International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.11 (2015), pp.243-254
http://dx.doi.org/10.14257/ijsip.2015.8.11.22

Topology Learning of Non-overlapping Multi-camera Network

Xiaolin Li[1], Wenhui Dong[1, 2], Faliang Chang[2] and Peishu Qu[1]

[1]College of Physics and Electronic engineering, Dezhou University, Dezhou 253023, China

[2]School of Control Science and Engineering, Shandong University, Jinan 250061, China

[1]lixiaolina@163.com,dongwh_81@163.com,
[2]Flchang@sdu.edu.cn,qupsh@163.com

Abstract

We focus on the issue of learning the topology of the non-overlapping multi-camera network, which includes recovering the nodes (entry and exit zones), transition time distribution and links. Firstly, the nodes associated with each camera view are identified using clustering method. Then, transition time distribution is modeled as a Gaussian distribution and is computed by accumulated cross correlation and Gaussian fitting. Finally, the mutual information is used to refine the possible links and the topology is recovered. Experimental results on simulated data and real scene demonstrate the effectiveness of the proposed method.

Keywords: *Multi-camera network, Non-overlapping view, Topology*

1. Introduction

Nowadays, more and more camera networks are used to surveillance the scenes, which can overcome the disadvantage of the single camera with a limited view. However, it also brings many new problems [1-3]. Take target tracking in a non-overlapping multi-camera network as an example, the target disappears in one camera may reappear in any view of cameras. In order to decrease the ambiguity and obtain the connectivity of the target, topology information of the network plays an important role [4-6]. For a non-overlapping multi-camera network, the topology information usually has three factors: the nodes (entry and exit zones), transition time distribution and links. The nodes mean the exit zones and entry zones. They are the zones that the targets exit or enter the camera view. If there is a path between two nodes, then a link between them exists. So links indicate the connectivity of each two nodes. Finally, the transition time distribution is used to describe the probability of transition time of an object moving from one node to another.

In this paper, we focus on recovering the topology information of the non-overlapping multi-camera network. It is a challenging problem, because there are “blind areas” between two adjacent cameras and even the same target may have different observations under different cameras. In order to recover the topology accurately, we exploit the statistical spatio-temporal information in the surveillance videos. Firstly, the nodes associated with each camera view are identified using clustering method. Then, transition time distribution is modeled as a Gaussian distribution and is computed by accumulated cross correlation and Gaussian fitting. Finally, the mutual information is used to refine the possible links and the topology is recovered. We evaluate our method and compare it with other two methods by the simulated data and the real scene. The experimental results demonstrate that the proposed method is effective.

This paper is organized as follows. Section 2 describes the related works of topology learning. Section 3 presents the details of the proposed topology recovering method. Experimental results are shown in Section 4. Finally, we conclude the paper in Section 5.

ISSN: 2005-4254 IJSIP

2. Related Work

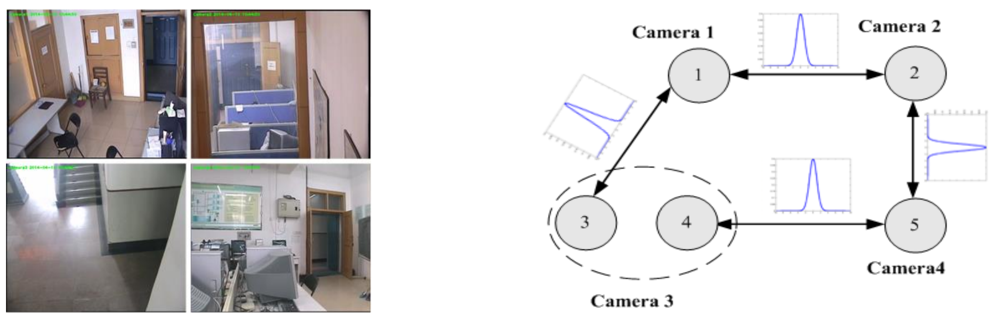
Many references focus on the problem of topology recovering of the camera networks. Some researchers learn the topology based on the camera calibration [7]. If camera calibration has been done in a 3D word coordinate system, the topology can be recovered straightly by geometric analysis and the view fields of cameras. However, it is a hard work to calibrate every camera in a large camera networks. Other researches try to recover the topology using the surveillance data without camera calibration. These methods can be divided into two categories: correspondence based methods [8-9] and correspondence-free methods [10]. Correspondence based methods need know whether the targets in different camera views actually correspond to the same target. This knowledge can be obtained manually or by some automatic identification methods. Javed *et. al.,* [9] present a supervised algorithm to track the target across the non-overlapping camera network. In order to recover the topology, Parzen windows are used during a training phase to find the correspondences. Because they use the manually labeled trajectories, this method is costly and not practical in real environment.

Correspondence-free methods relax the assumption of known data correspondence. In reference [10], the cross correlation function of the arrival time sequence at one node and the departure time sequence at another node is calculated in a time window. This method is based on the assumption that if there is a link between the two nodes, the peak of the cross correlation function will around the most popular transition time. However, due to the large number of false correspondence and large variance of transition time of different true correspondences, this method is not viable in most cases. Many methods are proposed to improve this method [11-12]. Niu and Grimson [11] propose a weighted cross correlation function on the vehicle tracking data to recover the topology. The primary idea of the method is to integrate the target appearance information to the cross correlation function. The appearance similarity is calculated as the product of the normalized color similarity and size similarity. This method can decrease the influence of false correspondences in certain degree. However, the appearances may vary under different condition. Recently, Chen *et. al.,* [13] propose a method based on N-neighbor accumulated cross correlations to learn the topology of the multi-camera networks. It focuses on finding the steadiest peak in the accumulated cross correlation function, which can overcome the disadvantage of the cross correlation function and can deal with large amounts of data or a long time window. Information-theoretic framework is also used by some researchers to infer the topology of the camera networks. Tieu *et. al.,* [14] measures the statistical dependency of the observations (transition time and color appearance of objects) in different cameras and use it to infer the topology. They measure the statistical dependence using non-parametric estimation and integrate the uncertainty of correspondence into a Bayesian manner. In reference [15], the Monte Carlo Expectation-Maximization algorithm is used to solve the data correspondence and network topology inference simultaneously. This approach works well when the number of the targets is limited, but it will be very slow when the number of the targets is large.

Our proposed method also belongs to the correspondence-free category and the three factors of the topology are learned. The entry and exit zones in each camera view are identified using clustering method. Then, transition time distribution is computed by accumulated cross correlation and Gaussian fitting. Finally, the detected links are refined by the mutual information. The most related work to our method is reference [13], but they only use thresholds to determine the variance of the transition time distribution and the link, which makes the method too relying on those thresholds.

3. Topology Learning

In this paper, we focus on recovering the topology of the non-overlapping camera network. Figure 1 shows an example of the non-overlapping network. Figure 1(a) is the views of a non-overlapping cameras network, which has four cameras. Figure 1(b) is the topology graph of the network. Nodes are entry or exit zones labeled by different numbers. In particular, there are two entry/exit zones in the view of camera 3. The arrows represent valid links between nodes across cameras. Transition time distributions of each valid link are also given as a Gaussian distribution. We will give the details of our topology recovering method in this section, including nodes (entry or exit zones) learning, transition time distribution learning and links refining.



(a) (b)

Figure 1. Topology Illustration: (a) Field of the Views of Four Cameras, (b) Topology of the Four Cameras

3.1. Nodes Learning

Nodes represent the entry zones and exit zones. So only the start points and end points of the trajectories are needed to learn the nodes [16]. After obtaining the entry point set and the exit point set, GMM model is used to clustering these points in this paper. The probability function of GMM model is defined:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathrm{N}(x \mid \mu_k, \Sigma_k) \tag{1}$$

Where K is the number of models, π_k is the weight, $\mathrm{N}(x \mid \mu_k, \Sigma_k)$ is the Gaussian distribution of each model and μ_k is the mean, Σ_k is the variance. The steps of GMM clustering are as following:

Step1. For each sample, estimate the production probability of each model.

$$\gamma(i,k) = \frac{\pi_k \mathrm{N}(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathrm{N}(x_i \mid \mu_j, \Sigma_j)} \tag{2}$$

Step 2. Estimate μ_j and Σ_j by maximum likelihood estimation.

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(i,k) x_i \tag{3}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(i,k)(x_i - \mu_k)(x_i - \mu_k)^T \tag{4}$$

Where $N_k = \sum_{i=1}^{N} \gamma(i,k)$ and $\pi_k = N_k / N$.

Step 3. Repeat the up two steps until the likelihood function convergences. The likelihood function is defined as:

$$\sum_{i=1}^{N} \log\{\sum_{j=1}^{K} \pi_j \mathrm{N}(x_i \mid \mu_j, \Sigma_j)\} \tag{5}$$

We record the entry points and exit points for 75 minutes out of a building using a camera installed on it and learn the nodes (entry and exit zone) using GMM clustering. The results are shown in Figure 2. The points in Figure (2a) are the entry and exit points. The clustering result is shown in Figure (2b). Finally, the exit and entry zones are described as ellipses. The centre of the ellipse is the mean of the Gaussian model. The major axis and minor axis are the eigenvalues of the covariance matrix.

(a) (b)

Figure 2. Entry and Exit Zones Learning: (a) Entry and Exit Points in the View, (b) Entry and Exit Zones

3.2. Transition Time Distribution Learning

In order to overcome the noises caused by the false correspondences and the large variances of the transition time of different true correspondences, transition time distribution is modeled as a Gaussian distribution. In this paper, we calculate the transition time distribution by accumulated cross correlation and Gaussian fitting.

Given $D_i(t)$ and $A_j(t)$ are the departure time sequence at node i and arrival time sequence at node j, respectively. The cross correlation function $C_{i,j}(T_0)$ can be calculated as:

$$C_{i,j}(T_0) = \sum_{t=-\infty}^{t=+\infty} D_i(t) A_j(t + T_0) \tag{6}$$

Based on the cross correlation, the accumulated cross correlation function can be calculated as:

$$R_{i,j}(T_n) = \sum_{T_0=T_n-n}^{T_0=T_n+n} C_{i,j}(T_0) \qquad T_n \geq n \tag{7}$$

For different n, the accumulated cross correlation function is calculated using (7). Then, we record the frequency of the time intervals corresponding to the peak values of $R_{i,j}(T_n)$. By doing so, we can find the most steady and frequent peak in accumulated cross correlation function rather than a very clear peak in the cross correlation. Figure 3(a) and Figure 3(b) demonstrate this issue. As can be seen, there are two clear peaks in the cross correlation function, which make it difficult to detect the transition time. While the accumulated cross correlation function can reduce the noises and reflect the true transition time.

However, transition time itself can not describe the relation of the two nodes accurately. In practice, the travel intervals will be different for different targets. So Gaussian distribution is more suitable to model it. In this paper, we use a Gaussian distribution to describe the relation, which is different from other methods that only use transition time. After obtaining the tran-sition time by finding the peak in the accumulated cross correlation function, Gaussian fitting is done using the transition time as the initial value to obtain the variance. Figure 3(c) shows the Gaussian fitting results of Figure 3(b).

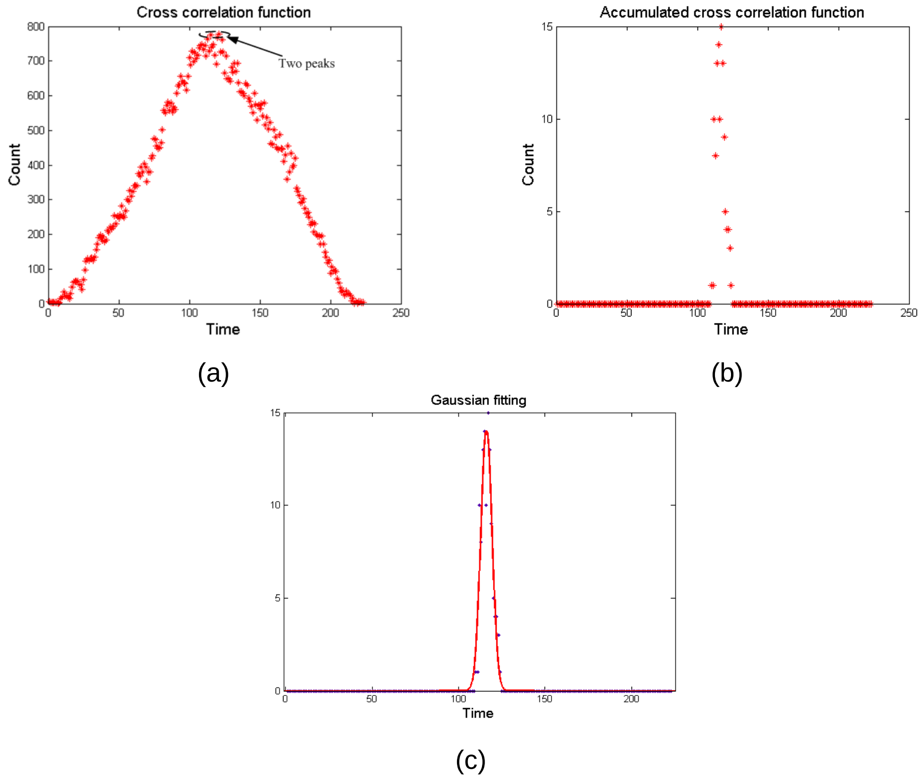


Figure 3. Transition Time Distribution Learning: (a) Cross Correlation Function, (b) Accumulated Cross Correlation Function, (c) Gaussian Fitting

3.3. Links Refining

By nodes learning and transition time distribution learning, many possible links can be found for disjoint views. However, there may be some false links in these detected links. So link refining is needed. We propose to use mutual information to refine these detected links.

Mutual information can measure the amount of information that one random variable contains about another variable. It also reflects the uncertainty reduction of one random variable due to the knowledge of the other.

$$\begin{aligned} I(X;Y) &= \int p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy \\ &= -\frac{1}{2}\log_2(1-\rho_{xy}^2) \end{aligned} \tag{8}$$

For a Markov chain type topology between three random variables $X \rightarrow Y \rightarrow Z$, we have $I(X,Y) \geq I(X,Z)$. So mutual information can help us to refine the links in the topology. If the mutual information of the two nodes in the link is above certain threshold, this link is true. Otherwise, it is a false link.

We can calculate the mutual information between the two nodes using Equation (8). ρ_{xy} is the correlation coefficient, which can be calculated as:

$$\rho_{i,j}^{2} = \frac{R_{i,j}(T_{peak}) - median(R_{i,j}(T))}{\sigma_{Di}\sigma_{Aj}} \tag{9}$$

where $R_{i,j}(T)$ is the accumulated cross correlation function. σ_{Di} and σ_{Aj} are the standard deviations of the departure time sequence and the arrival time sequence, respectively. If we use the whole sequence of $R_{i,j}(T)$ to calculate the correlation coefficient, the information of the false correspondences will also be absorbed. This will affect the accuracy of mutual information calculation. Because the transition time distribution is Gaussian distribution, we revise (9) by limiting the rang of T to $\left[T_{peak} - 3\sigma_T, T_{peak} + 3\sigma_T\right]$. Where σ_T is the standard deviation of the T. By doing so, we will get the most accurate mutual information.

4. Experimental Results

We evaluate the performance of the proposed method on simulated data and real scenes. We also compare it with the topology learning method in reference [10] and reference [13]. The details of the experiments are shown in the following subsections.

4.1. Experimental Results of Simulated Network

The simulation is based on a multi-camera network shown in Figure 4. In the network, 1 and 2 are both the exit and entry nodes and have 200 moving objects respectively; 3 is the exit node and has 400 moving objects; 4 is the entry node. The departure time sequences in node 1, 2 and 3 follow the uniform distribution $U(0,100)$, $U(100,200)$, and $U(200,300)$ respectively. The transition times between nodes 1 and 2, 2 and 4, 3 and 4, 4 and 1 follow the Gaussian distributions $N(20,6)$, $N(30,4)$, $N(5,6)$ and $N(10,4)$, respectively. Each object is equally to arrive at any connected node after leaving any node.

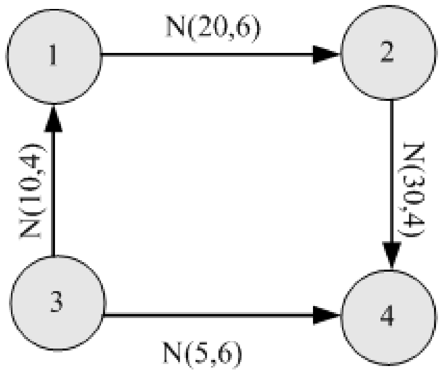


Figure 4. Simulated Multi-Camera Network

(1) Learning network topology using the proposed method

Figure 5 shows the frequency of the time interval corresponding to the peak values. We detect the transition time by finding the peaks. The Gaussian fitting is done using the transition time as the initial value to obtain the variance. Because there is no positive transition time between node 3 and 2, the link does not exists between them. Table 1 demonstrates the corresponding transition time and variance for each link. As can be observed, the false link between node 1 and 4 is also included.

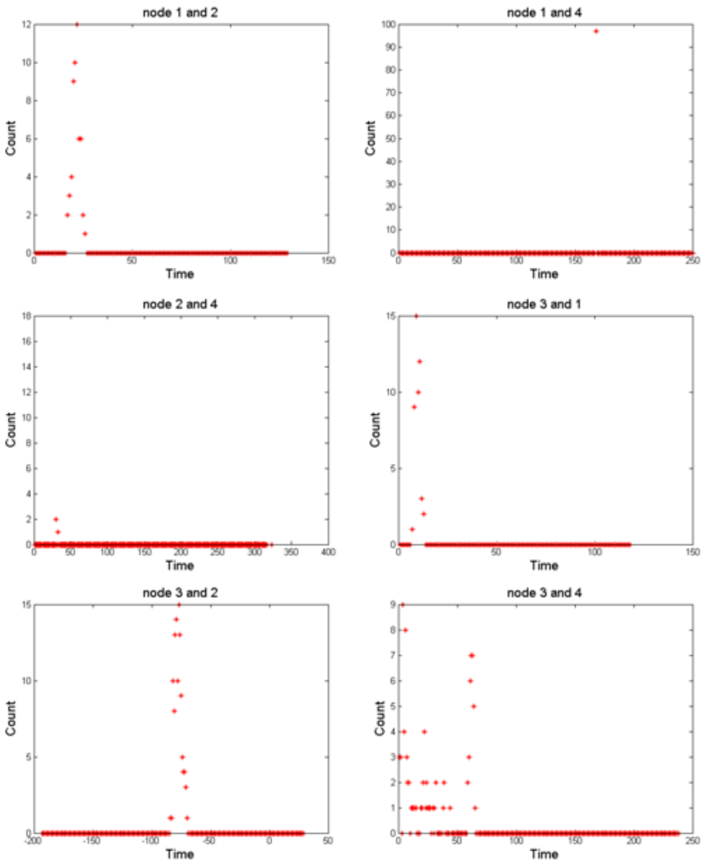


Figure 5. Transition Time Detection

Table 1. Transition Time Distribution of Each Link

	node1	node2	node3	node 4
node1	\	N(21,6.76)	\	N(233,5)
node2	\	\	\	N(29,6)
node3	N(9,4.41)	\	\	N(4.8,7.2)

Using formula (8) and (9), the possible links detected can be refined. Figure 6 demonstrates the mutual information using intensities corresponding to the magnitude of the mutual information in each link. As can been seen from it, the mutual information between node 1 and node 4 is the least and needed to be deleted. Finally, the simulated network can be fully recovered as Figure 7. As can be observed, the true links are all detected and the transition time for each link is very close to the true value (Compared to Figure 4).

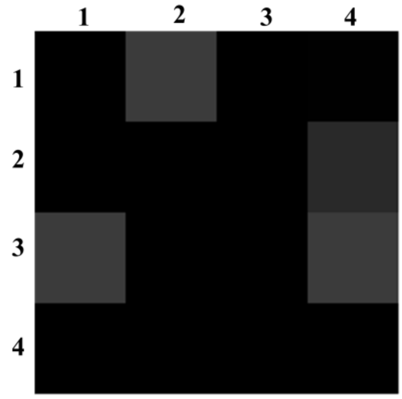


Figure 6. The Adjacency Matrix of the Mutual Information

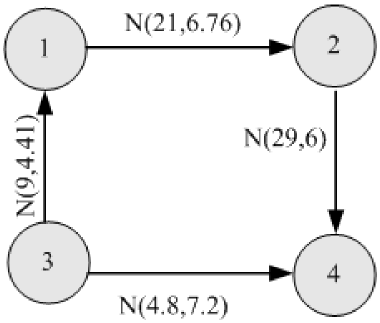


Figure 7. Topology Recovery Using the Proposed Method

(2) Compariation with Other Two Topology Learning Methods

We also use two other topology learning methods to recover the topology of the simulated network. One is the cross correlation method in reference [10], the other is the method in reference [13]. In cross correlation method, formula (6) is used to calculate cross correlation for every two nodes. Then, the transition time can be found using the threshold:

$$Thr = mean(C_{i,j}(T_0)) + \omega \cdot std(C_{i,j}(T_0)) \tag{10}$$

where ω is set to 1.5 in the experiment.

Figure 8 is the final topology learning result in reference [10]. Only transition times can be obtained using this method and there is a false link (the link between node 1 and node 4) in the topology recovery result. Reference [13] uses an iteration algorithm to estimate the co-nnectivity for each pair of nodes, and the parameters of the transition time distribution as we-ll. Thresholds are used to determine the variances of the transition time distributions and the links, which makes the method too relying on those thresholds. Figure 9 demonstrates the be-st result (different thresholds may have different results). It also cannot delete the false link between node 1 and 4.

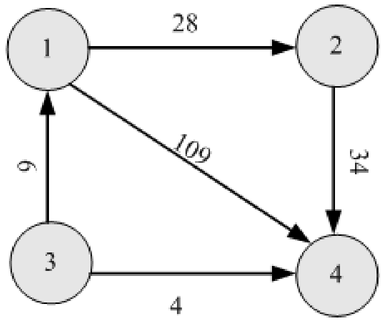


Figure 8. Topology Recovery using the Method in Reference [10]

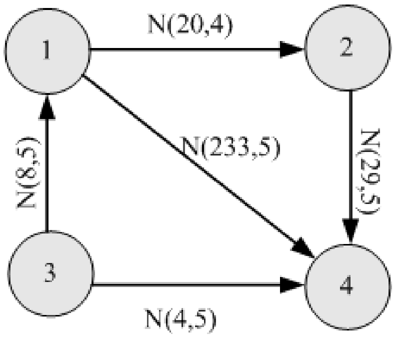


Figure 9. Topology Recovery using the Method in Reference [13]

From the compariation with other two topology learning methods (Figure 7, Figure 8, and Figure 9), we can conclude that the proposed method outperforms the other two methods. T-he method in reference [10] only can predict the transition time and can not make sure that whether the link exits. The method in reference [13] relies too much on the thresholds. Neither of them gives an accurate topology of the simulated network.

4.2. Experimental Results of Real Network

The real scene in Figure 1(a) is used to evaluate the proposed method. In this experiment, we first obtain the real factors of the topology using the training data. The real topology of t-he real network is shown in Figure 10. Then the proposed method and the methods in refere-nce [10] and [13] are tested to recover the topology. All the recovered topologies are compa-red with the real one.

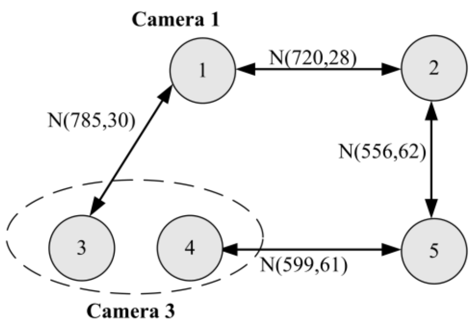


Figure 10. The Real Topology of the Camera Networks in Figure 1

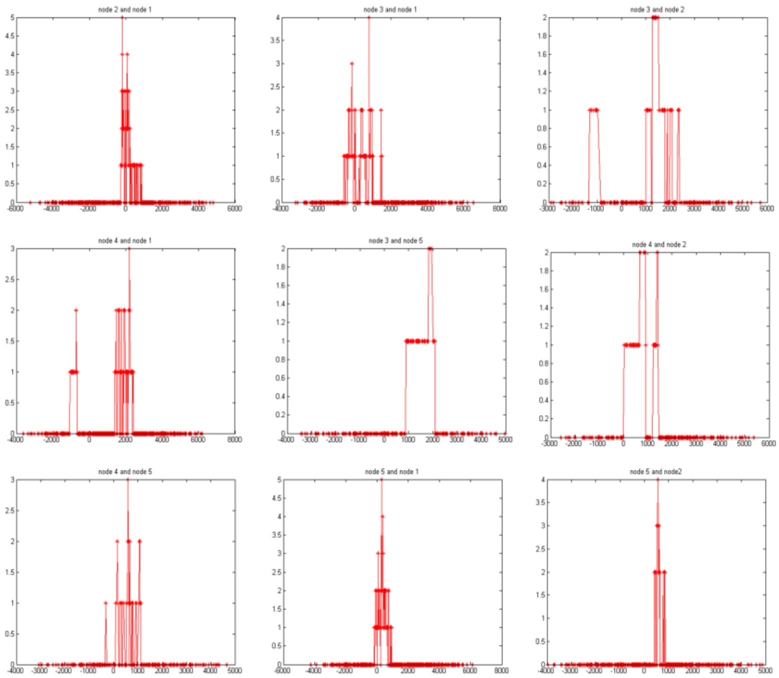


Figure 11. Transition Time Detection

(1) Learning Network Topology using the Proposed Method

The frequency of the time interval corresponding to the peak values are shown in Figure 11. All the possible links are learned and the transition times are detected by finding the peaks. T-he Gaussian fitting is done using the transition time as the initial value to obtain the variance. Table 2 shows the Gausssian distribution of each detected link. As can be observed, some fa-lse links are also detected (nodes 5 and 1, nodes 4 and 1, nodes 3 and 2, *etc*), so we need usi-ng the mutual information to refine the topology.

After transition time detection, all the possible links are refined using mutual information. Figure 12 shows the mutual information of each link couple. As can been observed, the mutual information of nodes 1 and 3, 2 and 3, 1 and 5, 3 and 5, 4 and 1 is so small that need to be deleted. Finally, the topology of this network can be fully recovered as Figure 13. The links are all detected and the transition time for each link is very close to the true value (compared to the ground truth in Figure 10).

Table 2. Transition Time Distribution of Each Detected Link

	node1	node2	node3	node4	node 5
node1	\	N(731,20)	N(788,25)	N(2206,10)	N(281,13)
node2	N(731,20)	\	N(1362,23)	N(867,45)	N(556,65)
node3	N(788,25)	N(1362,23)	\	\	N(1876,10)
node4	N(2206,10)	N(867,45)	\	\	N(599,63)
Node5	N(281,13)	N(556,65)	N(1876,10)	N(599,63)	\

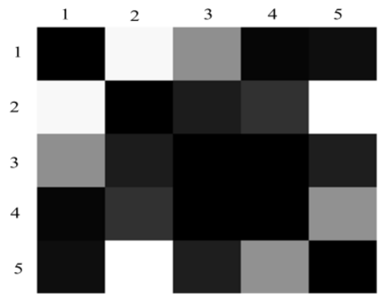


Figure 12. The Adjacency Matrix of the Mutual Information

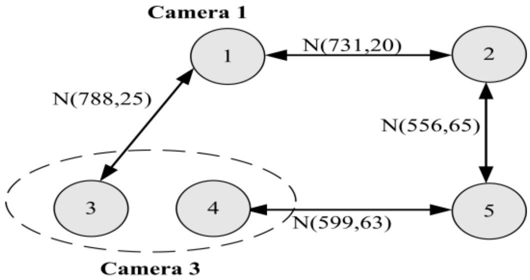


Figure 13. Topology Recovery Using the Proposed Method

(2) Compariation with other Topology Learning Methods

We also use the topology learning methods in references [10] and [13] to recover the topology of the real scene. The results are shown in Figure 14 and Figure 15. Compared to the proposed method (shown in Figure 13), the two approaches contain the false links and can not recover the topology correctly. So, the proposed method outperforms the two methods.

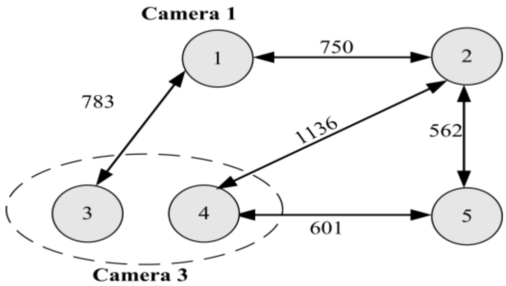


Figure 14. Topology Recovery Using the Method in Reference [10]

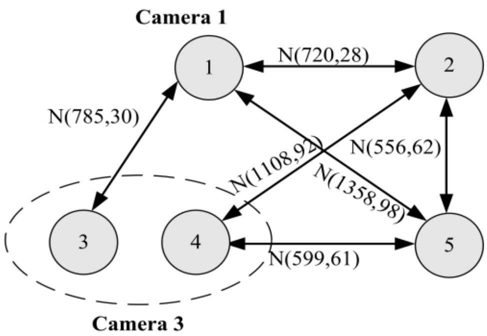


Figure 15. Topology Recovery Using the Method in Reference [13]

5. Conclusion

In this paper, a solution for automatically recovering the topology of a non-overlapping ca-mera network is proposed. By GMM clustering, we can identify the entry and exit zones that associated with each camera view. By using the accumulated cross correlation and Gaussian fitting, we can obtain the accurate time transition distribution of the links. By using mutual i-nformation, we can refine the links. Finally, the topology can be recovered accurately.

References

[1] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar, “Distributed localization of networked cameras”, 5th International Conference on Information Processing in Sensor Networks, Nashville, Tenn, USA, **(2006)** September, pp.34-42.
[2] A. Rahimi, B. Dunagan, and T. Darrell, “Simultaneous calibration and tracking with a network of non-overlapping sensors”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, vol. 1, **(2004)** June-July, pp.187-194.
[3] R. Fisher, “Self-organization of randomly placed sensors”, 7th European Conference on Computer Vision, Copenhagen, Denmark, vol. 2353, **(2002)** May, pp.146-160.
[4] D. Makris, T. Ellis, and J. Black, “Bridging the gaps between cameras,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, vol. 2, **(2004)** June-July, pp. 205-210.
[5] C. Niu and E. Grimson, “Recovering non-overlapping network topology using far-field vehicle tracking data”, International Conference on Pattern Recognition, Hong Kong, vol. 4, **(2006)** August, pp. 944-949.
[6] X. Zou, B. Bhanu, B. Song, and A. K. Roy-Chowdhury, “Determining topology in a distributed camera network”, International Conference on Image Processing, San Antonio, Tex, USA, vol. 5, **(2007)** September, pp. 133-136.
[7] G. A. Jones, J. R. Renno and P. Remagnino, “Auto-calibration in multiple-camera surveillance environments,” Process of Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, **(2002)**, pp.583-588.
[8] V. Kettnaker, R. Zabih, “Bayesian multi-camera surveillance”, IEEE International Conference on Computer Vision and Pattern Recognition, **(1999)**, pp. 253-259.
[9] O. Javed, Z. Rasheed, K. Shafique and M. Shah, “Tracking across multiple cameras with disjoint views”, IEEE International Conference on Computer Vision, **(2003)**, pp. 952-957.
[10] T. J. Ellis, D. Makris and J. Black, “Learning a multi-camera topology”, Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, **(2003).**
[11] Niu, C. and Grimson, E., “Recovering Non-overlapping Network Topology using Far-field Vehicle Tracking Data”, International Conference on Pattern Recognition, **(2006)**, pp. 944–949.
[12] Zou, X., Bhanu, B. and Roy-Chowdhury, A., “Continuous learning of a multilayered network topology in a video camera network”, EURASIP Journal on Image and Video Processing, **(2009)**, pp. 1-19.
[13] Xiaotang Chen, Kaiqi Huang, and Tieniu Tan, “Object tracking across on non-overlapping views by learning inter-camera transfer models”, Pattern Recogniton, vol. 47, **(2014)**, pp. 1126-1147.
[14] K. Tieu, G. Dalley and E. Grimson, “Inference of non-overlapping camera network topology by measuring statistical dependence”, Process of IEEE International Conference on Computer Vision, **(2005)**, pp.1842-1849.
[15] D. Marinakis, G. Dudek, and D. J. Fleet, “Learning sensor network topology through Monte Carlo expectation maximization”, IEEE International Conference on Robotics and Automation, Barcelona, Spain, **(2005)** April, pp. 4581-4587.
[16] D. Makris and T. Ellis, “Automatic learning of an activity-based semantic scene model”, 7th IEEE International Conference on Advanced Video and Signal-Based Surveillance, **(2003)**.

