MOT, 02 N Object Tracking

Yao Ge, alexgecontrol@qq.com

Jan. 2022

Contents

1 Introduction

In this document **the Bayesian algorithm workflow for N object tracking** under:

- possible miss detection
- possible clutter detection
- **NEW** fixed number of objects over the tracking horizon

is presented.

2 Assumptions

Assumptions below are made for single object tracking:

- **POINT OBJECT TRACKING** – **at most one detection** per object per measurement
- **FIXED NUMBER OF OBJECTS** – there are precisely **n objects** over the tracking horizon
- **INDEPENDENT OBJECTS** – which means:
  - the state of each object is independent of others
  - the motion of each object is independent of others

3 Extensions

To begin with, first let's have a look at **extensions of single object tracking**, which are needed by N object tracking.

3.1 State and Measurement

Assume the number of objects **n** is known and constant:

First, the new state should be extended as:

$$X_k = \begin{bmatrix} x_k^1 & x_k^2 & ... & x_k^n \end{bmatrix} \tag{1}$$

And the new measurement should also be extended as:

$$\begin{aligned} Z_k &= \begin{bmatrix} z_k^1 & z_k^2 & ... & z_k^{m_k} \end{bmatrix} \\ &= \prod \begin{bmatrix} O_k & C_k \end{bmatrix} \end{aligned}$$

In which:

$$\begin{aligned} O_k &= \begin{bmatrix} o_k^1 & o_k^2 & ... & o_k^n \end{bmatrix} \\ o_k^i &= \begin{cases} [], & \text{if object i has miss detection} \\ z_k^j, & \text{if the object is detected as the measurement} \end{cases} \end{aligned}$$

And C_k is generated from the same Poisson point process used to model clutter detection inside FOV as **Single Object Detection**.

3.2 Independent Objects

If we can further assume that **the n objects are independent**, which means:

$$\textbf{Prior}: \; p(X_k) = \prod_{i=1}^{n} p(x_k^i)$$

$$\textbf{Prediction}: \; p(X_{k+1|k}) = \prod_{i=1}^{n} p(x_{k+1|k}^i)$$

With this assumption **each candidate hypothesis can be updated independently using the standard Kalman update workflow**, which can greatly simplify the posterior computing.

4 Exact Solution

4.1 Challenge

Assume for the n objects of interest, m_k measurements are sensed at timestamp k. Under **point object tracking** assumption, the total number of possible data associations at timestamp k is:

- **num. object detections** $m_k^o = \begin{pmatrix} 0 & 1 & 2 & ... & min(n, m_k) \end{pmatrix}$
- **candidate measurement combinations** $\binom{m_k}{m_k^o}$
- **candidate object detection combinations** $\binom{n}{m_k^o}$
- **data association permutation** $m_k^o!$

So the total number of possible data associations at timestamp k is:

$$N_{DA}(n, m_k) = \sum_{m_k^0=0}^{min(n,m_k)} \binom{m_k}{m_k^o} \binom{n}{m_k^o} m_k^o! \tag{2}$$

And as timestamp goes by, the total number of possible data associations from timestamp 1 to timestamp k can grow astronomically as:

$$|\Theta| = N_{DA,1:k} = N_0 \times \prod_{t=1}^{k} N_{DA}(n, m_k) \tag{3}$$

4.2 Expression

The general **predict-update** workflow for N objects tracking can be expressed as follows.

$$\textbf{Prior State :}\ p_{k-1|k-1}(X_{k-1|k-1}) = \sum_{h} w^{h}_{k-1|k-1} \times p^{h}_{k-1|k-1}(X_{k-1|k-1})$$

$$\textbf{State Prediction :}\ p_{k|k-1}(X_{k|k-1}) = \sum_{h} w^{h}_{k|k-1} \times p^{h}_{k|k-1}(X_{k|k-1})$$

$$\textbf{Posterior State :}\ p_{k|k}(X_{k|k}) \sim \sum_{h} \sum_{\theta \in \Theta} p^{\theta,h}_{k|k}(X_{k|k}) \times w^{\tilde{\theta}|h} \times w^{h}_{k|k-1}$$

If

- **the n objects are independent**
- motion / measurement are both (can be approximated as) linear and Gaussian

for each hypothesis h and data association θ, the following standard Kalman update workflow can be used:

$$
\begin{aligned}
\textbf{Prior State :}\ & x_{k-1|k-1} = \mathcal{N}^h(x_{k-1}, \hat{x}^h_{k-1|k-1}, \hat{P}^h_{k-1|k-1}) \\
\text{State Update :}\ & x_k = F^h_{k-1} x_{k-1} + q_{k-1} \\
& q_{k-1} \sim \mathcal{N}(q_{k-1}, 0, Q^h_{k-1}) \\
\textbf{State Prediction :}\ & x_{k|k-1} \sim \mathcal{N}^h(x_{k|k-1}, \check{x}^h_{k|k-1}, \check{P}^h_{k|k-1}) \\
& x_{k|k-1} \sim \mathcal{N}^h(x_{k|k-1}, F^h_{k-1}\hat{x}^h_{k-1|k-1}, F^h_{k-1}\hat{P}^h_{k-1|k-1}{F^h_{k-1}}^T + Q^h_{k-1}) \\
\text{Measurement Prediction :}\ & x_k = H_k x_k + r_k \\
& r_k \sim \mathcal{N}(r_k, 0, R_k) \\
\textbf{Measurement Likelihood:}\ & z_{k|k-1} \sim \mathcal{N}(z_{k|k-1}, \check{z}^h_{k|k-1}, \check{S}^h_{k|k-1}) \\
& z_{k|k-1} \sim \mathcal{N}(z_{k|k-1}, H_k \check{x}^h_{k|k-1}, H_k \check{P}^h_{k|k-1} H_k^T + R_k) \\
\textbf{Posterior State :}\ & x_{k|k} = \mathcal{N}(x_{k|k}, x^{\hat{\theta}|h}_{k|k}, P^{\hat{\theta}|h}_{k|k}) \\
& K^h_k = \check{P}^h_{k|k-1} H_k^T {\check{S}^h_{k|k-1}}^{-1} \\
& x^{\hat{\theta}|h}_{k|k} = \check{x}^h_{k|k-1} + K^h_k (z^\theta_k - \check{z}^h_{k|k-1}) \\
& P^{\hat{\theta}|h}_{k|k} = \check{P}^h_{k|k-1} - K^h_k H_k \check{P}^h_{k|k-1} \\
\textbf{Posterior State, Probability:}\ & w^{\tilde{\theta}|h}_{k|k} = \begin{cases} 1 - p^D, & \text{miss detection} \\ p^D \frac{\mathcal{N}^h(z^\theta_k, \check{z}^h_{k|k-1}, \check{S}^h_{k|k-1})}{\lambda_C(z^\theta_k)}, & \text{object detection is } z^\theta_k \end{cases}
\end{aligned}
$$

5 Practical Solutions

5.1 Engineering Challenge

As timestamp goes by, the total number of possible data associations from timestamp 1 to timestamp k can grow astronomically as:

$$|\Theta| = N_{DA,1:k} = N_0 \times \prod_{t=1}^{k} N_{DA}(n, m_k) \tag{4}$$

will grow astronomically as timestamp goes by, which is **NOT ACCEPTABLE** for practical engineering system. So we need **approximation algorithms** to tackle this challenge, especially the huge number of possible data associations at each timestamp.

5.2 Approximation Strategy Overview

In practice the posterior is usually dominated by a few components. To reduce the number of candidate data associations, we need an approximation algorithm which can:

- Find a subset of candidate data associations $\tilde{\Theta_k} \in \Theta_k$, such that $|\tilde{\Theta_k}| << |\Theta_k|$
- And $w_{k|k}^{\tilde{\theta}|h}, \theta \in \tilde{\Theta_k}$ is the dominant compoent in posterior mixture

5.3 Measurement Reduction through Gating

For candidate data association z_k^θ, since

$$z_k^\theta \sim \mathcal{N}(z_k^\theta, z_{k|k-1}^{\check{h}}, S_{k|k-1}^{\check{h}}) \tag{5}$$

then the measurement's **squared Mahalanobis distance from predicted mean**

$$d^2_{h_{k-1},\theta_i} \sim \chi^2(n_Z) \tag{6}$$

in which n_Z is the dimensionality of measurement Z. With this, we can choose the gating threshold as:

$$\Pr(\text{discard object detection}) = Pr\{d^2_{h_{k-1},\theta_i} > G | h_{k-1}, \theta_i, n_Z\} \tag{7}$$

which can be calculated with Chi-Squared Cumulative Distribution Table

5.4 Data Association Reduction through Optimization

If we define **Likelihood Matrix**, $L^h \in R^{n\times(m+n)}$, as

$$L^h = \left[\begin{array}{ccc|ccc} -l^{1,1} & \dots & -l^{1,m} & -l^{1,0} & \dots & +\inf \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -l^{n,1} & \dots & -l^{n,m} & +\inf & \dots & -l^{n,0} \end{array}\right] \tag{8}$$

And corresponding **Assignment Matrix**, $A^h \in \{0,1\}^{n\times(m+n)}$, as

- **Each object has at most one detection :** $\sum_j A^h_{i,j} = 1$
- **Each measurement is clutter or object detection :** $\sum_i A^h_{i,j} \leq 1$

Since

$$\begin{aligned} w^{\tilde{\theta}|h}_{k|k} &= \prod_{i=1}^{n} w^{\tilde{\theta}|h}_{i,k|k} \sim \sum_{i=1}^{n} log(w^{\tilde{\theta}|h}_{i,k|k}) = -\sum_{i=1}^{n} -log(w^{\tilde{\theta}|h}_{i,k|k}) \\ &= -tr(A^T L) \end{aligned}$$

We can formulate **most likely DA identification** as **the optimization problem** below:

- **min** : $A^h \in \{0,1\}^{n\times(m+n)}$
- **s. t.** :
  - $A^h_{i,j} \in \{0,1\}$
  - $\sum_j A^h_{i,j} = 1$
  - $\sum_i A^h_{i,j} \leq 1$

Which can be solved efficiently using one of the following algorithms:

- **Most Likely**:
  - Hugarian Algorithm at $O(n^4)$
  - Jonker–Volgenant Castanon (JVC) variant of Hugarian at $O(n^3)$
- **Top K Most Likely, in Descending Order**:
  - Murty's Algorithm
- **MCMC**:
  - Deterministic Gibbs Sampling

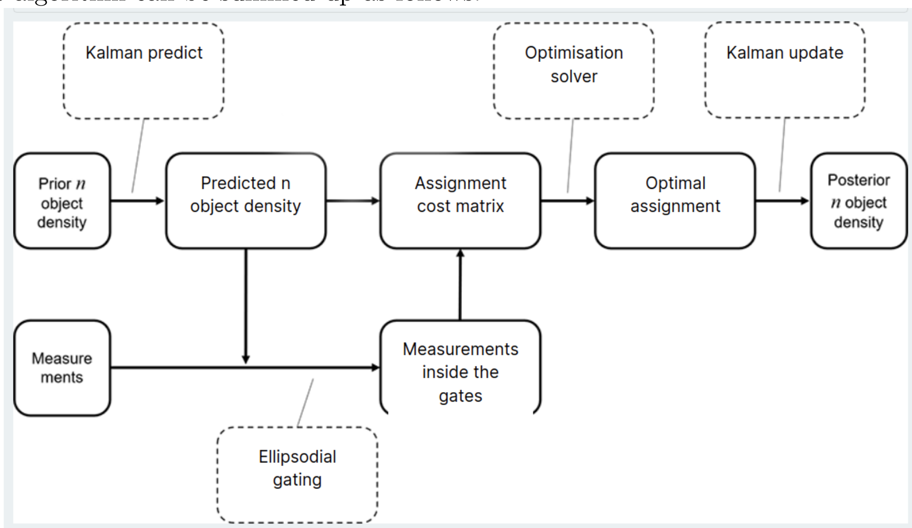
6 Representative Algorithms

Here **three representative algorithms** are presented:

- **Global Nearest Neighbor (GNN)** based on **pruning**
- **Joint Probabilistic Data Association (JPDA)** based on **merging**
- **Multi Hypotheses Tracker (MHT)** as the **benchmark algorithm**

6.1 Global Nearest Neighbor (GNN)

The algorithm can be summed up as follows:



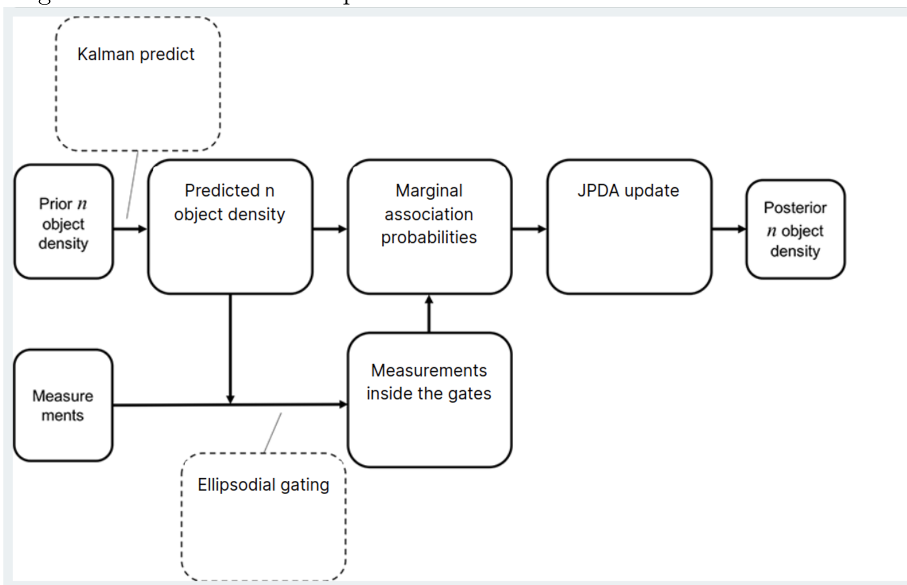
- Make prediction and filter measurements through gating
- Create **Likelihood Matrix L**
- Find optimal assignment θ^* and prune the rest data associations
- Use the resulting posterior from the above greedy assignment as posterior

Applicability:

- **Pros**
  - Simple
  - Works very well in high SNR scenarios, that is, high p^D, low λ_C and low R and when objects are well-separated from each other
- **Cons**
  - Not guaranteed to give global optimal estimation
  - Uni-modal posterior cannot model multi-modal actual posterior very well
  - Performance could deteriorates alot in middle-to-low SNR scenarios or when objects are clustered together

6.2 Joint Probabilistic Data Association (JPDA)

The algorithm can be summed up as follows:



- Make prediction and filter measurements through gating
- For each object, compute posterior for each qualified data association and approximate the resulting mixture through moment matching, which minimizes the Kullback-Leibler divergence between the two distributions
- Create joint posterior by multiplying all the marginal posteriors together

Applicability:

- **Pros**
  - Still simple to implement
  - Exact marginal posterior for each object can be approximated through
    - Cheap JDPA - Sub-optimal JDPA - Fast JDPA
    - Capping
  - Works better than GNN through merging, which maintains a higher variance to account for potential multi-modal posterior
- **Cons**
  - Uni-modal posterior cannot model multi-modal actual posterior very well
  - Performance could deteriorates alot when objects are clustered together

6.3 Multi Hypotheses Tracker (MHT)

In this section we begin with the direct implementation, **Hypothesis-Oriented MHT, HO-MHT**, then continue to a more efficient realization based on factorization of PDF, **Track-Oriented MHT, TO-MHT**.

6.3.1 Hypothesis-Oriented MHT, HO-MOT

The idea behind **HO-MHT** can be summed up as follows:

- Compare with **GNN** and **JPDA**, which only keep **1** active hypothesis, we keep $\mathcal{H}_{k-1}$ active prior hypotheses.
- In prediction step we proceed as follows. For **each active prior hypothesis** h_{k-1}
  - Predict each object independently. For each object, compute measurement likelihoods and update corresponding elements in **Likelihood Matrix** $L_{h_{k-1}}$
  - Compute $M_{h_{k-1}}$ most likely DAs using Murty's algorithm. $M_{h_{k-1}}$ can be:
    - either a fixed, hypothesis independent number
    - or $max(1, \left\lfloor M_{max} \times exp(l_{k-1}^{h_{k-1}}) \right\rfloor)$, which generates more candidate hypotheses for prior hypothesis with higher likelihood
- Then pool all candidate hypotheses together and perform the following post-processing:
  - First do **pruning**: normalize the weights then remove all hypotheses whose weight is below the threshold
  - Then do **capping**: only keep the top M_{max} most likely hypotheses
  - **done**: re-normalize the remaining weights
- Finally, for each survived candidate hypothesis, set posterior according to actual association.

6.3.2 Optimized Implementation through Factorization

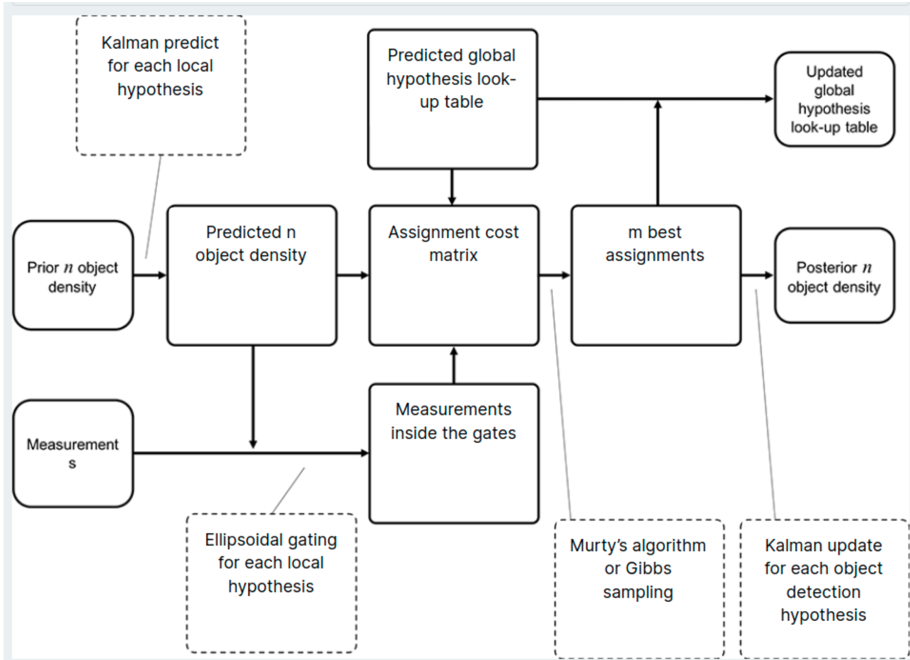
Under the assumption that **each object is independent from others**, the global hypothesis is just a product of composing local hypotheses. Therefore, we can only maintain

- component local hypotheses
- lookup table for global hypothesis construction

to efficiently represent all global hypotheses. In this way the runtime complexity can be greatly reduced.

6.4 Track-Oriented MHT, TO-MHT

The idea behind **TO-MHT** can be summed up as follows:



- Maintain the following structure rather than independent global hypotheses
  - Local hypotheses for each object
  - Global hypotheses lookup table
- In prediction step we proceed as follows.
  - Predict independently for each local hypothesis of each object. Compute measurement likelihoods and update corresponding elements in **Likelihood Matrix** L_i
- For each global hypothesis, compute $M_{h_{k-1}}$ most likely DAs and generate candidate global posteriors using Murty's algorithm. $M_{h_{k-1}}$ can be:
  - either a fixed, hypothesis independent number
  - or $max(1, \left\lfloor M_{max} \times exp(l_{k-1}^{h_{k-1}}) \right\rfloor)$, which generates more candidate hypotheses for prior hypothesis with higher likelihood
- Then pool all candidate global posteriors together and perform the following post-processing:

  - First do **pruning**: normalize the weights then remove all hypotheses whose weight is below the threshold
  - Then do **capping**: only keep the top M_{max} most likely hypotheses
  - **done**: re-normalize the remaining weights

- Finally finish update step as follows.
  - Identify composing local hypotheses using survived global hypotheses
  - Do Kalman update on local hypothesis according to actual DA
  - Re-index global hypotheses lookup table