Machine Learning Methods for Data Association in Multi-Object Tracking

PATRICK EMAMI, PANOS M. PARDALOS, LILY ELEFTERIADOU, and SANJAY RANKA,
University of Florida

Data association is a key step within the multi-object tracking pipeline that is notoriously challenging due to its combinatorial nature. A popular and general way to formulate data association is as the NP-hard multi-dimensional assignment problem. Over the past few years, data-driven approaches to assignment have become increasingly prevalent as these techniques have started to mature. We focus this survey solely on learning algorithms for the assignment step of multi-object tracking, and we attempt to unify various methods by highlighting their connections to linear assignment and to the multi-dimensional assignment problem. First, we review probabilistic and end-to-end optimization approaches to data association, followed by methods that learn association affinities from data. We then compare the performance of the methods presented in this survey and conclude by discussing future research directions.

CCS Concepts: • **Computing methodologies → Tracking**;

Additional Key Words and Phrases: Multi-object tracking, data association, machine learning, deep learning

ACM Reference format:
Patrick Emami, Panos M. Pardalos, Lily Elefteriadou, and Sanjay Ranka. 2020. Machine Learning Methods for Data Association in Multi-Object Tracking. *ACM Comput. Surv.* 53, 4, Article 69 (August 2020), 34 pages.
https://doi.org/10.1145/3394659

1 INTRODUCTION

The assignment problem is a classic combinatorial optimization problem where the goal is to find a weighted matching within a bipartite graph such that the sum of the weights is minimized. Within the field of computer vision, it is often used as a framework for tackling data association in multi-object tracking. In this survey, we set out to reexamine the data association problem through the lens of assignment problems as a means to abstract away details and to create a clear conceptual framework for unifying the many recently proposed learning-based data association algorithms. Visual multi-object tracking is a highly complex topic, so rather than attempt to provide a

This work was supported by the National Science Foundation under grant 1446813 and the Florida DOT under grant BVD31-977-45. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. P. M. Pardalos' research was supported by the Paul and Heidi Brown preeminent professorship at ISE, University of Florida.

Authors' addresses: P. Emami and S. Ranka, University of Florida, 432 Newell Drive, Gainesville, FL 32611; emails: pemami@ufl.edu, sanjayranka@gmail.com; P. M. Pardalos, University of Florida, 401 Weil Hall, Gainesville, FL 32611; email: p.m.pardalos@gmail.com; L. Elefteriadou, University of Florida, 512 Weil Hall, Gainesville, FL 32611; email: elefter@ce.ufl.edu.

0360-0300/2020/08-ART69 $15.00
https://doi.org/10.1145/3394659

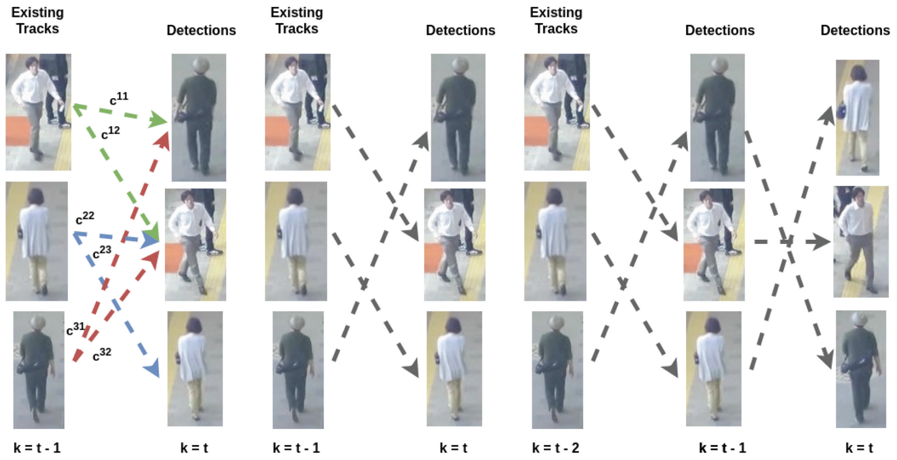
comprehensive overview, we instead take a closer look at solely the association step. Later, we will suggest surveys that review other aspects of the complete multi-object tracking problem for the interested reader. In this work, we argue that studying how machine learning can be used to solve data association is important for the following reasons. First, modern machine learning methods, particularly convolutional neural networks (CNNs), excel at learning discriminative features from raw sensor inputs for computing similarities between objects, which is an integral step for any data-driven matching task. For example, a recent study by Bergmann et al. [10] showed that a simple CNN bounding box regressor can be exploited to extend object tracks over time and drastically reduce the number of ID switches, putting into question the efficacy of sophisticated data association algorithms. Second, efficient probabilistic tools for approximate inference over highly structured models, such as those that arise in data association, have long been studied and are useful for dealing with noisy sensor measurements. Finally, there are many promising recent works on applying machine learning to directly solve a variety of combinatorial optimization problems [8], and it is interesting to ask whether assignment problems can be solved in a similar manner.

Multi-object tracking with one or more sensors plays a significant role in many surveillance and robotics applications. A tracking algorithm provides higher-level systems with the ability to make real-time decisions based on the state of the surrounding environment and is a core part of many scene understanding frameworks. Within intelligent transportation systems, it can be used for increasing pedestrian safety at traffic intersections [76], moving object awareness for self-driving cars [88], and for traffic surveillance [2, 52, 101, 138]. Multi-object tracking also has myriad other applications ranging from general security systems to tracking cells in microscopy images [70]. There are many sensor modalities that can be used for these applications; the most common are video, radar, and LiDAR. As a motivating example, consider a vision system that tracks vehicles and pedestrians at an urban traffic intersection. The real-time tracking data can be used for adaptive traffic signal control to optimize the flow of traffic at that intersection. However, intersections contain numerous challenges for multi-object tracking. Heavy traffic occupying multiple lanes and unpredictable pedestrian motion makes for a cluttered scene with lots of occlusion, false alarms, and missed detections. Variability in the appearance of targets caused by poor lighting and weather conditions is especially problematic for visual tracking. However, new technologies such as vehicle-to-infrastructure (V2I) communication enables vehicles to transmit information directly to traffic intersections, augmenting the data collected by traffic cameras and other sensors [32].

1.1 Data Association in Multi-Object Tracking

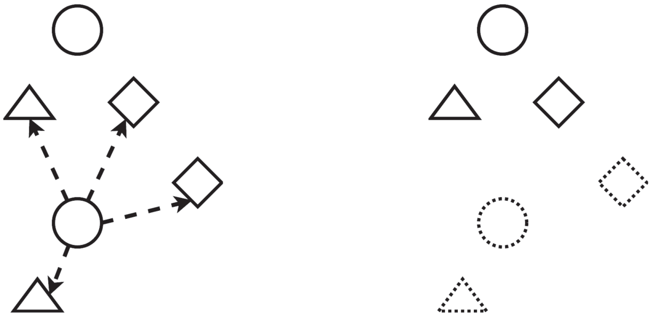
At the core of multi-object tracking lies the measurement-to-track and track-to-track association problems. The goal of measurement-to-track association is to identify a correspondence between a collection of new sensor measurements and preexisting tracks (Figure 1). New measurements can be generated by previously undetected targets, so care must be taken to not erroneously assign one of these measurements to a preexisting track. Likewise, the measurements that stem from clutter within the surveillance region must be identified to avoid false alarms. When there are multiple sensors, there is also the additional problem of track-to-track association. This problem seeks to find a correspondence between tracks that are generated by different sensors (Figure 2). Once the optimal assignment of the multi-sensor tracks has been found, all of the tracks assigned to a single track can be combined to produce the final estimate of that track's state. The sensors might be homogeneous or heterogeneous; in the latter case, the problem becomes even harder as the sensors could produce vastly different types of data.

Broadly speaking, algorithms for solving these two association tasks can be classified as single-scan, multi-scan, or batch. A single-scan algorithm only uses measurement or track information from the most recent timestep, whereas multi-scan algorithms use information from previous



(a) Linear Assignment (b) Linear Assignment (c) Multidimensional Assignment

Fig. 1. Data association in multi-object tracking. (a) In online tracking, new sensor detections are matched to existing tracks at each timestep by solving a LAP. The assignment hypotheses are the colored, dashed arrows. Each arrow is annotated with the cost c^{ij} of associating track i with detection j. (b) The optimal linear assignment. Notice how the assignment partitions the set of existing tracks and detections. (c) In batch, or offline single-sensor tracking, multiple sets of detections within a sliding window are associated all at once with a set of existing tracks. Here, the sliding window size T is 2 and the optimal assignment is shown. The images are taken from a random video in the MOT Challenge dataset [79].



(a) (b)

Fig. 2. Track-to-track association. There are three different sensors (circles, triangles, and diamonds) covering the surveillance region, each maintaining two tracks. Suppose there are two ground truth objects. (a) The dashed arrows show the possible ways of associating one of the circle tracks with the tracks from the triangle and diamond sensors. (b) The best track-to-track association hypothesis. The shapes with solid lines show all tracks, one per sensor, that have been assigned together as having originated from the same ground truth object, and likewise for the shapes with dotted lines. The solution effectively partitions each sensor's track lists.

and/or future timesteps. Batch, or offline multi-object tracking, is an extreme version of multi-scan where the entire sequence is available. Online multi-object tracking operates on one or a few of the most recent scans at a time. Generally, multi-scan methods are preferable in situations where the objects of interest are closely spaced and there are a lot of false alarms and missed detections. However, delaying the association to leverage future information negatively affects the real-time capabilities of the tracker. The accuracy and precision of the tracks produced by multi-scan

Table 1. Taxonomy of Assignment Problems in Multi-Object Tracking

	Measurement-to-Track Association	**Track-to-Track Association**
Single Scan	LAP (1–2 sensors), MDAP (≥3 sensors)	LAP (2 sensors), MDAP (≥3 sensors)
Multi-Scan	MDAP (≥1 sensors)	MDAP (≥2 sensors)

Note: The algorithms presented in this survey are mostly for solving the various MDAPs encountered in multi-object tracking, and are generally applicable (with modification) to both measurement-to-track and track-to-track association.

methods are usually superior, and they offer fewer track ID switches, track breaks, and missed targets [93]. Naturally, multi-scan methods are more computationally expensive and difficult to implement than their single-scan counterparts. The majority of the algorithms we will discuss in this survey are online algorithms, as offline algorithms typically involve sophisticated global optimization that as of yet is not data-driven.

Table 1 presents a categorization of the various data association problems mapped onto assignment problems. The easiest to solve is the bipartite matching or linear assignment problem (LAP), which seeks to match m tracks to n detections. Usually, the problem is constrained so that each track is assigned to exactly one measurement, but measurements are allowed to not be assigned (i.e., false alarms) or to be assigned to a "dummy track" (i.e., a missed detection). For multi-dimensional data association, such as the multi-scan extension of the aforementioned LAP, extra constraints ensure that each sensor measurement at each timestep is assigned to a track exactly once. Unfortunately, the multi-dimensional assignment problem (MDAP) is NP-hard for dimensions ≥3, whereas there exist many polynomial-time algorithms for the LAP such as the Hungarian method [83]. We will formulate these problems more rigorously in Section 2.

1.2 Comparison with Related Surveys

There are several related surveys to this one, and in this section we will highlight their main differences with ours. Both Poore [92] and Poore and Gadaleta [93] provide detailed treatments of how assignment problems are useful for multi-object tracking. They only go so far as to frame assignment problems in the context of multi-object tracking. There are several excellent general surveys on multi-object tracking [72, 139]; however, their focus is on all aspects of a multi-object tracking solution and they do not have any emphasis on machine learning methods. A survey on appearance matching in camera-based multi-object tracking discusses machine learning methods for improving data association, but it does not cover the recent advances in deep learning that have become ubiquitous in the computer vision tracking community [68]. The survey by Ciaparrone et al. [26] provides a general overview of deep learning in multi-object tracking.

1.3 Overview of MOT Benchmarks

In this section, we will briefly review the standard multi-object tracking benchmarks. Perhaps the most popular visual-based multi-object tracking set of benchmarks are the MOT challenges. The MOT15 challenge was first released in 2014 and consists of 22 video sequences of pedestrians [66]. Since then, the MOT16 and MOT17 challenges have been released, with each release also improving upon the annotation protocol and ground truth quality of the former [79]. These datasets are useful when proposing general improvements to multi-object tracking algorithms since results from many of the state-of-the-art trackers are publicly available for comparison. For an empirical comparison of state-of-the-art trackers on the MOT17 benchmark, see Leal-Taixé et al. [67]. A more recent comparison that focuses on various deep learning based trackers is available in Ciaparrone et al. [26]. The MOT datasets are particularly challenging because scenes are filmed from both static and moving vantage points, the density of the crowds of pedestrians is varied, and the appearances of pedestrians drastically changes between sequences. Previously, the PETS [33],

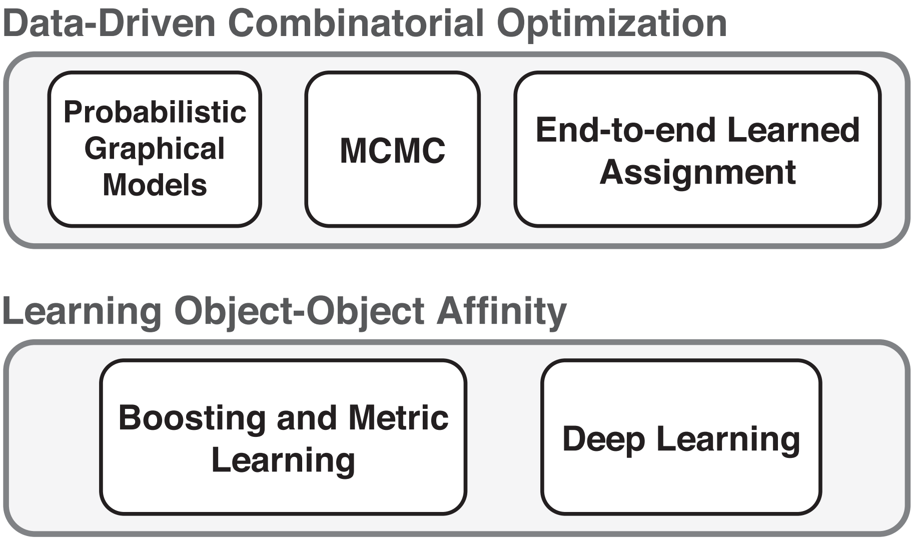

Data-Driven Combinatorial Optimization
Probabilistic Graphical Models
MCMC
End-to-end Learned Assignment
Learning Object-Object Affinity
Boosting and Metric Learning
Deep Learning

Fig. 3. Our categorization of machine learning methods for data association.

TUD Stadtmitte [3], and ETH Pedestrian [34] datasets were widely used as benchmarks. These offer a wide variety of multi-view, indoor, and outdoor scenes, and are still useful for training and testing, despite being less frequently used to assess state-of-the-art performance in recent works.

Other datasets of note include the KITTI benchmark [40], which is is focused on challenges for autonomous driving in urban environments and contains many tasks beyond multi-object tracking such as odometry, lane estimation, and orientation estimation. The UA-DETRAC benchmark [126] is a large-scale traffic surveillance benchmark of 10 hours of video that was recorded at 24 different locations in China and contains more than 8,250 vehicles that were manually annotated. For multi-sensor traffic surveillance, the Ko-PER intersection dataset [111] offers six sequences collected with multiple cameras and laser scanners; however, only two sequences currently have ground truth labels.

1.4 Roadmap

Our presentation of data-driven techniques for solving data association is split into two main sections. The first is focused on the combinatorial optimization aspect of the problem, and the second is concerned with learning features for the assignment cost function. Prior to this, in Section 2, we carefully present the connections between data association and assignment problems in multi-object tracking. Section 3 will present techniques for finding optimal assignments, with a focus on probabilistic and data-driven algorithms. Then, in Section 4, we present multiple methods for learning features for data association. This presentation is split between algorithms used in multi-object tracking prior to and after the introduction of deep learning. Section 5 includes a performance comparison of methods highlighted in this survey, and Section 6 presents our conclusion. For a visual representation of the organization of the technical contribution of the survey, see Figure 3.

2 DATA ASSOCIATION AS ASSIGNMENT

We will first formally introduce the LAP in the context of single-sensor data association and track-to-track association with two sensors. Following this, we will examine certain MDAP formulations for data association problems.

2.1 Linear Assignment

Consider a scenario where there are m existing tracks and n new sensor measurements at time k, $k = 1, \ldots, T$. We assume that there is a matrix $C_k \in \mathbb{R}^{m \times n}$, with entries $c_k^{ij} \in C$ representing the cost of assigning measurement j to track i at time k (Figure 1(a) and (b)). The goal is to find the optimal assignment of measurements to tracks so that the total assignment cost is minimized. Using binary decision variables $x^{ij} \in \{0, 1\}$ to represent an assignment of a measurement to a track, we end up with a 0-1 integer program

$$\min_{x \in X} \sum_{i=1}^{m} \sum_{j=1}^{n} c_k^{ij} x^{ij} \tag{1}$$

with constraints

$$\begin{aligned} \sum_{i=1}^{m} x^{ij} &= 1, \quad j = 1, \ldots, n \\ \sum_{j=1}^{n} x^{ij} &= 1, \quad i = 1, \ldots, m, \end{aligned} \tag{2}$$

where $x \in X$ is a binary assignment matrix. There are mn constraints forcing the rows and columns of X to sum to 1. Note that C_k is not required to be a square matrix. To capture the fact that some sensor measurements will either be false alarms or missed detections, a dummy track is added to the set of existing tracks so that C_k is now an $(m + 1) \times n$ matrix. The entries in the $(m + 1)^{\text{th}}$ row represent the costs of classifying measurements as false alarms. Missed detections are usually handled by forming validation gates around the m tracks (see Section 6.3 of Blackman and Popoli [13]). These gates can be used to determine, with some degree of confidence, whether any of the new measurements might have originated from a track. The canonical approach is to use elliptical gates, which are typically computed from the covariance estimates provided by a Kalman filter. In video-based tracking, a similar tactic is to suppress object detections with low confidence values.

Even though there are $\min(m, n)!$ possible assignments, many polynomial-time algorithms exist for finding the globally optimal assignment matrix. The most famous is the $O(n^3)$ Hungarian algorithm [59, 83]. Another popular method is the auction algorithm, introduced by Bertsekas [12]. These algorithms are fast and are easy to integrate into real-time multi-object tracking solutions. However, by only considering the previous timestep when assigning measurements or tracks, we are making a Markovian assumption about the information needed to find the optimal assignment. In situations with lots of clutter, false alarms, missed detections, and occlusion, the performance of these algorithms will significantly deteriorate. Indeed, it may be beneficial to instead use a sliding window of previous and/or future track states to construct assignment costs that model the relationship between tracks and new sensor measurements more accurately. As indicated in Table 1, the single-scan track-to-track association problem with two sensors is also a LAP, where m and n represent the sets of tracks maintained by each sensor. Similar methods for handling false alarms and missed detections in data association can be used for track-to-track association with uneven sensor track lists. If the assignment costs are known, an optimal track assignment can be found in polynomial time using one of the previously mentioned algorithms.

Instead of abandoning local data association in favor of more expensive global data association approaches, some have proposed heuristics involving solving a cascade of LAPs [1, 130]. In particular, DeepSORT [130] has gained in popularity due to its real-time speed and effective use of deep association features to achieve high-quality tracking.

2.2 Multi-Dimensional Assignment

Within the single-sensor and multi-sensor tracking paradigms, there are a few different ways to formulate measurement-to-track and track-to-track association as a MDAP (see Table 1). Each formulation seeks to optimize slightly different criteria, but each solution technique is generally applicable to all of them with minor modifications. We suggest further reading on the MDAP for more details [13, 53, 92].

2.2.1 Measurement-to-Track Association. We begin by considering the MDAP for measurement-to-track association with one sensor given multiple scans. Let the number of scans, or the temporal sliding window size, be given by T. Since the objective is to associate new sensor measurements with a set of existing tracks, the resulting MDAP has $T + 1$-dimensions (Figure 1(c)). When $T \geq 2$, the assignment problem is NP-hard [53].

Let the set of noisy measurements at time k be referred to as *scan* k and be represented by $Z_k = \{z_k^i\}$, where i is the i^{th} measurement of scan k, $i = 1, \ldots, M_k$. M_k is the number of measurements in each scan (i.e., $|Z_k| = M_k$). The main assumption we are making is that each object is responsible for at most one measurement within each scan. We let $Z^T = \{Z_1, \ldots, Z_T\}$ represent the collection of all measurements in the sliding window of size T.

Let Γ be the set of all possible partitions of the set Z^T. We seek an optimal partitioning $\gamma^* \in \Gamma$, also called a *hypothesis*, of Z^T into tracks. Note that a track is just an ordered set of measurements $\{z_1^i, z_2^i, \ldots, z_T^i\}$; one measurement from each scan at each timestep is attributed to each track. Hence, a partition γ represents a valid collection of tracks that adhere to the MDAP constraints. Now, we define γ^j to be the j^{th} track in γ. Following this, we can define a cost for each track γ^j in a partition as $c_{i_1, i_2, \ldots, i_T}$, where the indices $i_1, i_2, \ldots, i_T$ indicate which measurements from each scan belong to this particular track. This represents the cost of track j being assigned measurement i from scan 1, measurement i from scan 2, and so on. Crucially, the multi-dimensional constraints prevent measurements from being assigned to two different tracks and ensure that each measurement is matched to a track. If we use binary variables $\rho_{i_1, i_2, \ldots, i_T} \in \{0, 1\}$ to indicate if a track is present in a partition, then we can represent the MDAP objective as

$$\min_{\gamma \in \Gamma} \sum_{i_1=1}^{M_1} \cdots \sum_{i_T=1}^{M_T} c_{i_1, i_2, \ldots, i_T} \rho_{i_1, i_2, \ldots, i_T} \tag{3}$$

with constraints

$$\begin{aligned}
\sum_{i_2=1}^{M_1} \cdots \sum_{i_T=1}^{M_T} \rho_{i_1, i_2, \ldots, i_T} &= 1; \qquad & i_1 &= 1, \ldots, M_1 \\
\sum_{i_1=1}^{M_1} \cdots \sum_{i_T=1}^{M_T} \rho_{i_1, i_2, \ldots, i_T} &= 1; \qquad & i_2 &= 1, \ldots, M_2 \\
&\vdots & &\vdots \\
\sum_{i_1=1}^{M_1} \cdots \sum_{i_{T-1}=1}^{M_{T-1}} \rho_{i_1, i_2, \ldots, i_T} &= 1; \qquad & i_T &= 1, \ldots, M_T.
\end{aligned} \tag{4}$$

The solution ρ to this MDAP is the multi-dimensional extension of the binary assignment matrix. Simply, one may consider ρ as being a multi-dimensional array with binary entries such that the sum along each dimension is 1. Similarly to the LAP, we can augment each scan by including a z_k^0 dummy measurement in the set of detections at time k to address false alarms. This is useful for identifying track birth and track death as well, but care should be taken when defining the cost for assigning measurements as false alarms or missed detections to avoid high numbers of false positives and false negatives.

It is common to solve for an approximate solution within a fixed-sized sliding window T, then shift the sliding window forward in time by $t < T$ so that the new sliding window overlaps with the old region. This allows for tracks to be linked over time, and it provides a compromise between "offline" tracking, when T is set to the length of an entire sequence of measurements, and "online" tracking, when $T = 1$.

2.2.2 Track-to-Track Association. The other form of the MDAP we are interested in is multi-sensor association with $S \geq 3$ sensors. This scenario is common in centralized tracking systems, where sensors that are distributed around a surveillance region report raw measurements to a central node [14, 110]. When each sensor sends its local tracks to a central node for track association and fusion, an MDAP must be solved. In this case, the dimensionality of the MDAP is equal to S, and hence is NP-hard. Multi-scan track-to-track association with two sensors is also a MDAP, as well as multi-scan multi-sensor measurement-to-track association (Table 1).

Following Deb et al. [30], in this scenario there are $S \geq 3$ sensors, each maintaining a set of local tracks and using a sliding window of size $T \geq 1$. We define $X_k^s = \{x_k^{i,s}\}$, $s = 1, \ldots, S$, to represent the set of track state estimates produced by sensor s at time k. We have $i = 1, \ldots, N_s$, where N_s is the number of tracks being maintained by sensor s and $x_k^{i,s}$ interpreted as the i^{th} track of sensor s at scan k. Then, for each sensor, we have $X^{T,s} = \{X_1^s, \ldots, X_T^s\}$, which represents the collection of track state estimates within the sliding window. We seek an optimal partitioning $\gamma^* \in \Gamma$ of $X^T = \{X^{T,1}, \ldots, X^{T,S}\}$ of tracks over all scans and sensors that minimizes the total assignment cost, and we can define a partial assignment hypothesis in a partition γ as $\gamma^l = \{\{x_1^{j,1}, x_1^{j,2}, \ldots, x_1^{j,N_s}\}, \ldots, \{x_T^{j,1}, x_T^{j,2}, \ldots, x_T^{j,N_s}\}\}$. In words, this states that the j^{th} track of sensor 1 from scan 1, the j^{th} track of sensor 2 from scan 1, and so on, all correspond to the same underlying track l in scan 1. Likewise, this interpretation extends for all subsequent scans. As a quick example, suppose that there are three sensors each maintaining three tracks, and that $T = 1$. Then a potential hypothesis γ, or assignment, is $\{\{x^{1,1}, x^{2,2}, x^{1,3}\}, \{x^{2,1}, x^{1,2}, x^{2,3}\}, \{x^{1,3}, x^{2,3}, x^{3,3}\}\}$. This hypothesis makes the claim that track 1 from sensor 1, track 2 from sensor 2, and track 1 from sensor 3 all were generated by "true" track 1. The assignments for the other two tracks can be identified similarly. Note that the number of true targets in the surveillance region must either be known *a priori* or estimated. Considering the simplest case of $T = 1$, we can write the cost for a partial hypothesis as $c_{i_1, i_2, \ldots, i_{N_s}}$. Increasing T to include more than one scan corresponds to adding extra dimensions to the problem. We can use binary variables as before, $\rho_{i_1, i_2, \ldots, i_{N_s}} \in \{0, 1\}$, to indicate whether a particular partial hypothesis is present in γ. The MDAP can then be written as

$$\min_{\gamma \in \Gamma} \sum_{i_1=1}^{N_1} \cdots \sum_{i_{N_s}=1}^{N_s} c_{i_1, i_2, \ldots, i_{N_s}} \rho_{i_1, i_2, \ldots, i_{N_s}} \tag{5}$$

with constraints

$$\begin{aligned}
\sum_{i_2=1}^{N_1} \cdots \sum_{i_{N_s}=1}^{N_s} \rho_{i_1, i_2, \ldots, i_{N_s}} &= 1; \qquad i_1 = 1, \ldots, N_1 \\
\sum_{i_1=1}^{N_1} \cdots \sum_{i_{N_s}=1}^{N_s} \rho_{i_1, i_2, \ldots, i_{N_s}} &= 1; \qquad i_2 = 1, \ldots, N_2 \\
&\vdots \qquad\qquad \vdots \\
\sum_{i_1=1}^{N_1} \cdots \sum_{i_{N_{s-1}}=1}^{N_{s-1}} \rho_{i_1, i_2, \ldots, i_{N_s}} &= 1; \qquad i_{N_s} = 1, \ldots, N_s.
\end{aligned} \tag{6}$$

As with the multi-scan data association problem, the solution takes the form of a multi-dimensional binary array. As before, the number of potential assignment hypotheses in an MDAP can be reduced with gating. Even with gating, solving an MDAP for real-time tracking is infeasible. An analysis on the number of local minima in MDAPs with random costs shows that it increases exponentially in the number of dimensions [43]. Notably, the MDAP is closely related to other NP-hard combinatorial optimization problems, such as maximum-weight independent set and set packing [27]. In Section 3, we will show how the costs can be interpreted as probabilities; this will help motivate the use of approximate inference techniques for finding *maximum a posteriori* (MAP) solutions to MDAPs. However, we will begin our discussion of optimization approaches in Section 3 with techniques that do not require any assumptions about the nature of the cost function.

3 ALGORITHMS FOR FINDING OPTIMAL ASSIGNMENTS

We begin by briefly reviewing non-probabilistic optimization algorithms for solving the data association problem. These mostly fall into the category of offline data association. Next, our focus will shift to methods with a machine learning flavor. The techniques discussed in this section are quite general and in most cases can be used for both the measurement-to-track and track-to-track MDAPs with proper modification. The majority of these algorithms are developed for online MOT. We conclude by reviewing recent progress on end-to-end data association, which attempt to replace the combinatorial aspects of the problem with data-driven methods.

3.1 Non-Probabilistic Algorithms

3.1.1 Search Algorithms. Heuristically searching through the space of valid solutions within a time limit is an attractive way of ensuring both real-time performance and that a good local optima will be discovered. A search procedure for a MDAP takes as input a problem instance in the form of Equation (3) or Equation (5) and constructs a valid solution γ by adding each legal partial assignment incrementally. The most well-known method, the greedy randomized adaptive search procedure (GRASP), was originally introduced for multi-sensor multi-object tracking [84].

Other greedy search algorithms have been proposed [90, 105] based on the semi-greedy track selection (SGTS) algorithm [19]. SGTS-based algorithms first perform the usual greedy assignment algorithm step of sorting potential tracks by track score, then they generate a list of candidate hypotheses and return the locally optimal result.

The main strengths of search algorithms appear to be their simplicity and the extent to which they are embarrassingly parallel.

For a survey of research on GRASP for optimization, see the work of Resende and Ribeiro [98].

3.1.2 Lagrangian Relaxation. The multi-dimensional binary constraints 4 and 6 pose a significant challenge; a standard technique is to relax the constraints so that a polynomial-time algorithm can be used to find an acceptable sub-optimal solution. The existence of $O(n^3)$ algorithms [12, 59, 83] for the LAP suggests that if the constraints can be relaxed, a reasonably good solution to the MDAP should be obtainable within an acceptable amount of time. Indeed, Lagrangian relaxation algorithms for association in multi-object tracking [29, 30] involve iteratively producing increasingly better solutions to the MDAP by successively solving relaxed LAPs and reinforcing the constraints.

A parallel implementation of this method for the K-best case was developed [94, 95], which enables efficient implementations of multiple hypothesis tracking (MHT) algorithms. A variation on this approach using dual decomposition has been proposed as well [63].

Lagrangian relaxation has also been used to convert Equation (3) into a global network flow problem [18]. The motivation behind this approach is a desire to incorporate higher-order motion smoothness constraints beyond what is capable when only considering pairwise costs in multi-scan problems. The minimum-cost network flow problem that results from the relaxation can be solved in polynomial time; updates to the Lagrange multipliers enforcing the constraints are handled by sub-gradient methods. In the next section, we go into more detail on network optimization—one of the leading approaches to solving multi-object tracking association problems.

3.2 Probabilistic Graphical Models

3.2.1 Network Optimization. A popular approach (Equation (3)) in the multi-object tracking computer vision community is to transform the data association problem into finding a minimum-cost network flow [9, 18, 22, 50, 91, 103, 119, 122, 131, 137, 140]. In the corresponding network, detections at each discrete timestep generally become the nodes of the graph, and a complete flow path represents a target track, or trajectory. The amount of flow sent from the source node to the sink node corresponds to the number of targets being tracked, and the total cost of the flow on the network corresponds to the log-likelihood of the association hypothesis. The globally optimal solution to a minimum-cost network flow problem can be found in polynomial time, such as with the push-relabel algorithm.

Another benefit of using minimum-cost network flow is that the graph can be constructed to significantly reduce the potential number of association hypotheses by limiting transition edges between nodes with a spatiotemporal nearness criteria, similar to gating. Furthermore, occlusion can be explicitly modeled by adding nodes to the graph corresponding to the case where a target is partially or fully occluded by another target for some amount of time. A sliding window approach can be used for real-time performance rather than using the complete history of previous detections. To help illuminate the mapping from Equation (3) to a network flow problem, we adapt the following equations from Zhang et al. [140], rewritten using the notation from Section 2.

Recall that we defined a data association hypothesis γ as a partitioning of the set of all available measurements Z^T. Then, a MAP formulation of the MDAP for data association is given by

$$\begin{gathered}\gamma^* = \arg\max_{\gamma \in \Gamma} P(Z^T \mid \gamma) \prod_{\mathcal{T}_m \in \gamma} P(\mathcal{T}_m) \\ \text{s.t. } \mathcal{T}_m \cap \mathcal{T}_n = \emptyset, \forall m \neq n,\end{gathered} \tag{7}$$

where the product over tracks in the objective reflects an assumption of track motion independence, and the potentially prohibitive constraint guarantees that no two tracks ever intersect. It is possible to derive the measurement likelihood using Equation (22); in Zhang et al. [140], it is factored as $P(Z^T \mid \gamma) = \prod_z P(\{z \in Z^T\} \mid \gamma)$, where each term in this product is a Bernoulli distribution with parameter β encoding the probability of false alarm and missed detection. The track probabilities $P(\mathcal{T}_m)$ are modeled as Markov chains to capture track initialization, termination, and state transition probabilities. A network flow graph can now be defined as a graph with source s and sink t as follows. For every measurement $z_k^i \in Z^T$, create two nodes u_r, v_r, create an arc (u_r, v_r) with cost $c(u_r, v_r)$ and flow $f(u_r, v_r)$, an arc (s, u_r) with cost $c(s, u_r)$ and flow $f(s, u_r)$, and an arc (v_r, t) with cost $c(v_r, t)$ and flow $f(v_r, t)$. For every transition $P(z_{k+1}^i \mid z_k^i) \neq 0$, create an arc (v_r, u_s) with cost $c(v_r, u_s)$ and flow $f(v_r, u_s)$. An example of such a graph is given in Figure 4.

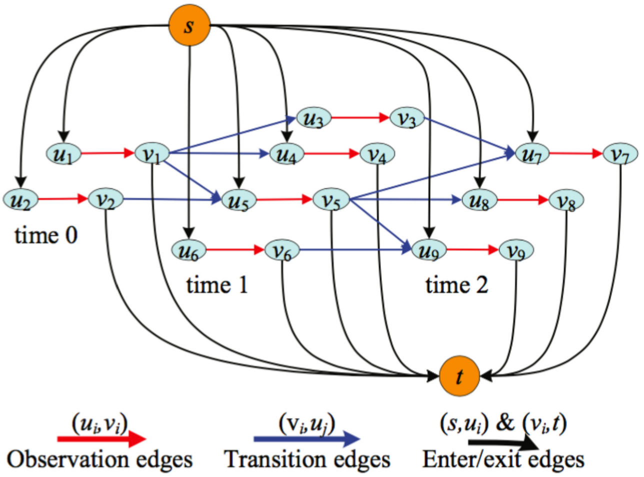


Fig. 4. A network flow graph for multi-scan data association (three scans depicted). The black arcs represent enter/exit edges for a potential track. The red arcs are measurement/observation edges, and the blue arcs are transition edges between measurements. Reproduced from Zhang et al. [140] with permission.

The flows f are indicator functions defined by

$$\begin{aligned} f(s,u_r) &= \begin{cases} 1 & \text{if } \exists \mathcal{T}_m \in \mathcal{T}, \mathcal{T}_m \text{ starts from } u_r \\ 0 & \text{otherwise} \end{cases} \\ f(v_r,t) &= \begin{cases} 1 & \text{if } \exists \mathcal{T}_m \in \mathcal{T}, \mathcal{T}_m \text{ ends at } v_r \\ 0 & \text{otherwise} \end{cases} \\ f(u_r,v_r) &= \begin{cases} 1 & \text{if } \exists \mathcal{T}_m \in \mathcal{T}, z_k^i \in \mathcal{T}_m \\ 0 & \text{otherwise} \end{cases} \\ f(v_r,u_s) &= \begin{cases} 1 & \text{if } \exists \mathcal{T}_m \in \mathcal{T}, z_{k+1}^i \text{ comes after } z_k^i \text{ in } \mathcal{T}_m \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

and the costs are defined as

$$\begin{aligned} c(s,u_r) &= -\log P_{\text{start}}(z_k^i) \quad & c(v_r,t) &= -\log P_{\text{end}}(z_k^i) \\ c(u_r,v_r) &= \log \frac{\beta_r}{1-\beta_r} \quad & c(v_r,u_s) &= -\log P_{\text{link}}(z_{k+1}^i \mid z_k^i), \end{aligned} \tag{9}$$

and can be derived by taking the logarithm of Equation (7); see Section 3.2 in the work of Zhang et al. [140] for more details. The minimum cost flow through the network corresponds to the assignment γ^* with the maximum log-likelihood.

Quite a few variations on this model have been proposed in the literature. In one case, a subgraph is created for each track in the surveillance region and occlusion is modeled by adding special nodes to the graphs [50]. A linear programming relaxation with a sliding-window heuristic then enables approximate global solutions to be found in real time. A limitation of this approach is the requirement of knowing *a priori* the number of tracks in the surveillance region, as well as the poor worst-case complexity of the simplex method. Another work further optimizes the approach introduced in Zhang et al. [140] to reduce the runtime complexity [91]. In a more drastic departure from previous works in this direction, the problem has also been formulated as a K-shortest paths through a flow graph [9]. One argument against the previously discussed network flow models is that they exhibit an over-reliance on appearance modeling and pairwise costs [27]. They offer a variation on the network flow approach that uses a more general cost function. In Section 4, we will go over the details of works that propose a variety of machine learning techniques to obtain the link costs (Equation (9)) in network flow graphs. Network optimization techniques offer a good trade-off between complexity, ease of implementation, and performance.

3.2.2 Conditional Random Fields. Probabilistic graphical models provide us with a powerful set of tools for modeling spatiotemporal relationships among sensor measurements in data association and among tracks in track-to-track association. Indeed, conditional random fields (CRFs), a class of Markov random fields [62], have been used extensively for solving MDAPs in visual tracking [22, 64, 82, 88, 135, 136]. A CRF is an undirected graphical model, often used for structured prediction tasks, that can represent a conditional probability distribution between sets of random variables. CRFs are well known for their ability to exploit grid-like structure in the underlying probabilistic model.

We define a CRF over a graph $G = (V, E)$ with nodes $x_{v \in V} \in X$ such that each node emits a label $y \in Y$. For simplicity of notation, we refer to nodes as x and omit the subscript. The labels take on values from a discrete set (e.g., $\{0, 1\}$); in the context of multi-object tracking, a realization of labels $\mathbf{y}$ usually corresponds to an assignment hypothesis. A key theorem concerning random fields states that the probability distribution being modeled can be written in terms of the cliques c of the graph [44]. For example, in chain-structured graphs, each pair of nodes and corresponding edge is a clique.

CRFs, like the probabilistic network flow models discussed in the previous section, are essentially a tool for modeling probabilistic relationships between a collection of random variables. They require a separate optimization process for handling training and inference (e.g., the graph cut algorithm [15] or message-passing algorithms). We will focus on presenting how the data association problem is mapped onto a CRF and direct the reader to other sources [15] for details on how exactly approximate inference is carried out for these models. One of the benefits of using graphical models is that we have the flexibility to construct our graph using either sensor measurements, tracklets (measurements that are partially associated to form a "sub"-track), or full tracks. Tracklets are a common choice for CRFs since they give an attractive hierarchical quality to the tracking solution; low-level measurements are first associated into tracklets via, for example, the Hungarian algorithm, and then stitched together into full tracks via a CRF. By working at a higher level of abstraction, the original MDAP constraints 4 and 6 are modified slightly; all that is needed at the higher level is to ensure that each tracklet is only associated to one and only one track. This can also help reduce processing time for running in real time.

Each clique c in the graph has a clique potential ψ_c associated with it; usually, the clique potentials are written as the product of unary terms ψ_s and pairwise terms $\psi_{s,t}$. It is common to assume a log-linear representation for the potentials (i.e., $\psi_c = \exp(w_c^\intercal \phi(x, y_c))$). Note that the implied normalization term in Equation (10) can be omitted when solving for the maximum-likelihood labeling $\mathbf{y}$ for a particular set of observations $\mathbf{x}$ such that

$$\begin{aligned} P(\mathbf{y} \mid \mathbf{x}, w) &\propto \prod_{c} \psi_c(y_c \mid \mathbf{x}, w) \\ &\propto \prod_{s \in V} \psi_s(y_s \mid \mathbf{x}, w) \prod_{s,t \in E} \psi_{s,t}(y_s, y_t \mid \mathbf{x}, w). \end{aligned} \tag{10}$$

Features ϕ must be provided (or can be extracted from data with supervised or unsupervised learning) and weights w are learned from data. The observations $\mathbf{x}$ can be either sensor measurements (for data association) or sensor-level tracks (for track-to-track association). The Markov property of CRFs can be interpreted in the context of multi-object tracking as assuming that the assignment of the observations to tracks within a particular spatiotemporal section of the surveillance region is independent of how they are assigned to tracks elsewhere—conditional on all observations. This adds an aspect of local optimality and, in a way, embeds similar assumptions as a gating heuristic. A solution to Equation (10) (i.e., the maximum-likelihood set of labels $\mathbf{y}$) can be used as a solution to the corresponding MDAP.

As is common with CRFs, the problem of solving for the most likely assignment hypothesis is cast as energy minimization. The objective to minimize is an energy function, computed by summing over the clique potentials; each potential is interpreted as contributing to the energy of the assignment hypothesis. Each clique consists of a set of vertices and edges, where each vertex is a pair of tracklets that could potentially be linked together. The corresponding labels for each vertex take values from the set $\{0, 1\}$ and indicate whether a pair of tracklets are to be linked or not. The energy term for each clique is decomposed into the sum of a unary term for the vertices and a pairwise term for the edges. In one instance, the weights w are learned with the RankBoost algorithm [135]. Other techniques for learning the parameters of a CRF that maximize the log-likelihood of the training data include iterative scaling algorithms [62] and gradient-based techniques. In Section 4, we will examine the problem of learning weights for assignment costs in more detail. The features used to construct these terms include appearance, motion, and occlusion information, among others. CRF and network optimization–based trackers are by nature global optimizers and must be run with a temporal sliding-window to get near real-time performance. For example, extensions to the generic CRF formulation have been developed that enable it to run in real time [136].

A particular CRF formulation, near-online multi-target tracking (NOMT) [22], also builds its graph of track hypotheses using tracklets. The novelty of this work is in the use of an affinity measure between detections called the *aggregated local flow descriptor*, and in the specific form of the unary and pairwise terms in the energy function of the CRF. Inference in the CRF is sped up by first analyzing the structure of the graphical model so that independent sub-graphs can be solved in parallel.

Other variations on the preceding approaches have been seen as well. In one such work, the energy term of a CRF is augmented with a continuous component to jointly solve the discrete data association and continuous trajectory estimation problems [82]. Another study embedded a factor graph in the CRF to add more structure and help model pairwise associations explicitly [46]. Based on the insight that the size of the bounding box is an indicator of object localization accuracy, asymmetric pairwise terms are added to the CRF that take this idea into account for better uncertainty management [141].

In the sequel, we will investigate how factor graphs, the belief propagation (BP) inference algorithm, and its variants can be used to solve the MDAP. To summarize, applying CRFs to a specific multi-object tracking problem involves defining how the graphical model will be constructed from the sensor data, specifying an objective function, selecting or learning features for the terms within the objective function, training the model to learn the weights, and then performing approximate inference to extract the predicted assignment hypothesis.

3.2.3 Belief Propagation. In this section, we highlight recent work that formulate the association problems as MAP inference and use BP or one of its variants to obtain a solution. Chen et al. [20] and Chena et al. [21] showed the effectiveness of BP at finding the MAP assignment hypothesis for the single- and multi-sensor data association problems. BP is a general message-passing algorithm that can carry out exact inference on tree-structured graphs and approximate inference on graphs with cycles, or “loopy” graphs. The types of graphs under consideration are once again Markov random fields, albeit more general ones than the ones discussed in the previous sections.

Indeed, BP can be used on graphs that model joint distributions $P(\mathbf{x}) = P(x_1, x_2, \ldots, x_N)$ that can be factorized into a product of clique potentials. As before, the clique potentials are assumed to be factorizable into pairwise terms. Therefore, for cliques c, we have

$$\begin{aligned} P(\mathbf{x}) &\propto \prod_{c} \psi_c(x_c) \\ &\propto \prod_{s \in V} \psi_s(x_s) \prod_{s,t \in E} \psi_{s,t}(x_s, x_t). \end{aligned} \tag{11}$$

It is common to use factor graphs to explicitly encode dependencies between variables. A factor graph decomposes a joint distribution into a product of several local functions $f_j(X_j)$, where each X_j is some subset of $\{x_1, x_2, \ldots, x_N\}$. The graph is bipartite and has nodes x (i.e., discrete random variables) and factors (i.e., dependencies) $f \in \mathcal{F}$, and edges between the nodes and factors. For example, the graph of $g(x_1, x_2, x_3) = f_A(x_1) f_B(x_2, x_3) f_C(x_1, x_3)$ has factors f_A, f_B, and f_C and nodes x_1, x_2, x_3. The joint distribution for a factor graph can be written similarly to Equation (11) as

$$P(\mathbf{x}) \propto \prod_{s \in V} \psi_s(x_s) \prod_{f \in \mathcal{F}} \psi_f(x_{\eta_f}), \tag{12}$$

where η_f represents the set of nodes x that are connected to factor f.

Parallel message-passing algorithms, such as BP, operate by having each node of the graph iteratively send messages to its neighbors simultaneously. We define messages from a node x_s to its neighbors $x_t \in \mathcal{N}(s)$ as $\mu_{s \to t}(x_s)$. In a factor graph, the set of neighbors $\mathcal{N}(s)$ for a node x_s are its corresponding factors. The max-product algorithm is useful for finding the MAP configuration $x^* = \{x^*{}_s \mid s \in V\}$ that corresponds to the best assignment hypothesis γ^*. In this algorithm, messages are computed recursively in general pairwise Markov random fields by

$$\mu_{s \to t}(x_s) = \max_{x_t} \left\{ \psi(x_t) \psi_{s,t}(x_s, x_t) \prod_{\xi \in N(t) \backslash s} \mu_{\xi \to t}(x_t) \right\}, \tag{13}$$

and at convergence, each x_s^* can be calculated by

$$x_s^* = \arg\max_{x_s \in X} \left\{ \psi_s(x_s) \prod_{\xi \in \text{nbr}(s)} \mu_{\xi \to s}(x_s) \right\} \tag{14}$$

for neighborhood set nbr(s). These updates are not guaranteed to converge for graphs with cycles, and even if they do, they may not compute the exact MAP configuration [20]. Williams and Lau [127] present a proof of convergence of loopy belief propagation (LBP) for data association. LBP simply applies the BP updates repeatedly until the messages all converge; interestingly, LBP has been shown to perform favorably in practice for association tasks [78, 128, 129]. An improvement over the max-product algorithm for LBP is tree-reweighted max-product [117]. This algorithm is used for data association to output a provably optimal MAP configuration or acknowledge failure [20]. The key idea of the tree-reweighted max-product algorithm is to represent the original problem as a combination of tree-structured problems that share a common optimum [20].

To illustrate the use of BP for solving MDAPs, we will present the graphical model formulation from Zhu et al. [142] for multi-sensor multi-object track-to-track association. The structure of the graphical model is decided on-the-fly by producing sets of independent association clusters consisting of multi-sensor tracks that could plausibly be associated with each other. This is accomplished by computing elliptical gates around each track and clustering together all such tracks whose gates overlap using, for example, kinematic information. The nodes of the graph are the track state estimates for $T = 1$ and $S \geq 3$ sensors (Section 2), $\{x^{i,j} \mid x^{i,j} \in X^1 = \{X^{1,1}, X^{1,2}, \ldots, X^{1,S}\}\}$, where each $x^{i,j}$ is the i^{th} track state estimate from sensor j, $i = 1, \ldots, N_j$ and $j = 1, \ldots, S$. Edges only exist between nodes from different sensors when their elliptic gates overlap. A random variable $Y^{i,j}$ corresponding to each node $x^{i,j}$ is defined as a vector of $S - 1$ dimensions and stores the indexes of the tracks from the other sensors associated with the i^{th} track from sensor j. The node potentials are defined as $\psi_{x^{i,j}}(Y^{i,j}) = \exp(\rho)$, where ρ is the sum of pairwise costs, given by Equation (23). Using the notation $Y_k^{i,j}$ to denote the k^{th} entry of the $S - 1$-dimensional vector $Y^{i,j}$ (the index of the local track from sensor k), the edge potentials can be defined to ensure that each track from each sensor is associated once and only once by

$$\psi_{x^{l,m} \to x^{n,o}}(Y_n^{l,m} = p, Y_l^{n,o} = q) = \begin{cases} 0 & p = n, q \neq l \\ 0 & p \neq n, q = l \\ 1 & \text{otherwise.} \end{cases} \tag{15}$$

If $w^{u,v}$ is the Mahalanobis distance between two tracks u, v, then messages between nodes can be initialized as

$$\mu_{x^{l,m} \to x^{n,o}}(Y_l^{n,o} = q) = \begin{cases} \exp(w^{u=(l,m);v=(n,o)}) & \text{if } q = l \\ 1 & \text{otherwise.} \end{cases} \tag{16}$$

Then, repeated applications of Equations (13) and (14) until the $Y^{i,j}$s converge will produce the MAP solution.

This approach has been extended for an unknown number of targets and multiple sensors [77] and applied to a multi-static sonar network [78]. For a general overview of graph techniques for the data association problem, including BP, see the work of Chong [23].

3.3 Markov Chain Monte Carlo

A principled approach to sampling from a complex, potentially high-dimensional distribution is Markov chain Monte Carlo (MCMC). MCMC methods construct a Markov chain on the state space $\mathcal{X}$ whose stationary distribution π^* is the target distribution. Decorrelated samples drawn from the chain can be used for approximate inference (i.e., integrating with respect to π^*). This is useful in the context of assignment problems for multi-object tracking when the goal is to estimate a posterior distribution over assignment hypotheses, from which a MAP hypothesis can be extracted. The Metropolis-Hastings algorithm has been used extensively for data association in single- and multi-sensor scenarios [7, 35, 85, 89]. Recently, a Gibbs sampler was derived for efficient implementations of the labeled multi-Bernoulli filter, which jointly addresses the data association and state estimation problems for single- and multi-sensor scenarios [99, 116]. We omit detailed descriptions of the Metropolis-Hastings and Gibbs sampling algorithms, and instead refer the reader to relevant work [85, 116].

MCMC is applied to the MDAP for data association (referred to as MCMCDA) and track-to-track association by designating the state space of the Markov chain to be all feasible assignment hypotheses and the stationary distribution of the Markov chain to be the posterior $P(\gamma \mid Z^T)$ or $P(\gamma \mid X^T)$. A MAP assignment hypothesis γ^* for the data association problem is

$$P(\gamma \mid Z^T) \propto P(Z^T \mid \gamma) \prod_{t=1}^{T} p_z^{z_t} (1 - p_z)^{c_t} p_d^{d_t} (1 - p_d)^{g_t} \lambda_b^{a_t} \lambda_f^{f_t} \tag{17}$$

$$\gamma^* = \arg\max_{\gamma} P(\gamma \mid Z^T). \tag{18}$$

Here, we define the survival probability as p_z and the detection probability as p_d. The number of targets at time $t - 1$ is e_{t-1}, the number of targets that terminate at time t is z_t, and $c_t = e_{t-1} - z_t$ is the number of targets from time $t-1$ that have not terminated at time t. We set a_t as the number of new targets at time t, d_t as the number of actual target detections at time t, and $g_t = c_t + a_t - d_t$ as the number of undetected targets. Finally, let $f_t = n_t - d_t$ be the number of false alarms, λ_b be the birth rate of new objects, and λ_f be the false alarm rate. Note that for the general case of unknown numbers of targets, the multi-scan MCMCDA will find an approximate solution of unknown quality at best. A bound on the quality of the approximation for the single-scan fixed target MCMCDA has been derived [85].

A Metropolis-Hastings algorithm for Equation (17) is as follows [85]. The proposal distribution q is associated with five types of moves, for a total of eight moves: a birth/death move pair, a split/merge move pair, an extension/reduction move pair, a track update move, and a track switch move. A move is accepted with acceptance probability $A(\gamma, \gamma')$, where

$$A(\gamma, \gamma') = \min\left(1, \frac{\pi(\gamma')q(\gamma', \gamma)}{\pi(\gamma)q(\gamma, \gamma')}\right). \tag{19}$$

Assuming a uniform proposal distribution q, the proposal distribution terms in the numerator and denominator cancel. The stationary distribution $\pi(\gamma)$ is $P(\gamma \mid Z^T)$ from Equation (17). Implementation details and descriptions of each type of move can be found in Section V-A in the work of Oh et al. [85]. Extensions to this algorithm have been proposed [7] to add a sliding-window version and to reduce the number of types of moves to three. For visual tracking [7], appearance information is fused with kinematic information to help improve performance. Sparse representations of detections and kinematic information have been used to define an energy objective that MCMCDA approximately optimizes [35]. This work deviates from its predecessors by allowing moves to be done not only forward in time but also backward to explore the solution space more efficiently. The use of a sliding window is once again crucial, enabling the trade-off between solution quality and a faster runtime.

3.4 End-to-End Data Association

Neural networks have a rich history of being used to solve combinatorial optimization problems. One of the earliest and most influential works in this line of research, by Hopfield and Tank [48], describes how to use Hopfield nets to approximately solve instances of the traveling salesperson problem (TSP). Despite the controversy associated with their results [108], this work inspired many others to pursue these ideas. This has led to the present day, where research on the use of deep neural networks to solve combinatorial optimization problems has started to pick up speed [8].

Following broad trends within the deep learning research community, many have recently asked whether the data association step in multi-object tracking can be solved in an almost entirely "end-to-end" fashion. In other words, given noisy measurements of the environment, the tracker should directly output filtered tracks, combining the association problem with state estimation into a monolithic learned module. In this section, we will present various recent works that attempt to learn the data association step from data using deep learning.

3.4.1 Data-driven Association. The deep affinity network (DAN) [112] is a deep neural network that explicitly learns the affinity between objects over time. It is trained to predict the optimal linear assignment using ground truth assignment matrices as supervision. Visual features are first extracted from a VGG network and then processed by DAN to output a matrix of soft assignments, which finally are stitched into tracks using the Hungarian algorithm. The main insight of this approach is that DAN is able to jointly learn good appearance features and features that are highly "matchable." They showed equal or better performance on MOT15 and UA-DETRAC with state-of-the-art methods. A closely related tracker is FAMNet [25], which learns to predict the assignment tensor for the MDAP directly. They use a sliding window to construct a set of hypothesis tracklets, for which an affinity network outputs the affinity tensor for the MDAP. An iterative and differentiable row/column tensor normalization layer is used to directly output the assignment, through which gradients from a loss computed with the ground truth assignment can be backpropagated. Another deep tracker similar to DAN is the deep Hungarian network (DHN) [134], which also attempts to predict the optimal linear assignment from a cost matrix between measurements and tracks. Interestingly, they derive a differentiable version of the multi-object tracking metrics MOTA and MOTP [11] to directly formulate the loss in terms of the MOT metrics given ground truth assignments. The reported performance on the MOT17 benchmark are inferior to DAN, however. The dual matching attention network (DMAN) [143] augments their data association algorithm by introducing spatial and temporal attention networks that refine candidate assignments. The spatial attention generates dual attention maps to exploit the strengths of discriminative CNN feature embeddings for re-ID, as is commonly done in single-object tracking. Tracking by animation [45] is a deterministic unsupervised model that uses attention and memory mechanisms to learn to track using only reconstruction error. It assumes rather simplistic scene compositions to be able to render the predicted scene in a differentiable way. The memory mechanism uses read/write operations to address data association and keep track of which objects have been attended to at each timestep. Although their experimental results were mainly on small-scale datasets, this direction is very promising as high-quality labeled data for multi-object tracking is scarce. Finally, we note that reinforcement learning has been applied successfully to multi-object tracking [132] where a policy is learned over a data association Markov decision process that handles track initialization, maintenance, and removal.

3.4.2 Recurrent Neural Networks. An investigation by Ondruska and Posner [86] revealed that a recurrent CNN is able to learn to track multiple targets from raw inputs in a synthetic problem without access to labeled training data. Crucially, rather than maximizing the likelihood of the next state of the system at each timestep, they modified the cost function to maximize the likelihood at some time $t + n$ in the future to force the network to learn a model of the system dynamics. More recently, they extended this work for use with raw LiDAR data collected by an autonomous vehicle [31]. Recurrent autoregressive networks [36] was designed as an approach to online multi-object tracking that seeks to incorporate internal and external memory components into a deep learning framework to help handle occlusion and appearance changes. They are able to show that the recurrent autoregressive network indeed makes use of its external memory to maintain tracks while the targets are occluded. Sadeghian et al. [102] present a closely related prior work that also explores the use of recurrent neural networks (RNNs). Recently, RNNs were also used to identify track failures (ID switches) within a set of tracklets so as to automatically correct such cases in a post-processing step [74]. Explicit learning of the assignment problem was attemped by Milan et al. [81], where they used deep learning to separately tackle the state estimation and data association problems. They designed a long short-term memory (LSTM) cell specifically for solving the MDAP in data association (Figure 5). Despite not using any visual features, their approach achieves reasonable performance relative to other similar systems on the MOT Challenge 2015 dataset [66].

3.4.3 Deep Generative Models. Advances in our ability to train and scale deep generative models such as variational autoencoders [57], generative adversarial networks (GANs) [42], and normalizing flows [100] has resulted in investigations on their use for multi-object modeling. The benefits of generative models with respect to multi-object tracking are that they can be used for trajectory prediction [51, 58] or as scene priors for robust handling of occlusion [38]. Sequential Attend, Infer, Repeat (SQAIR) [58] and a recent follow-up work, SCALOR [51], maintain sets of

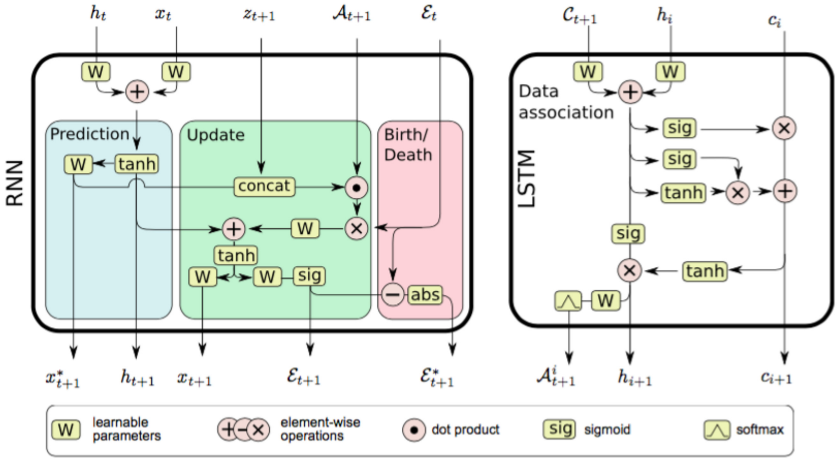


Fig. 5. An LSTM cell designed for multi-scan single-sensor data association (right). The input at each timestep is the matrix of pairwise distances C_{t+1}, along with the previous hidden state h_t and cell state c_t. The output A^i_{t+1} of the data association cell is a vector of assignment probabilities for each target and all available measurements, obtained by a log-softmax operation, and is subsequently fed into the state estimation recurrent network (left). The LSTM's non-linearities and memory are believed to provide the means for learning efficient solutions to the data association problem. Best viewed in color. Reproduced from Milan et al. [81] with permission.

latent variables corresponding to objects in the scene. The latent space of these generative models are structured to make it straightforward to differentiate through the rendering algorithm, allowing for them to be trained to maximize the evidence lower bound (ELBO) over a dataset of video sequences. Data association is addressed by a “glimpse” attention mechanism that sequentially attends to each object in an given frame. Notably, these models can handle objects that enter and leave the scene in the middle of a video sequence and have been applied to multi-pedestrian tracking. Relational-neural expectation maximization [115] uses iterative inference to assign pixels to object clusters in each image of sequence and captures interactions between objects using a neural relational dynamics component. The iterative inference is necessary to break the symmetry between the latent object components. Relational-neural expectation maximization learns to group the pixels belonging to a particular object to the same latent object component over time, forming a set of object tracks. Although these methods are theoretically interesting, an open problem is scaling them to real-world datasets. Deep generative models have been partially incorporated into existing multi-object tracking frameworks as well. A sequential GAN is used to improve the robustness of a pedestrian tracker in crowded scenes to occlusion and false detections [38]. They directly generate pedestrian heatmaps with the GAN's generator, which are used to associate new object detections. Then, they maintain a set of tracks by training LSTMs with attention to do short- and long-term trajectory prediction. They demonstrate slightly improved pedestrian detection performance compared to strong baselines on sequences from the PETS2009 benchmark.

To conclude, in this section, we reviewed a wide variety of machine learning approaches to the combinatorial optimization aspect of data association. We organized our presentation by describing how each fits into the framework of MDAPs. We first presented search algorithms and non-probabilistic discrete optimization methods to provide context for work done before recent data-driven approaches. Then, we discussed algorithms that fall broadly under the categories of network flow over probabilistic graphs, CRFs, BP over factor graphs, MCMC, and end-to-end learning. The end-to-end learning approaches can be contrasted with the other approaches for their abandonment of the structure provided by the combinatorial optimization framework in lieu of an almost complete reliance on data-driven techniques. In the next section, our focus shifts to reviewing recent works whose primary aim is to learn discriminative features for data association that can be used in tandem with some of the algorithms presented in this section.

4 LEARNING FEATURES FOR DATA ASSOCIATION

4.1 Assignment Costs

The particular choice of the data association cost function can have a large impact on the performance of a downstream task. We can observe from Equations (1), (3), and (5) that the cost functions for data association measure how "expensive" it is to include a particular assignment of detections (or tracks) to tracks in the solution. In this section, we introduce two perspectives toward formulating cost functions, specifically highlighting probabilistic approaches. Following that, we review machine learning methods for learning good features for data association, organized by non-deep learning and deep learning approaches.

4.1.1 Kinematic Costs. In situations where sensor measurements only consist of noisy estimates of kinematic data from targets (e.g., position and speed), a probabilistic framework can be used to recover the unobservable state of the targets. The most common approach is to handle the uncertainty in the sensor measurements and target kinematics with a stochastic Bayesian recursive filter; Mahler [75] presents a comprehensive overview. The Kalman filter—probably the most popular filter of this flavor—provides the means for updating a posterior distribution over the target state given the corresponding measurement likelihood (i.e., $P(x_k \mid z_k) \propto P(z_k \mid x_{k-1})P(x_{k-1} \mid z_{k-1})$). We are using the same notation as before such that x_k represents the target state at time k and z_k is the measurement at time k. One of the reasons for the popularity of the Kalman filter is that by assuming that all distributions of interest are Gaussian, the posterior update can be computed in closed form. Recall that a partial association hypothesis γ^j for the multi-scan single-sensor data association problem assigns T measurements to a single track within the sliding window of length T. The simplest cost function for data association is to minimize the following negative log-likelihood ratio:

$$c_{i_1,i_2,\ldots,i_T} = -\log \frac{P(\gamma^j \mid z_1^i, z_2^i, \ldots, z_T^i)}{P(\gamma^0 \mid z_1^i, z_2^i, \ldots, z_T^i)}, \quad (\gamma^j, \gamma^0) \in \gamma. \tag{20}$$

The partial hypothesis γ^j represents the j^{th} track of the hypothesis γ, and γ^0 represents a dummy track where all measurements attributed to it are considered false alarms. Assuming the sensor has a probability of 1 of detecting each target and a uniform prior over all assignment hypotheses, the likelihood that the j^{th} track generated the assigned measurements is

$$P(\gamma^j \mid z_1^i, z_2^i, \ldots, z_T^i) \propto P(z_1^i, z_2^i, \ldots, z_T^i \mid \gamma^j). \tag{21}$$

Assuming independence of the measurements and track states between timesteps, we can decompose the likelihood that the measurements originated from track γ^j as

$$P(z_1^i, z_2^i, \ldots, z_T^i \mid \gamma^j) = \prod_{k=1}^{T} P(z_k^i \mid x_k)P(x_k \mid j). \tag{22}$$

In the Kalman filter and its extensions, the right-hand side has an attractive closed-form representation as a Mahalanobis distance between the measurement predictions and the observed measurements, scaled in each dimension of the measurement space by the state and measurement covariances. This can easily be derived by taking Equation (22) and plugging it into the negative log-likelihood ratio in Equation (20).

In track-to-track association, the conventional cost function associated with a partial hypothesis is the likelihood that the tracks from multiple sensors were all generated by the same true target. When $S = 2$, the simplest approach is to consider the random variable $\triangle_{12} = x^1 - x^2$, which is the difference between the track state estimates from sensor 1 and sensor 2. When the track state estimates are Gaussian random variables, $\triangle_{12}$ is also Gaussian. The cost function becomes the likelihood that $\triangle_{12}$ has zero mean and covariance given by $\Sigma = \Sigma_1 + \Sigma_2 - \Sigma_{12} - \Sigma_{21}$ [6]. The first two terms of the covariance are the uncertainty around the track state estimates, and the second two terms are the cross covariances. A straightforward way to extend to the $S \geq 3$ case is to use star-shaped costs $\triangle_{1S} = \sum_{i=2}^{S} \triangle_{1i}$ [118]. For the Gaussian case, the cost can also be written in closed form as a Mahalanobis distance between the track state estimates [30, 54]:

$$c_{i_1, i_2, \ldots, i_S} = \sum_{j=2}^{S} \triangle_{1j}^{\mathsf{T}} \Sigma_{1j}^{-1} \triangle_{1j}. \tag{23}$$

In the Bayesian setting, minimizing Equations (20) and (23) is analogous to finding the MAP assignment hypothesis.

4.1.2 Feature-Augmented Costs. It is often the case in multi-object tracking that sensors generate high-dimensional observations of the surveillance region from which target information must be extracted. The most obvious example of this is the image data generated by a video surveillance system. This data, when featurized, can be used to augment or replace the kinematic costs mentioned in the previous section. The goal of doing this is to improve the association accuracy and ultimately the overall tracking performance.

Due to the high-dimensionality of the raw measurements, almost all such methods attempt to *learn* a pairwise cost between measurements or tracks using features extracted from the data. This pairwise cost can represent the association probability of the two objects, or simply some notion of similarity, such as a distance. There are many ways of formulating the problem of learning assignment costs and using it for solving data association or track-to-track association as a machine learning problem. For example, one technique is to use metric learning to transform the high-dimensional sensor measurements into a lower-dimensional geometric space where a Euclidean distance can be used as the assignment cost function. Learning pairwise costs from data is heavily used in the multi-object tracking computer vision community, partially due to the ease at which features can be extracted from images [68]. Of course, the main question is deciding what features to use, or whether to try to learn the best features for data association directly from data.

There are multiple ways to incorporate learned pairwise costs into data association when viewed as an MDAP. One common approach is as follows. The probability of association for a pair of measurements Λ_i and Λ_j (or tracks) can be written as a joint pdf [87]; assuming independence of the kinematic (K) and non-kinematic (NK) components of this probabilistic cost function, the resulting negative log-likelihood pairwise cost is

$$\begin{aligned} c_{ij} &= -\log P(\Lambda_i, \Lambda_j) \\ &= -\log(P_{\mathrm{K}}(\Lambda_i, \Lambda_j) P_{\mathrm{NK}}(\Lambda_i, \Lambda_j)) \\ &= -\log P_{\mathrm{K}}(\Lambda_i, \Lambda_j) - \log P_{\mathrm{NK}}(\Lambda_i, \Lambda_j). \end{aligned} \tag{24}$$

Usually, $P_{\mathrm{NK}}(\Lambda_i, \Lambda_j)$ is parameterized by weights θ and is a function of the features extracted from the sensor data and θ. For example, this probability could be represented as a neural network that outputs a similarity score between 0 and 1. The kinematic component of this pairwise cost, $P_{\mathrm{K}}(\Lambda_i, \Lambda_j)$, could be adapted from Equation (20).

Table 2. Features Used for Data-Driven Learning of Assignment Costs from a Representative Set of Works

Related Work	Method	Summary of Features Used
[69]	HybridBoost	Tracklet lengths, no. of detections in the tracklets, color histograms, frame gap between tracklets, no. of frames occluded, no. of missed detected frames, entry and exit proximity, motion smoothness
[60, 61, 136]	AdaBoost	Color histograms, covariance matrices, HOG
[135]	RankBoost	Tracklet lengths, no. of detections in the tracklets, color histograms, frame gap between tracklets, no. of frames occluded, no. of missed detected frames, entry and exit proximity, motion smoothness
[4]	ILDA	Templates from HSV color channel and tracklet ID
[122, 123]	Structured SVM	Off-the-shelf detector confidence (e.g., from DPM [37]), consecutive bounding box IOU, geometric relationships between all pairs of objects
[119, 120]	Metric learning	RGB, YCbCr, and HSV color histograms; HOG; two texture features extracted with Schmid and Gabor filters

Framing the problem of learning an assignment cost function for data association or track-to-track association is deeply intertwined with the choice of sensor(s). This section will mainly consist of recent work on this problem from the computer vision community, where machine learning is most heavily used. One reason for this is the relatively large amount of annotated video tracking datasets that are available. We divide the presentation of techniques into pre- and post-deep learning to provide a comprehensive perspective and to emphasize the shift to deep learning-based approaches in recent years.

4.2 Learning Features for Data Association, Pre-Deep Learning

The goal of learning features for data association is to use (usually labeled) training data to teach a model to output association scores at test time. These scores are then used to compute the assignment costs, as in Equation (24), and these costs are utilized by the optimization frameworks introduced in Section 3. In visual tracking, discriminative models have been commonly trained for predicting association scores based on appearance information. These models are typically adapted from popular classification and ranking models. Another learning paradigm (occasionally used in conjunction with discriminative models) is metric learning. In this case, the goal is to learn a distance metric between measurements or tracks, typically in the form of a parameterized Mahalanobis distance. The next two sections review these two learning techniques in the context of data association prior to the use of deep learning for feature extraction. As a key challenge for these methods was feature selection, we provide Table 2, which summarizes the various visual features used for learning association costs.

4.2.1 Discriminative Models. Boosting is one of the most powerful techniques in supervised learning and is a natural choice for learning discriminative models that approximate the true association costs. The general idea behind boosting is to produce a series of *weak* learners that are combined to form a single *strong* learner. The HybridBoost algorithm [69], one of the first applications of data-driven learning in multi-object tracking, is used to learn the link costs for a network flow graph (Equation (9)). The data association problem is decomposed into a hierarchy of association problems where the tracklet lengths successively increases [49]; furthermore, it is cast as a joint ranking and classification problem. The cost function is learned so that it can rank correct associations higher than incorrect ones, as well as reject some associations entirely (i.e., a binary classification to determine reasonable associations). Hence, HybridBoost is a combination of RankBoost and AdaBoost [39]. Their HybridBoost model is trained offline with videos paired with ground truth trajectories. In the work of Kuo et al. [60], a slightly different approach is taken; a hierarchical decomposition is used, but each stage of the hierarchy is linked by applying the Hungarian algorithm and the cost matrix for the Hungarian algorithm is learned online with AdaBoost. Online learning of the discriminative model within the sliding window is an attractive notion, since variations in appearance at test time can cause difficulty for systems that are trained offline. However, this comes at the cost of potentially sacrificing real-time capabilities. On a task involving tracking two to eight pedestrians at a time, this tracker runs at about 4 FPS. Other appearance models based on boosting have been proposed where the RankBoost algorithm is used with CRFs [135, 136]. In a follow-up work to Kuo et al. [60], ideas from person re-identification are used to improve the appearance model [61]. The features used by the boosting algorithms mentioned here are summarized in Table 2.

In efforts to improve upon boosting for online learning of appearance models, incremental linear discriminant analysis (ILDA) has been used by Bae and Yoon [4], who showed that ILDA outperforms boosting in their experiments in terms of identification accuracy and computational efficiency, partially because ILDA simply requires updating a single LDA projection matrix for distinguishing among the appearances of multiple objects. However, this approach makes the assumption that the featurized appearances of the tracked objects can be projected into a vector space where they are linearly separable. The assignment cost they used was

$$c_{ij} = \Lambda(x_i, x_j) = \Lambda^A(x_i, x_j)\Lambda^S(x_i, x_j)\Lambda^M(x_i, x_j) \tag{25}$$

for appearance, shape, and motion (kinematics) affinities. This form of the cost is similar to Equation (24) and is fairly common. The appearance affinity is the score computed by ILDA, and the shape and motion affinities are not learned from data. In this work, tracks are incrementally stitched together from tracklets by repeated application of the Hungarian algorithm. Another alternative to boosting that was explored for learning association costs within complex graphical models was the structured SVM [22, 56, 122, 123]. In general, however, the structured SVM approaches were restricted to linear cost functions.

4.2.2 Metric Learning. A different approach to addressing the problems of variability in object appearance is target-specific metric learning. Here, we define metric learning as the problem of learning a distance $d_{\mathbf{A}}(x, y) = \sqrt{(x-y)^\intercal \mathbf{A}(x-y)}$ parameterized by a positive semi-definite (PSD) matrix $\mathbf{A}$. An intuitive way of thinking about this is that the data points x, which might be featurized representations of tracked objects, are being mapped to $A^{1/2}x$ where a Euclidean distance metric can be applied to the rescaled data [133]. This is then cast as a constrained optimization problem to ensure that the solution $\mathbf{A}$ is valid (i.e., $\mathbf{A} \geq 0$). An early attempt at applying metric learning in multi-object tracking [124] combined the problem of learning a discriminative model for appearance matching given image patches with motion estimation and jointly optimized with gradient descent. Their formulation requires running the optimization at each timestep for all pairs of objects in the scene with a set of training samples that gets incrementally updated. A more efficient use of metric learning for multi-object tracking is learning link costs in a network flow graph [119, 120]. A regularized version of the constrained optimization problem is applied to learn a distance between feature vectors for an appearance affinity model. The intention is to learn a metric that returns a smaller distance for feature vectors within the same tracklet in the graph than for

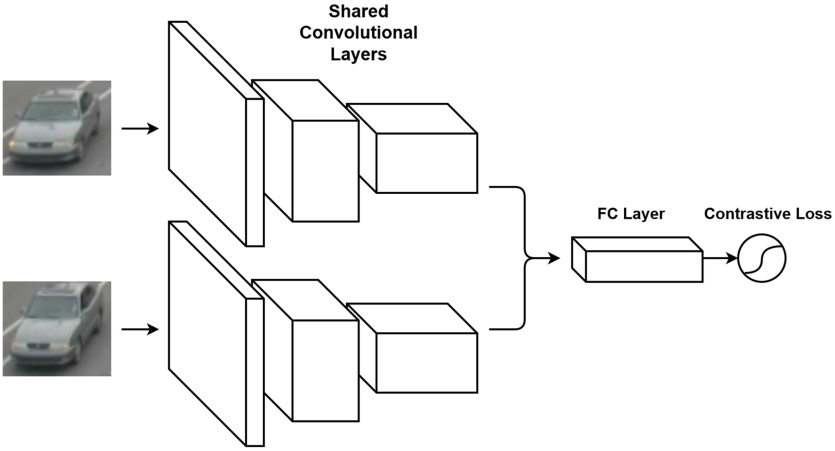


Fig. 6. The basic architecture of a Siamese network. The weights of the convolutional layers are shared between the two arms of the network. A contrastive loss can be used to train the network to predict the similarity of the two input images.

feature vectors that belong to different tracklets. The negative log-likelihood assignment cost for the network links is defined similarly to Equation (25).

4.3 Learning Features for Data Association, Post-Deep Learning

Tracking by detection is the current state-of-the-art approach for visual tracking, mainly due to the use of CNNs. The basic idea is to first leverage powerful deep networks for object detection to extract raw observations followed by an association step to produce object tracks. In this section, we will discuss the use of CNNs within the data association step.

CNNs learn features directly from raw images that are translation invariant and invariant to slight deformations, removing the need to hand pick features that may not generalize well. Another reason deep learning is an attractive option for multi-object tracking is because it is straightforward to take a CNN that has been pretrained on a massive image classification dataset and *transfer* the learned features to new tasks, including estimating association costs.

One of the first uses of deep learning in multi-object tracking was running image patches of detected objects obtained with, for example, the DPM [37], through a CNN to extract features. The CNNs were pretrained on the ImageNet and PASCAL visual object classification (VOC) datasets. In one instance, the features extracted from the CNN were used to train a multi-output regularized least-squares classifier [55]. Essentially, a 4,096-dimensional feature vector is first extracted from a CNN for each detection box, followed by an application of PCA to reduce the dimensionality to 256. The classifier is used to compute a log-likelihood cost for a track hypothesis given a set of detections. This work was unique in that it showed how the classic MHT algorithm, which performs MAP inference by updating sets of track hypothesis trees in real time, compares favorably with the modern approaches described in Section 3 when augmented with learned assignment costs. In fact, at the time of publishing, their method (referred to as MHT_DAM) outperformed the second-best tracker on the 2DMOT15 by 7% in multiple object tracking accuracy (MOTA).

4.3.1 Siamese Networks. A variation on the standard CNN architecture that has seen extensive use in multi-object tracking is the Siamese network. A Siamese network processes two inputs simultaneously using multiple layers with shared weights [65] (Figure 6). These networks can be used for a variety of tasks that involve comparing two image patches; this seems intuitively useful for the task of learning assignment costs, where we are interested in predicting the association likelihood for two inputs. Indeed, a technique was proposed to directly compute association scores for pairs of image patches [65]. First, two image patches are stacked, along with their optical flow information, and fed as input into a Siamese network. A separate network learns contextual features that encode relative geometry and position variations between the two inputs, and the final layers of these two networks are extracted and combined with a gradient-boosting classifier to produce a match prediction score. Tracks are ultimately obtained by solving a network flow problem (Section 3.2.1) using linear programming.

Siamese networks have also been used to learn embeddings for pairs of detections [121]. In this work, all parameters between the two arms of the CNN are shared and the features produced by the last layer are used as input to a metric learning loss. Specifically, a multi-task loss function for incorporating temporal constraints is combined with the regularized metric learning loss to jointly optimize the weights of the deep model. They use an online learning algorithm to address the issue of changing object appearance throughout a trajectory, but the deep networks are pretrained with auxiliary data. The learned affinity model is combined with the softassign algorithm [41] to find an optimal pairing of tracklets. For the task of underwater multi-object tracking, Siamese networks were shown to improve performance as well [96]. Instead of only considering pairs of images with Siamese networks, the Quad-CNN [109] aims to learn more sophisticated representations for metric learning by considering quadruplets of images. A bounding box regression loss and a multi-task ranking loss that considers appearance and temporal similarities between four images are used to jointly optimize a Quad-CNN end-to-end. The authors propose a sliding window minimax label propagation algorithm for data association.

4.3.2 Online Appearance Adaptation. The confidence-based robust online tracking approach [4] has been extended with a deep appearance model [5] resembling a Siamese network. The features from the last CNN layer are used to compute a metric over pairs of image patches such that the metric represents a regularized energy function where the lowest possible energy gets assigned to the optimal assignment hypothesis. They employ online transfer learning to update a small number of the higher layers in the network to adapt to changing object appearances. When the average affinity scores computed by the network fall below a threshold at runtime, training samples are collected and a pass of online transfer learning is carried out to adapt the network. To help reduce the runtime overhead introduced by online learning, the authors suggest using a parallelized implementation and performing the high-confidence and low-confidence tracklet associations once every 10 timesteps as opposed to every timestep. Another efficient online algorithm for updating appearance models has been proposed where a bilinear similarity function is learned between two feature vectors with constrained convex optimization [137]. The feature vectors are also aggregated from the last layer of a CNN. Ideas from single object tracking and reinforcement learning have been adapted for online multi-object tracking [24], where a policy is learned to decide whether the target-specific tracking models should be updated with the latest detections and features at predicted locations provided obtained by ROI pooling.

4.3.3 Deep Network Flow. The network flow approach popularized by Zhang et al. [140] is revisited again from a deep learning perspective [103, 106]. Effectively, the parameters of the unary and pairwise link costs are learned end-to-end with a deep neural network. The original linear program is converted into the following bi-level optimization problem

$$\begin{aligned}&\arg\min_{\Theta} \mathcal{L}(x^{gt}, x^*)\\ &\text{s.t. } x^* = \arg\min_{x} c(f, \Theta)^\intercal x \\ &\mathbf{A}x \leq \mathbf{b}, \mathbf{C}x = 0\end{aligned} \tag{26}$$

for parameters Θ, input data f, and ground truth network flow solutions x^{gt}. The M concatenated flow variables are $x \in \mathbb{R}^M$, $\mathbf{A} = [\mathbf{I}, -\mathbf{I}]^\intercal \in \mathbb{R}^{2M\times M}$ and $\mathbf{b} = [0, 1]^\intercal \in \mathbb{R}^M$ are box constraints, and $\mathbf{C} \in \mathbb{R}^{2K\times M}$ are the flow conservation constraints. The inner optimization problem is smoothed so that it is easily solvable with an off-the-shelf convex solver. The high-level optimization problem is then solved with gradient descent. The high-level optimization problem needs ground truth network flow labels x^{gt} during training; this is handled by manually annotating bounding boxes in sequences of frames. At test time, inference is performed in a sliding window.

4.3.4 Other Approaches. A variant of the data association problem for multi-object tracking as a minimum-cost graph multi-cut problem [113] has been explored in conjunction with learned features. The key differences here with the previously discussed optimization approaches are that multiple detections at a single timestep can be attributed to the same person; in addition, it is easier to allow edges to connect across multiple timesteps in this graph to handle occlusion. The edge costs are learned with logistic regression, with features obtained from the DeepMatching [125] algorithm. DeepMatching uses a CNN that has been trained to produce dense correspondences between image patches and was notably used in the DeepFlow [125] algorithm for learning large displacement optical flow. It is also used in another multi-object tracking system to compute temporal affinities between input features [47]. Related to this is recent work on examining the interplay between semantic segmentation and multi-object tracking [17, 80, 114]. In particular, a CNN is used to segment images, and then the optical flow between segmented object pairs in consecutive images is used to define an association cost matrix [17].

A noticeable trend is a gradual drift away from developing novel optimization algorithms that solve a MDAP; rather, recent works are relying more on powerful discriminative techniques, such as using features from pretrained CNNs, and combining this with linear assignment solvers. Advances in object detection such as Faster R-CNN [97] have almost single handedly improved the performance of multi-object trackers [26]. A recent work by Bergmann et al. [10] examines this trend in detail and introduces what they refer to as a new multi-object tracking paradigm. They propose to leverage bounding box regression to handle data association with a powerful object detection CNN, and to use a feature pyramid network [71] to robustly handle objects of variable sizes. They suggest that it is worthwhile to explore the limits of object detection within multi-object tracking. Their Tracktor model achieves equivalent or convincingly stronger or performance than many state-of-the-art online trackers that use the sophisticated data association methods described in Section 3.

We would like to provide further insight into the use of CNNs pretrained on image classification datasets for generating detections and learning assignment costs. To this end, we visualized CNN layer activations using the gradient-weighted class activation mapping technique [104] in Figure 7 when asked to classify images of vehicles at a traffic intersection.

To summarize, in this section, we first explained how probabilistic cost functions for data association are formulated with kinematic and non-kinematic components. Then, we reviewed machine learning algorithms for learning association features such as boosting and metric learning. Finally, we discussed a variety of deep learning methods that mainly fine tune pretrained CNNs to predict similarity or directly compute association scores.

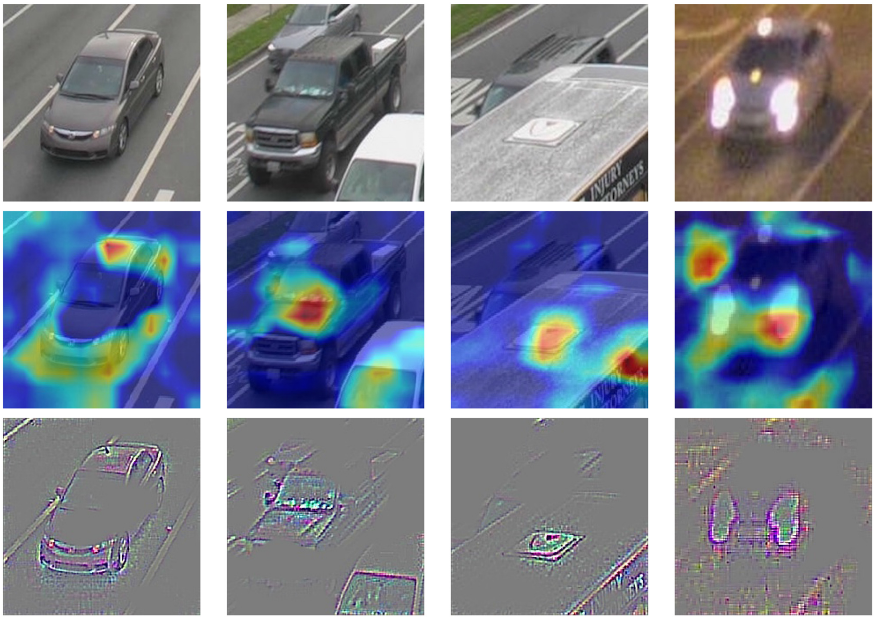


Fig. 7. Visualizations of ”important” regions for making predictions with the VGG16 network, generated with Grad-CAM [104] and pretrained VGG16 weights [107]. The first two images in the top left were correctly labeled as containing vehicles, and it can be seen that the CNN leverages interpretable features such as the car body, tires, and windshield to come to this conclusion. The CNN was not able to correctly classify the vehicles in the two images on the top right. Heavy occlusion and illumination changes still confuse a CNN if it has not been trained for these situations. The images were taken with a traffic camera by the authors. Best viewed in color.

5 EMPIRICAL COMPARISON

We have presented a large number of machine learning techniques for data association in multi-object tracking without yet addressing the question of when specific methods may be more preferable than others. We briefly touch on that topic here, focusing on reported results on the 2DMOT15 and MOT17 benchmarks. For an in-depth empirical comparison of deep learning–based multi-object trackers, we direct readers to a survey on this topic [26] and results from the recent 2018 UA-DETRAC competition [73, 126]. For reference, we have provided the results from the MOT15 and MOT17 leaderboards for methods discussed in this survey, organized by the data association method, in Tables 3 and 4.

If the tracking task has lots of labeled data available, such as pedestrian or vehicle tracking, and real-time performance is not required, currently the approach employed by Tracktor [10] of relying heavily on supervised object detection objectively performs best. It saves on the development cost incurred by sophisticated algorithms for stitching together tracklets while maintaining or exceeding their performance. Methods that learn to solve a custom linear assignment and are near real time tend to score highly (those with data association methods classified as “LA” and “E2E-LA” in the tables), emphasizing the use of deep learning for extracting both appearance and tracklet features.

If there is relatively little or no labeled data for a particular tracking task, there are certain avenues one can take besides conducting an expensive data collection and labeling effort. First, directly using features from pretrained CNNs without minimal fine tuning is still quite effective

Table 3. MOT15 Challenge Results

Tracker	DA	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID_Sw ↓	Frag ↓	HZ ↑
Tracktor++ [10]	LA	44.1	46.7	18.0	26.2	6477	26577	1318	1790	0.9
CNNTCM [121]	LA	29.6	36.8	11.2	44.0	7786	34733	712	943	1.7
RAR15pub [36]	LA	35.1	45.4	13.0	42.3	6771	32717	381	1523	5.4
AMIR15 [102]	LA	37.6	46.0	15.8	26.8	7933	29397	1026	2024	1.0
CDA_DDALpb [5]	LA	32.8	38.8	9.7	42.2	4983	35690	614	1583	2.3
RNN_LSTM [81]	E2E-LA	19.0	17.1	5.5	45.6	11578	36706	1490	2081	165.2
MDP [132]	E2E-LA	30.3	44.7	13.0	38.4	9717	32422	680	1500	1.1
MHT_DAM [55]	MHT	32.4	45.3	16.0	43.8	9064	32060	435	826	0.7
NOMT [22]	CRF	33.7	44.6	12.2	44.0	7762	32457	442	823	11.5
DCCRF [141]	CRF	33.6	39.1	10.4	37.6	5917	34002	866	1566	0.1
SiameseCNN [65]	NF	29.0	34.3	8.5	48.4	5160	37798	639	1316	52.8
TSMLCDEnew [119]	NF	34.3	44.1	14.0	39.4	7869	31908	618	959	6.5
HybridDAT [137]	NF	35.0	47.7	11.4	42.2	8455	31140	358	1267	4.6
LINF1 [35]	MCMC	24.5	34.8	5.5	64.6	5864	40207	298	744	7.5

Note: The symbols ↑ and ↓ respectively indicate that higher and lower values are preferred. LA, linear assignment; E2E-LA, end-to-end learned LA; E2E-MDAP, end-to-end MDAP; NF, network flow.

Table 4. MOT17 Challenge Results

Tracker	DA	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID_Sw ↓	Frag ↓	HZ ↑
Tracktor [10]	LA	53.5	52.3	19.5	36.6	12201	248047	2072	4611	1.5
DMAN [143]	LA	48.2	55.7	19.3	38.3	26218	263608	2194	5378	0.3
DAN [112]	E2E-LA	52.4	49.5	21.4	30.7	25423	234592	8431	14797	6.3
DeepMOT [134]	E2E-LA	48.1	43.0	17.6	38.6	26490	262578	3696	5353	4.9
MHT_DAM [55]	MHT	50.7	47.2	20.8	36.9	22875	252889	2314	2865	0.9
FAMNet [25]	E2D-MDAP	52.0	48.7	19.1	33.4	14138	253616	3072	5318	0.0

for estimating association scores. Another option is to leverage pretrained CNNs for dense correspondence or segmentation to extract flow or segmentation features as additional cues for data association [113]. Although there has been progress on end-to-end unsupervised approaches [38] (e.g., based on ability to reconstruct the scene), they are still only a promising research direction as opposed to being practically useful.

6 CONCLUSION

In this survey, we argue that viewing data association as an assignment problem helps conceptualize the large variety of data-driven techniques. We categorized many popular methods that address the combinatorial optimization and feature learning aspects of data association. One of the most exciting research directions that was discussed is the development of methods that attempt to learn the optimization algorithm and the features from data. The combinatorial nature of data association and the difficulty of learning a robust similarity metric for objects pose strong challenges, but recent work in this direction is promising. Broadly speaking, a common theme highlighted in this survey is the replacement of more and more parts of the typically cumbersome multi-object tracking pipeline with data-driven modules.

Broader impacts. Careful consideration is required when deploying these systems out in the real world. We do not yet have a perfect understanding of when data-driven systems fail, although we already know that such systems tend to reflect (potentially problematic) biases of our society stemming from, for example, the training data [28]. This is especially important to highlight due to the inherent *dual-use* nature of multi-object tracking [16]; in other words, it has the potential to be used by both benevolent and malicious actors. As smart surveillance systems are increasingly deployed in cities, it is important to be transparent about the capabilities and limitations of current and near-future multi-object tracking. However, there are many beneficial uses cases and outcomes for multi-object tracking, such as reducing traffic fatalities, monitoring endangered species, and improving real-time sports analysis.

REFERENCES

[1] Noor M. Al-Shakarji, Filiz Bunyak, Guna Seetharaman, and Kannappan Palaniappan. 2018. Multi-object tracking cascade with multi-step data association and occlusion handling. In *Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'18)*. IEEE, Los Alamitos, CA, 1–6.

[2] Thiemo Alldieck, Chris H. Bahnsen, and Thomas B. Moeslund. 2016. Context-aware fusion of RGB and thermal imagery for traffic monitoring. *Sensors* 16, 11 (2016), 1947.

[3] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. 2010. Monocular 3D pose estimation and tracking by detection. In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*. IEEE, Los Alamitos, CA, 623–630.

[4] Seung-Hwan Bae and Kuk-Jin Yoon. 2014. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14)*. 1218–1225.

[5] Seung-Hwan Bae and Kuk-Jin Yoon. 2017. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 3 (2017), 595–610.

[6] Yaakov Bar-Shalom and Huimin Chen. 2004. Multisensor track-to-track association for tracks with dependent errors. In *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC'04)*, Vol. 3. IEEE, Los Alamitos, CA, 2674–2679.

[7] Ben Benfold and Ian Reid. 2011. Stable multi-target tracking in real-time surveillance video. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. IEEE, Los Alamitos, CA, 3457–3464.

[8] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. 2018. Machine learning for combinatorial optimization: A methodological tour d'horizon. arXiv:1811.06128.

[9] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. 2011. Multiple object tracking using *k*-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 9 (2011), 1806–1819.

[10] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. 2019. Tracking without bells and whistles. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'19)*.

[11] Keni Bernardin and Rainer Stiefelhagen. 2008. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing* 2008, 1 (2008), Article 246309.

[12] Dimitri P. Bertsekas. 1992. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications* 1, 1 (1992), 7–66.

[13] Samuel Blackman and Robert Popoli. 1999. *Design and Analysis of Modern Tracking Systems*. Artech House.

[14] Vladimir L. Boginski, Clayton W. Commander, Panos M. Pardalos, and Yinyu Ye. 2011. *Sensors: Theory, Algorithms, and Applications*. Vol. 61. Springer Science & Business Media.

[15] Yuri Boykov and Vladimir Kolmogorov. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 9 (2004), 1124–1137.

[16] Miles Brundage, Shahar Avin, Jack Clark, Helen Toner, Peter Eckersley, Ben Garfinkel, Allan Dafoe, et al. 2018. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. arXiv:1802.07228.

[17] Sebastian Bullinger, Christoph Bodensteiner, and Michael Arens. 2017. Instance flow based online multiple object tracking. In *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP'17)*. IEEE, Los Alamitos, CA, 785–789.

[18] Asad A. Butt and Robert T. Collins. 2013. Multi-target tracking by Lagrangian relaxation to min-cost network flow. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*. IEEE, Los Alamitos, CA, 1846–1853.

[19] A. Caponi. 2004. Polynomial time algorithm for data association problem in multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems* 40, 4 (2004), 1398–1410.

[20] Lei Chen, Martin J. Wainwright, Müjdat Cetin, and Alan S. Willsky. 2006. Data association based on optimization in graphical models with application to sensor networks. *Mathematical and Computer Modelling* 43, 9 (2006), 1114–1135.

[21] Zhexu Chena, Lei Chen, Mujdat Cetin, and Alan S. Willsky. 2009. An efficient message passing algorithm for multi-target tracking. In *Proceedings of the 12th International Conference on Information Fusion (FUSION'09)*. IEEE, Los Alamitos, CA, 826–833.

[22] Wongun Choi. 2015. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*. 3029–3037.

[23] Chee-Yee Chong. 2012. Graph approaches for data association. In *Proceedings of the 15th International Conference on Information Fusion (FUSION'12)*. IEEE, Los Alamitos, CA, 1578–1585.

[24] Peng Chu, Heng Fan, Chiu C. Tan, and Haibin Ling. 2019. Online multi-object tracking with instance-aware tracker and dynamic model refreshment. In *Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV'19)*. IEEE, Los Alamitos, CA, 161–170.

[25] Peng Chu and Haibin Ling. 2019. FAMNet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *Proceedings of the 2019 IEEE International Conference on Computer Vision (ICCV'19)*.

[26] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera. 2019. Deep learning in video multi-object tracking: A survey. arXiv:1907.12740.

[27] Robert T. Collins. 2012. Multitarget data association with higher-order motion models. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. IEEE, Los Alamitos, CA, 1744–1751.

[28] David Danks and Alex John London. 2017. Algorithmic bias in autonomous systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 4691–4697.

[29] Somnath Deb, Krishna R. Pattipati, and Yaakov Bar-Shalom. 1993. A multisensor-multitarget data association algorithm for heterogeneous sensors. *IEEE Transactions on Aerospace and Electronic Systems* 29, 2 (1993), 560–568.

[30] Somnath Deb, Murali Yeddanapudi, Krishna Pattipati, and Yaakov Bar-Shalom. 1997. A generalized SD assignment algorithm for multisensor-multitarget state estimation. *IEEE Transactions on Aerospace and Electronic Systems* 33, 2 (1997), 523–538.

[31] Julie Dequaire, Peter Ondruska, Dushyant Rao, Dominic Wang, and Ingmar Posner. 2017. Deep tracking in the wild: End-to-end tracking using recurrent neural networks. *International Journal of Robotics Research* 37 (2017), 495–512.

[32] Soufiene Djahel, Nafaa Jabeur, Robert Barrett, and John Murphy. 2015. Toward V2I communication technology-based solution for reducing road traffic congestion in smart cities. In *Proceedings of the 2015 International Symposium on Networks, Computers, and Communications (ISNCC'15)*. IEEE, Los Alamitos, CA, 1–6.

[33] Anna Ellis and James Ferryman. 2010. PETS2010: Dataset and challenge. In *Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'10)*. IEEE, Los Alamitos, CA, 135–142.

[34] A. Ess, B. Leibe, K. Schindler, and L. van Gool. 2008. A mobile vision system for robust multi-person tracking. In *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. IEEE, Los Alamitos, CA.

[35] Loic Fagot-Bouquet, Romaric Audigier, Yoann Dhome, and Frederic Lerasle. 2016. Improving multi-frame data association with sparse representations for robust near-online multi-object tracking. In *Proceedings of the 14th European Conference on Computer Vision (ECCV'16)*. 774–790.

[36] Kuan Fang, Yu Xiang, and Silvio Savarese. 2017. Recurrent autoregressive networks for online multi-object tracking. arXiv:1711.02741.

[37] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. 2010. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 9 (2010), 1627–1645.

[38] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. 2018. Tracking by prediction: A deep generative model for multi-person localisation and tracking. In *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV'18)*. IEEE, Los Alamitos, CA, 1122–1132.

[39] Yoav Freund and Robert E. Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the European Conference on Computational Learning Theory*. 23–37.

[40] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*.

[41] Steven Gold and Anand Rangarajan. 1996. Softmax to softassign: Neural network algorithms for combinatorial optimization. *Journal of Artificial Neural Networks* 2, 4 (1996), 381–399.

[42] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2672–2680.

[43] Don A. Grundel, Pavlo A. Krokhmal, Carlos A. S. Oliveira, and Panos M. Pardalos. 2007. On the number of local minima for the multidimensional assignment problem. *Journal of Combinatorial Optimization* 13, 1 (2007), 1–18.

[44] John M. Hammersley and Peter Clifford. 1971. Markov fields on finite graphs and lattices. Unpublished manuscript.

[45] Zhen He, Jian Li, Daxue Liu, Hangen He, and David Barber. 2019. Tracking by animation: Unsupervised learning of multi-object attentive trackers. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 1318–1327.

[46] Alexandre Heili, Adolfo Lopez-Mendez, and Jean-Marc Odobez. 2014. Exploiting long-term connectivity and visual motion in CRF-based multi-person tracking. *IEEE Transactions on Image Processing* 23, 7 (2014), 3040–3056.

[47] Roberto Henschel, Laura Leal-Taixé, Daniel Cremers, and Bodo Rosenhahn. 2017. Improvements to Frank-Wolfe optimization for multi-detector multi-object tracking. arXiv:1705.08314.

[48] John J. Hopfield and David W. Tank. 1985. Neural computation of decisions in optimization problems. *Biological Cybernetics* 52, 3 (1985), 141–152.

[49] Chang Huang, Bo Wu, and Ramakant Nevatia. 2008. Robust object tracking by hierarchical association of detection responses. In *Proceedings of the 10th European Conference on Computer Vision (ECCV'08)*. 788–801.

[50] Hao Jiang, Sidney Fels, and James J. Little. 2007. A linear programming approach for multiple object tracking. In *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*. IEEE, Los Alamitos, CA, 1–8.

[51] Jindong Jiang, Sepehr Janghorbani, Gerard de Melo, and Sungjin Ahn. 2019. Scalable object-oriented sequential generative models. arXiv:1910.02384.

[52] Jean-Philippe Jodoin, Guillaume-Alexandre Bilodeau, and Nicolas Saunier. 2016. Tracking all road users at multimodal urban traffic intersections. *IEEE Transactions on Intelligent Transportation Systems* 17, 11 (2016), 3241–3251.

[53] Alla R. Kammerdiner. 2008. Multidimensional assignment problem. In *Encyclopedia of Optimization* (2nd ed.), C. A. Floudas and P. M. Pardalos (Eds.). Springer, 2396–2402.

[54] Lance M. Kaplan, Yaakov Bar-Shalom, and William D. Blair. 2008. Assignment costs for multiple sensor track-to-track association. *IEEE Transactions on Aerospace and Electronic Systems* 44, 2 (2008), 655–677.

[55] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M. Rehg. 2015. Multiple hypothesis tracking revisited. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*. 4696–4704.

[56] Suna Kim, Suha Kwak, Jan Feyereisl, and Bohyung Han. 2013. *Online Multi-Target Tracking by Large Margin Structured Learning*. Springer, Berlin, Germany, 98–111. https://doi.org/10.1007/978-3-642-37431-9_8

[57] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational Bayes. arXiv:1312.6114.

[58] Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. 2018. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*. 8606–8616.

[59] H. W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 1–2 (1955), 83–97. http://dx.doi.org/10.1002/nav.3800020109

[60] Cheng-Hao Kuo, Chang Huang, and Ramakant Nevatia. 2010. Multi-target tracking by on-line learned discriminative appearance models. In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*. IEEE, Los Alamitos, CA, 685–692.

[61] Cheng-Hao Kuo and Ram Nevatia. 2011. How does person identity recognition help multi-person tracking? In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. IEEE, Los Alamitos, CA, 1217–1224.

[62] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*. 282–289.

[63] Roslyn A. Lau and Jason L. Williams. 2011. Multidimensional assignment by dual decomposition. In *Proceedings of the 2011 7th International Conference on Intelligent Sensors, Sensor Networks, and Information Processing (ISSNIP'11)*. IEEE, Los Alamitos, CA, 437–442.

[64] Nam Le, Alexander Heili, and Jean-Marc Odobez. 2016. Long-term time-sensitive costs for CRF-based tracking by detection. In *Proceedings of the 14th European Conference on Computer Vision (ECCV'16)*. 43–51.

[65] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. 2016. Learning by tracking: Siamese CNN for robust target association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 33–40.

[66] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. 2015. MOTChallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942.

[67] Laura Leal-Taixé, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, and Stefan Roth. 2017. Tracking the trackers: An analysis of the state of the art in multiple object tracking. arXiv:1704.02781.

[68] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel. 2013. A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology* 4, 4 (2013), 58.

[69] Yuan Li, Chang Huang, and Ram Nevatia. 2009. Learning to associate: HybridBoosted multi-target tracker for crowded scene. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE, Los Alamitos, CA, 2953–2960.

[70] Liang Liang, Hongying Shen, Panteleimon Rompolas, Valentina Greco, Pietro De Camilli, and James S. Duncan. 2013. *A multiple hypothesis based method for particle tracking and its extension for cell segmentation. Information Processing in Medical Imaging* 23 (2013), 98–109. https://doi.org/10.1007/978-3-642-38868-2_9

[71] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 2117–2125.

[72] Wenhan Luo, Junliang Xing, Xiaoqin Zhang, Xiaowei Zhao, and Tae-Kyun Kim. 2014. Multiple object tracking: A literature review. arXiv:1409.7618[cs.CV].

[73] Siwei Lyu, Ming-Ching Chang, Dawei Du, Wenbo Li, Yi Wei, Marco Del Coco, Pierluigi Carcagnì, et al. 2018. UA-DETRAC 2018: Report of AVSS2018 & IWT4S challenge on advanced traffic monitoring. In *Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'18)*. IEEE, Los Alamitos, CA, 1–6.

[74] Cong Ma, Changshui Yang, Fan Yang, Yueqing Zhuang, Ziwei Zhang, Huizhu Jia, and Xiaodong Xie. 2018. Trajectory factory: Tracklet cleaving and re-connection by deep Siamese bi-GRU for multiple object tracking. In *Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME'18)*. IEEE, Los Alamitos, CA, 1–6.

[75] Ronald P. S. Mahler. 2007. *Statistical Multisource-Multitarget Information Fusion.* Artech House.

[76] Daniel Meissner, Stephan Reuter, and Klaus Dietmayer. 2012. Real-time detection and tracking of pedestrians at intersections using a network of laserscanners. In *Proceedings of the 2012 IEEE Intelligent Vehicles Symposium (IV'12)*. IEEE, Los Alamitos, CA, 630–635.

[77] Florian Meyer, Paolo Braca, Peter Willett, and Franz Hlawatsch. 2016. Tracking an unknown number of targets using multiple sensors: A belief propagation method. In *Proceedings of the 19th International Conference on Information Fusion (FUSION'16)*. IEEE, Los Alamitos, CA, 719–726.

[78] Florian Meyer, Paolo Braca, Peter Willett, and Franz Hlawatsch. 2017. A scalable algorithm for tracking an unknown number of targets using multiple sensors. *IEEE Transactions on Signal Processing* 65, 13 (2017), 3478–3493.

[79] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. 2016. MOT16: A benchmark for multi-object tracking. arXiv:1603.00831[cs.CV].

[80] Anton Milan, Laura Leal-Taixé, Konrad Schindler, and Ian Reid. 2015. Joint tracking and segmentation of multiple targets. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*. 5397–5406.

[81] Anton Milan, Seyed Hamid Rezatofighi, Anthony R. Dick, Ian D. Reid, and Konrad Schindler. 2017. Online multi-target tracking using recurrent neural networks. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*. 4225–4232.

[82] Anton Milan, Konrad Schindler, and Stefan Roth. 2016. Multi-target tracking by discrete-continuous energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 10 (2016), 2054–2068.

[83] James Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics* 5, 1 (1957), 32–38.

[84] Robert A. Murphey, Panos M. Pardalos, and Leonidas S. Pitsoulis. 1997. A greedy randomized adaptive search procedure for the multitarget multisensor tracking problem. *Network Design: Connectivity and Facilities Location* 40 (1997), 277–302.

[85] Songhwai Oh, Stuart Russell, and Shankar Sastry. 2004. Markov chain Monte Carlo data association for general multiple-target tracking problems. In *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC'04)*, Vol. 1. IEEE, Los Alamitos, CA, 735–742.

[86] Peter Ondruska and Ingmar Posner. 2016. Deep tracking: Seeing beyond seeing using recurrent neural networks. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*. 3361–3367.

[87] Richard W. Osbome, Yaakov Bar-Shalom, and Peter Willett. 2011. Track-to-track association with augmented state. In *Proceedings of the 14th International Conference on Information Fusion (FUSION'11)*. IEEE, Los Alamitos, CA, 1–8.

[88] Aljosa Osep, Wolfgang Mehner, Markus Mathias, and Bastian Leibe. 2017. Combined image-and world-space tracking in traffic scenes. In *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA'17)*. IEEE, Los Alamitos, CA, 1988–1995.

[89] Hanna Pasula, Stuart Russell, Michael Ostland, and Yaacov Ritov. 1999. Tracking many objects with many sensors. In *Proceedings of the 1999 International Joint Conference on Artificial Intelligence (IJCAI'99)*, Vol. 99. 1160–1171.

[90] Federico Perea and Huub W. De Waard. 2011. Greedy and *k*-greedy algorithms for multidimensional data association. *IEEE Transactions on Aerospace and Electronic Systems* 47, 3 (2011), 1915–1925.

[91] Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes. 2011. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. IEEE, Los Alamitos, CA, 1201–1208.

[92] Aubrey B. Poore. 1994. Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. *Computational Optimization and Applications* 3, 1 (1994), 27–57.

[93] Aubrey B. Poore and Sabino Gadaleta. 2006. Some assignment problems arising from multiple target tracking. *Mathematical and Computer Modelling* 43, 9 (2006), 1074–1091.

[94] Robert L. Popp, Krishna R. Pattipati, and Yaakov Bar-Shalom. 2001. M-best SD assignment algorithm with application to multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems* 37, 1 (2001), 22–39.

[95] Robert L. Popp, Krishna R. Pattipati, Yaakov Bar-Shalom, and Richard R. Gassner. 1998. An adaptive m-best SD assignment algorithm and parallelization for multitarget tracking. In *Proceedings of the IEEE Aerospace Conference*, Vol. 5. IEEE, Los Alamitos, CA, 71–84.

[96] M. V. Rahul, Revanur Ambareesh, and G. Shobha. 2017. Siamese network for underwater multiple object tracking. In *Proceedings of the 9th International Conference on Machine Learning and Computing*. ACM, New York, NY, 511–516.

[97] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*. 91–99.

[98] Mauricio G. C. Resende and Celso C. Ribeiro. 2016. *Optimization by GRASP*. Springer, New York, NY. DOI: https://doi.org/10.1007/978-1-4939-6530-4

[99] Stephan Reuter, Andreas Danzer, Manuel Stubler, Alexander Scheel, and Karl Granstrom. 2017. A fast implementation of the labeled multi-Bernoulli filter using Gibbs sampling. In *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV'17)*. IEEE, Los Alamitos, CA, 765–772.

[100] Danilo Jimenez Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. arXiv:1505.05770.

[101] Arunesh Roy, Nicholas Gale, and Lang Hong. 2011. Automated traffic surveillance using fusion of Doppler radar and video information. *Mathematical and Computer Modelling* 54, 1 (2011), 531–543.

[102] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. 2017. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV'17)*. 300–311.

[103] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. 2017. Deep network flow for multi-object tracking. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 6951–6960.

[104] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV'17)*.

[105] Samuel A. Shapero, Hunter Hughes, and Peter Tuuk. 2016. Adaptive semi-greedy search for multidimensional track assignment. In *Proceedings of the 19th International Conference on Information Fusion (FUSION'16)*. IEEE, Los Alamitos, CA, 409–415.

[106] Han Shen, Lichao Huang, Chang Huang, and Wei Xu. 2018. Tracklet association tracker: An end-to-end learning-based association approach for multi-object tracking. arXiv:1808.01562.

[107] K. Simonyan and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556[cs.CV].

[108] Kate A. Smith. 1999. Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing* 11, 1 (1999), 15–34.

[109] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. 2017. Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 5620–5629.

[110] Alexey Sorokin, Nikita Boyko, Vladimir Boginski, Stan Uryasev, and Panos M. Pardalos. 2009. Mathematical programming techniques for sensor networks. *Algorithms* 2, 1 (2009), 565–581.

[111] Elias Strigel, Daniel Meissner, Florian Seeliger, Benjamin Wilking, and Klaus Dietmayer. 2014. The Ko-PER intersection laserscanner and video dataset. In *Proceedings of the 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC'14)*. IEEE, Los Alamitos, CA, 1900–1901.

[112] ShiJie Sun, Naveed Akhtar, HuanSheng Song, Ajmal S. Mian, and Mubarak Shah. 2019. Deep affinity network for multiple object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Early Access. July 19, 2019. DOI: https://doi.org/10.1109/tpami.2019.2929520

[113] Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. 2016. Multi-person tracking by multicut and deep matching. In *Proceedings of the 14th European Conference on Computer Vision (ECCV'16)*. 100–111.

[114] Yicong Tian and Mubarak Shah. 2016. On duality of multiple target tracking and segmentation. arXiv:1610.04542[cs.CV].

[115] Sjoerd Van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. 2018. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *Proceedings of the 2018 International Conference on Learning Representations (ICLR'18)*.

[116] Ba-Ngu Vo, Ba-Tuong Vo, and Hung Gia Hoang. 2017. An efficient implementation of the generalized labeled multi-bernoulli filter. *IEEE Transactions on Signal Processing* 65, 8 (2017), 1975–1987.

[117] Martin Wainwright, Tommi Jaakkola, and Alan Willsky. 2002. MAP estimation via agreement on (hyper) trees: Message-passing and linear programming approaches. In *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, Vol. 40. 1565–1575.

[118] Jose L. Walteros, Chrysafis Vogiatzis, Eduardo L. Pasiliao, and Panos M. Pardalos. 2014. Integer programming models for the multidimensional assignment problem with star costs. *European Journal of Operational Research* 235, 3 (2014), 553–568.

[119] Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang. 2017. Tracklet association by online target-specific metric learning and coherent dynamics estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 3 (2017), 589–602.

[120] Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang. 2014. Tracklet association with online target-specific metric learning. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14)*. 1234–1241.

[121] Bing Wang, Li Wang, Bing Shuai, Zhen Zuo, Ting Liu, Kap Luk Chan, and Gang Wang. 2016. Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR'16)*. 1–8.

[122] Shaofei Wang and Charless C. Fowlkes. 2015. Learning optimal parameters for multi-target tracking. In *Proceedings of the British Machine Vision Conference (BMVC'15)*, Vol. 1. 4.1–4.13.

[123] Shaofei Wang and Charless C. Fowlkes. 2017. Learning optimal parameters for multi-target tracking with contextual interactions. *International Journal of Computer Vision* 122, 3 (2017), 484–501.

[124] Xiaoyu Wang, Gang Hua, and Tony X. Han. 2010. Discriminative tracking by metric learning. In *Proceedings of the 11th European Conference on Computer Vision (ECCV'10)*. 200–214.

[125] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. 2013. DeepFlow: Large displacement optical flow with deep matching. In *Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV'13)*. 1385–1392.

[126] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. 2015. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. arXiv:1511.04136[cs.CV].

[127] Jason L. Williams and Roslyn A. Lau. 2010. Convergence of loopy belief propagation for data association. In *Proceedings of the 6th International Conference on Intelligent Sensors, Sensor Networks, and Information Processing (ISSNIP'10)*. IEEE, Los Alamitos, CA, 175–180.

[128] Jason L. Williams and Roslyn A. Lau. 2010. Data association by loopy belief propagation. In *Proceedings of the 13th International Conference on Information Fusion (FUSION'10)*. IEEE, Los Alamitos, CA, 1–8.

[129] Jason L. Williams and Roslyn A. Lau. 2014. Approximate evaluation of marginal association probabilities with belief propagation. *IEEE Transactions on Aerospace and Electronic Systems* 50, 4 (2014), 2942–2959.

[130] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP'17)*. IEEE, Los Alamitos, CA, 3645–3649. DOI: https://doi.org/10.1109/ICIP.2017.8296962

[131] Zheng Wu, Ashwin Thangali, Stan Sclaroff, and Margrit Betke. 2012. Coupling detection and data association for multiple object tracking. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. IEEE, Los Alamitos, CA, 1948–1955.

[132] Yu Xiang, Alexandre Alahi, and Silvio Savarese. 2015. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*. 4705–4713.

[133] Eric P. Xing, Michael I. Jordan, Stuart J. Russell, and Andrew Y. Ng. 2003. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems*. 521–528.

[134] Yihong Xu, Yutong Ban, Xavier Alameda-Pineda, and Radu Horaud. 2019. DeepMOT: A differentiable framework for training multiple object trackers. arXiv:1906.06618.

[135] Bo Yang, Chang Huang, and Ram Nevatia. 2011. Learning affinities and dependencies for multi-target tracking using a CRF model. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. IEEE, Los Alamitos, CA, 1233–1240.

[136] Bo Yang and Ram Nevatia. 2012. An online learned CRF model for multi-target tracking. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. IEEE, Los Alamitos, CA, 2034–2041.

[137] Min Yang, Yuwei Wu, and Yunde Jia. 2017. A hybrid data association framework for robust online multi-object tracking. *IEEE Transactions on Image Processing* 26, 12 (2017), 5667–5679.

[138] Yuebin Yang and Guillaume-Alexandre Bilodeau. 2017. Multiple object tracking with kernelized correlation filters in urban mixed traffic. In *Proceedings of the 2017 14th Conference on Computer and Robot Vision (CRV'17)*. IEEE, Los Alamitos, CA, 209–216.

[139] Alper Yilmaz, Omar Javed, and Mubarak Shah. 2006. Object tracking: A survey. *ACM Computing Surveys* 38, 4 (2006), 13.

[140] Li Zhang, Yuan Li, and Ramakant Nevatia. 2008. Global data association for multi-object tracking using network flows. In *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. IEEE, Los Alamitos, CA, 1–8.

[141] Hui Zhou, Wanli Ouyang, Jian Cheng, Xiaogang Wang, and Hongsheng Li. 2018. Deep continuous conditional random fields with asymmetric inter-object constraints for online multi-object tracking. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 4 (2018), 1011–1022.

[142] Hongyan Zhu, Chongzhao Han, and Chen Li. 2007. Graphical models-based track association algorithm. In *Proceedings of the 10th International Conference on Information Fusion (FUSION'07)*. IEEE, Los Alamitos, CA, 1–8.

[143] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. 2018. Online multi-object tracking with dual matching attention networks. In *Proceedings of the 2018 European Conference on Computer Vision (ECCV'18)*. 366–382.

Received February 2018; revised March 2020; accepted April 2020