Learning to Associate: HybridBoosted Multi-Target Tracker for Crowded Scene

Yuan Li, Chang Huang and Ram Nevatia
University of Southern California, Institute for Robotics and Intelligent Systems
Los Angeles, CA 90089, USA
{yli8|huangcha|nevatia}@usc.edu

Abstract

We propose a learning-based hierarchical approach of multi-target tracking from a single camera by progressively associating detection responses into longer and longer track fragments (tracklets) and finally the desired target trajectories. To define tracklet affinity for association, most previous work relies on heuristically selected parametric models; while our approach is able to automatically select among various features and corresponding non-parametric models, and combine them to maximize the discriminative power on training data by virtue of a HybridBoost algorithm. A hybrid loss function is used in this algorithm because the association of tracklet is formulated as a joint problem of ranking and classification: the ranking part aims to rank correct tracklet associations higher than other alternatives; the classification part is responsible to reject wrong associations when no further association should be done. Experiments are carried out by tracking pedestrians in challenging datasets. We compare our approach with state-of-the-art algorithms to show its improvement in terms of tracking accuracy.

最佳关联

1. Introduction

The aim of multi-target tracking is to infer target trajectories from image observations in a video. It is a highly challenging problem in crowded environments where occlusions are frequent and many targets have similar appearance and intersecting trajectories. In such situations, tracking approaches that make online decisions on a frame-by-frame basis [1][2] may yield identity switches and trajectory fragmentations due to ambiguous and noisy observation. Considering more frames and performing global inference can give improved results. This has led to development of Data Association based Tracking (DAT) approaches [3][4][5][6][7][8], which link short track fragments (*i.e.*, tracklets) or detection responses into trajectories by global optimization based on position, size and appearance similarity.

There are two main components in DAT approaches: one

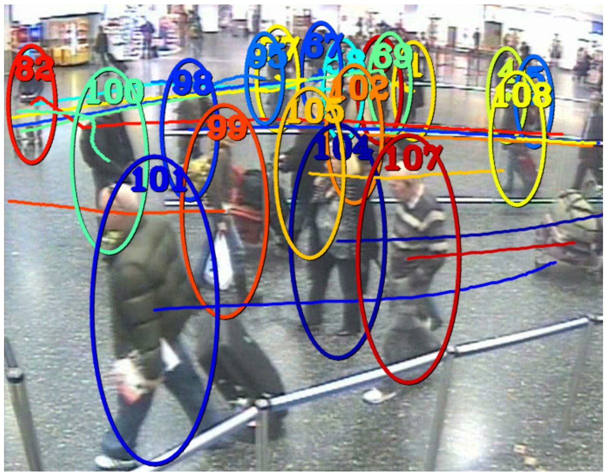

Figure 1. Sample tracking result of our approach.

追踪近似模型==>motion预测模型

is a tracklet affinity model that measures the likelihood that two tracklets belong to the same target; the other is the association optimization framework that determines which tracklets should be linked based on tracklet affinity measurements. While many algorithms have been proposed for the latter component (Hungarian algorithm [7], Linear Programming [4] and cost-flow network [6] are few representatives), there has been much less effort in improving the affinity model. Commonly used affinity models are simple parametric models, *e.g.*, zero-mean Gaussian distributions for object position change and distance between color histograms. Moreover, in many cases, the model parameters and the relative weights of different cues are determined based purely on prior knowledge or human observation of the data. When the application environment changes or multiple cues such as appearance, motion, and context are fused into one affinity model, the effort and expertise required in manual model tuning becomes unaffordable.

We propose an algorithm for learning an affinity model in DAT. Tracklet association is viewed as a joint problem of ranking and classification. The ranking part should rank the correct association higher than the wrong ones, while the classification part should reject further associa-

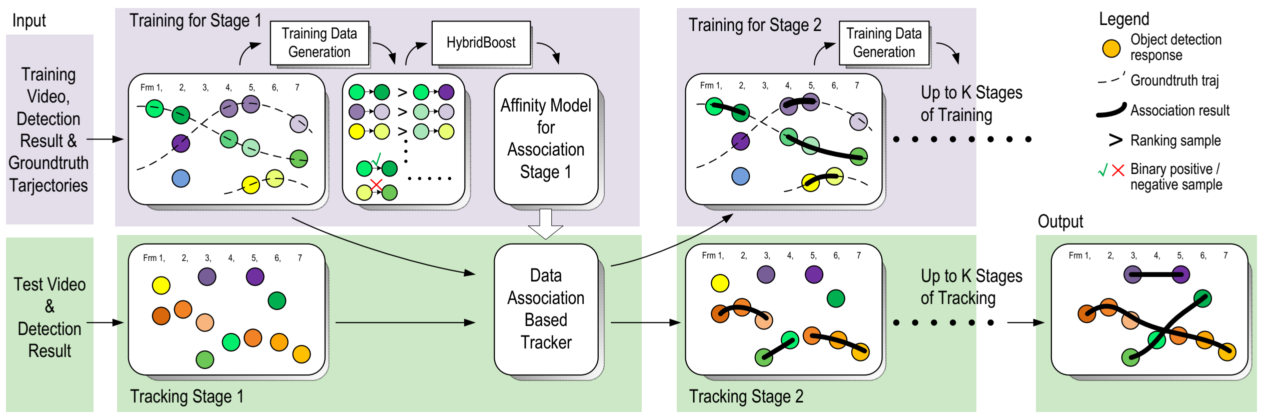
Input
Training Video, Detection Result & Groundtruth Tarjectories
Training for Stage 1
Training Data Generation
HybridBoost
Affinity Model for Association Stage 1
Training for Stage 2
Up to K Stages of Training
Legend
Object detection response
Groundtruth traj
Association result
Ranking sample
Binary positive / negative sample
Test Video & Detection Result
Data Association Based Tracker
Up to K Stages of Tracking
Output
Tracking Stage 1
Tracking Stage 2
Frm 1, 2, 3, 4, 5, 6, 7

Figure 2. Approach overview.

tions when not necessary. To learn an affinity model which has the above capabilities, we develop a new boosting algorithm called HybridBoost, which combines the merits of the RankBoost algorithm [9] and the AdaBoost algorithm [10]. By minimizing a hybrid loss function on training data, a strong *ranking classifier* [1] is boosted from a pool of weak ranking classifiers. Each weak ranking classifier is based on a single type of evidence (feature) for tracklet affinity measurement such as the differences between appearance descriptors, motion smoothness and the frame gap between a pair of tracklets. Model parameters of each feature and its voting weight in the strong ranking classifer are automatically learned based on the training data.

We adopt a hierarchical association framework similar to that of [8]. As shown in Figure 2, tracking is divided into multiple stages to progressively associate tracklets. The min-cost flow method [6] is used for optimization at each stage. Correspondingly, training is also done stage-wise: a strong ranking classifier is learned for each stage to compute the affinity between any two tracklets. One can think of this as being analogous to a cascade classifier, in the sense that the cascade classifier rejects false positives stage by stage, while our tracker gradually reduces trajectory fragmentation.

The rest of the paper is organized as follows. Related work is discussed in Section 2. The problem formulation and optimization framework are given in Section 3. Section 4 describes the HybridBoost algorithm and the training of strong ranking classifiers as the affinity models. Experimental results are shown in Section 5 and Section 6 concludes the paper.

[1]We call it a ranking classifier since it has the functionality of both a ranker and a classifier.

2. Related work

global optimal data association

To resolve the ambiguity in multi-target tracking, Multi-Hypothesis Tracking [11] and Joint Probabilistic Data Association Filters [12] are among the earliest to propose inference over multiple targets by looking at a longer period of time in contrast to frame-by-frame tracking. However, the search space of these methods grows exponentially with the number of frames. To overcome this, a category of Data Association based Tracking algorithms (DAT) has emerged. [3] combines Dynamic Programming with a greedy strategy to optimize one trajectory at a time. [4][5][6][7][8] aim at simultaneously optimizing all tracks in polynomial time by tailoring various optimization algorithms such as Linear Programming [4], Quadratic Boolean Programming [5], min-cost flow [6] and Hungarian algorithm [7] or its hierarchical version [8].

Although great progress has been made in global optimal data association, the affinity model (also referred to as link probability or association cost) of DAT has received relatively little attention. In previous work, the cues used to provide affinity measure include appearance (*e.g.*, color histogram similarity), motion (*e.g.*, speed), and frame gap between two tracklets. Typically, a simple parametric model is used for each cue (*e.g.*, Euclidean distance [2], single Gaussian distribution [6] and stump function for max speed [4]). The models are combined by product [6] or weighted sum [4]. Parameters are obtained empirically and in many cases, manually. To the best of our knowledge, there has been no extensive use of machine learning algorithm in building the affinity model, except for computing simple statistics such as the mean and variance of appearance difference and detection rate [6][8].

3. MAP formulation for Tracklet Association

In our approach, several stages of data association are performed hierarchically to progressively grow the tracklets as in [8]. At stage k, given the tracklet set $\mathcal{T}^{k-1} = \{T_i^{k-1}\}$ from the previous stage, the tracker tries to link the tracklets into longer ones that constitute a new tracklet set $\mathcal{T}^k = \{T_j^k\}$, in which $T_j^k = \{T_{i_0}^{k-1}, T_{i_1}^{k-1}, \cdots, T_{i_{l_j}}^{k-1}\}$.

To obtain an optimal association result, following [8], we formulate this process as an MAP problem

$$\begin{aligned}\mathcal{T}^{k*} &= \underset{\mathcal{T}^k}{\mathbf{argmax}}\, P(\mathcal{T}^{k-1}|\mathcal{T}^k)P(\mathcal{T}^k)\\ &= \underset{\mathcal{T}^k}{\mathbf{argmax}} \prod_{T_i^{k-1}\in\mathcal{T}^{k-1}} P(T_i^{k-1}|\mathcal{T}^k) \prod_{T_j^k\in\mathcal{T}^k} P(T_j^k), \quad (1)\end{aligned}$$

and model $P(T_j^k)$ as Markov Chains. Thus we have

$$\begin{aligned}\mathcal{T}^{k*} = \underset{\mathcal{T}^k}{\mathbf{argmax}} &\prod_{T_i^{k-1}:\forall T_j^k\in\mathcal{T}^k, T_i^{k-1}\notin T_j^k} P_-(T_i^{k-1})\\ &\prod_{T_j^k\in\mathcal{T}^k}\Big[P_{init}(T_{i_0}^{k-1})P_+(T_{i_0}^{k-1})P_{link}(T_{i_1}^{k-1}|T_{i_0}^{k-1})\cdots\\ &P_{link}(T_{i_{l_k}}^{k-1}|T_{i_{l_k-1}}^{k-1})P_+(T_{l_k}^{k-1})P_{term}(T_{i_{l_k}}^{k-1})\Big], \qquad (2)\end{aligned}$$

where $P_+(T_i^{k-1})$ is the likelihood that T_i^{k-1} is a true tracklet and $P_-(T_i^{k-1})$ is the likelihood that it is a false tracklet. Both can be modeled by Bernoulli distribution given the detection precison and tracklet length [6][8]. $P_{init}(T)$ and $P_{term}(T)$ are the initialization and termination probability, while $P_{link}(T|T')$ is the transition probability. Notice that some tracklets given by the previous stage may be excluded from the optimal tracklet set $\mathcal{T}^{k*}$ as being false tracks.

By defining an inner cost and a transition cost as

$$\begin{aligned}L_I(T_i^{k-1}) &= \ln\frac{P_{init}(T_i^{k-1})P_+(T_i^{k-1})P_{term}(T_i^{k-1})}{P_-(T_i^{k-1})},\\ L_T(T_j^{k-1}|T_i^{k-1}) &= \ln\frac{P_{link}(T_j^{k-1}|T_i^{k-1})}{P_{term}(T_i^{k-1})P_{init}(T_j^{k-1})}, \quad (3)\end{aligned}$$

we can rewrite (2) as

$$\begin{aligned}\mathcal{T}^{k*} &= \underset{\mathcal{T}^k}{\mathbf{argmax}} \prod_{T_i^{k-1}\in\mathcal{T}^{k-1}} P_-(T_i^{k-1}) \prod_{T_j^k\in\mathcal{T}^k}\Big[e^{L_I(T_{i_0}^{k-1})}\\ &\quad e^{L_T(T_{i_1}^{k-1}|T_{i_0}^{k-1})}e^{L_I(T_{i_1}^{k-1})}\cdots e^{L_I(T_{i_{l_k}}^{k-1})}\Big]\\ &= \underset{\mathcal{T}^k}{\mathbf{argmax}} \sum_{T_j^k\in\mathcal{T}^k}\Big[L_I(T_{i_0}^{k-1}) + L_T(T_{i_1}^{k-1}|T_{i_0}^{k-1})\\ &\quad + L_I(T_{i_1}^{k-1})\cdots + L_I(T_{i_{l_k}}^{k-1})\Big]. \qquad (4)\end{aligned}$$

The cost-flow method [6] and the Hungarian method [8] have shown their efficiency in solving this MAP problem. However, how to compute those probabilities (or costs) still remains arguable. In particular, $L_T(T_j^{k-1}|T_i^{k-1})$, as the affinity measurement between two tracklets, plays the most important role in the tracklet association. In the coming section, the HybridBoost algorithm is proposed to learn strong ranking classifiers for the computation of this crucial transition cost.

4. Affinity model for association

In this section we show that the association affinity model has the property of both a ranking function and a binary classifier, and propose a hybrid boosting algorithm to solve the ranking and classification problems simultaneously. Following that we give our design of feature pool, weak learner and training process.

4.1. The HybridBoost algorithm

A classic ranking problem involves an instance space $\mathcal{X}$ and a ranker H which defines a linear ordering of instances in $\mathcal{X}$. H typically takes the form of $H : \mathcal{X} \to \mathbb{R}$. Training data is given as a set of instance pairs $\mathcal{R} = \{\langle x_i, x_j\rangle | x_i, x_j \in \mathcal{X}\}$, meaning that x_j should be ranked higher than x_i, *i.e.* $H(x_j) > H(x_i)$. The goal is to find an H which describes the desired preference or ranking over $\mathcal{X}$ indicated by the data. RankBoost [9] is an algorithm developed for this purpose, which has been applied to web search, face recognition and other tasks.

The tracklet association problem can be mapped to the ranking problem as follows. Define the instance space to be $\mathcal{X} = \mathcal{T} \times \mathcal{T}$, where $\mathcal{T}$ is the set of tracklets to perform association on. For tracklets $T_1, T_2, T_3 \in \mathcal{T}$, if tracklet T_1 should be linked to T_2 to form a correct trajectory, and T_1 and T_3 should not be linked (*e.g.* belong to different targets), then there is a ranking preference $H(\langle T_1, T_2\rangle) > H(\langle T_1, T_3\rangle)$. Therefore H can be used to compute the transition cost $L_T(T|T')$ in (3).

Moreover, in the tracking problem, the affinity model is not limited to keeping the relative preference over any two tracklet pairs, but also needs to output a low value for any tracklet pair that should not be associated. To see why this is necessary, let T be the terminating tracklet of a target trajectory, the affinity model should prevent associating T to any other tracklet T'. In this case, no relative ranking preference is present, but it is desirable to keep $H(\langle T, T'\rangle) < \zeta, \forall T' \in \mathcal{T}$, where ζ is a certain rejection threshold. In this sense, this is no longer a simple ranking problem but a combination of ranking and binary classification.

To solve the two problems simultaneously, we combine RankBoost [9] with Adaboost [10] to form a HybridBoost algorithm. The training set includes both a ranking sample set $\mathcal{R}$ and a binary sample set $\mathcal{B}$. The ranking sample set is denoted by

$$\mathcal{R} = \{(x_{i,0}, x_{i,1}) | x_{i,0} \in \mathcal{X}, x_{i,1} \in \mathcal{X}\}, \qquad (5)$$

where $x_{i,0}$ and $x_{i,1}$ each represent a pair of tracklets as a candidate for association , and $(x_{i,0}, x_{i,1}) \in \mathcal{R}$ indicates that the association of $x_{i,1}$ should be ranked higher than $x_{i,0}$. The binary sample set is denoted by

$$\mathcal{B} = \{(x_j, y_j) | x_j \in \mathcal{X}, y_j \in \{-1, 1\}\}, \tag{6}$$

where $y_j = -1$ means that the corresponding x_j should not be associated at any time, while $y_j = 1$ means the contrary. In Section 4.3, we will discuss how to generate the training sets.

A new loss function for boosting is defined as a linear combination of ranking loss and binary classification loss:

$$\begin{aligned} Z = \beta & \sum_{(x_{i,0},x_{i,1})\in\mathcal{R}} w_0(x_{i,0}, x_{i,1}) \exp\big(H(x_{i,0}) - H(x_{i,1})\big) \\ & +(1-\beta) \sum_{(x_j,y_j)\in\mathcal{B}} w_0(x_j, y_j) \exp\big(-y_j H(x_j)\big), \end{aligned} \tag{7}$$

where β is a coefficient to adjust the emphasis on either part (choice of β discussed in Section 5.3). w_0 is the initial weight of each sample, which will be updated during boosting. The goal is to find an $H(x)$ that minimizes Z. As in traditional boosting, H is obtained by sequentially adding new weak ranking classifiers. In the t-th round, we try to find an optimal weak ranking classifier $h_t : \mathcal{X} \to \mathbb{R}$ and its weight α_t that minimizes

$$\begin{aligned} Z_t = \beta & \sum_{(x_{i,0},x_{i,1})\in\mathcal{R}} w_t(x_{i,0}, x_{i,1}) \exp\big(\alpha_t(h_t(x_{i,0}) - h_t(x_{i,1}))\big) \\ & +(1-\beta) \sum_{(x_j,y_j)\in\mathcal{B}} w_t(x_j, y_j) \exp\big(-\alpha_t y_j h_t(x_j)\big), \end{aligned} \tag{8}$$

and update the sample weight according to h_t and α_t to emphasize difficult ranking and binary samples. The final strong ranking classifier is the weighted combination of the selected weak ranking classifiers: $H(x) = \sum_{t=1}^{n} \alpha_t h_t(x)$, where n is the number of boosting rounds. The algorithm of HybridBoost is shown in Table 1.

Because of the hybrid form of our loss function, $H(x)$ has the merits of both a ranker and a classifier. When used in the affinity model, for any pair of tracklets $x = (T_1, T_2)$, we define the transition cost in (3) as

$$L_T(T_2|T_1) = \begin{cases} H(x) & \text{if } H(x) > \zeta, \\ -\infty & \text{otherwise.} \end{cases} \tag{9}$$

where ζ is a threshold which can conveniently control the trade off between trajectory fragmentation and risk of wrong association (ID switches). The default ζ learned by boosting is 0, which we use for all our experiments.

4.2. Feature pool and weak learner

The boosting algorithm works with a feature pool and a weak learner. Each feature is a function $f : \mathcal{X} \to \mathbb{R}$, namely

```
**Input**: ranking sample set $\mathcal{R} = \{(x_{i,0}, x_{i,1}) | x_{i,0} \in \mathcal{X}, x_{i,1} \in \mathcal{X}\}$,
binary sample set $\mathcal{B} = \{(x_j, y_j) | x_j \in \mathcal{X}, y_j \in \{-1, 1\}\}$.
**Output**:Strong ranking classifier $H : \mathcal{X} \to \mathbb{R}$.
**Initialize sample weight**:
    For each ranking sample $(x_{i,0}, x_{i,1})$, $w_0(x_{i,0}, x_{i,1}) = \frac{\beta}{|\mathcal{R}|}$.
    For each binary sample $(x_j, y_j)$, $w_0(x_j, y_j) = \frac{1-\beta}{|\mathcal{B}|}$.
**For** $t = 1, \ldots, n$ **do:**
  • On the current sample distribution, find optimal weak ranking classifier $h_t : \mathcal{X} \to \mathbb{R}$ and its weight $\alpha_t$ by weak learner (Table 2).
  • Update sample weight:
      For each ranking sample $(x_{i,0}, x_{i,1})$,
      $w_t(x_{i,0}, x_{i,1}) = w_{t-1}(x_{i,0}, x_{i,1}) \exp[\alpha_t(h_t(x_{i,0}) - h_t(x_{i,1}))]$.
      For each binary sample $(x_j, y_j)$,
      $w_t(x_j, y_j) = w_{t-1}(x_j, y_j) \exp[-\alpha_t y_j h_t(x_j)]$.
  • Normalize sample weight.
Output the final strong ranking classifier: $H(x) = \sum_{t=1}^{n} \alpha_t h_t(x)$.
```

Table 1. HybridBoost algorithm.

it takes a pair of tracklets $x = \langle T_1, T_2 \rangle$ as input and outputs a real value. The criterion for constructing the feature pool is to include any cue that can provide evidence for whether to associate two tracklets or not. In our implementation, five categories of 14 types of feature are used (Table 3); more can be added easily if necessary.

The weak learner aims at finding the optimal (h, α) to minimize the training loss Z in each boosting round. We use the stump function on a single feature as the weak ranking classifier:

$$h(x) = \begin{cases} 1 & \text{if } f(x) > \eta, \\ -1 & \text{otherwise.} \end{cases} \tag{10}$$

$h(x)$ monotonically increases with $f(x)$ because we use prior knowledge to pre-adjust the sign of each feature, *e.g.*, for the appearance feature, the smaller the χ^2 distance between two tracklets' color histograms is, the more likely that T_1 and T_2 should be associated.

To learn the optimal (h, α), we enumerate all the possible features f and a number of promising candidate thresholds η. Since h is fixed given f and η, we then compute

$$\hat{\alpha} = \mathop{\mathbf{argmin}}_{\alpha>0} Z(\alpha). \tag{11}$$

The algorithm of the weak learner is given in Table 2. To make the search for η and computation of Z more efficient, we build an adaptive histogram of the current training sample distribution on each feature f.

4.3. Training process

As mentioned in Section 3, we use multiple stages of data association, and for each stage an affinity model is learned. We hereby describe training sample generation for a single stage.

```
Input: ranking sample set $\mathcal{R} = \{(x_{i,0}, x_{i,1}) | x_{i,0} \in \mathcal{X}, x_{i,1} \in \mathcal{X}\}$; binary sample set $\mathcal{B} = \{(x_j, y_j) | x_j \in \mathcal{X}, y_j \in \{-1, 1\}\}$; current weight $w$ of each sample.
Output: weak ranking classifier $h : \mathcal{X} \to \mathbb{R}$ and its weight $\alpha$.
Initialize optimal loss $Z^* = \infty$.
For each feature $f \in$ feature pool $F$ do:
  For each candidate feature value threshold $\eta$ do:
    • Define $h(x) = \begin{cases} 1 & \text{if } f(x) > \eta, \\ -1 & \text{otherwise.} \end{cases}$
    • Compute loss function $Z$ as a function of $\alpha$ on current sample distribution:
      $Z(\alpha) = \beta \sum_i w(x_{i,0}, x_{i,1}) \exp[\alpha(h(x_{i,0}) - h(x_{i,1}))]$
      $\quad + (1-\beta) \sum_j w(x_j, y_j) \exp[-\alpha y_j h(x_j)]$
    • Compute $\hat{\alpha} = \mathbf{argmin}_{\alpha > 0} Z(\alpha)$, e.g., by Newton's method.
    • If $Z(\hat{\alpha}) < Z^*$, let $Z^* = Z(\hat{\alpha})$, $\alpha^* = \hat{\alpha}$, $h^* = h$.
Output the optimal weak ranking classifier and its $\alpha$: $h^*(x)$ and $\alpha^*$.
```

Table 2. Weak learner algorithm.

	Id	Feature description
Length	1	Length of T_1 or T_2.
	2	Number of detection responses in T_1 or T_2.
	3	Number of detection responses in T_1 or T_2 divided by length of T_1 or T_2.
Appearance	4	χ^2 distance between color histograms of the tail part of T_1 and the head part of T_2.
	5	The color histogram consistency of the object in the gap between T_1 and T_2 (the trajectory within the gap is interpolated assuming that T1 and T2 belong to the same object).
Gap	6	Frame gap between T_1's tail and T_2's head.
	7	Number of miss detected frames in the gap between T_1 and T_2.
	8	Number of frames occluded by other tracklets in the gap between T_1 and T_2.
	9	Number of miss detected frames in the gap divided by gap length.
	10	Number of frames occluded in the gap divided by gap length.
Entry Exit	11	Estimated time from T_1's head to the nearest entry point.
	12	Estimated time from T_2's tail to the nearest exit point.
Motion	13	Motion smoothness in image plane if connecting T_1 and T_2.
	14	Motion smoothness in ground plane if connecting T_1 and T_2.

Table 3. The list of feature types.

Let the tracklet set from the previous stage be $\mathcal{T}$ and the groundtruth track set be $\mathcal{G}$. First we compute a mapping

$$\varphi : \mathcal{T} \to \mathcal{G} \cup \{G_\emptyset\}, \tag{12}$$

so that $\varphi(T) = G \in \mathcal{G}$ indicates tracklet T is a correct tracklet matched with groundtruth track G, while $\varphi(T) = G_\emptyset$ indicates T is a false tracklet.

For each $T_i \in \mathcal{T}$, if $\varphi(T_i) = G \neq G_\emptyset$, we consider two possible association involving T_i: 1) connecting T_i's tail to the head of some other tracklet after T_i which is also matched to G; 2) connecting T_i's head to the tail of some other tracklet before T_i which is also matched to G.

For the first case, the possible correct association pairs and incorrect ones can be represented respectively by

$$\mathcal{X}_{i,1}^{tail} = \{\langle T_i, T_j\rangle | \varphi(T_j) = G, T_i \text{ is linkable to } T_j\}, \tag{13}$$

$$\mathcal{X}_{i,0}^{tail} = \{\langle T_i, T_j\rangle | \varphi(T_j) \neq G, T_i \text{ is linkable to } T_j\}. \tag{14}$$

Here T_i is linkable to T_j if and only if T_i occurs before T_j and the frame gap between them does not exceed the max allowed frame gap in the current association stage.

According to the definition of ranking problem, $\mathcal{X}_{i,0}^{tail} \times \mathcal{X}_{i,1}^{tail}$ will be valid ranking training samples. Similarly for connecting T_i's head with another tracklet's tail, we have

$$\mathcal{X}_{i,1}^{head} = \{\langle T_j, T_i\rangle | \varphi(T_j) = G, T_j \text{ is linkable to } T_i\}, \tag{15}$$

$$\mathcal{X}_{i,0}^{head} = \{\langle T_j, T_i\rangle | \varphi(T_j) \neq G, T_j \text{ is linkable to } T_i\}, \tag{16}$$

and $\mathcal{X}_{i,0}^{head} \times \mathcal{X}_{i,1}^{head}$ are ranking samples as well.

Therefore the ranking training sample set $\mathcal{R}$ is

$$\mathcal{R} = \bigcup_{T_i \in \mathcal{T}} \left(\mathcal{X}_{i,0}^{tail} \times \mathcal{X}_{i,1}^{tail}\right) \cup \left(\mathcal{X}_{i,0}^{head} \times \mathcal{X}_{i,1}^{head}\right). \tag{17}$$

As for the binary classification samples, positive samples include all the correct association pairs:

$$\mathcal{B}_1 = \bigcup_{T_i \in \mathcal{T}} \left(\mathcal{X}_{i,1}^{tail} \cup \mathcal{X}_{i,1}^{head}\right) \times \{1\}. \tag{18}$$

The negative samples are collected from those cases when all association choices are incorrect, to enforce the classification function to reject all of them:

$$\mathcal{B}_{-1} = \Bigg(\bigcup_{\substack{T_i \in \mathcal{T}, \\ \mathcal{X}_{i,1}^{tail} = \emptyset}} \mathcal{X}_{i,0}^{tail} \times \{-1\} \Bigg) \cup \Bigg(\bigcup_{\substack{T_i \in \mathcal{T}, \\ \mathcal{X}_{i,1}^{head} = \emptyset}} \mathcal{X}_{i,0}^{head} \times \{-1\} \Bigg). \tag{19}$$

Finally the binary classification sample set is a union of the two:

$$\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_{-1}. \tag{20}$$

Just as in the actual tracking process, training is also performed in a hierarchical way. As shown in Figure 2, for stage k, training consists of three steps: 1) use the groundtruth $\mathcal{G}$ and the tracklet set $\mathcal{T}_{k-1}$ obtained from stage $k-1$ to generate ranking and binary classification samples as described above; 2) learn a strong ranking classifier H_k by the HybridBoost algorithm; 3) apply the tracker using H_k as the affinity model to perform association on T_{k-1} and generate T_k, which is input to the next stage. This cycle is performed k times to build a k-stage tracker.

5. Experimental results

We applied our approach to human tracking and carried out the experiments on the CAVIAR dataset [13] and the TRECVID08 dataset [14] which features a much more challenging airport scene.

5.1. Implementation details

Given the detection results, we use the dual-threshold strategy to generate short but reliable tracklets on a frame-to-frame basis as done in [8]. After that, four stages of association are used. The maximum allowed frame gap for tracklet association in each stage is 16, 32, 64 and 128 respectively. For each stage, an affinity model (a strong ranking classifier H with 100 weak ranking classifiers) is trained to compute the transition cost $L_T(T|T')$, and the inner cost $L_I(T)$ in (3) is calculated in the way proposed by [8]. The combination coefficient β of the hybrid loss function in (8) is set to 0.75 for evaluation. The threshold ζ of each strong ranking classifier controls the tradeoff between fragmentation and ID switch. It can be either selected automatically based on training data or specified by the user. We simply use $\zeta = 0$ for all the strong ranking classifiers. There is no other parameter that needs human intervention.

5.2. Evaluation metrics

The evaluation metrics we use are listed in Table 4. A program is written to compute the metrics automatically. The key part of the evaluation program is the matching between groundtruth and tracking result, which is non-trivial itself. We implemented this part by Hungarian algorithm based on the the VACE evaluation software [15].

Our definitions of track fragments and ID switches are slightly different from those of [6] and [2] (referred to as traditional metric) as shown in Figure 3. Traditional ID switch is defined as "two tracks exchanging their ids". However the case in Figure 3(b) is not well-defined: TRK 1's identity changed but was not "exchanged" with others. We hereby define ID switch as a tracked trajectory changing its matched GT ID, *e.g.* in (a) there are two ID switches by our metric. Similar modification is made to the definition of fragments. Our definition is easier to implement but more strict and gives higher numbers in fragments and ID switches.

5.3. Analysis of the training process

Best features. For all of the four trained affinity models, the first three features selected in boosting are from motion smoothness (feature type 13 or 14), color histogram similarity (feature 4) and number of miss detected frames in the gap between the two trackelts (feature 7 or 9).

Strong ranking classifier output. Typically one strong ranking classifier includes multiple stump weak ranking classifiers on the same feature but with different thresholds. By combining them, the output on one feature takes the form of a piece-wise function. Figure 4 shows the output on two features (the color histogram similarity and the motion smoothness in image plane) learned in each stage. We can see that in early stages, very high appearance similarity is required for association; while in later stages the constraint becomes looser, allowing more tracklets to be linked. For motion smoothness, a decrease of importance of this feature is observed, because in later stages, tracklets are longer and more stable, the affinity model is placing more importance on motion smoothness in ground plane.

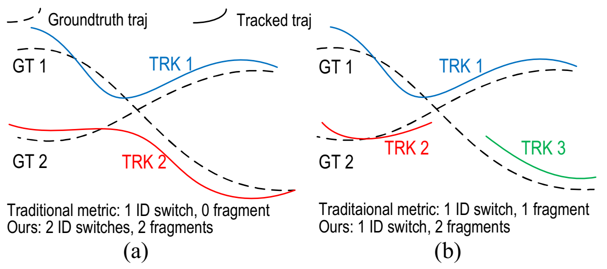
Groundtruth traj
Tracked traj
GT 1
TRK 1
GT 2
TRK 2
Traditional metric: 1 ID switch, 0 fragment
Ours: 2 ID switches, 2 fragments
(a)
GT 1
TRK 1
GT 2
TRK 2
TRK 3
Traditaional metric: 1 ID switch, 1 fragment
Ours: 1 ID switch, 2 fragments
(b)

Figure 3. Illustration of fragment and ID switch definitions.

Name	Definition
Recall	(Frame-based) correctly matched objects / total groundtruth objects.
Precision	(Frame-based) correctly matched objects / total output objects.
FA/Frm	(Frame-based) No. of false alarms per frame. *The smaller the better.*
GT	No. of groundtruth trajectories.
MT%	Mostly tracked: Percentage of GT trajectories which are covered by tracker output for more than 80% in length.
ML%	Mostly lost: Percentage of GT trajectories which are covered by tracker output for less than 20% in length. *The smaller the better.*
PT%	Partially tracked: 1.0-MT-ML.
Frag	Fragments: The total of No. of times that a groundtruth trajectory is interrupted in tracking result. *The smaller the better.*
IDS	ID switches: The total of No. of times that a tracked trajectory changes its matched GT identity. *The smaller the better.*

Table 4. Evaluation metrics.

Choice of β. In the hybrid loss function in equation (8), β is used to adjust the relative weights of the ranking part and the classification part. We tested trackers trained with different β, as shown in Figure 5. The best results are achieved by HybridBoost with β around 0.5 to 0.75, which outperforms the pure AdaBoost ($\beta = 0$) or pure RankBoost ($\beta = 1$).

5.4. Tracking performance

We report our tracking performance on 20 videos from the CAVIAR set[2], and 9 videos from the TRECVID08 set, which are from three different scenes, each of 5000 frames in length and different from the training data.

Table 5 shows the comparison among our approach, the hierarchical Hungarian algorithm based approach of [8][3],

[2]Results of [2] and [6] are reported on these 20 videos in [6].

[3]Results obtained by courtesy of authors of [8]. The high-level associa-

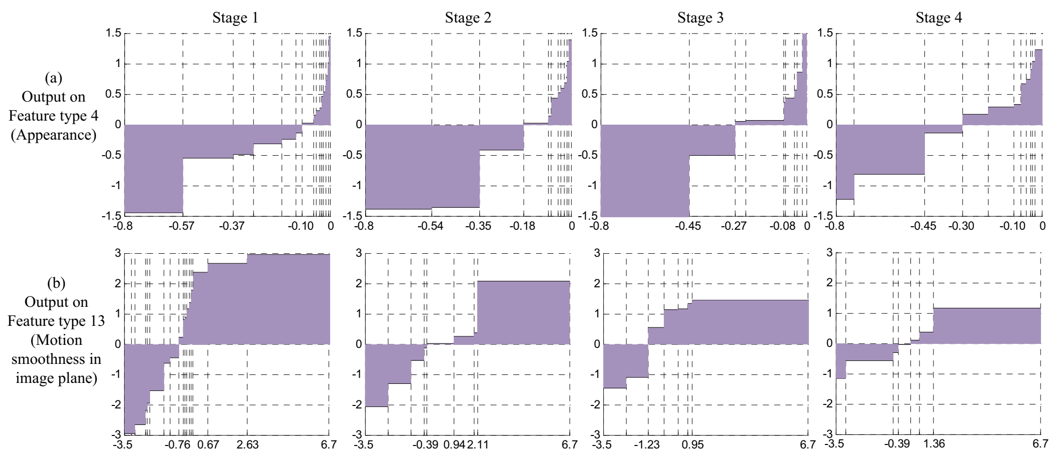

Figure 4. Comparison of $H_k(x)$'s output on one single feature in different stages. Here we show two of the most used features: χ^2 distance of color histogram and motion smoothness in image plane. The horizontal axis is feature value, the vertical axis is the output.

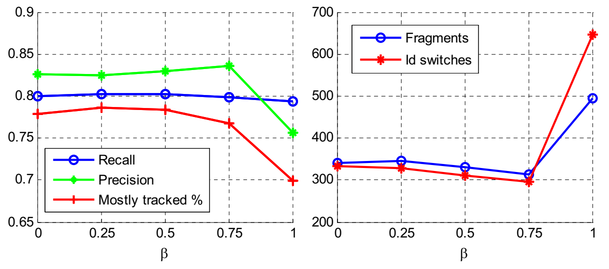

Figure 5. Tracking performance on TRECVID08 test set, using trackers trained with $\beta = 0$ (pure AdaBoost), 0.25, 0.5, 0.75 and 1 (pure RankBoost).

Method	Recall	FA/Frm	GT	MT	PT	ML	Frag	IDS
Wu *et al.* [2]	75.2%	0.281	140	75.7%	17.9%	6.4%	35*	17*
Zhang *et al.* [6]	76.4%	0.105	140	85.7%	10.7%	3.6%	20*	15*
Huang *et al.* [8][3]	86.3%	0.186	143	78.3%	14.7%	7.0%	54	12
Ours	89.0%	0.157	143	84.6%	14.0%	1.4%	17	11

Table 5. Results on subset of CAVIAR (20 videos selected by [6]). Numbers in the first two row are from [6]. Our GT trajectory number is higher than that used in [6] possibly because our minimal allowed GT object size is larger. * Fragment and ID switch numbers in [2] and [6] are evaluated using different metric definition from ours; our definition is more strict, see Section 5.2.

Method	Recall	Precision	GT	MT	PT	ML	Frag	IDS
Huang *et al.* [8][3]	71.6%	80.8%	919	57.0%	28.1%	14.9%	487	278
Ours	80.0%	83.5%	919	77.5%	17.6%	4.9%	310	288

Table 6. Results on TRECVID08 testing set.

the multi-hypothesis approach of [2] and the min-cost flow approach of [6] on the CAVIAR dataset. Among the four, our method achieves the best recall rate and the smallest numbers of fragments and ID switches. Our false alarm number per frame is 0.05 higher than the method of [6], but our recall rate is 12% higher. As for [8], which uses similar optimization framework as our approach, although its performance is already very good, our approach still reduces the mostly lost rate by 80% and improves the fragments by 69%, showing the effectiveness of the learned affinity model.

Since the CAVIAR set is relatively easy for our approach, we further test it on the much more challenging TRECVID08 dataset. This dataset features several airport scenes. They are crowded (10 to 30 people in each frame) and with very heavy inter-object occlusions and interactions. Table 6 shows the comparison between our approach and [8] on this test set. With only 10 more ID switches, our method gives 177 fewer fragments. Our recall rate is nearly 10% higher and the precision is also 3% higher. The mostly lost trajectories are reduced by 67% and the partially tracked trajectories reduced by nearly 40%.

tion part of their approach which involves the inference of scene occluders is excluded.

Some sample results are shown in Figure 6. (a) shows a man (pointed by a green arrow) walking against a crowd. He is correctly tracked through several instances of partial and full occlusion. Also, the tracker does not miss any person in the crowd. (b) features a scene with many small targets with different motion patterns: some are standing or lingering, others are walking across the hall; the tracker succeeds in tracking the targets through frequent inter-object occlusion (yellow and green arrows point to two examples). (c) compares the result of our tracker and that of [8]. The two targets are consistently tracked by our method while the other method experienced several instances of ID switches and target lost.

6. Conclusion and future work

We proposed a HybridBoost algorithm to learn the affinity model as a joint problem of ranking and classification. The affinity model is integrated in a hierarchical data association framework to track multiple targets in very crowded scenes. Promising results are shown on challenging data sets, and comparison with existing approaches is given. Further improvement can be achieved in several directions.

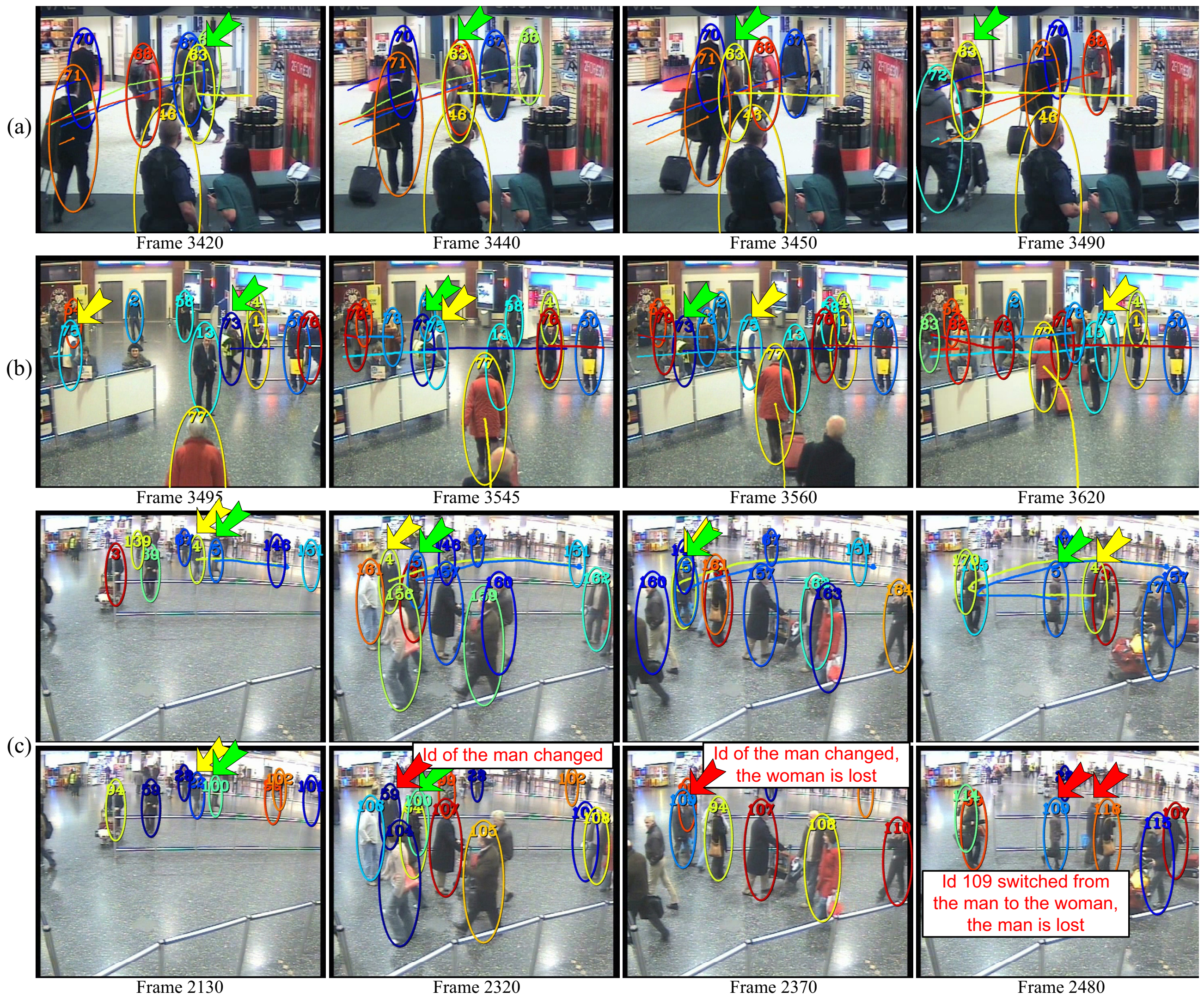

Figure 6. Sample results on TRECVID08 test videos. (a), (b) and (c) are from different scenes. The first row of (c) shows result of our approach and the second row shows result of the approach in [8].

More affinity features can be added, such as similarity of part appearance in addition to overall appearance. The learning algorithm should be able to learn the most important parts and assign proper weights to them. Also, other forms of weak ranking classifier can be explored to enhance to discriminative power of the strong ranking classifier.

Acknowledgements This research was sponsored by the U.S. Government in part by the VACE program and in part by the Office of Naval Research under Contract #N00014-06-1-0470.

References

[1] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2005.

[2] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *CVPR*, 2006.

[3] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR*, 2006.

[4] H. Jiang, S. Fels, and J. J. Little. A linear programming approach for multiple object tracking. In *CVPR*, 2007.

[5] B. Leibe, K. Schindler, and L. V. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007.

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

[7] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR*, 2006.

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

[9] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 2003.

[10] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 1999.

[11] D. Reid. An algorithm for tracking multiple targets. *IEEE Transaction on Automatic Control*, 24(6):843–854, December 1979.

[12] Y. Bar-Shalom, T. Fortmann, and M. Scheffe. Joint probabilistic data association for multiple targets in clutter. In *Information Sciences and Systems*, 1980.

[13] Caviar dataset. http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/.

[14] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, 2006.

[15] R. Kasturi, D. Goldgof, V. Manohar, M. Boonstra, and V. Korzhova. Performance evaluation protocol for face, person and vehicle detection and tracking in video analysis and content extraction. In *Workshop on classification of events, activities and relationships*, 2006.