An Online Learned CRF Model for Multi-Target Tracking

Bo Yang and Ram Nevatia
Institute for Robotics and Intelligent Systems, University of Southern California
Los Angeles, CA 90089, USA
{yangbo|nevatia}@usc.edu

Abstract

We introduce an online learning approach for multi-target tracking. Detection responses are gradually associated into tracklets in multiple levels to produce final tracks. Unlike most previous approaches which only focus on producing discriminative motion and appearance models for all targets, we further consider discriminative features for distinguishing difficult pairs of targets. The tracking problem is formulated using an online learned CRF model, and is transformed into an energy minimization problem. The energy functions include a set of unary functions that are based on motion and appearance models for discriminating all targets, as well as a set of pairwise functions that are based on models for differentiating corresponding pairs of tracklets. The online CRF approach is more powerful at distinguishing spatially close targets with similar appearances, as well as in dealing with camera motions. An efficient algorithm is introduced for finding an association with low energy cost. We evaluate our approach on three public data sets, and show significant improvements compared with several state-of-art methods.

1. Introduction

Tracking multiple targets is an important but difficult problem in computer vision. It aims at finding trajectories of all targets while maintaining their identities. Due to great improvements on object detection, association based tracking approaches have been proposed [11, 17, 2, 12, 18]. They often find proper linking affinities based on multiple cues between detection responses or tracklets, i.e., track fragments, and find a global solution with maximum probability using Hungarian algorithm, MCMC, etc.

Association based approaches are powerful at dealing with extended occlusions between targets and the complexity is polynomial in the number of targets. However, how to better distinguish different targets remains a key issue that limits the performance of association based tracking. It is difficult to find descriptors to distinguish targets in crowded scenes with frequent occlusions and similar appearances. In this paper, we propose an online learned condition random field (CRF) model to better discriminating different targets, especially difficult pairs, which are spatially near targets with similar appearance. Figure 1 shows some tracking examples by our approach.

To identify each target, motion and appearance information are often adopted to produce discriminative descriptors. Motion descriptors are often based on speeds and distances between tracklet pairs, while appearance descriptors are often based on global or part based color histograms to distinguish different targets.

In most previous association based tracking work, appearance models are pre-defined [17, 12] or online learned to discriminate all targets [7, 13] or to discriminate one target with all others [3, 8]. Though such learned appearance models are able to distinguish most targets, they are not necessarily capable of differentiating difficult pairs, i.e., close targets with similar appearances. The discriminative features between a difficult pair are possibly quite different with those for distinguishing with all other targets.

Linear motion models are widely used in previous tracking work [15, 19, 3]; linking probabilities between tracklets are often based on how well a pair of tracklets satisfies a linear motion assumption. However, as shown in the first row of Figure 2, if the view angle changes due to camera
motion, the motion smoothness would be impaired; it could be
compensated by frame matching techniques, but this is a
challenging task by itself. Relative positions between tar-
ggets are less dependent on view angles, and are often more
stable than linear motion models for dealing with camera
motions. For static cameras, relative positions are still help-
ful, as shown in the second row of Figure 2; some targets
may not follow a linear motion model, and relative posi-
tions between neighboring targets are useful for recovering
errors in a linear motion model.

Based on above observations, we propose an online
learning approach, which formulates the multi-target track-
ing problem as inference in a conditional random field
(CRF) model as shown in Figure 3. Our CRF framework
incorporates global models for distinguishing all targets and
pairwise models for differentiating difficult pairs of targets.

All linkable tracklet pairs form the nodes in this CRF
model, and labels of each node (1 or 0) indicate whether
two tracklets can be linked or not. The energy cost for each
node is estimated based on global appearance and motion
models similar to [7]. The energy cost for an edge is based
on discriminative pairwise models, i.e., appearance and mo-
tion descriptors, that are online learned for distinguishing
tracklets in the connected two CRF nodes. Global models
and pairwise models are used to produce unary and pairwise
energy functions respectively, and the tracking problem is
transformed into an energy minimization task.

The contributions of this paper are:

- A CRF framework for modeling both global tracklets
  affinity models and pairwise discriminative models.
- An online learning approach for producing unary and
  pairwise energy functions in a CRF model.
- An approximation algorithm for efficiently finding
  good tracking solutions with low energy costs.

The rest of the paper is organized as follows: related
work is discussed in Section 2; problem formulation is
given in Section 3; Section 4 describes the online learn-
ing approach for a CRF model; experiments are shown in
Section 5, followed by conclusion in Section 6.

2. Related Work

Multi-target tracking has been an important topic in com-
puter vision for several years. One key issue in tracking is
that how to distinguish targets with backgrounds and with
each other.

Most visual tracking methods focus on tracking single
object or multiple objects separately [9, 16]; they usually
try to find proper appearance models that distinguish one object
with all other targets or backgrounds, and adopt meanshift
target appearance models, and use updated models to con-
tinuously track targets.

On the other hand, most association based methods focus
on tracking multiple objects of a pre-known class simulta-
neously [11, 14, 2, 18]. They usually associate detection re-
sponses produced by a pre-trained detector into long tracks,
and find a global optimal solution for all targets. Appearance
models are often pre-defined [17, 12] or online learned
to distinguish multiple targets globally [13, 7]; in addition,
linear motion models between tracklet pairs [15, 3] are often
adopted to constrain motion smoothness. Though such ap-
proaches may obtain global optimized appearance and motion
models, they are not necessarily able to differentiate
difficult pairs of targets, i.e., close ones with similar appear-
ances, as appearance models for distinguishing a specific
pair of targets may be quite different with those used for
distinguishing all targets, and previous motion models are
not stable for non-static cameras. However, our online CRF
models consider both global and pairwise discriminative ap-
pearance and motion models.

Note that CRF models are also adopted in [19]. Both
[19] and this approach relax the assumption that associa-
tions between tracklet pairs are independent of each other.
However, [19] focused on modeling association dependen-
cies, while this approach aims at better distinction between
difficult pairs of targets and therefore the meanings of edges
in CRF are different. In addition, [19] is an offline ap-
proach that integrates multiple cues on pre-labeled ground
truth data, but our approach is an online learning method
that finds discriminative models automatically without pre-
labeled data.

3. CRF Formulation for Tracking

Given a video input, we first detect targets in each frame
by a pre-trained detector. Similar to [7], we adopt a low
level association process to connect detection responses in
neighboring frames into reliable tracklets, and then asso-
ciate the tracklets progressively in multiple levels. A track-
let \( T_i = \{ d_i^t, \ldots, d_i^{t_f} \} \) is defined as a set of detection or in-
terpolated responses in consecutive frames, where \( t_i^t \) and \( t_i^{t_f} \)
Training samples for
global appearance
+1 -1 Video
Input

Unary Energy Functions
Energy Minimization

Figure 3. Tracking framework of our approach. In the CRF model, each node denotes a possible link between a tracklet pair and has a unary energy cost based on global appearance and motion models; each edge denotes a correlation between two nodes and has a pairwise energy cost based on discriminative appearance and motion models specifically for the two nodes. Colors of detection responses indicate their belonging tracklets. Best viewed in color.

denote the start and end frames of \( T_i \) and \( d_i^t = \{ p_i^t, s_i^t, v_i^t \} \) denote the response at frame \( t \), including position \( p_i^t \), size \( s_i^t \), and velocity vector \( v_i^t \).

At each level, the input is the set of tracklets produced in previous level \( S = \{ T_1, T_2, \ldots, T_n \} \). For each possible association pair of tracklet \( (T_{i_1} \rightarrow T_{i_2}) \), we introduce a label \( l_i \), where \( l_i = 1 \) indicates \( T_{i_2} \) is linked to \( T_{i_1} \), and \( l_i = 0 \) indicates the opposite. We aim to find the best set of associations with the highest probability. We formulate the tracking problem as finding the best \( L \) given \( S \)

\[
L^* = \arg \max_L P(L|S) = \arg \max_L \frac{1}{Z} \exp(-\Psi(L|S))
\]

where \( Z \) is a normalization factor, and \( \Psi \) is a cost function. Assuming that the joint distributions for more than two associations do not make contributions to \( P(L|S) \), we have

\[
L^* = \arg \min_L \sum_i U(l_i|S) + \sum_{ij} B(l_i, l_j|S)
\]

where \( U(l_i|S) = -\ln P(l_i|S) \) and \( B(l_i, l_j|S) = -\ln P(l_i, l_j|S) \) denote the unary and pairwise energy functions respectively. In Eqn. 2, the first part defines the linking probabilities between any two tracklets based on global appearance and motion models, while the second part defines the correlations between tracklet pairs based on discriminative models especially learned for corresponding pairs of tracklets.

We model the tracking problem by a Conditional Random Field (CRF) model. As shown in Figure 3, a graph \( G = (V, E) \) is created for each association level, where \( V = \{ v_1, \ldots, v_p \} \) denotes the set of nodes, and \( E = \{ e_1, \ldots, e_q \} \) denotes the set of edges. Each node \( v_i = (T_{i_1} \rightarrow T_{i_2}) \) denotes a possible association between tracklets \( T_{i_1} \) and \( T_{i_2} \); each edge \( e_j = \{ (v_{j_1}, v_{j_2}) \} \) denotes a correlation between two nodes. A label \( L = \{ l_1, \ldots, l_p \} \) on \( V \) denotes an association result for current level. We assume that one tracklet cannot be associated with more than one tracklet, and therefore any valid label set \( L \) should satisfy

\[
\sum_{v_i \in Head_{i_1}} l_i \leq 1 \quad \text{and} \quad \sum_{v_i \in Tail_{i_2}} l_i \leq 1
\]

where the first constraint limits any tracklet \( T_{i_1} \) link to at most one other tracklet, and the second constraint limits that at most one tracklet may be link to any tracklet \( T_{i_2} \).

For efficiency, we track in sliding windows one by one instead of processing the whole video at one time. The CRF models are learned individually in each sliding window.

4. Online Learning of CRF Models

In this section, we introduce our tracking approach in several steps, including CRF graph creation, online learn-
Figure 5. Examples of head close and tail close tracklet pairs.

Figure 6. Global motion models for unary terms in CRF.

4.1. CRF Graph Creation for Tracklets Association

Given a set of tracklets $S = \{T_1, T_2, \ldots, T_n\}$ as input, we want to create a CRF graph for modeling all possible associations between tracklets and their correlations. Tracklet $T_i$ is linkable to $T_j$ if the gap between the end of $T_i$ and the beginning of $T_j$ satisfies

$$0 < t_j^s - t_i^e < t_{\text{max}}$$

where $t_{\text{max}}$ is a threshold for maximum gap between any linkable pair of tracklets, and $t_j^s$ and $t_i^e$ denote the start and end frames of $T_j$ and $T_i$ respectively. We create the set of nodes $V$ in CRF to model all linkable tracklets as

$$V = \{v_i = (T_i \rightarrow T_j)\} \text{ s.t. } T_i \text{ is linkable to } T_j$$

Instead of modeling association dependencies as in [19], edges in our CRF provide corresponding pairwise models between spatially close targets, and are defined between any nodes that have tail close or head close tracklet pairs.

As shown in Figure 4, two tracklets $T_i$ and $T_j$ are a head close pair if they satisfy (suppose $t_i^e \geq t_j^s$)

$$t_i^e < t_j^s \text{ } \&\text{ } ||p_i^t - p_j^t|| < \gamma \min(s_i^t, s_j^t)$$

where $\gamma$ is a distance control factor, set to 3 in our experiments. This definition indicates that the head part of $T_i$ is close to $T_j$ at $T_i$’s beginning frame. The definition of tail close is similar.

Then we define the set of edges $E$ as

$$E = \{(v_i, v_j) \mid \forall v_i, v_j \in V \} \text{ s.t. } T_i \text{ and } T_j \text{ are tail close, or } T_i \text{ and } T_j \text{ are head close.}$$

Such definition constraints the edges on difficult pairs where wrong associations are most likely to happen, so that edges produce proper pairwise energies to distinguish them.

4.2. Learning of Unary Terms

Unary terms in Equ. 2 define the energy cost for associating pairs of tracklets. As defined in section 3, $U(l_i|S) = -\ln P(l_i|S)$. We further divide the probability into motion based probability $P_m(\cdot)$ and appearance based probability $P_a(\cdot)$ as

$$U(l_i = 1|S) = -\ln(P_m(T_i \rightarrow T_j|S)P_a(T_i \rightarrow T_j|S))$$

$P_m$ is defined as in [7, 19, 8], which is based on the distance between estimations of positions based on linear motion models and the real positions. As shown in Figure 5, the motion probability between tracklets $T_1$ and $T_2$ are defined based on $\Delta p_1 = p_{\text{head}} - v_{\text{head}}\Delta t - p_{\text{tail}}$ and $\Delta p_2 = p_{\text{tail}} + v_{\text{tail}}\Delta t - p_{\text{head}}$ as

$$P_m(T_i \rightarrow T_j|S) = G(\Delta p_1, \Sigma_p)G(\Delta p_2, \Sigma_p)$$

where $G(\cdot, \Sigma)$ is the zero-mean Gaussian function, and $\Delta t$ is the frame difference between $p_{\text{tail}}$ and $p_{\text{head}}$.

For $P_a(\cdot)$, we adopt the online learned discriminative appearance models (OLDAMs) defined in [7], which focus on learning appearance models with good global discriminative abilities between targets.

4.3. Learning of Pairwise Terms

Similar to unary terms, pairwise terms are also decomposed into motion based and appearance based parts. Motion based probabilities are defined based on relative distance between tracklet pairs. Take two nodes $(T_1, T_3)$ and $(T_2, T_4)$ as an example. Suppose $T_1$ and $T_2$ are a tail close pair; therefore, there is an edge between the two nodes. Let $t_x = \min\{t_1^e, t_2^e\}$, and $t_y = \max\{t_3^e, t_4^e\}$.

As $T_1$ and $T_2$ are tail close, we estimate positions of both at frame $t_y$, as shown in dash circles in Figure 6. Then we can get the estimated relative distance between $T_1$ and $T_2$ at frame $t_y$ as

$$\Delta p_1 = (v_1^t - v_2^t + v_1^{tail}(t_y - t_1^e)) - (v_2^t + v_2^{tail}(t_y - t_2^e))$$

where $v_1^{tail}$ and $t_1^e$ are the tail velocity and end frames of $T_1$; $v_2^{tail}$ and $t_2^e$ are similar. We compare the estimated relative distance with the real one $\Delta p_2$, and use the same Gaussian function in Equ. 9 to compute the pairwise motion probability as $G(\Delta p_1 - \Delta p_2, \Sigma_p)$.

As shown in Figure 6, the difference between $\Delta p_1$ and $\Delta p_2$ is small. This indicates that if $T_1$ is associated to $T_3$, there is a high probability that $T_2$ is associated to $T_4$ and vice versa. Note that if $T_3$ and $T_4$ in Figure 6 are head close, we also do a similar computation as above; the final motion probability would be taken as the average of both.

Pairwise appearance models are designed for differentiating specific close pairs. For example, in Figure 6, $T_1$ and $T_2$ are a tail close pair; we want to produce an appearance model that best distinguishes the two targets without considering other targets or backgrounds.

Therefore, we online collect positive and negative samples only from the concerned two tracklets so that the learned appearance models are most discriminative for these two. Positive samples are selected from responses in the same tracklet; any pair of these responses should have
high appearance similarity. For a tail close tracklet pair $T_1$ and $T_2$, the positive sample set $S^+$ is defined as

$$S^+ = \{(d_{k,1}^1, d_{k,2}^2)\} \quad k \in \{1, 2\}$$

where $\theta$ is a threshold for the number of frames used for computing appearance models (set to $10$ in our experiments). The introduction of $\theta$ is because a target may change appearance a lot after some time due to illumination, view angle, or pose changes.

Negative samples are selected from responses in different tracklets, and they should have as much differences as possible in appearance. The negative sample set $S^-$ between $T_1$ and $T_2$ is defined as

$$S^- = \{(d_{k,1}^1, d_{k,2}^2)\}$$

where $\theta$ is a threshold for the number of frames used for computing appearance models (set to $10$ in our experiments). The introduction of $\theta$ is because a target may change appearance a lot after some time due to illumination, view angle, or pose changes.

Sample collection for head close pairs is similar, but detection responses are from the first $\theta$ frames of each tracklet.

With the positive and negative sample sets, we adopt the standard Real Boost algorithm to produce appearance models for best distinguishing $T_1$ and $T_2$; we adopt the features defined in [7], including color, texture, and shape for different regions of targets. Based on the pairwise model, we get new appearance based probabilities for $(T_1, T_3)$ and $(T_2, T_3)$ shown in Figure 6. If $T_3$ and $T_4$ are a head close pair, we adopt a similar learning approach to get appearance probabilities based on discriminative models for $T_3$ and $T_4$, and use the average of both scores as the final pairwise appearance probabilities.

Note that the discriminative appearance models between $T_1$ and $T_2$ are only learned once for all edges like $\{(T_1, T_3), (T_2, T_3)\} \forall T_3 \in S$. Therefore, the complexity is much less than the number of edges and becomes $O(n^2)$, where $n$ is the number of tracklets. Moreover, as only a few tracklet pairs are likely to be spatially close, the actual times of learning is often much smaller than $n^2$.

### 4.4. Energy Minimization

For CRF models with submodular energy functions, where $B(0,0) + B(1,1) < B(1,0) + B(1,1)$, a global optimal solution can be found by the graph cut algorithm. However, due to the constraints in Equ. 3, the energy function in our formulation is not sub-modular. Therefore, it is difficult to find the global optimal solution in polynomial time. Instead, we introduce a heuristic algorithm to find a good solution in polynomial time.

The unary terms in our CRF model have been shown to be effective for non-difficult pairs by previous work [7]. Considering this issue, we first use the Hungarian algorithm [11] to find a global optimal solution by only considering the unary terms and satisfying the constraints in Equ. 3. Then we sort the selected associations, i.e., nodes with labels of 1, according to their unary term energies from least to most as $A = \{(v_1 = (T_{i_1} \rightarrow T_{i_2}))\}$. Then for each selected node, we try to switch labels of it and each neighboring node, i.e., a node that is connected with current node by an edge in the CRF model; if the energy is lower, we keep the change. The energy minimization algorithm is shown in Algorithm 1.

\begin{algorithm}
\caption{Finding labels with low energy cost.}
\textbf{Input:} Tracklets from previous level $S = \{T_1, T_2, \ldots, T_n\}$; CRF graph $G = (V, E)$.
\begin{algorithmic}
\State Find the label set $L$ with the lowest unary energy cost by Hungarian algorithm, and evaluate its overall energy $\Psi$ by Equ. 2.
\State Sort nodes with labels of 1 according to their unary energy costs from least to most as $\{v_1, \ldots, v_k\}$.
\For{$i = 1, \ldots, k$} do:
\begin{itemize}
  \item Set updated energy $\Psi' = +\infty$
  \For{$j = 1, \ldots, t$ that $(v_1, v_j) \in E$ do:
    \begin{itemize}
      \item Switch labels of $l_i$ and $l_j$ under constraints in Equ. 3, and evaluate new energy $\Omega$.
      \item If $\Omega < \Psi'$, $\Psi' = \Omega$
    \end{itemize}
  \EndFor
\EndFor
\EndAlgorithmic
\end{algorithm}
\end{algorithm}

Note that the Hungarian algorithm has a complexity of $O(n^3)$, while our heuristic search process has a complexity of $O(|E|) = O(n^4)$. Therefore, the overall complexity is still polynomial. In addition, as nodes are only defined between tracklets with a proper time gap and edges are only defined between nodes with head or tail close tracklet pair, the actual number of edges is typically much smaller than $n^4$. In our experiments, the run time is almost linear in the number of targets.

### 5. Experiments

We evaluate our approach on three public pedestrian data sets: the TUD data set [2], Trecvid 2008 [1], and ETH mobile pedestrian [5] data set. We show quantitative comparisons with state-of-art methods, as well as visualized results of our approach. Though frame rates, resolutions and densities are different in these data sets, we use the same parameter setting, and performance improves compared to previous methods for all of them. This indicates that our approach has low sensitivity on parameters. All data used in our experiments are publicly available\(^1\).

\footnote{\url{http://iris.usc.edu/people/yangbo/downloads.html}}
### 5.1. Evaluation Metric

As it is difficult to use a single score to evaluate tracking performance, we adopt the evaluation metric defined in [10], including:

- **Recall**: correctly matched detections / total detections in ground truth.
- **Precision**: correctly matched detections / total detections in the tracking result.
- **FAF**: Average false alarms per frame.
- **GT**: The number of trajectories in ground truth.
- **MT**: The ratio of mostly tracked trajectories, which are successfully tracked for more than 80%.
- **ML**: The ratio of mostly lost trajectories, which are successfully tracked for less than 20%.
- **PT**: The ratio of partially tracked trajectories, i.e., $1 - MT - ML$.
- **Frag**: fragments, the number of times that a ground truth trajectory is interrupted.
- **IDS**: id switch, the number of times that a tracked trajectory changes its matched id.

For items with $\uparrow$, higher scores indicate better results; for those with $\downarrow$, lower scores indicate better results.

### 5.2. Results on Static Camera Videos

We first test our results on data sets captured by static cameras, *i.e.*, TUD [2] and Trecvid 2008 [1].

For fair comparison, we use the same TUD-Stadtmitte data set as used in [2]. It is captured on a street at a very low camera angle, and there are frequent full occlusions among pedestrians. But the video is quite short, and contains only 179 frames.

The quantitative results are shown in Table 1. We can see that our results are much better than that in [2], but the improvement is not so obvious compared with [8]: we have higher MT and recall and lower id switches, but PRIMPT has higher precision and lower fragments. This is because the online CRF model focuses on better differentiating difficult pairs of targets, but there are not many people in the TUD data set. Some visual results are shown in the first row of Figure 7; our approach is able to keep correct identities while targets are quite close, such as person 0, person 1, and person 2.

To see the effectiveness of our approach, we further evaluate our approach on the difficult Trecvid 2008 data set. There are 9 video clips in the data set, each of which has 5000 frames; these videos are captured in a busy airport, and have high density of people with frequent occlusions. There are lots of close track interactions in this data set, indicating huge number of edges in the CRF graph. The comparison results are shown in Table 2. Compared with up-to-date approaches, our online CRF achieves best performance on precision, FAF, fragments, and id switches, while keeping recall and MT competitive. Compared with [8], our approach reduces the fragments and the id switches by about 15% and 14% respectively. Row 2 and 3 in Figure 7 show some tracking examples by our approach. We can see that when targets with similar appearances get close, the online CRF can still find discriminative features to distinguish these difficult pairs. However, global appearance and motion models are not effective enough in such cases, such as person 106 and 109 in the third row of Figure 7, who are both in white, move in similar directions, and are quite close. The second row in Figure 2 shows an example where the approach in [7] produces a fragmentation due to non-linear motions while our approach has no fragments by considering pairwise terms in the CRF model.

### 5.3. Results on Moving Camera Videos

We further evaluate our approach on the ETH data set [5], which is captured by a pair of cameras on a moving stroller in busy street scenes. The stroller is mostly moving forward, but sometimes has panning motions, which makes

<table>
<thead>
<tr>
<th>Method</th>
<th>Recall</th>
<th>Precision</th>
<th>FAF</th>
<th>GT</th>
<th>MT</th>
<th>PT</th>
<th>ML</th>
<th>Frag</th>
<th>IDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy Minimization [2]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>9</td>
<td>60.0%</td>
<td>30.0%</td>
<td>0.0%</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>PRIMPT [8]</td>
<td>81.0%</td>
<td>99.5%</td>
<td>0.028</td>
<td>10</td>
<td>60.0%</td>
<td>30.0%</td>
<td>0.0%</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Online CRF Tracking</td>
<td>87.0%</td>
<td>96.7%</td>
<td>0.184</td>
<td>10</td>
<td>70.0%</td>
<td>30.0%</td>
<td>0.0%</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: Comparison of results on TUD dataset. The PRIMPT results are provided by courtesy of authors of [8]. Our ground truth includes all persons appearing in the video, and has one more person than that in [2].

<table>
<thead>
<tr>
<th>Method</th>
<th>Recall</th>
<th>Precision</th>
<th>FAF</th>
<th>GT</th>
<th>MT</th>
<th>PT</th>
<th>ML</th>
<th>Frag</th>
<th>IDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline CRF Tracking [19]</td>
<td>79.2%</td>
<td>85.8%</td>
<td>0.996</td>
<td>919</td>
<td>78.2%</td>
<td>16.9%</td>
<td>4.9%</td>
<td>319</td>
<td>253</td>
</tr>
<tr>
<td>OLDAMs [7]</td>
<td>80.4%</td>
<td>86.1%</td>
<td>0.992</td>
<td>919</td>
<td>76.1%</td>
<td>19.3%</td>
<td>4.6%</td>
<td>322</td>
<td>224</td>
</tr>
<tr>
<td>PRIMPT [8]</td>
<td>79.2%</td>
<td>86.8%</td>
<td>0.920</td>
<td>919</td>
<td>77.0%</td>
<td>17.7%</td>
<td>5.2%</td>
<td>283</td>
<td>171</td>
</tr>
<tr>
<td>Online CRF Tracking</td>
<td>79.8%</td>
<td>87.8%</td>
<td>0.857</td>
<td>919</td>
<td>75.5%</td>
<td>18.7%</td>
<td>5.8%</td>
<td>240</td>
<td>147</td>
</tr>
</tbody>
</table>

Table 2: Comparison of tracking results on Trecvid 2008 dataset. The human detection results are the same as used in [19, 7, 8], and are provided by courtesy of authors of [8].
the motion affinity between two tracklets less reliable.

For fair comparison, we choose the “BAHNHOFF” and “SUNNY DAY” sequences used in [8] for evaluation. They have 999 and 354 frames respectively, and people are under frequent occlusions due to the low view angles of cameras. For fair comparison with [8], we also use the sequence from the left camera; no depth and ground plane information are used.

The quantitative results are shown in Table 3. We can see that our approach achieves better or the same performances on all evaluation scores. The mostly tracked score is improved by about 10%; fragments are reduced by 17%; recall and precision are improved by about 2% and 4% respectively. The obvious improvement in MT and fragment scores indicate that our approach can better track targets under moving cameras, where the traditional motion models are less reliable.

The last two rows in Figure 7 show some visual tracking results by our online CRF approach. Both examples show obvious panning movements of cameras. Traditional motion models, i.e., unary motion models in our online CRF, would produce low affinity scores for tracklets belonging to the same targets. However, by considering pairwise terms, relative positions are helpful for connecting correct tracklets into one. This explains the obvious improvements on MT and fragments. The first row in Figure 2 shows an example that our approach successfully tracks persons 41 and 48 under abrupt camera motions, while the method in [7] fails to find the correct associations.

5.4. Speed

As discussed in Section 4.4, the complexity of our algorithm is polynomial in the number of tracklets. Our experiments are performed on a Intel 3.0GHz PC with 8G memory, and the codes are implemented in C++. For the less crowded TUD and ETH data sets, the speed are both about 10 fps; for crowded Trecvid 2008 data set, the speed is about 6 fps. Compared with the speed of 7 fps for Trecvid 2008 reported in [8], the online CRF does not add much to the computation cost (detection time costs are not included in either measurement).

6. Conclusion

We described an online CRF framework for multi-target tracking. This CRF considers both global descriptors for distinguishing different targets as well as pairwise descriptors for differentiating difficult pairs. Unlike global descriptors, pairwise motion and appearance models are learned from corresponding difficult pairs, and are further represented by pairwise terms in the CRF energy function. An effective algorithm is introduced to efficiently find associations with low energy, and the experiments show significantly improved results compared with up-to-date methods. Future improvement can be achieved by adding camera motion inference into pairwise motion models.

Acknowledgments

This paper is based upon work supported in part by Office of Naval Research under grant number N00014-10-1-0517.

References

<table>
<thead>
<tr>
<th>Method</th>
<th>Recall</th>
<th>Precision</th>
<th>FAF</th>
<th>GT</th>
<th>MT</th>
<th>PT</th>
<th>ML</th>
<th>Frag</th>
<th>IDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIMPT [8]</td>
<td>76.8%</td>
<td>86.6%</td>
<td>0.891</td>
<td>125</td>
<td>58.4%</td>
<td>33.6%</td>
<td>8.0%</td>
<td>23</td>
<td>11</td>
</tr>
<tr>
<td>Online CRF Tracking</td>
<td>79.0%</td>
<td>90.4%</td>
<td>0.637</td>
<td>125</td>
<td>68.0%</td>
<td>24.8%</td>
<td>7.2%</td>
<td>19</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 3. Comparison of tracking results on ETH dataset. The human detection results are the same as used in [8], and are provided by courtesy of authors of [8].

Figure 7. Tracking examples on TUD [2], Trecvid 2008 [1], and ETH [5] data sets.