Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor

Wongun Choi
NEC Laboratories America, Cupertino, CA, USA
wongun@nec-labs.com

Abstract

In this paper, we tackle two key aspects of multiple target tracking problem: 1) designing an accurate affinity measure to associate detections and 2) implementing an efficient and accurate (near) online multiple target tracking algorithm. As for the first contribution, we introduce a novel Aggregated Local Flow Descriptor (ALFD) that encodes the relative motion pattern between a pair of temporally distant detections using long term interest point trajectories (IPTs). Leveraging on the IPTs, the ALFD provides a robust affinity measure for estimating the likelihood of matching detections regardless of the application scenarios. As for another contribution, we present a Near-Online Multi-target Tracking (NOMT) algorithm. The tracking problem is formulated as a data-association between targets and detections in a temporal window, that is performed repeatedly at every frame. While being efficient, NOMT achieves robustness via integrating multiple cues including ALFD metric, target dynamics, appearance similarity, and long term trajectory regularization into the model. Our ablative analysis verifies the superiority of the ALFD metric over the other conventional affinity metrics. We run a comprehensive experimental evaluation on two challenging tracking datasets, KITTI [16] and MOT [2] datasets. The NOMT method combined with ALFD metric achieves the best accuracy in both datasets with significant margins (about 10% *higher MOTA) over the state-of-the-art.*

1. Introduction

The goal of multiple target tracking is to automatically identify objects of interest and reliably estimate the motion of targets over the time. Thanks to the recent advancement in image-based object detection methods [9, 13, 17, 34], *tracking-by-detection* [4, 6, 10, 25, 27] has become a popular framework to tackle the multiple target tracking problem. The advantages of the framework are that it naturally identifies new objects of interest entering the scene, that it can handle video sequences recorded using mobile platforms, and that it is robust to a target drift. The key challenge in this framework is to accurately group the detections into individual targets with high accuracy (*data association*), so one target could be fully represented by a single estimated trajectory. Mistakes made in the identity maintenance could result in a catastrophic failure in many high level reasoning tasks, such as future motion prediction, target behavior analysis, etc.

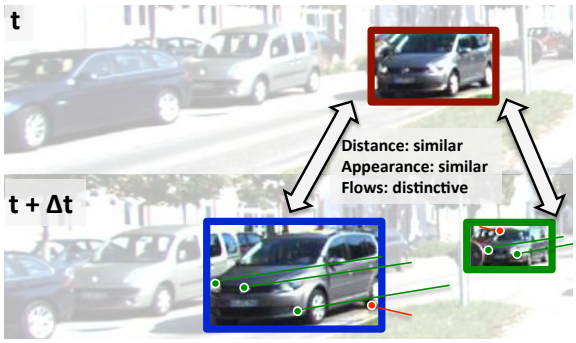

Figure 1. Bounding box distance and appearance similarity are popularly used affinity metrics in the multiple target tracking literature. However, in real-world crowded scenes, they are often ambiguous to successfully distinguish adjacent or similar looking targets. Yet, the optical flow trajectories provide more reliable measure to compare different detections across time. Although individual trajectory may be inaccurate (red line), collectively they provide strong information to measure the affinity. We propose a novel Aggregated Local Flow Descriptor that exploits the optical flow reliably in the multiple target tracking problem.

To implement a highly accurate multiple target tracking algorithm, it is important to have a robust data association model and an accurate measure to compare two detections across time (pairwise affinity measure). Recently, much work is done in the design of the data association algorithm using global (batch) tracking framework [4, 25, 27, 37]. Compared to the online counterparts [6, 7, 10, 22], these methods have a benefit of considering all the detections over entire time frames. With a help of clever optimization algorithms, they achieve higher data association accuracy than traditional online tracking frameworks. However, the application of these methods is fundamentally limited to post-analysis of video sequences. On the other hand, the pairwise affinity measure is relatively less investigated in the recent literature despite its importance. Most methods adopt weak affinity measures (see Fig. 1) to compare two detections across time, such as spatial affinity (e.g. bound-

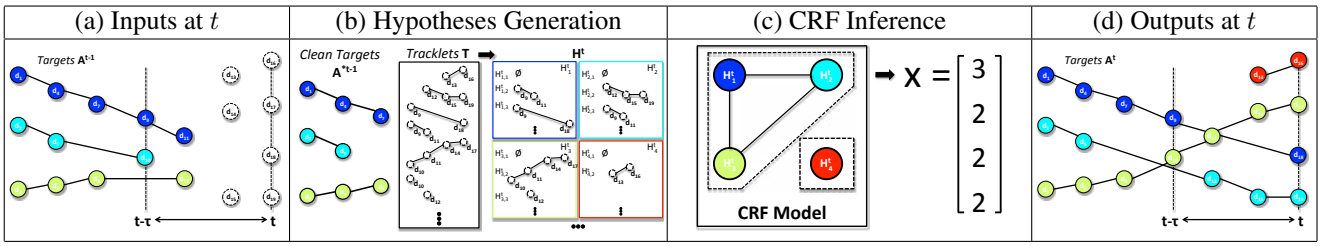

Figure 2. Schematic illustration of NOMT algorithm. (a) Given a set of existing targets $\mathbb{A}^{t-1}$ and detections $\mathbb{D}^t_{t-\tau}$, (b) our method generates a set of candidate hypotheses $\mathbb{H}^t$ using tracklets $\mathcal{T}$. Constructing a CRF model with the hypotheses, (c) we select the most consistent solution x using our inference algorithm and (d) output targets $\mathbb{A}^t$ are obtained by augmenting previous targets $\mathbb{A}^{t-1}$ with the solution $\mathbb{H}^t(\hat{x})$. See text (Sec. 4) for the details.

ing box overlap or euclidean distance [3, 4, 30]) or simple appearance similarity (e.g. intersection kernel with color histogram [31]). In this paper, we address the two key challenging questions of the multiple target tracking problem: 1) how to accurately measure the pairwise affinity between two detections (i.e. likelihood to link the two) and 2) how to efficiently apply the ideas in global tracking algorithms into an online application.

As for the first contribution, we present a novel *Aggregated Local Flow Descriptor* (ALFD) that encodes the relative motion pattern between two detection boxes in different time frames (Sec. 3). By aggregating multiple local interest point trajectories (IPTs), the descriptor encodes how the IPTs in a detection moves with respect to another detection box, and vice versa. The main intuition is that although each individual IPT may have an error, *collectively* they provide a strong cue for comparing two detections. With a learned model, we observe that ALFD provides a strong affinity measure. As for the second contribution, we propose an efficient *Near-Online Multi-target Tracking* (NOMT) algorithm. Incorporating the robust ALFD descriptor as well as long-term motion/appearance models, the algorithm produces highly accurate trajectories, while preserving the causality and real-time ($\sim$ 10 FPS) property. In every frame t, the algorithm solves the global data association problem between targets and all the detections in a temporal window $[t\text{-}\tau, t]$ of size τ (see Fig. 2). The key property is that the algorithm has the potential to fix any past association error within the temporal window when more detections are provided. In order to achieve both accuracy and efficiency, the algorithm generates candidate hypothetical trajectories using ALFD driven tracklets and solve the association problem with a parallelized junction tree algorithm (Sec. 4). We perform a comprehensive experimental evaluation on two challenging datasets: KITTI [16] and MOT Challenge [2] datasets. The proposed algorithm achieves the best accuracy with a large margin over the state-of-the-arts (including batch algorithms) in both datasets, demonstrating the superiority of our algorithm.

2. Background

Given a video sequence $V_1^T = \{I_1, I_2, ..., I_T\}$ of length T and a set of detection hypotheses $\mathbb{D}_1^T = \{d_1, d_2, ..., d_N\}$, where d_i is parameterized by the frame number t_i, a bounding box $(d_i[x], d_i[y], d_i[w], d_i[h])$[1], and the score s_i, the goal of multiple target tracking is to find a coherent set of targets (associations) $\mathbb{A} = \{A_1, A_2, ..., A_M\}$, where each target A_m are parameterized by a set of detection indices (e.g. $A_1 = \{d_1, d_{10}, d_{23}\}$) during the time of presence.

Data Association Models: Most of multiple target tracking algorithms/systems can be classified into two categories: online method and global (batch) method. Online algorithms [6, 7, 10, 22, 29] are formulated to find the association between existing targets and detections in the current time frame: $(V_t^t, \mathbb{D}_t^t, \mathbb{A}^{t-1}) \rightarrow \mathbb{A}^t$. The advantages of online formulation are: 1) it is applicable to online/real-time scenario and 2) it is possible to take advantage of targets' dynamics information available in $\mathbb{A}^{t-1}$. Such methods, however, are often prone to association errors since they consider only one frame when making the association. To avoid such errors, [6] adopts conservative association threshold together with detection confidence maps, or [7, 22, 29] model interactions between targets.

Recently, global algorithms [3, 4, 27, 30, 37] became much popular in the community, as more robust association is achieved when considering long-term information in the association process. One common approach is to formulate the tracking as the network flow problem to directly obtain the targets from detection hypothesis [4, 30, 37]; i.e. $(V_1^T, \mathbb{D}_1^T) \rightarrow \mathbb{A}^T$. Although they have shown promising accuracy in multiple target tracking, the methods are often over-simplified for the tractability concern. They ignore useful target level information, such as target dynamics and interaction between targets (occlusion, social interaction, etc). Instead of directly solving the problem at one step, other employ an iterative algorithm that progressively refines the target association [3, 19, 25, 27]; i.e. $(V_1^T, \mathbb{D}_1^T, \mathbb{A}_i^T) \rightarrow \mathbb{A}_{i+1}^T$, where i represent an iteration. Starting from short trajectories (tracklet), [19, 25] associate them into longer targets in a hierarchical fashion. [3, 27] iterate between two modes, association and continuous estimation. Since these methods obtain intermediate target information, targets' dynamics, interaction and high-order statistics on the trajectories could be accounted that can lead to a better association accuracy. However, it is unclear how

[1] $[x], [y], [w], [h]$ operators represent the x, y, width and height value, respectively.

to seamlessly extend such models to an online application.

We propose a novel framework that can fill in the gap between the online and global algorithms. The task is defined as to solve the following problem: $(V_1^t, \mathbb{D}_{t-\tau}^t, \mathbb{A}^{t-1}) \rightarrow \mathbb{A}^t$ in each time frame t, where τ is pre-defined temporal window size. Our algorithm behaves similar to the online algorithm in that it outputs the association in every time frame. The critical difference is that any decision made in the past is subject to change once more observations are available. The association problems in each temporal window are solved using a newly proposed global association algorithm. Our method is also reminiscent of iterative global algorithm, since we augment all the track iteratively (one iteration per frame) considering multiple frames, that leads to a better association accuracy.

Affinity Measures in Visual Tracking: The importance of a robust pairwise affinity measure (i.e. likelihood of d_i and d_j being the same target) is relatively less investigated in the multi-target tracking literature. Most of the recent literature [3, 4, 30, 31] employs a spatial distance and/or an appearance similarity with simple features (such as color histograms). In order to learn a discriminative affinity metric, Kuo *et al.* [25] introduces an online appearance learning with boosting algorithm using various feature inputs such as HoG [8], texture feature, and RGB color histogram. Milan *et al.* [27] and Zamir *et al.* [31] proposed to use a global appearance consistency measure to ensure a target has a similar (or smoothly varying) appearance over a long term. Although there have been many works exploiting appearance information or spatial smoothness, few attempts are made to incorporate optical flows in the multi-target tracking literature. Kalal *et al.* [20] showed a promising result in tracking a single target using optical flow trajectories together with median filtering. Similarly, Everingham *et al.* [11] calculates the portion of inlier trajectories over the outliers between face detections to cluster them in a movie. In contrast to these methods, ALFD encodes more fine grained description of the local motion patterns and enables us to learn a discriminative model via an explicit descriptor. Recently, Fragkiadaki *et al.* [14] introduced a method to track multiple targets while jointly clustering optical flow trajectories. The work presents a promising result, but the model is complicated due to the joint inference on both target and flow level association. In contrast, our ALFD provides a strong pairwise affinity measure that is generally applicable in any tracking model.

3. Aggregated Local Flow Descriptor

The Aggregated Local Flow Descriptor (ALFD) encodes the relative motion pattern between two bounding boxes in a temporal distance ($\Delta t = |t_i - t_j|$) given interest point trajectories [33]. The main intuition in ALFD is that if the two boxes belong to the same target, we shall observe many supporting IPTs in the same relative location with respect to the boxes. In order to make it robust against small localization errors in detections, targets' orientation change, and outliers/errors in the IPTs, we build the ALFD using spatial histograms. Once the ALFD is obtained, we measure the affinity between two detections ($a_A(d_i, d_j)$) using the linear product of a learned model parameter ($w_{\Delta t}$) and ALFD ($\rho(d_i, d_j)$), i.e. $a_A(d_i, d_j) = w_{\Delta t} \cdot \rho(d_i, d_j)$. In the following subsections, we discuss the details of the design.

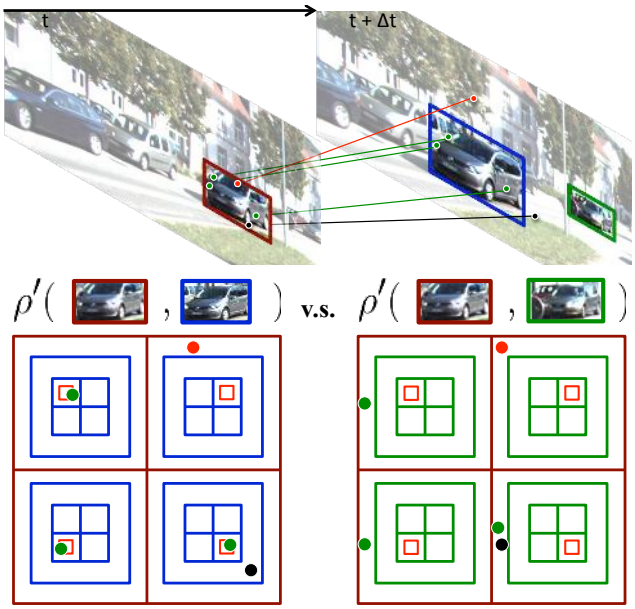

Figure 3. Illustrative figure for unidirectional ALFDs $\rho'(d_i, d_j)$. In the top figure, we show detections as colored bounding boxes (d_{red}, d_{blue}, and d_{green}). A pair of circles with connecting lines represent IPTs that are existing in both t and $t + \triangle t$ and located inside of the d_{red} at t. We draw the accurate (green), outlier (black), and erroneous (red) IPTs. In the bottom figure, we show two exemplar unidirectional ALFDs ρ' for (d_{red}, d_{blue}) and (d_{red}, d_{green}). The red grids (2×2) represent the IPTs' location at t relative to d_{red}. The blue and green grids inside of each red bin ($2 \times 2 + 2$ external bins) shows the IPTs' location at $t + \triangle t$ relative to the corresponding boxes. IPTs in the grid bins with a red box are the one observed in the same relative location. Intuitively, the more IPTs are observed in the bins, the more likely the two detections belong to the same target. In contrast, wrong matches will have more supports in the outside bins. The illustration is shown using 2×2 grids to avoid clutter. We use 4×4 in practice that yields a $16 \times (16 + 2) = 288$ dimensional vector.

3.1. Interest Point Trajectories

We obtain Interest Point Trajectories using a local interest point detector [5, 32] and optical flow algorithm [5, 12]. The algorithm is designed to produce a set of long and accurate point trajectories, combining various well-known computer vision techniques. Given an image I_t, we run the FAST interest point detector [5, 32] to identify "good

points" to track. In order to avoid having redundant points, we compute the distance between the newly detected interest points and the existing IPTs and keep the new points sufficiently far from the existing IPTs (> 4 px). The new points are assigned unique IDs. For all the IPTs in t, we compute the forward ($t \rightarrow t+1$) and backward ($t+1 \rightarrow t$) optical flow using [5, 12]. The starting points of backward flows are given by the forward flows' end point. Any IPT having a large disagreement between the two (> 10 px) is terminated.

3.2. ALFD Design

Let us define the necessary notations to discuss ALFD. $\kappa_{id} \in \mathcal{K}$ represents one IPT with a unique id. κ_{id} is parameterized by pixel locations $(\kappa_{id}(t)[x], \kappa_{id}(t)[y])$ during the time of presence. We define $\kappa_{id}(t)$ to denote the pixel location at the frame t. If κ_{id} does not exist at t (terminated or not initiated), $\emptyset$ is returned.

We first define a unidirectional ALFD $\rho'(d_i, d_j)$, i.e. motion pattern from d_i to d_j, by aggregating the information from all the IPTs that are located inside of d_i box and existing at t_j. Formally, we define the IPT set as $\mathcal{K}(d_i, d_j) = \{\kappa_{id} | \kappa_{id}(t_i) \in d_i \ \& \ \kappa_{id}(t_j) \neq \emptyset\}$. For each $\kappa_{id} \in \mathcal{K}(d_i, d_j)$, we compute the relative location $r_i(\kappa_{id}) = (x, y)$ of each κ_{id} at t_i by $r_i(\kappa_{id})[x] = (\kappa_{id}(t_i)[x]\text{-}d_i[x])/d_i[w]$ and $r_i(\kappa_{id})[y] = (\kappa_{id}(t_i)[y]\text{-}d_i[y])/d_i[h]$. We compute $r_j(\kappa_{id})$ similarly. Notice that $r_i(\kappa_{id})$ are bounded between $[0, 1]$, but $r_j(\kappa_{id})$ are not bounded since κ_{id} can be outside of d_j. Given the $r_i(\kappa_{id})$ and $r_j(\kappa_{id})$, we compute the corresponding spatial grid bin indices as shown in the Fig. 3 and accumulate the count to build the descriptor. We define 4×4 grids for $r_i(\kappa_{id})$ and $4 \times 4 + 2$ grids for $r_j(\kappa_{id})$ where the last 2 bins are accounting for the outside region of the detection. The first outside bin defines the neighborhood of the detection ($< width/4 \ \& \ < height/4$), and the second outside bin represents any farther region.

Using a pair of unidirectional ALFDs, we define the (undirected) ALFD as $\rho(d_i, d_j) = (\rho'(d_i, d_j) + \rho'(d_j, d_i)) \ / \ n(d_i, d_j)$, where $n(d_i, d_j)$ is a normalizer. The normalizer n is defined as $n(d_i, d_j) = |\mathcal{K}(d_i, d_j)| + |\mathcal{K}(d_j, d_i)| + \lambda$, where $|\mathcal{K}(\cdot)|$ is the count of IPTs and λ is a constant. λ ensures that the L1 norm of the ALFD increases as we have more supporting $\mathcal{K}(d_i, d_j)$ and converges to 1. We use $\lambda = 20$ in practice.

3.3. Learning the Model Weights

We learn the model parameters $w_{\Delta t}$ from a training dataset with a weighted voting. Given a set of detections $\mathbb{D}_1^T$ and corresponding ground truth (GT) target annotations, we first assign the GT target id to each detections. For each detection d_i, we measure the overlap with all the GT boxes in t_i. If the best overlap o_i is larger than 0.5, the corresponding target id (id_i) is assigned. Otherwise, -1 is assigned. For all detections that has $id_i \geq 0$ (positive detections), we collect a set of detections $\mathcal{P}_i^{\Delta t} = \{d_j \in \mathbb{D}_1^T | t_j - t_i = \Delta t\}$. For each pair, we compute the margin m_{ij} as follows: if id_i and id_j are identical, $m_{ij} = (o_i - 0.5) \cdot (o_j - 0.5)$. Otherwise, $m_{ij} = -(o_i - 0.5) \cdot (o_j - 0.5)$. Intuitively, m_{ij} shall have a positive value if the two detections are from the same target, while m_{ij} will have a negative value, if the d_i and d_j are from different targets. The magnitude is weighted by the localization accuracy. Given all the pairs and margins, we learn the model $w_{\Delta t}$ as follows:

$$w_{\Delta t} = \frac{\sum_{\{i \in \mathbb{D}_1^T | id_i \geq 0\}} \sum_{j \in \mathcal{P}_i^{\Delta t}} m_{ij} (\rho'(d_i, d_j) + \rho'(d_j, d_i))}{\sum_{\{i \in \mathbb{D}_1^T | id_i \geq 0\}} \sum_{j \in \mathcal{P}_i^{\Delta t}} |m_{ij}| (\rho'(d_i, d_j) + \rho'(d_j, d_i))} \quad (1)$$

where the division is performed element-wise. The algorithm computes a weighted average with a sign over all the ALFD patterns, where the weights are determined by the overlap between targets and detections. Intuitively, the ALFD pattern between detections that matches well with GT contributes more on the model parameters. The advantage of the weighted voting method is that each element in $w_{\Delta t}$ are bounded in $[-1, 1]$, thus the ALFD metric, $a_A(d_i, d_j)$, is also bounded by $[-1, 1]$ since $||\rho(d_i, d_j)||_1 \leq 1$. We learn $w_{\Delta t}$ using the KITTI 0000 sequence and kept the same parameter throughout all the experiments.

3.4. Properties

In this section, we discuss the properties of ALFD affinity metric $a_A(d_i, d_j)$. Firstly, unlike appearance or spatial metrics, ALFD implicitly exploit the information in all the images between t_i and t_j through IPTs. Secondly, thanks to the collective nature of ALFD design, it provides strong affinity metric over arbitrary length of time. We observe a significant benefit over the appearance or spatial metric especially over a long temporal distance (see Sec. 5.1 for the analysis). Thirdly, it is generally applicable to any scenarios (either static or moving camera) and for any object types (person or car). One disadvantage of the ALFD is that it may become unreliable when there is an occlusion. When an occlusion happens to a target, the IPTs initiated from the target tend to adhere to the occluder. It motivates us to combine target dynamics information discussed in Sec. 4.4.1.

4. Near Online Multi-target Tracking (NOMT)

We employ a near-online multi-target tracking framework that updates and outputs targets $\mathbb{A}^t$ in each time frame considering inputs in a temporal window $[t\text{-}\tau, t]$. We implement the NOMT algorithm with a hypothesis generation and selection scheme. For the convenience of discussion, we define *clean* targets $\mathbb{A}^{*t-1} = \{A_1^{*t-1}, A_2^{*t-1}, ...\}$ that exclude all the associated detections in $[t\text{-}\tau, t\text{-}1]$. Given a set of detections in $[t\text{-}\tau, t]$ and clean targets $\mathbb{A}^{*t-1}$, we generate

multiple target hypotheses $H_m^t = \{\emptyset, H_{m,2}^t, H_{m,3}^t ...\}$ for each target A_m^{*t-1} as well as newly entering targets, where $\emptyset$ (empty hypothesis) represents the termination of the target and each $H_{m,k}^t$ indicates a set of candidate detections in $[t\text{-}\tau, t]$ that can be associated to a target (Sec. 4.2). Each $H_{m,k}^t$ may contain 0 to τ detections (at one time frame, there can be 0 or 1 detection). Given the set of hypotheses for all the existing and new targets, the algorithm finds the most consistent set of hypotheses (MAP) for all the targets (one for each) using a graphical model (sec. 4.3). As the key characteristic, our algorithm is able to fix any association error (for the detections within the temporal window $[t\text{-}\tau, t]$) made in the previous time frames.

4.1. Model Representation

Before going into the details of each step, we discuss our underlying model representation. The model is formulated as an energy minimization framework; $\hat{x} = \text{argmin}_x \, E(\mathbb{A}^{*t-1}, \mathbb{H}^t(x), \mathbb{D}_{t-\tau}^t, V_1^t)$, where x is an integer state vector indicating which hypothesis is chosen for a corresponding target, $\mathbb{H}^t$ is the set of all the hypotheses $\{H_1^t, H_2^t, ...\}$, and $\mathbb{H}^t(x)$ is a set of selected hypothesis $\{H_{1,x_1}^t, H_{2,x_2}^t, ...\}$. Solving the optimization, the updated targets $\mathbb{A}^t$ can be uniquely identified by augmenting $\mathbb{A}^{*t-1}$ with the selected hypothesis $\mathbb{H}^t(\hat{x})$. Hereafter, we drop V_1^t and $\mathbb{D}_{t-\tau}^t$ to avoid clutters in the equations. The energy is defined as follows:

$$E(\mathbb{A}^{*t-1}, \mathbb{H}^t(x)) = \sum_m \Psi(A_m^{*t-1}, H_{m,x_m}^t) + \sum_{m,l} \Phi(H_{m,x_m}^t, H_{l,x_l}^t) \quad (2)$$

where $\Psi(\cdot)$ encodes individual target's motion, appearance, and ALFD metric consistency, and $\Phi(\cdot)$ represent an exclusive relationship between different targets (e.g. no two targets share the same detection). If there are hypotheses for newly entering targets, we define the corresponding target as an empty set, $A_m^{*t-1} = \emptyset$.

Single Target Consistency

The potential measures the compatibility of a hypothesis H_{m,x_m}^t to a target A_m^{*t-1}. Mathematically, this can be decomposed into unary, pairwise and high order terms as follows:

$$\Psi(A_m^{*t-1}, H_{m,x_m}^t) = \sum_{i \in H_{m,x_m}^t} \psi_u(A_m^{*t-1}, d_i) + \sum_{(i,j) \in H_{m,x_m}^t} \psi_p(d_i, d_j) + \psi_h(A_m^{*t-1}, H_{m,x_m}^t) \quad (3)$$

ψ_u encodes the compatibility of each detection d_i in the target hypothesis H_{m,x_m}^t using the ALFD affinity metric and Target Dynamics feature (Sec. 4.4.1). ψ_p measures the pairwise compatibility (self-consistency of the hypothesis) between detections within H_{m,x_m}^t (Sec. 4.4.2) using the ALFD metric. Finally, ψ_h implements a long-term smoothness constraint and appearance consistency (Sec. 4.4.3).

Mutual Exclusion

This potential penalizes choosing two targets with large overlap in the image plane (repulsive force) as well as duplicate assignments of a detection. The potential can be written as follows:

$$\Phi(H_{m,x_m}^t, H_{l,x_l}^t) = \sum_{f=t-\tau}^{t} \alpha \cdot o^2(d(H_{m,x_m}^t, f), d(H_{l,x_l}^t, f)) + \beta \cdot \mathbb{I}(d(H_{m,x_m}^t, f), d(H_{l,x_l}^t, f)) \quad (4)$$

where $d(H_{m,x_m}^t, f)$ gives the associated detection of H_{m,x_m}^t at time f (if none, $\emptyset$ is returned), $o^2(d_i, d_j) = 2 * IoU(d_i, d_j)^2$, and $\mathbb{I}(d_i, d_j)$ is an indicator function. The former penalizes having too much overlap between hypotheses and the later penalizes duplicate assignments of detections. We use $\alpha = 0.5$ and $\beta = 100$ (large enough to avoid duplicate assignments).

4.2. Hypothesis Generation

Direct optimization over the aforementioned objective function (eq. 2) is infeasible since the space of $\mathbb{H}^t$ is huge in practice. To cope with the challenge, we first propose a set of candidate hypotheses H_m for each target independently (Fig. 2(b)) and find a coherent solution (MAP) using a CRF inference algorithm (sec. 4.3). As all the subsequent steps depend on the generated hypotheses, it is critical to have a comprehensive set of target hypotheses. We generates the hypotheses of existing and new targets using *tracklets*. Notice that following steps could be done in parallel since we generate the hypotheses set per target independently.

Tracklet Generation

For all the confident detections ($\forall d_i \in \mathbb{D}_{t-\tau}^t, \; s.t. \; s_i > 0$), we build a tracklet using the ALFD metric a_A. Starting from one detection tracklet $\mathcal{T}_i = \{d_i\}$, we grow the tracklet by greedily adding the best matching detection d_k such that $k = \text{argmax}_{k \in \mathbb{D}_{t-\tau}^t \setminus \mathcal{T}_i} \, max_{j \in \mathcal{T}_i} a_A(d_j, d_k)$, where $\mathbb{D}_{t-\tau}^t \setminus \mathcal{T}_i$ is the set of detections in $[t\text{-}\tau, t]$ excluding the frames already included in $\mathcal{T}_i$. If the best ALFD metric is lower than 0.4 or $\mathcal{T}_i$ is full (has τ number of detections), the iteration is terminated. In addition, we also extracts the residual detections from each A_m^{t-1} in $[t\text{-}\tau, t]$ to obtain additional tracklets (i.e. $\forall m, A_m^{t-1} \setminus A_m^{*t-1}$). Since there can be identical tracklets, we keep only unique tracklets in the output set $\mathbb{T}$.

Hypotheses for Existing Targets

We generate a set of target hypotheses H_m^t for each existing target A_m^{*t-1} using the tracklets $\mathbb{T}$. In order to avoid having unnecessarily large number of hypotheses, we employ a gating strategy. For each target A_m^{*t-1}, we obtain a target predictor using the least square algorithm with polynomial function [26]. We vary the order of the polynomial depending on the dataset (1 for MOT and 2 for KITTI). If there is an overlap (IoU) larger than a certain threshold between the prediction and the detections in the tracklet $\mathcal{T}_i$ at

any frame in $[t\text{-}\tau, t]$, we add $\mathcal{T}_i$ to the hypotheses set H_m^t. In practice, we use a conservative threshold 0.1 to have a rich set of hypotheses. Too old targets (having no associated detection in $[t\text{-}\tau\text{-}T_{active}, t]$) are ignored to avoid unnecessary computational burden. We use $T_{active} = 1\ sec.$

New Target Hypotheses

Since new targets can enter the scene at any time and at any location, it is desirable to automatically identify new targets. Our algorithm can naturally identify the new targets by treating any tracklet in the set $\mathbb{T}$ as a potential new target. We use a non-maximum suppression on tracklets to avoid having duplicate new targets. For each tracklet $\mathcal{T}_i$, we simply add an empty target $A_m^{*t-1} = \emptyset$ to $\mathbb{A}^{*t-1}$ with an associated hypotheses set $H_m^t = \{\emptyset, \mathcal{T}_i\}$.

4.3. Inference with Dynamic Graphical Model

Once we have all the hypotheses for all the new and existing targets, the problem (eq. 2) can be formulated as an inference problem with an undirected graphical model, where one node represents a target and the states are hypothesis indices as shown in Fig. 2 (c). The main challenges in this problem are: 1) there may exist loops in the graphical model representation and 2) the structure of graph is different depending on the hypotheses at each circumstance. In order to obtain the exact solution efficiently, we first analyze the structure of the graph on the fly and apply appropriate inference algorithms based on the structure analysis.

Given the graphical model, we find independent subgraphs (shown as dashed boxes in Fig. 2 (c)) using connected component analysis [18] and perform individual inference algorithm per each subgraph in parallel. If a subgraph is composed of more than one node, we use junction-tree algorithm [23, 28] to obtain the solution for corresponding subgraph. Otherwise, we choose the best hypothesis for the target. Once the states x are found, we can uniquely identify the new set of targets by augmenting $\mathbb{A}^{*t-1}$ with $\mathbb{H}^t(x)$: $\mathbb{A}^{*t-1} + \mathbb{H}^t(x) \rightarrow \mathbb{A}^t$. This process allows us to adjust any associations of $\mathbb{A}^{t-1}$ in $[t\text{-}\tau, t]$ (i.e. addition, deletion, replacement, or no modification).

4.4. Model Details

4.4.1 Unary potential

As discussed in the previous sections, we utilize the ALFD metric as the main affinity metric to compare detections. The unary potential for each detection in the hypothesis is measured by:

$$\mu_A(A_m^{*t-1}, d_i) = -\sum_{\Delta t \in \mathcal{N}} a_A(d(A_m^{*t-1}, t_i - \Delta t), d_i) \tag{5}$$

where $\mathcal{N}$ is a predefined set of neighbor frame distances and $d(A_m^{*t-1}, t_i)$ gives the associated detection of A_m^{*t-1} at t_i. Although we can define an arbitrarily large set of $\mathcal{N}$, we choose $\mathcal{N} = \{1, 2, 5, 10, 20\}$ for computational efficiency while modeling long term affinity measures.

Although ALFD metric provides very strong information in most of the cases, there are few failure cases including occlusions, erroneous IPTs, etc. To complement such cases, we design an additional Target Dynamics (TD) feature $\mu_T(A_m^{*t-1}, d_i)$. Using the same polynomial least square predictor discussed in Sec. 4.2, we define the feature as follows:

$$\mu_T(A_m^{*t-1}, d_i) = \begin{cases} \infty, & if\ o^2(p(A_m^{*t-1}, t_i), d_i) < 0.5 \\ -\eta^{t_i - f(A_m^{*t-1})} o^2(p(A_m^{*t-1}, t_i), d_i), & otherwise \end{cases} \tag{6}$$

where η is a decay factor (0.98) that discounts long term prediction, $f(A_m^{*t-1})$ denotes the last associated frame of A_m^{*t-1}, o^2 represents IoU^2 discussed in the Sec. 4.1, and p is the polynomial least square predictor described in Sec. 4.2.

Using the two measures, we define the unary potential $\psi_u(A_m^{*t-1}, d_i)$ as:

$$\psi_u(A_m^{*t-1}, d_i) = \min(\mu_A(A_m^{*t-1}, d_i), \mu_T(A_m^{*t-1}, d_i)) - s_i \tag{7}$$

where s_i represents the detection score of d_i. The min operator enables us to utilize the ALFD metric in most cases, but *activate* the TD metric only when it is very confident (more than 0.5 overlap between the prediction and the detection). If A_m^{*t-1} is empty, the potential becomes $-s_i$.

4.4.2 Pairwise potential

The pairwise potential $\psi_p(\cdot)$ is solely defined by the ALFD metric. Similarly to the unary potential, we define the pairwise relationship between detections in H_{m,x_m}^t,

$$\psi_p(d_i, d_j) = \begin{cases} -a_A(d_i, d_j), & if\ |d_i - d_j| \in \mathcal{N} \\ 0, & otherwise \end{cases} \tag{8}$$

It measures the self-consistency of a hypothesis H_{m,x_m}^t.

4.4.3 High-order potential

We incorporate a high-order potential to regularize the target association process with a physical feasibility and appearance similarity. Firstly, inspired by [3, 31], we implement the physical feasibility by penalizing the hypotheses that present an abrupt motion. Secondly, we encodes long term appearance similarity between all the detections in A_m^{*t-1} and H_{m,x_m}^t similarly to [31]. The intuition is encoded by the following potential:

$$\psi_h(A_m^{*t-1}, H_{m,x_m}^t) = \gamma \cdot \sum_{i \in H_{m,x_m}^t} \xi(p(A_m^{*t-1} \cup H_{m,x_m}^t, t_i), d_i) + \epsilon \cdot \sum_{(i,j) \in A_m^{*t-1} \cup H_{m,x_m}^t} \theta - K(d_i, d_j) \tag{9}$$

where γ, ϵ, θ are scalar parameters, $\xi(a, b)$ measures the sum of squared distances in $(x, y, height)$ of the two boxes,

	KITTI 0001: Cars, Mobile camera				PETS09-S2L1: Pedestrians, Static camera			
Metric	$\triangle t:1$	$\triangle t:5$	$\triangle t:10$	$\triangle t:20$	$\triangle t:1$	$\triangle t:5$	$\triangle t:10$	$\triangle t:20$
ALFD	**0.91**	**0.84**	**0.80**	**0.71**	**0.88**	**0.83**	**0.78**	**0.68**
NDist2	0.81	0.32	0.15	0.06	0.85	0.67	0.55	0.41
HistIK	0.81	0.62	0.51	0.38	0.76	0.65	0.60	0.51

Table 1. AUC of affinity metrics for varying $\triangle t$. Notice that ALFD provides a robust affinity metric even at 20 frames distance. The results verify that ALFD provides stable affinity measure regardless of object type or the camera motion.

that is normalized by the mean height of p in $[t\text{-}\tau, t]$, and $K(d_i, d_j)$ represents the intersection kernel for color histograms associated with the detections. We use a pyramid of LAB color histogram where the first layer is the full box and the second layer is 3×3 grids. Only the A and B channels are used for the histogram with 4 bins per each channel (resulting in $4 \times 4 \times (1 + 9)$ bins). We use $(\gamma, \epsilon, \theta) = (20, 0.4, 0.8)$ in practice.

5. Experimental Evaluation

In order to evaluate the proposed algorithm, we use the KITTI object tracking benchmark [16] and MOT challenge dataset [2]. KITTI tracking benchmark is composed of about $19,000$ frames ($\sim$ 32 minutes). The dataset is composed of 21 training and 29 testing video sequences that are recorded using cameras mounted on top of a moving vehicle. Each video sequence has a variable number of frames from 78 to 1176 frames having a variable number of target objects (*Car, Pedestrian, and Cyclist*). The videos are recorded at 10 FPS. The dataset is very challenging since 1) the scenes are crowded (occlusion and clutter), 2) the camera is not stationary, and 3) target objects appears in arbitrary location with variable sizes. Many conventional assumptions/techniques adopted in multiple target tracking with a surveillance camera is not applicable in this case (e.g. fixed entering/exiting location, background subtraction, etc). MOT challenge is composed of $11,286$ frames ($\sim$ 16.5 minutes) with varying FPS. The dataset is composed of 11 training and 11 testing video sequences. Some of the videos are recorded using mobile platform and the others are from surveillance videos. All the sequences contain only Pedestrians. As it is composed of videos with various configuration, tracking algorithms that are particularly tuned for a specific scenario would not work well in general. For the evaluation, we adopt the widely used CLEAR MOT tracking metrics [21]. For a fair comparison to the other methods, we use the reference object detections provided by the both datasets.

5.1. ALFD Analysis

We first run an ablative analysis on our ALFD affinity metric. We choose two sequences, KITTI's *0001* and MOT's *PETS09-S2L1* both from the training sets, for the analysis. Given all the detections and the ground truth annotations, we first find the label association between detections and annotations. For each detection, we assign ground truth id if there is larger than 0.5 overlap. We collect all possible pairs of detections in $1, 5, 10, 20$ frame distance (Δt), to obtain the positive and negative pairs. As the baseline affinity measures, we use the L2 distance between bottom center of the detections that is normalized by the mean height of the two (NDist2) and the intersection kernel between the color histograms of the two (HistIK). Table. 1 shows the Area Under Curve (AUC) of each affinity metric [2]. We observe that ALFD affinity metric performs the best in all temporal distance regardless of the camera configuration and object type. As the temporal distance increases, the other metrics become quickly unreliable as expected, whereas our ALFD metric still provides a strong cue to compare different detections.

	Det.	Method	MOTA ↑	MOTP ↑	MT ↑	ML ↓	IDS ↓	FRG ↓
Car Tracking Benchmark								
DPMF [30]	[13]	Batch	36.62 %	78.49 %	11.13 %	39.18 %	2738	3240
TBD [15]	[13]	Batch	52.44 %	78.47 %	13.87 %	34.30 %	33	538
CEM [27]	[13]	Batch	48.23 %	77.26 %	14.48 %	33.99 %	125	398
RMOT [36]	[13]	Online	49.87 %	75.33 %	15.24 %	33.54 %	51	385
HM	[13]	Online	58.30 %	**78.79** %	26.98 %	30.18 %	28	251
NOMT	[13]	Online	**63.27** %	78.32 %	**31.55** %	**27.59** %	**13**	**155**
RMOT [36]	[34]	Online	60.46 %	75.57 %	26.98 %	**11.13** %	216	742
HM	[34]	Online	69.86 %	**80.10** %	38.72 %	15.09 %	109	372
NOMT	[34]	Online	**72.62** %	79.55 %	**43.14** %	14.48 %	**38**	**227**
Pedestrian Tracking Benchmark								
CEM [27]	[13]	Batch	18.18 %	**68.48** %	7.90 %	52.92 %	96	**610**
RMOT [36]	[13]	Online	25.47 %	68.06 %	9.97 %	47.42 %	81	692
HM	[13]	Online	17.26 %	67.99 %	11.34 %	51.55 %	73	743
NOMT	[13]	Online	**25.55** %	67.75 %	**14.43** %	**42.61** %	**34**	800
RMOT [36]	[34]	Online	36.42 %	71.02 %	16.84 %	41.24 %	156	760
HM	[34]	Online	31.43 %	71.14 %	17.18 %	42.27 %	186	870
NOMT	[34]	Online	**38.98** %	**71.45** %	**23.37** %	**34.71** %	**63**	**672**

Table 2. Multiple Target tracking accuracy for KITTI Car/Pedestrian tracking benchmark. ↑ represents that high numbers are better for the metric and ↓ means the opposite. The best numbers in each column are bold-faced. We use $\tau = 10$ for NOMT and NOMT+[34]. Numbers are updated.

5.2. KITTI Testing Benchmark Evaluation

Table. 2 summarizes the evaluation accuracy of our method (**NOMT**) and the other state-of-the-art algorithms on the whole 28 test video sequences[3]. We also implemented an online tracking algorithm with the Hungarian method [24] (**HM**) using our unary match function. Any match cost larger than -0.5 is set to be an invalid match. In following evaluations, we set the temporal window $\tau = 10$ and filter out targets that either have only one detection or a median detection score lower than 0. We use the Kalman Filter [35] to obtain continuous trajectories out of discrete detection sets $\mathbb{A}$. Since the KITTI evaluation system does not provide results on *Cyclist* category (due to lack of sufficient data), we report the accuracy of *Car* and *Pedestrian* categories. As for the detection inputs, we use two sets of reference detections ([13] and [34]) available at KITTI [1].

As shown in the table, we observe that our algorithm (NOMT) outperforms the other state-of-the-art methods

[2]Corresponding ROC curves are available at the supplemental material.
[3]Full analysis including detection measures and other methods is at [1].

	Method	MOTA ↑	MOTP ↑	MT ↑	ML ↓	IDS ↓	FRG ↓
Pedestrian Tracking Benchmark							
DP [30]	Batch	14.5 %	70.8 %	6.0 %	40.8 %	4,537	3,090
TBD [15]	Batch	15.9 %	70.9 %	6.4 %	47.9 %	1,939	1,963
RMOT [36]	Online	18.6 %	69.6 %	5.3 %	53.3 %	684	1,282
CEM [27]	Batch	19.3 %	70.7 %	8.5 %	46.5 %	813	1,023
HM	Online	26.7 %	71.5 %	11.2 %	47.9 %	669	916
NOMT	Online	**33.7** %	**71.9** %	**12.2** %	**44.0** %	**442**	**823**

Table 3. Multiple Target tracking accuracy for MOT Challenge.

in most of the metrics with significant margins. Our method produces much larger numbers of *mostly tracked targets* (MT) in both *Car* and *Pedestrian* experiments with smaller numbers of *mostly lost targets* (ML). This is thanks to the highly accurate identity maintenance capability of our algorithm demonstrated in the low number of *identity switch* (IDS) and *fragmentation* (FRG). In turn, our method achieves highest MOTA compared to other state-of-the-arts ($> 10\%$ for Car and $> 8\%$ for Pedestrian), which summarize all aspects of tracking evaluation. Our own HM baseline also performs better than the other methods, which demonstrates the robustness of ALFD metric. However, due to the nature of pure online association and lack of high order potential, it ends up missing more targets as shown in the MT and ML measures.

5.3. MOT Challenge Evaluation

Table. 3 summarizes the evaluation accuracy of our method (**NOMT**) and the other state-of-the-art algorithms on the MOT test video sequences[4]. The website provides a set of reference detections obtained using [9]. Similarly to the KITTI experiment, we observe that our algorithm outperforms the other methods with significant margins. Our method achieves the lowest *identity switch* and *fragmentation* while having more targets tracked (high MT and low ML). In turn, our method records the highest MOTA compared to the other state-of-the-arts with a significant margin ($> 14\%$). The two experiments demonstrate that our ALFD metric and NOMT algorithm is generally applicable to any application scenario. Fig. 4 shows some qualitative examples of our results.

5.4. Timing Analysis

In order to understand the timeliness of the NOMT method, we measure the latency by computing the difference between detection time (t_i of d_i in $\mathbb{A}^T$) and the last association time. The last association time is defined as: if a detection d_i is newly added to a target A_m^t or replace any other detection d_j (e.g. $t_i = t_j$) in A_m^{t-1} at t, t is recorded as the last association time for d_i. If d_i was in the A_m^{t-1}, no change is made to the last association time of d_i. The last association time tells us when the algorithm first recognizes the d_i as a part of A_m^T (the final trajectory output for the target m). The mean and standard deviation are 0.59 ± 1.75 and 0.66 ± 1.87 with [34] for the KITTI test set (84.7% and 83.9% with no latency) and 0.87 ± 2.04 for the MOT test set (77.6% with no latency). It shows that NOMT is indeed a near online method.

[4]Full analysis including detection measures and other methods is at [2].

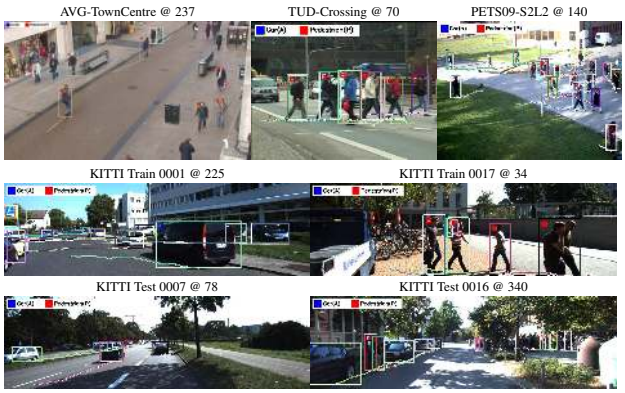

Figure 4. Qualitative examples of the tracking results. We show the bounding boxes together with the past trajectories (last 30 and 10 frames for MOT and KITTI, respectively). The color of the boxes and trajectories represents the identity of the targets. Notice that our method can generate long trajectories with consistent IDs in challenging situations, such as occlusion, fast motion, etc.

Dataset	FPS	IPT	CHist	Hypos	Infer	Total
KITTI (11,095)	10.27	644.2	238.8	236.0	15.6	1,080.2
KITTI+[34] (11,095)	10.15	615.6	161.5	144.9	40.3	1,092.5
MOT (5,783)	11.5	323.4	92.7	62.1	19.6	502.5

Table 4. Computation time on KITTI and MOT test datasets. The total number of images is shown in parentheses. We report the average FPS (images/total) and the time (seconds) spent in IPT computation (IPT), Color Histogram extraction (CHist), Hypothesis generation (Hypos) that includes all the potential computations, and the CRF inference (Infer). Total time includes file IO (reading images). The main bottleneck is the optical flow computation in IPT module, that can be readily improved using a GPU.

Our algorithm is not only highly accurate, but also very efficient. Leveraging on the parallel computation, we achieve a real-time efficiency ($\sim 10FPS$) using a 2.5GHz CPU with 16 cores. Table. 4 summarizes the time spent in each computational module.

6. Conclusion

In this paper, we propose a novel *Aggregated Local Flow Descriptor* that enables us to accurately measure the affinity between a pair of detections and a *Near Online Mutitarget Tracking* that takes the advantages of both the pure online and global tracking algorithms. Our controlled experiment demonstrates that ALFD based affinity metric is significantly better than other conventional affinity metrics. Equipped with ALFD, our NOMT algorithm generates significantly better tracking results on two challenging large-scale datasets. In addition, our method runs in real-time that enables us to apply it to various applications including autonomous driving, real-time surveillance, etc.

References

[1] KITTI tracking benchmark. http://www.cvlibs.

net/datasets/kitti/eval_tracking.php. Accessed: 2015-07-14. 7

[2] MOT challenge. http://nyx.ethz.ch/. Accessed: 2015-04-21. 1, 2, 7, 8

[3] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012. 2, 3, 6

[4] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *PAMI*, 2011. 1, 2, 3

[5] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008. 3, 4

[6] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009. 1, 2

[7] W. Choi, C. Pantofaru, and S. Savarese. A general framework for tracking multiple people from a moving camera. *Pattern Analysis and Machine Intelligence (PAMI)*, 2013. 1, 2

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 3

[9] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *PAMI*, 2014. 1, 8

[10] A. Ess, B. Leibe, K. Schindler, and L. van Gool. Robust multi-person tracking from a mobile platform. *PAMI*, 2009. 1, 2

[11] M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automated naming of characters in tv video. *Image and Vision Computing*, 27(5):545–559, 2009. 3

[12] G. Farneback. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 171–177. IEEE, 2001. 3, 4

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010. 1, 7

[14] K. Fragkiadaki, W. Zhang, G. Zhang, and J. Shi. Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In *ECCV (5)*, pages 552–565, 2012. 3

[15] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *PAMI*, 2014. 7, 8

[16] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1, 2, 7

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013. 1

[18] J. Hopcroft and R. Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Communications of the ACM*, 1973. 6

[19] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. *ECCV*, 2008. 2

[20] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2756–2759. IEEE, 2010. 3

[21] B. Keni and S. Rainer. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008. 7

[22] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *PAMI*, 2005. 1, 2

[23] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 6

[24] H. W. Kuhn. The hungarian method for the assignment problem. In *Naval Research Logistics Quarterly*, 1955. 7

[25] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, 2010. 1, 2, 3

[26] S. J. Leon. *Linear algebra with applications*. Macmillan New York, 1980. 5

[27] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *PAMI*, 2014. 1, 2, 3, 7, 8

[28] J. M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, Aug. 2010. Software available at https://staff.fnwi.uva.nl/j.m.mooij/libDAI/. 6

[29] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009. 2

[30] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. 2, 3, 7, 8

[31] A. Roshan Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. 2, 3, 6

[32] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006. 3

[33] C. Tomasi and T. Kanade. Detection and tracking of point features. In *Carnegie Mellon University Technical Report*, 1991. 3

[34] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, 2013. 1, 7, 8

[35] G. Welch and G. Bishop. An introduction to the kalman filter, 1995. 7

[36] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015. 7, 8

[37] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 1, 2