Joint Probabilistic Data Association Revisited

Seyed Hamid Rezatofighi[1]   Anton Milan[1]   Zhen Zhang[2]   Qinfeng Shi[1]   Anthony Dick[1]   Ian Reid[1]

[1]School of Computer Science, The University of Adelaide, Australia

[2]School of Computer Science and Technology, Northwestern Polytechnical University, Xian, China

hamid.rezatofighi@adelaide.edu.au

Abstract

In this paper, we revisit the joint probabilistic data association (JPDA) technique and propose a novel solution based on recent developments in finding the m-best solutions to an integer linear program. The key advantage of this approach is that it makes JPDA computationally tractable in applications with high target and/or clutter density, such as spot tracking in fluorescence microscopy sequences and pedestrian tracking in surveillance footage. We also show that our JPDA algorithm embedded in a simple tracking framework is surprisingly competitive with state-of-the-art global tracking methods in these two applications, while needing considerably less processing time.

1. Introduction

Despite significant technical advances made in automated tracking of moving objects, multi-target tracking remains a challenging task. Within computer vision, applications of multi-target tracking are exemplified by the tasks of surveillance of a crowd of pedestrians [5, 25, 28, 29, 33, 44], and of tracking dense cellular and sub-cellular structures in biological sequences [11, 35] (Fig. 1). The main challenge in these applications is to estimate the state of an unknown and time-varying number of targets from a set of noisy and uncertain measurements. Targets often remain undetected due to occlusion, strong variation in appearance or other detector failures. Moreover, the observations generally include a set of spurious measurements (clutter) not originating from any target. Therefore, one of the crucial steps in the development of a reliable multi-target tracking systems is *data association*, which assigns the detected measurements to the existing targets in the presence of noise, clutter and detection uncertainty.

Joint probabilistic data association (JPDA) [16], is an elegant method of associating the detected measurements in each time frame with existing targets using a joint probabilistic score. Proposed in the early 1980s, it is widely accepted as a reliable data association technique and it has in-

Figure 1. Two sample frames from challenging multi-target tracking applications: Pedestrian tracking in a surveillance camera (left) and spot tracking in fluorescence microscopy (right).

fluenced a degree of the recent literature in the visual tracking community [11, 28, 29, 32, 35, 39, 42]. However, naive JPDA suffers from combinatorial complexity as it considers all possible assignments of measurements to targets to calculate the joint probabilistic score. Therefore, with an increasing number of targets and/or clutter, the technique is intractable in almost all practical applications without the use of heuristics such as gating. Even when gating is used, for any given gate size there will be a degree of target and/or clutter density that renders the method impractical. As a result, usually its application domain has been restricted to multi-target tracking scenarios with few, well separated targets.

In this paper, we revisit the JPDA formulation but address the issue of its complexity by leveraging the latest developments in finding the m-best solutions of an integer linear program. We propose a computationally tractable approximation to the original JPDA algorithm and show that it takes only a fraction of the time to compute without forfeiting performance. We demonstrate its applicability in practical applications with numerous targets and measurements such as fluorescence spot tracking in biological sequences and pedestrian tracking in crowded scenes. Moreover, we show that our JPDA algorithm, along with a simple tracking framework, can surprisingly perform on par, or even outperform state-of-the art multi-target tracking methods with a considerable gain in processing time.

We make the following contributions: (*i*) We reformulate the calculation of individual JPDA assignment scores

as a series of integer linear programs (ILPs) and approximate the joint score by the m-best solutions. This allows us to obtain an extremely accurate estimate of the complete JPDA by only considering a tiny fraction of its entire solution space. (*ii*) We propose a generic and highly efficient way to calculate the m-best solutions for any binary linear program by using a binary tree partition method. (*iii*) With a computationally tractable JPDA solution, we extend our implementation to multi-frame (MF) JPDA to increase robustness of data association. To the best of our knowledge, this is the first practical implementation of MF-JPDA for real-world applications. (*iv*) We show state-of-the-art performance on cell and pedestrian tracking using only a fraction of the computational time of previous methods.

2. Related Work

One of the earliest approaches for online tracking (or state estimation of a dynamic target) is the Kalman filter [21]. This recursive Bayesian filter computes the optimal state posterior when dealing with linear observation and transition models as well as Gaussian noise. In contrast, the particle filter [14] approximates the state density by a finite number of samples (or particles). Both methods are inherently designed to deal with one target only. To manage a scenario with multiple objects, typically a greedy or local assignment process is used to resolve data association [8].

The multi-hypothesis tracker (MHT) [11, 12, 34] is a more principled formulation for data association based on the Bayesian framework. It hypothesizes all possible data associations over time and uses measurements that arrive later in time to resolve ambiguities in the current frame. However, the complexity of the algorithm and the computational costs of this exhaustive search are considerable. In practice, heuristic pruning and merging techniques are usually combined with the MHT to restrict the exponentially growing number of hypotheses. In contrast to the (multi-frame) JPDA that maintains the contributions from all potential hypotheses from all tracks, the MHT prunes out invalid hypotheses for each track independently and deleted terms are completely discarded [7], making it impossible to recover from errors.

The joint probabilistic data association (JPDA) filter [16], which we review in more detail in the next section, is another approach for finding an optimal target-to-measurement assignment. Unfortunately, in its pure form, its computational complexity prohibits many real-world applications with a large number of targets. To alleviate the computational burden, different approximations of JPDA have been proposed. However, many of them use heuristic techniques and often sacrifice the tracking accuracy to make their algorithm computationally tractable [2, 38]. Oh *et al.* [31] proposed a more principled JPDA approximation based on Markov chain Monte Carlo (MCMC) data

association. While such sampling schemes offer a practical approach to approximating high-dimensional problems, they may suffer from poor mixing leading to slow convergence and the random elements can make reproducing experiments difficult.

Many of the most successful recent approaches in the vision literature [*e.g.* 5, 10, 24, 28, 32, 46] are so-called offline, or batch processing techniques and follow a rather different strategy from the ones described above. Typically, a sequence of frames is considered at once and the state and data association of all targets are inferred jointly by optimizing a predefined objective. The main differences between methods lie in the exact formulations of the objective and the trade-off between modeling accuracy and tractability.

Discretizing the state space and making simplifying assumptions about conditional dependences reduces the complexity of multi-target tracking and allows one to achieve the global optimum by LP-relaxation [20, 43], min-cost flow [10, 32, 46], or k-shortest paths algorithms [5]. Moving to a continuous state representation [28] or including more sophisticated terms, such as exclusion constraints [29], leads to more complex optimization problems that can only be solved to local optimality. Further examples that belong to that second class include quadratic boolean programming [24], generalized clique problems [45], maximum weight-independent set [9] and many more. While such methods show remarkable performance, the introduced delay in the output limits their applicability to offline applications in surveillance or video analysis.

In this work we revisit JPDA, a classical *online* approach, and demonstrate its power when combined with recent advances in optimization. Surprisingly, when combined with our novel principled approximation method, it is able to outperform many recent techniques while taking only a fraction of their time to process.

3. Joint Probabilistic Data Association

Let $x_t^1, ..., x_t^N$ and $z_t^1, ..., z_t^M$ be the states of all N targets and all M measurements at time t, respectively. The state vector x_t^j contains all relevant dynamic information about the jth target, *e.g.* its position and velocity, while the measurements contain what can be directly observed from the sequences, *e.g.* noisy and cluttered detected positions.

Let $p_t(d_i^j = 1)$, simply denoted by $p_t(d_i^j)$, be the assignment (or data association) probability representing that the measurement index $i \in [M]_0 \triangleq \{0, 1, ..., M\}$[1] is generated by target $j \in [N] \triangleq \{1, ..., N\}$ at time t. Here, 0 is a placeholder for a 'dummy' (or missed) detection. Under a linear Gaussian model, $p_t(d_i^j)$ is obtained as follows:

---

[1]For notational simplicity, we assume that all measurements can be assigned to all targets. However, if JPDA is followed by gating, only a subset of measurements can be assigned to each individual target.

$$p_t\left(d_i^j\right) \propto \begin{cases} (1 - p_D)\beta & \text{if } i = 0, \\ p_D \cdot \mathcal{N}(z_t^i; \hat{x}_t^j, \Sigma_S) & \text{otherwise,} \end{cases} \quad (1)$$

where $\hat{x}_t^j$ is the predicted position of the jth target at time t, p_D is the detection probability and β is the false detection (clutter) density. Here, $\mathcal{N}(\cdot)$ is the normal distribution and Σ_S is the innovation covariance matrix of the Kalman filter.

The joint probabilistic data association (JPDA) algorithm calculates a marginalized probability $q_t(d_i^j = 1)$, simply denoted by $q_t(d_i^j)$, on the joint data association space $\boldsymbol{\Theta}$. By definition, $\boldsymbol{\Theta}$ consists of *all* possible combinations of measurement-to-target assignments such that *(a)* each measurement (except for the dummy hypothesis $i = 0$) is assigned to at most one target, and *(b)* each target is uniquely assigned to a measurement. This space can be described by a set of binary vectors as follows:

$$\boldsymbol{\Theta} = \left\{ \theta = \left(d_i^j\right)_{i \in [M]_0, \, j \in [N]} \mid d_i^j \in \{0, 1\} \right. \quad (2)$$

$$\wedge \quad \sum_{j=1}^{N} d_i^j \leqslant 1, \quad \forall i \in [M] \quad (a)$$

$$\left. \wedge \quad \sum_{i=0}^{M} d_i^j = 1, \quad \forall j \in [N] \right\}, \quad (b)$$

where $|\boldsymbol{\Theta}| =: n_h$ is the total number of joint assignments and $\theta \in \boldsymbol{\Theta} \subseteq \mathbb{B}^{N \times (M+1)}$ is a binary vector representing one possible solution to the data association problem.

Let $\Theta_i^j \subset \boldsymbol{\Theta}$ be a subset that includes all hypotheses that assign the measurement i to target j such that $\Theta_i^j = \{\theta \in \boldsymbol{\Theta} \mid d_i^j = 1\}$. The JPDA probability $q_t(d_i^j)$ is calculated by marginalizing over this subset:

$$q_t\left(d_i^j\right) = \sum_{\theta \in \Theta_i^j} p(\theta), \quad (3)$$

where

$$p(\theta) = \prod_{\substack{\forall r \in [M]_0 \\ \forall k \in [N]}} \left(p_t\left(d_r^k\right)\right)^{d_r^k}. \quad (4)$$

Finally, all joint data association probabilities $q_t(d_i^j)_{i \in [M]_0}$ are normalized and used to update the jth target's state [16].

The accuracy of JPDA can be enhanced by taking the assignments in the subsequent frames into consideration (analogous to so-called Kalman smoothing vs. Kalman filtering). This extension, known as the JPDA-smoothing [26] or multi-frame JPDA (MF-JPDA) [37], conditions the probability $q_t(d_i^j)$ on both future *and* past information. However, this exacerbates the combinatorial explosion, and so has almost never been used in a practical application.

4. Our Solution

Even the traditional (single-frame) JPDA is often intractable because the sum over all possible combinations in
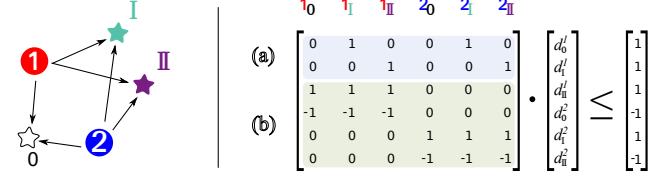

Figure 2. Rewriting the data association in Eq. (6) as an ILP. In this example, two targets (circles) and two measurements (I,II) and a dummy node (0) yield a 6×6 constraint matrix $\mathbf{A}$ that ensures that at most one incoming edge for each measurement (a) and that exactly one outgoing edge for each target (b) can be selected. Note that equality constraints (6b) are introduced by including negative entries so that both $\leqslant$ and $\geqslant$ constraints are enforced. The dummy node corresponds to a missed detection.

Eq. (3) involves a potentially huge number of terms. Our approach to address this is to approximate $q_t(\cdot)$ as the sum over the m highest probability hypotheses, which in most cases account for all but a tiny fraction of the total probability mass:

$$q_t\left(d_i^j\right) \approx \sum_{\theta \in \Delta_i^j} p(\theta). \quad (5)$$

Here, $\Delta_i^j = \{\theta \in \boldsymbol{\Theta}^m \mid d_i^j = 1\}$ and $\boldsymbol{\Theta}^m \subset \boldsymbol{\Theta}$ is a subset of the m most likely hypotheses ($|\boldsymbol{\Theta}^m| = m \ll n_h$). In other words, if we sorted all possible solutions in $\boldsymbol{\Theta}$ according to their association probability $q_t(\cdot)$, $\boldsymbol{\Theta}^m$ would contain only the top m entries that carry most of the probability mass.

We approach this in two stages. First, we reformulate the data association problem as an integer linear program (ILP). Solving this ILP will yield the best (*i.e.* maximum likelihood) data association. We then show how the second, third, *etc.* best solutions can be obtained successively in an efficient manner, yielding an approximation of $q_t(d_i^j)$ in Eq. (5) based on the m-best solutions of the problem $\max_{\theta \in \boldsymbol{\Theta}} p(\theta)$.

4.1. Data association as an integer linear program

We first rewrite the data association problem as a minimization:

$$C_1^* = \min_{\theta \in \boldsymbol{\Theta}} -\log(p(\theta)) \quad (6)$$

$$= \min \sum_{\substack{\forall r \in [M]_0 \\ \forall k \in [N]}} -\left(\log\left(p_t(d_r^k)\right) \cdot d_r^k\right)$$

$$\text{s. t.} \quad \sum_{k=1}^{N_t} d_r^k \leqslant 1 \quad \forall r \in [M] \quad (a)$$

$$\sum_{r=0}^{M_t} d_r^k = 1 \quad \forall k \in [N], \quad (b)$$

The constraints (6a), (6b) ensure that at most one target is associated with each measurement and exactly one measurement is associated with each target. The value of θ which attains the minimum value of (6) is the maximum

likelihood data association. It is easy to see that this problem can be reformulated exactly as an integer linear program (ILP) [40]:

$$C_1^* = \min_{\mathbf{y} \in \{0,1\}^n} C^T \mathbf{y} \qquad \text{s.t.} \quad \mathbf{A}\mathbf{y} \leqslant \mathbf{b}, \qquad (7)$$

where $\mathbf{y} = [y_1, \cdots, y_n]^T$ is a binary vector of length $n = N(M+1)$ such that $y_l = d_r^k$, and $C = [c_1, \cdots, c_n]^T$ is the cost vector with $c_l = -\log\left(p_t(d_r^k)\right)$. Fig. 2 illustrates the form of $\mathbf{A}$, $\mathbf{y}$ and $\mathbf{b}$ for a toy example.

5. Approximation by the m-best Solutions

Solving the ILP in Eq. (7) is straightforward using LP-relaxation[2]. However, recall that we seek, not just the best assignment, but the best m solutions to obtain an accurate approximation of the JPDA assignment probability. Let C_m^* denote the mth smallest objective value, and $\mathbf{y}^{(m)}$ the solution that attains this value:

$$C_m^* = C^T \mathbf{y}^{(m)}, \qquad (8a)$$

$$\mathbf{y}^{(1)} = \operatorname{argmin}_{\mathbf{y}} C^T \mathbf{y}, \text{ s. t. } \mathbf{A}\mathbf{y} \leqslant \mathbf{b}, \qquad (8b)$$

$$\mathbf{y}^{(m)} = \operatorname{argmin}_{\mathbf{y}} C^T \mathbf{y}, \text{ s. t. } \begin{cases} \mathbf{A}\mathbf{y} \leqslant \mathbf{b}, \\ \forall k < m : \mathbf{y} \neq \mathbf{y}^{(k)}. \end{cases} \qquad (8c)$$

The inequality constraints in Eq. (8c) can not be handled by general ILP solvers. However, since $\mathbf{y}$ is binary in our case, the constraints $\mathbf{y} \neq \mathbf{y}^{(k)}$ can be reformulated as

$$\langle \mathbf{y}, \mathbf{y}^{(k)} \rangle < \|\mathbf{y}^{(k)}\|_1, \qquad (9)$$

i.e. $\mathbf{y}$ differs from $\mathbf{y}^{(k)}$ in at least one bit[3]. As a result, a naive approach to find the m-best solutions suggests itself: for $k = 1, ..., m$: *(i)* solve an ILP using standard solvers such as [18] to obtain $\mathbf{y}^{(k)}$; *(ii)* add constraints (9) and repeat.

This kind of approach has been taken in some previous work for finding m-best solutions, [*e.g.* 3, 4]. However, the number of constraints grows with k. In the next section we present a much more efficient strategy that removes redundant constraints and inactive variables, thereby simplifying the problem with each kth-best iteration instead of aggravating it, yielding sub-linear increase in running time.

5.1. Binary Tree Partition method

Instead of iteratively adding new constraints, Fromer and Globerson [17] show that m-best problems can be solved by iteratively solving a series of constrained second-best problems. Solutions are found in order from $k = 1$ (best) to $k = m$. Finding the kth-best solution assumes that the feasible set $\mathcal{F}_k$ has been partitioned into $k-1$ disjoint sets $\mathcal{F}_k^1, ..., \mathcal{F}_k^{k-1}$. The kth solution $\mathbf{y}^{(k)}$ is found by searching

---

[2]The relaxation is tight because $\mathbf{A}$ is an assignment matrix (*cf.* [19]).
[3]In practice, $\langle \mathbf{y}, \mathbf{y}^{(k)} \rangle \leq \|\mathbf{y}^{(k)}\|_1 - 1$ is used instead.

Algorithm 1: Binary Tree Partition for JPDA

input : $C, \mathbf{A}, \mathbf{b}, m$
output: $\mathbf{y}^{(k)}, k = 1, \ldots, m$

1   $\mathbf{y}^{(1)} = \operatorname{argmin}_{\mathbf{y}} C^T \mathbf{y}$ s. t. $\mathbf{A}\mathbf{y} \leqslant \mathbf{b}$;
2   $\mathbf{y}^{(2)} = \operatorname{argmin}_{\mathbf{y}} C^T \mathbf{y}$ s. t. $\mathbf{A}\mathbf{y} \leqslant \mathbf{b}, \langle \mathbf{y}^{(1)}, \mathbf{y} \rangle < \|\mathbf{y}^{(1)}\|_1$;
3   Select arbitrary $j \in \{i | y_i^{(1)} \neq y_i^{(2)}\}$;
4   $\mathcal{F}_3^1 = \{\mathbf{y} \in \mathbb{B}^n | \mathbf{A}\mathbf{y} \leqslant \mathbf{b}, \langle \mathbf{y}, \mathbf{y}^{(1)} \rangle < \|\mathbf{y}^{(1)}\|_1, y_j = y_j^{(1)}\}$;
5   $\mathcal{F}_3^2 = \{\mathbf{y} \in \mathbb{B}^n | \mathbf{A}\mathbf{y} \leqslant \mathbf{b}, \langle \mathbf{y}, \mathbf{y}^{(2)} \rangle < \|\mathbf{y}^{(2)}\|_1, y_j = y_j^{(2)}\}$;
6   $\mathbf{y}_3^1 = \operatorname{argmin}_{\mathbf{y} \in \mathcal{F}_3^1} C^T \mathbf{y}$;
7   $\mathbf{y}_3^2 = \operatorname{argmin}_{\mathbf{y} \in \mathcal{F}_3^2} C^T \mathbf{y}$;
8   **for** $k = 3, \ldots, m$ **do**
9     $l_k = \operatorname{argmin}_l C^T \mathbf{y}_k^l$, $\mathbf{y}^{(k)} = \mathbf{y}_k^{l_k}$;
10    $\mathcal{F}_{k+1}^l = \mathcal{F}_k^l, y_{k+1}^l = y_k^l, \forall l < k, l \neq l_k$;
11   Select arbitrary $j_k \in \{i | y_i^{(l_k)} \neq y_i^{(l)}\}$;
12    $\mathcal{F}_{k+1}^{l_k} = \mathcal{F}_k^{l_k} \cap \{\mathbf{y} | \langle \mathbf{y}^{(k)}, \mathbf{y} \rangle < \|\mathbf{y}^{(l_k)}\|_1, y_{j_k} = y_{j_k}^{(l_k)}\}$;
13    $\mathcal{F}_{k+1}^k = \mathcal{F}_k^k \cap \{\mathbf{y} | \langle \mathbf{y}^{(k)}, \mathbf{y} \rangle < \|\mathbf{y}^{(k)}\|_1, y_{j_k} = y_{j_k}^{(k)}\}$;
14   Remove constraints $\langle \mathbf{y}^{(k)}, \mathbf{y} \rangle < \|\mathbf{y}^{(k)}\|_1$ from $\mathcal{F}_{k+1}^{l_k}$;
15   Remove constraints $\langle \mathbf{y}^{(l_k)}, \mathbf{y} \rangle < \|\mathbf{y}^{(l_k)}\|_1$ from $\mathcal{F}_{k+1}^k$;
16    $\mathbf{y}_{k+1}^l = \operatorname{argmin}_{\mathbf{y} \in \mathcal{F}_{k+1}^l} C^T \mathbf{y}, l \in \{l_k, k\}$;

---

all sets. To proceed to the $(k+1)^{\text{th}}$ solution they add the constraint $\mathbf{y} \neq \mathbf{y}^{(k)}$; however this is redundant for all sets except for $\mathcal{F}_k^{l_k}$, which contains $\mathbf{y}^{(k)}$. Therefore, the previous solutions can be retained for all sets except this one, which is partitioned into two disjoint sets. The optimal value of the objective over these two sets can then be obtained via solving two second-best problems. This results in k disjoint sets, whose union is the feasible set of $(k+1)$-best problem, and the process is repeated.

As [17] is designed for multi-label integer programming (IP), they need a specific IP solver that can handle constraints like $\mathbf{y} \neq \mathbf{y}^{(k)}$ and $y_i \neq y_i^{(k)}$. However, in our case the variables are binary, so we note that 1) the constraint $\mathbf{y} \neq \mathbf{y}^{(k)}$ is redundant in all sets except that containing $\mathbf{y}^{(k)}$, and 2) $y_i \neq y_i^{(k)}$ fixes the value of y_i as $y_i = 1$ or $y_i = 0$. This is an assignment rather than a constraint and therefore reduces the dimensionality of $\mathbf{y}$ by 1. Combining these observations leads to our streamlined form of Fromer and Globerson's approach, summarized in Alg. 1.

5.2. Calculation of m

In the previous sections we presented an efficient way to obtain the m-best solutions of an ILP with the assumption that m is known. However, we want to calculate m such that the probability mass error between the approximated and exact JPDA scores for all target-to-measurements assignments is less than a small threshold ϵ:

$$\mathcal{E} = \sum_{\theta \subseteq \Theta} p(\theta) - \sum_{\theta \subseteq \Theta^m} p(\theta) < \epsilon. \qquad (10)$$
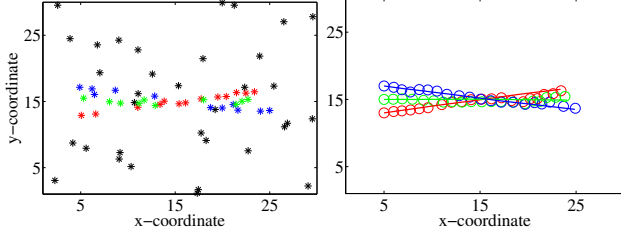
Figure 3. Left: Noisy and cluttered detections. Right: Ground truth trajectories (solid lines) versus the tracking results (circle markers) using 3F-JPDA$_{100}$.

It can be proved[4] that a tight upper bound for this error is $\mathcal{E} \leqslant (n_h - m) \exp(-C_m^*)$. Therefore, for any case, m can be automatically calculated such that this tight upper bound error is less ϵ.

6. Experimental Results

6.1. Evaluation on simulations

To evaluate the speed and accuracy of our m-best JPDA tracker, we first apply it on a simulated scenario with three moving targets crossing each other (Fig. 3). Each target's state is given by its position and velocity $x_t = (x_p, \dot{x}_v, y_p, \dot{y}_v)$ and the motion is modeled by the discrete update equation $x_t = \mathbf{F}x_{t-1} + \eta$, where $\mathbf{F} = \mathrm{diag}[\mathbf{F}_1, \mathbf{F}_1]$ is a constant velocity model and η is Gaussian noise with covariance $\mathbf{Q} = \mathrm{diag}[\mathbf{Q}_1, \mathbf{Q}_1]$ representing unmodeled acceleration. $\mathbf{F}$ and $\mathbf{Q}$ take their textbook values as in:

$$\mathbf{F}_1 = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q}_1 = q_d \begin{bmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{bmatrix}, \quad (11)$$

where $\tau = 1$ is the sampling period and $q_d = 0.02$ is the process noise parameter. Both noisy and spurious detection points $z_t = (\hat{x}_p, \hat{y}_p)$ were generated according to a detection probability $p_D = 0.7$ with added zero-mean Gaussian noise with covariance $q_m = 0.1$ and a uniform clutter density β with a false positive rate $\lambda = 3$. To simulate long term occlusion, when multiple targets come very close to one another (distance less than 1), only one of them is detected and the others are missed (see Fig. 3 left). In the following, we report averaged results over 100 Monte Carlo experiments. We evaluate based on observations from individual frames (JPDA, JPDA$_m$) and by including neighboring frames (3F-JPDA, 3F-JPDA$_m$).

Fig. 4 (left) represents the probability mass approximation error $\mathcal{E}$ from Eq. (10), which is introduced by our approximation. As expected, the error decreases exponentially with growing m. Empirically, the error reaches less than 1% after $m \geqslant 30$ best solutions. This figure also shows how the averaged processing time (green line) increases sub-linearly with m. In this experiment, our 3F-JPDA$_m$
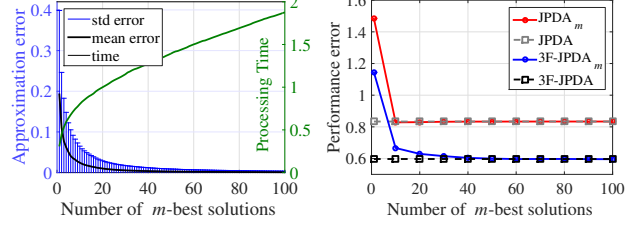
---

[4]Proof provided in the supplementary files.

Figure 4. Left: Probability mass approximation error $\mathcal{E}$ (*cf.* Eq. 10) and processing time in seconds for 3F-JPDA. Right: Tracking error measured as the OSPA-T location error versus the number of m-best solutions for JPDA$_m$ and 3F-JPDA$_m$. In both cases the solution converges to the minimum error for small m.

requires less than 2 seconds for any value of $m \leqslant 100$ whilst 3F-JPDA takes on average 57 seconds per simulation run to complete. To show how the approximation error affects the tracking results, Fig. 4 (right) depicts the location error of the OSPA-T metric [36], representing both track accuracy and label switching as a single number, versus the value of m. This confirms our claim from Eq. 5 that by selecting the best few solutions, we can reach the same accuracy as JPDA and 3F-JPDA, but with considerably lower processing time. Note that we deliberately designed a computationally tractable scenario for 3F-JPDA and compared with our 3F-JPDA$_m$ over 100 experiments. The difference in processing time can be significantly larger in real-world applications, as we will see in the following section.

6.2. Evaluation on real-world data

Implementation details. The core JPDA algorithm does not include a mechanism to deal with a time-varying number of targets. A principled extension of JPDA, known as integrated JPDA (IJPDA) [13], has been proposed for that purpose. However, IJPDA adds a considerable computational complexity to the JPDA algorithm. For simplicity, we use a heuristic initiation and termination scheme proposed in [35]: *(i)* any detection that is not claimed by an existing target is initiated as a new target; *(ii)* a track is terminated if the number of consecutive missed detection assignments reaches a specified threshold T_d. In this latter case the estimated states for this track are deleted from the frame where the missed detection first occurred. In addition, all tracks with a life span shorter than a threshold L_S are removed. We will show that even with this simple scheme, we can perform as well as, or even better than, the state-of-the art methods in real-world applications using *only* a set of sparse detections and a simple dynamic model.

For practical reasons we make use of a gating procedure that excludes the set of detections whose Mahalanobis distance exceeds a predefined threshold d_G. As noted in Sec. 1, gating can make JPDA tractable in cases where the number of interacting targets and the measurements inside their gates are small. Thus we only use our approximation when

the estimated number of possible assignments exceeds a threshold. Since the total number of hypotheses n_h cannot be accurately predicted when JPDA is followed by gating, we cannot make direct use of the calculation in Eq. (10). Therefore, we fix $m = 100$ throughout all our experiments. In our experience this value is enough to reach the same tracking accuracy as complete JPDA.

Complexity and runtime. The time complexity to find the kth-best solution of an ILP for the naive approach (see Sec. 5) is $\mathcal{O}((A+k)^{1.5}B^2)$, where $B = MN$ and $A = M + N$ [30]. This becomes computationally prohibitive for large k. In contrast, our proposed approach takes $\mathcal{O}(A^{1.5}(B - D_k)^2)$ time, where D_k is a monotonic increasing function of k. This yields a sub-linear increase in running time as k increases (*cf.* Sec. 6.1, Fig. 4).

Our code was implemented using MATLAB and was run on a desktop PC (Intel Core $i7 - 4790$, 3.60 GHz CPU, 16 GB RAM), making use of the Gurobi ILP solver (version 5.6.3, 64bit). We report the average processing time per frame of the tracking methods in the following experiments.

Evaluation performance measures. To evaluate the performance of the tracking methods in the fluorescence spot tracking application, we employ the same Optimal Sub-Pattern Assignment metric for tracks (OSPA-T), used in [11] and [35]. This error metric can be seen as the sum of cardinality and location errors. The cardinality error can be interpreted as errors related to missed or false tracks while the location error combines both track accuracy and the labeling (or mismatch) errors.

For quantitative comparison with previous pedestrian tracking methods and for consistency with their evaluation scheme, we used the popular CLEAR MOT performance measures [6]. The multi-object tracking accuracy (MOTA) combines errors such as false positives (FP), false negatives (FN) and identity switches (IDs) into a single number. The multi-object tracking precision (MOTP) measures the localization accuracy of trajectories. Mostly lost (ML) and mostly tracked (MT) scores [25] respectively represent how many targets are tracked for less than 20% and more than 80% of their life span based on ground truth trajectories (GT). We also report the tracking recall and precision.

6.2.1 Fluorescence spot tracking

We first apply our proposed JPDA$_m$ and 3F-JPDA$_m$ in a challenging biological application: tracking numerous subcellular structures in fluorescence microscopy sequences. These structures are seen as small moving bright spots that can appear or disappear from the field of view or be occluded by other structures. Our sequences include 300 frames and comprise a time-varying number of targets (on average ≈ 204 spots per frame) moving inside a cell membrane with an effective region $\approx 230 \times 230$ pixels.

Method	Location↓ (Pixel)	Cardinality↓ (Pixel)	OSPA-T↓ (Pixel)	Time↓ (Sec.)
MHT [11]	5.38	**1.94**	7.32	0.23
JPDA [35]	2.14	4.06	6.20	2.38
JPDA$_{100}$	2.14	4.06	6.20	**0.20**
3F-JPDA$_{100}$	**1.94**	3.22	**5.16**	3.13

Table 1. The averaged location, cardinality and OSPA-T errors and processing time per frame of the spot trackers. JPDA$_{100}$ matches the error of JPDA with lower computation time, while 3F-JPDA$_{100}$ reduces overall error at higher computational cost.

We compare the results of our proposed single frame JPDA (JPDA$_{100}$) and three frames JPDA (3F-JPDA$_{100}$) on these sequences against two state-of-the-art spot tracking methods: IMM-JPDA [35] and MHT [11]. To evaluate the performance of all algorithms fairly, the same detections were provided for all competing tracking methods using the spot detector proposed in [11]. Moreover, since a single motion model (constant velocity) describes the dynamic behavior of our structures in this application sufficiently well, a single motion model implementation of all tracking methods was used[5]. All parameters for all methods were either directly estimated or tuned manually on a training sequence (a 30 frame movie). We used the same value for all parameters that are in common between the methods: detection probability $p_D = 0.77$; clutter rate $\lambda = 1$; gate size $d_G = 4$; dynamic noise $q_d = 0.1$; and measurement noise $q_m = 1$. For all JPDA techniques including the IMM-JPDA method, the termination and track deletion parameters were set as $T_d = 8$ and $L_S = 2$. Note that increasing the depth or the gate size for MHT does not yield noticeable performance improvement, but significantly slows down the computation.

In Tab. 1, the average processing time[6] and the tracking results for all aforementioned methods are reported. According to the OSPA-T value, all JPDA algorithms track more accurately than MHT in this application. However, since we used a heuristic for track initiation and termination, all JPDA algorithms have higher cardinality errors due to higher numbers of false tracks compared to MHT, which has a principled way for target initiation and termination. The overall performance superiority of JPDA is mainly due to its reliability of dealing with long occlusion and complex data association, which are frequent in this application.

As expected, our JPDA$_{100}$ performs as accurately as JPDA [35], but more than 10 times faster on average. This faster performance is mainly due to a few frames involving many interacting targets, which take around 640 seconds for JPDA while our JPDA$_m$ requires only 1.5 seconds. In this application, 3F-JPDA is computationally intractable; based on our knowledge of its cost relative to the JPDA, we estimate that it would take several weeks to complete on this

---

[5]Therefore, we abbreviate IMM-JPDA [35] as JPDA in Table 1.

[6]The average processing time for MHT is reported based on a Java implementation, available on http://icy.bioimageanalysis.org.

Figure 5. Our tracking results on a fluorescence microscopy sequence (top row, cropped for better readability) using 3F-JPDA$_{100}$ and PETS S2.L2 (second row) and TUD-Stadtmitte (bottom) using JPDA$_{100}$.

dataset. However, 3F-JPDA$_{100}$ is highly efficient and has overall a lower OSPA-T value compared to other methods.

6.2.2 Pedestrian tracking

We demonstrate the performance of our method for visual tracking in surveillance by evaluating our results on the popular PETS 2009 video sequences [15], in particular those with high target density: S1.L1-2, S1.L2-1, S2.L2 and S2.L3. We also include the sequence (TUD-Stadtmitte) [1] with a very different setup captured from a low angle. In all videos, tracking is performed in image coordinates using publicly available detections provided by [28] as input. As discussed above, we use a constant velocity model for tracking pedestrians in image space. Empirically, we find our results are not sensitive to the exact parameter settings. To achieve best performance, we manually tune them on a different PETS sequence (S1.L2-2), and then fix them for all test sequences at $p_D = 0.89$, $\lambda = 3$, $d_G = 5.48$, $q_d = 0.5$, $q_m = 7$, $T_d = 45$ and $L_S = 15$.

In Tab. 2, we compare our results against several state-of-the art methods applied on the same sequences. Previous figures are taken from [28, 29, 41] and the same evaluation script is used to quantify our results. All metrics are computed in 3D with a 1m hit/miss threshold. For a meaningful comparison to other methods, we present the results for the entire image, and for a cropped tracking area for each sequence. Since the number of occlusions and crossing targets in this application is significantly lower than in fluores-

cence spot tracking, 3F-JPDA$_{100}$ does not yield a noticeable improvement compared to JPDA$_{100}$. Moreover, full JPDA has exactly the same results as JPDA$_{100}$, but requires higher processing time. Therefore, we only report the results of JPDA$_{100}$ in this setting.

The performance measures in Tab. 2 indicate that the results of JPDA$_{100}$ produce an increased number of false tracks (higher FP) compared to the other methods. As discussed, this is mainly due to the heuristic track initiation and termination used. Nevertheless, we can still outperform state-of-the-art methods w.r.t. MOTA in most sequences. The main reason is JPDA's ability to robustly maintain targets' identities through long occlusions resulting in higher MT and lower FN and IDs. The processing time of JPDA$_{100}$ is between 0.001 and 0.046 seconds per frame, easily enabling its use in real-time applications.

MOTChallenge. In addition to the above experiments, we also present our results[7] on *MOTChallenge*, a recent multi-target tracking benchmark [23], featuring a number of sequences with substantially varying properties, such as the number of targets present, camera motion, target density, *etc*. Tab. 2 (bottom) shows our performance along with the top three competitors with available corresponding publications at the time of submission. Although we only rely on the provided detections and a simple dynamic model, our approach shows very competitive performance, while being one to two orders of magnitude faster. We achieve the over-

---

[7]http://motchallenge.net/results/2D_MOT_2015/

Dataset (Sequence)	Method	MOTA %↑	MOTP %↑	GT	MT ↑	ML ↑	FP ↑	FN ↑	IDs ↓	Recall %↑	Prec. %↑
PETS (S1.L1-2)	Pirsiavash et al. [32]	45.4	66.8	36	9	14	6	1367	38	47.1	99.5
	Berclaz et al. [5]	51.5	64.8	36	16	14	98	1151	4	55.5	93.6
	Wen et al. [41]	57.1	54.8	36	18	8	34	1071	4	58.6	97.8
	Milan et al. [28]	57.9	59.7	36	19	11	148	918	21	64.5	91.8
	JPDA₁₀₀	**70.0**	64.8	36	**21**	**5**	108	**658**	10	**74.5**	94.7
	Milan et al. [29]	60.0	61.9	44	**21**	11	169	1349	22	64.9	93.7
	JPDA₁₀₀	**63.5**	**64.5**	44	17	**9**	**112**	**1279**	**13**	**66.7**	95.8
PETS (S1.L2-1)	Berclaz et al. [5]	19.5	60.6	43	4	29	64	2950	7	21.4	92.6
	Milan et al. [28]	30.8	49.0	43	7	20	227	2308	61	38.5	86.4
	JPDA₁₀₀	**32.8**	59.8	43	9	20	230	**2238**	52	**40.3**	86.8
	Milan et al. [29]	29.6	58.8	42	2	21	27	3494	42	30.9	98.3
	JPDA₁₀₀	**32.8**	57.6	42	5	15	218	**3108**	76	**38.6**	89.9
PETS (S2.L2)	Pirsiavash et al. [32]	45.0	64.1	74	7	17	199	4257	137	49.0	95.4
	Berclaz et al. [5]	24.2	60.9	74	7	40	193	6117	22	26.8	92.1
	Wen et al. [41]	62.1	52.7	74	27	3	640	2402	125	71.2	90.3
	Milan et al. [28]	56.9	59.4	74	28	12	622	2881	99	65.5	89.8
	JPDA₁₀₀	58.3	59.3	74	22	6	910	2468	103	70.5	86.6
	Milan et al. [29]	58.1	59.8	43	11	1	549	3592	167	65.1	92.4
	JPDA₁₀₀	58.2	58.5	43	11	0	1051	**3108**	143	69.8	87.2
PETS (S2.L3)	Pirsiavash et al. [32]	43.0	63.0	44	5	18	46	1760	52	46.0	97.0
	Berclaz et al. [5]	28.8	61.8	44	5	31	45	2269	7	30.4	95.7
	Wen et al. [41]	55.3	53.2	44	12	9	149	1272	36	61.0	93.0
	Milan et al. [28]	45.4	64.6	44	9	18	169	1572	38	51.8	90.9
	JPDA₁₀₀	53.9	61.6	44	15	17	162	1320	20	59.5	92.3
	Milan et al. [29]	39.8	65.0	44	8	19	115	2493	27	43	94.2
	JPDA₁₀₀	48.0	62.3	44	13	18	161	2092	23	52.2	93.4
TUD (Stadtmitte)	Berclaz et al. [5]	45.8	56.7	9	1	1	117	261	5	63.1	79.2
	Milan et al. [28]	71.1	65.5	9	7	0	92	108	4	84.7	86.7
	JPDA₁₀₀	57.9	60.0	9	4	1	120	172	6	75.7	81.7
	Milan et al. [29]	56.2	61.6	10	4	0	134	357	15	69.1	85.6
	JPDA₁₀₀	58.9	59.8	10	4	1	116	349	10	69.8	87.4
2D MOT Challenge (Benchmark)	CEM [28] (1.1 fps)	19.3	70.7	721	8.5	46.5	14180	34591	813	43.7	65.4
	SegTrack [27] (0.2 fps)	22.5	71.7	721	5.8	63.9	7890	39020	697	36.5	74.0
	MotiCon [22] (1.4 fps)	23.1	70.9	721	4.7	52.0	10404	35844	1018	41.7	71.1
	JPDA₁₀₀ (32.6 fps)	23.8	68.2	721	5.0	58.1	6373	40084	365	34.8	77.0

Table 2. Quantitative comparison results of our JPDA₁₀₀ with other state-of-the-art trackers on the pedestrian datasets. The red and blue colors indicate the **best** and the second best performing tracker on each metric. For each sequence, results above the line are for a cropped tracking region, while below the line use the entire frame.

all lowest number of ID switches, which once again confirms the power of joint data association. Please note that the JPDA algorithm solves the data association problem in an online manner, whereas the closest previous approaches belong to the class of batch processing techniques.

7. Conclusion

In this paper, we revisited the JPDA algorithm and proposed an efficient and accurate approximation. We demonstrated the validity of our approach on two challenging multi-target tracking applications with noisy detections and substantial occlusion. In spite of the heuristic nature of the track initiation scheme, we showed that JPDA performs on par or even better than state-of-the-art methods in molecular applications and pedestrian tracking.

Our future work will explore more general applications of this approach. JPDA is just one example of an association/matching method used in computer vision, and we believe that our method can also be used to increase the scale and speed at which other such methods can be applied.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Monocular 3D pose estimation and tracking by detection. In *CVPR 2010*.

[2] Y. Bar-Shalom. Multitarget-multisensor tracking: advanced applications. *Norwood, MA, Artech House*, 1, 1990.

[3] D. Batra. An efficient message-passing algorithm for the M-best MAP problem. *arXiv:1210.4841*, 2012.

[4] D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich. Diverse M-best solutions in Markov random fields. In *ECCV 2012*, pages 1–16, 2012.

[5] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *PAMI*, 33(9):1806–1819, Sept. 2011.

[6] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *Image and Video Processing*, 2008(1):1–10, May 2008.

[7] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[8] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV 2009*.

[9] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR 2011*.

[10] A. A. Butt and R. T. Collins. Multi-target tracking by Lagrangian relaxation to min-cost network flow. In *CVPR 2013*.

[11] N. Chenouard, I. Bloch, and J.-C. Olivo-Marin. Multiple hypothesis tracking for cluttered biological image sequences. *PAMI*, 35(11):2736–2750, 2013.

[12] I. J. Cox and S. L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *PAMI*, 18(2), 1996.

[13] J. Dezert, N. Li, and X.-R. Li. Theoretical development of an integrated JPDAF for multitarget tracking in clutter. In *Proc. Workshop ISIS-GDR/NUWC, ENST*, 1998.

[14] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[15] J. Ferryman and A. Shahrokni. PETS2009: Dataset and challenge. In *Winter-PETS*, 2009.

[16] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J. Oceanic Eng.*, 8(3):173–184, 1983.

[17] M. Fromer and A. Globerson. An LP view of the M-best MAP problem. *NIPS*, 22:567–575, 2009.

[18] Gurobi, Inc. Gurobi optimizer reference manual, 2015.

[19] I. Heller and C. B. Tompkins. An extension of a theorem of dantzig. *Annals of Mathematics Studies.*, 38(1), 1956.

[20] H. Jiang, S. Fels, and J. J. Little. A linear programming approach for multiple object tracking. In *CVPR 2007*.

[21] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[22] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *CVPR 2014*.

[23] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, Apr. 2015.

[24] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV 2007*.

[25] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR 2009*.

[26] A. Mahalanabis, B. Zhou, and N. Bose. Improved multi-target tracking in clutter by PDA smoothing. *IEEE Trans. Aerosp. Electron. Syst.*, 26(1):113–121, 1990.

[27] A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid. Joint tracking and segmentation of multiple targets. In *CVPR 2015*.

[28] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *PAMI*, 36(1):58–72, 2014.

[29] A. Milan, K. Schindler, and S. Roth. Detection- and trajectory-level exclusion in multiple object tracking. In *CVPR 2013*.

[30] A. Nemirovski. Interior point polynomial time methods in convex programming. *Lecture Notes*, 2004.

[31] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for multi-target tracking. *IEEE Trans. Autom. Control*, 54(3):481–497, 2009.

[32] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*.

[33] H. Possegger, T. Mauthner, P. Roth, and H. Bischof. Occlusion geodesics for online multi-object tracking. In *CVPR'14*.

[34] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control*, 24(6):843–854, 1979.

[35] S. H. Rezatofighi, S. Gould, R. Hartley, K. Mele, and W. E. Hughes. Application of the IMM-JPDA filter to multiple target tracking in total internal reflection fluorescence microscopy images. In *MICCAI*, pages 357–364, 2012.

[36] B. Ristic, B. Vo, D. Clark, and B. Vo. A metric for performance evaluation of multi-target tracking algorithms. *IEEE Trans. Signal Process.*, 59(7):3452–3457, 2011.

[37] J. Roecker. Multiple scan joint probabilistic data association. *IEEE Trans. Aerosp. Electron. Syst.*, 31(3):1204–1210, 1995.

[38] J. Roecker and G. Phillis. Suboptimal joint probabilistic data association. *IEEE Trans. Autom. Control*, 29(2), 1993.

[39] A. V. Segal and I. Reid. Latent data association: Bayesian model selection for multi-target tracking. In *ICCV 2013*.

[40] P. Storms and F. Spieksma. An LP-based algorithm for the data association problem in multitarget tracking. In *Third International Conference on Information Fusion*, July 2000.

[41] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *CVPR 2014*.

[42] Z. Wu, T. H. Kunz, and M. Betke. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *CVPR 2011*.

[43] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *CVPR 2012*.

[44] Z. Wu, J. Zhang, and M. Betke. Online motion agreement tracking. In *BMVC 2013*.

[45] A. R. Zamir, A. Dehghan, and M. Shah. GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV 2012*, volume 2, pages 343–356.

[46] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR 2008*.