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\cite{5,25,28,29,33,44}, and of tracking dense cellular and sub-cellular structures in biological sequences\cite{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)\cite{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 influenced a degree of the recent literature in the visual tracking community\cite{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 lin-
ear 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 know-
edge, 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 fil-
ter [21]. This recursive Bayesian filter computes the opti-
mal 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 inher-
etly 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. How-
ever, the complexity of the algorithm and the computa-
tional costs of this exhaustive search are considerable. In
practice, heuristic pruning and merging techniques are usu-
ally 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 po-
tential hypotheses from all tracks, the MHT prunes out in-
valid 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) fil-
ter [16], which we review in more detail in the next sec-
tion, is another approach for finding an optimal target-to-
measurement assignment. Unfortunately, in its pure form,
its computational complexity prohibits many real-world ap-
lications with a large number of targets. To alleviate the
computational burden, different approximations of JPDA
have been proposed. However, many of them use heuris-
tic techniques and often sacrifice the tracking accuracy to
make their algorithm computationally tractable [2, 38]. Oh
et al. [31] proposed a more principled JPDA approxima-
tion based on Markov chain Monte Carlo (MCMC) data
association. While such sampling schemes offer a practi-
cal approach to approximating high-dimensional problems,
they may suffer from poor mixing leading to slow conver-
gence and the random elements can make reproducing ex-
periments 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 optimiz-
ing 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 as-
sumptions about conditional dependences reduces the com-
plexity 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 in-
cluding more sophisticated terms, such as exclusion con-
straints [29], leads to more complex optimization prob-
lems 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 applicabil-
ity to offline applications in surveillance or video anal-
ysis.

In this work we revisit JPDA, a classical online
approach, and demonstrate its power when combined with
recent advances in optimization. Surprisingly, when com-
bined 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 \( x_{i1}^1, ..., x_{iN}^1 \) and \( z_{i1}^1, ..., z_{iM}^1 \) be the states of all \( N \) tar-
gets and all \( M \) measurements at time \( t \), respectively.
The state vector \( x_{it} \) contains all relevant dynamic information
about the \( j \)-th 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 \( p_i(d_{it}^j = 1) \), simply denoted by \( p_{ij}(d_{it}^j) \), be the assign-
ment (or data association) probability representing that the
measurement index \( i \in [M] \) \( \triangleq \{0, 1, ..., M\} \) is generated
by target \( j \in [N] \) \( \triangleq \{1, ..., N\} \) at time \( t \). Here, 0 is a place-
holder for a ‘dummy’ (or missed) detection. Under a linear
Gaussian model, \( p_{ij}(d_{it}^j) \) is obtained as follows:

\[ p_{ij}(d_{it}^j) = \]
\[ p_t(d_i^j) \propto \begin{cases} (1 - p_D) \beta^i & \text{if } i = 0, \\ p_D \cdot N(z_i^j; \hat{x}_i^j, \Sigma_S) & \text{otherwise,} \end{cases} \tag{1} \]

where \( \hat{x}_i^j \) is the predicted position of the \( j \)-th target at time \( t \), \( p_D \) is the detection probability and \( \beta \) is the false detection (clutter) density. Here, \( N(\cdot) \) is the normal distribution and \( \Sigma_S \) 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 \( \Theta \). By definition, \( \Theta \) 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:

\[
\Theta = \left\{ \theta = (d_i^j)_{i \in [M], j \in [N]} \left| d_i^j \in \{0, 1\} \right. \right\} 
\setminus \left\{ \sum_{j=1}^{N} d_i^j \leq 1, \quad \forall i \in [M] \right\} \quad \text{(a)} \\
\setminus \left\{ \sum_{i=0}^{M} d_i^j = 1, \quad \forall j \in [N] \right\} \quad \text{(b)}
\]

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

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

\[ q_t(d_i^j) = \sum_{\theta \in \Theta_i^j} p(\theta), \tag{3} \]

where

\[ p(\theta) = \prod_{r \in [M], k \in [N]} \left( p_t(d_k^r) \right)^{d_k^r}. \tag{4} \]

Finally, all joint data association probabilities \( q_t(d_i^j)_{i \in [M]} \) are normalized and used to update the \( j \)-th 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

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure2.png}
\caption{Rewriting the data association in Eq. (6) as an ILP. In this example, two targets (circles) and two measurements (II, II) and a dummy node \( 0 \) yield a \( 6 \times 6 \) constraint matrix \( A \) 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 \( \leq \) and \( \geq \) constraints are enforced. The dummy node corresponds to a missed detection.}
\end{figure}

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:

\[ q_t(d_i^j) \approx \sum_{\theta \in \Delta_i^j} p(\theta). \tag{5} \]

Here, \( \Delta_i^j = \{ \theta \in \Theta^m \mid d_i^j = 1 \} \) and \( \Theta^m \subset \Theta \) is a subset of the \( m \) most likely hypotheses (\( |\Theta^m| = m < n_h \)). In other words, if we sorted all possible solutions in \( \Theta \) according to their association probability \( q_t(\cdot) \), \( \Theta^m \) 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 \( \max_{\theta \in \Theta} p(\theta) \).

4.1. Data association as an integer linear program

We first rewrite the data association problem as a minimization:

\[
C^*_1 = \min_{\theta \in \Theta} - \log (p(\theta)) = \min_{\theta \in \Theta} \sum_{r \in [M], k \in [N]} - \left( \log (p_t(d_k^r)) \right) \cdot d_k^r \\
\text{s. t.} \quad \sum_{k=1}^{N} d_k^r \leq 1 \quad \forall r \in [M] \quad \text{(a)} \\
\sum_{r=0}^{M} d_k^r = 1 \quad \forall k \in [N], \quad \text{(b)}
\]

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 \( \theta \) 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]:

\[ C_1^* = \min_{y \in \{0,1\}^n} C^T y \quad \text{s.t.} \quad Ay \leq b, \quad (7) \]

where \( y = [y_1, \ldots, y_n]^T \) is a binary vector of length \( n = N(M + 1) \) such that \( y_i = d_i^k \), and \( C = [c_1, \ldots, c_n]^T \) is the cost vector with \( c_l = -\log(p_l(d_i^k)) \). Fig. 2 illustrates the form of \( A, y \) and \( b \) 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 \( m \)-th smallest objective value, and \( y^{(m)} \) the solution that attains this value:

\[ C_m^* = C^T y^{(m)}, \]

\[ y^{(1)} = \arg\min_y C^T y \text{ s.t. } Ay \leq b, \]

\[ y^{(m)} = \arg\min_y C^T y \text{ s.t. } \{Ay \leq b, \forall k < m : y \neq y^{(k)}\}. \quad (8c) \]

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

\[ \langle y, y^{(k)} \rangle < \|y^{(k)}\|_1, \quad (9) \]

i.e. \( y \) differs from \( y^{(k)} \) in at least one bit\(^3\). As a result, a naive approach to find the \( m \)-best solutions suggests itself: for \( k = 1, \ldots, m \): (i) solve an ILP using standard solvers such as [18] to obtain \( y^{(k)} \); (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 \( k \)-th 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 \( k \)-th best solution assumes that the feasible set \( F_k \) has been partitioned into \( k - 1 \) disjoint sets \( F_1^k, \ldots, F_{k-1}^k \). The \( k \)-th solution \( y^{(k)} \) is found by searching all sets. To proceed to the \((k+1)\)-th solution they add the constraint \( y \neq y^{(k)} \); however this is redundant for all sets except for \( F_k \), which contains \( y^{(k)} \). 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 \( y \neq y^{(k)} \) and \( y_i \neq y_i^{(k)} \). However, in our case the variables are binary, so we note that 1) the constraint \( y \neq y^{(k)} \) is redundant in all sets except that containing \( y^{(k)} \), 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 \( y \) 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 \( \epsilon \):

\[ \mathcal{E} = \sum_{\theta \subseteq \Theta} p(\theta) - \sum_{\theta \subseteq \Theta^m} p(\theta) < \epsilon. \quad (10) \]
It can be proved\(^4\) that a tight upper bound for this error is  
\[ E \leq (n_h - m) \exp(-C_m^*). \]
Therefore, for any case, \( m \) can be automatically calculated such that this tight upper bound error is less \( \epsilon \).

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 \( x_t = (x_p, \dot{x}, y_p, \dot{y}) \) and the motion is modeled by the discrete update equation \( x_t = F x_{t-1} + \eta \) where \( F = \text{diag}[F_1, F_1] \) is a constant velocity model and \( \eta \) is Gaussian noise with covariance \( Q = \text{diag}[Q_1, Q_1] \) representing unmodeled acceleration. \( F \) and \( Q \) take their textbook values as in:

\[
F_1 = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \quad Q_1 = q_d \begin{bmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{bmatrix},
\]

where \( \tau = 1 \) is the sampling period and \( q_d = 0.02 \) is the process noise parameter. Both noisy and spurious detection points \( z_t = (\tilde{x}_p, \tilde{y}_p) \) 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 \( \beta \) 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 \( E \) 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 \( m \gtrsim 30 \) best solutions. This figure also shows how the averaged processing time (green line) increases sub-linearly with \( m \). In this experiment, our 3F-JPDA\(_m\) requires less than 2 seconds for any value of \( m \leq 100 \) 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

\[ \text{Proof provided in the supplementary files.} \]
the estimated number of possible assignments exceeds a threshold. Since the total number of hypotheses \(n_k\) 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 \(k^{th}\)-best solution of an ILP for the naive approach (see Sec. 5) is \(O((A + k)^{1.5}B^2)\), where \(B = MN\) and \(A = M + N\) [30]. This becomes computationally prohibitive for large \(k\). In contrast, our proposed approach takes \(O(A^{1.5}(B - D_k)^2)\) 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 \(\approx 204\) spots per frame) moving inside a cell membrane with an effective region \(= 230 \times 230\) pixels.

<table>
<thead>
<tr>
<th>Method</th>
<th>Location(_\downarrow) (Pixel)</th>
<th>Cardinality(_\downarrow) (Pixel)</th>
<th>OSPA-T(_\downarrow) (Pixel)</th>
<th>Time(_\downarrow) (Sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHT [11]</td>
<td>5.38</td>
<td>1.94</td>
<td>7.32</td>
<td>0.23</td>
</tr>
<tr>
<td>JPDA [35]</td>
<td>2.14</td>
<td>4.06</td>
<td>6.20</td>
<td>2.38</td>
</tr>
<tr>
<td>JPDA(_m)</td>
<td>2.14</td>
<td>4.06</td>
<td>6.20</td>
<td>0.20</td>
</tr>
<tr>
<td>3F-JPDA(_m)</td>
<td>1.94</td>
<td>3.22</td>
<td>5.16</td>
<td>3.13</td>
</tr>
</tbody>
</table>

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

We compare the results of our proposed single frame JPDA (JPDA\(_m\)) and three frames JPDA (3F-JPDA\(_m\)) 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\(_m\) 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](http://icy.bioimageanalysis.org).
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-Stadmitte) [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 fluorescence 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/](http://motchallenge.net/results/2D_MOT_2015/)
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.

Acknowledgment: This work was supported by ARC Linkage Project LP130100154.
References