a_i}(a_i) = \sum_{b_m=0}^{I} \Psi_{im}(a_i,b_m) \prod_{\substack{i'=1\\ i'\neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm}\to b_m}(b_m)$$

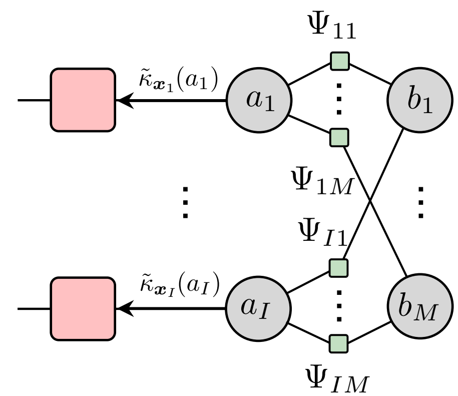
$$\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\,\Psi_{im}(a_i,b_m) \prod_{\substack{m'=1\\ m'\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'}\to a_i}(a_i)$$

- Initialization at $\ell = 0$: $\phi^{[0]}_{\Psi_{im}\to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\,\Psi_{im}(a_i,b_m)$

Loopy SPA for Joint Probabilistic Data Association

- Stretching the graph enables calculation of approximate $\tilde{\kappa}_{\boldsymbol{x}_{i,n}}(a_i)$ by means of the loopy SPA

- At message passing iteration $\ell \in \{1, \ldots, L\}$ we calculate the following SPA messages in parallel $(i \in \{1, \ldots, I\}, m \in \{1, \ldots, M\})$



$$\phi^{[\ell]}_{\Psi_{im} \to a_i}(a_i) = \sum_{b_m=0}^{I} \Psi_{im}(a_i, b_m) \prod_{\substack{i'=1 \\ i' \neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm} \to b_m}(b_m)$$

$$\phi^{[\ell]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\, \Psi_{im}(a_i, b_m) \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'} \to a_i}(a_i)$$

- Initialization at $\ell = 0$: $\phi^{[0]}_{\Psi_{im} \to b_m}(b_m) = \sum_{a_i=0}^{M} \phi_{a_i}(a_i)\, \Psi_{im}(a_i, b_m)$

- Result after $\ell = L$ iterations: $\tilde{\kappa}_{\boldsymbol{x}_i}(a_i) = \prod_{m=1}^{M} \phi^{[L]}_{\Psi_{im} \to a_i}(a_i)$

The Sum-Product Algorithm for Data Association (SPADA)

- The complexity of the probabilistic assignment algorithm can be reduced further by performing the following steps

  1. Since the constraint $\Psi_{im}(a_i, b_m)$ is binary, messages can be represented by only two different values

$$\phi^{[\ell]}_{\Psi_{im} \to a_i}(a_i) = \begin{cases} \prod_{\substack{i'=1 \\ i' \neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm} \to b_m}(b_m = i), & \text{for } a_i = m \\ \sum_{\substack{b_m=0 \\ b_m \neq i}}^{I} \prod_{\substack{i''=1 \\ i'' \neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i''m} \to b_m}(b_m), & \text{for } a_i \neq m \end{cases}$$

$$\phi^{[\ell]}_{\Psi_{im} \to b_m}(b_m) = \begin{cases} \phi_{a_i}(m) \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'} \to a_i}(m), & \text{for } b_m = i \\ \sum_{\substack{a_i=0 \\ a_i \neq m}}^{M} \phi_{a_i}(a_i) \prod_{\substack{m''=1 \\ m'' \neq m}}^{M} \phi^{[\ell]}_{\Psi_{im''} \to a_i}(a_i), & \text{for } b_m \neq i \end{cases}$$

The Sum-Product Algorithm for Data Association (SPADA)

- The complexity of the probabilistic assignment algorithm can be reduced further by performing the following steps

   2. Since messages can be multiplied by an arbitrary constant, we divide each message by one of its values

$$\phi^{[\ell]}_{\Psi_{im}\to a_i}(a_i) \propto \begin{cases} \dfrac{\prod_{\substack{i'=1\\ i'\neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm}\to b_m}(i)}{\sum_{\substack{b_m=0\\ b_m\neq i}}^{I} \prod_{\substack{i''=1\\ i''\neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i''m}\to b_m}(b_m)}, & \text{for } a_i = m \\ 1, & \text{for } a_i \neq m \end{cases}$$

$$\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m) \propto \begin{cases} \dfrac{\phi_{a_i}(m) \prod_{\substack{m'=1\\ m'\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im'}\to a_i}(m)}{\sum_{\substack{a_i=0\\ a_i\neq m}}^{M} \phi_{a_i}(a_i) \prod_{\substack{m''=1\\ m''\neq m}}^{M} \phi^{[\ell]}_{\Psi_{im''}\to a_i}(a_i)}, & \text{for } b_m = i \\ 1, & \text{for } b_m \neq i \end{cases}$$

The Sum-Product Algorithm for Data Association (SPADA)

- The complexity of the probabilistic assignment algorithm can be reduced further by performing the following steps

3. Messages can now be replaced by there normalized counterpart

$$\phi^{[\ell]}_{\Psi_{im}\to a_i}(a_i) \propto \begin{cases} \frac{1}{1+\sum_{\substack{i'=1\\ i'\neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm}\to b_m}(i')}, & \text{for } a_i = m \\ 1, & \text{for } a_i \neq m \end{cases}$$

$$\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m) \propto \begin{cases} \frac{\phi_{a_i}(m)}{\phi_{a_i}(0)+\sum_{\substack{m'=1\\ m'\neq m}}^{M} \phi_{a_i}(m')\,\phi^{[\ell]}_{\Psi_{im'}\to a_i}(m')}, & \text{for } b_m = i \\ 1, & \text{for } b_m \neq i \end{cases}$$

The Sum-Product Algorithm for Data Association (SPADA)

- The complexity of the probabilistic assignment algorithm can be reduced further by performing the following steps

4. Each message can be represented by a single value $\big(i \in \{1, \ldots, I\}, m \in \{1, \ldots, M\}\big)$

$$\phi^{[\ell]}_{\Psi_{im} \to a_i} = \frac{1}{1 + \sum_{\substack{i'=1 \\ i' \neq i}}^{I} \phi^{[\ell-1]}_{\Psi_{i'm} \to b_m}(i')} \qquad \phi^{[\ell]}_{\Psi_{im} \to b_m} = \frac{\phi_{a_i}(m)}{\phi_{a_i}(0) + \sum_{\substack{m'=1 \\ m' \neq m}}^{M} \phi_{a_i}(m')\, \phi^{[\ell]}_{\Psi_{im'} \to a_i}(m')}$$

Initialization: $$\phi^{[0]}_{\Psi_{im} \to b_m} = \frac{\phi_{a_i}(m)}{\phi_{a_i}(0) + \sum_{\substack{m'=1 \\ m' \neq m}}^{M} \phi_{a_i}(m')}$$

Result after L iterations:

$$\tilde{\kappa}_{\boldsymbol{x}_i}(a_i) = \begin{cases} \phi^{[L]}_{\Psi_{im} \to a_i}(m), & \text{for } a_i = m \in \{1, \ldots, M\} \\ 1, & \text{for } a_i = 0 \end{cases}$$

Properties

- The complexity of the probabilistic assignment algorithm is essentially determined by that of calculating the sums $\sum_{i=1}^{I} \phi^{[\ell]}_{\Psi_{im} \to b_m}(i)$ and $\phi^{[\ell]}_{\Psi_{im} \to b_m} = \sum_{m=1}^{M} \phi_{a_i}(m) \phi^{[\ell]}_{\Psi_{im} \to a_i}(m)$, which scales as $\mathcal{O}(IM)$
- It can be shown that the loopy SPA algorithms for joint probabilistic data association
  – solves a convex optimization problem
  – is guaranteed to converge
  – provides the correct MAP solution

M. Bayati, D. Shah, and M. Sharma, "Max-product for maximum weight matching: Convergence, correctness, and LP duality," *IEEE Trans. Inf. Theory*, no. 3, pp. 1241–1251, Mar. 2008.

J. L. Williams and R. A. Lau, "Multiple scan data association by convex variational inference," *IEEE Trans. Signal Process.*, vol. 66, no. 8, pp. 2112–2127, Apr. 2018.

F. Meyer, T. Kropfreiter, J. L. Williams, R. A. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, Feb. 2018.

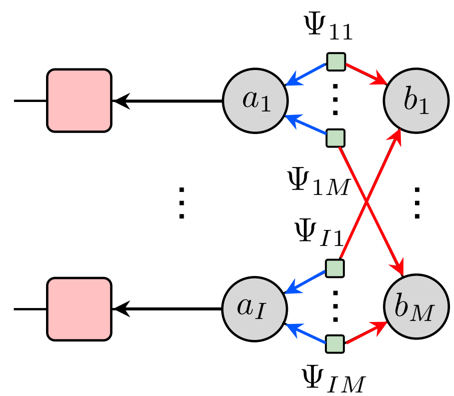
Joint Probabilistic Data Association

- The sum-product message passing rules are applied to the stretched factor graph we obtain $\phi^{[\ell]}_{\Psi_{i,m}\to a_i}(a_i)$ and $\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m)$ for the ℓth iteration

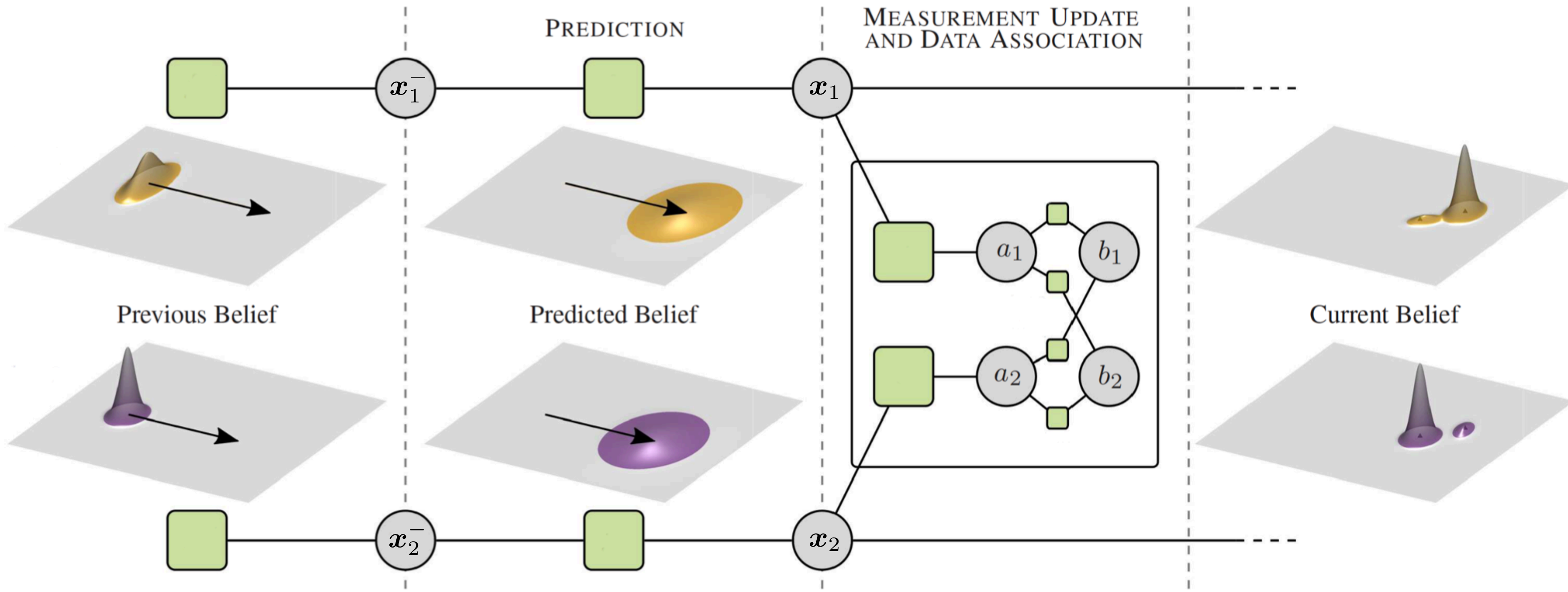
- Due to the binary consistency constrains, $\phi^{[\ell]}_{\Psi_{im}\to b_m}(b_m)$ takes only two values (one for $b_m = i$ and one for $b_m \neq i$); similarly $\phi^{[\ell]}_{\Psi_{i,m}\to a_i}(a_i)$ takes one value for $a_i = m$ and one value for $a_i \neq m$

- Can be implemented by performing pointwise operations on $I \times M$ matrices

- All $\tilde{\kappa}_{\boldsymbol{x}_i}(a_i)$ needed for multiobject tracking can be obtained with a complexity that only scales as $\mathcal{O}(IM)$



Multiobject Tracking Example



D. Gaglione, G. Soldi, F. Meyer, F. Hlawatsch, P. Braca, A. Farina, and M. Z. Win, *Bayesian information fusion and multitarget tracking for maritime situational awareness*, IET Radar Sonar Navi., Nov. 2020.

Hard Measurement Validation

- To further reduce computational complexity of multiobject tacking, measurements that with a high probability have not been generate by an object, can be removed in a suboptimum preprocessing step

- For each object a multidimensional gate is set up and only measurements that fall within the gate are considered association candidates

- Joint probabilistic data association has only to be performed for objects that share association candidates; thus its complexity is $\mathcal{O}(I'M')$ with $I' \leqslant I$ and $M' \leqslant M$

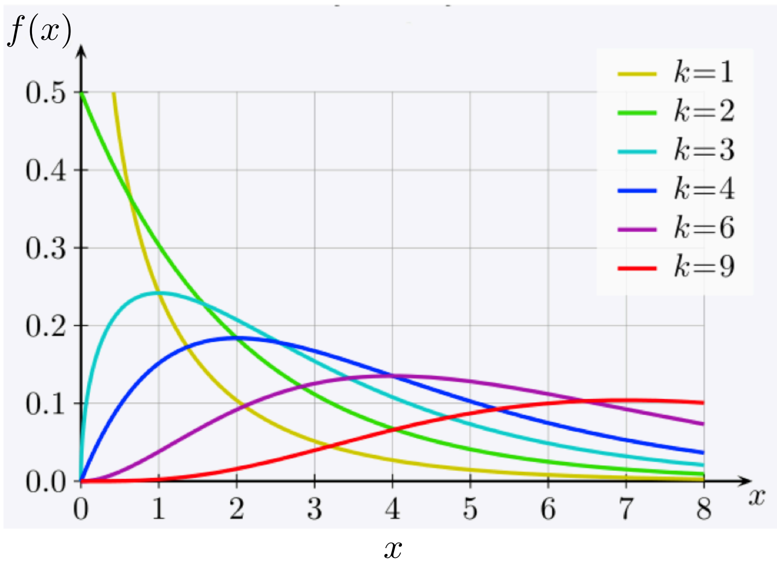
The Chi-Square Distribution

- The chi-square distribution with k degrees of freedom is the distribution of a sum of the squares of k independent normal random with unit variance

$$f(x) = \begin{cases} \frac{x^{\frac{k}{2}-1}\exp(-\frac{x}{2})}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})}, & x > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\Gamma(n) = (n-1)! \qquad \text{for } n \in \mathbb{N}_0$$

Mean: k
Variance: $2k$



Hard Measurement Validation

- **Assumption:** The measurement that is originated by object i at time n is distributed according to

$$
\begin{aligned}
f(\boldsymbol{z}_{m,n}|\boldsymbol{z}_{1:n-1}) &= f_{\mathrm{g}}(\boldsymbol{z}_{m,n}; \boldsymbol{H}_n \boldsymbol{\mu}^{-}_{\boldsymbol{x}_{i,n}}, \boldsymbol{H}_n \boldsymbol{\Sigma}^{-}_{\boldsymbol{x}_{i,n}} \boldsymbol{H}_n^{\mathrm{T}} + \boldsymbol{\Sigma}_{\boldsymbol{v}_n}) \\
&= f_{\mathrm{g}}(\boldsymbol{z}_{m,n}; \boldsymbol{\mu}^{-}_{\boldsymbol{z}_{i,n}}, \boldsymbol{\Sigma}^{-}_{\boldsymbol{z}_{i,n}})
\end{aligned}
$$

- The true measurement will be in the following set

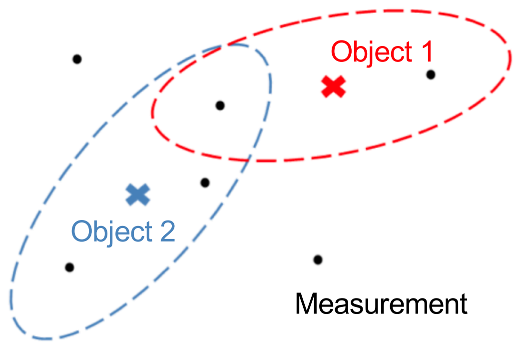
$$
\mathcal{V}_{i,n}(\gamma) = \left\{ \boldsymbol{z}_{m,n} \middle| (\boldsymbol{z}_{m,n} - \boldsymbol{\mu}_{\boldsymbol{z}_{i,n}})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{z}_{i,n}} (\boldsymbol{z}_{m,n} - \boldsymbol{\mu}_{\boldsymbol{z}_{i,n}}) \leqslant \gamma \right\}
$$

  with probability determined by the threshold γ

- The region that contains validated measurements is an ellipsoid with semiaxes given by the square roots of the eigenvalues of $\gamma \boldsymbol{\Sigma}_{\boldsymbol{z}_{i,n}}$

Hard Measurement Validation

- The quadratic form that defines the validation region is chi-square distributed with the number of degrees of freedom equal to the dimension of a measurement d
- Thus, the probability p_{g} that a measurement lies in the validation region or ``gate'' can be obtain from the cumulative distribution function of the chi-square distribution, i.e., $p_{\mathrm{g}} = \mathtt{chi2cdf}(\gamma, d)$
- Hard measurement validation trades of computational complexity and sensor performance since p_{d} is reduced to $p'_{\mathrm{d}} = p_{\mathrm{g}} p_{\mathrm{d}}$
- Example with two objects →



Summary

- Computational complexity of joint probabilistic data association can be reduced from $\mathcal{O}\big((M_n+1)^I\big)$ to $\mathcal{O}(IM_n)$ by performing a highly optimized loopy sum-product algorithm

- Hard measurement validation (``gating'') can further reduce computational complexity by extracting association candidates from the joint measurement vector and thus reducing the dimension of the data association problem

Graph-Based Multiobject Tracking

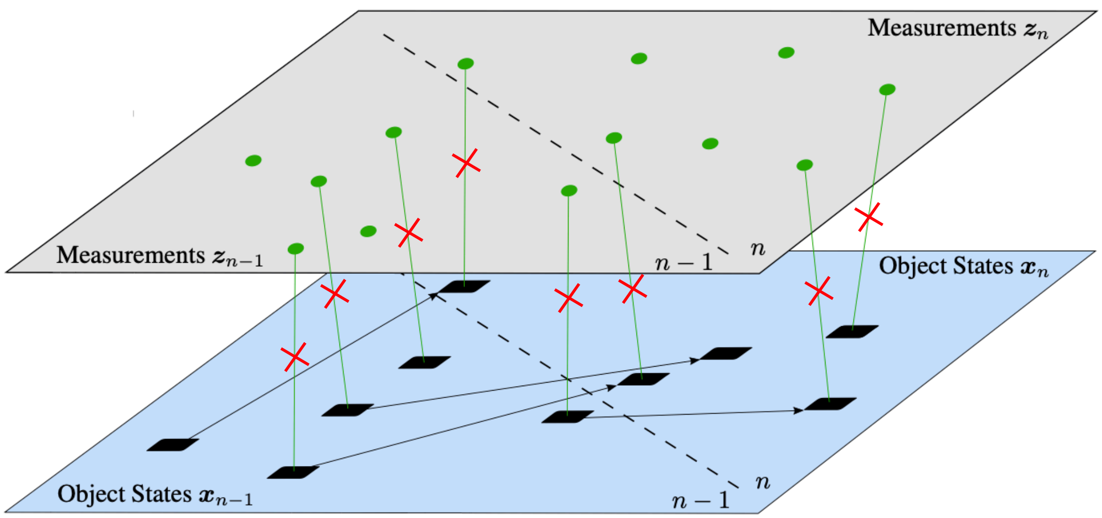
Part 6: Track Management and Partitioning of Measurements

Florian Meyer
joint work with Jason Williams, Paolo Braca, Peter Willett, and Franz Hlawatsch

Scripps Institution of Oceanography
Electrical and Computer Engineering Department
University of California San Diego

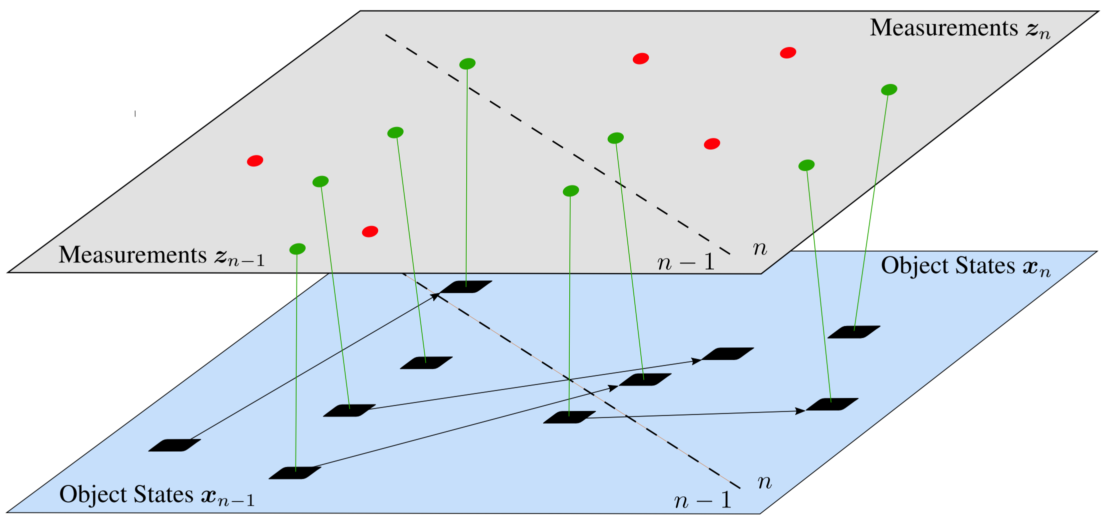
The Multiobject Tracking Problem

- At each time n: localize and track an unknown number of objects $\boldsymbol{x}_n = [\boldsymbol{x}_{n,1}^{\mathrm{T}} \ldots \boldsymbol{x}_{n,I_n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{n,1}^{\mathrm{T}} \ldots \boldsymbol{z}_{n,M_n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin



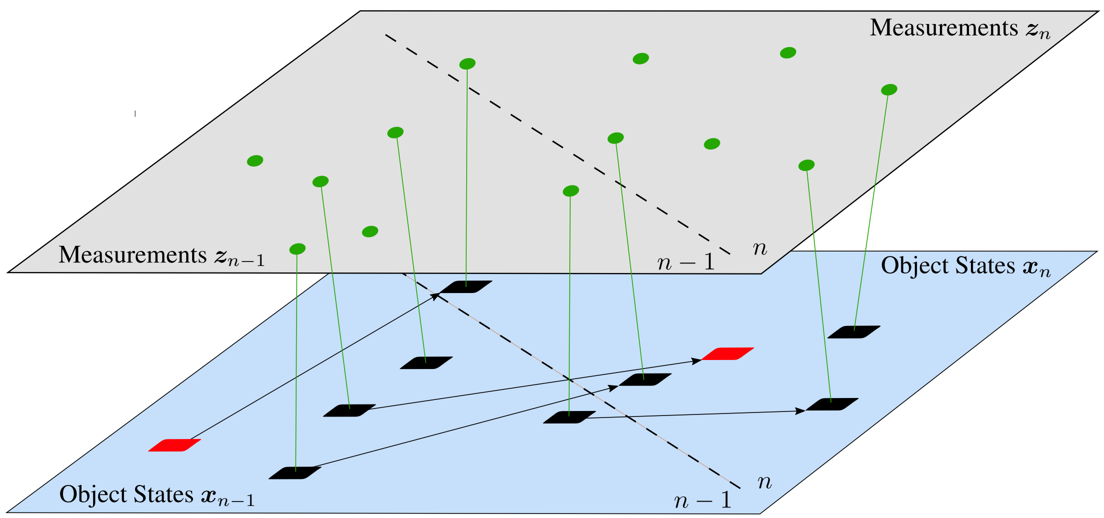
The Multiobject Tracking Problem

- At each time n: localize and track an unknown number of objects $\boldsymbol{x}_n = [\boldsymbol{x}_{n,1}^{\mathrm{T}} \ldots \boldsymbol{x}_{n,I_n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{n,1}^{\mathrm{T}} \ldots \boldsymbol{z}_{n,M_n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin
- Data association is challenging because of clutter measurements, missing measurements, object births, and object deaths



The Multiobject Tracking Problem

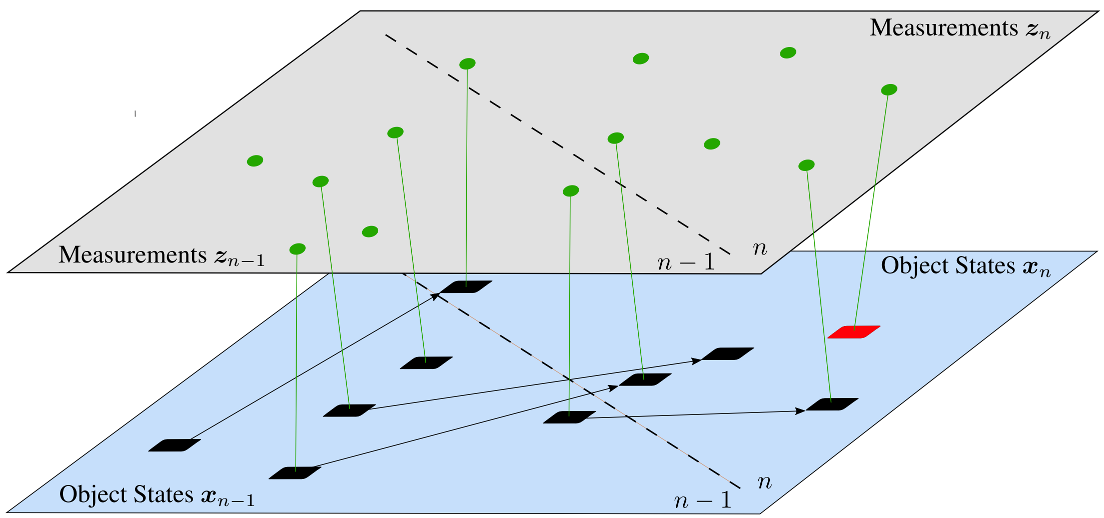
- At each time n: localize and track an unknown number of objects $\boldsymbol{x}_n = [\boldsymbol{x}_{n,1}^{\mathrm{T}} \ldots \boldsymbol{x}_{n,I_n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{n,1}^{\mathrm{T}} \ldots \boldsymbol{z}_{n,M_n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin
- Data association is challenging because of clutter measurements, missing measurements, object births, and object deaths



The Multiobject Tracking Problem

- At each time n: localize and track an unknown number of objects $\boldsymbol{x}_n = [\boldsymbol{x}_{n,1}^{\mathrm{T}} \dots \boldsymbol{x}_{n,I_n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{n,1}^{\mathrm{T}} \dots \boldsymbol{z}_{n,M_n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin

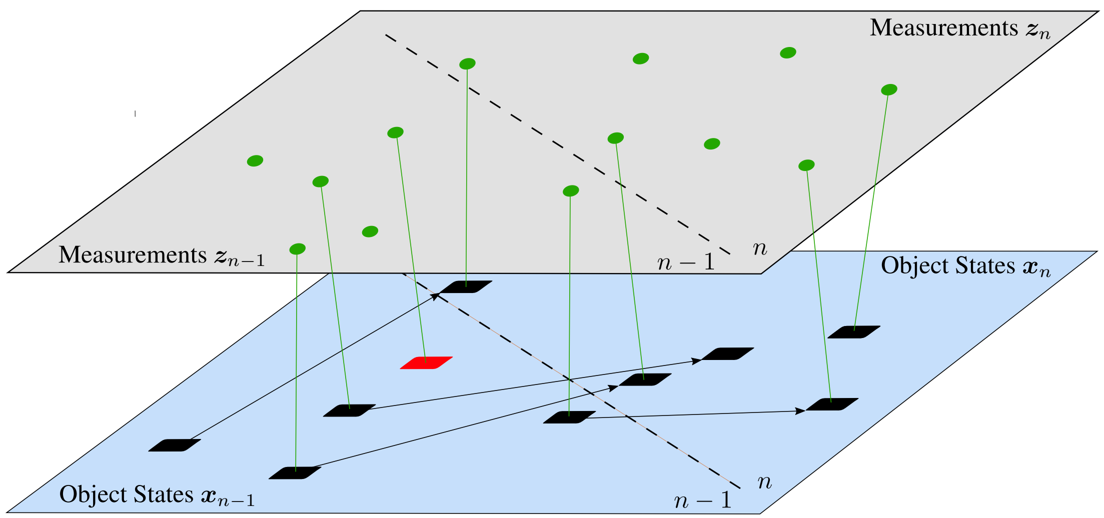
- Data association is challenging because of clutter measurements, missing measurements, object births, and object deaths



The Multiobject Tracking Problem

- At each time n: localize and track an unknown number of objects $\boldsymbol{x}_n = [\boldsymbol{x}_{n,1}^{\mathrm{T}} \ldots \boldsymbol{x}_{n,I_n}^{\mathrm{T}}]^{\mathrm{T}}$ from measurements $\boldsymbol{z}_n = [\boldsymbol{z}_{n,1}^{\mathrm{T}} \ldots \boldsymbol{z}_{n,M_n}^{\mathrm{T}}]^{\mathrm{T}}$ with uncertain origin

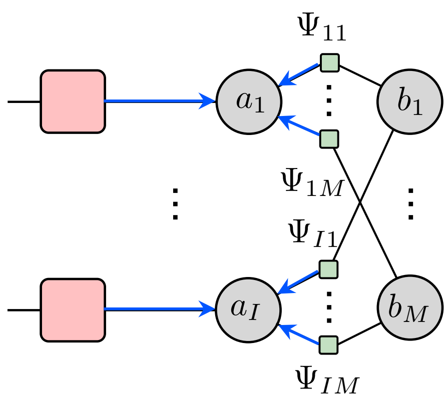
- Data association is challenging because of clutter measurements, missing measurements, object births, and object deaths



Association Probabilities

- Approximate object-oriented marginal association probabilities after $\ell = L$ iterations

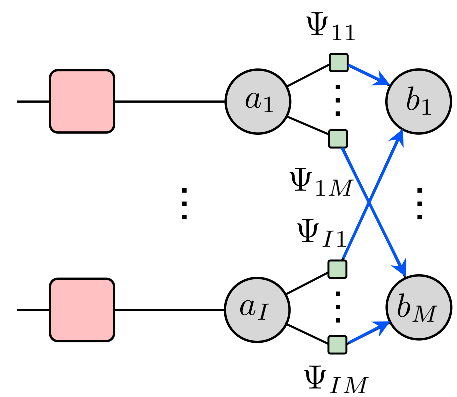
$$\tilde{p}(a_i|\boldsymbol{z}) \propto \phi_{a_i}(a_i) \prod_{m=1}^{M} \phi^{[L]}_{\Psi_{im}\rightarrow a_i}(a_i)$$



Association Probabilities

- Approximate object-oriented marginal association probabilities after $\ell = L$ iterations

$$\tilde{p}(a_i|\boldsymbol{z}) \propto \phi_{a_i}(a_i) \prod_{m=1}^{M} \phi^{[L]}_{\Psi_{im} \rightarrow a_i}(a_i)$$



- Approximate measurement-oriented marginal association probabilities after $\ell = L$ iterations

$$\tilde{p}(b_m|\boldsymbol{z}) \propto \prod_{i=1}^{I} \phi^{[L]}_{\Psi_{im} \rightarrow b_m}(b_m)$$

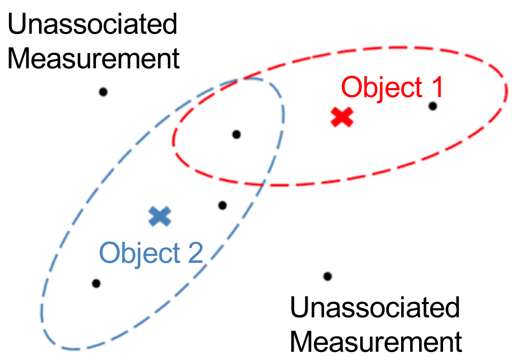
- Note that $\tilde{p}(a_i = 0|\boldsymbol{z})$ is the probability that object i did not generate a measurement and $\tilde{p}(b_m = 0|\boldsymbol{z})$ is the probability that measurement m was not generated by an object

$\longrightarrow$ potentially useful for generating or terminating tracks

Unassociated Measurements

- Unassociated measurements are measurements that with high probability have not been originated by an object

- Can be determined by
  - hard measurement evaluation: measurements that are outside the gates of all the objects are declared unassociated
  - joint probabilistic data association: all measurements with $\tilde{p}(b_m = 0|\boldsymbol{z})$ larger than a certain threshold are declared unassociated



Track Formation and Termination Heuristics

- A heuristic to initialize a new track for a newborn object is referred to as track formation in clutter
- The logic-based approach uses gates to search for sequence of measurements that are not associated to any existing object
- If a requirement is satisfied, then the measurement sequence is accepted as a valid track and initialized by increasing the state space and extracting a prior distribution from the sequence of measurements
- A track is terminated if for a number of time steps N no measurement is associated to it

K/N Formation Heuristic

1. Every unassociated measurement is an ``initiator'' -- it yields a tentative track

2. At the time step following the detection of an initiator, a gate is set up based on the
   – assumed maximum and minimum object motion parameters
   – the measurement noise variances

   such that, if there is a target that gave rise to the initiator, the measurement from it in this second time step (if detected) will fall in the gate with nearly unity probability

3. If there is a measurement, this tentative track becomes a preliminary track. If there is no measurements, this track is dropped

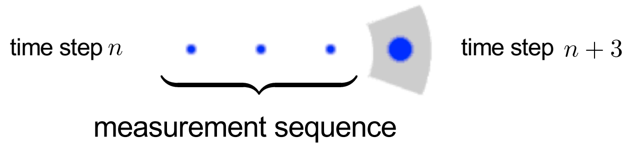
4. Since a preliminary track has two measurements, a sequential Bayesian estimation can be initialized and used to set up a gate for the next (third) time step

K/N Formation Heuristic

5. Starting from the third scan a logic of K detections out of N time steps is used for subsequent gates

6. If at the end (scan $N+2$ at the latest) the logic requirement is satisfied, the track becomes a confirmed track; otherwise it is dropped

- The requirement of two initial detections reduces the probability of false tracks
- Typical values for K/N: 3/5, 4/6, …
- **Advantages:** Easy to implement
- **Disadvantages:** Heuristic, performance analysis difficult

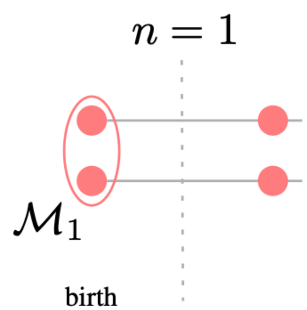


Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. YBS, 2011.

Bayesian Initialization of New Tracks

- Consider time n and *potential object states* $\boldsymbol{y}_{i,n} = [\boldsymbol{x}_{i,n}^{\mathrm{T}} \; r_{i,n}]^{\mathrm{T}}, \; i \in \mathcal{I}_n$ where existence is modeled by a Bernoulli variable $r_{i,n} \in \{0, 1\}$

- **Potential object birth:** For each measurement $z_{m,n}, m \in \mathcal{M}_n$ introduce a new state

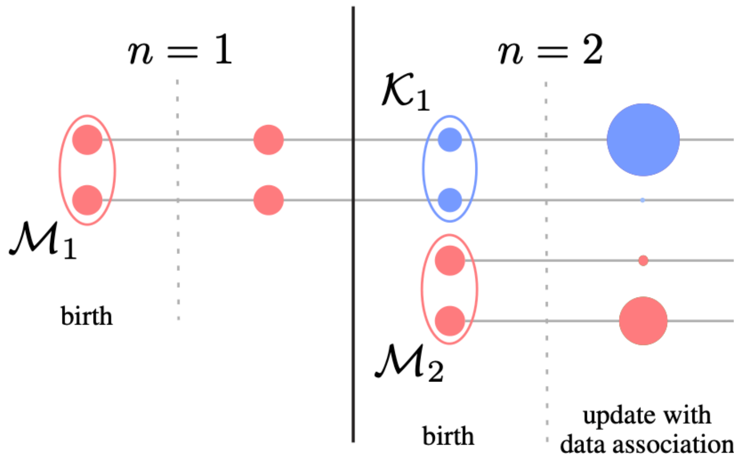


Bayesian Initialization of New Tracks

- Consider time n and *potential object states* $\boldsymbol{y}_{i,n} = [\boldsymbol{x}_{i,n}^{\mathrm{T}}\, r_{i,n}]^{\mathrm{T}},\ i \in \mathcal{I}_n$ where existence is modeled by a Bernoulli variable $r_{i,n} \in \{0, 1\}$

- **Potential object birth:** For each measurement $\boldsymbol{z}_{m,n}, m \in \mathcal{M}_n$ introduce a new state

- **Potential object death:** Remove states i with low probability of existence $p(r_{i,n} = 1 | \boldsymbol{z}_{1:n})$

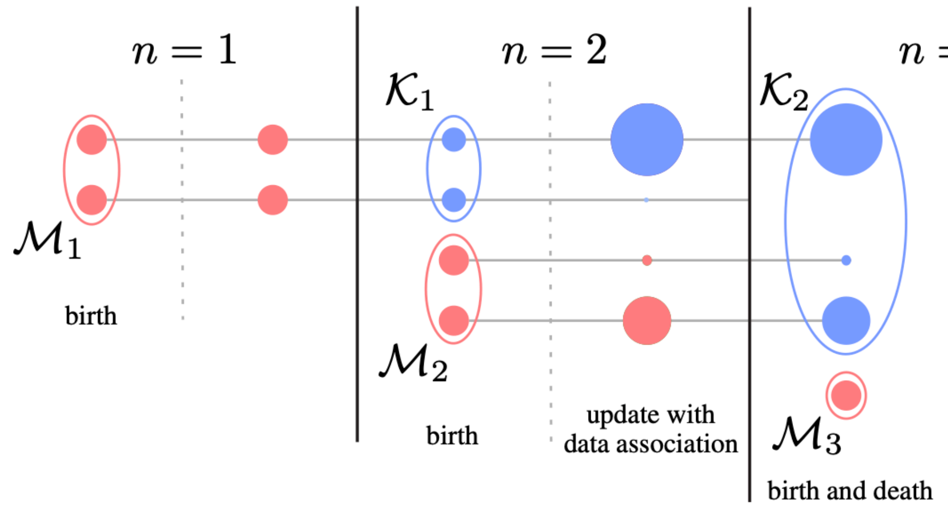


Bayesian Initialization of New Tracks

- Consider time n and *potential object states* $\boldsymbol{y}_{i,n} = [\boldsymbol{x}_{i,n}^{\mathrm{T}}\; r_{i,n}]^{\mathrm{T}},\; i \in \mathcal{I}_n$ where existence is modeled by a Bernoulli variable $r_{i,n} \in \{0, 1\}$

- **Potential object birth:** For each measurement $\boldsymbol{z}_{m,n}, m \in \mathcal{M}_n$ introduce a new state

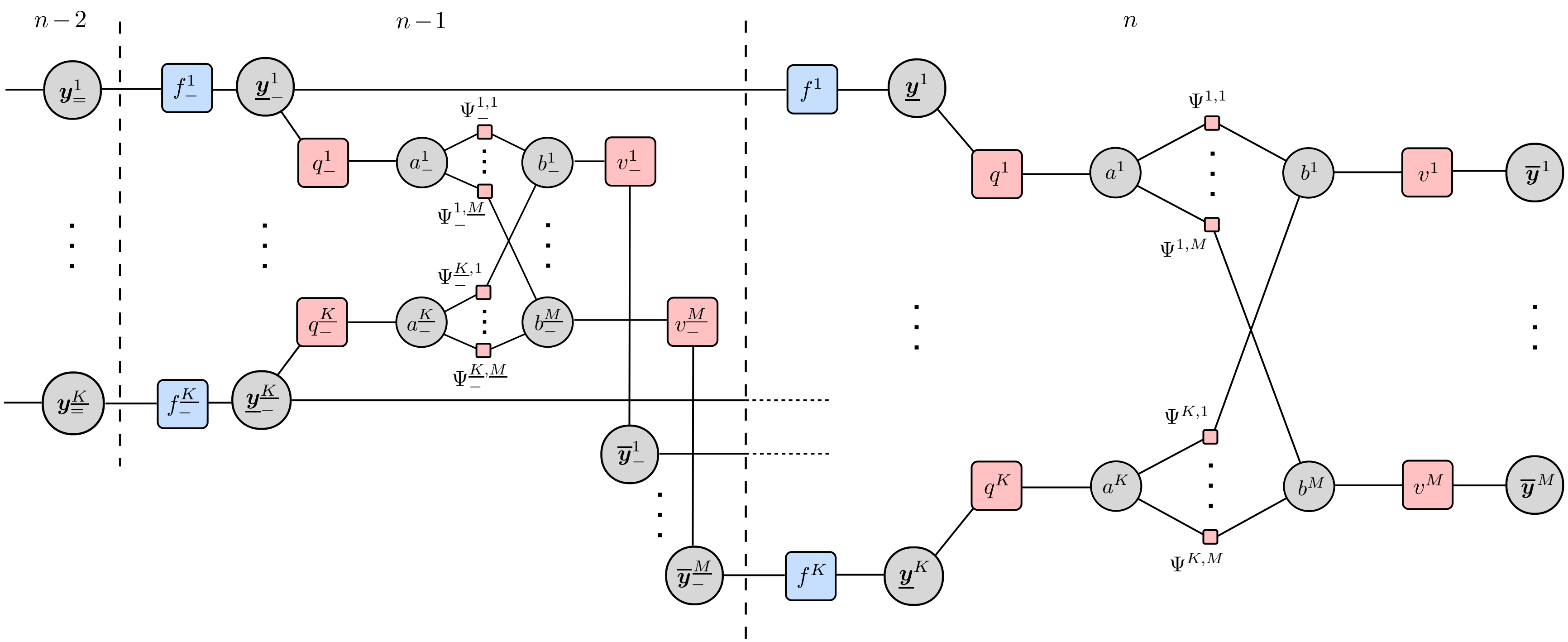
- **Potential object death:** Remove states i with low probability of existence $p(r_{i,n} = 1 | \boldsymbol{z}_{1:n})$



- **Localizing an unknown number of objects:**
  – determine existence of object i by comparing $p(r_{i,n} = 1 | \boldsymbol{z}_{1:n})$ to threshold, e.g., $P_{\mathrm{th}} = 0.5$
  – estimate the states $\boldsymbol{x}_{i,n}$ of existing objects by using, e.g., the MMSE estimator

Multiobject Tracking – Factor Graph

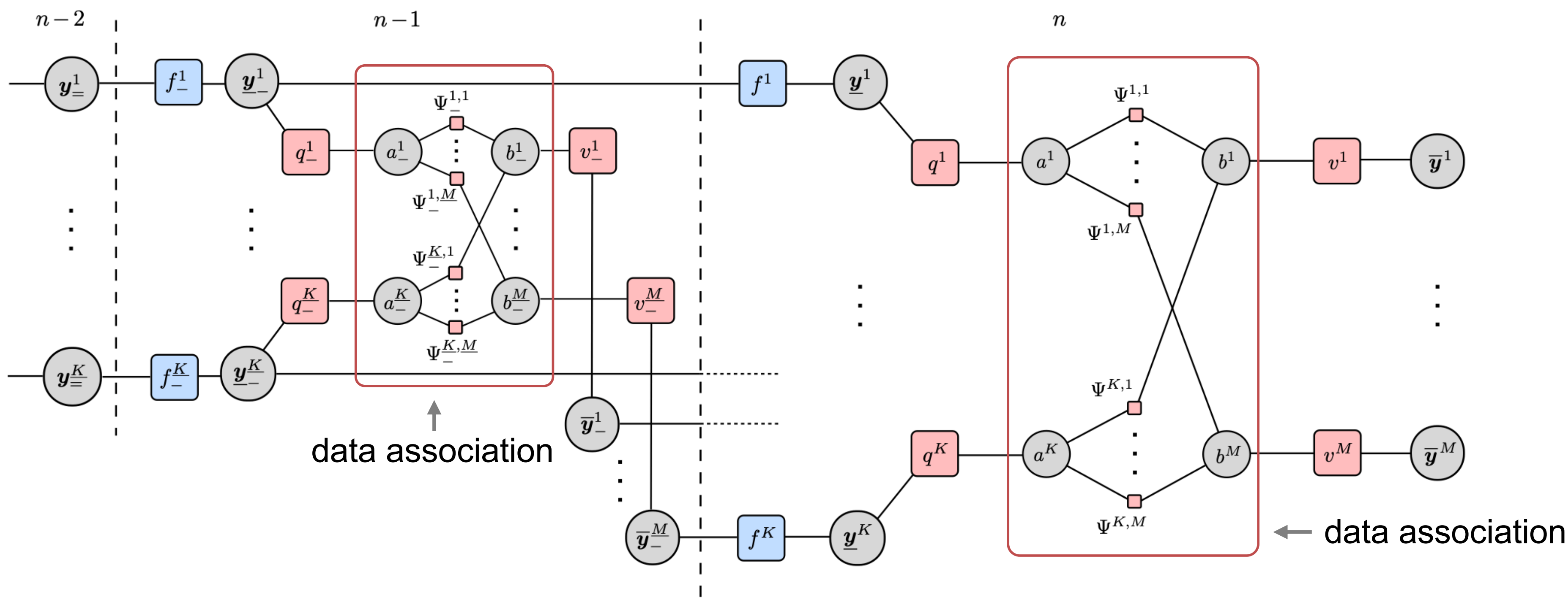
- Complete graph for a sequence of measurements:



F. Meyer, T. Kropfreiter, J. L. Williams, R. A. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, Feb. 2018.

Multiobject Tracking – Factor Graph

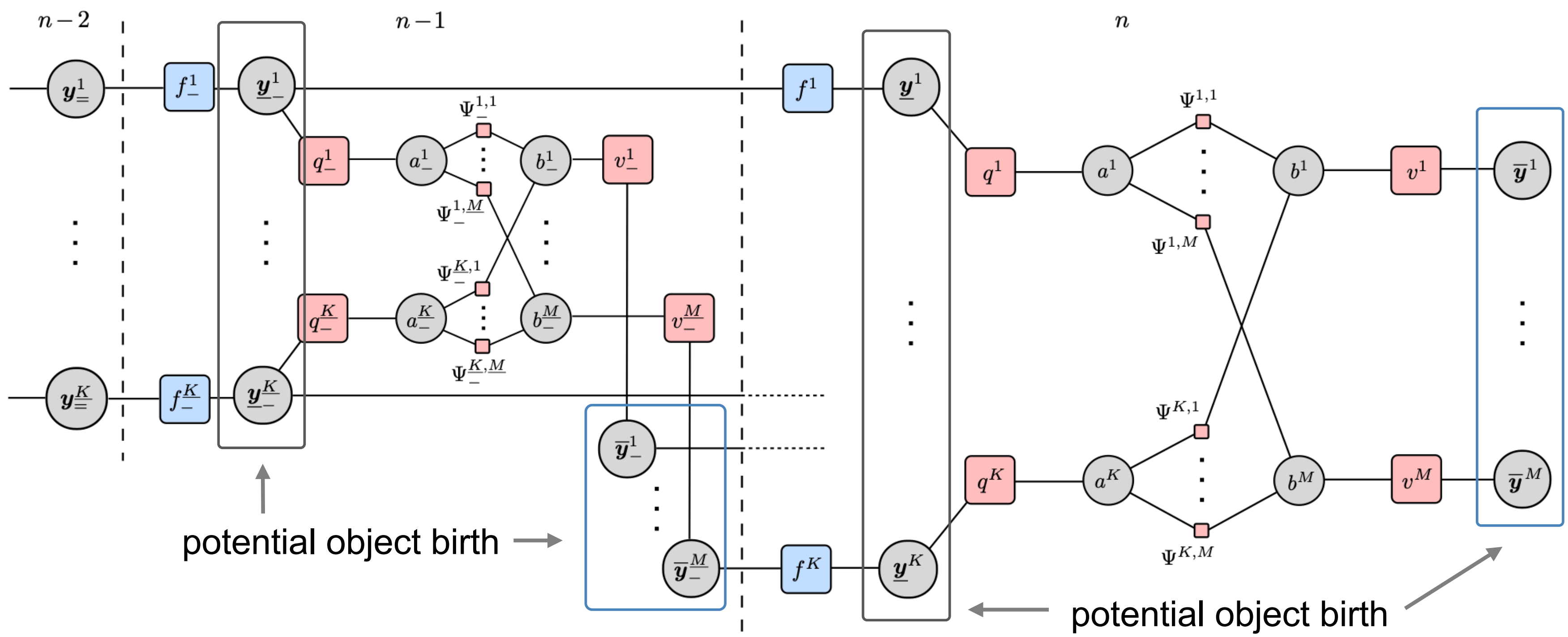
- Complete graph for a sequence of measurements:



F. Meyer, T. Kropfreiter, J. L. Williams, R. A. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, “Message passing algorithms for scalable multitarget tracking,” *Proc. IEEE*, Feb. 2018.

Multiobject Tracking – Factor Graph

- Complete graph for a sequence of measurements:



F. Meyer, T. Kropfreiter, J. L. Williams, R. A. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, Feb. 2018.

Distance Partitioning

- Recall data association assumptions: An (i) object can generate at most one measurement and a (ii) measurement can be generated by at most one object

- For high-resolution sensors (e.g., LIDAR), (i) is typically not satisfied

- Statistical model for case where (i) is not satisfied results in very challenging data association problem $\longrightarrow$ extended object tracking

- Heuristic preprocessing stage that aims to enforce (i):
  - Partition the set of measurements $\mathcal{Z} = \{\boldsymbol{z}_m \,|\, m \in \{1, \ldots, M\}\}$ into disjoint subsets (cells) $\mathcal{Z}^{(c)}, \; c \in 1, \ldots, C$, where each subset contains spatially close measurements that are likely to be generated by the same object $(C \leqslant M)$
  - Use "hyper measurements" $\boldsymbol{z}^{(c)}$ related to cells $\mathcal{Z}^{(c)}$ as measurements for multiobject tracking

Distance Partitioning

- Let us assume $d(\cdot,\cdot)$ is a distance measure and Δ_{m_1,m_2} is the distance of measurement pair $\boldsymbol{z}_{m_1}$ and $\boldsymbol{z}_{m_2}$
- The set the measurements $\mathcal{Z} = \{\boldsymbol{z}_m \big| m \in \{1,\dots,M\}\}$ can be partitioned into disjoint subsets (cells) based on the following theorem
- **Theorem:** A distance threshold d_ℓ defines a unique partition of that leaves all pairs of measurements (m_1, m_2) satisfying $\Delta_{m_1,m_2} < d_\ell$ in the same cell (see references for detailed version of theorem)
- If the measurements noise is additive Gaussian, the Mahalanobis distance can be used

$$d(\boldsymbol{z}_{m_1}, \boldsymbol{z}_{m_2}) = \sqrt{(\boldsymbol{z}_{m_1} - \boldsymbol{z}_{m_2})^{\mathrm{T}} \boxed{\boldsymbol{\Sigma}_{\boldsymbol{v}}^{-1}} (\boldsymbol{z}_{m_1} - \boldsymbol{z}_{m_2})}$$

Measurement noise covariance matrix

K. Granström, C. Lundquist, and O. Orguner, "Extended target tracking using a Gaussian-mixture PHD filter," *IEEE Trans. Aerosp. Electron. Syst.,* Oct. 2012.

K. Granström, O. Orguner, R. Mahler, and C. Lundquist, Corrections on: "Extended target tracking using a Gaussian-mixture PHD filter," *IEEE Trans. Aerosp. Electron. Syst.,* Apr. 2017.

Distance Partitioning

- Distance threshold: d_ℓ
- Number of measurements: $N_{\boldsymbol{z}}$
- Distance between measurement $\boldsymbol{z}_i$ and measurement $\boldsymbol{z}_j$: $\Delta_{i,j}$

Distance Partitioning

Require: d_ℓ, $\Delta_{i,j}$, $1 \le i \ne j \le N_z$.
CellNumber(i) = 0, $1 \le i \le N_z$ {Set cells of all measurements to null}
CellId = 1 {Set the current cell id to 1}
%Find all cell numbers
for $i = 1 : N_z$ **do**
  if CellNumbers(i) = 0 **then**
    CellNumbers(i) = CellId
    CellNumbers = FindNeighbors(i,CellNumbers,CellId)
    CellId = CellId+1
  end if
end for

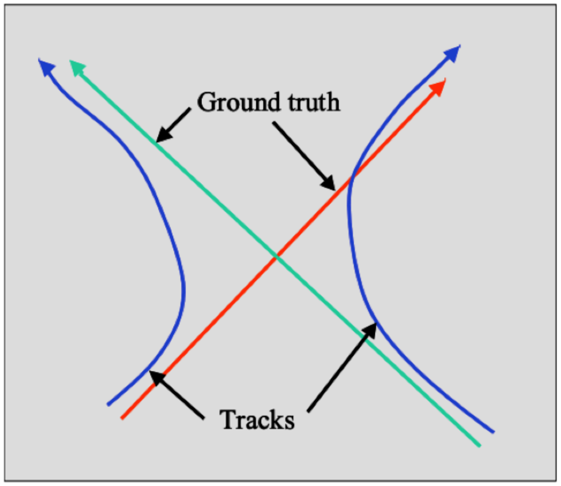
The recursive function FindNeighbors($\cdot,\cdot,\cdot$) is given as

function CellNumbers = FindNeighbors(i,CellNumbers,CellId)
for $j = 1 : N_z$ **do**
  if $j \ne i$ & $\Delta_{ij} \le d_\ell$ & CellNumbers(j) = 0 **then**
    CellNumbers(j) = CellId
    CellNumbers = FindNeigbors(j,CellNumbers,CellId)
  end if
end for

Optimal Subpattern Assignment Metric

- Mean square error is not a suitable metric for many multiobject tracking applications
  - not defined if estimated number of objects is different than the true number of objects
  - track swapping leads to large errors

- Parameters
  - Metric order p
  - Cutoff parameter η
  - Inner metric $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$



D. Schuhmacher, B.-T. Vo, B.-N. Vo, "A Consistent Metric for Performance Evaluation of Multi-Object Filters," *IEEE Trans. Signal Process., Jul*. 2008.

Optimal Subpattern Assignment Metric

- Let $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_I]^{\mathrm{T}}$ be the true joint object state vector and $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_{\hat{I}}]^{\mathrm{T}}$ be the estimated joint object state vector

- Simple version for the case $I = \hat{I}$, $p = 2$, $\eta = \infty$, and $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$

$$d_2^{(\infty)} = \frac{1}{I}\Big(\min_{\pi \in \Pi_I} \sum_{i=1}^{I} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_{\pi(i)}\|^2 \Big)^{1/2}$$

  – Π_I is the set of all permutations of $[1, 2, \ldots, I]^{\mathrm{T}}$

Optimal Subpattern Assignment Metric

- General version for $\hat{I} \leqslant I$

$$d_p^{(\eta)}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{I}\Big(\min_{\pi \in \Pi_I} \sum_{i=1}^{\hat{I}} d^{(\eta)}(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_{\pi(i)})^p + \eta^p (I - \hat{I}) \Big)^{1/p}$$

where $d^{(\eta)}(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_j) = \boxed{\min(\eta, d(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_j))}$

Individual object state errors are cutoff at η

Optimal Subpattern Assignment Metric

- General version for $\hat{I} \leqslant I$

$$d_p^{(\eta)}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{I}\Big(\min_{\pi \in \Pi_I} \sum_{i=1}^{\hat{I}} d^{(\eta)}(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_{\pi(i)})^p + \boxed{\eta^p (I - \hat{I})}\Big)^{1/p}$$

Penalty for dimension mismatch

where $d^{(\eta)}(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_j) = \min(\eta, d(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_j))$

Optimal Subpattern Assignment Metric

- General version for $\hat{I} \leqslant I$

$$d_p^{(\eta)}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{I}\bigg(\min_{\pi \in \Pi_I} \sum_{i=1}^{\hat{I}} d^{(\eta)}(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_{\pi(i)})^p + \eta^p (I - \hat{I}) \bigg)^{1/p}$$

where $d^{(\eta)}(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_j) = \min(\eta, d(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_j))$

- General version for $\hat{I} > I$

$$d_p^{(\eta)}(\hat{\boldsymbol{x}}, \boldsymbol{x}) = \frac{1}{\hat{I}}\bigg(\min_{\pi \in \Pi_{\hat{I}}} \sum_{i=1}^{I} d^{(\eta)}(\hat{\boldsymbol{x}}_i, \boldsymbol{x}_{\pi(i)})^p + \eta^p (I - \hat{I}) \bigg)^{1/p}$$

Summary

- Marginal association probabilities and gating are useful to introduce and remove objects states from the state space (initiate and terminate tracks)
- Distance partitioning can be used to as a preprocessing stage to ``enforce'' the property that each object just produces one measurement
- The very general optimal subpattern assignment metric makes it possible to quantify estimation errors in arbitrary multiobject tracking problems