Object tracking across non-overlapping views by learning inter-camera transfer models

Xiaotang Chen*, Kaiqi Huang, Tieniu Tan

Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, No.95 ZhongGuoCun East Street, HaiDian District, P.O. Box 2778, Beijing, PR China

Abstract

In this paper, we introduce a novel algorithm to solve the problem of object tracking across multiple non-overlapping cameras by learning inter-camera transfer models. The transfer models are divided into two parts according to different kinds of cues, i.e. spatio-temporal cues and appearance cues. To learn spatio-temporal transfer models across cameras, an unsupervised topology recovering approach based on N-neighbor accumulated cross-correlations is proposed, which estimates the topology of a non-overlapping multi-camera network. Different from previous methods, the proposed topology recovering method can deal with large amounts of data without considering the size of time window. To learn inter-camera appearance transfer models, a color transfer method is used to model the changes of color characteristics across cameras, which has an advantage of low requirements to training samples, making update efficient when illumination conditions change. The experiments are performed on different datasets. Experimental results demonstrate the effectiveness of the proposed algorithm.

1. Introduction

As the number of cameras used in the wide area video surveillance increases, multi-camera object tracking plays a more important role in understanding and analyzing the scenes. It is a challenging problem. Especially when there are non-overlapping views among cameras, the observations of the same object under different cameras are often widely separated in time and space. Based on this fact, the problem of object tracking across non-overlapping cameras is quite different from single camera object tracking or overlapping multi-camera object tracking.

Over the last few years, many approaches have been proposed to solve this problem. Two kinds of cues are usually employed: spatio-temporal cues across cameras and appearance cues of objects.

1.1. Spatio-temporal cues across cameras

To model the spatio-temporal relationships across cameras, various strategies are proposed to recover the topology graph of the non-overlapping multi-camera network. The topology graph usually has three main factors: firstly, the nodes, from which objects enter or exit; secondly, the links between nodes, indicating the connectivity of each two nodes and corresponding to the real paths in the environment which can be followed by objects; thirdly, the transition time distribution for each link across cameras, demonstrating the probability of transition time of an object moving from one node to another. If an object leaves the FoV of a camera at a moment, then we can predict the object’s re-appearance after some time under certain cameras using the knowledge of topology.

Generally, the nodes are defined as entry/exit zones in the FoVs of cameras, which can be learned by clustering the starting or ending points of trajectories observed by single camera tracking [2–4], or defined as single cameras [5]. To estimate the existence of link between two nodes and the transition time distribution for each link, the methods can be put into two categories. The first one is based on solving the correspondence problem [3,6,7] or object tracking [8]. Javed et al. [6] use Parzen windows to estimate the inter-camera space-time probabilities from training data, assuming the correspondences are known. These methods usually have good estimations of the transition time distributions, however, solving the problem of correspondences or object tracking itself is complicated and challenging.

The second one does not require establishing correspondences between observations or object tracking [2,4,9,10]. Makris et al. [9] calculate a cross-correlation function of two signals which represent the arrival event sequence observed at one node and the departure event sequence observed at the other node in a time
window. Ideally if a link exists, then the cross-correlation has a clear peak around the most popular transition time. However, in most cases, the peak is not so clear due to the large variance of transition time of true correspondences and a large number of false correspondences which result from a large traffic flow or a long time window. To make the peak sharp, methods [4,10] add similarity in appearance to weight the cross-correlation model. These methods are usually easy to be implemented. However, few of them consider the estimation of transition time distributions. To estimate the transition time distribution, Zou et al. [10] fit K Gaussian functions to a normalized cross-correlation using the EM algorithm, which is not proper for considering both true and false correspondences.

Based on cross-correlation functions, we present a topology recovering method by decreasing the large variance of transition time of true correspondences, which can compensate for the influence caused by large-scale false correspondences to a certain degree. Thus, the proposed topology recovering method can deal with large amounts of data or a long time window. It estimates the transition time distribution for each valid link based on an iteration, which is different from other cross-correlation based methods. In addition, the proposed topology recovering method avoids solving the problem of establishing correspondences between non-overlapping views, making it easy to be implemented.

1.2. Appearance cues of objects

For the appearance cues, methods generally use one or multiple kinds of features to represent the appearance of an object. However, the appearance varies a lot across cameras, which is influenced by many factors, such as the illumination, camera properties, viewpoints, poses and nonuniform clothing, as shown in Fig. 2. Various appearance descriptors are proposed to be robust to the challenges mentioned above. A major color spectrum histogram (MCSH) [11] is introduced to represent a moving object by using a normalized geometric distance between two points in the RGB space. D’Angelo et al. present a probabilistic color histogram (PCH) to describe the color appearance of the object [12], which is built by using a fuzzy K-Nearest Neighbors classifier. Yu et al. [13] model the appearance based on spatial/color statistical features. To incorporate structural information and achieve invariance to motion and pose, they use an additional feature of path-length besides color features. Wang et al. [14] introduce the concept of shape and appearance context by modeling the spatial distribution of the appearance relative to each of the object parts. Instead of directly exploiting robust features in object representation, several approaches [15–17] use machine learning tools (i.e. AdaBoost algorithm) to learn the similarity or distance between any two objects based on color and texture histogram features.

To the best of our knowledge, color-based features are the most widely used features in solving the problem of tracking pedestrians across non-overlapping cameras. Colors are easily influenced by illumination changes. Thus, alleviating the influence of illumination variance across cameras becomes necessary and important. Methods to this problem can be divided into two groups. The first group is color transfer across cameras. One of the typical methods is learning brightness transfer functions (BTFs) [18], which handles the change in observed colors of an object as it moves from one camera to another. Javed et al. [18] show that all brightness transfer functions from a given camera to another camera lie in a low dimensional subspace and demonstrate that this subspace can be used to compute appearance similarity. They learn BTFs for each pair of cameras from the training data by using probabilistic principal component analysis. However, their method relies on training subjects with a good range of brightness values to give an accurate mean BTF (MBTF), which implicitly assumes both extensive color variations on object clothing and very large number of objects being sampled [19]. To extend the work in [18], Prosser et al. [19] compute a cumulative BTF (CBTF) by accumulating the brightness values of the whole training set before the BTF computation instead of computing a BTF for each training pair, thus, brightness values that are not common in the training set are still preserved. Mazzeo et al. [20] compare the performance of two different brightness transfer functions, i.e. the MBTF and the CBTF, which demonstrates quite similar behaviors of the two methods when the simple association problem has to be solved. Jeong et al. [21] learn color transfer functions by operating

Fig. 1. The topology graph of a non-overlapping multi-camera network. Nodes are entry/exit zones labeled by different numbers. The green solid arrows denote visible paths within the field of view (FoV) of each camera, which can be detected by single camera tracking. The red dotted arrows represent valid links between nodes across cameras, which depend on methods of recovering the topology to estimate the existence and corresponding transition time distributions. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)
on chromaticity samples to increase the temporal stability of the transfer function between any pair of cameras. Their method also needs a training phase with known correspondences.

Although these color transfer approaches [18,19,21] can improve the object recognition accuracy, they only work well when given a suitable training set with a good range of brightness values and known correspondences between training samples. Thus a small training set or not highly reliable correspondences can cause degradation of performance. Moreover, once the unknown functions for any pair of cameras are learned during the training phase, the illumination conditions at both cameras should remain unchanged. When the illumination condition changes at either camera, the functions are not applicable until they are learned again based on newly collected training samples. Hence, these color transfer approaches are not applicable to real systems of uncontrollable illumination conditions due to their high requirements to the training set.

For the second group of approaches, the aim is to correct images independent of cameras, i.e., correcting images for deviations from a canonical light source using computational color constancy algorithms. One well-known color constancy method is gamut mapping proposed by Forsyth [22]. The algorithm computes the mapping relationships from an observed gamut into the canonical gamut and derives the illuminant color based on the relationships. Gijsenij et al. [23] extend gamut mapping to incorporate the statistical nature of images whose focus is on the local n-jet describing the derivative structure of an image. However, these color constancy algorithms are complex and require an image dataset with known light sources for calibration, which is very difficult to obtain for non-overlapping multi-camera systems. Only color constancy algorithms which have no need of a training dataset with known light sources can be applied to multi-camera systems to solve the problem of illumination variance across cameras, such as gray-world [24], max-RGB [25], shades of gray [26] and gray-edge [27].

In this paper, to alleviate the influence of illumination variance across cameras, we model the inter-camera appearance transformations based on a color transfer method proposed in [28] belonging to the first group. This method imposes one image's color characteristics on another in the $L^a b$ color space, so color transfer between cameras is realized by color characteristic transfer (described as CCT in the following sections), which applies the color characteristics of one observation of an object at one camera to another observation of the same object at a different camera. Different from other color transfer approaches, a single pair of corresponding observations can meet the requirement of the CCT algorithm, making update efficient in multi-camera systems.

The main contributions in this paper lie in two aspects: (1) an unsupervised topology recovering approach is proposed for learning the inter-camera spatio-temporal transfer models, which can deal with large amounts of data or a long time window and avoid solving the problem of establishing correspondences between non-overlapping views; (2) CCT method [28] is applied to model the inter-camera appearance transformations, which can update the models efficiently when illumination conditions change.

This paper is organized as follows. Section 2 gives an overview of the multi-camera object tracking algorithm. Section 3 proposes a topology recovering method to learn the inter-camera spatio-temporal transfer models. In Section 4, the CCT method is explained in detail and used to model the inter-camera appearance transformations. Experimental results and conclusions are given in Sections 5 and 6 respectively.

2. Multi-object tracking across non-overlapping views

In this paper, we propose a novel algorithm to solve the problem of object tracking across non-overlapping views. Both inter-camera spatio-temporal relationships and appearance relationships are modeled to provide more information. Fig. 3 shows the flowchart of the method.

Considering the application that tracking pedestrians across non-overlapping cameras, we only deal with one traffic pattern (pedestrians), assuming that the transition time between each two cameras follows a normal distribution. In the preprocessing phase, departure object sequences, arrival object sequences and the corresponding time sequences are collected from off-line videos. Based on the data, an unsupervised topology recovering algorithm is performed to learn spatio-temporal transfer models, such as transition time distributions. To learn the initial inter-camera appearance transfer models, a color characteristic transfer (CCT) method [28] is applied according to the color characteristics of two corresponding hand-labeled observations or full images of FoVs of cameras when hand-labeled observations are unavailable. If more than one pair of observations are provided, the learned appearance transfer model (described as CCT model) is the average.

In the multi-camera object tracking system, we assume that the problem of single camera object tracking has already been solved. When an object $Object^t$ enters the FoV of one camera at time $t$, it needs to be recognized as the same object which has left the FoV.
of some camera or identified as a new object if it appears in the camera network for the first time. Firstly, the topology recovering algorithm provides inter-camera spatio-temporal cues to select candidate objects. Specifically, given Seq, the blob sequence of Object, we match it against blob sequences of labeled objects (e.g., Object) leaving the connected entry/exit zones during the time window $[t-\mu_{ZbZa}-3\sigma_{ZbZa}, t-\mu_{ZbZa}+3\sigma_{ZbZa}]$, where $Z$ and $Z'$ denote the entry/exit zone which Object enters and the entry/exit zone which Object exits from respectively. $N(\mu_{ZbZa}, \sigma_{ZbZa}^2)$ is the learned transition time distribution of the path from $Z'$ to $Z$. Note that the weight of the mean square deviation $\sigma_{ZbZa}$ is set to 3 because $P(\mu-3\sigma < X \leq \mu + 3\sigma)$, the probability between $\mu - 3\sigma$ and $\mu + 3\sigma$, is more than 99.7% according to characteristics of the normal distribution. Secondly, to alleviate the influence of illumination variance across cameras, learned inter-camera appearance transfer models are applied to estimate the appearance of each blob in Seq at the camera which Object exits from before feature extraction. Further operations are done on the estimated appearances of Object rather than its observed appearances.

Thirdly, the problem of object re-recognition across cameras is done following the work in [29], which formulates it as a binary classification problem based on an AdaBoost classifier. A camera variable indicating the entry/exit cameras is treated as the feature to represent different pairs of cameras. For each labeled object, an adaptive model describing the intraclass similarity is computed.
and used as an adaptive threshold to deal with the outputs of AdaBoost. If no one successfully matches Seqa, then Object a is identified to be a new object and a new label is assigned to it. Otherwise, it retains the same label as the one with a maximal matching score. Finally, the tracking results can be treated as new training samples for updating the appearance transfer models. As illumination changes over time, for each pair of connected cameras, the most reliable pair of observations corresponding to the highest matching score is used to update the inter-camera appearance transfer models using CCT method at regular intervals.

The technical details of how to learn the inter-camera spatio-temporal and appearance transfer models are discussed in the next two sections.

**Fig. 5.** The procedure of the CCT method.

**Fig. 6.** An example of color transfer using CCT method. (a) The original source image. (b) The target image. (c) the transferred image.

**Fig. 7.** Experimental setups. (a) The network topology. (b) The Gaussian models of similarity. The green area demonstrates the possibility of failures in object matching, which is true to fact. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

**Fig. 8.** Recovered topology graphs. (a) By the method in [9]. (b) By our method without similarity cues. (c) By the method in [4]. (d) By our method with similarity cues.
Fig. 9. Estimated cross-correlations. (a) By the method in [9]. (b) By our method without similarity cues ($R_n^{21}(c_n), n=22$). (c) By the method in [4]. (d) By our method with similarity cues ($R_n^{21}(c_n), n=15$).
3. Learning inter-camera spatio-temporal transfer models

The inter-camera spatio-temporal transfer models are learned by recovering the topology of the camera network. In this paper, the entry/exit zones in the FoVs of cameras are represented as nodes in the topology. The locations of nodes are estimated by clustering the starting or ending points of trajectories observed by single camera tracking, similar to the node estimation in [2–4]. As mentioned before, directly estimating the topology from cross-correlations suffers from large-scale false correspondences and the large variance of transition time of true correspondences.

We assume that the transition time of true correspondences is within a restricted range of some popular transition time while transition time caused by false correspondences is irregular and widespread in the whole time axis. Based on this reasonable assumption, an N-neighbor accumulated cross-correlation function is computed for every two nodes across cameras to reduce the influence caused by large variance of transition time of true correspondences, which makes the peak clear and sharp.

Given node i and node j from two cameras, we observe objects departing at node i and arriving at node j within a time window which is long enough. Let \( D_i(t) \) and \( A_j(t) \) denote the departure time sequence at node i and arrival time sequence at node j respectively. Then the N-neighbor accumulated cross-correlation function of \( D_i(t) \) and \( A_j(t) \) is computed as follows:

\[
R_{ij}^N(\tau_n) = \sum_{\tau_0 = -\infty}^{\tau_n} \sum_{\tau = -\infty}^{r_n} E[D_i(t) \cdot A_j(t + \tau_0)]
\]

where \( R_{ij}^N(\tau_0) \) represents the cross-correlation function of \( D_i(t) \) and \( A_j(t) \).

As mentioned previously, we assume that the transition time across cameras follows a normal distribution. An iteration is performed to estimate the connectivity for each pair of nodes, and the parameters of the transition time distribution as well. The main idea of this iteration is to find the most steady and frequent peak in \( R_{ij}^N(\tau_n) \) rather than a very clear peak in the cross-correlation. The details of the proposed method are summarized in Algorithm 1. Parameters such as \( M_1 \), \( M_2 \), \( T_1 \) and \( T_2 \) are set empirically. \( M_1 \) and \( M_2 \) set the upper and lower limits of \( n \) respectively. \( M_1 \) should be big when the distance between nodes is very long, and we usually set it to be no less than 100. \( T_1 \) is a threshold above which a valid link is believed to exist, which is empirically set to be no less than 0.9, as some invalid links may be introduced using a low-value \( T_1 \). The parameter \( T_2 \) allows a small fluctuation of the average transition time, for which the best setting is from 20 to 100. \( N(\tau_n, n) \) is the estimated transition time distribution for a valid link, where \( \tau_n \) demonstrates the average transition time.

**Algorithm 1.** The estimation of valid links and transition time distributions.

1: \( \text{input: } D_i(t), A_j(t) \)
2: \( \text{Initialize } Link = False, MaxList(r) = 0 \text{ where } r \geq 0. \)
3: \( \text{for } n \text{ from } 1 \text{ to } M_1 \text{ do} \)
4: \( \text{Compute } R_{ij}^N(\tau_n) \text{ according to Eq. (1).} \)
5: \( \text{find } \tau_n^* = \arg \max R_{ij}^N(\tau_n) \)
6: \( \text{MaxList}(\tau_n^*) = \text{MaxList}(\tau_n^*) + 1 \)
7: \( \text{if } n \geq M_2 \text{ then} \)
8: \( \text{StepMaxList}(\tau) = \sum_{\tau_n^*}^{\tau} \text{MaxList}(\tau), \tau \geq T_2 \)
9: \( \text{if } \text{ratio} \geq T_1 \text{ then} \)
10: \( \text{Link} = True, \text{break} \)
11: \( \text{end if} \)
12: \( \text{end if} \)
13: \( \text{end for} \)
14: \( \text{output: } N(\tau_n^*, n), \text{ when } \text{Link} = True \)

Fig. 4 gives an example using the proposed method to detect a valid link under the condition that the pedestrian flow is large and the time window is very long (far longer than the average transition time 885), while general cross-correlation [9] fails. There are two clear peaks in the cross-correlation using the method [9], while neither of them is around actual average transition time. Estimating transition time distributions by the EM algorithm based on the cross-correlations is improper because of so much noise, while using the proposed method, the transition time distribution of this link is estimated to be N(898, 139), very close to the ground truth N(885, 148).

Combined with similarity cues, the proposed method can be applied to the weighted cross-correlation model [4,10], by transforming the computation of \( R_{ij}^N(\tau_n) \) from Eqs. (1) to (2):

\[
R_{ij}^N(\tau_n) = \sum_{\tau_0 = -\infty}^{\tau_n} \sum_{\tau = -\infty}^{\infty} \text{Sim}(O_i(t), O_j(t + \tau_0)), \quad \tau_n \geq 0
\]
where \( O_i(t) \) and \( O_j(t) \) denote the departure object sequence at node \( i \) and arrival object sequence at node \( j \) respectively. \( \text{Sim}(i,j) \) measures the similarity between each object in \( O_i(t) \) and each object in \( O_j(t) \). For this similarity measurement, any effective feature can be used.

4. Learning inter-camera appearance transfer models

In order to model the appearance variance across cameras, the inter-camera appearance transfer models are learned by applying the CCT method [28] to the multi-camera object tracking system. When a typical three channel image is represented in any of the most well-known color spaces, i.e. RGB color space, there will be correlations between the different channels’ values [28]. Correlations between different color channels make the operation of color correction or color transfer more complicated. Thus, an orthogonal color space without correlations between channels is preferred for it does not cause undesirable cross-channel artifacts when different operations are applied in different color channels with some confidence. Ruderman et al. [30] develop a new color space (named \( l\alpha\beta \) color space) by minimizing correlation between channels for many natural scenes. As there is little correlation between color channels in \( l\alpha\beta \) space, CCT method borrows one of these spaces to \( l\alpha\beta \) space (named \( l\alpha\beta \) color space) by minimizing correlation between channels for many natural scenes. As there is little correlation between color channels in \( l\alpha\beta \) space, CCT method borrows one of these spaces to \( l\alpha\beta \) space where \( l\alpha\beta \) space is used to impose color characteristics of the target image are applied to the source image without alleviating the influence of illumination variance.

After both the source and target images are converted to \( l\alpha\beta \) space, the mean and standard deviation are computed for each color channel separately. Let \( m_l^i, m_\alpha^i, m_\beta^i \) and \( \sigma_l^i, \sigma_\alpha^i, \sigma_\beta^i \) be the means of different color channels of the source (target) image. \( \bar{s}_l^i, \bar{s}_\alpha^i, \bar{s}_\beta^i \) represent the standard deviations of different color channels of the source (target) image. Then the color characteristics of the target image are applied to the source image as follows:

\[
\begin{align*}
\bar{s}_l^t &= \frac{s_l^t}{s_l^s}(l_i - m_l^i) + m_l^i \\
\bar{s}_\alpha^t &= \frac{s_\alpha^t}{s_\alpha^s}(\alpha_i - m_\alpha^i) + m_\alpha^i \\
\bar{s}_\beta^t &= \frac{s_\beta^t}{s_\beta^s}(\beta_i - m_\beta^i) + m_\beta^i
\end{align*}
\]

where \( l_i, \alpha_i, \beta_i \) are the color values corresponding to different channels of the source (transferred) image. Finally, the transferred image is converted back to RGB color space via LMS space using Eqs. (6) and (7). An example of color transfer using CCT method is shown in Fig. 6:

\[
\begin{bmatrix}
l_L \\
l_M \\
l_S \\
\end{bmatrix} =
\begin{bmatrix}
\frac{1}{3} & 0 & 0 \\
0 & \frac{1}{3} & 0 \\
0 & 0 & \frac{1}{3}
\end{bmatrix}
\begin{bmatrix}
l_t \\
\alpha_t \\
\beta_t
\end{bmatrix}
\]

CCT algorithm uses a simple statistical analysis to impose one image’s color characteristics on another, thus the CCT model for each pair of connected cameras can be learned by two

Fig. 13. CMC curves of different methods. Despite evaluating the performance of various color transfer or color constancy methods, we also give the result on original images without alleviating the influence of illumination variance.
observations of the same objects viewed from these two cameras respectively. In the preprocessing phase, full images of FoVs of cameras are used to initialize CCT models when hand-labeled objects are not provided. Obviously, there are nine critical parameters constituting the CCT model \( s_1^l, s_1^r, s_2^l, s_2^r, m_1^l, m_1^r, m_2^l, m_2^r, m_3^l \). Once this CCT model is learned, it can be applied to dealing with blobs observed under different cameras to alleviate the influence of illumination variance across cameras. Specifically, given a newcomer Object\(^a\) detected at Camera \( x \), we estimate the appearance of Object\(^a\) at Camera \( y \) based on the observed

---

**Fig. 12.** Some examples of processed images using different methods. The first two rows are the original images from Cam. \( x \) and Cam. \( y \) respectively.
appearance of Object\textsuperscript{a} at Camera \( x \) and the CCT model before matching it against candidate objects which have left Camera \( y \). The color transfer is performed as follows:
\[
\begin{align*}
\ell^a_y = & \frac{\sigma^2_a}{\sigma^2_y} (\ell^a_x - m^a_l) + m^a_t \\
\alpha^a_y = & \frac{\sigma^2_a}{\sigma^2_a} (\alpha^a_x - m^a_y) + m^a_t \\
\beta^a_y = & \frac{\sigma^a}{\sigma^a} (\beta^a_x - m^a_y) + m^a_t
\end{align*}
\]
where \( \ell^a_y, \alpha^a_y, \) and \( \beta^a_y \) are the color values corresponding to different channels of the observed (estimated) appearance of Object\textsuperscript{a}. Note that different pairs of cameras have different CCT models. The CCT model used in this case is from Camera \( x \) to Camera \( y \), which is learned by taking the appearance of an object observed at Camera \( x \) as the source image and the appearance of the same object at Camera \( y \) as the target image.

5. Experimental results

We conduct multiple experiments to give a thorough performance test of the proposed topology recovering method, color transfer across cameras using CCT method and object tracking across non-overlapping cameras.

The first experiment is to estimate the performance of the proposed topology recovering method on simulated data. The simulation is based on a multi-camera network shown in Fig. 7 (a). In the network, the nodes in a closed dotted curve belong to the same camera. The departure time of 1000 moving objects follows a uniform distribution \( U(0, 1500) \), and the transition time between nodes follows a normal distribution \( N(300, 20) \). Each object is equally likely to arrive at any connected node after leaving any node (in the same camera or a different camera), \( M_1 \) and \( M_2 \) are set to 100 and 10 respectively in this case. The threshold \( T_1 \) is set to 0.98, and \( T_2 \) is set to 20. First, the proposed method is compared with a benchmark method [9]. In our experiments, the parameter \( \omega \) in method [9] is set to 2, which controls the peak detection threshold. Then, we extend the proposed method into the weighted cross-correlation model according to Eq. (2), and compare it with the method [4]. The similarity between the same objects and between two different objects follows \( N(0.7, 0.1) \) and \( N(0.4, 0.1) \) respectively, shown in Fig. 7(b).

Fig. 8 shows the recovered topology graphs using different methods. Since we focus on estimating the connectivity between every two nodes across cameras, the links within the same FoVs are neglected. Previous methods [4, 9] have unsatisfactory performance, due to the large amounts of traffic data and a long time window. Extended to the weighted cross-correlation model, the proposed method fully recovers the topology of the simulated network, as shown in Fig. 8(d). Our method also recovers the transition time distribution, not only an average transition time for each valid link. Estimated cross-correlations for the link from node 2 to node 1 are shown in Fig. 9. Only the proposed topology recovering method successfully detects this valid link, and returns the average transition time after the \( m \)th iteration.

The second experiment is conducted on real data collected from off-line videos to further demonstrate the effectiveness of the proposed topology recovering method. The real-life experimental setup of the network is shown in Fig. 10. The network has three non-overlapping cameras, containing two, two and four entry/exit zones respectively. We use two-hour-long videos to recover the topology of the network. Gaussian Mixture Model is used to detect every pedestrian entering or leaving each node and the corresponding arrival time or departure time is also recorded. We do not set limits for the size of time window, so all the departure events and arrival events happened in the videos are used to recover the topology. \( M_1 \) and \( M_2 \) are set to 1000 and 10 respectively in this case. The threshold \( T_1 \) is set to 0.9, and \( T_2 \) is set to 50.

The recovered topology is shown in Fig. 11. The learned transition time distributions (black) provide good estimates of the ground truth. Although there is a real path from node 5 to node 4, the proposed method fails to detect this valid link because there are only two pedestrians walking from node 5 to node 4 among 53 departure events detected in node 5 and 100 arrival events detected in node 4 in the whole videos. It is very difficult to detect this valid link based on so few true correspondences. Estimated cross-correlations for the link from node 2 to node 3 using the proposed method and the method [9] are shown in Fig. 4. Previous method [9] can hardly work in this case, which has
a long time window and a large pedestrian flow, while the proposed method detects the valid link successfully.

The third experiment is conducted on a pedestrian dataset to demonstrate the effectiveness of alleviating the influence of illumination variance across cameras by learning appearance transfer models using CCT method [28]. The dataset contains 67 pedestrians which are collected from off-line videos. Each person has two images viewed from Camera x and Camera y respectively. There is a large illumination variance between these two cameras, which has great influence on the pedestrians' appearances. The CCT method is compared with six different kinds of color transfer or color constancy methods: MBTF [18], CBTF [19], gray-edge [27], gray-world [24], max-RGB [25] and shades of gray [26]. For each method, the pedestrian re-recognition accuracy is tested based on images processed by this method to evaluate its ability to solve the problem of alleviating the influence of illumination variance. For each image, only the areas which the object occupies are considered. Some examples of processed images are shown in Fig. 12. In this experiment, 7 pedestrians are randomly selected to learn the color transfer models for MBTF, CBTF and CCT, while the other 60 pedestrians are taken as testing samples for all the methods. The major color spectrum histogram (MCSH) [11] is used to measure the similarity between two objects. Images from Camera x and Camera y are used as the probe set and the gallery set respectively. The results are presented using cumulative matching characteristic (CMC) curves, as shown in Fig. 13.

Note that color transfer methods have directions while color constancy methods deal with single images without considering the appearance relationships between cameras. For example, in Fig. 13, CCT(x→y) means we transfer the images viewed from Camera x according to their color characteristics and the learned CCT model from Camera x to Camera y. The corresponding CMC curve is evaluated by taking the transferred images as the probe set and the original images viewed from Camera y as the gallery set. MBTF has an unsatisfied performance because the training set is too small for it. However, CCT method performs best using the same training set. Different color constancy methods give very close results and make small improvements over the original images. For each color transfer method, the performance varies a lot with the transfer direction. Thus, making use of both directions may provide more information for object recognition.

To evaluate the tracking performance of the proposed multi-camera object tracking algorithm, we conduct the fourth experiment on a real-life multi-camera network. As shown in Fig. 1, the camera network has five non-overlapping cameras containing one indoor camera and four outdoor cameras. In this camera network, non-overlapping areas between cameras are very large and illumination variance across cameras is great. At first, off-line videos with a length of about 20 min are collected. Gaussian Mixture Model is used to detect every pedestrian entering or leaving each node and the corresponding arrival time or departure time is also recorded. All the departure events and arrival events happened in the videos are used to recover the topology. Parameter settings are the same as the second experiment.

The recovered topology is shown in Fig. 14. Although there is a real path between node 6 and node 8, it fails to be detected as there are too few pedestrians walking from node 6 to node 8 or from node 8 to node 6. To learn initial appearance transfer models across cameras, we take the full images as the source and target images for each pair of connected cameras like Fig. 6. The proposed object tracking algorithm is tested on videos [31] containing 14 pedestrians and 44 transitions across cameras. Forty-three transitions of them are correctly re-recognized. The only error is occurred when Person 7 enters Camera 1 from Camera 2, as he is mistaken for Person 13. Some examples of tracking results are shown in Fig. 15.

6. Conclusions

In this paper, we have presented a solution for non-overlapping multi-camera object tracking by learning inter-camera spatio-temporal and appearance transfer models. The inter-camera spatio-temporal transfer models are learned by automatically recovering the topology of a non-overlapping camera network. Unlike previous cross-correlation based work, our topology recovering method can deal with large amounts of data without considering the size of time window. The connectivity for each pair of nodes is estimated based on the stability and frequency of peaks in the N-neighbor accumulated cross-correlations, which is more robust. The appearance transfer models across cameras are learned by learning inter-camera color characteristic transformations using CCT method. Experimental results have demonstrated that the proposed algorithm performs well in tracking multiple objects across non-overlapping cameras based on inter-camera transfer models, which are effectively learned by the topology recovering method and CCT method.

Experimental results have also shown that the performance of color transfer varies a lot with the transfer direction between camera. Thus, the relationships between different transfer directions can be further exploited and made use of to improve the performance. Future work will focus on large-scale multi-camera networks.

Conflict of Interest

None declared.

Acknowledgements

This work is funded by the National Natural Science Foundation of China (Grant nos. 61105009, 61135002), the National Key Technology R&D Program (Grant no. 2012BAH07B01), the National Basic Research Program of China (Grant no. 2012CB316302).

References


Xiaotang Chen received the BE degree in Xidian University in 2008. She is currently a PhD candidate in the National Laboratory of Pattern Recognition, Chinese Academy of Sciences. Her research interests are pattern recognition, image processing, and computer vision.

Kaiqi Huang received the MS degree in Nanjing University of Science & Technology and the PhD degree in Signal and Information Processing from Southeast University. Then he joined in National Laboratory of Pattern Recognition, Chinese Academy of Sciences and now he has been a professor.

Tieniu Tan received the MSc and PhD degrees in electronic engineering from Imperial College of Science, Technology and Medicine, London, United Kingdom, in 1986 and 1989, respectively. He is currently a professor and the director of the National Laboratory of Pattern Recognition, Chinese Academy of Sciences.