Learning the Three Factors of a Non-overlapping Multi-camera Network Topology

Xiaotang Chen, Kaiqi Huang, and Tieniu Tan

National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences
{xtchen,kqhuang,tnt}@nlpr.ia.ac.cn

Abstract. In this paper, we propose an unsupervised approach for learning the three factors of the topology of a non-overlapping multi-camera network, which are nodes, links, and transition time distributions. It is a cross-correlation based method. Different from previous methods, the proposed method can deal with large amounts of data without considering the size of time window. The connectivity between nodes is estimated based on the N-neighbor accumulated cross-correlations, as well as the transition time distribution for each link. Furthermore, integrated with similarity cues, the proposed method can be extended into weighted cross-correlation models for better performance. Experimental results both on simulated and real-life datasets demonstrate the effectiveness of the proposed method.

Keywords: Topology recovering, Transition time distribution, Camera network, Non-overlapping views.

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 cameras have non-overlapping views among them, the observations of the same object under different cameras are often widely separated in time and space. Thus lack of spatio-temporal cues between cameras makes it different from single camera object tracking or overlapping multi-camera object tracking.

To compensate for lack of spatio-temporal cues across cameras, various strategies are proposed to recover the topology graph of the non-overlapping multi-camera network. Figure 1 shows a topology graph of a 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 we observe an object

C.-L. Liu, C. Zhang, and L. Wang (Eds.): CCPR 2012, CCIS 321, pp. 104–112, 2012.

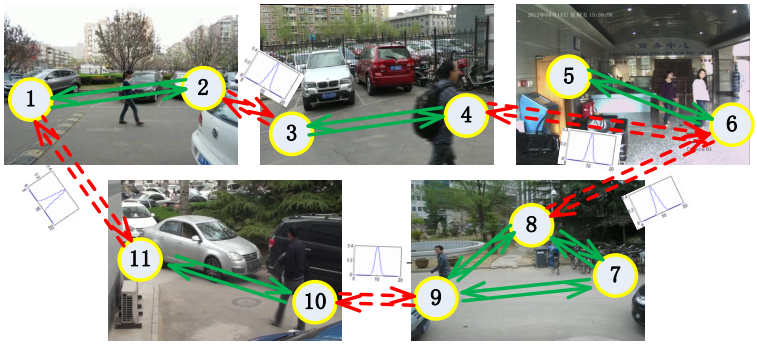

Fig. 1. The topology graph of a non-overlapping multi-camera network. Nodes are entry/exit zones labeled by different numbers. The green solid arrows denote visible paths within the field of view (FoV) of each camera, which can be detected by single camera tracking. The red dotted arrows represent valid links between nodes across cameras, which depend on methods of recovering the topology to estimate the existence and corresponding transition time distributions.

leaves the field of view of a camera at a moment, we can predict the object's re-appearance after some time under certain cameras using the knowledge of topology.

2 Related Work and Contributions

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 [1–3], or defined as single cameras [4]. 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 [2, 5, 6] or object tracking [7]. Javed, O. *et al* [5] 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 [1, 3, 8, 9]. Makris, D. *et al* [8] 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 [3, 9] 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, X. *et al* [9] 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 function, our method focuses on 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 method can deal with large amounts of data or a long time window. Based on an iteration, our method can estimate the transition time distribution for each valid link, which is different from other cross-correlation based methods. In addition, the proposed method avoids solving the problem of establishing correspondences between non-overlapping views, making it easy to be implemented.

3 Proposed Method

In this paper, we represent the entry/exit zones in the FoVs of cameras as nodes in the topology, and estimate the locations of nodes by clustering the starting or ending points of trajectories observed by single camera tracking, similar to the node estimation in [1–3]. As we 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, we compute an N-neighbor accumulated cross-correlation function 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 $A_j(t)$ denotes 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:

$$\begin{aligned} R_n^{i,j}(\tau_n) &= \sum_{\tau_0=\tau_n-n}^{\tau_n+n} R_0^{i,j}(\tau_0) \\ &= \sum_{\tau_0=\tau_n-n}^{\tau_n+n} E[D_i(t)\cdot A_j(t+\tau_0)] \\ &= \sum_{\tau_0=\tau_n-n}^{\tau_n+n} \sum_{t=-\infty}^{+\infty} D_i(t)\cdot A_j(t+\tau_0), \tau_n \geq n \end{aligned} \tag{1}$$

where $R_0^{i,j}(\tau_0)$ represents the cross-correlation function of $D_i(t)$ and $A_j(t)$.

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

input: $D_i(t)$, $A_j(t)$
Initialize $Link = False$, $MaxList(\tau) = 0$, where $\tau \geq 0$.
for n from 1 to M_1 **do**
  Compute $R_n^{i,j}(\tau_n)$ according to Eq.1.
  find $\tau_n^* = arg\max_{\tau_n} R_n^{i,j}(\tau_n)$
  $MaxList(\tau_n^*) \leftarrow MaxList(\tau_n^*) + 1$
  if $n \geq M_2$ **then**
    $StepMaxList(\tau') = \sum_{\tau=\tau'-T_2}^{\tau'+T_2} MaxList(\tau), \tau' \geq T_2$
    $ratio = \max(StepMaxList)/\Sigma(MaxList)$
    if $ratio \geq T_1$ **then**
      $Link = True$, **break**
    end if
  end if
end for
output: $N(\tau_n^*, n)$, when $Link = True$

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 nodes across cameras follows a normal distribution. We process an iteration 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 $R_n^{i,j}(\tau_n)$ rather than a very clear peak in the cross-correlation. The details of the proposed method are summarized in Algorithm 1, where M_1 and M_2 set the upper and lower limits of n respectively. T_1 is a threshold above which a valid link is believed to exist. The parameter T_2 allows a small fluctuation of the average transition time. $N(\tau_n^*, n)$ is the estimated transition time distribution for a valid link, where τ_n^* demonstrates the average transition time.

Figure 2 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 [8] fails. There are two clear peaks in the cross-correlation using the method [8], 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 [3, 9], by transforming the computation of $R_n^{i,j}(\tau_n)$ from Eq.1 to Eq.2:

$$R_n^{i,j}(\tau_n) = \sum_{\tau_0=\tau_n-n}^{\tau_n+n} \sum_{t=-\infty}^{+\infty} Sim(O_i(t), O_j(t+\tau_0)), \tau_n \geq n \tag{2}$$

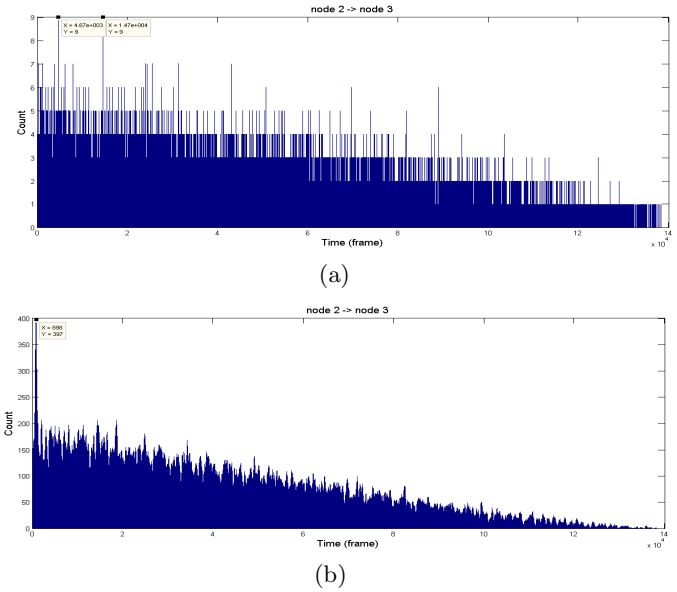

Fig. 2. The estimated cross-correlations. (a) By the method in [8]; (b) By our method without similarity cues ($R_n^{2,3}(\tau_n)$, $n = 139$).

where $O_i(t)$ and $O_j(t)$ denotes the departure object sequence at node i and arrival object sequence at node j respectively. $Sim(*,*)$ 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 Experimental Results

We evaluate the performance of the proposed method both on simulated data and real-life data.

4.1 Simulated Experiments

The simulation is based on a multi-camera network shown in Figure 3 (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 is 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 [8]. In our experiments, the parameter ω in method [8] is set to 2, which controls the peak detection threshold. Then, we extend the proposed method into the weighted

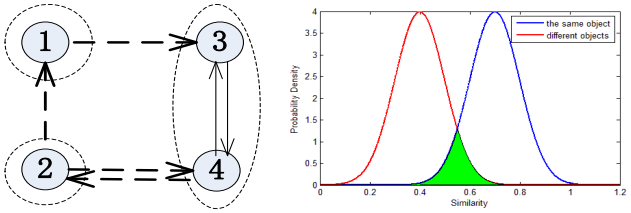

Fig. 3. Experimental setups. (a) The network topology; (b) The Gaussian models of similarity. The green area demonstrates the possibility of failures in object matching, which is true to fact.

cross-correlation model according to Eq.2, and compare it with the method [3]. 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 Figure 3 (b).

Figure 4 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 [3, 8] have bad 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 Figure 4 (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 Figure 5. Only our method successfully detects this valid link, and returns the average transition time after the nth iteration.

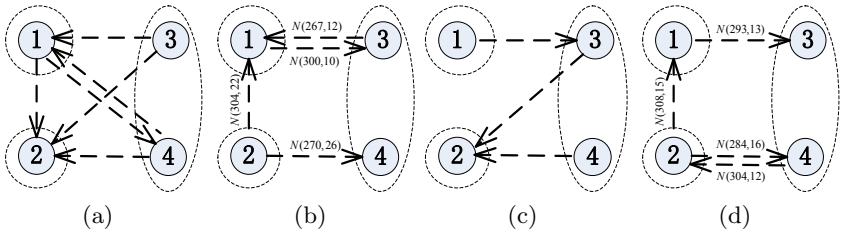

Fig. 4. The recovered topology graphs. (a) By the method in [8]; (b) By our method without similarity cues; (c) By the method in [3]; (d) By our method with similarity cues.

4.2 Real-Life Experiments

The real-life experimental setup of the network is shown in Figure 6. 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.

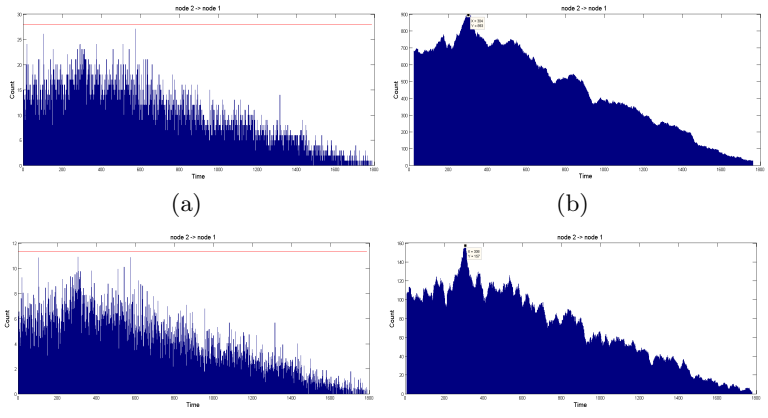

Fig. 5. The estimated cross-correlations. (a) By the method in [8]; (b) By our method without similarity cues ($R_n^{2,1}(\tau_n)$, $n = 22$); (c) By the method in [3]; (d) By our method with similarity cues ($R_n^{2,1}(\tau_n)$, $n = 15$).

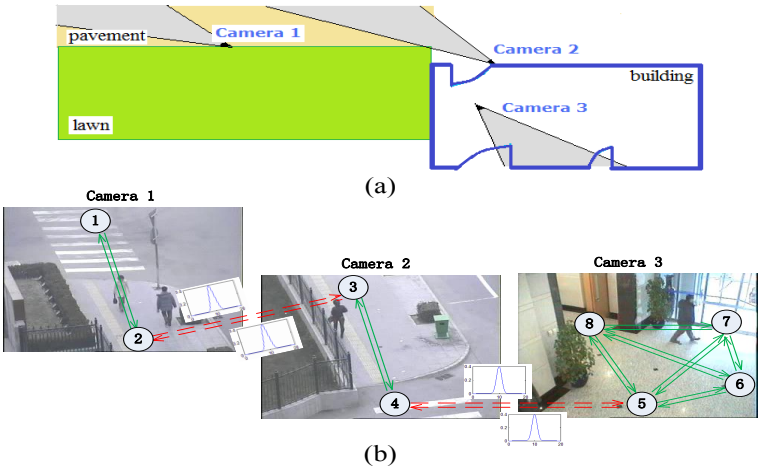

Fig. 6. (a) The layout of a non-overlapping multi-camera network; (b) The corresponding topology graph

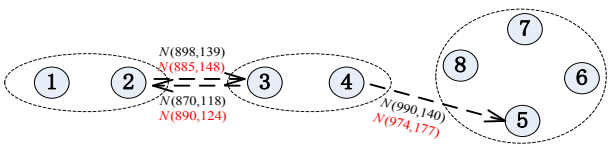

Fig. 7. The recovered topology graph. The ground truth (red) is estimated by the EM algorithm based on only true correspondences labeled by hand.

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 is 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 Figure 7. 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 the 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 [8] are shown in Figure 2.

5 Conclusions

In this paper, we have presented a solution for automatically recovering the three factors of the topology of a non-overlapping camera network. Unlike previous cross-correlation based work, the proposed 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. Combined with similarity cues, the proposed method can be applied to weighted cross-correlation models, which improves the performance. Future work will focus on extending the proposed method to large-scale multi-camera networks.

Acknowledgement. This work is funded by National Natural Science Foundation of China (Grant No.61175007,61175002), the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No.XDA06030300), the National Basic Research Program of China (Grant No.2012CB316302), the Tsinghua National Lab for Information Science and Technology Cross-discipline Foundation (Grant No.Y2U10 11MC1).

References

1. Ellis, T.J., Makris, D., Black, J.K.: Learning a Multi-camera Topology. In: Joint IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), pp. 165–171 (2003)
2. Cai, Y., Huang, K., Tan, T., Pietikäinen, M.: Recovering the Topology of Multiple Cameras by Finding Continuous Paths in a Trellis. In: ICPR, pp. 3541–3544 (2010)
3. Niu, C., Grimson, E.: Recovering Non-overlapping Network Topology using Far-field Vehicle Tracking Data. In: ICPR, pp. 944–949 (2006)
4. Marinakis, D., Giguère, P., Dudek, G.: Learning network topology from simple sensor data. In: Kobti, Z., Wu, D. (eds.) Canadian AI 2007. LNCS (LNAI), vol. 4509, pp. 417–428. Springer, Heidelberg (2007)

5. Javed, O., Rasheed, Z., Shafique, K., Shah, M.: Tracking across Multiple Cameras with Disjoint Views. In: ICCV, pp. 952–957 (2003)
6. Tieu, K., Dalley, G., Grimson, W.E.L.: Inference of Non-overlapping Camera Network Topology by Measuring Statistical Dependence. In: ICCV, pp. 1842–1849 (2005)
7. Nam, Y., Ryu, J., Choi, Y., Cho, W.: Learning Spatio-temporal Topology of a Multicamera Network by Tracking Multiple People. Proceedings of World Academy of Science, Engineering and Technology 24, 175–180 (2007)
8. Makris, D., Ellis, T., Black, J.: Bridging the Gaps between Cameras. In: CVPR, vol. 2, pp. 205–210 (2004)
9. Zou, X., Bhanu, B., Roy-Chowdhury, A.: Continuous Learning of a Multilayered Network Topology in a Video Camera Network. EURASIP Journal on Image and Video Processing (2009)