Learning to Track Multi-Target Online by Boosting and Scene Layout

Guang Chen, Feihu Zhang, Daniel Clarke, Alois Knoll

***Abstract*— We address two principal difficulties of multi-target tracking in a real traffic scenario. Firstly, fast moving traffic scenarios lead to large displacements and complex interactions with occlusions and ambiguities. Secondly, the tracking application for real traffic scenarios has the online requirement. To surmount these difficulties, we propose an approach to track the multi-target online by Boosting and scene context reasoning. To this end, we use a two-stage system, where the first stage learns a non-linear classifier which is capable of generating the observation similarities. In the second stage, we demonstrate a novel relationship between observations and the scene layout parameters. Using a probabilistic formulation and the above relationship, our method has the unique ability to handle exceptions. To evaluate our method, we create three real traffic data sets, covering urban, rural, and highway conditions. We hope that these datasets will push forward the performance of tracking systems when being moved outside the laboratory to the real world.**

I. Introduction

Tracking objects is important for many computer vision applications. This is an easy task when the objects are isolated and easily distinguished from the background. However, in real world scenes, the following often result in incorrect trajectories: strong occlusions, illumination changes, driving at a high speed, similar object appearance and false observations often result in incorrect trajectories.

Recent studies focus on tracking-by-detection methods as a result of significant improvement in object detection algorithms [6], [5], [18]. These techniques often deal with the imperfect detections by global optimization such as integer linear programming [12], [1], and finding min-cost flow on the flow network model [25], [19]. Robust tracking has been achieved by hierarchical tracklet association based on the Munkres algorithm [11], [17], conditional random fields [23], maximum weight independent set [3]. However, although there has been significant improvement in tracking performance, many of these methods are limited to offline processing as they consider future information for optimization. Although a few techniques perform multi-target tracking online, most of these methods are developed under a controlled environment (i.e. stationary camera or slowly moving) [13], [24], [22].

The key factor to guarantee a successful and robust tracking method is the association cost. For this reason, various affinity measures between a pair of observations have been studied in the past. Distances between the descriptors of the detections are often used as the affinity measurements [16], [11], [17], [25], [12]. For example, Mahalanobis distance between descriptors is defined as the affinity updated by online learning metrics during tracking. Affinity is often determined by learning the discrimination among targets online [23], [15], or jointly solving classification and ranking of associations [17]. However, the definition of this kind of affinity may not be sufficiently rigorous especially when multiple features are incorporated and the dependencies between each other are difficult to be predicted properly.

To this end, we provide an online algorithm to track multiple objects through data association based on Boosting and scene layout without the construction of tracklets (See Fig. 1). The data association is achieved by the predicted structure in the Boosting framework [7]. Furthermore, we claim that jointly tracking objects and reasoning scene context are critical. The idea that recognition and reconstruction are mutually beneficial has been investigated in [9], [10], [2]. We start from the intuition that, in a traffic scenario, an object's location and pose are not arbitrarily distributed but rather constrained by the fact that objects must lie on the ground. We formulate the problem of joint object tracking and scene reconstruction as a novel relationship between tracking objects and the scene layout parameters. Using a novel probabilistic formulation our method has the ability to to handle the exceptions and achieve robust tracking performance.

Recently, [8] found that algorithms such as stereo matching and object detections ranking high on existing benchmarks often failed when confronted with more realistic scenarios. To evaluate our method, we create three challenging sequences, named RURALseq, URBANseq and HIGHWAYseq. Most of current datasets are simplistic, e.g. ETH [4] is captured by cameras installed on a slowly moving chariot. Ours aim at real traffic applications.

The structure of the paper is as follows. In Section II we decribe how our multi-target tracking algorithm is formulated by boosting. Section III describes the exceptions handling by our scene layout model. The data sets for the real traffic scenarios are described in Section IV. Results for the these data sets are presented in Section V, before the paper is concluded in Section VI.

II. Multi-Target tracking by Boosting

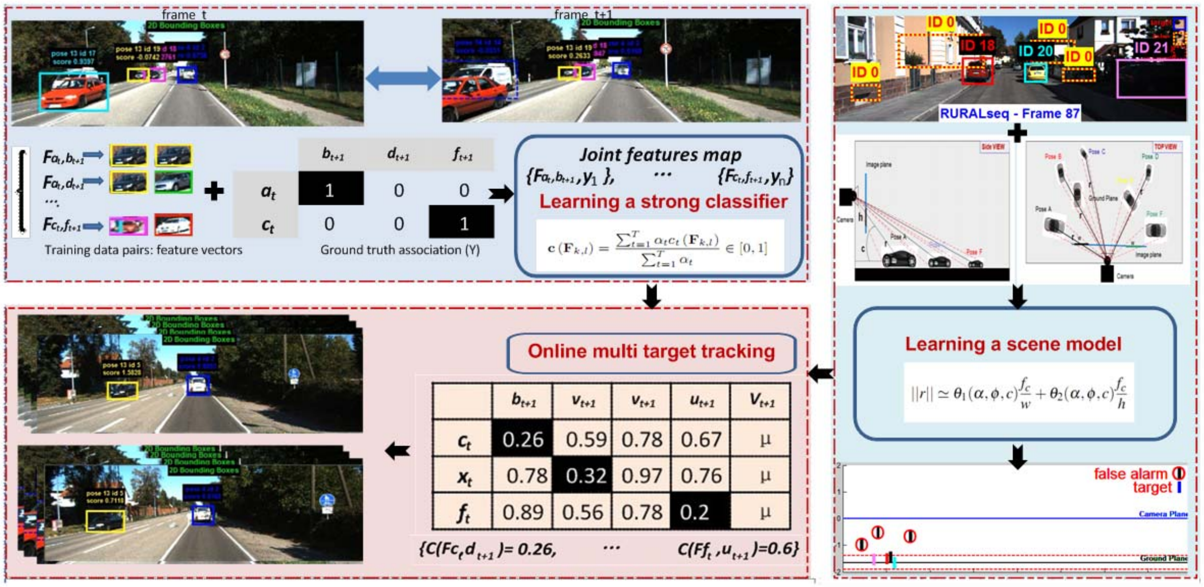
Let $X_k = \{x_1^k, ..., x_{n_k}^k\}$ be a set of detected objects of interest at frame k. In the *golablly-optimal* data association, an association hypothesis $\mathcal{T}$ is defined as a set of single tracking hypotheses,*i.e.* $\mathcal{T} = \{T_k\}$. The objective of data association is to maximize the posteriori probability of $\mathcal{T}$

Guang Chen, Feihu Zhang, and Alois Knoll are with the Technische Universität München, Garching bei München, Germany, e-mail: {guang,zhangf,knoll}@in.tum.de.

Daniel Clarke is with the fortiss GmbH, München, e-mail: daniell@fortiss.org.

978-0-7695-5144-9/13 $31.00 © 2013 IEEE
DOI 10.1109/ICMLA.2013.41

IEEE computer society


frame t
frame t+1
2D Bounding Boxes
pose 13 id 17
score 0.9397
pose 13 id 19
score -0.0742
pose 13 id 19
score 0.2633
Fa_t,b_{t+1}
Fa_t,d_{t+1}
…,
Fc_t,f_{t+1}
Training data pairs: feature vectors
Ground truth association (Y)
Joint features map
$\{Fa_t,b_{t+1},y_1\}$, … $\{Fc_t,f_{t+1},y_n\}$
Learning a strong classifier
$\mathbf{c}(\mathbf{F}_{k,l}) = \frac{\sum_{t=1}^{T} \alpha_t c_t(\mathbf{F}_{k,l})}{\sum_{t=1}^{T} \alpha_t} \in [0,1]$
Online multi target tracking
pose 13 id 5
score 1.5828
pose 13 id 5
score 0.7118
$\{C(Fc_t d_{t+1}) = 0.26,$ … $C(Ff_t, u_{t+1}) = 0.6\}$
ID 0
ID 18
ID 20
ID 21
RURALseq - Frame 87
Side VIEW
TOP VIEW
Image plane
Camera
h
c
r
Pose A
Pose B
Pose C
Pose D
Pose F
Ground Plane
Learning a scene model
$||r|| \simeq \theta_1(\alpha,\phi,c)\frac{f_c}{w} + \theta_2(\alpha,\phi,c)\frac{f_c}{h}$
false alarm
target
Camera Plane
Ground Plane
2
1
0
-1
-2

	b_{t+1}	d_{t+1}	f_{t+1}
a_t	1	0	0
c_t	0	0	1

	b_{t+1}	v_{t+1}	v_{t+1}	u_{t+1}	V_{t+1}
c_t	0.26	0.59	0.78	0.67	µ
x_t	0.78	0.32	0.97	0.76	µ
f_t	0.89	0.56	0.78	0.2	µ

Fig. 1. Overview of our multi target tracking system by boosting and scene layout. Firstly, we learn the strong classifier and predict the matching structure. Secondly, we learn the scene model to handle the exceptions. We can efficiently reject the false observations and model the long-term occlusion/coexistent occlusions.

given the detection set $\mathcal{X} = \{X_k\}$. This kind of data association is able to find the global optimum efficiently but is limited to *offline* batch processing. To achieve the online requirement, we present how the Boosting formulates the data association problem for multi-target tracking.

A. Features representation

We compute a feature vector Ψ_{ij}^{kk+1} between any pairs of detections x_i^k and x_j^{k+1} from a pair of adjacent frames k and $k+1$. To learn the structure of data association between two adjacent frames, we assign each feature vector Ψ_{ij}^{kk+1} a corresponding class label f_{ij}^{kk+1}. Here, we have two classes, called positive and negative, or p and n, respectively. The positive class consists of detection pairs from the same target, the negative class consists of detection pairs from different targets. For each pair of adjacent frame k and frame $k+1$, we define an augmented feature map F_{kk+1}, which is a collection of the feature vectors and class labels between all pairs of detections.

To learn a strong classifier using boosting, κ hand-labeled feature maps from the training data are provided. Let N_p and N_n be the number of training feature vectors belonging to p and n, respectively, We write Φ as a combination of feature vectors and class labels from κ hand-labeled feature maps,

$$\begin{aligned} &\Phi = \{(\Psi_1,f_1),\ldots,(\Psi_k,f_k),\ldots,(\Psi_N,f_N)\} \\ &N = N_p + N_n \end{aligned} \tag{1}$$

B. Association using Boosting

Our data association algorithm uses the pairwise comparison of detections. All dimensions of the feature vector Ψ_{ij}^{kk+1} are combined in a non-linear manner using a boosting classifier. There are two main parts of the algorithm, the first part is the learning phase where a classifier learns from the training data. The second part is the classification phase, where the learned classifier is used to classify feature vectors in multi-target tracking experiments.

The learning phase of boosting is an iterative procedure that consecutively adds weak classifiers to a set of previously added weak classifiers. The weak classifiers used here are decision stumps. In each iteration, the weak classifier that minimizes the weighted classification error is chosen. Given the parameters of the best weak classifier, the training data is classified and the weights of the correctly classified data are decreased.This procedure is repeated until *T* weak classifiers have been computed. Weak classifiers can be added several times in each dimension of Ψ_i. The weighted combination of *T* weak classifier together create the strong classifier. The closer the strong classifier $C(\Psi)$ is to zero, the higher the likelihood of the feature vector Ψ belonging to positive class is. A detailed presentation of Boosting for data association is given in Algorithm 1.

C. Optimization

During prediction, within the strong classifier it is easy to transform the feature map F_{kk+1} to the probability map P_{kk+1}. Let C_{ij} and $\hat{f}_{ij}$ be a compact way of writing $C(\Psi_{ij}^{kk+1})$ and $\hat{f}_{ij}^{kk+1}$.

$$P_{kk+1} =$$

Algorithm 1 Boosting for data association used in multi-target tracking algorithm

```
Input: $\{(\Psi_1,f_1),\ldots,(\Psi_i,f_i),\ldots,(\Psi_N,f_N)$ with the number of corresponding class $N_p$, $N_s$ $\}$
  Weights initialization: $w_1^i = 1/(N_p + N_s)$, $\forall i$
  for $\tau = 1$; $\tau \le T$; $\tau + +$ do
    Weights normalization:
      $$\bar{w}_\tau^i = \frac{w_\tau^i}{\sum_{n=1}^{N} w_\tau^j}, \qquad \forall i \qquad (2)$$
    Select the best weak classifier: the one that minimizes the weighted error,
      $$\Theta_\tau = \arg\min_{\Theta} \sum_{i=1}^{N} \bar{w}_\tau^i | \, c(\Psi_\tau, \Theta) - f_i| \qquad (3)$$
    Define $e_\tau$ the corresponding weighted error.
    Update the weights:
      $$w_\tau^{i+1} = \bar{w}_\tau^i (\frac{e_\tau}{1-e_\tau})^{1-\delta_i} \qquad \forall i \qquad (4)$$
      where
      $$\delta_i = \begin{cases} 1 & c(\Psi_i, \Theta_\tau) = f_i \\ 0 & c(\Psi_i, \Theta_\tau) \neq f_i \end{cases}$$
  end for
Output:
  $$C(\Psi) = 1 - \frac{\sum_{\tau=1}^{T} log(\frac{1-e_\tau}{e_\tau}) c(\Psi, \Theta_\tau)}{\sum_{\tau=1}^{T} log(\frac{1-e_\tau}{e_\tau})} \qquad (5)$$
```

$$\begin{bmatrix} (C_{11},\hat{f}_{11}) & (C_{12},\hat{f}_{12}) & \ldots & (C_{1n_{k+1}},\hat{f}_{1n_{k+1}}) \\ \vdots & \vdots & \ddots & \vdots \\ (C_{n_k1},\hat{f}_{n_k1}) & (C_{n_k2},\hat{f}_{n_k2}) & \ldots & (C_{n_kn_{k+1}},\hat{f}_{n_kn_{k+1}}) \end{bmatrix} \tag{6}$$

To solve the data association and achieve the global optimum between a pair of frame k and $k+1$, we solve the following optimization problem,

$$\hat{f}^{\,*} = \arg\max_{\hat{f}} \sum \hat{f} C \tag{7}$$

with

$$\sum_i \hat{f}_{ij} \le 1, \quad \sum_j \hat{f}_{ij} \le 1, \quad \hat{f}_{ij} \in [0,1], \quad \hat{f}_{ij} C_{ij} \ge \mu \tag{8}$$

where $\hat{f}_{ij}$ denotes the predicted class label that is 1 when the feature vector Ψ_{ij} belonging to positive class. We interpret the likelihood of the feature vector C_{ij} as the weight of edges in a bipartite graph. The optimization problem in (7) is equivalent to finding the maximum weight matching under the constraint (8).

$$\mathcal{P}_{kk+1} = \left(\begin{array}{c|c} P_{kk+1} & P_{kk+1}^{mi} \\ \hline P_{kk+1}^{en} & -\infty \end{array} \right)$$

$$= \left(\begin{array}{ccc|ccc} C_{11} & \ldots & C_{1n_{k+1}} & \mu & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ C_{n_k1} & \ldots & C_{n_kn_{k+1}} & 0 & \ldots & \mu \\ \hline \mu & \ldots & 0 & & & \\ \vdots & \ddots & \vdots & & -\infty & \\ 0 & \ldots & \mu & & & \end{array} \right) \tag{9}$$

To solve the constraint (8), we construct an extended probability map $\mathcal{P}_{kk+1}$ (9). For clarity, we omit the predicted class label $\hat{f}_{ij}$ in $\mathcal{P}_{kk+1}$. We introduce two groups of virtual maps P_{kk+1}^{en} and P_{kk+1}^{mi} . Munkres algorithm [14] can be applied to the extended probability map $\mathcal{P}_{kk+1}$, solving the optimization problem in (7) with the constraint (8).

III. Modeling scene layout to handle exceptions

Based on the Boosting algorithm described in Section II, we can track all possible pairs of targets between two adjacent frames. However, for the applications under the real traffic scenarios, the rapidly changing traffic environment, fast moving targets and long-time occlusions exist. Therefore, we apply the scene layout model to handle this kind of exceptions. Specifically, we advocate the importance of geometric contextual reasoning for object detection and tracking. We discuss the functionality of the scene layout to model the false observations and occlusions

A. Modeling objects and scene layout(MOSL)

We combine available prior knowledge with image evidence to reconstruct the 3D positions of all objects in the scene. Following that we first introduce notations and assumptions and then formulate the problem.

Assumptions and notations We assume that each object lies on the ground at an upright pose. This assumption is satisfied in most real world scenes. For the traffic scenarios, a vehicle is usually touching the ground by four tyres rather than only one and a pedestrian is usually standing vertically rather than obliquely. The ground plane is parameterized by its surface normal $\vec{n}$ and its distance h_n to the origin of the coordinate system (e.g. the camera). Let r be the ray that connects the objects' center *O* and the camera center. Let zenith angle ϕ be the angle between ray r and $\vec{n}$. We define α the object's observation pose. We define *bbox* by the height *h* and width *w* of the object on the image coordinate framework. We denote scene information $\mathcal{L} = (\vec{n}, h_n, f_c)$.

Modeling objects and scene layout The goal of this work is to infer the relationship between $\mathcal{L}$ and objects and how to locate objects in the 3D camera reference system. Denote by $||r||$ the distance between the object location *O* and the camera. Assuming that we have some prior knowledge about the real size of the 3D object, the object distance $||r||$ can be estimated from the object scale in the image by means of an inversely proportional relationship. Specifically,if an object's *bbox*'s height and width are h and w, its category is *c*, and given pose α

and ϕ, we just use the linear regression to approximate its distance $||r||$ by the following linear combination:

$$||r|| \simeq \theta_1(\alpha, \phi, c)\frac{f_c}{w} + \theta_2(\alpha, \phi, c)\frac{f_c}{h} \tag{10}$$

where θ_1 and θ_2 are functions of the object's pose and category. A more precise modeling of this relationship goes beyond the scope of this paper. We instead use linear regression to learn θ_1 and θ_2 for each set of (α, ϕ, c) where ground truth pose and distance $||r||$ are available. As a result, given the detection and its image coordinates (u, v) from the detector, its 3D coordinates $\mathcal{D}$ can be estimated in the camera coordinates as follows:

$$\mathcal{D} = \frac{||r||}{\sqrt{v/{f_c}^2 + v/{f_c}^2 + 1}} \begin{pmatrix} u/f_c \\ v/f_c \\ 1 \end{pmatrix} \tag{11}$$

This allows us to relate the 3D coordinates of detections, the scene information $\mathcal{L}$, and the distance d between detection and the ground as $d = \mathcal{D}^T \vec{n} + h_n$

B. False Detections Rejecting(FDR)

False detection is common since the object detection algorithms can not provide perfect detection results. [13] assumes that false detections occur randomly, and have short trajectories. However, this may remove some true-positive objects which have short trajectories. [6] extracts additional features like shape features to provide shape-constraints among objects. However, this is computationally expensive. To effectively reject false detections, we propose to reason which detections may be rejected by MOSL constructed in III-A.

The first step of FDR is to generate *false detections hypothesis* $\hat{x}_i^t$. We say that detection x_i^t is considered false detections if and only if the observation likelihood of x_i^t is smaller than a certain threshold. In the second step of FDR, the proposed MOSL formulations (10) and (11) are applied to the *false detections hypothesis* $\hat{x}_i^t$.This allows us to obtain the distance d_i^t between $\hat{x}_i^t$ and the ground. We say that *false detections hypothesis* $\hat{x}_i^t$ is considered false detection if and only if the distance d_i^t is below a certain threshold.

C. Occlusion/Reappearing model

Long-term occlusion and coexistent occlusions are another critical challenge in visual tracking. To effectively handle this kind of problems, we propose to reason explicitly about which objects may be occluded by which others and which objects may be reappearing at which time. Following that we construct an explicit occlusion/reappearing model (ORM) in two steps.

Occlusion hypotheses generation The ORM generates a set of occlusion hypotheses. Only occlusions between tracked objects are addressed. We obtain a missing object set X_{k-1}^{mi} from frame $k-1$. For each missing object x_{k-1}^i, x_{k-1}^i is directly occluded by a tracked object x_k^i if and only if $||r_{k-1}^i||$ is greater than $||r_k^j||$, and the expected visibility $\mathcal{V}_{k-1}^i$ is below a certain threshold $\mathcal{V}_{min}$. Given the observation x_{k-1}^i , we can compute the expected visibility :

$$\mathcal{V}_{k-1}^i = Area(x_{k-1}^i \cap x_k^j)/Area(x_{k-1}^i) \tag{12}$$

where $Area(x_k^i)$ denotes the image area in pixels covered by the projections of x_k^i. We then add it to the occlusion hypotheses set $\tilde{\mathcal{X}}_k$ and update h_i^{k-1}. Repeat this until no new hypotheses can be generated.

Occlusion/Reappearing hypotheses verification We apply the same constraints in occlusion hypotheses generation to $\tilde{\mathcal{X}}_k$ and X_{k+1}. For the hypotheses which pass the constraints, we update them to set $\tilde{\mathcal{X}}_{k+1}$. Let X_{k+1}^r be the observation residuals (unassociated observations) of X_{k+1}. The proposed optimization (Sec.II-C) is applied again to the probability map generated by $\tilde{\mathcal{X}}_{k+1}$ and X_{k+1}^r. Hypotheses in $\tilde{\mathcal{X}}_{k+1}$ that are matched with detections in X_{k+1}^r are considered reappearing objects. The unmatched hypotheses are maintained for potential hypotheses in the future.

IV. Data set

We augment a subset of the KITTI vision benchmark [8] with rough detections and annotated track ID. The KITTI vision benchmark is captured by driving around a mid-size city, in rural areas and on highways. Three challenging sequences under different scenarios are selected to evaluate our approach, named RURALseq, URBANseq and HIGHWAYseq. We run an "out-of-the-box" joint object and pose vehicle detector [18] with a lower threshold. This means to run an object detector with more false detections. We manually annotate the track IDS on these three sequences.

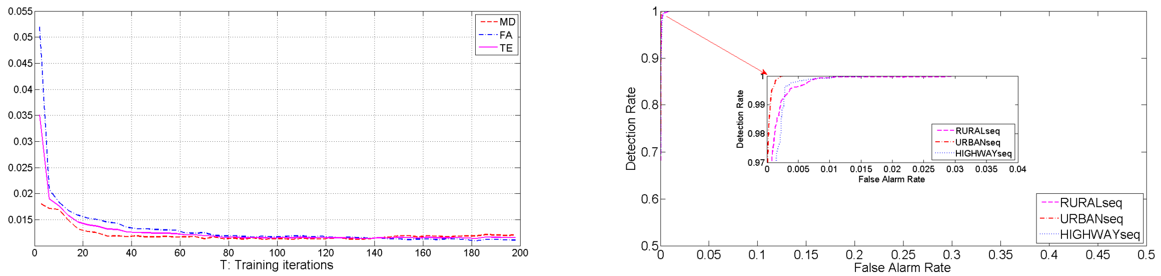
V. Experiments

Our multi-target tracking technique based on the Boosting and Scene layout is applied to track multiple vehicles in the real traffic scenarios, and its performance is validated on RURALseq, URBANseq and HIGHWAYseq. In this section, we first describe the features used in our approach. Next we quantitatively evaluate our training process. Finally, we present the performance of our approach and analyze the experimental results.

A. Features

We compute the feature vector Ψ_{ij} between each pair of detections from a pair of consecutive frames. Three different kinds of features are employed in our application: *detection features*, *layout features* and *image features*. The intuition behind the choises of feature is that the computational cost of the feature vector Ψ_{ij} is minimal and different kinds of features are complementary to each other.

For *detection features*, we calculate the pose similarity of the pair detections. We also compute the difference of the observation likelihood of the pair detections given the penalty to their sum. For *layout features*, we compute the Euclidean distances of the pair observations under the vehicle and image coordinates. The visible coverage of the pair observations



(a) Error rates against the training rounds T

(b) Receiver operating characteristic curves

Fig. 2. Classifier performance

$Area(x^i_{k-1} \cap x^j_k)/Area(x^i_{k-1} \cup x^j_k)$ are also used. For the *image features*, we employ the histogram of LBP feature [20] and HSV features. We first define the inside region and outside region in the *bbox*. Then, we divide both inside and outside regions into 2×2 and 3×3 sub windows. We compute the histogram of LBP feature and HSV feature at each sub window. The affinity between a pair of histograms is computed by Bhattacharyya coefficient.

The final feature vector Ψ_{ij} is 51 dimensional – 2 dimensions for *detection features*, 3 dimensions for *layout features* and 46 dimensions for *image features*

B. Analysis of training process

In this section we evaluate the classifiers learned in Sec II-B. When learning a classifier, an initial important step is to determine an appropriate number of training rounds *T*. The strong classifier's receiver operating characteristics (ROC) are evaluated since the performance of the classifier directly affects the tracking results. The classifier is evaluated in terms of detection rate (*TD*), missed detection rate (*MD*) and false alarms rate (*FA*). The experiments in this section were conducted using 10-fold cross validation on the data sets.

Number of training rounds Strong classifiers were trained for different values of *T*, the resulting error rates are shown in Fig. 2a. The validation error levels decrease as the learning algorithm iterates up until about 80 training iterations. Hence, *T*=80 was chosen for all subsequent experiments.

Classifier performance In this section we present the performance of the classifiers in terms of *TD* and *FA*, as defined above. Fig. 2b shows ROC curves for the classifiers . The area under the ROC-curve is approximately 1 for all data sets. Good levels of detection are achieved.

C. Tracking performance

Evaluation metrics To evaluate our system, we use the performance metrics described in [21]: Rec - correctly matched objects to the ground truth; Prec - correctly matched objects to output objects; GT - number of trajectories in the ground truth; MT - percentage of trajectories tracked for more than 80%; ML - percentage of trajectories tracked for less than 20% ; Frag - times that a trajectory is interrupted; IDS - times that a tracked trajectory changes its matched GT identity.

Table 1: Tracking results in RURALseq (**R**), URBANseq(**U**) and HIGHWAYseq (**H**)

Method	Test	Train	Rec	Prec	GT	MT	ML	Frag	IDS
Boost	**R**	U	0.933	0.892	41	0.926	0.073	**1**	1
BoostScene	**R**	U	0.977	0.939	41	0.951	0.048	2	1
Boost	**R**	H	0.943	0.892	41	0.950	0.049	2	1
BoostScene	**R**	H	**0.982**	**0.942**	41	**0.976**	**0.024**	2	2
Boost	**U**	R	0.795	0.936	56	0.607	0.160	7	0
BoostScene	**U**	R	0.979	**0.960**	56	**0.964**	**0.017**	7	0
Boost	**U**	H	0.827	0.938	56	0.660	0.142	7	0
BoostScene	**U**	H	**0.980**	0.959	56	**0.964**	**0.017**	7	1
Boost	**H**	U	**0.944**	0.836	24	0.875	**0.041**	1	0
BoostScene	**H**	U	0.937	0.891	24	**0.917**	0.083	1	0
Boost	**H**	R	0.942	0.814	24	0.833	0.083	1	0
BoostScene	**H**	R	0.937	**0.892**	24	**0.917**	0.083	**0**	0

Results and Discussion We report our tracking performance on our new datasets by 3-fold cross validation. Examples of the tracking results are shown in Fig. 3. Table 1 shows the quantitative results obtained for the RURALseq, URBANseq and HIGHWAYseq datasets. Each test data contains four different results by our algorithm since two different training set were tested. Among them, ”Boosting” denotes our tracking method without modeling scene layout, while ”BoostingScene” means our final method. As shown in Table 1, from the Boost to BoostScene, the tracking performance is progressively increased. BoostScene holds superiority in most of the metrics. Especially, BoostScene tends to achieve high precision and recall compared to Boost, which means that our scene layout model can reject most of the false detections. Note that our method tends to have more fragments but fewer ID switches. This is natural as our algorithm is an online method that achieves global optimum between adjacent frames.

VI. Conclusion

We present a framework to track multi target in real traffic scenarios. Our method integrated the Boosting algorithm and the scene layout to enhance the performance of the tracker. Our algorithm can deal with various exceptions such as false alarms, long-term occlusions and occlusion

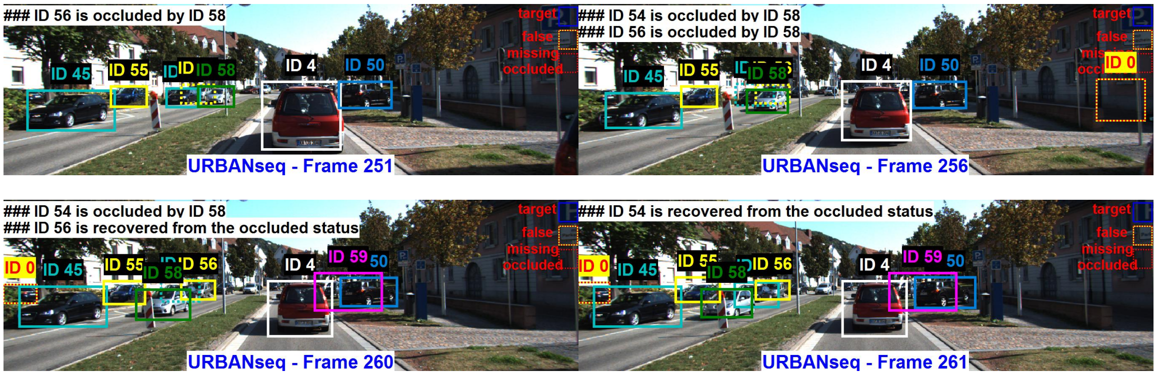

### ID 56 is occluded by ID 58
target
false
missing
occluded
ID 45
ID 55
ID
ID 58
ID 4
ID 50
URBANseq - Frame 251
### ID 54 is occluded by ID 58
### ID 56 is occluded by ID 58
ID 0
URBANseq - Frame 256
### ID 54 is occluded by ID 58
### ID 56 is recovered from the occluded status
ID 56
ID 59
URBANseq - Frame 260
### ID 54 is recovered from the occluded status
URBANseq - Frame 261

Fig. 3. Tracking results of our method: show occlusion/reappearing model. ID 56 is occluded by others for 10 frames. Although two occlusions coexist in our scenario. Our method successfully tracks these targets.

reappearing. Furthermore, we provide three new data sets of the real world scene. We hope that these data sets could push forward the performance of tracking systems when being moved outside the laboratory to the real world.

References

[1] Anton Andriyenko and Konrad Schindler. Globally optimal multi-target tracking on a hexagonal lattice. In *ECCV*, volume 1, pages 466–479, 2010.

[2] Sid Yingze Bao, Yu Xiang, and Silvio Savarese. Object co-detection. In *Proc. of European Conference of Computer Vision*, 2012.

[3] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1273–1280, 2011.

[4] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[5] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[6] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[7] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361, 2012.

[9] Varsha Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1849–1856, 2009.

[10] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. *Int. J. Comput. Vision*, 80(1):3–15, October 2008.

[11] Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 788–801, Berlin, Heidelberg, 2008. Springer-Verlag.

[12] Hao Jiang, S. Fels, and J.J. Little. A linear programming approach for multiple object tracking. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[13] Suna Kim, Suha Kwak, Jan Feyereisl, and Bohyung Han. Online multi-target tracking by large margin structured learning. In *ACCV (3)*, pages 98–111, 2012.

[14] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97, 1955.

[15] Cheng-Hao Kuo, Chang Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 685–692, 2010.

[16] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007.

[17] Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR'09*, pages 2953–2960, 2009.

[18] R.J. Lopez-Sastre, T. Tuytelaars, and S. Savarese. Deformable part models revisited: A performance evaluation for object category pose estimation. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1052–1059, 2011.

[19] H. Pirsiavash, D. Ramanan, and C.C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1201–1208, 2011.

[20] Ojala T. and Pietikinen M & Menp T. A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. 2001. In: Advances in Pattern Recognition, ICAPR 2001 Proceedings, Lecture Notes in Computer Science 2013, Springer, 397 - 406.

[21] Bo Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 951–958, 2006.

[22] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *Int. J. Comput. Vision*, 75(2):247–266, November 2007.

[23] Bo Yang and R. Nevatia. An online learned crf model for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2034–2041, 2012.

[24] Ming Yang, Fengjun Lv, Wei Xu, and Yihong Gong. Detection driven adaptive multi-cue integration for multiple human tracking. In *ICCV'09*, pages 1554–1561, 2009.

[25] Li Zhang, Yuan Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.