Tracking Pedestrians across Multiple Cameras via Partial Relaxation of Spatio-Temporal Constraint and Utilization of Route Cue

Toru Kokura†, Yasutomo Kawanishi††, Masayuki Mukunoki†††,
Michihiko Minoh†††

Graduate School of Informatics, Kyoto University†
Institute of Innovation for Future Society, Nagoya University††
Academic Center for Computing and Media Studies, Kyoto University†††

Abstract. We tackle multiple people tracking across multiple non-overlapping surveillance cameras installed in a wide area. Existing methods attempt to track people across cameras by utilizing appearance features and spatio-temporal cues to re-identify people across adjacent cameras. However, in relatively wide public areas like a shopping mall, since many people may walk and stay arbitrarily, the spatio-temporal constraint is too strict, which results in matching errors. Additionally, appearance features can be severely influenced by illumination conditions and camera viewpoints against people, making it difficult to match tracklets by appearance features. These two issues cause fragmentation of tracking trajectories across cameras. We deal with the former issue by selectively relaxing the spatio-temporal constraint and the latter one by introducing a route cue. We show results on data captured by cameras in a shopping mall, and demonstrate that the accuracy of across-camera tracking can be significantly increased under considered settings.

1 Introduction

We address the problem of tracking pedestrians across multiple non-overlapping surveillance camera views in a wide area. It has a lot of commercial applications in practice and great importance for business growth. For example, in shopping malls, global tracking results can be valuable for finding similar spots and planning shop reallocation for sales growth. Since acquiring such global tracking results demands enormous time, labor and cost, a method for doing it automatically is required.

To this end, existing methods attempt to track pedestrians across camera views by utilizing appearance features and certain spatio-temporal cue to re-identify and associate pedestrians across adjacent camera views [1–5]. Appearance features are usually based on color histograms and texture descriptors, and the spatio-temporal cue is commonly based on the travel time between adjacent camera views, respectively.

In a shopping mall, since many pedestrians walk and stay arbitrarily no matter whether they are under the view of some camera(s) or outside of the views of all the cameras, their travel time between adjacent camera views varies from time to time. Additionally, appearance features can be influenced by camera viewpoint changes and illumination condition variations. These issues result in fragmented trajectories making difficult the global tracking across camera views, so that the existing methods fail to work when pedestrians walk and stay arbitrarily in wide public areas like a shopping mall. We deal with the former issue by relaxing the spatio-temporal cue selectively, and the latter one by introducing a route cue. Our goal is to make possible the global tracking in such important real scenarios.

2 Reated work

In this paper, we assume that we have already acquired cropped pedestrian image sequences by intra-camera pedestrian tracking for each camera. Each sequence of pedestrian images is called a *tracklet*. Tracklets are gained by conducting tracking within a camera view[6–8]

Tracking pedestrians across multiple camera views can be achieved if for each pair of tracklets acquired in adjacent camera views, we are able to correctly judge whether they correspond to the same pedestrian or not , in another word, to successfully perform person re-identification. To this end, Farenzena *et al.* [2] proposed an effective feature which accumulates various kinds of information including clothings' color and texture.

Javed *et al.* [3] presented a method for tracking pedestrians across two adjacent camera views by utilizing a spatio-temporal cue. The spatio-temporal cue consists of a spatio-temporal likelihood and a spatio-temporal constraint. The former is the likelihood of travel time between two adjacent camera views, and is described as a probability distribution. The spatio-temporal likelihood is used to compare similarity between tracklets. The latter is the constraint that the travel time in which a pedestrian moves from a camera view to an adjacent camera view must be within a time span, namely, between a given *minimum travel time* and a given *maximum travel time* for the camera pair. The spatio-temporal constraint is used to reduce matching candidates for computational efficiency.

Liana *et al.* [4] increased the accuracy of tracking across two adjacent camera views by optimizing matching of pedestrians. They track multiple pedestrians simultaneously and acquire the optimal matching. If we can match tracklets across each pair of adjacent camera views, tracking across multiple camera views will be possible, however, once a matching error occurs, a tracking trajectory may get fragmented or wrongly connected, which leads to the failure in tracking over a wide area.

Song *et al.* [5] proposed a method for tracking pedestrians across multiple camera views. This method achieved higher matching accuracy between each camera pair by utilizing additional information; pedestrian appearance and observing time of tracklets acquired by other cameras than the camera pair, however, it requires vast computational cost. Chen *et al.* [1] presented a method for tracking pedestrians across multiple camera views with less computational complexity by restricting matching candidates with a spatio-temporal constraint.

Alahi *et al.* [9] introduced a concept of data association [10] for the across-camera tracking, and proposed a method to optimize all trajectories. In the method, optimal trajectories are acquired by calculating the posterior probability of each considerable trajectory.

3 Tracking pedestrians using a spatio-temporal cue

For each tracklet $r_i(i \in \mathbb{N})$, those methods[1, 5] focus on only a set of tracklets which were acquired before r_i was acquired. They choose tracklets for the set using a spatio-temporal constraint. By using the spatial constraint, they exclude tracklets acquired by a camera view not adjacent to the camera view where tracklet r_i was acquired. They also exclude tracklets whose travel time is not in a given time span. The time span is characteristic to each camera pair.

The left tracklets after the exclusion process are the candidate set for the tracklet r_i denoted by H_i . They compute similarities between tracklet r_i and every tracklet in H_i. For the computation of the similarity, they use two kinds of information; one is the spatio-temporal likelihood, more precisely, the likelihood of transition time; and the other is the appearance likelihood, i.e., the similarity of appearance features. If the similarity between tracklet r_i and its most similar tracklet in the candidate set H_i is greater than a given threshold, the two tracklets are considered to be matched. Otherwise, no matching is found and the tracklet r_i is considered to be the first tracklet of a trajectory, that is, the starting tracklet of a pedestrian in the camera network.

Such kind of methods have the following two drawbacks.

Firstly, when an observed pedestrian is significantly delayed between a pair of adjacent camera views, the travel time may go out of the given time span. A delay often happens when there are places visitable between the adjacent camera views, such as stores, restrooms, signboards, smoking areas, and exhibitions.

Secondly, except in a very simple environment such as a straight road, it is often difficult to install cameras to observe pedestrians from similar viewpoints. Thus, each pedestrian's appearance varies across camera views. In general, additionally, lighting condition varies with observation time because of weather, light intensity and existence of the sunlight. Since these factors may cause significant appearance change, they may lead to the wrong judgment that the corresponding tracklet does not exist in the candidate set, even though it actually does.

We give examples of these two problems. To make it easy for understanding, we focus on only three cameras from a possible complex camera network as shown in Fig. 1. The views of Cameras 1 and 2, and those of Cameras 2 and 3 are adjacent, and there is no direct path linking the views of Cameras 1 and 3 without passing other views.

The two problems we've concerned are summarized below:

- Problem(i): Because of the delay, the tracklet which should be matched is excluded from the candidate set.

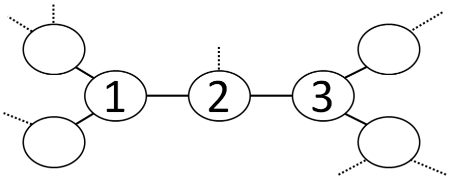

Fig. 1. adjacency of camera views in a camera network

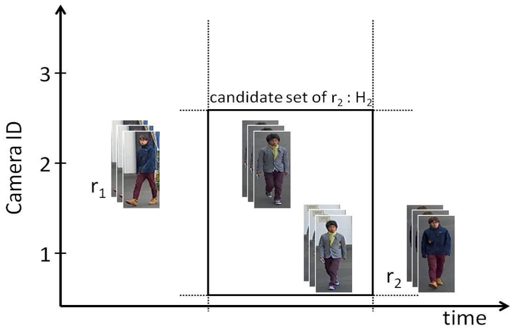

Fig. 2. Problem(i) : delay

- Problem(ii): Because of appearance variation, the tracklet which should be matched is considered not to exist in the candidate set, even though it does.

An example of the problem(i) is shown in Fig. 2. In the figure, each tracklet is shown as a set of stacked images of a pedestrian. The figure indicates that the tracklets r_1 and r_2 are acquired when the pedestrian P_A passed through the views of Camera 2 and Camera 1 respectively. The pedestrian corresponding to the other two tracklets are different from the pedestrian P_A. For tracklet r_2, the candidate set H_2 consists of tracklets existing in the rectangle in the figure. H_2 is the set to an extent satisfying the spatio-temporal constraint of the tracklets. The pedestrian got delayed, and therefore tracklet r_1 is not included in the rectangle. This results in a matching error.

An example of the problem(ii) is illustrated in Fig 3. The figure indicates that the tracklets r_3, r_4 and r_5 are acquired when the pedestrian P_B passed through the views of Cameras 1, 2 and 3 respectively. The pedestrian corresponding to the rest one tracklet is different from the pedestrian P_B. In this case, the candidate set H_5 consists of tracklets existing in the rectangle as marked in the figure. Although the tracklet which should match to r_5 is actually included in the candidate set, the pedestrian looks different because she is observed from different viewpoints by Camera 2 and Camera 3. Thus, appearances of the tracklets r_4 and r_5 are not similar enough, making the matching of them very difficult.

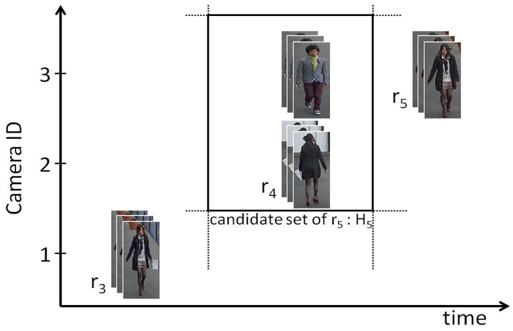

Fig. 3. Problem(ii) : appearance variation

4 Tracking pedestrians via partial relaxation of spatio-temporal cue and utilization of route cue

4.1 How to solve the problems

The proposed method copes with the two problems via the following two ideas respectively;

- Idea(i): By partially relaxing the spatio-temporal constraint only against pairs of tracklets whose appearance similarities are significantly high, we include the corresponding tracklets in the candidate set. The tracklet pairs are chosen from every camera in the camera network, not only from adjacent cameras.
- Idea(ii): By utilizing a route cue, we predict that the corresponding tracklet actually exists in the candidate set.

Details of the ideas are given below. Let r_i be a tracklet denoted by $r_i = (\mathbf{f}_i, s_i, e_i, c_i)$, where $\mathbf{f}_i$ is the sequence of the pedestrian's appearance features extracted from the cropped pedestrian image sequence within the view of camera c_i, s_i is the time when the first frame of this sequence is captured, e_i is the time when the last frame of the sequence is captured. Using this notation, the candidate set H_i of a tracklet r_i mentioned in the previous section can be defined as

$$\begin{aligned} H_i = \{r_m \mid (c_i, c_m) \in E, \\ t_{\min}(c_i, c_m) < s_i - e_m < t_{\max}(c_i, c_m)\}, \end{aligned} \tag{1}$$

where $t_{\min}(c_i, c_m)$ denotes the minimum travel time between the views of camera c_i and camera c_m, and $t_{\max}(c_i, c_m)$ denotes the maximum travel time between the two cameras, and E denotes a set of camera pairs which have a direct path between them.

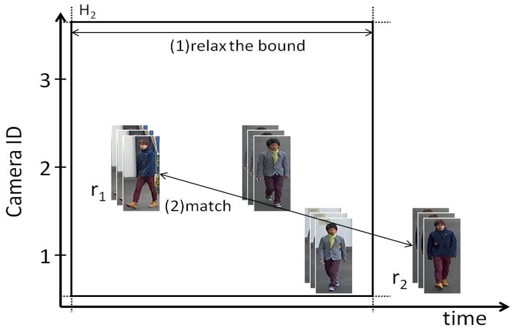

Fig. 4. Idea(i) : partial relaxation of spatio-temporal cue

Here, we assume that when a pedestrian moved from one camera view to another camera view, a tracklet r_j is acquired from the former camera view and another tracklet $r_i (i \neq j)$ is from the latter. We focus on the situations when the problem(i) or problem(ii) occurs between the two tracklets.

Firstly, by selectively relaxing the spatio-temporal constraint in matching tracklets, we tackle the problem(i), where the corresponding tracklet deviates from the candidate set. When a tracklet pair that does not fulfill the spatio-temporal constraint because of delay, the tracklet r_j is excluded from the candidate set H_i. If the similarity of the two tracklets r_i and r_j is significantly high, we relax the spatio-temporal constraint for the tracklet r_j, namely setting $t_{\max}$ as ∞, to include the tracklet into the candidate set H_i. In this case, the spatio-temporal likelihood is not used to calculate similarity between tracklets r_i and r_j, and only appearance likelihood is used. This enables us to match two tracklets regardless of the duration of the delay.

The example of this process is shown in Fig. 4. The situation of this figure is the same as the situation of Fig. 2. In this figure, by relaxing the spatio-temporal constraint, the tracklet r_1 will be found in the relaxed candidate set H_2. Since appearance similarity of tracklets r_1 and r_2 is significantly high, they can be matched correctly.

Secondly, we tackle the appearance variation by introducing a route cue. The route cue is a novel constraint that when a pedestrian is observed by two cameras whose views are not adjacent, the pedestrian should be observed by the other cameras whose views exist on a must-pass route between the two camera views. This constraint is an enhanced version of the spatio-temporal constraint, and ensures that the tracklet r_j actually exists in the candidate set. Here, let r_k, r_j and r_i be the tracklets acquired when a pedestrian P_C passed through the views of three cameras respectively, and $s_k < s_j < s_i$. This process narrows the number of elements of H_i, and reduces the existence probability of a tracklet $r_l \in H_i$ where similarity(r_i, r_l) and similarity(r_j, r_l) are higher than similarity(r_i, r_k) and similarity(r_j, r_k). Here, similarity() means the similarity between two tracklets.

Purging such a tracklet r_l from the candidate set makes it possible to match r_j correctly. Additionally, when the existence of the tracklet r_j is ensured, it can be matched even if r_j's similarity against r_i and r_k is lower than the matching threshold. Even if appearance change happens, r_j belongs to the same person as r_k and r_j, thus r_j's similarity against r_i and r_k is still high to some extent.

Here, we will describe the process in detail. We consider the situation where there is a tracklet r_k acquired in the view of a camera $c_k(\neq c_i)$ before a tracklet r_i is acquired, and the views of cameras c_i and c_k are not adjacent. Let r_m be a candidate tracklet of the matching partner of r_i, the tracklet r_m must fulfill following two constraints;

- s_m and e_m of the tracklet r_m must fulfill following inequities:

$$\begin{cases} s_m - e_k > t_{\min}(c_k, c_m) \\ s_i - e_m > t_{\min}(c_m, c_i). \end{cases} \tag{2}$$

  These inequities mean that the tracklet r_m is acquired at a time between the time when the tracklets r_k and r_i are acquired with consideration of the minimum travel time between pairs of the cameras..
- The view of camera c_m must exist on a route from the view of camera c_k to the view of camera c_i.

We consider only the tracklets fulfilling these two constraints as the candidate set. The tracklet which has the highest similarity to r_i in the candidate set H_i is matched with r_i.

The tracklet r_k very similar to r_i can be found by executing the matching with partial relaxation of the spatio-temporal constraint, if the similarity between r_i and r_k is higher than a given threshold. If the camera views which the tracklets c_k and c_i are acquired are not adjacent, we conduct the matching mentioned above.

By following the two constraints, we can reduce the number of elements in the candidate set H_i. Additionally, we can ensure that the candidate set H_i includes the tracklet r_j, thus even if similarity(r_i, r_j) and similarity(r_k, r_j) are lower than the given threshold, we can match the tracklets.

An example of this process is shown in Fig. 5. The situation of this figure is as same as the situation of Fig. 3. Because the appearances of tracklet r_3 and r_5 are similar enough, they are matched by relaxed spatio-temporal constraint. Since the views of camera $c_3(=1)$ and $c_5(=3)$ are not adjacent, by using the route cue, the corresponding tracklet to the tracklet r_5 must exist in the area enclosed by the rectangle in the figure. By optimally selecting the tracklet which has the highest similarity to r_3 and r_5 from the candidate set H_5, the tracklet r_4 can be correctly matched even if the similarity of the tracklet r_5 and r_4 is lower than the given threshold.

4.2 Proposed tracking procedure

Including these two ideas, we propose a novel tracking method. One of the novelties of the method is its deliberate procedure. We describe the procedure below.

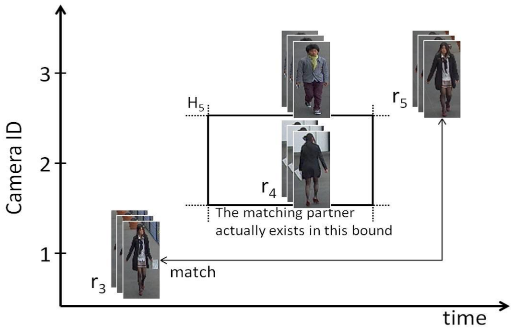

Fig. 5. Idea(ii) : utilization of route cue

We apply matching procedures using idea (i)(ii) to a tracklet set before applying the existing method[1]. Here, in order to apply matching with utilization of the route cue, the tracklet r_k must be found beforehand. Therefore, we should first perform matching with partial relaxation of the spatio-temporal cue. Before the two above-mentioned procedures, pairs of tracklets which have neither problem(i) nor problem(ii) should be matched with each other. This is easily achieved by searching for tracklet pairs whose spatio-temporal likelihood and appearance similarity are high. Here, the spatio-temporal likelihood between two tracklets means how similar the two tracklets are when considering a transition time distribution between two cameras which acquired the two tracklets.

Accordingly, the steps of matching should be the followings;

- Step 1: For each pair of tracklets acquired by adjacent cameras, if the wighted average of their spatio-temporal likelihood and appearance likelihood are higher than a given threshold, match them.
- Step 2: By relaxing the spatio-temporal constraint for all tracklets, if there are pairs of tracklets whose similarity of appearance is significantly high, match them.
- Step 3: Execute matchings utilizing the route cue.
- Step 4: By applying the existing method[1], try to match tracklets while fixing the matched pairs in above steps.

We empirically tuned the thresholds used in the step 1 and step 2 so strictly that false matchings don't happen. Matching tracklets with low similarity invokes false matchings, thus strict thresholding can prevent them.

Each process conducts locally optimal matching, and after all 4 steps finish, an optimal solution is obtained.

5 Benchmark Datasets

To evaluate the robustness of the proposed method against the delay and the appearance change, we should evaluate on many datasets collected from various environments which have different rates of the delay and appearance change.

However, these datasets are difficult to prepare, especially when controlling the rates of the delay and the appearance change are considered. Instead of trying hard to build such “real” datasets, we generate multiple “virtual” datasets and use them for evaluation. If we prepare some “real” datasets, very similar datasets must be included in the virtual datasets.

Here, we describe how to generate the virtual datasets from the Shinpuhkan 2014 dataset collected from surveillance cameras mounted in a real shopping mall[11]. We denote the original Shinpuhkan 2014 dataset by $\tilde{D}$. This dataset $\tilde{D}$ consists of tracklets $\tilde{R}$ of 24 pedestrians, which are captured by 16 non-overlapping camera views. These cameras cover both shade areas and sunny areas within the mall.

We consider the rate of the delay and the appearance change happen as parameters of a virtual dataset and generate various virtual datasets which contain tracklets generated virtually while changing the parameters. Here, we first describe the parameters for the virtual dataset, and then we describe how to generate a virtual dataset with these parameters.

5.1 Parameters for the virtual dataset

We want to generate virtual datasets with different rates of the delay and appearance change. We define a virtual dataset by $D(\beta, N_{cam}, N_{sun}; \tilde{D})$, where β denotes the parameter of the delay happening, N_{cam} denotes the number of camera views we use in the dataset and N_{sun} denotes the parameter controlling the appearance change.

To simulate the delay of pedestrians, we parametrize the probability of the delay happening at $\beta \in [0, 1]$. In the virtual dataset, the pedestrians delay at a probability of β. Once a pedestrian delays, the pedestrian spent more t_{delay} seconds between two camera views. t_{delay} is sampled randomly from a given uniform distribution.

To simulate the appearance change, we parameterize the probability of the appearance change happening by the number of camera views N_{sun} which are under the sunlight. The greater the value of N_{sun}/N_{cam} is, the more often appearance changes happen. It is because the appearance of a pedestrian is heavily affected by the illumination condition, especially whether it is under the sunlight or not. Though also the direction variation of people against cameras causes appearance variations, we didn't control it. It is because the dataset $\tilde{D}$ originally contains the direction variation, which makes the direction variation automatically contained in generated datasets, and it is difficult to control how much is contained.

If a pedestrian is observed in two camera views which observe shaded areas, appearances of the pedestrian in the two images captured by the two cameras are similar. However, if the pedestrian is observed under the sunlight in one camera view, appearance of the pedestrian will be different from the one observed under the other camera. We show those examples in Fig. 6. Three images in Fig. 6 represent the same person. Fig. 6 (a) is captured in the shaded area and Fig. 6 (b)(c) are captured under the sunlight. We can see the big difference between Fig. 6 (a) and the others. Appearance variation also exists between Fig.6 (b) and Fig.6 (c) caused by the direction of the sunlight against the person.

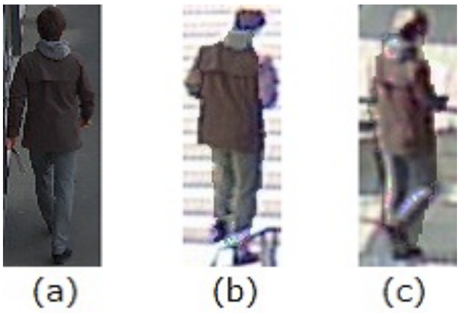

Fig. 6. Illumination change and appearance variation

How to generate a virtual dataset We generate a virtual dataset by simulating pedestrians' walk over a virtual camera network. Therefore, the dataset generation consists of two parts: the virtual camera network generation and the simulation of pedestrians' walk, namely, the virtual tracklets generation.

Virtual camera network generation First, we divide the vertices $\tilde{V}$ in the camera network $\tilde{G}$ of the original dataset $\tilde{D}$ into two groups $\tilde{V}_{sun}$ and $\tilde{V}_{shade}$ by lighting condition of the camera. We select N_{sun} vertices from $\tilde{V}_{sun}$ and $N_{cam} - N_{sun}$ vertices from $\tilde{V}_{shade}$ and we denote the set of the selected vertices by V.

Then, we generate N_{path} pedestrian path candidates over V. For each subset $V_1, \dots, V_{N_{path}}$ of V, we generate a Hamiltonian paths $\Pi_1, \dots, \Pi_{N_{path}}$. Letting all the edges contained in the generated Hamiltonian paths be an edge set E, we get a camera network $G = (V, E)$.

Virtual tracklets generation The original dataset $\tilde{D}$ consists of the tracklets $\tilde{R}$ of 24 pedestrians. We generate a set of virtual tracklets R by generating virtual tracklets for each pedestrian in $\tilde{D}$.

We randomly ordered the pedestrians and denote them as $\{p_1, \dots, p_{24}\}$. For each pedestrian $p_k \in \{p_1, \dots, p_{24}\}$, we randomly selected a path $\pi_k \in$

$\{\Pi_1, \ldots, \Pi_{N_{path}}\}$. For each camera view corresponding to a vertex in the path π_k, we select a tracklet of the person p_k, and we select a sequence of real tracklets of the pedestrian p_k along the path π_k from $\tilde{R}$.

To generate virtual tracklets, we need to change the observation time for each tracklets. The observation time of tracklets of a pedestrian p_k is determined by the time point $t_0^{p_k}$ when the pedestrian is observed in the camera network at the first time, the time span $\Delta t_i^{p_k}$ in which the pedestrian passes through the camera view corresponding to the vertex $v_i^{p_k}$, and the time span $\Delta t_{i,i+1}^{p_k}$ in which the pedestrian travels from the camera view corresponding to the vertex $v_i^{p_k}$ to the next camera view corresponding to the vertex $v_{i+1}^{p_k}$ of the camera network along the path $\pi_k = (v_1^{p_k}, \ldots, v_{N_{\pi_k}}^{p_k})$.

We set a given time point to $t_0^{p_0}$. Then we determine the time points $t_0^{p_k}$ iteratively by sampling the time span $(t_0^{p_k} - t_0^{p_{k+1}})$ from a given gamma distribution. Gamma distribution is often used to model a time span in which something occurs.

For the time span $\Delta t_i^{p_k}$, we simply used the original time span which the pedestrian p_k actually spent to pass through the camera view.

To determine the time span $\Delta t_{i,i+1}^{p_k}$, we need to model the travel time of pedestrians between two camera views. Here, we model it by sum of the time span $t_{ordinary}(v_i^{p_k}, v_{i+1}^{p_k})$, which pedestrians ordinary spent to travel from the camera view denoted by the vertex $v_i^{p_k}$ to the next camera view denoted by the vertex $v_{i+1}^{p_k}$ and the time span t_{delay} which denotes the delay time as;

$$\Delta t_{i,i+1}^{p_k} = t_{ordinary}(v_i^{p_k}, v_{i+1}^{p_k}) + \delta t_{delay}, \tag{3}$$

where $t_{ordinary}(v_i^{p_k}, v_{i+1}^{p_k})$ is sampled from a given gamma distribution for each pair $(v_i^{p_k}, v_{i+1}^{p_k})$ and t_{delay} is sampled from a given uniform distribution between 5 minutes to 60 minutes. δ is a controlling parameter sampled from a Bernoulli distribution with the parameter β.

Then we can calculate the observation time for each virtual tracklet from $t_0^{p_k}$, $\Delta t_i^{p_k}$, and $\Delta t_{i,i+1}^{p_k}$, that is, we can calculate the entrance time s_i and the exit time e_i of a virtual tracklet r_i. Finally, we get a set of virtual tracklets R.

6 Evaluation

6.1 Experiment configurations

We generated various virtual datasets by the procedure introduced in the previous section to show the effectiveness of our proposed method.

Virtual dataset parameters We changed the delay parameter β from 0% to 100% by 10%. Camera networks were randomly generated for each virtual dataset with the parameters $N_{cam} = 5$ and $N_{path} = 4$. We had two paths randomly generated and the rest two set to be inverse of them. We set the start vertices of the former two paths to be the same vertex, and also set the end vertices of the paths to be the same vertex. We also changed the appearance parameter N_{cam} from 0 to 5. Parameters of every distribution were determined empirically. We generated 50 different datasets for each parameter set of β and N_{sun}.

Calculation of similarity between tracklets We calculated the similarity between tracklets as a weighted average of the appearance likelihood and the temporal likelihood.

We calculated appearance likelihood $l_{app}(r_i, r_j)$ between tracklets r_i and r_j as follows;

$$l_{app}(r_i, r_j) = \min_{a,b} d(\mathbf{f}_i^a, \mathbf{f}_j^b), \tag{4}$$

where $\mathbf{f}_i^a$ is the appearance feature extracted the a_{th} frame of tracklet r_i, and d is the Bhattacharyya coefficient, which is used to calculate distance between histograms or probability distributions. We described the appearance feature of pedestrian images by using Weighted Color Histogram [2] .

Methods for comparison We compared our method with the traditional method [1] which was proposed by Chen *et al.*. The method is equivalent to the step 4 of our method. The the spatio-temporal constraint works well under low rate of delay happening, but it makes the result worse under high rate of delay happening. To evaluate the effect of the spatio-temporal constraint, we compared the two methods with 2 different settings; with/without the spatio-temporal constraint in the step 4. For the "without the spatio-temporal constraint" setting, we set the maximum travel time t_{max} to ∞.

- traditional : The traditional method [1] without the spatio-temporal constraint.
- proposed : The proposed method without the spatio-temporal constraint.
- traditionalST : The traditional method [1] with the spatio-temporal constraint.
- proposedST : The proposed method with the spatio-temporal constraint.

6.2 Evaluation Criterion

We evaluated the tracking results by measuring the correctness for each sequence of tracklets. As the evaluation criterion, we utilized F-measure, which is usually used for the accuracy evaluation in the field of information retrieval. To define F-measure in this evaluation, we introduce the label series $\boldsymbol{l}$ over a tracking result, where matched tracklets should have the same label and not matched tracklets should have different labels.

F-measure is defined as the harmonic average of the precision and the recall. Let the series of these labels $\boldsymbol{l} = (l^1, \ldots, l^N)$, where l^i represents the label assigned to a tracklet r_i. We defined F-measure as follows;

$$F\text{-}measure(\boldsymbol{l}) = \frac{2 \cdot precision(\boldsymbol{l}) \cdot recall(\boldsymbol{l})}{precision(\boldsymbol{l}) + recall(\boldsymbol{l})}. \tag{5}$$

We defined the precision as the percentage of correct matchings, and defined the recall as the percentage of conducted matchings in ground truth as follows;

$$precision(\boldsymbol{l}) = \frac{|TP(\boldsymbol{l})|}{|L(\boldsymbol{l})|}, \tag{6}$$

$$recall(\boldsymbol{l}) = \frac{|TP(\boldsymbol{l})|}{|L(\hat{\boldsymbol{l}})|}. \tag{7}$$

where $TP(\boldsymbol{l})$ and $FP(\boldsymbol{l})$ denote the set of true positives and the set of false positives respectively, and $L(\boldsymbol{l})$ is the linkage set. $\hat{\boldsymbol{l}}$ is the ground truth of the label series.

$TP(\boldsymbol{l})$ and $FP(\boldsymbol{l})$ are defined using the linkage set $L(\boldsymbol{l})$ as follows respectively;

$$TP(\boldsymbol{l}) = \{x | x \in L(\boldsymbol{l}) \ \wedge \ x \in L(\hat{\boldsymbol{l}})\}, \tag{8}$$

$$FP(\boldsymbol{l}) = \{x | x \in L(\boldsymbol{l}) \ \wedge \ x \notin L(\hat{\boldsymbol{l}})\}. \tag{9}$$

We defined the linkage set $L(\boldsymbol{l})$ as follows;

$$L(\boldsymbol{l}) = \{x = (g_q^j, g_q^{j+1}) | \ q < \max_i l^i \ \wedge \ (g_q^j, g_q^{j+1}) \in G_q\}, \tag{10}$$

where

$$G_q = \{i | l^i = q\}. \tag{11}$$

Based on the definition, we calculated F-measure against all of the 4 methods listed in the previous subsection.

6.3 Result

We show the result of the case $N_{sun} = 0$ and $N_{sun} = 2$ in Fig. 7 and Fig. 8 respectively. Both figures are plotted with F-measure as the vertical axis and the delay probability β as the horizontal axis. Each line represents the average of the results of 50 trials.

Whatever value the delay probability β is, F-measure of the proposed method exceeds that of the existing method. The difference of F-measure between the proposed method and the existing method grows greater as the delay probability β gets larger. This suggests that our method can deal with the problem (i) : the problem of delay happening.

Furthermore, whether there are cameras observing the sunshine in the datasets or not, F-measure of the proposed method is greater than that of the existing method. From this, we confirmed that our method can deal with the problem (ii) : the problem of appearance variation.

Based on the above results, we can conclude that our tracking method can deal with both problems of delay happening and appearance changes.

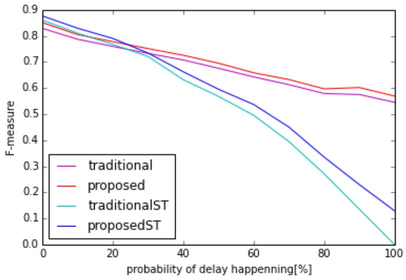

Fig. 7. The result where $N_{sun} = 0$

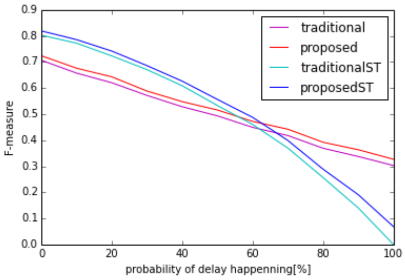

Fig. 8. The result where $N_{sun} = 2$

7 Conclusion

This paper presents a method for tracking pedestrians across multiple non-overlapping camera views by selectively relaxing the spatio-temporal cue and introducing the route cue. We showed that the proposed method can consistently improve the tracking accuracy under different parameter settings of delay and appearance change.

The possible future work includes modeling the travel time statistically in the situation where a person has a heavy delay and setting the criteria of relaxation ratio for the spatio-temporal cue.

Acknowledgement

This work was supported by "R&D Program for Implementation of Anti-Crime and Anti-Terrorism Technologies for a Safe and Secure Society," Funds for integrated promotion of social system reform and research and development of the Ministry of Education, Culture, Sports, Science and Technology, the Japanese Government.

References

1. Chen, K.Y., Huang, C.L., Hsu, S.C., Chang, I.C.: Multiple objects tracking across multiple non-overlapped views. In Ho, Y.S., ed.: PSIVT (2). Volume 7088 of Lecture Notes in Computer Science., Springer (2011) 128–140
2. Farenzena, M., Bazzani, L., Perina, A., Murino, V., Cristani, M.: Person re-identification by symmetry-driven accumulation of local features. In: CVPR, IEEE (2010) 2360–2367
3. Javed, O., Rasheed, Z., Shafique, K., Shah, M.: Tracking across multiple cameras with disjoint views. In: ICCV, IEEE Computer Society (2003) 952–957
4. Lian, G., Lai, J.H., Zheng, W.S.: Spatial-temporal consistent labeling of tracked pedestrians across non-overlapping camera views. Pattern Recognition **44** (2011) 1121–1136

5. Song, B., Chowdhury, A.K.R.: Robust tracking in a camera network: A multi-objective optimization framework. J. Sel. Topics Signal Processing **2** (2008) 582–596
6. Hofmann, M., Haag, M., Rigoll, G.: Unified hierarchical multi-object tracking using global data association. In: Performance Evaluation of Tracking and Surveillance (PETS), 2013 IEEE International Workshop on, IEEE (2013) 22–28
7. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–8
8. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1201–1208
9. Alahi, A., Ramanathan, V., Fei-Fei, L.: Socially-aware large-scale crowd forecasting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 2203–2210
10. Bar-Shalom, Y., Tse, E.: Tracking in a cluttered environment with probabilistic data association. Automatica **11** (1975) 451–460
11. Kawanishi, Y., Wu, Y., Mukunoki, M., Minoh, M.: Shinpuhkan2014: A multi-camera pedestrian dataset for tracking people across multiple cameras. In: 20th Korea-Japan Joint Workshop on Frontiers of Computer Vision. (2014) 232–236