Trajectory Ensemble: Multiple Persons Consensus Tracking across Non-overlapping Multiple Cameras over Randomly Dropped Camera Networks

Yasutomo Kawanishi  Daisuke Deguchi  Ichiro Ide  Hiroshi Murase  
Nagoya University  
Furo-cho, Chikusa-ku, Nagoya, Aichi, JAPAN  
kawanishi@i.nagoya-u.ac.jp

Abstract

Multiple person tracking over a camera network is usually performed by matching person images between adjacent cameras. It easily fails by a temporal appearance change of the persons caused by environmental illumination and observation orientation of a camera. To solve this problem, matching person images across not only adjacent cameras but also cameras multiple hops away in the camera network is effective, but such relaxation of spatio-temporal cues also cause tracking failure due to the increase of matching candidates. To avoid the failure, we introduce “Random Camera Drop” to generate different camera networks which relax the spatio-temporal cues partially and randomly. We then, integrate tracking results over the networks to a consensus tracking result by a novel concept “Trajectory Ensemble”, an extension of unsupervised ensemble learning for the multiple person tracking over a camera network problem. We evaluated the framework on several virtual datasets generated from a public dataset, “Shinpuhkan 2014 dataset” and confirmed that the proposed method achieves the highest tracking results among several comparative methods.

1. Introduction

Many surveillance cameras have been installed in our daily environment and utilized for observing activities of persons (Figure 1). The goal of our research is to obtain trajectories of persons by using multiple surveillance cameras whose views do not overlap. In this paper, the term “trajectory” stands for a sequence of camera views where a person visited. It has a lot of potential commercial applications and great importance for business growth. If we could obtain trajectories of multiple persons, we can utilize them for various purposes such as finding suspicious persons in a shopping mall, finding similar spots, or planning shop reallocation for sales growth. For obtaining trajectories of multiple persons, multiple person tracking across multiple cameras is a fundamental technique. For statistical analysis of trajectories, the trajectories do not need to be obtained in real-time. Thus, we focus on off-line tracking.

Most existing methods attempt to track persons across multiple camera views by utilizing appearance features and certain spatio-temporal cues to re-identify and associate persons across adjacent camera views [1, 2, 3, 4, 5]. Appearance features are usually based on color histograms and texture descriptors [6, 7, 8]. Recently, as same as other computer vision applications, Deep Learning based approaches are also applied to this field [9, 10, 11]. Spatio-temporal cues are commonly based on the adjacency of camera views and the distribution of travel time between adjacent camera views [12, 13, 14, 15, 16].

Appearance features are easily affected by illumination in a camera view and the positional relationship between a camera and a person. Illumination in a camera view and direction of the camera are different among cameras, while appearance features vary even for the same person. Therefore, in case of multiple person tracking across multiple cameras based on similarity comparison, temporal change
in the appearance of a person could easily cause an interruption in the tracking or switching to another person (Figure 2 (i)).

To avoid the interruption of tracking across multiple cameras, we considered that matching person images not only between two adjacent camera views but also between other camera views several hops away over a camera network could be a solution. However, it will also cause more failure in the matching due to the increase of candidates.

Skipping cameras in the network is likely to relax spatio-temporal cues partially (Figure 2 (ii)). Instead of skipping cameras, we consider to drop cameras from the network. We introduce “Random Camera Drop” to generate different camera networks which randomly relaxes the spatio-temporal cues. By tracking multiple persons over the generated networks, we can obtain multiple tracking results. By introducing a novel concept “Trajectory Ensemble”, which is an extension of unsupervised ensemble learning [17, 18] for the multiple person tracking over a camera network problem, the tracking results are integrated and the consensus tracking result is obtained.

The rest of the paper is organized as follows: First, we define the problem of multiple person tracking over a camera network and discuss the difficulty of a straightforward approach in Section 2. In Section 3, details of the proposed method “Trajectory Ensemble” are introduced. Experimental results are reported in Section 4. Finally, we conclude this paper in Section 5.

Figure 2. Matching over a camera network. White circles connected by lines show adjacent camera views.

(i) Matching images across two adjacent cameras. Matching fails and tracking across multiple cameras is interrupted twice.

(ii) Matching images across two non-adjacent cameras. Matching is successfully performed.

Figure 3. “Multiple Person Tracking across Multiple Cameras.” Each number enclosed with a circle is a label assigned for the corresponding tracklet. In this example, tracking is successfully performed and consistent labels are assigned to the persons.

2. Multiple person tracking over a camera network

2.1. Problem setting and basic approach

Let us consider the situation that multiple cameras whose views do not overlap are installed in an environment like a shopping mall, and multiple persons walk in the environment as shown in Figure 1. When a person enters a camera view, the person is detected and tracked until the person exits the camera view. Then an image sequence of the person (tracklet) \( r_i = \{ m_{ij} \}_{j=1}^{n_i} \) is obtained. Here, \( m_{ij} \) denotes an image of the cropped person and \( n_i \) denotes the number of frames the person was tracked in the camera view. Here, for simplicity, we do not consider false detections, namely, detection of non-persons and misdetections. In this paper, “multiple person tracking across multiple cameras” is a problem that assigns consistent label of the same person to tracklets \( \{ r_1, r_2, \ldots \} \) obtained by multiple cameras \( \{ c_1, c_2, \ldots, c_N \} \in C \) (Figure 3).

Since it is difficult to decide if two tracklets are of the same person only by image features, spatio-temporal cues are introduced. Here, spatio-temporal cues consist of adjacency of cameras and traveling time between two camera pairs, which are described by a directed graph \( G = (C, E) \) where \( C = \{ c_1, c_2, \ldots, c_N \} \) and \( E = \{ (c_j, c_k) | c_j, c_k \in C \} \) denote a set of vertices and a set of edges, respectively. Each vertex corresponds to a camera view, and two cameras connected by an edge are adjacent. We define “adjacent cameras” as two cameras where persons can travel between them without crossing other camera views. Each edge is assigned parameters that represent the distribution of traveling time between the two cameras connected by the edge. In this problem setting, the distribution of traveling time is
Figure 4. The concept of Trajectory Ensemble. These images are observed by different cameras. Dashed circles indicate that the tracklets are not used for “weak” tracking since the corresponding cameras are dropped in the sub-networks.

Figure 5. Sub-networks generated by Random Camera Drop. Some camera pairs have multiple edges to keep the original routes.

3. Trajectory Ensemble

3.1. Overview

Multiple persons tracking over a camera network can easily fail when the appearance of a person changes due to environmental illumination and observation settings.

We introduce “Random Camera Drop” to generate several different camera networks which relax spatio-temporal cues partially and randomly. Here, we call each randomly dropped camera network as a “sub-network”. In different sub-networks, since the network topology is different, candidates for the tracklet matching are different. Therefore, we can obtain different tracking results from sub-networks. The proposed method integrates these tracking results by “Trajectory Ensemble” (Figure 4), which is an extended concept of Cluster Ensemble [17]. Note that it is a different approach to several existing methods based on combining multiple hypotheses [21, 22, 23].

Cluster Ensemble is an instance of Unsupervised Ensemble Learning. The main concept of Cluster Ensemble is the integration of data in terms of several “weak” clustering results clustered by changing the clustering criterion. We extend this concept to multiple person tracking over a camera network. The proposed method tracks persons over multiple sub-networks (“weak” tracking) and integrates the “weak” tracking results.

3.2. Sub-network generation by Random Camera Drop

From an initial camera network \( G = (C, E) \), each sub-network is generated by the following procedure.

1. A camera \( c_j \in C \) is selected randomly.
2. Let \( \mathcal{N}(c_j) \) be a set of adjacent cameras of camera \( c_j \).
3. For all pairs of \( c_k, c_l \in N(c_j), k \neq l \), add an edge \((c_k, c_l)\) to the network. Let the parameters of the edge \((c_k, c_l)\) be the sum of the parameters of edges \((c_k, c_j)\) and \((c_j, c_l)\). If the edge \((c_k, c_l)\) already exists, the network contains multiple edges \((c_k, c_l)\) assigned with different parameters.

4. Remove camera \( c_j \) and edges connected to \( c_j \).

We show an example of an initial camera network and sub-networks generated by the procedure in Figure 5.

### 3.3. Multiple Person Tracking over Each Sub-network

Persons are tracked over each sub-network \( G_s (s = 1, 2, \ldots, S) \). For these “weak” tracking, tracklets observed by the dropped cameras are ignored and not used.

By the “weak” tracking over each sub-network \( G_s \), labels \( L_s \) are assigned for all tracklets \( r_i (i = 1, 2, \ldots, M) \). The labels are identical to those of tracklets determined as the same person’s. For the tracklets which are not used for the “weak” tracking, a missing value (NA) is assigned. This label assignment \( L_s \) can be considered as a “weak” tracking result.

### 3.4. Trajectory Ensemble

The final result is calculated by integrating the “weak” tracking results using the extended concept of Cluster Ensemble [17]. The original Cluster Ensemble clusters samples into the same cluster when the samples are clustered into the same cluster in most “weak” clustering results. By clustering vectors whose elements are labels of weak clustering results, the consensus clustering results are obtained.

A number of labels are assigned to a tracklet \( r_i \) by the “weak” tracking results \( L_s (s = 1, 2, \ldots, S) \). For each tracklet \( r_i \), let \( \ell_i = (\ell_{i1}, \ell_{i2}, \ldots, \ell_{is}) \) be a vector consisting of labels assigned by the “weak” tracking results.

For a pair of tracklets \( r_{i1} \) and \( r_{i2} \), the pair can be considered that they are of the same person when the same labels are assigned in most of the “weak” tracking results. Therefore, by clustering all tracklets in terms of the similarity of the vectors \( \ell_{i1} \) and \( \ell_{i2} \), the consensus tracking results, namely, the final label assignment is obtained.

In this case, since elements of a vector \( \ell_i \) are labels, the difference between them are meaningless. Additionally, the vector contains the missing value NA. Therefore, we define a similarity function \( \text{sim}(\cdot, \cdot) \) of vectors \( \ell_{i1} \) and \( \ell_{i2} \) by modifying L_0-norm considering missing values as follows:

\[
\text{sim}(\ell_{i1}, \ell_{i2}) = \sum_{s=1}^{S} I(l_{i1s}, l_{i2s}) \\
I(a, b) = \begin{cases} 
1 & \text{if } a \neq b, a \neq \text{NA, } b \neq \text{NA} \\
0 & \text{otherwise}
\end{cases}
\] (1)

Figure 6. Virtual dataset generation. Once a simulated person is observed by a camera, an image corresponding to the person is sampled from the Shinpuhkan 2014 dataset.

By clustering based on this similarity function, the final label assignment for all tracklets are obtained. Here, we simply use agglomerative hierarchical clustering. We set the number of clusters for this as the average of the number of labels assigned for each “weak” tracking.

### 4. Evaluation

#### 4.1. Dataset

Although there are some publicly available image datasets for multiple person tracking across multiple cameras [9, 24, 25, 26, 27, 28], each of them contains just one or few scenarios of person movements. To evaluate on many scenarios of person movements, we generated several virtual datasets from a publicly available dataset, “Shinpuhkan 2014 dataset” [28]. As similar to Kokura et al. [29], we randomly generated the structure of a camera network and simulated the movement of persons. In the simulation, first, cameras are randomly placed in a scene, neighboring cameras are connected by edges, and then some edges are randomly removed. For each pedestrian, the source and the destination cameras are randomly selected and a path between them is randomly selected. We assumed that when a person passes in a camera view, the person is detected by the camera virtually. At that time, an image of the corresponding person is selected from Shinpuhkan 2014 dataset as shown in Figure 6.

As same as Shinpuhkan 2014 dataset, the number of persons in the simulation was 24 and the number of cameras was 16. An example of a generated camera network is shown in Figure 7. Parameters representing the traveling time distribution between each adjacent camera pair were determined based on the distance between them. The minimum traveling time of each adjacent camera pair varied depending on the distance between the cameras. The maximum of the minimum traveling time was about 5 minutes.
The intervals of new person arrival in the observation area were set between 20 seconds and 5 minutes. Once a person entered an observation area, the person traveled between more than 7 camera views. For each person, the traveling time between a camera pair is randomly selected according to the traveling time distribution of the camera pair. Under this setting, the average traveling time of a person over the camera network was about 15 minutes.

4.2. Features and comparison methods for the evaluation

Since developing a new image feature is out of the scope of this paper, we simply used an HSV color histogram for the image feature. To suppress affection of the background, we cropped the center half regions (half in both height and width) of input images before the feature extraction. To make it robust to illumination change, Adaptive Histogram Equalization [30] was applied to the input images before conversion to HSV color space.

The similarity of two tracklets \( f(r_{i_1}, r_{i_2}) \) was calculated by multiplying the appearance similarity and the likelihood of the temporal relationship as follows:

\[
f(r_{i_1}, r_{i_2}) = f_{\text{app}}(r_{i_1}, r_{i_2}) f_{\text{temp}}(r_{i_1}, r_{i_2}|e),
\]

(3)

where \( e \) denotes the edges between two cameras where the tracklets \( r_{i_1} \) and \( r_{i_2} \) were observed, respectively. The appearance similarity of two tracklets was determined by selecting the maximum appearance similarity between images of the two tracklets as follows:

\[
f_{\text{app}}(r_{i_1}, r_{i_2}) = \max_{m_1 \in r_{i_1}, m_2 \in r_{i_2}} f_{\text{app}}(m_1, m_2).
\]

(4)

The appearance similarity of two images was calculated by histogram intersection of HSV color histograms. The likelihood of the temporal relationship was calculated by the probability density function of the Gamma distributions whose parameters were assigned to the edge \( e \in e \) as follows:

\[
f_{\text{temp}}(r_{i_1}, r_{i_2}|e) = \max_{e \in e} f_{\text{pdf}}(r_{i_1}, r_{i_2}|e).
\]

(5)

4.3. Evaluation criterion

We generated five virtual datasets and evaluated on the averages of the five tracking results.

As the evaluation criterion, we used Adjusted Rand Index (ARI) [31, 32]. ARI measures the similarity between two label assignments. The value becomes 1 when all the label assignments are the same and it can be less than 0 when the assignments are worse than chance rate.

Let us assume label sets \( X = \{X_1, X_2, \ldots, X_A\} \) and \( Y = \{Y_1, Y_2, \ldots, Y_B\} \) are assigned to \( n \) elements, \( n_{i,j} \) be the number of elements where both \( X_i \) and \( Y_j \) are assigned. Let \( n_i \) and \( n_j \) be the number of elements where labels \( X_i \) and \( Y_j \) are assigned respectively. Then, ARI of label assignments \( X \) and \( Y \) is defined as follows:

\[
\text{ARI}(X, Y) = \frac{\sum_i \sum_j \binom{n_{i,j}}{2} - \left( \sum_i \binom{n_i}{2} \right) \left( \sum_j \binom{n_j}{2} \right) / \binom{n}{2} )}{\frac{1}{2} \left( \sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2} \right) - \left( \sum_i \binom{n_i}{2} \right) \left( \sum_j \binom{n_j}{2} \right) / \binom{n}{2} }.
\]

(6)

We used this criterion to measure similarities between tracking results (assigned labels) and the ground-truth (true person IDs).

In the evaluation, since the proposed method randomly drops cameras, the result changes among trials of Random Camera Drop. Therefore, we performed the tracking evaluation ten times and averaged ARIs over their results.

4.4. Comparative methods

For the “weak” tracking of the proposed method, we used a Minimum Cost Flow based method. As comparative methods, we selected a greedy method (Greedy) and a Minimum Cost Flow based method without Trajectory Ensemble (MinCostFlow). All of them used the same features and similarity comparison method for tracklets.

4.5. Result and discussion

The results are shown in Table 1. As shown here, we confirmed that the proposed method achieved the highest
Figure 8. Examples of tracking results.

Table 2. Averaged ARIs in different parameters.

<table>
<thead>
<tr>
<th>Number of dropped cameras $C_d$</th>
<th>Number of sub-networks $S$</th>
<th>50</th>
<th>100</th>
<th>150</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/16</td>
<td>0.548</td>
<td>0.556</td>
<td>0.556</td>
<td></td>
</tr>
<tr>
<td>7/16</td>
<td>0.573</td>
<td>0.572</td>
<td>0.576</td>
<td></td>
</tr>
<tr>
<td>9/16</td>
<td>0.575</td>
<td><strong>0.581</strong></td>
<td>0.576</td>
<td></td>
</tr>
<tr>
<td>11/16</td>
<td>0.457</td>
<td>0.526</td>
<td>0.578</td>
<td></td>
</tr>
<tr>
<td>13/16</td>
<td>0.314</td>
<td>0.455</td>
<td>0.495</td>
<td></td>
</tr>
</tbody>
</table>

ARI. Comparing with the MinCostFlow, we confirmed the effectiveness of employing Trajectory Ensemble.

The tracking results by the MinCostFlow, which is a comparative method, and Trajectory Ensemble, which is the proposed method, are shown in Figure 8 (i) and (ii), respectively. The tracking target switched to another person in the comparative method, while the proposed method could keep tracking the same person.

To evaluate the effectiveness of Trajectory Ensemble, we further evaluated the tracking results while changing parameters of the number of sub-networks and the number of dropped cameras. As shown in Table 2, parameters $C_d = 9$, $S = 100$ achieved the highest ARI.

The higher $C_d$ is, the more tracklets are not used for tracking. The more such tracklets are, the more missing values exist in vector $\ell$, which consists of labels assigned by weak tracking results. Since it is hard to compare two vectors which contain many missing values, the tracking accuracy degrades.

If we set a higher value to the number of sub-networks $S$, various sub-networks are generated. Therefore we consider that Trajectory Ensemble works more effectively. In the evaluation, setting $S = 100$ achieved the highest accuracy. However, since the number of combinations of dropped cameras is $\binom{10}{9} = 11,440$ when the total number of cameras is 16, we need further evaluation by tuning $S$ to higher values. On the other hand, when we set a higher value to $S$, since we need to process many sub-networks, it linearly increases the computation cost according to $S$. Therefore, we need to consider the trade-off between accuracy and computation cost.

5. Conclusion

Multiple person tracking over a camera network can fail by a temporal appearance change of the persons caused by environmental illumination and observation orientation. To solve this problem, comparing person images across not only adjacent cameras but also cameras multiple hops away is effective. However, such relaxation of spatio-temporal cues will also cause tracking failure. We introduced “Random Camera Drop” to generate different camera networks which relax the spatio-temporal cues partially and randomly. We also introduced a novel concept “Trajectory Ensemble”, an extension of unsupervised ensemble learning for multiple person tracking over a camera network problem. Using this concept, we integrated multiple “weak” tracking results over randomly dropped camera networks. We achieved the best performance among comparative methods on some virtually generated datasets.

Further analysis on the relationship between the ratio of dropping cameras and the number of randomly dropped camera networks is our future work.

Acknowledgement

Parts of this research were supported by MEXT, Grant-in-Aid for Scientific Research.

References


