Learning Optimal Parameters For Multi-target Tracking

Shaofei Wang
shaofeiw@uci.edu
Charless Fowlkes
fowlkes@ics.uci.edu

Dept of Computer Science
University of California
Irvine, CA, USA

Abstract

We describe an end-to-end framework for learning parameters of min-cost flow multi-target tracking problem with quadratic trajectory interactions including suppression of overlapping tracks and contextual cues about co-occurrence of different objects. Our approach utilizes structured prediction with a tracking-specific loss function to learn the complete set of model parameters. Under our learning framework, we evaluate two different approaches to finding an optimal set of tracks under quadratic model objective based on an LP relaxation and a novel greedy extension to dynamic programming that handles pairwise interactions. We find the greedy algorithm achieves almost equivalent accuracy to the LP relaxation while being 2-7x faster than a commercial solver. We evaluate trained models on the challenging MOT and KITTI benchmarks. Surprisingly, we find that with proper parameter learning, our simple data-association model without explicit appearance/motion reasoning is able to outperform many state-of-the-art methods that use far more complex motion features and affinity metric learning.

1 Introduction

Thanks to advances of object detector performance, "tracking-by-detection" approaches that build tracks on top of a collection of candidate object detections have shown great promise. Tracking-by-detection avoids some classic difficulties such as drift, and is often able to recover from extended periods of occlusion since it is "self-initializing". Finding an optimal set of detections corresponding to each track can be formulated as a discrete optimization problem of locating low-cost paths through a graph of candidate detections for which there are often efficient combinatorial algorithms such as min-cost matching or min-cost network-flow (*e.g.*, [27, 36]). However, unlike generative formulations of multi-target tracking that estimate and score latent continuous trajectories for each object (*e.g.*, [2, 25, 32]), trajectories in tracking-by-detection approaches are implicitly defined by the selected set of detections. This immediately raises difficulties, both in (1) encoding strong trajectory models with only pairwise potentials and (2) identifying the parameters of these potentials from training data.

One approach to these issues is to first group detections into candidate tracklets and then perform scoring and association of these tracklets [5, 30, 33]. This allows tracklets to be scored with richer trajectory and appearance models. Another approach is to attempt to include higher-order constraints directly in a combinatorial framework [6, 7]. In either case,

Pages 4.1-4.13
DOI: https://dx.doi.org/10.5244/C.29.4

Figure 1: Our tracking framework incorporates quadratic interactions between objects in order to resolve appearance ambiguity and to boost weak detections. The parameters of the interactions are learned from training examples, allow the tracker to successfully learn mutual exclusion between cyclist and pedestrian, and boost to intra-class co-occurrence of nearby people.

there are a large number of parameters associated with these richer models which become increasingly difficult to set by hand and necessitate the application of machine learning techniques.

The contribution of this paper is in demonstrating that aggressively optimizing the parameters of relatively simple combinatorial models can yield state-of-the-art performance on difficult tracking benchmarks. We introduce a simple multi-target, multi-category tracking model that extends min-cost flow with quadratic interactions between tracks in order to capture contextual interactions within a frame. To perform inference, we propose a novel, greedy-dynamic programming algorithm that produces high-quality solutions on par with linear programming relaxations of the quadratic tracking objective while being substantially faster than a commercial LP solver. For learning, we use a structured prediction SVM [29] to optimize the complete set of tracking parameters from labeled data.

Structured prediction has been applied in tracking to learning inter-frame affinity metrics [17] and association [23] as well as a variety of other learning tasks such as fitting CRF parameters for segmentation [28] and word alignment for machine translation [18]. A related paper [8] also used a structured SVM to learn parameters for multi-target tracking with quadratic interactions for the purpose of activity recognition. Our work differs in that we choose a novel loss function that penalizes false transition and id-errors based on the MOTA tracking score. In particular, we show experimental results that demonstrate the learned models produce state-of-the-art performance on multi-target, multi-category tracking benchmarks.

2 Model

We begin by formulating multi-target tracking and data association as a min-cost network flow problem equivalent to that of [36], where individual tracks are described by a first-order Markov Model whose state space is spatial-temporal locations in videos. This framework incorporates a state transition likelihood that generates transition features in successive frames, and an observation likelihood that generates appearance features for objects and background.

Tracking by Min-cost Flow For a given video sequence, we consider a discrete set of candidate object detection sites V where each candidate site $x = (l, \sigma, t)$ is described by its

location, scale and frame number. We write $\Phi = \{\phi_a(x) | x \in V\}$ for the image evidence (appearance features) extracted at each corresponding spatial-temporal location in a video. A single object track consists of an ordered set of these detection sites: $T = \{x_1, ..., x_n\}$, each of which independently generates foreground object appearances at the corresponding sites according to distribution $p_{fg}(\phi_a)$ while the remaining site appearances are generated by a background distribution $p_{bg}(\phi_a)$.

The set of optimal (most probable) tracks can be found by solving an integer linear program (ILP) over flow variables $\mathbf{f}$.

$$\min_{\mathbf{f}} \sum_i c_i^s f_i^s + \sum_{ij \in E} c_{ij} f_{ij} + \sum_i c_i f_i + \sum_i c_i^t f_i^t \tag{1}$$

$$\text{s.t.} \quad f_i^s + \sum_j f_{ji} = f_i = f_i^t + \sum_j f_{ij} \tag{2}$$

$$f_i^s, f_i^t, f_i, f_{ij} \in \{0,1\} \tag{3}$$

where E is the set of valid transitions between sites in successive frames and the costs are given by

$$c_i = -\log \frac{p_{fg}(\phi_a(x_i))}{p_{bg}(\phi_a(x_i))}, \quad c_{ij} = -\log p_t(x_j|x_i), \quad c_i^s = -\log p_s(x_i), \quad c_i^t = -\log p_e(x_i) \tag{4}$$

p_s, p_e and p_t represent the likelihoods for tracks starting, ending and transitioning between given sites. This ILP is a well studied problem known as minimum-cost network flow [1]. The constraints satisfy the *total unimodularity* property and thus can be solved exactly using any LP solver or via various efficient specialized solvers, including network simplex, successive shortest path and push-relabel with bisectional search [36].

Track interdependence The aforementioned model assumes tracks are independent of each other, which is not always true in practice. In order to allow interactions between multiple objects, we add a pairwise cost term denoted q_{ij} and q_{ji} for jointly activating a pair of flows f_i and f_j corresponding to detections at sites $x_i = (l_i, \sigma_i, t_i)$ and $x_j = (l_j, \sigma_j, t_j)$. Adding this term to 1 yields an Integer Quadratic Program (IQP):

$$\min_{\mathbf{f}} \sum_i c_i^s f_i^s + \sum_{ij \in E} c_{ij} f_{ij} + \sum_i c_i f_i + \sum_{ij \in EC} q_{ij} f_i f_j + \sum_i c_i^t f_i^t \tag{5}$$

$$\text{s.t.} \quad (Eq.\ 2),\ (Eq.\ 3)$$

In our experiments, we only consider pairwise interactions between pairs of sites in the same video frame which we denote by $EC = \{ij : t_i = t_j\}$. One could easily extend such formulation to include transition-transition interaction to model high order dynamics.

The addition of quadratic terms makes this objective hard to solve in general. In the next section we discuss two different approximations for finding high quality solutions $\mathbf{f}$. In Section 5 we describe how the costs $\mathbf{c}$ can be learned from data.

3 Inference

Unlike min-cost flow (Eq. 1), finding the global minimum of the IQP problem (Eq. 5) is NP-hard [35] due to the quadratic terms. We evaluate two different schemes for finding high-quality approximate solutions. The first is a standard approach of introducing auxiliary

variables and relaxing the integral constraints to yield a linear program (LP) that lower-bounds the original objective. We also consider a greedy approximation based on successive rounds of dynamic programming that also yields good solutions while avoiding the expense of solving a large scale LP. The resulting tracks (encoded by the optimal flows $\mathbf{f}$) are used for both test-time track prediction as well as for optimizing parameters during learning (see Section 5).

LP Relaxation and Rounding If we relax the integer constraints and deform the costs as necessary to make the objective convex, then the global optimum of 5 can be found in polynomial time. For example, one could apply Frank-Wolfe algorithm to optimize the relaxed, convexified QP while simultaneously keeping track of good integer solutions [16]. However, for real-world tracking over long videos, the relaxed QP is still quite expensive. Instead we follow the approach proposed by Chari *et al*. [7], reformulating the IQP as an equivalent ILP problem by replacing the quadratic terms $f_i f_j$ with a set of auxiliary variables u_{ij}:

$$\begin{aligned} &\min_{\mathbf{f}} \sum_i c_i^s f_i^s + \sum_{ij\in E} c_{ij} f_{ij} + \sum_i c_i f_i + \sum_{ij\in EC} q_{ij} u_{ij} + \sum_i c_i^t f_i^t \\ &\text{s.t.} \quad (Eq.\ 2),\ (Eq.\ 3), \quad u_{ij} \le f_i, u_{ij} \le f_j \quad f_i + f_j \le u_{ij} + 1 \end{aligned} \tag{6}$$

The new constraint sets enforce u_{ij} to be 1 only when f_i and f_j are both 1. By relaxing the integer constraints, program 6 can be solved efficiently via large scale LP solvers such as CPLEX or MOSEK. We use two different rounding heuristics proposed by [7] to produce integral solutions and take the solution with lower cost.

Greedy Sequential Search As an alternative to the LP relaxation, we describe a simple greedy algorithm inspired by the combination of dynamic programming (DP) and non-maximal suppression proposed in [27]. The detailed algorithm is described in Algorithm 1. The algorithm sequentially instances tracks by finding shortest paths through a min-cost flow graph and then updates estimated costs for future tracks to include the quadratic penalties incurred by the instanced track.

Algorithm 1 Greedy Sequential Search (DP with Quadratic Cost Update)

Input: A Directed-Acyclic-Graph G with edge weights c_i, c_{ij}
initialize $\mathcal{T} \leftarrow \emptyset$
repeat
  Find shortest start-to-end path p on G via dynamic programming
  $track\_cost = cost(p)$
  if $track\_cost < 0$ **then**
    for all locations x_i in p **do**
      $c_j = c_j + q_{ij} + q_{ji}$ for all $ij, ji \in EC$
      $c_i = +\infty$
    end for
    $\mathcal{T} \leftarrow \mathcal{T} \cup p$
  end if
until $track\_cost \ge 0$
Output: track collection $\mathcal{T}$

In the absence of quadratic terms, this algorithm corresponds to the 1-pass DP approximation of the successive-shortest paths (SSP) algorithm. Hence it does not guarantee an

optimal solution, but, as we show in the experiments, it performs well in practice. An implementation difference from a pure dynamic programming approach such as [27], is that updating costs with the quadratic terms when a track is instanced has the (unfortunate) effect of invalidating cost-to-go estimates which could otherwise be cached and re-used between successive rounds of dynamic programming.

4 Features for Tracking Potential Functions

In order to learn the tracking potentials ($\mathbf{c}$ and $\mathbf{q}$) we parameterize the flow cost objective by a vector of weights $\mathbf{w}$ and a set of features $\Psi(X,\mathbf{f})$ that depend on features extracted from the video, the spatio-temporal relations between candidate detections, and which tracks are instanced. With this linear parameterization we write the cost of a given flow as $C(\mathbf{f}) = \mathbf{w}^T\Psi(X,\mathbf{f})$ where the vector components of the weight and feature vector are given by:

$$\mathbf{w} = \begin{bmatrix} w_S \\ w_t \\ w_s \\ w_a \\ w_E \end{bmatrix} \qquad \Psi(X,\mathbf{f}) = \begin{bmatrix} \sum_i \phi_S(x_i^s) f_i^s \\ \sum_{ij\in E} \psi_t(x_i,x_j) f_{ij} \\ \sum_{ij\in EC} \psi_s(x_i,x_j) f_i f_j \\ \sum_i \phi_a(x_i) f_i \\ \sum_i \phi_E(x_i^t) f_i^t \end{bmatrix} \tag{7}$$

Here w_a represents local appearance template for the tracked objects of interest, w_t represents weights for transition features, w_s represents weights for pairwise interactions, w_S and w_E represents weights associated with track births and deaths. $\Psi(X,\mathbf{f})$ are corresponding features, which are described as below:

Local appearance and birth/death model We make use of off-the-shelf detectors [10, 11, 31] to capture local appearance. Our local appearance feature thus consists of the detector score along with a constant 1 to allow for a variable bias. In applications with static cameras it can be useful to learn a spatially varying bias to model where tracks are likely to appear or disappear. However, most videos in our experiments are captured from moving platforms, we thus use a single constant value 1 for the birth and death features.

Transition model We connect a candidate x_i at time t_i with another candidate x_j at a later time t_i+n, only if the overlap ratio between x_i's window and x_j's window exceeds 0.3. The overlap ratio is defined as two windows' intersection over their union. We use this overlap ratio as a feature associated with each transition link. The transition link's feature will be 1 if this ratio is lower than 0.5, and 0 otherwise. In our experiments, we allow up to 7 frames occlusion for all the network-flow methods. We append a constant 1 to this feature and bin these features according to the length of transition. This yields a 16 dimensional feature for each transition link.

Pairwise interactions w_s is a weight vector that encodes valid geometric configurations of two objects. $\psi_s(x_i,x_j)$ is a discretized spatial-context feature that bins relative location of detection window at location x_i and window at location x_j into one of the D relations including on top of, above, below, next-to, near, far and overlap (similar to the spatial context of [9]). To mimic the temporal NMS described in [27] we add one additional relation, strictly overlap, which is defined as the intersection of two boxes over the area of the first box; we set the corresponding feature to 1 if this ratio is greater than 0.9 and 0 otherwise. Now assume that we have K classes of objects in the video, then w_s is a DK^2 vector,

$w_s = [w_{s11}^T, w_{s12}^T, ..., w_{sij}^T, ..., w_{sKK}^T]^T$, in which w_{sij} is a length of D column vector that encodes valid geometric configurations of object of class i w.r.t. object of class j. In such way we can capture intra- and inter-class contextual relationships between tracks.

5 Learning

We formulate parameter learning of tracking models as a structured prediction problem. With some abuse of notation, assume we have N training videos $(X_n, \mathbf{f}_n) \in \mathcal{X} \times \mathcal{F}, n = 1, ..., N$. Given ground-truth tracks in training videos specified by flow variables $\mathbf{f}_n$, we discriminatively learn tracking model parameters $\mathbf{w}$ using a structured SVM with margin rescaling:

$$\mathbf{w}^* = \underset{\mathbf{w}, \xi_n \geq 0}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_n \xi_n \tag{8}$$
$$\text{s.t. } \mathbf{w}^T\Psi(X_n, \widehat{\mathbf{f}}) - \mathbf{w}^T\Psi(X_n, \mathbf{f}_n) \geq L(\mathbf{f}_n, \widehat{\mathbf{f}}) - \xi_n \quad \forall n, \widehat{\mathbf{f}}$$

where $\Psi(X_n, \mathbf{f}_n)$ are the features extracted from nth training video. $L(\mathbf{f}_n, \widehat{\mathbf{f}})$ is a loss function that penalizes any difference between the inferred label $\widehat{\mathbf{f}}$ and the ground truth label $\mathbf{f}_n$ and which satisfies $L(\mathbf{f}_n, \mathbf{f}_n) = 0$. The constraint on the slack variables ξ_n ensure that we pay a cost for any training videos in which the cost of the flow associated with ground-truth tracks under model $\mathbf{w}$ is higher than some other incorrect flow $\widehat{\mathbf{f}}$. We use a standard cutting plane approach [15] to optimize the objective by performing loss-augmented inference to find flows $\widehat{\mathbf{f}}$ that violate the constraint. We note that this formulation allows for constraints corresponding to non-integral flows $\widehat{\mathbf{f}}$ so we can directly use the LP relaxation (Eq. 6) to generate violated constraints during training. [12] points out that besides optimality guarantees, including non-integral constraints naturally pushes the SVM optimization towards solutions that produce integer solution even before rounding.

Tracking Loss Function We find that a critical aspect for successful learning is to use a loss function that closely resembles major tracking performance criteria, such as Multiple Object Tracking Accuracy (MOTA [4]). Metrics such as false positive, false negative, true positive, true negative and true/false birth/death can be easily incorporated using a standard Hamming loss on the flow vector. However, id switches and fragmentations [21] are determined by looking at labels of two consecutive transition links simultaneously, and hence cannot be optimized by our inference routine which only considers pairwise relations between detections within a frame. Instead, we propose a decomposable loss for transition links that attempts to capture important aspects of MOTA by taking into account the length and localization of transition links rather than using a constant (Hamming) loss on mislabeled links.

We define a weighted Hamming loss to measure distance between ground-truth tracks $\mathbf{f}$ and inferred tracks $\widehat{\mathbf{f}}$ that includes detections/birth/death, f_i, and transitions, f_{ij}. Let

$$L(\widehat{\mathbf{f}}, \mathbf{f}) = \sum_i l_i \left| f_i - \widehat{f_i} \right| + \sum_{ij} l_{ij} \left| f_{ij} - \widehat{f_{ij}} \right|$$

where $\mathbf{l} = \{l_1, ..., l_i, ..., l_{ij}, ..., l_{|\mathbf{f}|}\}$ is a vector indicating the penalty for differences between the estimated flow $\widehat{\mathbf{f}}$ and the ground-truth $\mathbf{f}$.

In order to describe our transition loss, let us first denote four types of transition links: *NN* is the link from a false detection to another false detection, *PN* is the link from a true

detection to a false detection, NP is the link from a false detection to a true detection, PP^+ is the link from a true detection to another true detection with the same identity, and PP^- is the link from a true detection to another true detection with a different identity. For all the transition links, we interpolate detections between its start detection and end detection (if their frame numbers differ more than 1); the interpolated virtual detections are considered either true virtual detection or false virtual detection, depending on whether they overlap with a ground truth label or not. Loss for different types of transition is defined as:

1. For NN links, the loss will be (number of true virtual detections + number of false virtual detections)
2. For PN and NP links, the loss will be (number of true virtual detections + number of false virtual detections + 1)
3. For PP^+ links, the loss will be (number of true virtual detections)
4. For PP^- links, the loss will be (number of true virtual detections + number of false virtual detections + 2)

Training data Available training datasets specify ground-truth bounding boxes that need to be mapped onto ground-truth flow variables $\mathbf{f_n}$ for each video. To do this mapping, we first consider each frame separately. We take the highest scoring detection window that overlaps a ground truth label as true detection and assigned it a track identity label which is the same as the ground truth label it overlaps. Next, for each track identity, we run a simplified version of the dynamic programming algorithm to find the path that claims the largest number of true detections. After we iterate through all id labels, any instanced graph edge will be a true detection/transition/birth/death while the remainder will be false.

An additional difficulty of training which arises on the KITTI tracking benchmark are special evaluation rules for ground truth labels such as small/truncated objects and vans for cars, sitting persons for pedestrians. This is resolved in our training procedure by removing all detection candidates that correspond to any of these "ambiguous" ground truth labels during training; in this way we avoid mining hard negatives from those labels. To speed up training on both MOT and KITTI dataset, we partition full-sized training sequences in to 10-frame-long subsequences with a 5-frame overlap, and define losses on each subsequence separately.

6 Experimental results

We have focused our experiments on two challenging datasets: the Multiple Object Tracking Benchmark[1] [19], which focuses primarily on pedestrian tracking; and the KITTI Tracking Benchmark[2] [13], which involves multi-category tracking of cars, pedestrians and cyclists. For both datasets we allow occlusion of up to 7 frames in our tracking graph. Best regularization parameters are obtained via leave-one-video-out cross-validation on training data. All results on test set were submitted to the respective test servers *only once*.

The MOT Benchmark For the MOT Benchmark, we only use a subset of contextual features that includes the overlap and near relationships due to the varying view angle of benchmark videos. Surprisingly, we achieve the best tracking results on MOTA among all published

[1] http://nyx.ethz.ch/
[2] http://www.cvlibs.net/datasets/kitti/eval_tracking.php

MOT dataset

Benchmark on test set							Cross-validation on training set						
Method	MOTA	MOTP	MT	ML	IDSW	FRAG	Method	MOTA	MOTP	MT	ML	IDSW	FRAG
TC_ODAL [8]	15.1%	70.5%	3.2%	55.8%	**637**	1716	SSP	**28.7%**	**72.9%**	15.1%	50.5%	440	**541**
RMOT [30]	18.6%	69.6%	5.3%	53.3%	684	1282	LP+Hamming	25.3%	72.4%	**17.4%**	**46.5%**	567	604
CEM [23]	19.3%	70.7%	**8.5%**	**46.5%**	813	1023	LP	28.5%	72.8%	15.1%	48.9%	**440**	563
SegTrack [24]	22.5%	71.7%	5.8%	63.9%	697	**737**	DP	27.6%	72.4%	15.5%	49.1%	492	626
MotiCon [20]	23.1%	70.9%	4.7%	52.0%	1018	1061							
Ours(LP)	**25.2%**	**71.7%**	5.8%	53.0%	646	849							

Table 1: Benchmark and cross-validation results on MOT dataset. We denote variants of our model as follows: 1) SSP are models without pairwise terms, learned and tested with successive shortest path algorithm. 2) LP are models with pairwise terms, learned with LP-Relaxation while tested with LP-Rounding. 3) DP are models with pairwise terms, learned with LP-Relaxation while tested with Greedy Sequential Search. 4) LP+Hamming is the same as LP, except that models are learned using Hamming loss instead of the loss described in Section 5.

results, without employing any explicit appearance/motion model. We expect this is not because appearance/motion features are useless but rather that the parameters in these features have not been optimally learned/integrated into in competing tracking methods.

The KITTI Tracking Benchmark Due to the high speed motion of vehicle platforms, for the KITTI dataset we use pre-computed frame-wise optical flow [22] to predict candidates' locations in future frames in order to generate candidate links between frames. We evaluated two different detectors, DPM and the regionlets detector [31] which produced the best result in terms of MOTA, IDs and FRAG during cross-validation. Result on test set is summarized in the upper part of Table 2.

Diagnostic Analysis We conduct cross-validations on the training set for each dataset to study the effect of quadratic terms, loss function and inference algorithm. The results are summarized in Table 1 and 2. As shown in right side of Table 1, our novel loss function is superior to traditional Hamming loss in terms of maximizing MOTA. The DP algorithm proposed in section 3 achieves 2-7x speedup (Fig 2) with negligible loss in most metrics; it is even slightly better than LP inference when tracking cars with DPM detector (Table 2).

We found SSP (min-cost flow without quadratic terms) achieves slightly better overall accuracy on the MOT dataset. MOT only contains a single object category and includes videos from many different viewpoints (surveillance, vehicle, street level) which limits the potential benefits of simple 2D context features. However, by properly learning the detector confidence and transition smoothness in the SSP model, many false tracks can be pruned even without contextual knowledge.

For traditional multi-category detector such as DPM [11], quadratic interactions were very helpful to improve the tracking performance on KITTI; this is most evident for tracking cyclist, as shown in Table 2. However, this benefit seems to disappear when we switch to the more powerful regionlets-based detector where the SSP approach (min-cost flow without quadratic terms) achieves best performance by a noticeable margin. We conclude that when the detector itself is good enough to resolve ambiguity caused by similar appearance or limited resolution, the benefits of quadratic terms are outweighed by the difficulties of approximate inference. Unlike the LP relaxation or greedy sequential search, SSP always produces a globally optimal set of tracks which appears to benefit parameter learning.

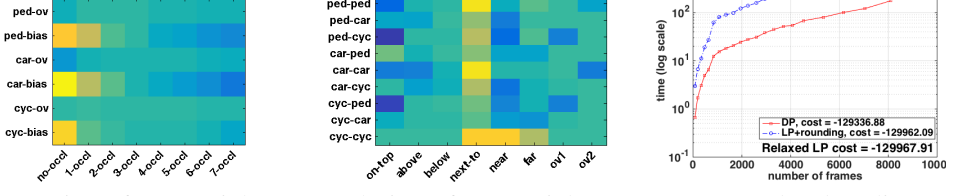
Benchmark on KITTI test set

Benchmark on Car, DPM detections							Benchmark on Pedestrian, DPM detections						
Method	MOTA	MOTP	MT	ML	IDSW	FRAG	Method	MOTA	MOTP	MT	ML	IDSW	FRAG
TBD [ICCV09]	51.7%	**78.5%**	13.8%	34.6%	33	540	TBD	NA	NA	NA	NA	NA	NA
CEM	47.8%	77.3%	14.4%	34.0%	125	401	CEM	36.2%	**74.6%**	8.0%	53.0%	221	**1011**
RMOT	49.3%	75.3%	15.2%	33.5%	51	**389**	RMOT	39.9%	72.9%	10.0%	47.5%	132	1081
Ours(LP)	**57.3%**	77.2%	**27.9%**	**23.4%**	**18**	449	Ours(LP)	**43.7%**	74.1%	**10.4%**	**43.4%**	**87**	1291
Benchmark on Car, Regionlet detections							Benchmark on Pedestrian, Regionlet detections						
RMOT	60.3%	75.6%	27.0%	11.4%	216	755	RMOT	51.1%	74.2%	16.9%	41.3%	372	1515
Ours(SSP)	**70.4%**	**77.7%**	**41.6%**	**9.4%**	**73**	**579**	Ours(SSP)	**56.5%**	**75.4%**	**18.7%**	**33.7%**	**181**	**1448**

Cross-validation result on KITTI training set

Benchmark on Car, DPM detections							Benchmark on Car, Regionlet detections						
Method	MOTA	MOTP	MT	ML	IDSW	FRAG	Method	MOTA	MOTP	MT	ML	IDSW	FRAG
SSP	63.4%	**78.3%**	27.4%	20.0%	**2**	**179**	SSP	**78.9%**	**80.8%**	**45.0%**	**8.1%**	**22**	**308**
LP	64.5%	78.1%	30.1%	18.9%	4	206	LP	77.4%	80.2%	44.1%	8.1%	152	515
DP	**65.1%**	78.0%	**30.3%**	**18.7%**	16	224	DP	76.2%	80.2%	42.7%	8.4%	163	517
Benchmark on Pedestrian, DPM detections							Benchmark on Pedestrian, Regionlet detections						
Method	MOTA	MOTP	MT	ML	IDSW	FRAG	Method	MOTA	MOTP	MT	ML	IDSW	FRAG
SSP	51.2%	**73.2%**	19.2%	24.6%	**16**	**230**	SSP	**73.0%**	**76.6%**	**58.1%**	**7.8%**	**67**	**369**
LP	**53.0%**	72.9%	**22.8%**	21.0%	24	263	LP	69.5%	76.1%	51.5%	8.4%	130	493
DP	52.7%	73.0%	19.2%	**21.0%**	30	269	DP	67.8%	76.2%	55.7%	7.8%	138	470
Benchmark on Cyclist, DPM detections							Benchmark on Cyclist, Regionlet detections						
Method	MOTA	MOTP	MT	ML	IDSW	FRAG	Method	MOTA	MOTP	MT	ML	IDSW	FRAG
SSP	47.4%	**79.7%**	35.1%	32.4%	**5**	**10**	SSP	**83.7%**	**82.5%**	**75.7%**	**2.7%**	**8**	**18**
LP	**58.1%**	79.5%	40.5%	**29.7%**	8	15	LP	79.9%	81.9%	73.0%	2.7%	8	28
DP	57.8%	79.5%	**40.5%**	32.4%	7	13	DP	78.1%	81.9%	70.3%	5.4%	15	34

Table 2: Benchmark and cross-validation results on KITTI data set. We evaluate two different detectors (DPM and Regionlet) and three different inference models (SSP,LP,DP) each trained using SSVM.



(a) inter-frame weights (b) intra-frame weights (c) speed and quality

Figure 2: Visualization of the weight vector learned on KITTI dataset for the DPM detector. Yellow has small cost, blue has large cost. (a) shows transition weights for different length of frame jumps. (b) shows learned pairwise contextual weights between objects. The model encourages intra-class co-occurrence when objects are close but penalizes overlap and objects on top of others. Note the strong negative interaction learned between cyclist and pedestrian (two classes which are easily confused by their respective detectors.). (c) Speed and quality comparison of proposed DP and traditional LP approximation. Over the 21 training sequences in KITTI dataset, LP+rounding produces solutions with an upper-bound cost that is very close to relaxed global optimum. Sequential DP with quadratic cost estimates gives an upper-bound that is within 1% of relaxed global optimum, while being 2 to 7 times faster than a commercial LP solver (MOSEK). Figure best be viewed in color.

7 Conclusion and Future Work

In summary, our underlying tracking model is a rather straight-forward extension of previously published approaches [27, 36], yet it is able to outperform many far more complex state-of-the-art methods on both MOT and KITTI benchmarks. However, we note that simple application of the DP-based tracker described in [27] does quite poorly on these datasets (*e.g.*, MOTA=14.9 on the MOT benchmark). We thus attribute the performance boost to our learning framework which produce much better parameters than those estimated by hand-tuning or piece-wise model training.

We want to stress that our work is also complimentary to other existing methods. While we did not see significant benefits to adding simple appearance-based affinity features (e.g., RGB histogram or HOG) to our model, many state-of-the-art systems perform hierarchical or streaming data-association which involves collecting examples from extended period of trajectory to train target specific appearance models in an online fashion. Such models can potentially fit into our framework, providing a way to explore more complicated affinity features while estimating hyper-parameters automatically from data. One could also introduce richer, trajectory level contextual features under such a hierarchical learning framework.

8 Acknowledgement

This work was supported by the US National Science Foundation through awards IIS-1253538 and DBI-1053036

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-617549-X.

[2] Anton Andriyenko, Konrad Schindler, and Stefan Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012.

[3] Seung-Hwan Bae and Kuk-Jin Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *J. Image Video Process.*, 2008:1:1–1:10, January 2008. ISSN 1687-5176.

[5] William Brendel, Mohamed Amer, and Sinisa Todorovic. Multiobject tracking as maximum weight independent set. In *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[6] Asad A. Butt and Robert T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[7] Visesh Chari, Simon Lacoste-Julien, Ivan Laptev, and Josef Sivic. On pairwise cost for multi-object network flow tracking. *CoRR*, abs/1408.3304, 2014.

[8] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*, 2012.

[9] Chaitanya Desai, Deva Ramanan, and Charless Fowlkes. Discriminative models for multi-class object layout. In *IEEE International Conference on Computer Vision*, 2009.

[10] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *PAMI*, 2014.

[11] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/.

[12] T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *International Conference on Machine Learning (ICML)*, pages 304–311, 2008.

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[14] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3d traffic scene understanding from movable platforms. *Pattern Analysis and Machine Intelligence (PAMI)*, 2014.

[15] T. Joachims, T. Finley, and Chun-Nam Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

[16] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *European Conference on Computer Vision (ECCV)*, 2014.

[17] Suna Kim, Suha Kwak, Jan Feyereisl, and Bohyung Han. Online multi-target tracking by large margin structured learning. In *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part III*, ACCV'12, pages 98–111, Berlin, Heidelberg, 2013. Springer-Verlag.

[18] Simon Lacoste-Julien, Ben Taskar, Dan Klein, and Michael I. Jordan. Word alignment via quadratic assignment. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 112–119, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

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

[20] Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an image-based motion context for multiple people tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[21] Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *In CVPR*, 2009.

[22] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[23] Xinghua Lou and Fred A. Hamprecht. Structured Learning for Cell Tracking. In *Twenty-Fifth Annual Conference on Neural Information Processing Systems (NIPS 2011)*, 2011.

[24] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *IEEE TPAMI*, 36(1):58–72, 2014. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.103.

[25] Anton Milan, Konrad Schindler, and Stefan Roth. Detection- and trajectory-level exclusion in multiple object tracking. In *CVPR*, 2013.

[26] Anton Milan, Laura Leal-Taixé, Konrad Schindler, and Ian Reid. Joint tracking and segmentation of multiple targets. In *CVPR*, 2015.

[27] Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[28] Martin Szummer, Pushmeet Kohli, and Derek Hoiem. Learning crfs using graph cuts. In *European Conference on Computer Vision*, October 2008.

[29] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. MIT Press, 2003.

[30] Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang. Tracklet association with online target-specific metric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[31] Xiaoyu Wang, Ming Yang, Shenghuo Zhu, and Yuanqing Lin. Regionlets for generic object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[32] Z. Wu, A. Thangali, S. Sclaroff, , and M. Betke. Coupling detection and data association for multiple object tracking. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Rhode Island, June 2012.

[33] Bo Yang and Ram Nevatia. An online learned crf model for multi-target tracking. In *In CVPR*, 2012.

[34] Ju Hong Yoon, Ming-Hsuan Yang, Jongwoo Lim, and Kuk-Jin Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2014, Waikoloa, HI, USA, January 5-9, 2015*, pages 33–40, 2015.

[35] Abdel Nasser H. Zaied and Laila Abd El fatah Shawky. Article: A survey of quadratic assignment problems. *International Journal of Computer Applications*, 101(6):28–36, September 2014.

[36] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 0:1–8, 2008.