PKF: Probabilistic Data Association Kalman Filter for Multi-Object Tracking

Hanwen Cao, *Graduate Student Member, IEEE*, George J. Pappas, *Fellow, IEEE*, and Nikolay Atanasov, *Senior Member, IEEE*

***Abstract*—In this letter, we derive a new Kalman filter (KF) with probabilistic data association between measurements and states. We formulate a variational inference problem to approximate the posterior density of the state conditioned on the measurement data. We view the unknown data association as a latent variable and apply Expectation Maximization (EM) to obtain a filter with the update step in the same form as the Kalman filter but with an expanded measurement vector of all potential associations. We show that the association probabilities can be computed as permanents of matrices with measurement likelihood entries. We name our probabilistic data association Kalman filter the *PKF* with P emphasizing both the *probabilistic* nature of the data association and the matrix *permanent* computation of the association weights. We compare PKF with the well-established Probabilistic Multi-Hypothesis Tracking (PMHT) and Joint Probabilistic Data Association Filter (JPDAF) in both theory and simulated experiments. The experiments show that we can achieve lower tracking errors than both. We also demonstrate the effectiveness of our filter in multi-object tracking (MOT) on multiple real-world datasets, including MOT17, MOT20, and DanceTrack. We can achieve comparable tracking results with previous KF-based methods without using velocities or doing multi-stage data association and remain real-time. We further show that our PKF can serve as a backbone for other KF-based trackers by applying it to a method that uses varieties of features for association, and improving its results.**

***Index Terms*—Visual tracking, probability and statistical methods, computer vision for automation.**

I. Introduction

In estimation tasks where ambiguity exists between measurements and variables of interest, probabilistic data association can prevent catastrophic estimation failures. For example, in multi-object tracking (MOT), there can be a lot of occlusions, causing high ambiguity in data association. An illustration of how probabilistic data association improves MOT is shown in Fig. 1. Methods utilizing the Kalman filter (KF) [1], [2], [3], [4] have achieved outstanding performance in MOT but have not considered the impact of probabilistic data association on the tracking process.

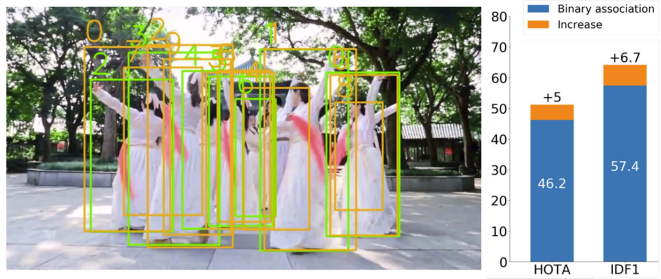


Fig. 1. Example scene with high ambiguity in DanceTrack [11] with green box detections and orange box tracks. Applying our probabilistic data association Kalman filter on this sequence significantly increases the association quality according to IDF1 [12], and the combined HOTA metric [13].

We derive a new form of Kalman filter with probabilistic data association. Previous works [5], [6] show that Kalman filter can be derived using variational inference (VI) by maximizing the evidence lower bound (ELBO) of the input and measurement likelihood. To deal with ambiguous associations, we approach the VI problem using Expectation-Maximization (EM) with the data association as the latent variable. In the E-step, we show that association weights can be computed as *permanents* of matrices with measurement likelihood entries using accelerated algorithms [7], [8]. We also show that the weight computation can be extended to cases with missing or false detections. In the M-step, optimizing the EM objective leads to a Kalman filter with the usual prediction and update steps but with an extended measurement vector and noise covariance matrix capturing all possible data associations. Our formulation is related to Probabilistic Multi-Hypothesis Tracking (PMHT) [9] and the Joint Probabilistic Data Association Filter (JPDAF) [10]. PMHT assumes independence of measurements and tracks in both E-step and M-step, which can easily be violated in some problems. The JPDAF computes a posterior given each associated measurement, i.e., maximizes the ELBO of each measurement, and then computes the weighted average of the posteriors. Our filter, instead, directly optimizes the overall ELBO weighted by the data association weights. We show that our algorithm achieves lower tracking errors than both PMHT and JPDAF, while running at comparable speeds to JPDAF.

We also apply our filter to MOT and test our algorithm on multiple real-world datasets, including MOT17 [14], MOT20 [15], and DanceTrack [11]. Without using techniques like multi-stage data association, KF re-updates, object recovery, or considering

Received 16 April 2025; accepted 25 August 2025. Date of publication 10 September 2025; date of current version 30 September 2025. This article was recommended for publication by Associate Editor R. Talak and Editor L. Pallottino upon evaluation of the reviewers' comments. This work was supported in part by NSF FRR CAREER under Grant 2045945 and in part by ARL DCIST CRA under Grant W911NF-17-2-0181. *(Corresponding author: Hanwen Cao.)*

Hanwen Cao and Nikolay Atanasov are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093 USA (e-mail: h1cao@ucsd.edu; natanasov@ucsd.edu).

George J. Pappas is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: pappasg@seas.upenn.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3608646, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3608646

2377-3766 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:04:07 UTC from IEEE Xplore. Restrictions apply.

velocities during association, our method achieves comparable performance to previous KF-based methods [2], [3] while maintaining almost the same inference speed. We further show the compatibility of our PKF to varieties of features for data association by applying it to Hybrid-SORT [4], which uses the techniques from [2], [3] in addition to using detection scores and neural features. By replacing the KF in [4] with our PKF, we can further improve its results. Our contributions are summarized as follows.

- We formulate state estimation with ambiguous measurements as a VI problem and use EM to derive the PKF, a new Kalman filter with probabilistic data association.
- We show that association probabilities can be computed using matrix permanents, which can accelerate the association weight computation.
- We demonstrate the effectiveness of the PKF in comparison to PMHT and JPDAF on simulated data. On multiple real-world MOT datasets, PKF achieves comparable results with previous state-of-the-art algorithms by associating only bounding boxes, and can improve the results by replacing the KF part, and runs at 250+ fps on a single laptop CPU, given offline detections[1].

II. Related Work

A. Probabilistic Data Association

Data association is a key challenge in estimation problems where measurements need to be related to estimated variables. The most straightforward association approach is by a nearest neighbor rule based on some distance metric, such as Euclidean or Mahalonobis [16]. However, nearest-neighbor associations are prone to mistakes. The joint compatibility branch and bound (JCBB) algorithm [17] computes a joint Mahalonobis distance and performs a joint χ^2 test for the association event to obtain robust associations. To deal with ambiguity, Bar-Shalom and Tse developed the probabilistic data association filter (PDAF) [18] for tracking single objects in the presence of clutter measurements. In the Kalman filter update step, the innovation terms are computed for each of the possible associations and are averaged weighted by the association probabilities. The performance of PDAF can degrade in the multi-object case because its separate association process can cause multiple tracks to latch onto the same object. The JPDAF [10] extends the PDAF by associating all measurements and tracks jointly. In each association event, a measurement can be associated with at most one track, and vice versa. Musicki et al. [19] develop an integrated probabilistic data association (IPDA) method using a Markov chain modeling probabilistic data association, track initialization, and termination. JIPDA [20] extends IPDA [19] to the multi-object case by performing data association jointly. There are also improvements on the original JPDAF [10] like avoiding coalescence [21] and speeding up the inference [22], [23]. Meyer et al. [24] approach probabilistic data association using a factor graph where the variables (including objects and association events) and functions (motion, observation, et al.) are nodes and the edges model their relationships. The association probabilities can be computed efficiently by message passing in the graph. A survey of data association techniques can be found in [25]. Besides MOT, probabilistic data association is also applied in other estimation tasks like biological data processing [26] and simultaneous localization and mapping [27], [28].

B. Multi-Object Tracking

State-of-the-art MOT algorithms are dominated by the tracking-by-detection approach, which consists of two steps: object detection [29], [30] and data association. SORT [1] uses the Hungarian algorithm [31] to associate bounding-box object detections and updates the object tracks using the Kalman filter to achieve real-time online tracking. DeepSORT [32] extends SORT [1] by introducing deep visual features for data association. ByteTrack [2] makes an observation that the detectors can make imperfect predictions in complex scenes and proposes a second round of data association for low-confidence detections. OC-SORT [3] analyzes the limitations of SORT [3], namely sensitivity to state noise and temporal error magnification, and proposes an observation-centric re-update step and object momentum for data association.

A second class of algorithms for MOT performs tracking-by-regression. Feichtenhofer et al. [33] use a correlation function between the feature maps of two frames to predict the transformation of bounding boxes in different frames. Bergmann et al. [34] uses the regression head of Faster-RCNN [29] to predict bounding-box offsets from the previous image to the current image. CenterTrack [35] adopts the same network architecture as CenterNet [36] but takes two consecutive images to predict object offsets between frames. Braso and Leal-Taixe [37] approach MOT with a graph neural network, which stores object features in the nodes and predicts whether two bounding boxes in different frames are the same object by edge features.

A third set of methods performs tracking-by-attention [38] using transformer models [39] for joint detection and tracking based on new object queries and tracked object queries, which solves the creation and association of objects implicitly. Although these methods utilize different techniques for data association, either explicit or implicit, they all consider deterministic data association.

III. Problem Formulation

Consider N time-varying variables that need to be estimated, denoted as $\mathbf{x}_{t,j} \in \mathbb{R}^n$, $j = 1, \ldots, N$. We refer to their concatenation into a single vector at time t:

$$\mathbf{x}_t = [\mathbf{x}_{t,1}^\top \quad \cdots \quad \mathbf{x}_{t,N}^\top]^\top \in \mathbb{R}^{nN} \tag{1}$$

as the state of a dynamical system. Suppose that $\mathbf{x}_t$ evolves according to a discrete-time linear motion model:

$$\mathbf{x}_{t+1} = F\mathbf{x}_t + G\mathbf{u}_t + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, W), \tag{2}$$

where $\mathbf{u}_t \in \mathbb{R}^l$ is the known input and $\mathbf{w}_t$ is zero-mean Gaussian noise with covariance W. At each time t, we receive M_t measurements

$$\mathbf{z}_t = [\mathbf{z}_{t,1}^\top \quad \cdots \quad \mathbf{z}_{t,M_t}^\top]^\top \in \mathbb{R}^{mM_t}. \tag{3}$$

The data association between $\mathbf{z}_t$ and $\mathbf{x}_t$ is defined as follows.

Definition 1: The *data association* of $\mathbf{z}_t$ to $\mathbf{x}_t$ is a function $\delta_t : \{1, \ldots, M_t\} \to \{0, 1, \ldots, N\}$ that either associates an element $\mathbf{z}_{t,k}$ to an element $\mathbf{x}_{t,\delta_t(k)}$ or indicates via $\delta_t(k) = 0$ that there is no matching element in $\mathbf{x}_t$.

Given association $\delta_t(k) = j$, the measurement model is

$$\mathbf{z}_{t,k} = H\mathbf{x}_{t,j} + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, V), \tag{4}$$

[1]Code is at https://github.com/hwcao17/pkf.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:04:07 UTC from IEEE Xplore. Restrictions apply.

where $\mathbf{v}_t$ is zero-mean Gaussian noise with covariance V.

Problem 1: Given an estimate of the state $\mathbf{x}_t$ at time t, an input $\mathbf{u}_t$, and measurements $\mathbf{z}_{t+1}$, obtain an estimate of $\mathbf{x}_{t+1}$ with *unknown association* between the measurements $\mathbf{z}_{t+1,k}$ and the variables $\mathbf{x}_{t+1,j}$.

IV. Methodology

We formulate a Kalman filter with probabilistic data association to solve the estimation problem defined in Section III.

A. Data Association

Following Definition 1, δ_t is the data association at time t indicating that measurement $\mathbf{z}_{t,k}$ is assigned to variable $\mathbf{x}_{t,\delta_t(k)}$. We denote the set of all possible association functions at time t as $\mathcal{D}_t = \{\delta_t : \{1,\dots,M_t\} \to \{1,\dots,N\}\}$ and the set of data associations across all times as $\mathcal{D} = \cup_{t=1}^{T} \mathcal{D}_t$.

We consider a probabilistic setting, in which we are given a prior probability density $p(\mathbf{x}_t)$ of the state $\mathbf{x}_t$, and aim to compute the posterior density $p(\mathbf{x}_{t+1}|\mathbf{z}_{t+1}, \mathbf{u}_t)$ of $\mathbf{x}_{t+1}$ conditioned on the input $\mathbf{u}_t$ and the measurements $\mathbf{z}_{t+1}$. At each time t, we treat the data association δ_t as a random variable with a uniform prior $p(\delta_t)$ independent of $\mathbf{x}_t$. Using Bayes' rule and assuming the measurements are mutually independent conditioned on the variables that generated them, the density of δ_t conditioned on $\mathbf{z}_t$ satisfies:

$$p(\delta_t|\mathbf{z}_t) = \frac{p(\mathbf{z}_t|\delta_t)p(\delta_t)}{p(\mathbf{z}_t)} \propto p(\mathbf{z}_t|\delta_t) = \prod_{k=1}^{M_t} p\left(\mathbf{z}_{t,k}|\delta_t(k)\right). \quad (5)$$

Let $p_{\mathcal{N}}(\cdot; \boldsymbol{\mu}, \Sigma)$ denote the density of a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance Σ. Assuming Gaussian prior $p(\mathbf{x}_{t,\delta_t(k)}) = p_{\mathcal{N}}(\mathbf{x}_{t,\delta_t(k)}; \boldsymbol{\mu}_{t,\delta_t(k)}, \Sigma_{t,\delta_t(k)})$ and measurement $p(\mathbf{z}_{t,k}|\mathbf{x}_{t,\delta_t(k)}) = p_{\mathcal{N}}(\mathbf{z}_{t,k}; H\mathbf{x}_{t,\delta_t(k)}, V)$, we have:

$$p(\mathbf{z}_{t,k}|\delta_t(k)) = p_{\mathcal{N}}(\mathbf{z}_{t,k}; H\boldsymbol{\mu}_{t,\delta_t(k)}, H\Sigma_{t,\delta_t(k)}H^\top + V).$$

Consider the likelihood that a specific measurement $\mathbf{z}_{t,k}$ is generated by $\mathbf{x}_{t,j}$, denoted by $w_{k,j}^t$. Using (5), we have:

$$w_{k,j}^t = \sum_{\delta_t \in \mathcal{D}_t(k,j)} p(\delta_t|\mathbf{z}_t) = \sum_{\delta_t \in \mathcal{D}_t(k,j)} \prod_{r=1}^{M_t} p\left(\mathbf{z}_{t,r}|\delta_t(r)\right), \quad (6)$$

where $\mathcal{D}_t(k,j)$ is the set of data association functions that assign measurement $\mathbf{z}_{t,k}$ to variable $\mathbf{x}_{t,j}$. We show that $w_{k,j}^t$ can be computed using the *permanent* of a matrix Q containing the measurement likelihood $p(\mathbf{z}_{t,k}|\mathbf{x}_{t,j})$ as its entries $Q(k,j)$.

Definition 2: The permanent of a matrix $Q = [Q(k,j)] \in \mathbb{R}^{M\times N}$ with $M \le N$ is:

$$\text{per}(Q) := \sum_{\delta \in \mathcal{D}} \prod_{k=1}^{M} Q(k, \delta(k)), \quad (7)$$

where $\mathcal{D}$ is the set of injective functions $\delta : \{1,\dots,M\} \to \{1,\dots,N\}$.

Given this definition, we have the following proposition to compute the data association weights.

Proposition 1: Let $Q^t \in \mathbb{R}^{M_t \times N}$ be a matrix with elements $Q^t(k,j) = p(\mathbf{z}_{t,k}|\delta_t(k) = j) = p(\mathbf{z}_{t,k}|\mathbf{x}_{t,j})$. The data association weight $w_{k,j}^t$ in can be computed as:

$$w_{k,j}^t \propto Q^t(k,j)\text{per}\left(Q_{-kj}^t\right),$$

where Q_{-kj}^t is the matrix Q^t with the k-th row and j-th column removed.

Proof: See Supplementary Material Section VII. ∎

Proposition 1 allows us to compute data association weights. Next, we derive our PKF.

B. Probabilistic Data Association Kalman Filter

To derive a filter, we consider two time steps, namely t and $t+1$, and define a lifted form of the state:

$$\bar{\mathbf{x}} = \begin{bmatrix}\mathbf{x}_t^\top & \mathbf{x}_{t+1}^\top\end{bmatrix}^\top. \quad (8)$$

Given the prior density $p(\mathbf{x}_t)$, the input $\mathbf{u}_t$, and the measurement $\mathbf{z}_{t+1}$, we aim to find an estimate $q(\bar{\mathbf{x}})$ of the joint posterior $p(\bar{\mathbf{x}}|\mathbf{z}_{t+1}, \mathbf{u}_t)$. To determine $q(\bar{\mathbf{x}})$, we use variational inference [40, Ch. 10], which maximizes the lower bound of the evidence $\log p(\mathbf{z}_{t+1}, \mathbf{u}_t)$ with respect to $q(\bar{\mathbf{x}})$. To save space, we denote $q(\bar{\mathbf{x}})$ by q. The evidence can be decomposed as [40, Ch. 10]:

$$\log p(\mathbf{z}_{t+1}, \mathbf{u}_t) = \text{KL}(q||p(\bar{\mathbf{x}}|\mathbf{z}_{t+1}, \mathbf{u}_t)) + \mathcal{L}(q), \quad (9)$$

$$\mathcal{L}(q) = \mathbb{E}_q\left[\log \frac{p(\bar{\mathbf{x}}, \mathbf{z}_{t+1}, \mathbf{u}_t)}{q(\bar{\mathbf{x}})}\right], \quad (10)$$

where $\mathcal{L}(q)$ is the ELBO (evidence lower bound) since the Kullback–Leibler divergence $\text{KL}(q||p) \ge 0$. Following standard practice [40, Ch. 10], we optimize the ELBO

$$q^* \in \arg\max_q \mathcal{L}(q) = \arg\max_q \mathbb{E}_q\left[\log \frac{p(\bar{\mathbf{x}}, \mathbf{z}_{t+1}, \mathbf{u}_t)}{q(\bar{\mathbf{x}})}\right]. \quad (11)$$

To account for probabilistic data association, we formulate the optimization by the EM algorithm [40, Ch. 9]. EM determines the maximum likelihood (ML) or maximum a posteriori (MAP) in the presence of unobserved variables. It contains an E-step that computes the expectation w.r.t. the unobserved variables and an M-step that maximizes the log-likelihood. Letting the data association δ_{t+1} be the unobserved variable, we consider the following problem:

$$\begin{aligned} q^{(i+1)} &\in \arg\max_q f^{(i)}(q), \\ &= \arg\max_q \mathbb{E}_{\delta_{t+1}}\mathbb{E}_{\bar{\mathbf{x}}^{(i)}}\left[\mathcal{L}(q, \delta_{t+1}) \mid \bar{\mathbf{x}}^{(i)}, \mathbf{z}_{t+1}\right] \\ &= \arg\max_q \mathbb{E}_{\delta_{t+1}}\mathbb{E}_{\bar{\mathbf{x}}^{(i)}}\left[\mathbb{E}_q\left[\log\frac{p(\bar{\mathbf{x}}, \mathbf{z}_{t+1}, \mathbf{u}_t, \delta_{t+1})}{q(\bar{\mathbf{x}})}\right]|\bar{\mathbf{x}}^{(i)}, \mathbf{z}_{t+1}\right] \\ &= \arg\max_q \sum_{k=1}^{M_t}\sum_{j=1}^{N} \omega_{kj}^{t+1} \log p(\mathbf{z}_{t+1,k}|\mathbf{x}_{t+1,j}), \end{aligned} \quad (12)$$

where the expectation $\mathbb{E}_{\bar{\mathbf{x}}^{(i)}}$ is with respect to $\bar{\mathbf{x}}^{(i)} \sim q^{(i)}(\bar{\mathbf{x}})$. EM splits the optimization in (12) in two steps. The E-step requires computing the data association likelihood $p(\delta_{t+1}|\mathbf{z}_{t+1})$. This can be obtained from (5) and Proposition 1. Given the data association weights $w_{k,j}^t$, the M-step performs the optimization in (12) to determine $q^{(i+1)}$.

We show that performing the E and M steps for one iteration, with initialization $\bar{\mathbf{x}}^{(i)}$ given by the prior state and the predicted state in (2), is equivalent to a Kalman filter with probabilistic data association. We first define an expanded observation model that captures all possible ways of generating the measurements

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:04:07 UTC from IEEE Xplore. Restrictions apply.

at each time t:

$$\bar{\mathbf{z}}_t = \bar{H}_t \mathbf{x}_t + \bar{\mathbf{v}}_t, \qquad \bar{\mathbf{v}}_t \sim \mathcal{N}(\mathbf{0}, \bar{V}_t), \tag{13}$$

where $\bar{\mathbf{z}}_t = (I_{M_t} \otimes \mathbf{1}_N \otimes I_m)\mathbf{z}_t \in \mathbb{R}^{mM_tN}$, $\bar{H}_t = \mathbf{1}_{M_t} \otimes \mathbf{I}_N \otimes H \in \mathbb{R}^{mM_tN \times nN}$, $\mathbf{1}_N \in \mathbb{R}^N$ is a vector with elements equal to one, $\mathbf{I}_N \in \mathbb{R}^{N \times N}$ is the identity matrix, $\otimes$ is the Kronecker product, and $\bar{\mathbf{v}}_t$ is zero-mean Gaussian noise with covariance:

$$\bar{V}_t = \begin{bmatrix} \bar{V}_{t,1} & & \\ & \ddots & \\ & & \bar{V}_{t,M_t} \end{bmatrix}, \bar{V}_{t,k} = \begin{bmatrix} V/w^t_{k,1} & & \\ & \ddots & \\ & & V/w^t_{k,N} \end{bmatrix}, \tag{14}$$

where $w^t_{k,j}$ are the data association weights in (6). Using the expanded measurement model in (13), we obtain a Kalman filter with probabilistic data association.

Proposition 2: Given prior $\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$ and input $\mathbf{u}_t$, the predicted Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^+_{t+1}, \Sigma^+_{t+1})$ of $\mathbf{x}_{t+1}$ computed by the Kalman filter with motion model in (2) has parameters:

$$\begin{aligned} \boldsymbol{\mu}^+_{t+1} &= F\boldsymbol{\mu}_t + G\mathbf{u}_t, \\ \Sigma^+_{t+1} &= F\Sigma_t F^\top + W. \end{aligned} \tag{15}$$

Given measurements $\mathbf{z}_{t+1}$, the updated Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{t+1}, \Sigma_{t+1})$ of $\mathbf{x}_{t+1}$ conditioned on $\mathbf{z}_{t+1}$ computed by the Kalman filter with probabilistic data association is obtained from the expanded measurement model in (13) as:

$$\begin{aligned} \boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}^+_{t+1} + \bar{K}_{t+1}(\bar{\mathbf{z}}_{t+1} - \bar{H}_{t+1}\boldsymbol{\mu}^+_{t+1}), \\ \Sigma_{t+1} &= (I - \bar{K}_{t+1}\bar{H}_{t+1})\Sigma^+_{t+1}, \end{aligned} \tag{16}$$

where the Kalman gain is:

$$\bar{K}_{t+1} = \Sigma^+_{t+1}\bar{H}^\top_{t+1}\left(\bar{H}_{t+1}\Sigma^+_{t+1}\bar{H}^\top_{t+1} + \bar{V}_{t+1}\right)^{-1}. \tag{17}$$

Proof: See Supplementary Material Section VIII. ■

C. Comparison With PMHT and JPDAF

We discuss the relationship of PKF to PMHT [9] and JPDAF [10] and compare them in simulated experiments.

1) Comparison With PMHT: The difference between our PKF and PMHT [9] is that PMHT assumes independence in both the E-step (association) and M-step (update), while the PKF does not. During association, PMHT computes the probability of each measurement and object individually without considering if the measurement is already assigned to another object. In contrast, PKF does the association in a joint manner by evaluating probabilities of the injective association functions, where a measurement can not be assigned to another object after being assigned to one. During the update step, PMHT updates each object individually, while PKF updates objects jointly, i.e., there are cross-correlation blocks is the object covariances.

2) Comparison With JPDAF: The major difference is in the update step, we first show the update scheme difference, then show that the association weights in JPDAF can be computed using Proposition 1.

Update scheme Both JPDAF [10] and PKF can be decomposed into two stages. The JPDAF first computes a posterior mean as the normal Kalman filter with each associated measurement, and then averages the posterior means with the association weights and computes the covariance correspondingly. Instead, our PKF first constructs an expanded measurement model and then performs a single update in the same form as the normal Kalman filter but with an expanded measurement vector. The JPDAF maximizes the ELBO with regard to each associated measurement and then averages those estimates, while our PKF maximizes the total weight-averaged ELBO, which does not necessarily maximize each measurement's individual ELBO but may lead to a higher total ELBO. This difference is underscored by the lower tracking errors in the simulations in Section IV-C3.

Both algorithms can be performed in a coupling or decoupling way. In the coupling way, all the objects are stored in one filter and updated jointly. In the decoupling way, each object is stored and updated separately, i.e., setting the cross-correlation blocks of Σ_t in Proposition 2 to 0. In the experiments, we do decoupling updates just like JPDAF. Regarding the complexity, the Kalman gain in JPDAF is the same as the normal Kalman filter but JPDAF needs all the measurement residuals to update the mean and covariance. In PKF, the complexity of the Kalman gain computation increases due to the expanded measurement model. The original Kalman filter has complexity $\mathcal{O}(m^{2.4} + n^2)$ [41] in terms of the measurement dimension m and state dimension n. Given the number of associated measurements M_t, the complexity of PKF is $\mathcal{O}((M_t m)^{2.4} + M_t n^2)$.

Weight computation Following JPDAF [10], we make the assumption that 1) the number of established objects is known and 2) there can be missing or false measurements (clutter). Using our notation, the probability of an association event δ_t in [10, Eq. 47] can be written as

$$p(\delta_t|\mathbf{z}_t) \propto \prod_k \left(\frac{1}{\lambda} p(\mathbf{z}_{t,k}|\mathbf{x}_{t,\delta_t(k)})\right)^{\tau_k} \prod_j (p^t_D)^{\sigma_j}(1 - p^t_D)^{1-\sigma_j}$$

where λ is the spatial density of the false measurements Poisson pmf, p^t_D is the detection probability, τ_k is an indicator of whether $\mathbf{z}_{t,k}$ is associated in the event, i.e. $\mathbf{z}_{t,k}$ is not treated as clutter, and σ_j is an indicator of whether the object $\mathbf{x}_{t,j}$ is associated. Suppose that the number of clutter points in δ_t is Φ_t, the number of measurements is M_t, and the number of objects is N. Since the number of established objects is assumed known, we have $M_t - \Phi_t \leq N$. With the number of clutter points fixed, $p(\delta_t|\mathbf{z}_t)$ can be simplified as

$$\begin{aligned} p(\delta_t|\mathbf{z}_t) &\propto c^{\Phi_t}_t \prod_k p(\mathbf{z}_{t,k}|\mathbf{x}_{t,\delta_t(k)}), \quad \delta_t \in \mathcal{D}^{\Phi_t}_t, \\ c^{\Phi_t}_t &= \lambda^{-\Phi_t}(p^t_D)^{M_t - \Phi_t}(1 - p^t_D)^{N_t - M_t + \Phi_t}, \end{aligned} \tag{18}$$

where $\mathcal{D}^{\Phi_t}_t$ is the set of association events that have Φ_t false measurements. The probability of measurement $\mathbf{z}_{t,k}$ being assigned to object $\mathbf{x}_{t,j}$ is computed in [10, Eq. 51] as

$$\begin{aligned} w^t_{k,j} &\triangleq \sum_{\delta_t \in \mathcal{D}_t(k,j)} p(\delta_t|\mathbf{z}_t) = \sum_{\Phi_t} \sum_{\delta_t \in \mathcal{D}^{\Phi_t}_t(k,j)} p(\delta_t|\mathbf{z}_t) \\ &\propto \sum_{\Phi_t} \sum_{\delta_t \in \mathcal{D}^{\Phi_t}_t(k,j)} c^{\Phi_t}_t \prod_k p(\mathbf{z}_{t,k}|\mathbf{x}_{t,j}), \end{aligned} \tag{19}$$

where $\mathcal{D}^{\Phi_t}_t(k,j)$ is the set of associations having Φ_t false measurements and assigning $\mathbf{z}_{t,k}$ to $\mathbf{x}_{t,j}$. Selecting a series of sets of false measurements of size $\Phi_t = \max\{0, M_t - N\} \cdots M_t$, for each fixed Φ_t, the second sum on the last line of (19) can be computed via Proposition 1.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:04:07 UTC from IEEE Xplore. Restrictions apply.

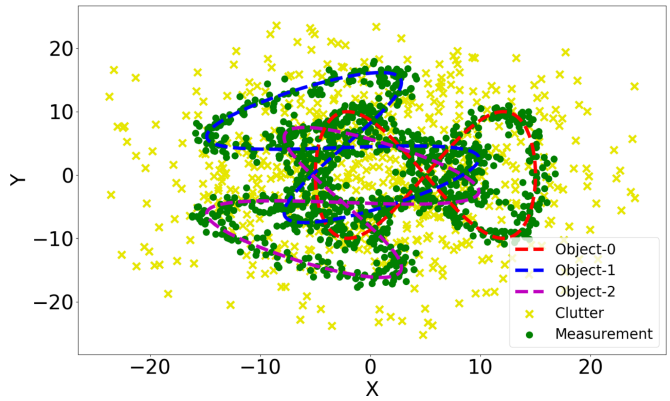


Fig. 2. Object tracking simulation showing ground-truth object trajectories (dashed lines), object measurements (green dots), and clutter (false) measurements (yellow crosses).

TABLE I
COMPARISON OF TRACKING ERRORS ($l2$ NORM IN METERS)

N_{obj}	Method	Obj 1	Obj 2	Obj 3	Obj 4	Obj 5	Avg
3	Binary	2.16	1.36	1.80	-	-	1.78
	PMHT [9]	1.50	1.29	1.55	-	-	1.45
	JPDAF [10]	0.70	0.60	0.65	-	-	0.65
	PKF	**0.70**	**0.57**	**0.58**	-	-	**0.62**
5	Binary	8.37	19.36	17.34	11.37	14.53	14.20
	PMHT [9]	12.41	12.34	12.43	12.10	11.21	12.10
	JPDAF [10]	0.68	0.73	0.62	0.63	0.66	0.66
	PKF	**0.60**	**0.68**	**0.58**	**0.61**	**0.61**	**0.62**

3) Tracking Error: We compare our PKF to PMHT [9] and JPDAF [10] with 2D simulated data, where each object is a point and moves along an 8-shape trajectory. The detection probability of each object is $p_D^t = 0.9$, the measurement noise is a zero-mean Gaussian with covariance $\text{diag}(0.75, 0.75)$, and clutter is sampled uniformly in the range of $[-10, 10]^2$ around each ground-truth point. We use $\lambda = 0.125$ for the spatial density of the Poisson pmf. A visualization of 3 simulated objects can be found in Fig. 2.

The errors of tracking 3 and 5 objects in simulation are shown in Table I. PMHT [9] works slightly better than binary ssociation but much worse than PKF and JPDAF [10] due to the non-exclusive data association. PKF achieves lower tracking errors than JPDAF implemented by [42]. Our intuition as to why PKF achieves lower errors is provided in the first paragraph of Section IV-C2. We also compare PKF to JPDAF under different levels of measurement noise with 10 objects. In this setting, we increase the detection probability to 0.95 to avoid frequent tracking failures. The clutter generation remains the same, and we test with measurement noises from 0.2 to 0.75. The results can be found in Fig. 3. We can see that our filter obtains lower tracking errors in most cases. The number of failed tracks is almost the same as JPDAF [10].

4) Update Speed: We compare the update speeds of these methods with different numbers of objects. The results can be found in Table II. The experiments show that PKF can track at a comparable speed with JPDAF.

V. APPLICATION TO MULTI-OBJECT TRACKING

In this section, we apply our PKF to the MOT task.

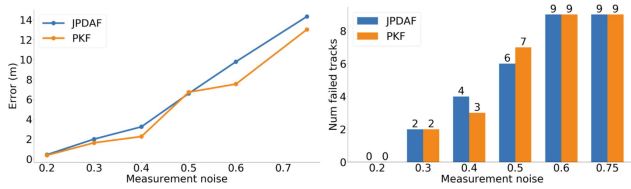


Fig. 3. Ablation study with 10 objects in the simulation under different noise scales. Left: tracking errors in meters. Right: number of failed tracks, i.e., those with tracking errors larger than 5.

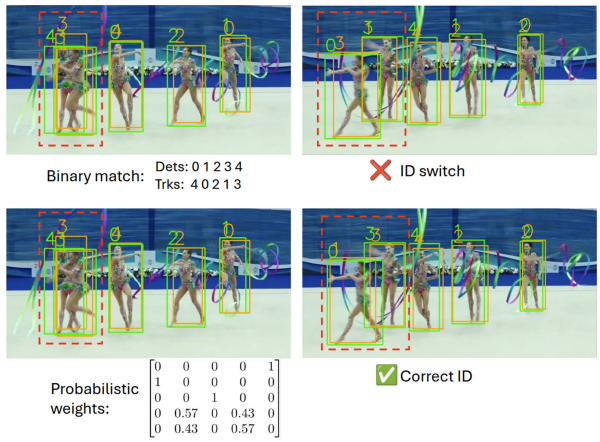


Fig. 4. Illustration of our data association. The detections and tracks are shown in green and orange. The first row shows the result of binary association via [48]. The second row is our proposed data association. In the first image, the detections highly overlap, and our ambiguity check recognizes these as ambiguous (dashed red boxes). The binary association causes an ID switch, but our method can track the objects correctly, as shown in the third image.

TABLE II
UPDATE TIME (MS) PER FRAME GIVEN ASSOCIATION WEIGHTS

Number of objects	3	5	10	20	Avg
Vanilla Kalman filter	0.08	0.12	0.20	0.36	0.19
PMHT [9]	0.11	0.16	0.28	0.55	0.28
JPDAF [10]	0.28	0.39	0.78	1.47	0.73
PKF	0.31	0.46	0.93	1.76	0.87

A. Algorithm Design

To make the filter work efficiently in practice, we introduce an ambiguity check to avoid outliers and unnecessary computation. Also, we force the state covariance matrix to be block diagonal so that each track is estimated by a separate filter, although the states are still considered jointly when evaluating the association probabilities. Following SORT [1], we have $\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^\top$ and $\mathbf{z} = [u, v, s, r]^\top$ where u and v are the horizontal and vertical pixel location of the center of the object box, s and r represent the scale (area) and the aspect ratio of the object's box, and $\dot{()}$ terms are velocities. Details about the motion and measurement model matrices can be found in Section IX.

Ambiguity check Gating is very important in Kalman filtering [10]. We propose a simple yet efficient way to find ambiguous measurements in the meantime of removing potential outliers (gating). Given measurements $\mathbf{z}_{t,k}$, $k = 1, \ldots, M_t$ and states $\mathbf{x}_{t,j}$, $j = 1, \ldots, N$, we first compute a score matrix $S_t \in \mathbb{R}^{M_t \times N}$ with entries $[S_t]_{kj}$ equal to the intersection over

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:04:07 UTC from IEEE Xplore. Restrictions apply.



Fig. 5. Visualizations of different datasets.

union (IoU) for each measurement-object bounding-box pair. Then, we perform an ambiguity check. For each row k of S_t (corresponding to measurement $\mathbf{z}_{t,k}$), we sort the scores from high to low. Supposing the rank is $\mathbf{x}_{t,j_1}\ \mathbf{x}_{t,j_2}\cdots\mathbf{x}_{t,j_N}$, if object $\mathbf{x}_{t,j_2}$ has a score over a threshold τ_{ambig}, e.g., 90% the score of object $\mathbf{x}_{t,j_1}$, then $\mathbf{x}_{t,j_1}$ and $\mathbf{x}_{t,j_2}$ are ambiguous objects, and the measurement $\mathbf{z}_{t,k}$ is an ambiguous measurement. Then, we compare $\mathbf{x}_{t,j_2}$ and $\mathbf{x}_{t,j_3}$ and so on. We repeat this for all objects and measurements to obtain an ambiguous set. Measurements and objects with their matches marked as ambiguous will also be added to the ambiguous set. Proposition 1 is only applied to the ambiguous set of measurements and objects. The rest of the measurements and objects are associated with binary association [31], [48]. An illustration of how our proposed probabilistic data association Kalman filter benefits the ambiguous case is shown in Fig. 4, where binary association causes a wrong ID switch while our proposed association can obtain a correct ID.

We create a filter for each new object (equivalent to enforcing a block-diagonal covariance matrix). This avoids a large state covariance and observation Jacobian $\bar{H}_{t+1}$ in (17), which speeds up the tracking especially when the number of objects is large. We still compute the association weight matrix W jointly, each column vector $\mathbf{w}_j$ corresponds to a tracked object with measurement association weights. Measurements with a weight larger than a threshold τ_{weight} are used to update the tracker by Proposition 2, which helps exclude potential outliers and achieve stable computation.

B. Evaluation

Datasets We evaluate PKF on multiple datasets, including MOT17 [14], MOT20 [15], and DanceTrack [11]. MOT17 and MOT20 focus on crowded pedestrian tracking in public places. DanceTrack is a dancing-scene dataset, where the objects have similar appearances, move fast, highly non-linearly, and cross over each other frequently, thus imposing a higher requirement on data association.

Metrics We use higher order tracking accuracy (HOTA) [13] as our main metric because it balances between detection accuracy and association accuracy. We also report the AssA [13] and IDF1 [12] metrics, which emphasize data association accuracy and the MOTA [49] metric, which emphasizes the detection accuracy.

Implementation In performing the data association ambiguity check, we compute the IoU between detections and tracked bounding boxes. To construct the matrix Q in Proposition 1, we treat $UoI := 1/IoU$ as a distance (not an actual distance function mathematically) and compute the conditional probability as $p(\mathbf{z}_{t,k}|\mathbf{x}_{t,j}) \propto \exp(-\alpha UoI)$, where α is a scaling factor. We found $\alpha = 2$ to work well in practice. On MOT17 [14] and DanceTrack [11], the ambiguity check threshold was set to $\tau_{ambig} = 0.9$. On MOT20, we set $\tau_{ambig} = 0.95$ since the pedestrians are more crowded. The weight threshold was set at $\tau_{weight} = 0.25$. We used the publicly available YOLOX weights by ByteTrack [2]. Following the common practice of SORT [1], we set the detection confidence threshold at 0.4 for MOT20 and 0.6 for other datasets. A new track is created if a detected box has IoU lower than 0.3 with all the tracks. All experiments were conducted on a laptop with i9-11980HK@2.60 CPU, 16 GB RAM, and RTX 3080 GPU.

C. Benchmark Results

In this section, we first show that PKF can get comparable results to other methods [2], [3] by only associating bounding boxes, i.e., without using any other techniques or features. We then show that PKF is compatible with more advanced object features can serve as a backbone, same as SORT [1].

MOT17 and MOT20 The quantitative results on MOT17 [14] and MOT20 [15] are shown in Table III and IV. To get a fair comparison, we use the same detections as ByteTrack [2] and OC-SORT [3] and inherit the linear interpolation. While the direct comparison to our PKF is the vanilla Kalman filter, we can achieve comparable results to ByteTrack [2] and OC-SORT [3], which do multiple rounds of associations [2], make use of velocities [3], and re-updates [3]. We can see that PKF is able to achieve better data association, as indicated by IDF1 and AssA, and achieves slightly higher HOTA results.

DanceTrack The results on DanceTrack [11] are shown in Table V. We can see that PKF achieves better results than SORT [1] and ByteTrack [2]. In this dataset, there are a lot of occlusions and the dancers can move dramatically. The increase of AssA and IDF1 over SORT [1] shows again the advantage of PKF in terms of association.

Application to Hybrid-SORT-ReID Besides data association techniques from [2], [3], Hybrid-SORT-ReID [4] further uses detection scores, a height-enhanced IoU, and a ReID module with deep neural features. We show that our PKF is compatible with these features, just as SORT [1] is, by replacing the SORT part (Kalman filter) with PKF. We name the new method Hybrid-PKF-ReID. With the score matrix computed by Hybrid-SORT-ReID using all these features, we compute the measurement probability as $p(\mathbf{z}_{t,k}|\mathbf{x}_{t,j}) \propto \exp(-\alpha/s_{k,j})$, where $s_{k,j}$ is the association score of $\mathbf{z}_{t,k}$ and $\mathbf{x}_{t,j}$. We test Hybrid-PKF-ReID on MOT20 and Dancetrack, which have more ambiguous scenarios. Table VI shows that adding probabilistic association improves Hybrid-SORT-ReID's results. We show some sequences that PKF improves in Fig. 5, which are highly occluded. This suggests that PKF is effective in highly ambiguous cases.

D. Ablation Study

We compare the effect of ambiguity checks and gating, and analyze the computation time of the matrix permanent.

Ambiguity check We compare the performance of PKF with and without ambiguity check in Table VII. We observe that associating all measurements and tracks directly degrades the performance due to updating with low-weight measurements, which are mostly outliers. Introducing the ambiguity check in PKF to discard low-probability measurements before computing

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:04:07 UTC from IEEE Xplore. Restrictions apply.

TABLE III
RESULTS ON MOT17 [14] TESTSET WITH PRIVATE DETECTIONS. METHODS IN BLUE SHARE DETECTIONS

Tracker	HOTA ↑	MOTA ↑	IDF1 ↑	FP(10^4) ↓	FN(10^4) ↓	IDs ↓	Frag ↓	AssA ↑	AssR ↑
FairMOT [43]	59.3	73.7	72.3	2.75	11.7	3303	8073	58.0	63.6
QDTrack [44]	53.9	68.7	66.3	2.66	14.7	3378	8091	52.7	57.2
MOTR [45]	57.2	71.9	68.4	2.11	13.6	2115	3897	55.8	59.2
TransMOT [46]	61.7	76.7	75.1	3.62	9.32	2346	7719	59.9	66.5
MeMOT [47]	56.9	72.5	69.0	2.72	11.5	2724	-	55.2	-
ByteTrack [2]	63.1	80.3	77.3	2.55	8.37	2196	2277	62.0	68.2
OC-SORT [3]	63.2	78.0	77.5	1.51	10.8	1950	2040	63.2	67.5
PKF	**63.3**	**78.3**	**77.7**	**1.56**	**10.5**	**2130**	**2754**	**63.4**	**67.4**

TABLE IV
RESULTS ON MOT20 [15] TESTSET WITH PRIVATE DETECTIONS. METHODS IN BLUE SHARE DETECTIONS

Tracker	HOTA ↑	MOTA ↑	IDF1 ↑	FP(10^4) ↓	FN(10^4) ↓	IDs ↓	Frag ↓	AssA ↑	AssR ↑
FairMOT [43]	54.6	61.8	67.3	10.3	8.89	5243	7874	54.7	60.7
TransMOT [46]	61.9	77.5	75.2	3.42	8.08	1615	2421	60.1	66.3
MeMOT [47]	54.1	63.7	66.1	4.79	13.8	1938	-	55.0	-
ByteTrack [2]	61.3	77.8	75.2	2.62	8.76	1223	1460	59.6	66.2
OC-SORT [3]	62.1	75.5	75.9	1.80	10.8	913	1198	62.0	67.5
PKF	**62.3**	**75.4**	**76.3**	**1.73**	**10.9**	**980**	**1584**	**62.7**	**67.6**

TABLE V
DANCETRACK [11] RESULTS. BLUE MEANS SHARED DETECTIONS

Tracker	HOTA ↑	DetA ↑	AssA ↑	MOTA ↑	IDF1 ↑
FairMOT [43]	39.7	66.7	23.8	82.2	40.8
QDTrack [44]	45.7	72.1	29.2	83.0	44.8
TraDes [50]	43.3	74.5	25.4	86.2	41.2
MOTR [45]	54.2	73.5	40.2	79.7	51.5
SORT [1]	50.6	80.2	32.0	89.2	48.9
DeepSORT [32]	45.6	71.0	29.7	87.8	47.9
ByteTrack [2]	47.3	71.6	31.4	89.5	52.5
OC-SORT [3]	54.5	80.4	37.1	89.4	54.0
PKF	**53.7**	**79.7**	**36.2**	**89.1**	**53.5**

TABLE VI
APPLICATION TO HYBRID-SORT-REID [4]

Dataset	Tracker	HOTA ↑	AssA ↑	MOTA ↑	IDF1 ↑
MOT20	Hybrid-SORT-ReID [4]	63.9	64.7	76.7	78.4
	Hybrid-PKF-ReID	**64.3**	**65.1**	76.7	**79.0**
DanceTrack	Hybrid-SORT-ReID [4]	65.5	52.2	91.8	67.2
	Hybrid-PKF-ReID	**66.4**	**53.9**	91.6	**68.7**

TABLE VII
ABLATION STUDY OF AMBIGUITY CHECK (A.C.).

	MOT17			DanceTrack		
Metrics	HOTA ↑	IDF1 ↑	t (ms)	HOTA ↑	IDF1 ↑	t (ms)
Binary	64.9	76.9	**5.9**	47.8	48.3	**3.4**
PKF w/o A.C.	44.9	52.8	4970.6	44.2	45.2	121.2
PKF	**66.8**	**78.5**	6.2	**53.5**	**53.3**	3.7

the matrix permanents, leads to better results than binary association with almost the same tracking speed. Table VIII shows that our ambiguity check has a similar effect to Chi-squared gating in the JPDAF [10] with 5 objects in simulation. We do not do gating in real-world data since $1/IoU$ is ill-conditioned for Chi-squared test while the ambiguity check is more flexible with different features.

TABLE VIII
EFFECT OF GATING AND AMBIGUITY CHECK IN THE JPDAF [10] ON THE $l2$ TRACKING ERROR (METERS) IN SIMULATION

Method	Obj 1	Obj 2	Obj 3	Obj 4	Obj 5	Avg
Binary	8.37	19.36	17.34	11.37	14.53	14.20
JPDAF [10] w/o gating	19.32	18.31	10.60	16.15	11.63	15.21
JPDAF [10] w/ gating	0.68	**0.73**	**0.62**	**0.63**	0.66	**0.66**
JPDAF [10] w/ A.C.	**0.62**	1.47	0.62	0.63	**0.58**	0.79

TABLE IX
COMPUTATION TIME (MS) OF MATRIX PERMANENT AND OVERALL ASSOCIATION WEIGHTS WITH DIFFERENT MATRIX SIZES

Matrix size	5×5	10×10	15×15	20×20	30×30
Nijenhuis et al. [7]	0.003	0.01	2.0	26.5	31510.9
Huber et al. [8]	6.5	20.1	21.7	35.4	67.8
Best	0.003	0.01	2.0	26.5	67.8
Weight compute	0.03	0.2	14.5	1073.6	1766.6

Matrix permanent computation time We analyze the computation time of the matrix permanent used to obtained the association weights. We take the faster matrix permanent method between [7] and [8] (switch at size 20×20). Table IX shows that the speed is very fast for matrices smaller than 15×15 and computations on matrices up to 30×30 run at over 0.5 fps.

Discussion We observe that achieving effective use of probabilistic data association requires a balance between removing outliers (gating) and keeping in ambiguous measurements. If there is no gating, the outliers will bring bias to the measurements. On the other hand, if only the most likely measurements are considered, wrong hard decisions could harm the estimation. The probability and gating model can be improved to unlock further potential of PKF.

VI. CONCLUSION

We derived a new formulation of the Kalman filter with probabilistic data association by formulating a variational inference

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:04:07 UTC from IEEE Xplore. Restrictions apply.

problem and introducing data association as a latent variable in the EM algorithm. We showed that, in the E-step, the association probabilities can be computed as matrix permanents, while the M-step led to the usual Kalman filter prediction and update steps but with an extended measurement model. Our experiments demonstrated that our filter can outperform the JPDAF and can achieve good results on MOT benchmarks. Our algorithm is not restricted to the MOT application and can serve as a general method for estimation problems with measurement ambiguity.

REFERENCES

[1] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. Int. Conf. Image Process.*, 2016, pp. 3464–3468.

[2] Y. Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 1–21.

[3] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 9686–9696.

[4] M. Yang et al., "Hybrid-sort: Weak cues matter for online multi-object tracking," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, pp. 6504–6512.

[5] T. D. Barfoot, J. R. Forbes, and D. J. Yoon, "Exactly sparse Gaussian variational inference with application to derivative-free batch nonlinear state estimation," *IEEE Int. J. Robot. Res.*, vol. 39, no. 13, pp. 1473–1502, 2020.

[6] H. Cao, S. Shreedharan, and N. Atanasov, "Multi-robot object SLAM using distributed variational inference," *IEEE Robot. Automat. Lett.*, vol. 9, no. 10, pp. 8722–8729, Oct. 2024.

[7] A. Nijenhuis and H. S. Wilf, *Combinatorial Algorithms: For Computers and Calculators*. Amsterdam, the Netherlands: Elsevier, 2014.

[8] M. Huber and J. Law, "Fast approximation of the permanent for very dense problems," in *Proc. 19th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2008, pp. 681–689.

[9] R. L. Streit and T. E. Luginbuhl, "Maximum likelihood method for probabilistic multihypothesis tracking," in *Proc. SPIE*, vol. 2235, pp. 394–405, 1994.

[10] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Syst. Mag.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.

[11] P. Sun et al., "DanceTrack: Multi-object tracking in uniform appearance and diverse motion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 20993–21002.

[12] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 17–35.

[13] J. Luiten et al., "HOTS: A higher order metric for evaluating multi-object tracking," *Int. J. Comput. Vis.*, vol. 129, pp. 548–578, 2021.

[14] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*.

[15] P. Dendorfer et al., "MOT20: A benchmark for multi object tracking in crowded scenes," 2020, *arXiv:2003.09003*.

[16] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[17] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robot. Automat.*, vol. 17, no. 6, pp. 890–897, Dec. 2001.

[18] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.

[19] D. Musicki, R. Evans, and S. Stankovic, "Integrated probabilistic data association," *IEEE Trans. Automat. Control*, vol. 39, no. 6, pp. 1237–1241, Jun. 1994.

[20] D. Musicki and R. Evans, "Joint integrated probabilistic data association: JIPDA," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 3, pp. 1093–1099, Jul. 2004.

[21] H. A. Blom and E. A. Bloem, "Probabilistic data association avoiding track coalescence," *IEEE Trans. Automat. Control,*, vol. 45, no. 2, pp. 247–259, Feb. 2000.

[22] S. Yang, K. Thormann, and M. Baum, "Linear-time joint probabilistic data association for multiple extended object tracking," in *Proc. IEEE 10th Sensor Array Multichannel Signal Process. Workshop*, 2018, pp. 6–10.

[23] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3047–3055.

[24] F. Meyer et al., "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

[25] L. Rakai, H. Song, S. Sun, W. Zhang, and Y. Yang, "Data association in multiple object tracking: A survey of recent techniques," *Expert Syst. Appl.*, vol. 192, 2022, Art. no. 116300.

[26] W. J. Godinez and K. Rohr, "Tracking multiple particles in fluorescence time-lapse microscopy images via probabilistic data association," *IEEE Trans. Med. Imag.*, vol. 34, no. 2, pp. 415–432, Feb. 2015.

[27] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1722–1729.

[28] K. Doherty, D. Fourie, and J. Leonard, "Multimodal semantic SLAM with probabilistic data association," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 2419–2425.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Neural Inf. Process. Syst.*, 2015, vol. 28, pp. 91–99.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[31] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, no. 1/2, pp. 83–97, 1955.

[32] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEE Int. Conf. Image Process.*, 2017, pp. 3645–3649.

[33] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3038–3046.

[34] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 941–951.

[35] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 474–490.

[36] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.

[37] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6247–6257.

[38] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "TrackFormer: Multi-object tracking with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8844–8854.

[39] A. Vaswani et al., "Attention is all you need," in *Proc. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 5998–6008.

[40] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[41] C. Montella, "The Kalman filter and related algorithms: A literature review," in *Res. Gate*, pp. 1–17, 2011.

[42] S. Hiscocks et al., "Stone Soup: No longer just an appetiser," in *Proc. 26th Int. Conf. Inf. Fusion*, 2023, pp. 1–8.

[43] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, pp. 3069–3087, 2021.

[44] J. Pang et al., "Quasi-dense similarity learning for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 164–173.

[45] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "MOTR: End-to-end multiple-object tracking with transformer," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 659–675.

[46] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, "TransMOT: Spatial-temporal graph transformer for multiple object tracking," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 4870–4880.

[47] J. Cai et al., "MeMOT: Multi-object tracking with memory," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8090–8100.

[48] R. Jonker and T. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, 1987, pp. 325–340.

[49] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP J. Image Video Process.*, vol. 2008, 2008, Art. no. 246309.

[50] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12352–12361.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:04:07 UTC from IEEE Xplore. Restrictions apply.