Prioritizing the propagation of identity beliefs for multi-object tracking

Amit Kumar K.C.
Christophe De Vleeschouwer
Université catholique de Louvain
Organization

• **Context**
  – Grouping detections into tracklets
  – Assigning prior identity distribution to the tracklets

• **Standard belief propagation**
  – Formalization of the labeling problem
  – Message passing

• **Priority belief propagation**
  – Graph structure
  – Potential function definition
  – Scheduling of the nodes

• **Experimental results**
Context

• A basketball game captured by multiple cameras

• Players have been already detected.

• Detections are grouped into segments → tracklets.

• **Objective**: to recognize the players
Multi-object tracking: Two stage process

1. **Detect candidate locations and extract features**
2. **Link the candidate detections into **tracklets**
3. **Assign Identity to tracklets**

Input video → Detections → Tracklets → Labels
Multi-object tracking: Two stage process

1. Detect candidate locations and extract features
2. Link the candidate detections into *tracklets*
3. Assign Identity to tracklets
Multi-object tracking: Two stage process

- Detect candidate locations and extract features
- Link the candidate detections into *tracklets*
- Assign identity to tracklets
Detect candidate locations and extract features

Link the candidate detections into *tracklets*

Assign identity to tracklets

Fewer entities to label.
Multi-object tracking: Two stage process


Features cannot always be reliably estimated.

Digit feature can be read only when facing the camera.

Color feature is noisy because of occlusion, clutter, etc.
Inferring identity of tracklets from appearance features is difficult.

Availability and reliability of appearance features vary across time.
How to exploit noisy/missing features?

Each feature $f$ has a confidence value $0 \leq c \leq 1$.

Damien Delannay, Nicolas Danhier and Christophe De Vleeschouwer, “Detection and recognition of sports (wo)men from multiple views”, ICDSC 2009, Como, Italy
How to exploit noisy/missing features?

Each tracklet $v$ has:
- Accumulated appearance features, $\overrightarrow{F}(v) = \{\overrightarrow{f}(v), \ldots, \overrightarrow{f}(v)^{K}\}$
- Their confidence values, $\mathbf{C} = (c_1, \ldots, c_K)$
Assigning prior identity distribution (1/2)

• Assumptions
  – $N$ targets
  – $K$ appearance features for target, $j$
    
    \[ \mathcal{F}(j) = \{ f_1^{(j)}, \ldots, f_K^{(j)} \} \]

• Example: Basketball game
  – $K=2$, \( \mathcal{F} = \{ \text{color}, \text{digit} \} \)

• Average features of a tracklet, $v$
    
    \[ \overline{\mathcal{F}}(v) = \{ \overline{f}_1^{(v)}, \ldots, \overline{f}_K^{(v)} \} \]
Assigning prior identity distribution (2/2)

Probability of tracklet $v$ having $j$th identity

$$p_v(j) \propto \prod_{i=1}^{K} \exp\left[ - \frac{\| \mathbf{f}_i(j) - \overline{f}_i^{(v)} \|_1}{\tau_i^{(v)}} \right]$$

for $1 \leq j \leq N$

- Decreases when the feature $i$ becomes more reliable.
Some tracklets have more ambiguous labels.
How to infer the identity of more ambiguous tracklets?

Gather identity information from less ambiguous neighbor tracklets.
Identity assignment: an inference problem

• Graph-based formalism
  – Node $\rightarrow$ tracklet
  – Edge $\rightarrow$ identities of nodes are dependent
    • Nodes co-exist in time
    • Nodes are close in time, space, appearance

**Problem:** Estimate the most likely identity of a node given the identities of other nodes.

**Approach:** Exchange messages between nodes
Standard belief propagation

How are messages formed and exchanged?
Message is exchanged between nodes.
Formally...

Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

A node $v \in \mathcal{V}$, characterized by $\phi_v(l_v)$

An edge $e = (u, v) \in \mathcal{E}$, characterized by $\phi_{uv}(l_u, l_v)$

$l_u, l_v \in \mathcal{L} = \{1, \ldots, N\}$

Likelihood that node $v$ has a label $l_v$

Likelihood that nodes $u$ and $v$ have a labels $l_u$ and $l_v$
Objective: label the nodes

Find a labeling function $l$ that labels each node $v$ with $l_v$ so as to maximize the joint likelihood function:

$$p(l) \propto \prod_{v \in \mathcal{V}} \left[ \phi_v(l_v) \prod_{u \in \mathcal{N}_v} \phi_{uv}(l_u, l_v) \right]$$

Iteratively exchange messages:

$$m_{u \rightarrow v}^{(t)}(l_v) \propto \sum_{l_u \in \mathcal{L}} \left[ \phi_{uv}(l_u, l_v) \phi_u(l_u) \prod_{s \in \mathcal{N}_u \setminus \{v\}} m_{s \rightarrow u}^{(t-1)}(l_u) \right]$$

Message sent by $u$ to $v$ at iteration $t$ about label $l_v$
4 steps to compute message

1. Gather messages from neighbors

2. Multiply by corresponding likelihoods

3. Marginalize (or, maximize) over all possible labels

4. Normalize the message

\[
m_{u \rightarrow v}^{(t)}(l_v) \propto \sum_{l_u \in \mathcal{L}} \left[ \phi_{uv}(l_u, l_v) \phi_u(l_u) \prod_{s \in \mathcal{N}_u \setminus v} m_{s \rightarrow u}^{(t-1)}(l_u) \right]
\]
After $T$ iterations, compute belief and assign label

$$b_v(l_v) \propto \phi_v(l_v) \prod_{s \in \mathcal{N}_v} m_{s\rightarrow u}^{(T)}(l_v)$$

1. Gather messages from neighbors
2. Multiply by likelihood

$$l_v^* = \arg \max_{l_v \in \mathcal{L}} b_v(l_v)$$

3. Choose the label that maximizes the belief
How our graph looks like.

- Temporal edges
- Mutex edges
- Flat distribution
- Peaky distribution
- Intermediate distribution
How our graph looks like.

Mutex edge:
Nodes (or, tracklets) co-exist in time $\rightarrow$ different labels

Temporal edge:
Sufficiently "close" $\rightarrow$ same label
Far apart $\rightarrow$ different labels

- **Temporal edges**
- **Mutex edges**
- **Flat distribution**
- **Peaky distribution**
- **Intermediate distribution**
Potential definition (1/2)

\[ \phi_v(l_v) := p_v(l_v), l_v \in \mathcal{L} \]

\[ \phi_{uv}(l_u, l_v) = \begin{cases} 
\epsilon & \text{if } l_u = l_v \\
1 - \epsilon & \text{otherwise,}
\end{cases} \]

Small positive constant

\[ \phi_{uv}(l_u, l_v) = \begin{cases} 
\exp(-d_{uv}/\tau_{\text{dist}}) & \text{if } l_u = l_v \\
1 - \exp(-d_{uv}/\tau_{\text{dist}}) & \text{otherwise,}
\end{cases} \]
Potential definition (2/2)

If \( E(u) < \tau_{TH} \) and \( E(v) < \tau_{TH} \), then

\[
d_{uv} = \left[ 1 - \sum_{l_v \in \mathcal{L}} \sqrt{b_u(l_v)b_v(l_v)} \right]^{1/2}
\]

If both nodes have unambiguous identity information, then computing Bhattacharyya distance suffices.

If \( |t_v^{(s)} - t_u^{(e)}| < \tau_{\text{max}} \), then

\[
d_{uv} = \left\| \mathbf{x}_v^{(s)} - \mathbf{x}_u^{(e)} \right\|_2
\]

Position feature helps only when nodes are not far in time.
From Standard BP to Priority BP

Some node have more ambiguous labels than others.
Motivations

Two observations

1. Standard BP does not prioritize nodes
   – Nodes are selected arbitrarily to transmit messages.

2. During message construction, a node gathers message from all of its neighbors.
   – Messages from more ambiguous nodes are usually uninformative.
Priority belief propagation

• Prioritize the nodes according to their ambiguity levels
  – Less ambiguous nodes transmit messages first.
    • Faster convergence

• Avoid transmission of uninformative messages
  • Gather(disseminate) messages only from(to) the less(more) ambiguous nodes
  • Fewer message construction and dissemination
Algorithm 1 Priority Belief Propagation

Initialize: \( b_v(l_v) \leftarrow p_v(l_v) \) and \( \phi_v(l_v) \leftarrow p_v(l_v) \) \( \forall v \in \mathcal{V}, l_v \in \mathcal{L} \)

for \( t = 1 \) to \( T \) do

\( v.\text{committed} \leftarrow \text{false} \ \forall v \in \mathcal{V} \)

\( \mathcal{R} \leftarrow \mathcal{V} \)

while \( \mathcal{R} \neq \emptyset \) do

\( u \leftarrow \text{Schedule}(\mathcal{R}) \)

\( u.\text{committed} \leftarrow \text{true} \)

\( S \leftarrow \{ s \mid s \in \mathcal{N}_u, s.\text{committed} = \text{true} \} \) \( \{/*/ \text{Less ambiguous neighbors of } u */\} \)

\( h_u \leftarrow \text{Compute the pre-message of } u \text{ from } S \)

\( Q \leftarrow \{ v \mid v \in \mathcal{N}_u, v.\text{committed} = \text{false} \} \) \( \{/*/ \text{More ambiguous neighbors of } u */\} \)

for \( v \in Q \) do

\( \mathbf{m}_{u \rightarrow v}^{(t)} \leftarrow \text{ComputeMessage}(u, v, h_u, \tau_{\text{max}}) \)

\( b_v \leftarrow \text{Compute belief for node } v \)

end for

\( \mathcal{R} \leftarrow \mathcal{R} \setminus u \)

end while

end for
General idea

You look MORE like me
msg=[70% 1, 30% 2]

You look LESS like me
msg=[30% 1, 70% 2]

You CANNOT be me
msg=[0% 1, 100% 2]

Mutual exclusion

Less ambiguous label

More ambiguous label
Scheduling of the nodes

• Node ambiguity level estimation

• Two approaches
  – Entropy based
    \[ E(v) = - \sum_{l_v \in \mathcal{L}} b_v(l_v) \log b_v(l_v) \]
    • Low entropy \(\rightarrow\) Reliable identity assignment
    • Sort the nodes in ascending order of entropy
  – Confusion set based
Confusion set based

Given the belief vector, $b_v$

- Count the number of likely labels, i.e., those whose belief value exceed a certain threshold.

- More the number of labels $\Rightarrow$ more ambiguous identity

\[
\mathcal{CS}(v) = \left\{ l_v | b_v^{(rel)}(l_v) \geq \tau_b \right\}
\]

\[
b_v^{(rel)}(l_v) = b_v(l_v) - \min \{ b_v(l_v) \}
\]

- Sort the node $v$ in ascending order of $|\mathcal{CS}(v)|$
Experimental validation

• Dataset
• Performance metrics
• Results
Dataset

• 10 mins video of APIDIS dataset
  – http://www.apidis.org/Dataset/

• 7 cameras
  – loosely synchronized
  – Asymmetrical distribution of cameras around the court
    • Observation of features is more reliable on one side of the court than the other.

• Manual annotation of player IDs at each second ➔ Ground Truth
Performance metrics

Ground Truth

Hypothesis

Miss detection (MS)

Correct detection but Wrong identification (WI)

False Positive (FP)
Results: different scheduling mechanisms

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (%)</th>
<th>FP (%)</th>
<th>WI (%)</th>
<th>MS(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No BP</td>
<td>68.14</td>
<td>0.13</td>
<td>1.48</td>
<td>30.25</td>
</tr>
<tr>
<td>Std BP (seq.)</td>
<td>73.59</td>
<td>0.15</td>
<td>1.76</td>
<td>24.50</td>
</tr>
<tr>
<td>Std BP (rand).</td>
<td>83.69</td>
<td>0.16</td>
<td>2.36</td>
<td>13.79</td>
</tr>
<tr>
<td>Priority BP</td>
<td><strong>89.04</strong></td>
<td>0.15</td>
<td>2.54</td>
<td>8.27</td>
</tr>
</tbody>
</table>

Scheduling affects the quality of solution.
### Results: different graphical models

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (%)</th>
<th>FP (%)</th>
<th>WI (%)</th>
<th>MS (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No edges</td>
<td>68.14</td>
<td>0.13</td>
<td>1.48</td>
<td>30.25</td>
</tr>
<tr>
<td>Mutex edges only</td>
<td>80.32</td>
<td>0.15</td>
<td>2.03</td>
<td>17.50</td>
</tr>
<tr>
<td>Temporal edges only</td>
<td>84.49</td>
<td>0.15</td>
<td>2.71</td>
<td>12.65</td>
</tr>
<tr>
<td><strong>Both edges</strong></td>
<td><strong>89.04</strong></td>
<td><strong>0.15</strong></td>
<td><strong>2.54</strong></td>
<td><strong>8.27</strong></td>
</tr>
</tbody>
</table>

- Temporal edges seem to bring more information as compared to mutex edges.

- However, it cannot be generalized.
## Results: team-wise recognition results

<table>
<thead>
<tr>
<th></th>
<th>Blue team (%)</th>
<th>Yellow team (%)</th>
<th>Average (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No BP</strong></td>
<td>48.74</td>
<td>87.54</td>
<td>68.14</td>
</tr>
<tr>
<td><strong>Std. BP</strong></td>
<td>55.54</td>
<td>91.64</td>
<td>73.59</td>
</tr>
<tr>
<td><strong>Priority BP</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mutex edges</td>
<td>70.11</td>
<td>90.53</td>
<td>80.32</td>
</tr>
<tr>
<td>Temporal edges</td>
<td>77.45</td>
<td>91.54</td>
<td>84.49</td>
</tr>
<tr>
<td>Both edges</td>
<td><strong>86.37</strong></td>
<td><strong>91.71</strong></td>
<td><strong>89.04</strong></td>
</tr>
</tbody>
</table>

- Yellow team is recognized more often.
- Possible reason(s):
  - More cameras on the yellow side.
  - Silhouette of blue team player is more degraded than the yellow team player.
Results: confusion set based scheduling

<table>
<thead>
<tr>
<th>$T_b$</th>
<th>0.01</th>
<th>0.03</th>
<th>0.1</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>90.16</td>
<td>89.05</td>
<td>86.23</td>
<td>69.84</td>
</tr>
<tr>
<td>FP (%)</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.13</td>
</tr>
<tr>
<td>WI (%)</td>
<td>2.52</td>
<td>2.45</td>
<td>2.24</td>
<td>1.85</td>
</tr>
<tr>
<td>MS (%)</td>
<td>7.17</td>
<td>8.35</td>
<td>11.38</td>
<td>28.18</td>
</tr>
</tbody>
</table>

- Performance is comparable to that of entropy-based scheduling.
- Easier to implement than entropy.
Demo video
Conclusions

• Identity assignment problem in which observations have various degree of reliabilities.

• Priority BP to address the problem.

• The order in which the messages are exchanged between the nodes affects the solution.

• 89% accurate recognition
Thank you for the kind attention!