Machine Learning Methods for Solving Assignment Problems in Multi-Target Tracking

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

Data association and track-to-track association, two fundamental problems in single-sensor and multi-sensor multi-target tracking, are instances of an NP-hard combinatorial optimization problem known as the multidimensional assignment problem (MDAP). Over the last few years, data-driven approaches to tackling MDAPs in tracking have become increasingly popular. We argue that viewing multi-target tracking as an assignment problem conceptually unifies the wide variety of machine learning methods that have been proposed for data association and track-to-track association. In this survey, we review recent literature, provide rigorous formulations of the assignment problems encountered in multi-target tracking, and review classic approaches used prior to the shift towards data-driven techniques. Recent attempts at using deep learning to solve NP-hard combinatorial optimization problems, including data association, are discussed as well. We highlight representation learning methods for multi-sensor applications and conclude by providing an overview of current multi-target tracking benchmarks.

CCS Concepts: • Computing methodologies → Tracking;

Additional Key Words and Phrases: multi-target tracking; multidimensional assignment; machine learning; deep learning

ACM Reference Format:

Patrick Emami, Panos M. Pardalos, Lily Elefteriadou, and Sanjay Ranka. 2018. Machine Learning Methods for Solving Assignment Problems in Multi-Target Tracking. 1, 1 (February 2018), 35 pages. https://doi.org/000001.0000001

1 INTRODUCTION

1.1 Background

Multi-target 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 an accurate picture of the surrounding environment. Within ITS, it can be used for pedestrian detection at intersections [81], self-driving cars [97], and for traffic surveillance [110] [1] [146] [60]. Multi-target tracking also has a myriad of other applications ranging from general security systems to tracking cells in microscopy images [77]. There are many different sensor modalities that can be used for these applications; the most common are

Authors' addresses: Patrick Emami, University of Florida, 432 Newell Dr, Gainesville, FL, 32611, USA, pemami@ufl.edu; Panos M. Pardalos, University of Florida, 401 Weil Hall, Gainesville, FL, 32611, USA, p.m.pardalos@gmail.com; Lily Elefteriadou, University of Florida, 512 Weil Hall, Gainesville, FL, 32611, USA, elefter@ce.ufl.edu; Sanjay Ranka, University of Florida, 432 Newell Dr, Gainesville, FL, 32611, USA, sanjayranka@gmail.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

XXXX-XXXX/2018/2-ART \$15.00

https://doi.org/0000001.0000001

:2 P. Emami et al.

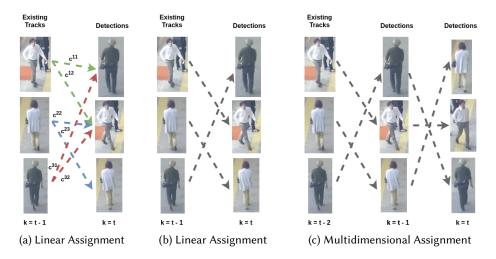


Fig. 1. A visual depiction of data association. a) In online tracking, new sensor detections are matched to existing tracks at each time step by solving a linear assignment problem. The assignment hypotheses are the colored, dashed arrows. Each arrow is annotated with the cost c^{ij} of associating track 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 [85].

video, radar, and LiDAR. As a motivating example, consider a vision system that tracks all traffic participants at an urban intersection. The real-time tracking data can be used for adaptive traffic signal control to optimize the flow of traffic at that intersection. However, urban traffic intersections contain numerous challenges for multi-target 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. On the other hand, 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 [34]. However, tall buildings, trees, and other vehicles can increase GPS signal interference, a phenomenon known as multipath, that can corrupt the data [37]. Identifying and filtering out the effects of multipath is still an ongoing area of research [23].

Prior to the proliferation of vision-based tracking, tracking methods primarily relied on kinematic data. A sensible intuition is that combining kinematic information with the learned representations of high-dimensional sensor data will improve tracking performance. The aim of this survey is to review the algorithms used in data-driven multi-target tracking and discuss recently proposed extensions. We believe that considering tracking from the perspective of an assignment problem is a good way to abstract away a lot of application-specific details and unify the many different approaches.

1.2 Assignment Problems in Multi-Target Tracking

At the core of multi-target tracking lies the data association and track-to-track association problems. The goal of data association is to identify a correspondence between a collection of new sensor

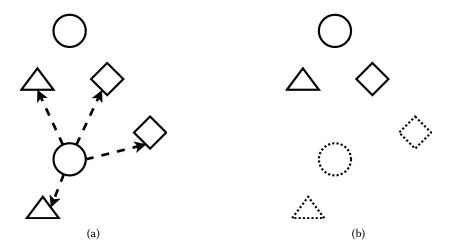


Fig. 2. a) There are three different sensors (circles, triangles, and diamonds) covering the surveillance region, each maintaining two tracks. In this scenario, the sliding window is of size T=1. The dashed arrows in (a) depict the assignment hypothesis corresponding to associating one of the tracks from the circle sensor with tracks from the triangle and diamond sensors. We don't show the other arrows that originate from each track in (a) for visual clarity. b) The best track-to-track association hypothesis; the two groups of tracks associated together are indicated by the solid and dashed lines. The solution effectively partitions each sensor's track lists.

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 of the same target that were generated by different sensors (Figure 2). This is a necessary step before track-to-track fusion; 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. Note that in this work, we use *detections* and *measurements* interchangeably; similarly, we equivocate *targets* with the term *objects*. We will attempt to be as consistent as possible with our usage while also adhering to the norms of the different tracking communities when appropriate. For example, in vision-based tracking, the term *detections* is typically used instead of *measurements*.

Broadly speaking, algorithms for solving these two association tasks can be classified as either *single-scan* or *multi-scan*. A single-scan algorithm only uses track information from the most recent time step, whereas multi-scan algorithms use track information from multiple previous or future time steps. 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 methods are usually superior, and they offer fewer track ID switches, track breaks, and missed targets [102]. Naturally, multi-scan methods are more computationally expensive and difficult to implement than their single-scan counterparts.

:4 P. Emami et al.

Table 1. Taxonomy of assignment problems in multi-target tracking. LAP := linear assignment problem and MDAP := multidimensional assignment problem. The algorithms presented in this survey are for solving the various MDAPs encountered in multi-target tracking, and are generally applicable (with modification) to both data association and track-to-track association.

	Data Association	Track-to-Track Association
U	LAP (1-2 sensors), MDAP (\geq 3 sensors) MDAP (\geq 1 sensors)	LAP (2 sensors), MDAP (\geq 3 sensors) MDAP (\geq 2 sensors)

A common way to formulate these association tasks is as an assignment problem. See Table 1 for a categorization of the various association tasks mapped onto assignment problems. The simplest version is the 2D assignment problem, also known as bipartite matching or linear assignment (LAP), which seeks to match m workers, e.g., tracks, to n jobs, e.g., sensor measurements. This combinatorial optimization problem constrains the space of solutions 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). The multidimensional extension to the assignment problem for track-to-track association stipulates that each track from each sensor be assigned exactly once. For multidimensional data association, constraints ensure that each sensor measurement at each time step is assigned to a track exactly once. Unfortunately, the MDAP is NP-hard for dimensions ≥ 3 , whereas there exists many polynomial-time algorithms for the LAP. We will formulate these problems more rigorously in Section 2. The algorithms presented in this survey are for solving the various MDAPs encountered in multi-target tracking, and are generally applicable (with modification) to both data association and track-to-track association.

It has been suggested that non-kinematic data obtained from sensors can be incorporated into association algorithms to improve performance [7] [96] [90] [26]. For example, a classifier can be used to prevent two sensor tracks with different target class labels from being associated, which reduces the number of potential assignments. Appearance information has been used extensively in the computer vision community to improve the performance of data association; see [75] for an in-depth survey. We will be discussing data-driven approaches for discovering features to augment association algorithms. Additionally, we will survey optimization algorithms for finding the solution to a MDAP.

1.3 Comparison with Related Surveys

There are several related surveys to ours, and we wish to highlight the relationship between the contributions of these surveys and those of our own. Both [101] and [102] provide a detailed treatment of how assignment problems are useful for multi-target tracking. They only go so far as to frame assignment problems in the context of multi-target tracking. There are a number of excellent general surveys on multi-target tracking [79] [147]; however, the scope of these studies is limited to only vision-based tracking and the focus is on all aspects of a tracking solution, whereas our focus is specifically on the data association and track-to-track association problems. The survey on appearance matching in camera-based multi-target tracking [75] 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. Finally, [42] surveys recent advances in applying machine learning techniques to graph matching, but the connection to multi-sensor multi-target tracking is not mentioned.

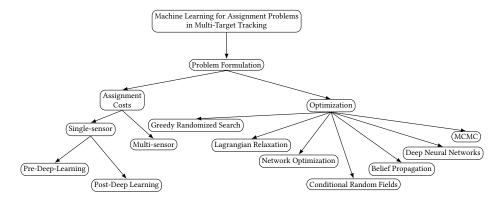


Fig. 3. Organization of the survey. We begin by surveying a wide variety of optimization techniques, followed by learning algorithms for assignment costs.

1.4 Roadmap

The presentation of the techniques for solving MDAPs is split into two parts. The first part is focused on the optimization problem of finding an assignment for data association and track-to-track association, and the second part is concerned with methods for learning the assignment costs from data. Hence, the rest of the survey will be organized in the following manner. In Section 2, the various assignment problems in multi-target tracking are carefully defined for the reader. Section 3 begins with a presentation of techniques for solving MDAPs in multi-target tracking that were proposed early on but still remain relevant today, followed by an examination of machine learning-based approaches that are now more prominent. In Section 4, we present multiple methods for learning assignment costs in single-sensor and multi-sensor tracking from data. Section 5 provides a brief overview of available datasets for single-sensor and multi-sensor multi-target tracking, with emphasis placed on ITS applications. Finally, Section 6 concludes with a discussion on future research directions. For a visual representation of the organization of the survey, see Figure 3.

2 PROBLEM FORMULATION

We will first formally introduce the linear assignment problem (LAP) for single-sensor data association and track-to-track association with two sensors. Following this, we will examine the various MDAP formulations.

2.1 Linear Assignment

Consider a scenario where there are m existing tracks and n new sensor measurements at time k, k=1,...,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 (Figures 1a and 1b). 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

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

P. Emami et al.

with constraints

$$\sum_{i=1}^{m} x^{ij} = 1, \quad j = 1, ..., n$$

$$\sum_{j=1}^{n} x^{ij} = 1, \quad i = 1, ..., m$$
(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 $(m+1)^{th}$ row represent the costs of classifying measurements as false alarms. Missed detections are usually handled by forming validation gates around the m tracks (see [12], Section 6.3). 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 mn! possible assignments, many polynomial-time algorithms exist for finding the globally optimal assignment matrix. Most famous is the $O(n^3)$ Hungarian, or Kuhn-Munkres, algorithm [66] [91]. Another popular method is the Auction algorithm, introduced by Bertsekas in [11]. These algorithms are fast and are easy to integrate into real-time multitarget tracking solutions. However, by only considering the previous time step 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 future track states to construct assignment costs that model the relationship between tracks and new sensor measurements more accurately. Instead of updating the assignment within the sliding window at each time step, an alternative approach is to simply delay making a decision within the sliding window. In the sequel, we describe how this affects the formulation of the assignment problem. As indicated in Table 1, the single-scan track-to-track association problem with two sensors is also a LAP, where m and nrepresent 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, i.e., $m \neq n$. If the assignment costs are known, an optimal track assignment can be found in polynomial-time using one of the previously mentioned algorithms.

2.2 Multidimensional Assignment

Within the single-sensor and multi-sensor tracking paradigms, there are a few different ways to formulate data association 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. For further reading on the MDAP, see [61], [101], [12].

We begin by considering the MDAP for multi-scan data association with one sensor. This scenario is the one most commonly encountered, especially in video-based tracking. 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 (Figures 1c). When $T \ge 2$, the assignment problem is NP-hard [61].

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 ith measurement of scan k, $i = 1, ..., M_k$. 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 $Z^T = \{Z_1, ..., Z_T\}$ 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 $\{z_1^i, z_2^i, ..., z_T^i\}$; one measurement from each scan at each time step 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 $c_{i_1,i_2,...,i_T}$, where the indices $i_1,i_2,...,i_T$ 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 multidimensional 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 $\rho_{i_1,i_2,...,i_T} \in \{0,1\}$ to indicate if a track is present in a partition, then we can represent the MDAP objective as

$$\min_{\gamma \in \Gamma} \sum_{i_1=1}^{M_1} \dots \sum_{i_T=1}^{M_T} c_{i_1, i_2, \dots, i_T} \rho_{i_1, i_2, \dots, i_T}$$
(3)

with constraints

$$\sum_{i_{2}=1}^{M_{1}} \dots \sum_{i_{T}=1}^{M_{T}} \rho_{i_{1}, i_{2}, \dots, i_{T}} = 1; \qquad i_{1} = 1, \dots, M_{1}$$

$$\sum_{i_{1}=1}^{M_{1}} \dots \sum_{i_{T}=1}^{M_{T}} \rho_{i_{1}, i_{2}, \dots, i_{T}} = 1; \qquad i_{2} = 1, \dots, M_{2}$$

$$\vdots \qquad \vdots$$

$$\sum_{i_{1}=1}^{M_{1}} \dots \sum_{i_{T-1}=1}^{M_{T}-1} \rho_{i_{1}, i_{2}, \dots, i_{T}} = 1; \qquad i_{T} = 1, \dots, M_{T}$$

$$(4)$$

The solution ρ to this MDAP is the multidimensional extension of the binary assignment matrix. Simply, one may consider ρ as being a multidimensional 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.

The other form of the MDAP we are interested in is multi-sensor association with $S \ge 3$ 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 [119] [13]. When each sensor sends its local tracks to a central node for track association and fusion, a MDAP must be solved. In this case, the dimensionality of the MDAP is equal to S, and hence, is NP-hard. The main difference between this problem and the previous data association problem is that it deals solely with tracks,

:8 P. Emami et al.

as opposed to new sensor measurements from all sensors. Multi-scan track-to-track association with two sensors is also a MDAP, as well as multi-scan multi-sensor data association (Table 1), but we omit these cases for brevity in our formulation and for the fact that they can be defined quite similarly from what is presented next.

Following [32], in this scenario there are $S \ge 3$ sensors, each maintaining a set of local tracks and using a sliding window of size $T \ge 1$. We define $X_k^s = \{x_k^{i,s}\}$, s = 1, ..., S, to represent the set of track state estimates produced by sensor s at time k. We have $i = 1, ..., N_s$, 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 $X^{T,s} = \{X_1^s, ..., X_T^s\}$, which represents the collection of track state estimates within the sliding window. We seek an optimal partitioning $\gamma^* \in \Gamma$ of $X^T = \{X^{T,1},...,X^{T,S}\}$ 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 $\gamma^l = \{\{x_1^{j,1}, x_1^{j,2}, ..., x_1^{j,N_s}\}, ..., \{x_T^{j,1}, x_T^{j,2}, ..., x_T^{j,N_s}\}\}$. In words, this states that the j^{th} track of sensor 1 from scan 1, jth 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 3 sensors each maintaining 3 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 $c_{i_1,i_2,...,i_{N_s}}$. Increasing T to include more than one scan corresponds to adding extra dimensions to the problem. We can use binary variables as before, $\rho_{i_1,i_2,...,i_{N_s}} \in \{0,1\}$, to indicate whether a particular partial hypothesis is present in γ . The MDAP can then be written as

$$\min_{\gamma \in \Gamma} \sum_{i_1=1}^{N_1} \dots \sum_{i_{N_s}=1}^{N_s} c_{i_1, i_2, \dots, i_{N_s}} \rho_{i_1, i_2, \dots, i_{N_s}}$$
 (5)

with constraints

$$\sum_{i_{2}=1}^{N_{1}} \dots \sum_{i_{N_{s}}=1}^{N_{s}} \rho_{i_{1}, i_{2}, \dots, i_{N_{s}}} = 1; \qquad i_{1} = 1, \dots, N_{1}$$

$$\sum_{i_{1}=1}^{N_{1}} \dots \sum_{i_{N_{s}}=1}^{N_{s}} \rho_{i_{1}, i_{2}, \dots, i_{N_{s}}} = 1; \qquad i_{2} = 1, \dots, N_{2}$$

$$\vdots \qquad \vdots$$

$$\sum_{i_{s}=1}^{N_{1}} \dots \sum_{i_{N_{s}}=1}^{N_{s-1}} \rho_{i_{1}, i_{2}, \dots, i_{N_{s}}} = 1; \qquad i_{N_{s}} = 1, \dots, N_{s}$$

$$(6)$$

As with the multi-scan data association problem, the solution takes the form of a multidimensional binary array. As before, the number of potential assignment hypotheses in a MDAP can be reduced with gating. Even with gating, solving a 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 [49]. Notably, the MDAP is closely related to other NP-Hard combinatorial optimization problems, such as Maximum-Weight Independent Set and Set Packing [27]. In the next subsection, 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.

2.3 Assignment Costs

The assignment cost function has a massive impact on tracking performance. In this subsection, we will introduce various perspectives towards defining assignment costs, specifically highlighting probabilistic approaches.

2.3.1 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; see [80] for 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., $P(x_k|z_k) \propto P(z_k|x_{k-1})P(x_{k-1}|z_{k-1})$. 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. Now, 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 canonical cost function for data association is to minimize the following negative log-likelihood ratio:

$$c_{i_1,i_2,...,i_T} = -\log \frac{P(\gamma^j | z_1^i, z_2^i, ..., z_T^i)}{P(\gamma^0 | z_1^i, z_2^i, ..., z_T^i)}, \quad (\gamma^j, \gamma^0) \in \gamma.$$
 (7)

The partial hypothesis γ^j represents the jth 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 jth track generated the assigned measurements is

$$P(\gamma^{j}|z_{1}^{i}, z_{2}^{i}, ..., z_{T}^{i}) \propto P(z_{1}^{i}, z_{2}^{i}, ..., z_{T}^{i}|\gamma^{j}).$$
 (8)

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

$$P(z_1^i, z_2^i, ..., z_T^i | \gamma^j) = \prod_{k=1}^T P(z_k^i | x_k) P(x_k | j).$$
(9)

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 9 and plugging it into the negative log-likelihood ratio in Equation 7.

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

:10 P. Emami et al.

likelihood that Δ_{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 $\Delta_{1S} = \sum_{i=2}^{S} \Delta_{1i}$ [127]. For the Gaussian case, the cost can also be written in closed form as a Mahalanobis distance between the track state estimates [62] [32]:

$$c_{i_1, i_2, \dots, i_S} = \sum_{j=2}^{S} \Delta_{1j}^{\mathsf{T}} \Sigma_{1j}^{-1} \Delta_{1j}$$
 (10)

In the Bayesian setting, minimizing Equations 7 and 10 is analogous to finding the MAP assignment hypothesis; this will be covered in more detail in Section 3.

2.3.2 Feature-augmented Costs. It is often the case in multi-target 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 subsection. 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, e.g., 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; the goal of Section 4 is to highlight the approaches that have proven to be the most useful. 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-target tracking computer vision community, partially due to the ease at which features can be extracted from images [75].

There are multiple ways to incorporate learned pairwise costs into a MDAP solver. One common approach is as follows. The probability of association for a pair of measurements or tracks Λ_i and Λ_j can be written as a joint pdf [96]; assuming independence of the kinematic (K) and non-kinematic (NK) components of this probabilistic cost function, the resulting negative log-likelihood pairwise cost is:

$$c_{ij} = -\log P(\Lambda_i, \Lambda_j)$$

$$= -\log \left(P_K(\Lambda_i, \Lambda_j) P_{NK}(\Lambda_i, \Lambda_j) \right)$$

$$= -\log P_K(\Lambda_i, \Lambda_j) - \log P_{NK}(\Lambda_i, \Lambda_j)$$
(11)

Usually, $P_{NK}(\Lambda_i, \Lambda_j)$ 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, $P_K(\Lambda_i, \Lambda_j)$, could be adapted from Equation 7.

3 OPTIMIZATION

In this section, we will review recent work on a variety of optimization algorithms for solving MDAPs in real-time multi-target tracking systems. Our focus will be on approaches with a machine learning flavor, e.g., approximate inference techniques and deep neural networks, as well as the probabilistic modeling aspects of the problem. We will start by briefly covering non-probabilistic

methods that are useful for contrasting with what is currently popular. The techniques discussed in this section are quite general, and in most cases can be used for both the data association and track-to-track association problems with proper modification. It is important to notice that certain modeling assumptions, such as how the assignment cost function is defined, can cause a tracker to make errors regardless of how strong the optimization approach is.

3.1 Greedy Randomized Search

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 in [92] for multi-sensor multi-target tracking. The idea behind GRASP is to randomly select each partial assignment from lists of greedily chosen candidates to form a solution γ . Then, a local random search is conducted to attempt to improve this solution. This procedure is repeated until the alloted time runs out or a maximum number of iterations is reached, at which point the best solution that was discovered is returned. GRASP algorithms also use gating techniques to help reduce the search space, and conduct the local searches by permuting a small number of entries within some of the assignments. A parallel implementation of GRASP is described in [94]. In [103] it is suggested that GRASP produces suboptimal solutions of a quality that is not acceptable for real-time performance; however, experiments on modern parallel computer architectures are needed to verify this claim.

Other greedy search algorithms have been proposed in [99] and [115], based on the semi-greedy track selection (SGTS) algorithm introduced in [20]. 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. This process is repeated iteratively in a manner so that candidate hypotheses are generated that best represent the solution space. In [99], an extension for the K-best case is also provided, which enables the use of SGTS-esque algorithms for multiple hypothesis tracking (MHT) [12]. The construction of SGTS and its extensions are such that they can provide a solution that is within a guaranteed factor of the optimal solution [99].

The main strength of search algorithms appear to be their simplicity and the extent to which they are embarrassingly parallel. Despite being quite general, the advent of more sophisticated techniques that can leverage problem-specific information and the necessary hardware necessary to run them in real-time has most likely contributed to the lack of continued research on GRASP algorithms in the academic tracking community. For a survey of research on GRASP for optimization, see [107].

3.2 Lagrangian Relaxation

The multidimensional 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 [66] [91] [11] 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 [14] algorithms for association in multi-target tracking, proposed in [31] and [32], involve iteratively producing increasingly better solutions to the MDAP by successively solving relaxed LAPs and reinforcing the constraints. A set of Lagrange multipliers for the N-dimensional case, $\mathbf{u} = [u_3, u_4, ..., u_N]$, are introduced to incorporate the relaxed set of constraints into the cost function. Since there are potentially multiple constraints that are not being enforced at each iteration, the obtained solution is an optimistic lower bound on the actual optimal solution, referred to as the dual solution. When the constraints

:12 P. Emami et al.

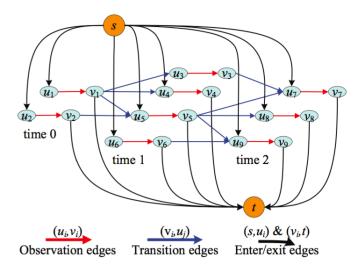


Fig. 4. A network flow graph for multi-scan data association (three scans depicted). The black arcs represent enter/exit edges for a potential track. The red arcs are measurement/observation edges, and the blue arcs are transition edges between measurements. Reproduced from [149] with permission.

are reapplied, a valid solution is obtained that is an upper bound on the optimal solution, referred to as the primal solution. The idea is then to update **u** using subgradient methods (see Appendix A of [32]) and to repeat the procedure until the *duality gap*, the difference between the primal and dual solutions, is below a threshold. To formulate this algorithm for real-time applications, it can also be set to terminate after a maximum number of iterations.

A parallel implementation of this method for the K-best case was developed in [104] and [103], which enables efficient implementations of MHT algorithms. A variation on this approach using dual decomposition is proposed in [70] where the original MDAP is separated into subproblems that contain copies of the original variables; a constraint is introduced via Lagrangian relaxation that requires copies of the same variable to share the same value. In experiments evaluating the performance of the dual decomposition method on a generic tracking problem with six closely spaced targets, it performed comparably with the Lagrangian relaxation algorithm from [32].

Lagrangian relaxation has also been used to convert Equation 3 into a global network flow problem in [19]. 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 subgradient methods. In the next subsection, we go into more detail on network optimization, one of the leading approaches to solving multi-target tracking association problems.

3.3 Network Optimization

A popular approach to solving MDAPs for data association (Equation 3) in the computer vision tracking community is to transform the problem into finding a minimum-cost network flow [58] [149] [100] [10] [131] [128] [139] [19] [113] [24]. In the corresponding network, detections at each discrete time step 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, e.g., 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 hypothesis by limiting transition edges between nodes with a spatio-temporal 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 [149], 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

$$\gamma^* = \arg\max_{\gamma \in \Gamma} \frac{P(Z^T | \gamma)}{\mathcal{T}_m \in \gamma} P(\mathcal{T}_m)
\text{s.t. } \mathcal{T}_m \cap \mathcal{T}_n = \emptyset, \forall m \neq n$$
(12)

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 9; in [149], it is factored as $P(Z^T|\gamma) = \prod_z P(\{z \in Z^T\}|\gamma)$, 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 $P(z_{k+1}^i|z_k^i) \neq 0$, 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. The flows f are indicator functions defined by

$$f(s, u_r) = \begin{cases} 1 & \text{if } \exists \mathcal{T}_m \in \mathcal{T}, \mathcal{T}_m \text{ starts from } u_r \\ 0 & \text{otherwise} \end{cases}$$

$$f(v_r, t) = \begin{cases} 1 & \text{if } \exists \mathcal{T}_m \in \mathcal{T}, \mathcal{T}_m \text{ ends at } v_r \\ 0 & \text{otherwise} \end{cases}$$

$$f(u_r, v_r) = \begin{cases} 1 & \text{if } \exists \mathcal{T}_m \in \mathcal{T}, z_k^i \in \mathcal{T}_m \\ 0 & \text{otherwise} \end{cases}$$

$$f(v_r, u_s) = \begin{cases} 1 & \text{if } \exists \mathcal{T}_m \in \mathcal{T}, z_{k+1}^i \text{ comes after } z_k^i \text{ in } \mathcal{T}_m \\ 0 & \text{otherwise} \end{cases}$$

$$(13)$$

and the costs are defined as

$$c(s, u_r) = -\log P_{\text{start}}(z_k^i) \quad c(v_r, t) = -\log P_{\text{end}}(z_k^i)$$

$$c(u_r, v_r) = \log \frac{\beta_r}{1 - \beta_r} \quad c(v_r, u_s) = -\log P_{\text{link}}(z_{k+1}^i | z_k^i)$$
(14)

:14 P. Emami et al.

and can be derived by taking the logarithm of Equation 12; see Section 3.2 in [149] 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. One is described in [58], in which a subgraph is created for each track in the surveillance region and occlusion is modeled by adding special nodes to the graphs. 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. The method from [100] further optimizes the approach introduced in [149] to reduce the run-time complexity. A comparable approach to this one, from [10], formulates the problem as finding the K-shortest paths through the flow graph. In [27], the argument that the popular network flow model exhibits an over-reliance on appearance modeling and pairwise costs is made. They offer a variation on the network flow approach that uses a more general cost function. In Section 4, we will go into the details of recent works that propose a variety of machine learning techniques to obtain the link costs (Equation 14) in network flow graphs. Network optimization techniques offer a good trade-off between complexity, ease of implementation, and performance.

3.4 Conditional Random Fields

Probabilistic graphical models provide us with a powerful set of tools for modeling spatio-temporal relationships amongst sensor measurements in data association and amongst tracks in track-to-track association. Indeed, conditional random fields (CRFs), a class of Markov random fields [69], have been used extensively for solving MDAPs in visual tracking [89] [144] [143] [71] [24] [97]. 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 graphical 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-target tracking, a realization of labels 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 x0 of the graph [53]. For example, in chain-structured graphs, each pair of nodes and corresponding edge is a clique.

CRFs, like the network flow models discussed in the previous subsection, are essentially a tool for modeling probabilistic relationships between a collection of random variables, and hence still require a separate optimization process for handling training and inference (such as 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 such as [15] for details on how to do approximate inference 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, e.g., 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 reformulated; 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., $\psi_c = \exp(w_c^{\mathsf{T}} \phi(x, y_c))$. Note that the implied normalization term in Equation 15 can be omitted when solving for the maximum-likelihood labeling y for a particular set of observations x.

$$P(\mathbf{y}|\mathbf{x}, w) \propto \prod_{c} \psi_{c}(y_{c}|\mathbf{x}, w)$$

$$\propto \prod_{s \in V} \psi_{s}(y_{s}|\mathbf{x}, w) \prod_{s, t \in E} \psi_{s, t}(y_{s}, y_{t}|\mathbf{x}, w)$$
(15)

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 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-target tracking as assuming that the assignment of the observations to tracks within a particular spatio-temporal 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 15, i.e., the maximum-likelihood set of labels 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 indicates 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 [143], the weights w are learned with the RankBoost algorithm. Other techniques for learning the parameters of a CRF that maximize the log-likelihood of the training data include iterative scaling algorithms [69] 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, in [144] extensions to the generic CRF formulation are presented that enable it to run in real-time.

A CRF formulation, Near Online Multi-Target Tracking (NOMT), is proposed in [24] that 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 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 subgraphs can be solved in parallel.

Other variations on the approaches above have been seen as well. In [89], the energy term of a CRF is augmented with a continuous component to jointly solve the discrete data association and continuous trajectory estimation problems. A factor graph is embedded in the CRF in [54] to add more structure and help model pairwise associations explicitly. In the next subsection, we will investigate how factor graphs, the belief propagation inference algorithm, and its variants can be used to solve the MDAP. To summarize, applying CRFs to a specific multi-target 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.

:16 P. Emami et al.

3.5 Belief Propagation

In this section, we highlight recent work that formulate the association problems as MAP inference and use belief propagation (BP) or one of its variants to obtain a solution. Chen et al. [21] [22] 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 subsections. Indeed, BP can be used on graphs that model joint distributions $P(\mathbf{x}) = P(x_1, x_2, ..., x_N)$ 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

$$P(\mathbf{x}) \propto \prod_{c} \psi_{c}(x_{c})$$

$$\propto \prod_{s \in V} \psi_{s}(x_{s}) \prod_{s,t \in E} \psi_{s,t}(x_{s}, x_{t})$$
(16)

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 $\{x_1, x_2, ..., x_N\}$. 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 16 as

$$P(\mathbf{x}) \propto \prod_{s \in V} \psi_s(x_s) \prod_{f \in \mathcal{F}} \psi_f(x_{\eta_f})$$
 (17)

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 $x^* = \{x^*_{s} | s \in V\}$ which corresponds to the best assignment hypothesis γ^* . In this algorithm, messages are computed recursively in general pairwise Markov random fields by

$$\mu_{s \to t}(x_s) = \max_{x_t} \left\{ \psi(x_t) \psi_{s,t}(x_s, x_t) \prod_{\xi \in N(t) \setminus s} \mu_{\xi \to t}(x_t) \right\}$$
(18)

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

$$x_s^* = \operatorname*{arg\,max}_{x_s \in X} \left\{ \psi_s(x_s) \prod_{\xi \in \operatorname{nbr}(s)} \mu_{\xi \to s}(x_s) \right\} \tag{19}$$

for neighborhood set nbr(s). As indicated in [21], these updates are not guaranteed to converge for graphs with cycles, and even if they do, they may not compute the exact MAP configuration. A proof of convergence for a specific loopy belief propagation (LBP) formulation for data association presented in [137]. 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 [138] [136] [84]. An improvement over the max-product algorithm for LBP is tree-reweighted max-product [126]. This algorithm is used for data association in [21] to output a provably optimal MAP

configuration or acknowledge failure. 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 [21].

To illustrate the use of BP for solving MDAPs, we will present the graphical model formulation from [151] for multi-sensor multi-target track-to-track association. The structure of the graphical model is decided on-the-fly by producing sets of independent association clusters consisting of multisensor 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; in [151], the gates are computed from purely kinematic information. The nodes of the graph are the track state estimates for T=1 and $S\geq 3$ sensors (Section 2), $\left\{x^{i,j}|x^{i,j}\in X^1=\{X^{1,1},X^{1,2},...,X^{1,S}\}\right\}$, where each $x^{i,j}$ is the i^{th} track state estimate from sensor j, $i=1,...,N_j$ and j=1,...,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 pair-wise costs, given by Equation 10. 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

$$\psi_{\chi^{l,m} \to \chi^{n,o}}(Y_n^{l,m} = p, Y_l^{n,o} = q) = \begin{cases} 0 & p = n, q \neq l \\ 0 & p \neq n, q = l \\ 1 & \text{otherwise} \end{cases}$$
 (20)

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

$$\mu_{\chi^{l,m} \to \chi^{n,o}}(Y_l^{n,o} = q) = \begin{cases} \exp(w^{u=(l,m);v=(n,o)}) & \text{if } q = l\\ 1 & \text{otherwise} \end{cases}$$
 (21)

Then, repeated application of Equation 18 until the $Y^{i,j}$ s converge will produce the MAP solution to the MDAP.

Examples of factor graphs for the data association MDAP can be found in [138], and examples of pairwise Markov random fields formulations of the data association MDAP are in [21] and [22]. An extension to [138] for an unknown number of targets and multiple sensors is presented in [83] and applied to a multistatic sonar network in [84]. As shown in [138], a *hybrid* factor graph that encodes the constraints (that each measurement be associated to at most one target and each target give rise to at most one measurement) with two different sets of constraint variables exhibited the strongest performance. A useful overview of graph techniques for the data association problem, including BP, is [25]. See [24] for an example of how BP can be used as a general inference technique for MAP inference on a network flow graph.

3.6 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-target tracking when the goal is to estimate a posterior distribution over assignment hypotheses, from which a MAP hypothesis can

:18 P. Emami et al.

be extracted. The Metropolis-Hastings algorithm has been used extensively for data association in single and multi-sensor scenarios [9] [98] [93] [39]. 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 [108] [125]. We omit detailed descriptions of the Metropolis-Hastings and Gibbs sampling algorithms, and instead refer the reader to the explanations in [125] and [93].

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 $P(\gamma|Z^T)$ or $P(\gamma|X^T)$. A MAP assignment hypothesis γ^* for the data association problem is [93]:

$$P(\gamma | Z^T) \propto P(Z^T | \gamma) \prod_{t=1}^T p_z^{z_t} (1 - p_z)^{c_t} p_d^{d_t} (1 - p_d)^{g_t} \lambda_b^{a_t} \lambda_f^{f_t}$$
(22)

$$\gamma^* = \arg\max_{\gamma} P(\gamma | Z^T) \tag{23}$$

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 is provided in [93].

A Metropolis-Hastings algorithm for Equation 22 is described in [93] as follows. 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

$$A(\gamma, \gamma') = \min\left(1, \frac{\pi(\gamma')q(\gamma', \gamma)}{\pi(\gamma)q(\gamma, \gamma')}\right)$$
(24)

Assuming a uniform proposal distribution q, the proposal distribution terms in the numerator and denominator cancel. The stationary distribution $\pi(\gamma)$ is $P(\gamma|Z^T)$ from Equation 22. Implementation details and descriptions of each type of move can be found in Section V-A of [93]. Extensions to this algorithm have been proposed in [9] to add a sliding-window flavor and to reduce the number of types of moves to three. Since the application in [9] is visual tracking, appearance information is fused with kinematic information to help improve performance. [39] uses sparse representations of detections and kinematic information to define an energy objective that MCMCDA approximately solves. They deviate from prior work by allowing moves to be done not only forward in time, but also backwards as well 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 run-time.

3.7 Deep Learning

Neural networks have a rich history of being used to solve combinatorial optimization problems. One of the most influential papers in this line of research, by Hopfield and Tank [56], describes how to use Hopfield nets to approximately solve instances of the Traveling Salesman Problem (TSP). Despite the controversy associated with their results [117], this work inspired many others

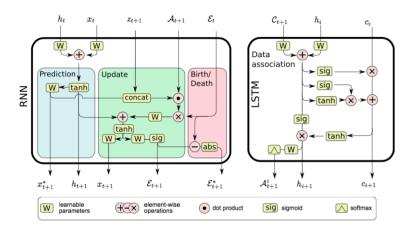


Fig. 5. An LSTM cell designed for multi-scan single-sensor data association (right). The input at each time step is the matrix of pairwise distances C_{t+1} , along with the previous hidden state h_t and cell state c_t . The output A_{t+1}^i of the data association cell is a vector of assignment probabilities 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 nonlinearities and memory are believed to provide the means for learning efficient solutions to the data association problem. Reproduced from [87] with permission.

to pursue these ideas. This has lead to the present day, where research on the use of deep neural networks to solve problems like the TSP has started to pick up speed.

Deep Reinforcement Learning. The assignment problems in multi-target tracking are, at their core, combinatorial problems. Naturally, the question of whether deep neural networks are useful for finding near-optimal solutions to the LAP or MDAP is of significant interest. A preliminary answer to this question, in recent work by [88], suggests the affirmative; they used a recurrent neural network to solve a small MDAP in a simulated multi-target tracking scenario. Impressively, they were also able to get good performance on a quadratic assignment problem that involved finding point correspondences between pairs of images. In [88] it was also suggested that using a problem-specific objective function for training neural networks in a supervised manner, as opposed to using, e.g., a regression loss [124], is preferable. One of the key challenges that supervised learning approaches face in this domain is obtaining labeled ground-truth samples, since generating optimal solutions to NP-hard combinatorial optimization problems can be timeconsuming or even impossible. To address this, [8] and [29] use reinforcement learning to avoid the requirement of labeled data. The main difficulties here are deciding how to represent the data for efficient learning and enforcing the original constraints of the problem during training, e.g., Equations 4 and 6. Naively searching in the space of assignment hypotheses forces a reinforcement learning agent to select an action from an action space of size n!. Furthermore, if the agent's policy is parameterized by a deep neural network, as is the case in deep reinforcement learning [3], the output of the policy network (if searching directly in the space of valid solutions) is a permutation matrix; more formally, an extreme point of the Birkhoff polytope [78]. This has been known to be quite difficult to do with neural networks [44]. An alternative to this is the approach in [29], where a Deep O-Network augmented with a graph-embedding layer is used to greedily construct valid solutions to graph combinatorial optimization problems. Principled approaches for doing inference over permutations have been proposed in [47], [78], and [82] based on annealing a

:20 P. Emami et al.

temperature-controlled parameter to produce a discrete permutation matrix from a continuous doubly-stochastic matrix. However, this technique has yet to be extended to the reinforcement learning setting. We note that reinforcement learning has already been applied successfully to multi-target tracking by [141], where a policy is learned to control track initialization, maintenance, and removal.

Deep Learning on Graphs and Sets. Featurization of the assignment hypothesis graph (e.g., Figure 2) seems useful for a deep learning-based approach. Graph-embedding techniques can potentially enable deep neural networks to handle missed detections and false alarms by providing the means to model missing edges in the assignment hypothesis graph. However, learning useful inductive representations of graph-structured data is still an open problem in machine learning; see [46] and [51] for recent progress on this. Notably, the deep reinforcement learning algorithm from [29] makes use of a powerful graph embedding technique called struct2vec proposed in their earlier work [28]. Also, see [17] for a general discussion on applying deep learning to graphs, including the recently proposed Graph Convolutional Networks (GCNs) [65]. In particular, it is observed that the transductive nature of the graph embeddings learned by GCNs prohibits them from generalizing to graphs with different structure at test time, rendering this approach unusable for multi-target tracking. The review papers by Goyal et. al. [48] and Hamilton et. al. [52] are also useful for learning more about recent efforts at embedding graphs into a feature space. When applying deep neural networks to combinatorial optimization problems where the solution space consists of permutations or subsets of the input, Vinyals et. al. [124] [123] proposed Pointer Networks, which leverage attention mechanisms and the powerful seq2seq architecture to greedily construct valid solutions. In [109], a deep learning architecture inspired by the theory of Random Finite Sets [80] is proposed to predict outputs that are structured as sets by simultaneously predicting the cardinality of the set. They include promising results from pedestrian detection benchmarks, showing results that are slightly worse than state-of-the-art.

3.7.3 End-to-End Multi-target Tracking. As is common with deep learning research, some have already gone further to ask whether multi-target tracking can be solved in an entirely end-to-end fashion [95]. In other words, given noisy measurements of the environment, the objective is for a deep learning system to directly output the filtered tracks, combining the association problem with state estimation. An investigation by [95] revealed that a recurrent-convolutional neural network (RCNN) 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 time step, as would be natural for standard Bayesian recursive filtering, 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 [33]. In short, they showed that their system is able to predict an unoccluded version of the occupancy grid derived from the sensor's field-of-view. Recently, [40] proposed Recurrent Autoregressive Networks (RAN), an approach to online multi-target tracking that seeks to incorporate internal and external memory components into a deep learning framework to help handle occlusion and appearance changes. Crucially, they are able to show that RAN indeed makes use of its external memory to maintain tracks while the targets are occluded. The appearance and motion features in the external memory and the hidden state of the recurrent network used for the internal memory are combined to produce association scores for data association with the Hungarian algorithm. See [111] for a closely related prior work that also explores the use of recurrent networks.

Instead of pursuing the monolithic end-to-end approach, [87] represents the state estimation and data association problems separately in their deep learning architecture, arguing that doing so

provides the means to separately train and debug these components. They design 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 [73].

Research on applying deep learning to the LAP and MDAP in multi-target tracking is still in its infancy; based on the flurry of recent work on this problem, it is likely that we will see significant progress on this in the near future. However, using data-driven solutions brings up the question about whether such a system could generalize to any environment it may be deployed in. A interesting research direction for addressing this is zero-shot learning [140]. Next, we will tackle the other major machine learning task in multi-target tracking—learning the assignment costs.

4 LEARNING ASSIGNMENT COSTS

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 large amounts of annotated datasets that are freely 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. Following this, we will conclude the section by highlighting recent research from the multi-sensor data fusion community on representation learning.

4.1 Learning Assignment Costs in Multi-Target Tracking, Pre-Deep Learning

Data-driven approaches to multi-target tracking are becoming popular due to learning algorithms that can take advantage of the increased availability of high-quality datasets. In essence, the goal of data-driven multi-target tracking is to use labeled datasets to train a model to output association costs at test time, where the cost might look similar to Equation 11. These learned functions are then used in the optimization frameworks introduced in Section 3. It is common to use discriminative models for learning appearance affinity; these models attempt to learn a conditional distribution P(Y|X). Y could be a categorical random variable for classification, or it could be real-valued for regression. Basically, discriminative models in visual tracking are used to predict an association likelihood based on appearance information. A simple example would be a neural network or Support Vector Machine (SVM) trained on a dataset of pairs of detections to output a score between 0 and 1. The score corresponds to the model's confidence about whether a pair of detections were generated by the same object. Another learning paradigm that has been used in conjunction with discriminative models for this task is metric learning. In this setting, a distance metric between measurements or tracks, typically in the form of a parameterized Mahalanobis distance, is learned from training data. We discuss a variety of machine learning techniques in this subsection to provide a brief historical context to frame our presentation of deep learning-based methods in the next subsection. We provide Table 2, which summarizes the various visual features used for learning association costs with the methods mentioned in this subsection.

4.1.1 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 introduced in [76], one of the first applications of data-driven learning to multi-target tracking, is used to learn the link costs for a network flow graph (Equation 14). The data association problem is decomposed into a hierarchy of association problems where the tracklet lengths successively increases [57]; furthermore, it is

:22 P. Emami et al.

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 [43]. Their HybridBoost model is trained offline with videos paired with ground-truth trajectories. In [67], a slightly different approach is taken; a hierarchical decomposition in the same vein as [57] and [76] is used, but each stage of the hierarchy is linked by applying the Hungarian algorithm. 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 2-8 pedestrians at a time, this tracker runs at about 4 fps. Other appearance models based on boosting include [143] and [144], where the RankBoost algorithm is used with CRFs. The online-learned discriminative appearance model from [67] is adopted in [144]. In an extension to [67], ideas from person re-identification are embedded into the system to improve the appearance model [68]. The features used to construct the parameterized learners for the boosting algorithms mentioned here are summarized in Table 2.

In efforts to improve upon boosting for online learning of appearance models, [4] proposed the use of incremental linear discriminant analysis (ILDA). They showed that ILDA outperforms boosting in their experiments in terms of identification accuracy and computational efficiency, partially due to the fact that ILDA simply requires updating a single LDA projection matrix for distinguishing amongst 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

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

for appearance, shape, and motion (kinematics) affinities. This form of the cost is similar to Equation 11 and is fairly common. The appearance affinity is the score computed by ILDA, and the shape and motion affinities are not learned from data; details about those can be found in [4]. In this work, tracks are incrementally stitched together from tracklets by repeated application of the Hungarian algorithm. Another alternative to boosting methods, which is especially useful for learning the parameters of association cost functions embedding within complex graphical models, is the structured SVM [64] [131] [132] [24]. This approach, however, typically limits the cost functions to a linear parameterization.

4.1.2 Metric Learning. A different approach to addressing the problems of variability in object appearance and representation learning is target-specific metric learning. Here, we define metric learning as the problem of learning a distance $d_A(x,y) = \sqrt{(x-y)^\intercal}A(x-y)$ parameterized by a positive semi-definite (PSD) matrix 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 [142]. This is then cast as a constrained optimization problem to ensure that the solution A is valid, i.e., $A \ge 0$. An early attempt at applying metric learning in multi-target tracking was [133], where the problem of learning a discriminative model for appearance matching given image patches is combined with motion estimation and jointly optimized with gradient descent. Their formulation requires running the optimization at each time step 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-target tracking is learning link costs in a network flow graph [129] [128]. Here, a regularized version of the aforementioned constrained optimization problem is applied to learn a distance between feature vectors for an appearance

Related Work	Method	Summary of Features Used
[76]	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
[67], [68], [144]	AdaBoost	color histograms, covariance matrices, HOG
[143]	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
[131] [132]	Structured SVM	Off-the-shelf detector confidence (e.g., from DPM [41]), consecutive bounding box IOU, geometric relationships between all pairs of objects
[129] [128]	Metric learning	RGB, YCbCr, and HSV color histograms, HOG, two texture features extracted with Schmid and Gabor filters

Table 2. Features used for data-driven learning of assignment costs from a representative set of works.

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

We will revisit metric learning when we discuss learning representations of multi-sensor data in Section 4.3. The topic of the next subsection transitions over to the use of deep learning for learning assignment costs.

4.2 Learning Assignment Costs in Multi-Target Tracking, Post-Deep Learning

Tracking-by-detection has solidified its position as the primary tracking paradigm for visual tracking, especially now that convolutional neural networks (CNNs) are widely used for learning assignment costs. CNNs are a special class of deep neural network that can learn hierarchical features which are translation invariant and invariant to slight deformations. For object detection and recognition, augmenting the training set by varying orientation, scale, and color can help to further increase robustness. CNNs learn incredibly rich representations directly from raw images. Another reason why deep learning is an attractive option for multi-target tracking is because it is straightforward to take a CNN that has been pre-trained on massive datasets and re-purpose it for new tasks by only re-training a few of the layers. In this subsection, we will cover recent research that leverages CNN-based neural network architectures to learn deep discriminative assignment costs.

One of the first uses of deep learning in multi-target tracking is running image patches of detected objects obtained with, e.g., the DPM [41], through a CNN to extract features. The CNNs are usually pre-trained 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 [63]. Here, a 4096-dimensional feature vector is first extracted from the CNN for each detection box, followed by an application of PCA to reduce the dimensionality to

:24 P. Emami et al.

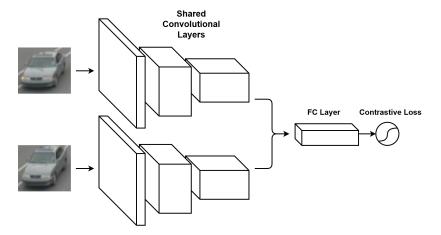


Fig. 6. The basic architecture of a siamese network. The weights of the convolutional layers are shared between the two arms of the network. A contrastive loss can be used to train the network to predict the similarity of the two input images.

256. The classifier is used to compute a log-likelihood cost for a track hypothesis given a set of sensor detections. This paper 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, this approach outperformed the second-best tracker on the 2D MOT 2015 Challenge [73] by 7% in multiple object tracking accuracy (MOTA).

A variation on the standard CNN architecture that has seen extensive use in multi-target tracking is the siamese network. As nicely summarized in [72], a siamese network processes two inputs simultaneously using multiple layers with shared weights (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, [72] proposed a technique where 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; the final layers of these two networks are extracted and combined with a gradient-boosting classifier to produce a match prediction score. Tracks are obtained by solving a network flow problem (Section 3.3) using Linear Programming. Siamese networks are also used in [130] to learn an embedding of two detections into a metric space where their affinity can be easily discriminated. In this work, all parameters between the two arms of the CNN are shared; the features produced by the last layers are used as input to a metric learning loss. A multi-task loss function for incorporating temporal constraints is combined with the regularized metric learning loss (Section 4.1.2) to jointly optimize the weights of the deep model with stochastic gradient descent. They use an online learning algorithm to address the issue of changing object appearance throughout a trajectory, but the deep networks are pre-trained with auxiliary data. The learned affinity model is used with the softassign algorithm [47] to solve a LAP to find an optimal pairing of tracklets. For the task of underwater multi-target tracking, siamese networks were shown to improve performance as well [105]. Instead of only considering pairs of images with siamese networks, the Quad-CNN [118] 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 to use a minimax label propagation algorithm that makes use the trained Quad-CNN for data association in a sliding window.

The confidence-based robust online tracking approach from [4] has been extended by adding a discriminative deep appearance model in [5]. Similarly to the siamese network approach, they pass two image patches through a CNN to automatically featurize them. Then the features from the last CNN layer are used to compute a distance with the squared L2 norm; this distance is used to define a regularized energy function such that the lowest possible energy is assigned to the optimal assignment hypothesis. The deep network is once again pre-trained on a large dataset, and online transfer learning is leveraged to update a small number of the higher layers in the network to adapt to changing object appearances. In particular, when the average affinity scores computed by the network falls 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 run-time 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 time steps, as opposed to every time step. An efficient online algorithm for updating appearance models is described in [145]; here, the problem is cast as learning a bilinear similarity function between two feature vectors with constrained convex optimization. The feature vectors are aggregated from the last convolutional layer of a CNN pre-trained on ImageNet and fine-tuned on the PASCAL VOC dataset.

Rather than formulate the data association problem for multi-person tracking as a MDAP, [121] defines it as a minimum-cost graph multi-cut problem. The key differences here with previously discussed optimization approaches are that multiple detections at a time step can be attributed to the same person; also, it is easy to allow edges to connect across multiple time steps in this graph to handle occlusion. The edge costs are learned with logistic regression, with features obtained from the DeepMatching [134] algorithm. DeepMatching uses a CNN that has been trained to produce dense correspondences between image patches, and was notably used in the DeepFlow [134] algorithm for learning to do large displacement optical flow. It is also used in the multi-person tracking system in [55] to compute temporal affinities between input features. Related to this is recent work on examining the interplay between semantic segmentation and multi-target tracking [86] [122] [18]. Indeed, [18] uses a CNN to segment images, and then computes the optical flow between segmented object pairs in consecutive images to define an association cost matrix for the LAP.

The network optimization approach from [149] is revisited once again in [113], where 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

$$\underset{\Theta}{\operatorname{arg\,min}} \mathcal{L}(x^{gt}, x^*)$$
s.t. $x^* = \underset{x}{\operatorname{arg\,min}} c(f, \Theta)^{\mathsf{T}} x$

$$(26)$$

$$Ax \le b, Cx = 0$$

for parameters Θ , input data f, ground truth network flow solutions x^{gt} , $x \in \mathbb{R}^M$ are the M concatenated flow variables, $\mathbf{A} = [\mathbf{I}, -\mathbf{I}]^\intercal \in \mathbb{R}^{2M \times M}$ 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

:26 P. Emami et al.

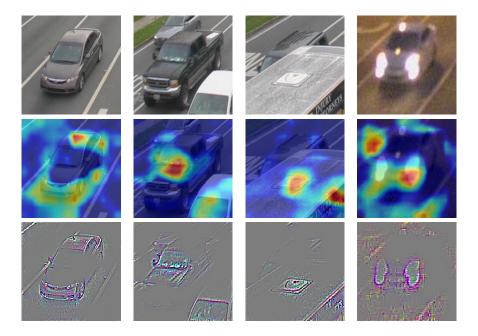


Fig. 7. Visualizations of "important" regions for making predictions about the class label with the VGG16 network, generated with Grad-CAM [114] and pre-trained VGG16 weights [116]. The first two images on the left were correctly labeled as containing vehicles, and it can be seen that the CNN leverages key features such as the car body shape, tires, and windshield to come to this conclusion. The CNN was not able to correctly classify the vehicles in the two images on the right. Heavy occlusion and illumination changes at night can still confuse a CNN that hasn't been trained for these situations. The images were taken with a traffic camera by the authors.

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.

A noticeable trend is the gradual drift away from developing novel optimization algorithms that attempt to solve the MDAP within a sliding window. Rather, recent solutions are relying more on powerful discriminative techniques, such as using features from pre-trained CNNs, and combining this with efficient LAP solvers. Advances in object detection such as Faster R-CNN [106] have almost single-handedly improved the performance of multi-target trackers. To offer some insight into the widely popular approach of using pre-trained CNN for generating detections and learning assignment costs, we present visualizations of CNN activations using the Gradient-weighted Class Activation Mapping technique [114] in Figure 7.

4.3 Multi-Sensor Representation Learning

To wrap-up our discussion of learning assignment costs for multi-target tracking, we will introduce the basic ideas behind representation learning for multi-sensor data and discuss recent progress in this area. Our presentation will focus on applications to multi-target tracking, as well as related tasks such as multi-sensor classification. There are many theoretical and engineering challenges to multi-sensor multi-target tracking, and we hope that this subsection helps generate more discussion on this topic. As stated in Equation 5, for the multi-sensor multi-target track-to-track association problem, we are interested in learning a cost $c_{i_1,i_2,...,i_{N_s}}$ for an assignment of tracks $i_1,i_2,...,i_{N_s}$,

where each track originates from one of N_s sensors. This problem is challenging for a number of reasons; for example, beyond the practical issues of temporally aligning the data, the raw data from each sensor may live in vastly different geometric spaces. Unfortunately, defining a measure of similarity between these spaces is usually non-trivial. Of course, the simple work-around is to independently map each sensor's raw data to the desired low-dimensional representation needed for tracking (e.g., $[x, y, \dot{x}, \dot{y}]^{T} \in \mathbb{R}^{4}$ for position and speed), and then to define a cost function for this representation (e.g., Equation 10). We find that this isn't satisfying; indeed, we would instead like to learn a joint representation of the multi-sensor data that outperform the aforementioned simplistic approach in terms of tracking performance. Consider the example of a video camera, radar, and LiDAR observing one lane at an urban traffic intersection. In this scenario, we can assume that the surveillance regions of each sensor are overlapping. Then, one approach to multi-sensor representation learning would be to map the time-aligned images, point clouds, and radar ranges and azimuths to a single geometric space such that a cost function defined on this space assigns low costs to measurements that are generated by the same vehicle. A connection can be made here with the siamese networks mentioned before where two images are processed by twin pathways in a CNN and mapped to a single vector space, from which a similarity score can be produced. In the remainder of this section, we will discuss different research directions that formalize these ideas for multi-sensor multi-target tracking as well as the related task of multi-sensor classification.

In many multi-sensor multi-target tracking scenarios, there is a network of sensors that are streaming high-dimensional data to a central processing unit for high-level fusion. When the surveillance regions are overlapping, the sensors might be tracking one or more targets from multiple perspectives; however, the raw data streams may live in vastly different geometric spaces when the network consists of heterogeneous sensors. Taking intuitions from manifold learning and dimensionality reduction, [30] introduces the idea of a *joint* manifold that captures a low-dimensional representation of the related data streams. The authors propose a distributed data fusion procedure that uses random projections to efficiently map the data streams to *K*-dimensional component manifolds, which are then linearly combined. They presented an application to tracking where they recorded themselves moving a coffee mug along an "R"-shaped trajectory on a planar surface with 4 cameras. They were able to learn a 2D joint manifold of the data generated by the 4 cameras that visually re-created the "R"-shaped path in the plane. An interesting research direction is thereby extending the theory of joint manifolds for augmenting multi-sensor multi-target tracking algorithms. For an extensive discussion on dimensionality reduction for multi-sensor fusion, we direct the reader to the following thesis [112].

A different perspective on fusion in heterogeneous multi-sensor networks is taken by [148] [150]. In particular, Heterogeneous Multi-Metric Learning for classification [148] involves learning S projection matrices for a classification task, where S is the number of sensors and the target metric space is one where training samples are encouraged to have the same labels as their k-nearest neighbors. Likewise, training samples with different labels are pushed away from each other in the learned space to help optimize the classification performance. To learn the projection matrices, the algorithm takes in a training set of multi-sensor data points and alternates between gradient descent steps over a hinge loss and projection steps onto the positive semi-definite cone to maintain the metric properties for the S matrices. They later strengthen their results in [150] by suggesting the use of the kernel trick to learn the S projection matrices in a Reproducing Kernel Hilbert Space. Related to this work is that of [16], which introduces cross-modality similarity-sensitive hashing. Here, boosting is used to learn two maps that take data from two different geometric spaces and project them onto a single space. The motivation behind using boosting is that a Hamming distance metric on this learned space can be defined as a weighted sum of weak binary classifiers.

:28 P. Emami et al.

In conclusion, we can see that there have been multiple algorithms proposed for multi-sensor representation learning, but they have yet to be fully integrated into multi-sensor multi-target tracking. The works we have described do not represent a comprehensive overview of the subject of multi-sensor fusion, but they suggest many interesting ideas for learning assignment costs. An important aspect of multi-sensor fusion that was not discussed is embedding robustness to temporal misalignment amongst sensors directly in the data fusion algorithm. Without specialized hardware, it can be difficult to precisely align data coming from multiple heterogeneous sensors, which in turn can have a drastic impact on the performance of track-to-track association.

5 BENCHMARKS

In this section, we will briefly review the multi-target tracking benchmarks; for a focused examination, we refer readers to the recent surveys [74] [79]. Following this, we discuss benchmarks pertaining specifically to multi-target tracking in ITS applications.

Perhaps the most popular vision-based multi-object tracking benchmarking as of late is the MOT challenge. The MOT15 challenge was first released in 2014 and consists of 22 video sequences of pedestrians. Since then, the MOT16 [85] and MOT17 challenges have been released, with each release also improving upon the annotation protocol and ground truth quality of the former. These datasets are particularly useful when proposing general improvements to multi-target tracking algorithms, since carefully evaluated results from many of the state-of-the-art trackers are available for comparison. 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 [36], TUD Stadtmitte [2], and ETH Pedestrian [38] 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 as of late. The KITTI benchmark [45] is focused on challenges for autonomous driving in urban environments, and contains many tasks beyond multi-target tracking such as odometry, lane estimation, and orientation estimation.

Traffic surveillance is an application of multi-target tracking that is in desperate need of more high-quality single and multi-sensor datasets. Unfortunately, mounting sensors in areas of heavy traffic flow and collecting and cleaning the data is not an easy task, and collaboration with industry and government entities is crucial. On the other hand, there already exists plenty of datasets for pedestrian tracking, which is also important to ITS applications. Tracking vehicles is useful at traffic intersections as this information can be used for applications such as adaptive traffic signal control and collision detection; this is the area where high-quality datasets are most needed. Tracking both vehicles and pedestrians from the vantage point of an autonomous vehicle is still a challenge as well. The UA-DETRAC benchmark [135] is an excellent large-scale traffic surveillance benchmark that was recently proposed. It consists of 10 hours of video that was recorded at 24 different locations in China, and contains over 8,250 vehicles that were manually annotated. The dataset comes with some reference implementations of popular trackers, an evaluation tool, and detections. Another useful dataset for video-based traffic surveillance research is UrbanTracker [59], which comes with sequences from 4 different intersections, as well as an annotation tool and a metrics tool. For multi-sensor traffic surveillance, the Ko-PER intersection dataset [120] offers 6 sequences collected with multiple cameras and laser scanners; however, only 2 sequences currently have ground-truth labels. Due to the difficulty of collecting, synchronizing, and labeling data across multiple sensors, datasets such as this one are hard to find and extremely valuable. The KITTI tracking dataset also contains synchronized camera and laser scans, but it is slightly less useful for traffic surveillance since it is recorded from the perspective of an autonomous vehicle. Another

video surveillance dataset that is of interest is GRAM Road-Traffic Monitoring [50], which contains 3 sequences recorded under different conditions and with different visual platforms. The benefits of benchmarking across multiple datasets are apparent; in real-world scenarios, traffic surveillance systems will need to generalize to all manners of environments.

Recently, realistic urban driving simulators have become available to advance research in autonomous vehicles [35]. These simulators are typically built on top of game engines and have the ability to generate sensor data. A promising future direction may be leveraging these tools for research on single and multi-sensor multi-target tracking systems, especially if the research seeks to explore augmenting the tracker with vehicle-to-infrastructure communication.

6 CONCLUSIONS

In this survey we argued that considering multi-target tracking as an assignment problem helps to conceptualize the large variety of existing solution techniques. We presented details for the most popular machine learning methods that address the MDAP underlying many single and multisensor multi-target tracking problems. The material was presented by distinguishing between optimization methods for finding the MAP assignment and learning algorithms for the assignment costs, and included a discussion on recent progress in applying deep learning to these tasks. Indeed, the latter is one of the most promising research directions that the field is taking. However, due to the current limited theoretical understanding of deep learning, careful consideration is required before it is deployed in real-world scenarios. The study of some of the failure modes of deep learning (e.g., fooling deep neural networks with adversarial inputs and its poor interpretability) as well as a detailed understanding of its generalization capabilities is still a work in progress. Another interesting research direction that was discussed is the development of solutions for end-to-end multi-target tracking; in particular, data-driven multi-target tracking systems that bundle the series of complex sub-problems into a single, monolithic solution. The fact that deep learning has already been successful in other areas such as machine translation and speech recognition is further evidence that this is a research direction that should be pursued. A large number of other open challenges were also highlighted in this survey, such as handling occlusion, changes in target appearance, and balancing the use of multiple scans of measurements with real-time performance. We used the application to ITS to help motivate many of these, as these problems involve tracking both vehicles and humans in a variety of environmental settings.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under grant 1446813 and the Florida DOT under grant BDV31-977-45. 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 is supported by the Paul and Heidi Brown preeminent professorship at ISE, University of Florida.

REFERENCES

- [1] 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.
- [2] 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, 623–630.
- [3] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38.
- [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.

:30 P. Emami et al.

[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 (2017).

- [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, 2674–2679.
- [7] Yaakov Bar-Shalom and Huimin Chen. 2007. Track-to-Track Association Using Attributes. Information Fusion 2, 1 (2007), 49–59.
- [8] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. 2017. Neural combinatorial optimization with reinforcement learning. In Workshop Track of the International Conference on Learning Representations (ICLR'17).
- [9] 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, 3457–3464.
- [10] 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.
- [11] Dimitri P Bertsekas. 1992. Auction algorithms for network flow problems: A tutorial introduction. *Computational optimization and applications* 1, 1 (1992), 7–66.
- [12] Samuel Blackman and Robert Popoli. 1999. Design and analysis of modern tracking systems. (1999).
- [13] Vladimir L Boginski, Clayton W Commander, Panos M Pardalos, and Yinyu Ye. 2011. Sensors: theory, algorithms, and applications. Vol. 61. Springer Science & Business Media.
- [14] Stephen Boyd and Lieven Vandenberghe. 2004. Convex optimization. Cambridge university press.
- [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] Michael M Bronstein, Alexander M Bronstein, Fabrice Michel, and Nikos Paragios. 2010. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10). IEEE, 3594–3601.
- [17] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine 34, 4 (2017), 18–42.
- [18] Sebastian Bullinger, Christoph Bodensteiner, and Michael Arens. 2017. Instance Flow Based Online Multiple Object Tracking. arXiv:1703.01289 [cs.CV] (2017).
- [19] 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, 1846–1853.
- [20] A Caponi. 2004. Polynomial time algorithm for data association problem in multitarget tracking. *IEEE Trans. Aerospace Electron. Systems* 40, 4 (2004), 1398–1410.
- [21] 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.
- [22] 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, 826–833.
- [23] Cheng Cheng, Jean-Yves Tourneret, Quan Pan, and Vincent Calmettes. 2016. Detecting, estimating and correcting multipath biases affecting GNSS signals using a marginalized likelihood ratio-based method. Signal Processing 118 (2016), 221–234.
- [24] 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.
- [25] Chee-Yee Chong. 2012. Graph approaches for data association. In Proceedings of the 15th International Conference on Information Fusion (FUSION'12). IEEE, 1578–1585.
- [26] Chee-Yee Chong and Shozo Mori. 2006. Metrics for feature-aided track association. In *Proceedings of the 9th International Conference on Information Fusion (FUSION'06)*. IEEE, 1–8.
- [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, 1744–1751.
- [28] Hanjun Dai, Bo Dai, and Le Song. 2016. Discriminative embeddings of latent variable models for structured data. In Proceedings of the 33rd International Conference on Machine Learning (ICML'16). 2702–2711.
- [29] Hanjun Dai, Elias B Khalil, Yuyu Zhang, Bistra Dilkina, and Le Song. 2017. Learning Combinatorial Optimization Algorithms over Graphs. *Advances in Neural Information Processing Systems* (2017).
- [30] Mark A Davenport, Chinmay Hegde, Marco F Duarte, and Richard G Baraniuk. 2010. Joint manifolds for data fusion. *IEEE Transactions on Image Processing* 19, 10 (2010), 2580–2594.
- [31] Somnath Deb, Krishna R Pattipati, and Yaakov Bar-Shalom. 1993. A multisensor-multitarget data association algorithm for heterogeneous sensors. IEEE Trans. Aerospace Electron. Systems 29, 2 (1993), 560–568.

- [32] Somnath Deb, Murali Yeddanapudi, Krishna Pattipati, and Yaakov Bar-Shalom. 1997. A generalized SD assignment algorithm for multisensor-multitarget state estimation. *IEEE Trans. Aerospace Electron. Systems* 33, 2 (1997), 523–538.
- [33] 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. *The International Journal of Robotics Research* (2017).
- [34] 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, 1–6.
- [35] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An Open Urban Driving Simulator. In Proceedings of the 1st Annual Conference on Robot Learning. 1–16.
- [36] Anna Ellis and James Ferryman. 2010. PETS2010: Dataset and challenge. In 2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 135–142.
- [37] Patrick Emami, Lily Elefteriadou, and Sanjay Ranka. 2017. Tracking Vehicles Equipped with Dedicated Short-Range Communication at Traffic Intersections. In Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications (DIVANet '17). ACM, New York, NY, USA, 9–16. https://doi.org/10.1145/3132340.3132356
- [38] 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 Press.
- [39] 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). Springer, 774–790.
- [40] Kuan Fang, Yu Xiang, and Silvio Savarese. 2017. Recurrent Autoregressive Networks for Online Multi-Object Tracking. arXiv:1711.02741 [cs.CV] (2017).
- [41] 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.
- [42] Pasquale Foggia, Gennaro Percannella, and Mario Vento. 2014. Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence* 28, 01 (2014), 1450001.
- [43] Yoav Freund and Robert E Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In European Conference on Computational Learning Theory. Springer, 23–37.
- [44] Andrew H Gee and Richard W Prager. 1994. Polyhedral combinatorics and neural networks. Neural computation 6, 1 (1994), 161–180.
- [45] 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)*.
- [46] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural Message Passing for Quantum Chemistry. In Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research), Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, International Convention Centre, Sydney, Australia, 1263–1272.
- [47] Steven Gold, Anand Rangarajan, et al. 1996. Softmax to softassign: Neural network algorithms for combinatorial optimization. Journal of Artificial Neural Networks 2, 4 (1996), 381–399.
- [48] Palash Goyal and Emilio Ferrara. 2017. Graph Embedding Techniques, Applications, and Performance: A Survey. arXiv:1705.02801 [cs.SI] (2017).
- [49] Don A Grundel, Pavlo A Krokhmal, Carlos AS 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.
- [50] R. Guerrero-Gomez-Olmedo, R. J. Lopez-Sastre, S. Maldonado-Bascon, and A. Fernandez-Caballero. 2013. Vehicle Tracking by Simultaneous Detection and Viewpoint Estimation. In *IWINAC 2013, Part II, LNCS 7931*. 306–316.
- [51] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In Advances in Neural Information Processing Systems. 1025–1035.
- [52] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation Learning on Graphs: Methods and Applications. arxiv:1709.05584 [cs.SI] (2017).
- [53] John M Hammersley and Peter Clifford. 1971. Markov fields on finite graphs and lattices. (1971).
- [54] 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.
- [55] Roberto Henschel, Laura Leal-Taixe, Daniel Cremers, and Bodo Rosenhahn. 2017. Improvements to Frank-Wolfe optimization for multi-detector multi-object tracking. arXiv:1705.08314 [cs.CV] (2017).
- [56] John J Hopfield and David W Tank. 1985. Neural computation of decisions in optimization problems. Biological cybernetics 52, 3 (1985), 141–152.

:32 P. Emami et al.

[57] 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). Springer, 788–801.

- [58] 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, 1–8.
- [59] Jean-Philippe Jodoin, Guillaume-Alexandre Bilodeau, and Nicolas Saunier. 2014. Urban tracker: Multiple object tracking in urban mixed traffic. In 2014 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 885–892.
- [60] 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.
- [61] Alla R. Kammerdiner. 2008. Encyclopedia of Optimization (2nd ed.). Chapter Multimdimensional Assignment Problem, 2396–2402.
- [62] Lance M Kaplan, Yaakov Bar-Shalom, and William D Blair. 2008. Assignment costs for multiple sensor track-to-track association. IEEE Trans. Aerospace Electron. Systems 44, 2 (2008).
- [63] 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.
- [64] Suna Kim, Suha Kwak, Jan Feyereisl, and Bohyung Han. 2013. Online Multi-target Tracking by Large Margin Structured Learning. Springer Berlin Heidelberg, 98–111. https://doi.org/10.1007/978-3-642-37431-9_8
- [65] Thomas N Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [stat.ML] (2016).
- [66] 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
- [67] 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, 685–692.
- [68] 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, 1217–1224.
- [69] 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 Eighteenth International Conference on Machine Learning (ICML '01). San Francisco, CA, USA, 282–289.
- [70] Roslyn A Lau and Jason L Williams. 2011. Multidimensional assignment by dual decomposition. In Proceedings of the 2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'11). IEEE, 437–442.
- [71] 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). Springer, 43–51.
- [72] Laura Leal-Taixe, Cristian Canton-Ferrer, and Konrad Schindler. 2016. Learning by Tracking: Siamese CNN for Robust Target Association. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR'16)*.
- [73] Laura Leal-Taixe, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. 2015. Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942 [cs.CV] (2015).
- [74] Laura Leal-Taixe, 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 [cs.CV] (2017).
- [75] 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.
- [76] 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, 2953–2960.
- [77] 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. 98–109. https://doi.org/10.1007/978-3-642-38868-2 9
- [78] Scott W Linderman, Gonzalo E Mena, Hal Cooper, Liam Paninski, and John P Cunningham. 2017. Reparameterizing the Birkhoff Polytope for Variational Permutation Inference. arXiv:1710.09508 [stat.ML] (2017).
- [79] Wenhan Luo, Junliang Xing, Xiaoqin Zhang, Xiaowei Zhao, and Tae-Kyun Kim. 2014. Multiple object tracking: A literature review. arXiv:1409.7618 [cs.CV] (2014).
- [80] Ronald PS Mahler. 2007. Statistical multisource-multitarget information fusion. Artech House, Inc.
- [81] 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, 630–635.

- [82] Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. 2018. Learning Latent Permutations with Gumbel-Sinkhorn Networks. In *International Conference on Learning Representations*.
- [83] 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, 719–726.
- [84] 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.
- [85] 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] (2016).
- [86] Anton Milan, Laura Leal-Taixe, 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.
- [87] 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.
- [88] Anton Milan, Seyed Hamid Rezatofighi, Ravi Garg, Anthony R Dick, and Ian D Reid. 2017. Data-Driven Approximations to NP-Hard Problems.. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*. 1453–1459.
- [89] 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.
- [90] Shozo Mori, Kuo-Chu Chang, and Chee-Yee Chong. 2014. Performance prediction of feature-aided track-to-track association. IEEE Trans. Aerospace Electron. Systems 50, 4 (2014), 2593–2603.
- [91] James Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics* 5, 1 (1957), 32–38.
- [92] 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.
- [93] 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, 735–742.
- [94] Carlos AS Oliveira and Panos M Pardalos. 2004. Randomized parallel algorithms for the multidimensional assignment problem. *Applied Numerical Mathematics* 49, 1 (2004), 117–133.
- [95] Peter Ondruska and Ingmar Posner. 2016. Deep Tracking: Seeing Beyond Seeing Using Recurrent Neural Networks. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16). AAAI Press, 3361–3367.
- [96] 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, 1–8.
- [97] 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, 1988–1995.
- [98] 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.
- [99] Federico Perea and Huub W De Waard. 2011. Greedy and K-Greedy Algorithms for Multidimensional Data Association. IEEE Trans. Aerospace Electron. Systems 47, 3 (2011), 1915–1925.
- [100] 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, 1201–1208.
- [101] 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.
- [102] Aubrey B Poore and Sabino Gadaleta. 2006. Some assignment problems arising from multiple target tracking. Mathematical and Computer Modelling 43, 9 (2006), 1074–1091.
- [103] Robert L Popp, Krishna R Pattipati, and Yaakov Bar-Shalom. 2001. m-best SD assignment algorithm with application to multitarget tracking. IEEE Trans. Aerospace Electron. Systems 37, 1 (2001), 22–39.
- [104] 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 IEEE Aerospace Conference, Vol. 5. IEEE, 71–84.
- [105] MV 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, 511–516.
- [106] 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.

:34 P. Emami et al.

[107] Mauricio G.C. Resende and Celso C. Ribeiro. 2016. Optimization by GRASP. Springer New York. https://doi.org/10. 1007/978-1-4939-6530-4

- [108] 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, 765–772.
- [109] S. H. Rezatofighi, V. Kumar BG, A. Milan, E. Abbasnejad, A. Dick, and I. Reid. 2017. DeepSetNet: Predicting Sets with Deep Neural Networks. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV'17)*.
- [110] 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.
- [111] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. 2017. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. arXiv:1701.01909 [cs.CV] (2017).
- [112] Alon Schclar. 2012. Multi-sensor fusion via reduction of dimensionality. arXiv:1211.2863 [cs.CV] (2012).
- [113] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. 2017. Deep Network Flow for Multi-Object Tracking. arXiv:1706.08482 [cs.CV] (2017). Update to CVPR?
- [114] 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)*.
- [115] 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, 409–415.
- [116] K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. arxiv:1409.1556 [cs.CV] (2014).
- [117] 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.
- [118] 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.
- [119] 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.
- [120] Elias Strigel, Daniel Meissner, Florian Seeliger, Benjamin Wilking, and Klaus Dietmayer. 2014. The ko-per intersection laserscanner and video dataset. In 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 1900–1901.
- [121] 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). Springer, 100–111.
- [122] Yicong Tian and Mubarak Shah. 2016. On duality of multiple target tracking and segmentation. arXiv:1610.04542 [cs.CV] (2016).
- [123] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order matters: Sequence to sequence for sets. In 4th International Conference on Learning Representations (ICLR'16).
- [124] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In Advances in Neural Information Processing Systems. 2692–2700.
- [125] 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.
- [126] 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. The University; 1998, 1565–1575.
- [127] 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.
- [128] 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.
- [129] 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.
- [130] 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.
- [131] Shaofei Wang and Charless C Fowlkes. 2015. Learning Optimal Parameters For Multi-target Tracking. In *Proceedings* of the British Machine Vision Conference (BMVC), Vol. 1. BMVA Press, 4.1–4.13.

- [132] 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.
- [133] 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). Springer, 200–214.
- [134] 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
- [135] 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] abs/1511.04136 (2015).
- [136] Jason Williams and Roslyn Lau. 2014. Approximate evaluation of marginal association probabilities with belief propagation. IEEE Trans. Aerospace Electron. Systems 50, 4 (2014), 2942–2959.
- [137] 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, 175–180.
- [138] 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, 1–8.
- [139] 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, 1948–1955.
- [140] Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-Shot Learning the Good, the Bad and the Ugly. In In the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17).
- [141] 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.
- [142] 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.
- [143] 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, 1233–1240.
- [144] 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, 2034–2041.
- [145] Min Yang, Yuwei Wu, and Yunde Jia. 2017. A Hybrid Data Association Framework for Robust Online Multi-Object Tracking. arXiv:1703.10764 [cs.CV] (2017). Uses CNN for appearance matching.
- [146] Yuebin Yang and Guillaume-Alexandre Bilodeau. 2016. Multiple Object Tracking with Kernelized Correlation Filters in Urban Mixed Traffic. arXiv:1611.02364 [cs.CV] (2016). Uses deep semantic segmentation to find correspondences and define association cost matrix for a LAP.
- [147] Alper Yilmaz, Omar Javed, and Mubarak Shah. 2006. Object tracking: A survey. Acm computing surveys (CSUR) 38, 4 (2006), 13.
- [148] Haichao Zhang, Thomas S Huang, Nasser M Nasrabadi, and Yanning Zhang. 2011. Heterogeneous multi-metric learning for multi-sensor fusion. In *Proceedings of the 14th International Conference on Information Fusion (FUSION'11)*. IEEE, 1–8.
- [149] 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, 1–8.
- [150] Yanning Zhang, Haichao Zhang, Nasser M Nasrabadi, and Thomas S Huang. 2013. Multi-metric learning for multi-sensor fusion based classification. *Information Fusion* 14, 4 (2013), 431–440.
- [151] 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, 1–8.