Multi-Target Tracking by On-Line Learned Discriminative Appearance Models

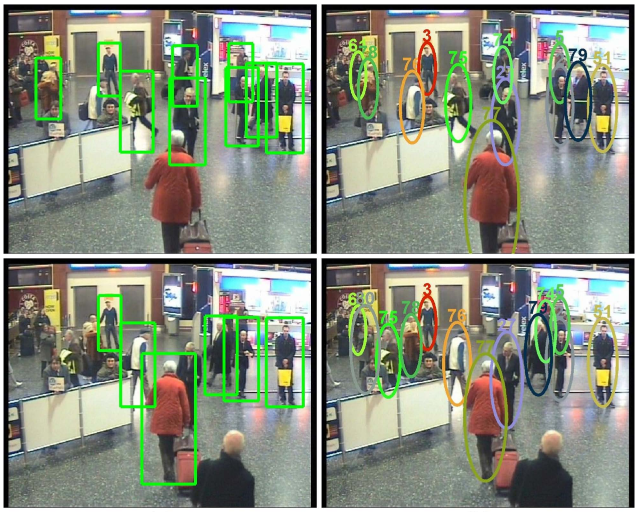
Cheng-Hao Kuo, Chang Huang, and Ramakant Nevatia
University of Southern California, Institute for Robotics and Intelligent Systems
Los Angeles, CA 90089, USA
{chenghak|huangcha|nevatia}@usc.edu

Abstract

We present an approach for online learning of discriminative appearance models for robust multi-target tracking in a crowded scene from a single camera. Although much progress has been made in developing methods for optimal data association, there has been comparatively less work on the appearance models, which are key elements for good performance. Many previous methods either use simple features such as color histograms, or focus on the discriminability between a target and the background which does not resolve ambiguities between the different targets. We propose an algorithm for learning a discriminative appearance model for different targets. Training samples are collected online from tracklets within a time sliding window based on some spatial-temporal constraints; this allows the models to adapt to target instances. Learning uses an AdaBoost algorithm that combines effective image descriptors and their corresponding similarity measurements. We term the learned models as OLDAMs. Our evaluations indicate that OLDAMs have significantly higher discrimination between different targets than conventional holistic color histograms, and when integrated into a hierarchical association framework, they help improve the tracking accuracy, particularly reducing the false alarms and identity switches.

1. Introduction

Multi-target tracking is important for many applications such as surveillance and human-computer interaction systems. Its aim is to locate the targets, retrieve their trajectories, and maintain their identities through a video sequence; this is a highly challenging problem in crowded environments when the occlusions of targets are frequent. In particular, similar appearance and complicated interactions between different targets often result in incorrect tracking results such as track fragmentation and identity switches. Figure 1 features a busy airport terminal [2] which is a challenging case for multi-target tracking.



(a) (b)

Figure 1. Sample detection results in column (a) and sample tracking results in column (b).

Detection-based tracking methods have become popular due to recent improvements in object detection performance. These methods integrate several cues such as appearance, motion, size, and location into an affinity model to measure similarity between detection responses or between tracklets in an association optimization framework. While many algorithms have been proposed for the association framework, there has been relatively less effort addressed to develop improved appearance models. Many previous methods simply compute the distance between two holistic color histograms for consistency measurement.

A recent paper, [15], shows impressive tracking results on difficult datasets. It uses a hierarchical association framework to progressively link tracklets into longer ones to form the final tracking result. At each stage, given the tracklet set provided by previous stage, the tracklet association is formulated as a MAP problem, which is solved by Hungarian algorithm with the link probabilities between tracklets from previous stage. In [15], this link probability is defined by an affinity model comprising three affinity terms for motion, time and appearance respectively:

$$P_{link}(T_j|T_i) = A_m(T_j|T_i)A_t(T_j|T_i)A_a(T_j|T_i) \qquad (1)$$

each of them measures the likelihood of tracklet T_i and T_j belonging to the same target according to their own features. We follow this formulation but with the goal of replacing the appearance probability with a more significantly discriminative model.

We propose an algorithm for online learning of discriminative appearance models; resulting models are called OLDAMs. They are designed to give a high affinity score for the tracklets which belong to the same target and low score for the tracklets which belong to different targets. Online learning is more effective than off-line learning for this task as it is specifically designed for the targets present in the current scene. Given short but reliable tracklets in a time sliding window, spatial-temporal constraints are applied to collect positive and negative samples. Several image descriptors at multiple locations and the corresponding similarity measurements are computed as features, which are combined into OLDAMs by AdaBoost algorithm. We compare the discriminative ability of OLDAMs to color histograms. We also integrated them into a hierarchical association framework, similar to that in [15]. The block diagram of our proposed system is shown in Figure 2. The usage of OLDAMs shows considerable improvements in tracking performance on CAVIAR and TRECVID08 data sets, particularly in metrics of false alarms and identity switches.

Notice that although the learning approach proposed by [18] and OLDAMs are both designed to improve the tracklet affinity model, the two methods focus on different aspects: [18] aims at off-line learning of a general affinity model that combines multiple types of cues such as tracklet length, motion, and time while using conventional appearance descriptors such as color histograms; OLDAMs are designed to discriminate specifically among the targets in the current time window according to their appearance and update when the targets change. Thus, both approaches are complementary; OLDAMs can be incorporated into the approach of [18] in future work.

The rest of the paper is organized as follows. Related work is discussed in Section 2. The overview of our approach is given in Section 3. Our framework for learning a discriminative appearance model is presented in Section 4. The experimental results are shown in Section 5. The conclusion is given in Section 6.

2. Related Work

Tracking multiple objects has been an active research topic in computer vision. There are two main streams in detection-based tracking methods: one considers only past and current frames to make association decisions [7, 8, 20, 25, 27]; the other takes information from future frames also [3, 5, 15, 16, 18, 21, 26, 28]. The former usually adopts a particle filtering framework and uses detection responses to guide the tracker. It is suitable for time-critical applications since no clues from future are required; however, it may be prone to yield identity switches and trajectory fragments due to noisy observation and long occlusion of targets; The latter method considers both past frames and future frames, and then performs a global optimization which is more likely to give improved results.

There has been relatively little attention given to development of discriminative appearance models among different targets. [15, 18, 25, 26, 28] use only a color histogram as their appearance model with different affinity measures such as the χ^2 distance, Bhattacharyya coefficient, and correlation coefficient. To enhance the appearance model for tracking, several methods [4, 7, 9, 13] obtain dynamic feature selection or the target observation model by online learning techniques. Several features, *e.g.*, Haar-like features [24], SIFT-like features [12, 19], orientation histograms [10, 17], spatiograms [6], are integrated in this framework. However, the appearance models in these methods are aimed at making the targets distinguishable from their neighborhood in the background, rather than from each other.

3. Overview of our approach

There are two main components in our approach: one is the strategy for online sample collection, the other is the appearance model learning method.

Sample collection strategy is based on examining spatial-temporal relations between tracklets in a time window. We rely on a dual-threshold association method [15] that is conservative and generally provides tracklets that correctly correspond to a single object. Positive samples are collected by extracting pairs of different detection responses within the same tracklet; negative samples are collected by extracting pairs of detection responses from tracklets that can not belong to the same target based on their spatial-temporal relationships.

Model learning problem is transformed into a binary classification problem: determine whether two tracklets belong to the same target or not according to their appearance descriptors. For each detection response, appearance descriptors consisting of the color histogram, the covariance matrix, and the HOG feature, are computed at multiple locations. Similarity measurements among the training samples establish the feature pool. The AdaBoost algorithm is adopted to select discriminative features from this pool and combine them into a strong classifier; the prediction confidence output by this classifier is transformed to a probability, which cooperates with other cues (*e.g.* motion and time) to compute the link probability between tracklets for their

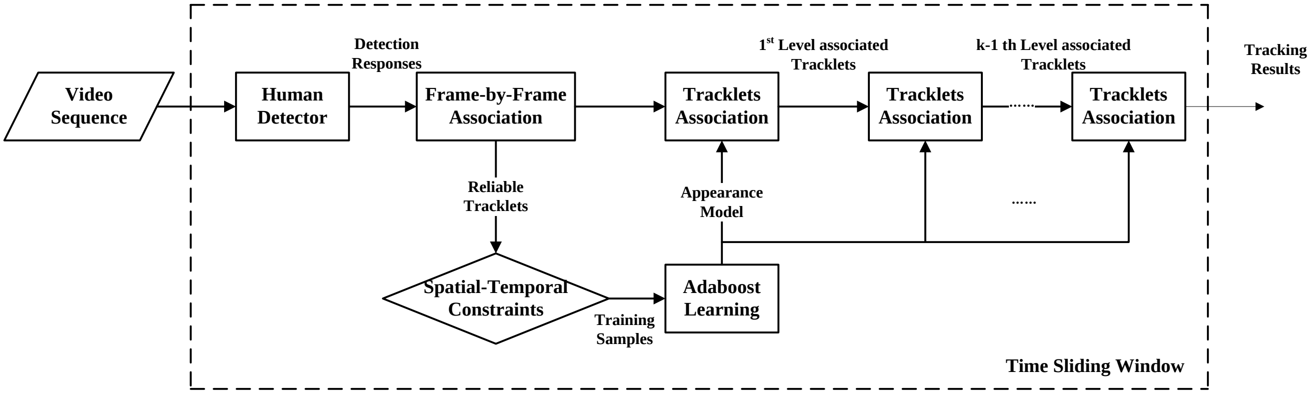


Figure 2. The block diagram of our multi-object tracking system.

association. In this way, the OLDAMs are capable of effectively distinguishing different targets within the current sliding window, and automatically adapting to new targets and varying environments in the coming sliding windows.

4. Online Learned Discriminative Appearance Models(OLDAMs)

The learning of OLDAMs involves four parts: samples collection, descriptor extraction, similarity measurement, and the learning algorithm. Before elaborating them, we briefly describe the dual-threshold method used to generates the reliable tracklets.

4.1. Reliable Tracklets

Given the detection responses, a dual-threshold strategy is used to generate short but reliable tracklets as in [15]. Note that the association here is only between two consecutive frames. The affinity of two responses is defined as the product of three measurements based on their position, size, and color histogram. Given all detection responses in two neighboring frames, a matching score matrix S can be formed. Two responses r_i and r_j are linked if their affinity score $S(i,j)$ is higher than a threshold θ_1 and exceeds any other elements in the i-th row and j-th column of S by another threshold θ_2. This strategy is conservative and biased to link only reliable associations.

4.2. Collecting training samples

We propose a method to collect positive and negative training samples using spatial-temporal constraints. Based on the tracklets generated by the dual-threshold strategy, we make two assumptions: 1) responses in one tracklet describe the same target. 2) any responses in two different tracklets which overlap in time represent different targets. The first one results from observing that the tracklets are reliable; the second one is based on the observation that one object can not belong to two different trajectories at the same time. Additionally, we denote certain tracklets as not being associable based on their spatial relations; if the frame gap between two tracklets is small but they are spatially separated, we consider them to belong to different targets based on the observation that tracked objects have limited velocities.

Figure 3 illustrates the process of collecting training samples. Some sample detection results, in a window, are given in Figure 3(a). Associations generated by dual-threshold strategy are made between the consecutive frames as in Figure 3(b). From those associated tracklets, we can use the spatial-temporal constraints to collect training samples. For example, T_3 is an associated tracklet so that the link between any different responses in T_3, *e.g.* r_1 and r_3, are labeled as positive samples. On the other hand, T_1 and T_2 overlap in time so that the link between any response in T_1 and T_2, *e.g.* r_1 and r_8, are labeled as negative samples. Besides, T_1 and T_6 are too far in the spatial domain so that the link between any responses in T_1 and T_6, *e.g.* r_4 and r_{12}, are labeled as negative samples as well. Based on these training samples, the learned appearance model which has discriminative power between T_8 and T_9 is able to prevent the wrong link between T_3 and T_5; this happens when target of T_3 is occluded by target of T_5 for a while.

In our implementation, a discriminative set is formed by the negative constraints. For a certain tracklet T_j, each element in the discriminative set $\mathcal{D}_j$ indicates a different target from T_j by spatial-temporal information. For example, $\mathcal{D}_1 = \{T_2, T_3, T_5, T_6\}$. Therefore, we can extract any two different responses from one tracklet as a positive training sample and two responses from two tracklets which belong to different targets as a negative training sample. We can define the instance space to be $\mathcal{X} = \mathcal{R} \times \mathcal{R}$, where $\mathcal{R}$ is the set of detection responses in tracklets. The sample set

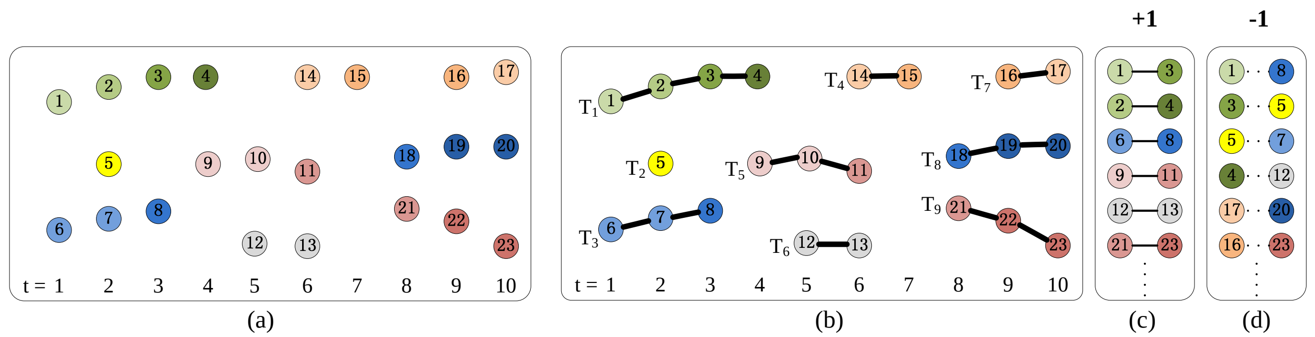


Figure 3. The overview of the process of obtaining on-line training samples. (a): The raw detection responses. (b) The result of reliable tracklets. (c) Positive training samples. (d) Negative training samples.

$\mathcal{B} = \mathcal{B}^+ \cup \mathcal{B}^-$ can be denoted by

$$\begin{aligned}\mathcal{B}^+ =& \{x_i : (r_m, r_n), y_i : +1 | r_m, r_n \in T_j\} \\ \mathcal{B}^- =& \{x_i : (r_m, r_n), y_i : -1 | r_m \in T_j, r_n \in T_k, \\ & T_j \in \mathcal{D}_k \text{ or } T_k \in \mathcal{D}_j\}\end{aligned} \tag{2}$$

where $x \in \mathcal{X}$, $r_m, r_n \in \mathcal{R}$, and $m \neq n$.

4.3. Representation of appearance model

To build a strong appearance model, we begin by computing several local features to describe a tracked target. In our implementation, color histograms, covariance matrixes, and histogram of gradients (HOG) constitute the feature pool. Given a detection response r, each feature is evaluated at different locations and different scales to increase the descriptive ability.

We use standard color histograms to represent the color appearance of a local image patch. Histograms have the advantage of being easy to implement and having well studied similarity measures. We adopt RGB color space for simplicity. Single channel histograms are concatenated to form a single vector $\mathbf{f}_{RGB_i}$, but any other suitable color space can be used. In our implementation, we use 8 bins for each channel to form a 24-element vector.

To describe the image texture, we use a descriptor based on covariance matrices of image features proposed in [23]. It has been shown to give good performance for texture classification and object categorization. The texture descriptor $\mathbf{C}_R$ corresponds to the covariance matrix:

$$\mathbf{C}_R = \frac{1}{n-1} \sum_{k=1}^{n} (\mathbf{z}_k - \boldsymbol{\mu})(\mathbf{z}_k - \boldsymbol{\mu})^T \tag{3}$$

where

$$\mathbf{z}_k = \left[\frac{\partial I}{\partial x} \; \frac{\partial I}{\partial y} \; \frac{\partial^2 I}{\partial x^2} \; \frac{\partial^2 I}{\partial y^2} \; \frac{\partial^2 I}{\partial xy} \right]^T \tag{4}$$

is the vector containing first and second derivatives of image at k-th pixel in the Region R, $\boldsymbol{\mu}$ is the mean vector over R, and n is the number of pixels.

To capture shape information, we choose the Histogram of Gradients (HOG) Feature proposed in [10]. In our design, a 32D HOG feature $\mathbf{f}_{HOG_i}$ is extracted over the region R; it is formed by concatenating 8 orientations bins in 2×2 cells over R.

In summary, the appearance descriptor of a tracklet T_i can be written as:

$$\mathcal{A}_i = (\{\mathbf{f}^l_{RGB_i}\}, \{\mathbf{C}^l_i\}, \{\mathbf{f}^l_{HOG_i}\}) \tag{5}$$

where $f^l_{RGB_i}$ is the feature vector for color histogram, $\mathbf{C}^l_i$ is the covariance matrix, and $f^l_{HOG_i}$ is the 32D HOG feature vector. The superscript l means that the features are evaluated at region R^l. In our design, we choose the number of regions to be 15 so that the feature pool contains 45 cues in total.

4.4. Similarity of appearance descriptors

Given the appearance descriptors explained above, we can compute similarity between two patches. The color histogram and HOG feature are histogram-based features so standard measurements, such as χ^2 distance, Bhattacharyya distance, and correlation coefficient can be used. In our implementation, correlation coefficient is chosen for simplicity. We denote the similarity of these two descriptors as $\rho(\mathbf{f}_{RGB_i}, \mathbf{f}_{RGB_j})$ and $\rho(\mathbf{f}_{HOG_i}, \mathbf{f}_{HOG_j})$.

The distance measurement of covariance matrices is described in [23]:

$$\sigma(\mathbf{C}_i, \mathbf{C}_j) = \sqrt{\sum_{k=1}^{5} ln^2 \lambda_k(\mathbf{C}_i, \mathbf{C}_j)} \tag{6}$$

where $\{\lambda_k(\mathbf{C}_i, \mathbf{C}_j)\}$ are the generalized eigenvalues of C_i and C_j, computed from

$$\lambda_k \mathbf{C}_i \mathbf{x}_k - \mathbf{C}_j \mathbf{x}_k = 0 \qquad k = 1 \dots 5 \tag{7}$$

and $\mathbf{x}_k \neq 0$ are generalized eigenvectors.

After computing the appearance model and the similarity between appearance descriptors at different regions, we form a feature vector

$$\begin{aligned}\mathbf{h}(\mathcal{A}_i,\mathcal{A}_j) = [&\rho(\mathbf{f}^1_{RGB_i},\mathbf{f}^1_{RGB_j}),\ldots,\rho(\mathbf{f}^L_{RGB_i},\mathbf{f}^L_{RGB_j}),\\ &\sigma(\mathbf{C}^1_i,\mathbf{C}^1_j),\ldots,\sigma(\mathbf{C}^L_i,\mathbf{C}^L_j),\\ &\rho(\mathbf{f}^1_{HOG_i},\mathbf{f}^1_{HOG_j}),\ldots,\rho(\mathbf{f}^L_{HOG_i},\mathbf{f}^L_{HOG_j})]\end{aligned} \tag{8}$$

by concatenating the similarity measurements with different appearance descriptors at multiple locations. This feature vector gives us a feature pool so that we can use Adaboost algorithm to combine those cues into a strong classifier.

4.5. Learning Algorithm

Our goal is to design a strong model which determines the affinity score of appearance between two instances. It takes a pair of instances as input and returns a real value to distinguish positive pairs from negative pairs. The larger the $H(\mathcal{A}_i,\mathcal{A}_j)$ is, the more likely that $\mathcal{A}_i$ and $\mathcal{A}_j$ represent the same target. We adopt the learning framework of binary classification and transfer the confidence score into the probability space. The affinity model is designed to be a linear combination of the similarity measurements computed in (8). We choose Adaboost algorithm [11, 22] to learn the coefficients. The strong classifier takes the following form:

$$H(\mathcal{A}_i,\mathcal{A}_j) = \sum_{t=1}^{T} \alpha_t h_t(\mathcal{A}_i,\mathcal{A}_j) \tag{9}$$

In our framework, the weak hypothesis is from the feature pool obtained by (8). We adjust the sign and normalize $h(x)$ to be in the restricted range $[-1,+1]$. The sign of $h(x)$ is interpreted as the predicted label and the magnitude $|h(x)|$ as the confidence in this prediction.

The loss function for Adaboost algorithm is defined as:

$$Z = \sum_i w_i^0 exp\big(-y_i H(x_i)\big) \tag{10}$$

where w^0 is the initial weight for each training sample, which will be updated during boosting. Our goal is to find $H(x)$ which minimizes Z, where $H(x)$ is obtained by sequentially adding new weak classifiers. In the t-th round, we aim at learning the optimal (h_t,α_t) to minimize the loss

$$Z_t = \sum_i w_i^t exp\big(-\alpha_t y_i h_t(x_i)\big) \tag{11}$$

The algorithm proposed in [22] is adopted to find α_t in an analytical form. We then update the sample weights according to h_t and α_t to focus on the misclassified samples. The learning procedure is summarized in Algorithm 1.

Algorithm 1 Learning the appearance model

```
Input:  $\mathcal{B}^+$ $=\{(x_i,+1)\}$:   Positive samples
        $\mathcal{B}^-$ $=\{(x_i,-1)\}$:   Negative samples
        $\mathcal{F}$   $=\{\mathbf{h}(x_i)\}$:   Feature pools

Set $w_i = \frac{1}{2|\mathcal{B}^+|}$ if $x_i \in \mathcal{B}^+$, $w_i = \frac{1}{2|\mathcal{B}^-|}$ if $x_i \in \mathcal{B}^-$
for $t = 1$ to $T$ do
    for $k = 1$ to $K$ do
        $r = \sum_i w_i y_i h_k(x_i)$
        $\alpha_k = \frac{1}{2} ln(\frac{1+r}{1-r})$
    end for
    Choose $k^* = \arg\min_k \sum_i w_i exp[-\alpha_k y_i h_k(x_i)]$
    Set $\alpha_t = \alpha_{k^*}$ and $h_t = h_{k^*}$
    Update $w_i \leftarrow w_i exp[-\alpha_t y_i h_t(x_i)]$
    Normalize $w_i$
end for

Output: $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$
```

5. Experimental results

We evaluate the effectiveness of OLDAMs incorporated in a hierarchical tracking algorithm applied to two public surveillance datasets: the CAVIAR dataset [1] and the TRECVID08 [2] dataset. Performance of our method is compared with several state-of-art methods and with other appearance models using our implementations; we also provide some graphical examples.

5.1. Discrimination Comparison

We first evaluate the discriminative power of OLDAMs, independent of the tracking system that they may be embedded in. For each tracklet pair in a given temporal sliding window, the affinity score based on OLDAMs and correlation coefficient of color histogram are computed. We manually label which tracklet pairs should be associated to form the ground truth. A distribution of scores is displayed in Figure 4; it is generated by 192 positive pairs and 2,176 negative pairs of detection responses extracted from a window of 200 frames. The horizontal axis represents the affinity scores and vertical axis denotes the density distribution. The Bhattacharyya distance between negative samples and positive samples is 0.284 by correlation coefficient of color histogram and 0.689 by OLDAMs. The equal error rate is 0.265 by correlation coefficient of color histogram and 0.125 by OLDAMs. It can be seen that the OLDAMs are significantly more discriminative than color histograms as the positive and negative samples are separated better.

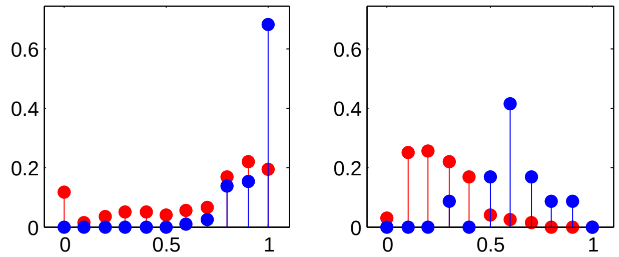


Figure 4. The sample distribution based on correlation coefficients of color histogram (left) and OLDAMs (right). Blue represents positive samples and red represents negative ones. The figure gives an example to show that OLDAMs are more discriminative than color histogram.

5.2. Evaluation metrics

We adopt the commonly used metrics [25] which include the evaluation of performance both in tracking and detection. Note that for the track fragments and ID switches, we follow the definition proposed in [18]; it is more strict but better-defined than the definition in [25]. The metrics in tracking evaluation are:

• Ground Truth(GT): The number of trajectories in the ground truth.

• Mostly tracked trajectories(MT), the percentage of trajectories that are successfully tracked for more than 80% divided by GT.

• Partially tracked trajectories(PT), the percentage of trajectories that are tracked between 20% and 80% divided by GT.

• Mostly lost trajectories(ML), the percentage of trajectories that are tracked for less than 20% divided by GT.

• Fragments(Frag): The total number of times that a trajectory in ground truth is interrupted by the tracking results.

• ID switches(IDS): The total number of times that a tracked trajectory changes its matched GT identity.

Since multi-object tracking can be viewed as a method which is able to recover missed detections and remove false alarms from the raw detection responses, we also provide the metrics for detection evaluation:

• Recall: The number of correctly matched detections divided by the total number of detections in ground truth.

• Precision: The number of correctly matched detections divided by the number of output detections.

• False Alarm per Frame(FAF): The number of false alarms per frame.

The higher value, the better is the performance in MT, recall, and precision; The lower value, the better is the performance in ML, Frag, IDS, and FAF.

5.3. Results for the CAVIAR dataset

The CAVIAR dataset contains 26 video sequences of a corridor in a shopping center taken by a single camera with frame size of 384×288 and frame rate of 25fps. We use the method of [14] as our pedestrian detector. For comparison with the state-of-arts, we conduct our experiment on the 20 videos selected by [28]. Tracking evaluation results are presented in Table 1. OLDAM's results have the best recall rate, the smallest False Alarm per Frame, and the second best mostly tracked trajectories. The number of fragments and ID switches in our approach are lower than most previous methods and competitive with [18]. Some tracking results are shown in Figure 5. The two targets are consistently tracked by our method while they exchange their IDs in other methods.

5.4. Results of TRECVID08 dataset

The CAVIAR dataset is relatively easy and several methods achieve good results. To test the effectiveness of the OLDAM approach, we further evaluate our method on a much more challenging TRECVID 2008 event detection [2] dataset which consists of videos captured in a major airport.

As in [18], the experiment is conducted on 9 videos with frame size of 720×576 chosen from the TRECVID08 set, which contains three different scenes, each being 5000 frames in length. This dataset has a high crowd density and inter-object occlusions and interactions occur often; thus, stronger appearance models should be of greater value. Table 2 shows comparison between the OLDAM approach and [15, 18] on this dataset. Our result achieves the best recall rate and precision rate. With similar mostly tracked trajectories and fragments compared to [18], we have significant reduction in ID switches. It shows that on-line learned discriminative appearance model prevents wrong association between different targets which a simple appearance model fails to do. Notice that [18] and our approach focus on different aspects as mentioned in introduction, thus they are complementary works; OLDAMs can be incorporated into the approach of [18] in future work. Some tracking results are shown in Figure 6.

We also compare our approach with different appearance models using our own implementation. The results are also shown in Table 2. Ours (a) represents the result that only color histogram is used as the appearance model. In the result of ours (b), our proposed appearance model is used but learned in an off-line environment, which means the coefficients α_t are fixed. It shows that our proposed method outperforms these two appearance models. This comparison justifies that our stronger appearance model with online learning improves the tracking performance.

5.5. Computational Speed

We measure the execution time using OLDAMs on 20 videos in the evaluation from CAVIAR dataset, which typically has 2 to 8 pedestrians to be tracked in each frame. The tracking speed of our system is about 4 fps, on a 3.0GHz

Method	Recall	Precision	FAF	GT	MT	PT	ML	Frag	IDS
Wu *et al.* [25]	75.2%	-	0.281	140	75.7%	17.9%	6.4%	35*	17*
Zhang *et al.* [28]	76.4%	-	0.105	140	85.7%	10.7%	3.6%	20*	15*
Xing *et al.* [26]	81.8%	-	0.136	140	84.3%	12.1%	3.6%	24*	14*
Huang *et al.* [15]	86.3%	-	0.186	143	78.3%	14.7%	7.0%	54	12
Li *et al.* [18]	89.0%	-	0.157	143	84.6%	14.0%	1.4%	17	11
OLDAM	89.4%	96.9%	0.085	143	84.6%	14.7%	0.7%	18	11

Table 1. Tracking results on CAVIAR dataset. *The numbers of Frag and IDS in [25] [28] [26] are obtained by different metrics from what we adopt [18], which is more strict.

Method	Recall	Precision	FAF	GT	MT	PT	ML	Frag	IDS
Huang *et al.* [15]	71.6%	80.8%	-	919	57.0%	28.1%	14.9%	487	278
Li *et al.* [18]	80.0%	83.5%	-	919	77.5%	17.6%	4.9%	310	288
Ours (a)	77.0%	85.3%	1.015	919	71.6%	23.1%	5.4%	496	303
Ours (b)	80.5%	83.9%	1.181	919	77.8%	18.7%	3.4%	475	286
OLDAM	80.4%	86.1%	0.992	919	76.1%	19.3%	4.6%	322	224

Table 2. Tracking results on TRECVID08 dataset.

PC with the program being coded in Matlab; this does not count the processing time of the human detection step. In fact, most of the processing time is spent in extracting appearance descriptors from the videos, which are shared by the online learning and the tracklets association. We also test our implementation using the off-line learned appearance model on the same videos. By removing the online learning, the execution time is reduced by 17%. This indicates that the online learning does not significantly increase the computational load of the tracking system.

6. Conclusion

We present an approach of online learning a discriminative appearance model for robust multi-target tracking. Unlike previous methods, our model is designed to distinguish different targets from each other, rather than from the background. Spatial-temporal constraints are used to select training samples automatically at runtime. Experiments on challenging datasets show clear improvements by our proposed OLDAMs.

Acknowledgements

This paper is based upon work supported in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under contract number W911NF-08-C-0068.

References

[1] Caviar dataset. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/. 5

[2] National institute of standards and technology: Trecvid 2008 evaluation for surveillance event detection. http://www.nist.gov/speech/tests/trecvid/2008/. 1, 5, 6

[3] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR 2008*. 2

[4] S. Avidan. Ensemble tracking. In *CVPR 2005*. 2

[5] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR 2006*. 2

[6] S. T. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *CVPR 2005*. 2

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

[8] Y. Cai, N. de Freitas, and J. J. Little. Robust visual tracking for multiple targets. In *ECCV 2006*. 2

[9] R. T. Collins and Y. Liu. On-line selection of discriminative tracking features. In *ICCV 2003*. 2

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR 2005*. 2, 4

[11] J. Friedman, T. Hastie, and R. Tibshirani. Additive Logistic Regression: a Statistical View of Boosting. *The Annals of Statistics*, 38(2), 2000. 5

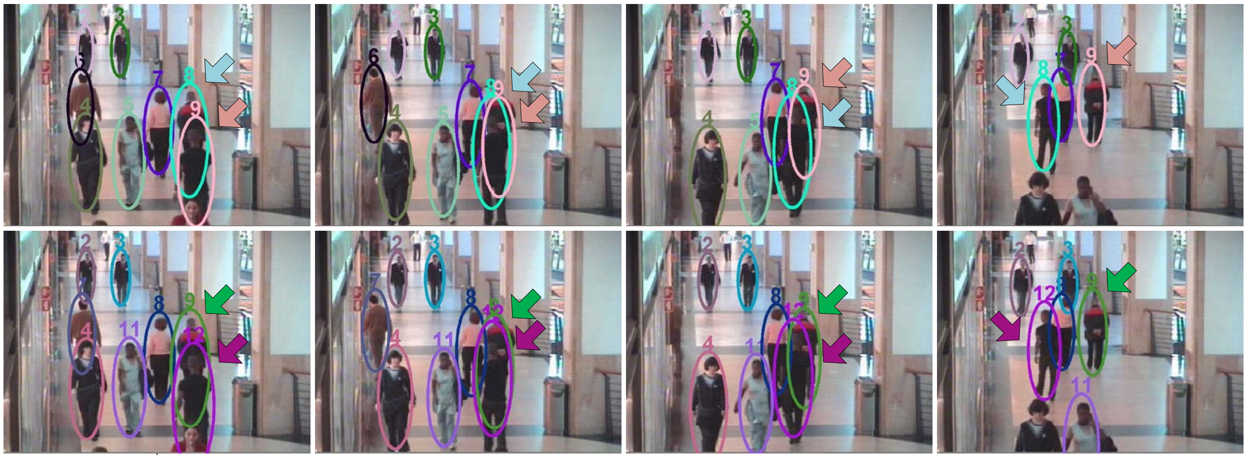
[12] B. Fulkerson, A. Vedaldi, and S. Soatto. Localizing objects with smart dictionaries. In *ECCV 2008*. 2

[13] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR 2006*. 2

[14] C. Huang and R. Nevatia. High performance object detection by collaborative learning of joint ranking of granule features. In *CVPR 2010*. 6

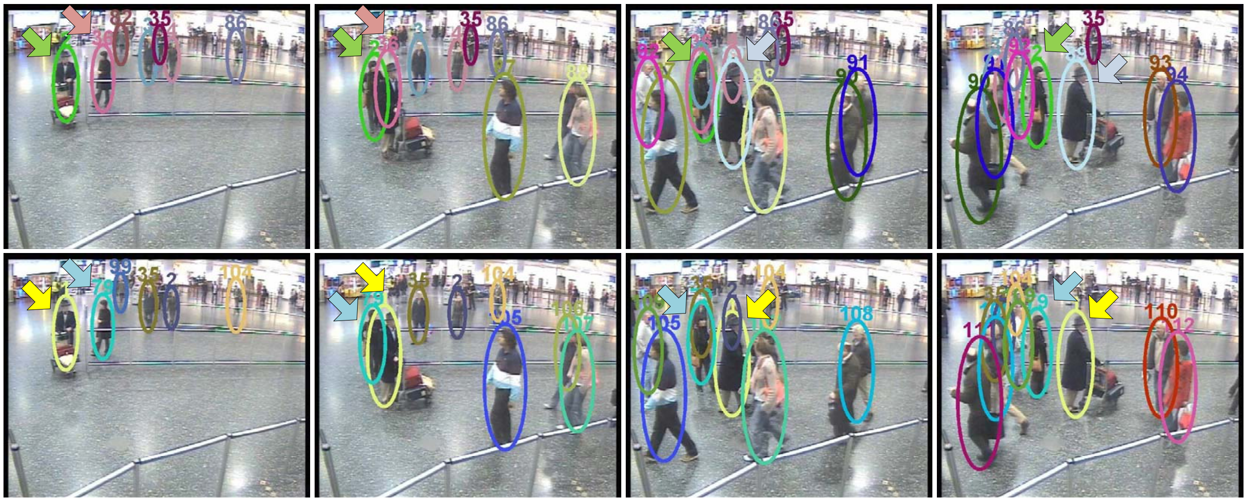
[15] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV 2008*. 1, 2, 3, 6, 7

[16] B. Leibe, K. Schindler, and L. V. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV 2007*. 2



Frame 399 Frame 424 Frame 439 Frame 519

Figure 5. Sample tracking result on CAVIAR dataset. Top row: Two targets indicated by arrows switch their IDs where color histogram serves the appearance model. Bottom row: These two targets are tracked successfully as the OLDAMs are used.



Frame 2170 Frame 2250 Frame 2300 Frame 2360

Figure 6. Sample tracking result on TRECVID08 dataset. The top row shows the result of [18] that a man has a new ID and his old ID is transferred to the lady behind. The bottom row shows that they are consistently tracked in our method.

[17] K. Levi and Y. Weiss. Learning object detection from a small number of examples: the importance of good features. In *CVPR 2004*. 2

[18] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR 2009*. 2, 6, 7, 8

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, November 2004. 2

[20] K. Okuma, A. Taleghani, O. D. Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV 2004*. 2

[21] 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*. 2

[22] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Machine Learning*, pages 80–91, 1999. 5

[23] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV 2006*. 4

[24] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR 2001*. 2

[25] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *IJCV*, 75(2):247–266, November 2007. 2, 6, 7

[26] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *CVPR 2009*. 2, 7

[27] M. Yang, F. Lv, W. Xu, and Y. Gong. Detection driven adaptive multi-cue integration for multiple human tracking. In *ICCV 2009*. 2

[28] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR 2008*. 2, 6, 7