Learning a Proposal Classifier for Multiple Target Tracking
(Supplementary Material)

Peng Dai1 Renliang Weng2 Wongun Choi2 Changshui Zhang1 Zhangping He2
Wei Ding2
1Tsinghua University, Beijing, China. 2Aibee Inc
1{daip2020, zcs}@mail.tsinghua.edu.cn 2{rlweng, wgchoi, zphe, weiding}@aibee.com

A. Detailed Algorithm

In this section, we first detail the gating strategy in affinity graph construction, and then provide the pseudocode of the algorithms presented in the main paper.

A.1. Gating Strategy

To reduce the complexity of the graph, we adopt a simple gating strategy to remove the edges exceeding the thresholds. Specifically, let $O_i$ represent the valid neighbors of vertex $v_i$, and $O_i$ is obtained by:

$$O_i = \{\forall v_j: I^t(t_i, t_j, \tau_t) \& I^p(p_i, p_j, \tau_p) \& I^a(a_i, a_j, \tau_a)\}$$

(1)

where $I^t$ is an indicator function to check if the minimum time gap between vertex $v_i$ and $v_j$ is less than $\tau_t$, $I^p$ is also an indicator function to check if the location distance is less than $\tau_p$ when having the minimum time gap, and $I^a$ checks if the appearance distance is less than $\tau_a$. The thresholds $\tau_t$, $\tau_p$, and $\tau_a$ determine the radius of the gate.

A.2. Proposal Generation and Deoverlapping

Algorithm 1 and Algorithm 2 show the detailed procedures to generate proposals. In these algorithms, $s_{max}$ (maximum cluster size) and $\Delta$ (cluster threshold step) are utilized to improve the purity of the generated clusters in the early iterations. It should be noted that we adopt a compatible function to keep all pairwise vertices within a cluster to be temporally compatible, i.e., no temporally overlapping vertices are allowed within the same cluster.

Algorithm 3 provides a summary of the de-overlapping procedures to generate the final tracking output.

B. Parameter Sensitivity Analysis

Here, we investigate the effects of different settings on parameter $s_{max}$, $\Delta$ and $K$ (the maximum number of edges linked to one vertex) to the tracking performance. The parameter $s_{max}$ and $\Delta$ are used to control the growth speed of the proposals. The results in Figure 1 and Figure 2 show that we can choose $s_{max} \in [2, 4]$, $\Delta \in [0.02, 0.06]$ to achieve the satisfactory and stable performance. With the increasing $s_{max}$ or $\Delta$, more noises will be introduced to the proposals in early iterations, hence reducing the performance. The parameter $K$ controls the number of edges in the graph construction. The results in Figure 3 show that a satisfactory and stable performance can be achieved when $K > 1$.

C. Evaluation Results on MOT16

We also report the quantitative results obtained by our method on MOT16 in Table 1 and compare it to methods that are officially published on the MOTChallenge benchmark. Our method can also obtain state-of-the-art IDF1 score on MOT16.

D. Qualitative Analysis

Figure 4 and Figure 5 give a qualitative comparison between MPNTrack[3] and our method on MOT17. It validates that our method has better performance in handling long-term occlusions, hence achieving higher IDF1 score.
Algorithm 2: Cluster Nodes

Input: Symmetric affinity matrix \( G \), maximum cluster size \( s_{\text{max}} \), cluster threshold step \( \Delta \).

Output: Clusters \( C \)

1. function main:
   2. \( C = \emptyset \), \( R = \emptyset \), \( \tau = \min(G) \);
   3. \( C', R = \text{FindClusters}(G, \tau, s_{\text{max}}) \);
   4. \( C = C \cup C' \);
   5. while \( R \neq \emptyset \) do
      6. \( \tau = \tau + \Delta \);
      7. \( C', R = \text{FindClusters}(G_R, \tau, s_{\text{max}}) \);
      8. \( C = C \cup C' \);
   9. end
10. return \( C \);

11. function FindClusters \((G, \tau, s_{\text{max}})\):
   12. \( G' = \text{PruneEdge}(G, \tau) \);
   13. \( S = \text{FindConnectedComponents}(G') \);
   14. \( C' = \{ c | c \in S, |c| < s_{\text{max}} \text{ and } \text{Compatible}(c) \} \);
   15. \( R = S \setminus C' \);
   16. return \( C', R \);

17. function Compatible \((c)\):
   18. if \( d(t_i, t_j) > 0, \forall i, j \in c, i \neq j \) then
      19. return True ;
   20. else
      21. return False ;
   22. end

Algorithm 3: De-overlapping

Input: Ranked Proposals \( \{\hat{P}_1, \hat{P}_2, \cdots, \hat{P}_{N_p}\} \)

Output: Tracking Results \( T \)

1. Dictionary \( T = \{\} \), Occupied Set \( I = \emptyset, i = 1 \);
2. while \( i < N_p \) do
   3. \( C_i = \hat{P}_i \setminus I \);
   4. for \( v_i \) in \( C_i \) do
      5. \( T[v_i] = i \);
   6. end
   7. \( I = I \cup C_i \);
   8. \( i = i + 1 \);
9. end
10. Return \( T \);

E. Further Performance Comparison

We also noticed that MPNTrack [3] used a different Reidentification (ReID) model from our method. In order to achieve a completely fair comparison, we also provide the comparison results between our method and MPNTrack using our ReID model on the training set of MOT17. Table 2 shows the detailed results. By comparing our method with MPNTrack\(^2\), it is clear that our method achieves better performance on identity preservation, improving the IDF1
Table 1. Performance comparison with start-of-the art on MOT16 (top: offline methods; bottom: online methods).

<table>
<thead>
<tr>
<th>Method</th>
<th>MOTA↑</th>
<th>IDF1↑</th>
<th>MT↑</th>
<th>ML↓</th>
<th>FP↓</th>
<th>FN↓</th>
<th>IDs↓</th>
<th>Hz↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>58.8</td>
<td>67.6</td>
<td>27.3</td>
<td>35.0</td>
<td>6167</td>
<td>68432</td>
<td>435</td>
<td>4.3</td>
</tr>
<tr>
<td>Lif_T [5]</td>
<td>61.3</td>
<td>64.7</td>
<td>27.0</td>
<td>34.0</td>
<td>4844</td>
<td>65401</td>
<td>389</td>
<td>0.5</td>
</tr>
<tr>
<td>MPNTrack [3]</td>
<td>58.6</td>
<td>61.7</td>
<td>27.3</td>
<td>34.0</td>
<td>4949</td>
<td>70252</td>
<td>354</td>
<td>6.5</td>
</tr>
<tr>
<td>HDTR [1]</td>
<td>53.6</td>
<td>46.6</td>
<td>21.2</td>
<td>37.0</td>
<td>4714</td>
<td>79353</td>
<td>618</td>
<td>3.6</td>
</tr>
<tr>
<td>TPM [9]</td>
<td>51.3</td>
<td>47.9</td>
<td>18.7</td>
<td>40.8</td>
<td>2701</td>
<td>85504</td>
<td>569</td>
<td>0.8</td>
</tr>
<tr>
<td>CRF_TRACK [10]</td>
<td>50.3</td>
<td>54.4</td>
<td>18.3</td>
<td>35.7</td>
<td>7148</td>
<td>83614</td>
<td>614</td>
<td>19.2</td>
</tr>
<tr>
<td>NOTA [4]</td>
<td>49.8</td>
<td>55.3</td>
<td>17.9</td>
<td>37.7</td>
<td>7248</td>
<td>83614</td>
<td>614</td>
<td>19.2</td>
</tr>
<tr>
<td>UnsupTrack [6]</td>
<td>62.4</td>
<td>58.5</td>
<td>27.0</td>
<td>31.9</td>
<td>5909</td>
<td>61981</td>
<td>588</td>
<td>1.9</td>
</tr>
<tr>
<td>GNNMatch [8]</td>
<td>57.2</td>
<td>55.0</td>
<td>22.9</td>
<td>34.0</td>
<td>3905</td>
<td>73493</td>
<td>559</td>
<td>0.3</td>
</tr>
<tr>
<td>Tracktor [2]</td>
<td>56.2</td>
<td>54.9</td>
<td>20.7</td>
<td>35.8</td>
<td>2394</td>
<td>76844</td>
<td>517</td>
<td>6.5</td>
</tr>
<tr>
<td>TrectD16 [11]</td>
<td>54.8</td>
<td>53.4</td>
<td>19.1</td>
<td>37.0</td>
<td>2955</td>
<td>78765</td>
<td>645</td>
<td>1.6</td>
</tr>
<tr>
<td>PV [7]</td>
<td>50.4</td>
<td>50.8</td>
<td>14.9</td>
<td>38.9</td>
<td>2600</td>
<td>86780</td>
<td>1061</td>
<td>7.3</td>
</tr>
</tbody>
</table>

Figure 4. A qualitative example showing (a) a failure case of MPNTrack [3] in handling long-term occlusions, which reduces the IDF1 score; (b) our method can effectively handle this case. The numbers are the object IDs. Best viewed in color.

Figure 5. A qualitative example showing (a) a failure case of MPNTrack [3] in handling occlusions, which leads to an identity transfer when one person passes the other and a fragmentation when one is fully occluded; (b) our method can effectively handle this case. The numbers are the object IDs. Best viewed in color.

Table 2. Further performance comparison on the training set of MOT17.

<table>
<thead>
<tr>
<th>Method</th>
<th>MOTA↑</th>
<th>IDF1↑</th>
<th>MT↑</th>
<th>ML↓</th>
<th>FP↓</th>
<th>FN↓</th>
<th>IDs↓</th>
<th>Hz↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>63.9</td>
<td>71.8</td>
<td>647</td>
<td>377</td>
<td>7176</td>
<td>10500</td>
<td>728</td>
<td></td>
</tr>
<tr>
<td>MPNTrack1</td>
<td>64.0</td>
<td>70.0</td>
<td>648</td>
<td>362</td>
<td>619</td>
<td>114509</td>
<td>602</td>
<td></td>
</tr>
<tr>
<td>MPNTrack2</td>
<td>63.9</td>
<td>70.3</td>
<td>634</td>
<td>365</td>
<td>6228</td>
<td>114723</td>
<td>523</td>
<td></td>
</tr>
</tbody>
</table>

1 with their own ReID model
2 with our ReID model

References


