Adaptive topology learning of camera network across non-overlapping views

Yang Biao[1]　　Lin Guoyu[2]　　Zhang Weigong[2]

([1] Faculty of Information Science and Engineering, Changzhou University, Changzhou 213164, China)
([2] School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: An adaptive topology learning approach is proposed to learn the topology of a practical camera network in an unsupervised way. The nodes are modeled by the Gaussian mixture model. The connectivity between nodes is judged by their cross-correlation function, which is also used to calculate their transition time distribution. The mutual information of the connected node pair is employed for transition probability calculation. A false link eliminating approach is proposed, along with a topology updating strategy to improve the learned topology. A real monitoring system with five disjoint cameras is built for experiments. Comparative results with traditional methods show that the proposed method is more accurate in topology learning and is more robust to environmental changes.
Key words: non-overlapping views; mutual information; Gaussian mixture model; adaptive topology learning; cross-correlation function
doi: 10. 3969/j. issn. 1003 − 7985. 2015. 01. 011

W̲ide range monitoring with a camera network is a useful method for addressing the safety issue. However, covering all areas with cameras is not realistic and necessary. A multi-camera monitoring network is always built by covering key areas with cameras and leaving general areas un-covered. Thus, the camera network topology, including connectivity and transition time, is significant for continuous tracking across a camera network with non-overlapping views.

There are mainly two approaches for topology learning, including a supervised approach and an un-supervised one. The former learns the topology by tracking labeled objects during a training stage, such as the Parzen window topology learning approach proposed by Javed et al[1–2]. However, the need for mass labeled data and the sensitivity to environmental changes make this approach unsuitable for realistic applications. The latter learns the topology adaptively with unlabeled data. The cross-corre-

lation function (CCF) [3] is employed to calculate the connectivity between the nodes of different camera views. To improve its accuracy in connectivity inferring, color information is introduced into the traditional CCF by Niu et al[4]. Face detection[5] is employed to improve the color-based CCF used on humans. The latter can precisely infer the connectivity if the identities of tracked people are recognized. However, face detection is difficult to popularize due to its requirements for imaging angle and precision.

A camera network topology learning approach is proposed in an un-supervised way. It is represented by a graph model $G = \langle V, E, W \rangle$ [6]. The set of nodes is represented by $V = \{v_i\}$, where v_i indicates the i-th entering/leaving location. The set of edges is represented by $E = \{e_{ij}\}$, where e_{ij} indicates the connectivity between nodes v_i and v_j. The set of weights is represented by $W = \{w_{ij}\}$, where w_{ij} indicates the transition probability of e_{ij}. A false link eliminating approach is proposed to improve the learned topology and an update strategy is employed to ensure the robustness of the learned topology to environmental changes.

1　Joint Appearance Similarity

1.1　Color similarity

Color feature is commonly used for matching. Major color[7] is employed due to its robustness to affine transformation. A patch-based strategy is employed to calculate the color similarity P_c, which is defined as

$$P_c = \alpha P_{head} + \beta P_{body} + \gamma P_{legs} \tag{1}$$

where P_{head}, P_{body} and P_{legs} are color similarities of the head, the body and the legs, respectively; α, β and γ are the corresponding weights (α, β and γ are set to be 0. 2, 0. 4 and 0. 4, respectively.).

1.2　Textural similarity

Texture feature is always used as detailed information. The histogram of gradient (HOG) [8] is employed as texture feature due to its superiority in human representation. Supposed that $O_{a,i}$ and $O_{b,j}$ are two observations detected in v_i and v_j, the textural similarity P_t is defined as u

$$P_t = \exp\left(-\frac{\sqrt{1 - \sum_{u=1}^{k_{HOG}} \sqrt{H_{a,i}(u) H_{b,j}(u)}}}{\sigma_t}\right) \tag{2}$$

where $H_{a,i}$ and $H_{b,j}$ are the HOG descriptors of $O_{a,i}$ and $O_{b,j}$, respectively; σ_t is a pre-defined bandwidth and k_{HOG} is the dimension of HOG descriptor.

1.3 Saliency similarity

The saliency feature proposed by Zhao et al.[9] can be used to acquire identity information instead of face detection. Adjacency constrained patch matching is employed to build dense correspondence between image pairs, which shows effectiveness in handling misalignment caused by large viewpoint and pose variations. Human saliency is then extracted automatically by the k-nearest-neighbor algorithm. Finally, the saliency similarity P_s of two observations is calculated by a bi-directional weighting approach as proposed by Zhao et al[9].

1.4 Joint appearance similarity

The joint appearance similarity is calculated based on color similarity P_c, textural similarity P_t and saliency similarity P_s. If $O_a = O_b$, which indicates that observations $O_{a,i}$ and $O_{b,j}$ belong to the same object, the joint appearance similarity $P_a(O_{a,i}, O_{b,j})$ can be defined as

$$P_a(O_{a,i}, O_{b,j}) = P(O_{a,i}, O_{b,j} \mid O_a = O_b) = P_c P_t P_s \quad (3)$$

2 Learning Camera Network Topology

2.1 Learning topology nodes

The set of nodes $V = \{v_i\}$ is learned based on the results of single camera tracking. In each view, the entry/exit zones are modeled as a Gaussian mixture model (GMM) with parameters $Z = \{z, \lambda_z, \mu_z, \sigma_z^2\}$. For each GMM, z indicates the number of Gaussian distributions and λ_z represents the weight of the z-th Gaussian distribution. μ_z and σ_z^2 represent the mean and variance of the z-th Gaussian distribution. z is automatically determined according to the Bayesian information criterion (BIC). Other parameters $\{\lambda_z, \mu_z, \sigma_z^2\}$ can be estimated by the expectation maximization (EM) algorithm.

2.2 Learning topology edges

The set of edges $E = \{e_{ij}\}$ is learned based on the CCF, which is used for connectivity inferring. For two nodes from different camera views, a prominent single peak of their CCF curve indicates that they are connected. The traditional CCF of v_i and v_j can be defined as

$$R_{ij}(T) = E\{X_i(t)Y_j(t+T)\} = \sum_{t=-\infty}^{t=\infty} \|X_i(t)\| \|Y_j(t+T)\| \quad (4)$$

where the departure events of v_i and arrival events of v_j at time t are represented as $X_i(t)$ and $Y_j(t)$, respectively. To improve the traditional CCF, joint appearance similarity is used to build a new CCF, which is defined as

$$R_{ij}(T) = \sum_{t=-\infty}^{t=\infty} \sum_{O_{a,i} \in X_i(t)} \sum_{O_{b,j} \in Y_j(t+T)} P_a(O_{a,i}, O_{b,j})$$
$$\text{s.t. } P_a(O_{a,i}, O_{b,j}) > \delta \quad (5)$$

where $P_a(O_{a,i}, O_{b,j})$ is added into the CCF if and only if it is greater than a given threshold δ, which is employed to guarantee the high correlation in CCF.

Two nodes are connected if and only if the peak value of $R_{ij}(t)$ is greater than a threshold T_c, which is defined as

$$T_c = \text{mean}(R_{ij}(T)) + \omega \text{std}(R_{ij}(T)) \quad (6)$$

where ω is an user-defined parameter.

The transition time distribution (TTD) between two connected nodes is acquired by normalizing their CCF. A TTD can be built as a Gaussian model whose parameters are learned by the EM algorithm.

$$P_{i,j}(T) = R_{i,j}(T) / \|R_{i,j}(T)\| \sim N(\mu, \sigma^2) \quad (7)$$

2.3 Learning topology weights

The set of weights $W = \{w_{ij}\}$ is learned based on the mutual information (MI)[10] between two connected nodes v_i and v_j. Letting $X_i(t)$ and $Y_j(t+T)$ be two temporal sequences from v_i and v_j, their mutual information $I(X, Y)$ is defined as

$$I(X, Y) = \int p(X, Y) \log \frac{p(X, Y)}{p(X)p(Y)} dXdY = -\frac{1}{2} \log_2(1 - \rho_{X,Y}^2) \quad (8)$$

where $\rho_{X,Y}^2$ represents the correlation coefficient, which can be estimated as

$$\rho_{X,Y}^2 = \frac{R_{i,j}(T_{peak}) - \text{mean}(R_{i,j}(T))}{\sigma(X_i(t))\sigma(Y_j(t+T))} \quad (9)$$

where T_{peak} indicates a clear peak in $R_{i,j}(T)$ at time $T = T_{peak}$.

2.4 False link elimination

The learned topology can be represented as a link matrix or topological graph, as shown in Tabs. 1 and 2. For a link matrix, the value of row m column n denotes the transition probability from v_m to v_n. For a more intuitional topological graph, connectivity between different nodes
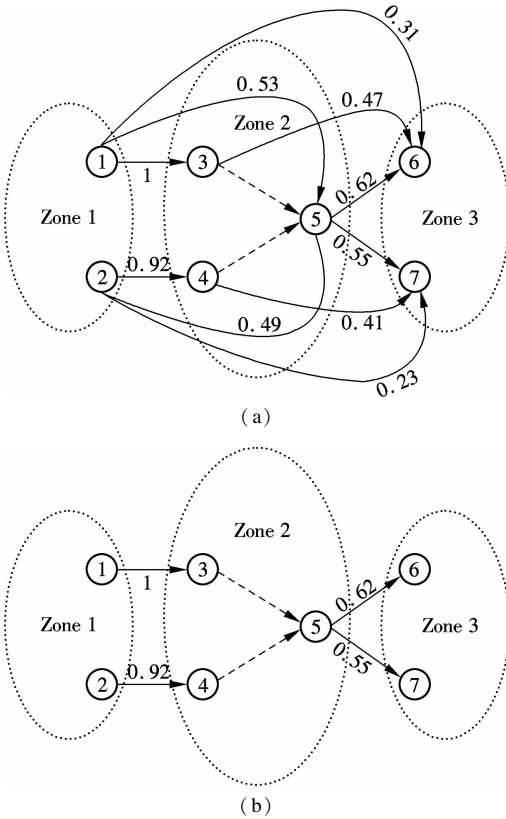
Tab. 1 Learned topology with "false link"

Zone	Node	Node						
		1	2	3	4	5	6	7
Zone 1	1	0	0	1	0	0.53	0.31	0
	2	0	0	0	0.92	0.49	0	0.23
Zone 2	3	0	0	0	0	0	0.47	0
	4	0	0	0	0	0	0	0.41
	5	0	0	0	0	0	0.62	0.55
Zone 3	6	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0

Tab. 2 Learned topology without "false links"

Zone	Node	Node						
		1	2	3	4	5	6	7
Zone 1	1	0	0	1	0	0	0	0
	2	0	0	0	0.92	0	0	0.23
Zone 2	3	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
	5	0	0	0	0	0	0.62	0.55
Zone 3	6	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0

is indicated by a solid arrow with its transition probability. Connectivity between nodes from the same camera view is indicated by a dotted arrow which is obtained from the results of single camera tracking.

However, false links always exist in the learned topology. As shown in Fig. 1(a), v_1 and v_5 are not connected without v_3. False links should be removed in order to reduce ambiguity in object matching across disjoint camera views[11]. For instance, if an object disappears in v_1, it will only reappear in v_3 rather than v_5 or v_6.

Fig. 1 Topological graphs of the learned topology. (a) Topological graph with a "false link"; (b) Topological graph without a "false link"

Mutual information and TTD are utilized to eliminate false links. Mutual information of definite false links is less than a low threshold T_{F1} (0.15 in this paper). Mutual information of probable false links is between a low threshold T_{F1} and a high threshold T_{F2} (0.8 in this paper). Meanwhile, the probable false links must contain a set of true sub-links whose mutual information is larger than

T_{F2}. The probable false link is regarded as a definite false link when its TTD is similar to the TTDs of all its sub-links. Kullback-Leibler divergence is employed to calculate the similarity between two distributions. Fig. 1(b) demonstrates the improved topology after false link elimination.

2.5 Update strategy

The topology of practical camera network, including TTD and transition probability, should be updated to fit the environmental changes. Let t_k be the passing time of the k-th object passing two nodes whose original TTD is $N(\mu, \sigma^2)$ and ρ be the updating rate. Then, the new parameters of TTD are defined as

$$\mu^* = (1 - \rho)\mu + \rho t_k \qquad (10)$$

$$\sigma^{*2} = (1 - \rho)\sigma^2 + \rho(\mu^* - t_k)^2 \qquad (11)$$

The transition probability between two nodes is updated based on all objects passing them. Letting the updating rate be κ, the new transition probability is defined as
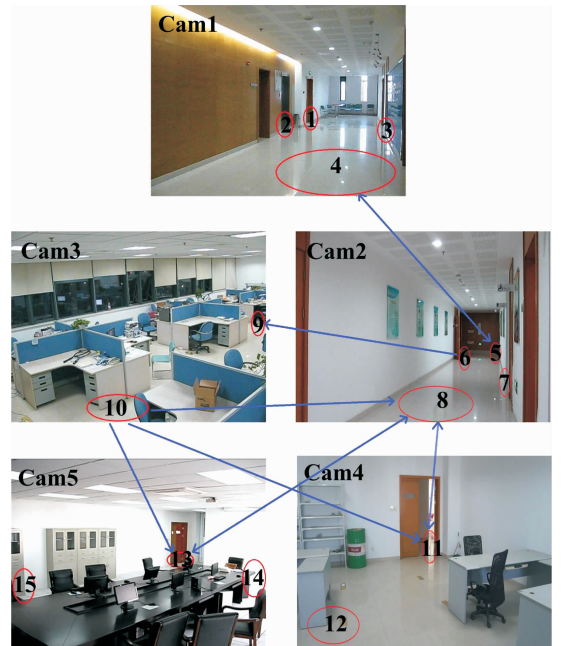
$$w_{ij}^* = (1 - \kappa)w_{ij} + \kappa P_{ij}(T_w) \qquad (12)$$

where $P_{ij}(T_w)$ is the ratio between the object number from v_i to v_j and the object number exiting from v_i at time segment T_w.

3 Experimental Analysis

3.1 Experimental environment

Fig. 2 shows a camera network which contains five cameras with non-overlapping views. Each camera is hung on the ceiling of different places and looks down at no specific angle. Lighting conditions in different places

Fig. 2 The camera network used in the experiments

are similar to ensure the robustness of appearance similarity. Data is continuously acquired from 8:00 to 12:00. CamShift[12] is employed as the single camera tracking algorithm. Entry/exit nodes are denoted in numbers and true connectivity is also given by arrows.
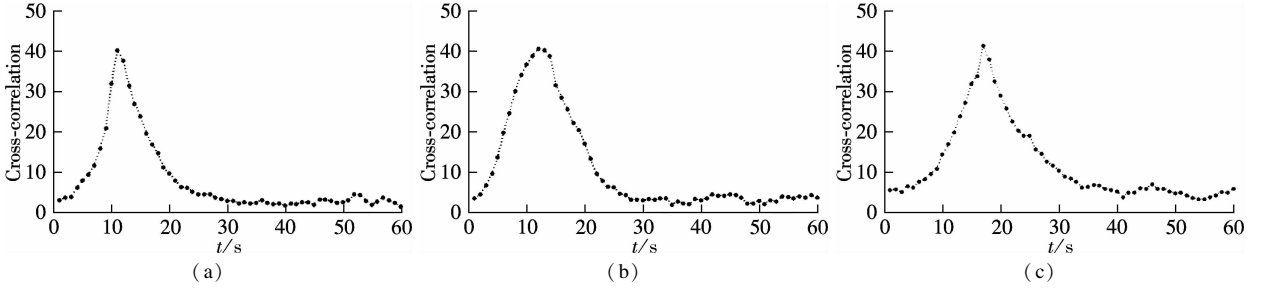
3.2   Performances of different CCFs

Figs. 3 to 8 give the comparative results of different CCFs which are utilized to learn topology edges. Six pairs of nodes are used, including three connected pairs ($V_4 \rightarrow V_5$, $V_6 \rightarrow V_9$, $V_{10} \rightarrow V_{11}$) and three disconnected ones ($V_1 \rightarrow V_{14}$, $V_2 \rightarrow V_{12}$, $V_3 \rightarrow V_{15}$). Fig. 3 shows the power of our CCF for connectivity inferring due to the prominent single peak. As shown in Fig. 4, double peaks exist in the curve of Fig. 4(c) due to the ambiguity of the color-
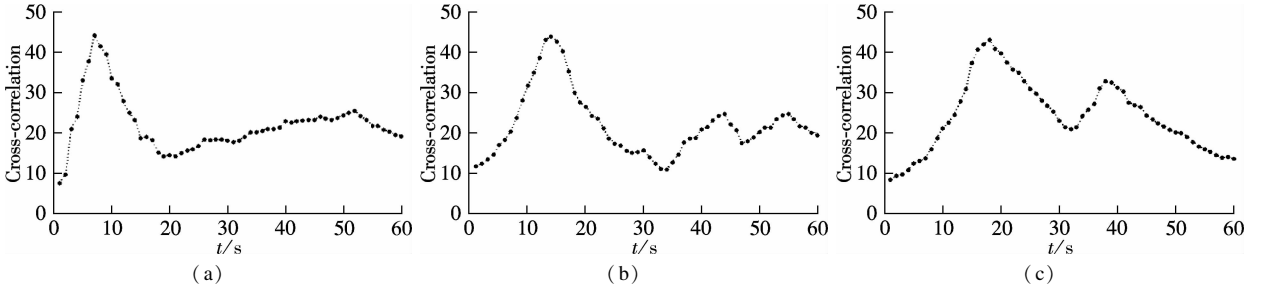
based CCF. Peaks are obscure in all curves of Fig. 5 because no appearance information is added into the traditional CCF. Figs. 6 to 8 show the curves of disconnected node pairs. Compared to the traditional CCF and the color-based CCF, our CCF can restrain curve peaks of disconnected nodes and then reduce the probability of learning a wrong connectivity. Thus, topology edges learned by our CCFs are more accurate than those by the other two commonly used CCFs.
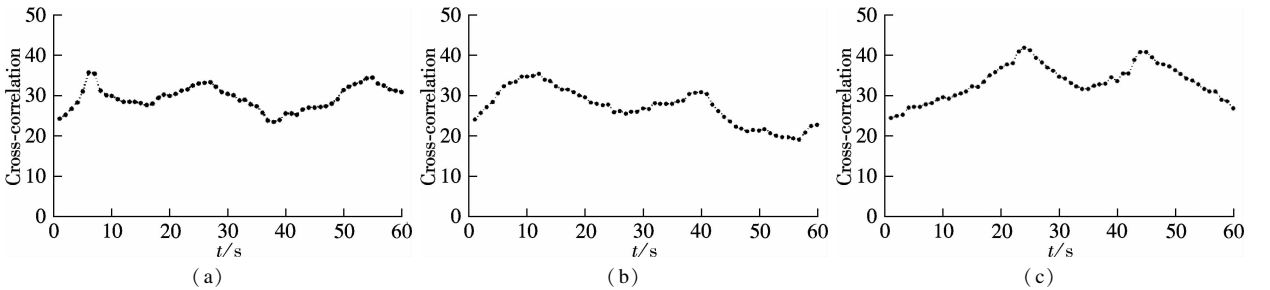
3.3   Experiments of topology learning

The topology of camera network as shown in Fig. 2 is learned using the proposed algorithm. Fig. 9 shows the learned topology without false links elimination and Fig. 10 shows the learned topology after false link elimination.
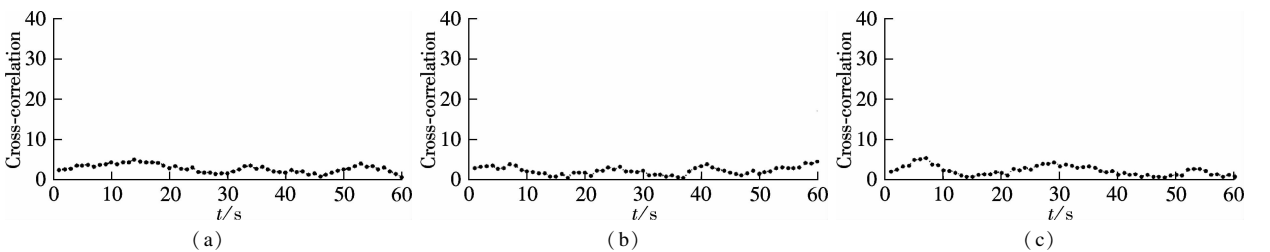

Fig. 3   Curves of our CCF between connected nodes. (a) $V_4 \rightarrow V_5$; (b) $V_6 \rightarrow V_9$; (c) $V_{10} \rightarrow V_{11}$
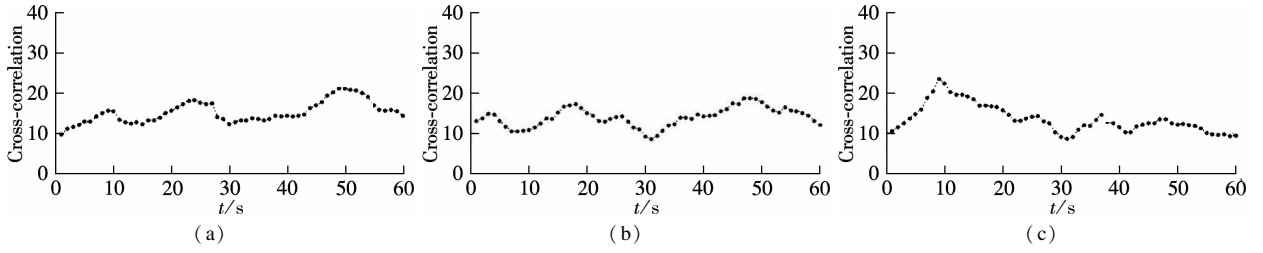

Fig. 4   Curves of the color-based CCF between connected nodes. (a) $V_4 \rightarrow V_5$; (b) $V_6 \rightarrow V_9$; (c) $V_{10} \rightarrow V_{11}$
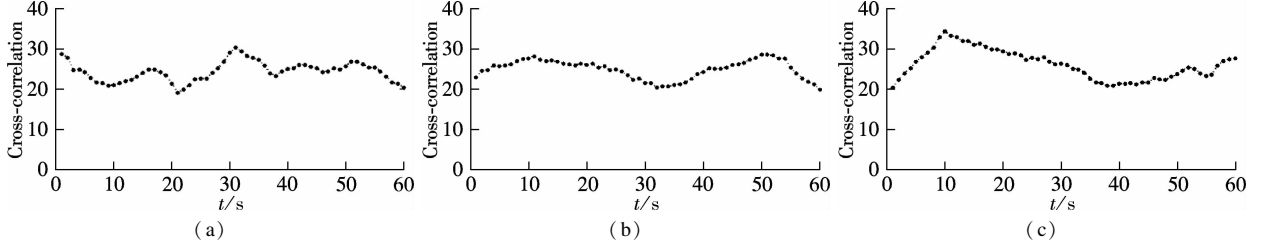

Fig. 5   Curves of the traditional CCF between connected nodes. (a) $V_4 \rightarrow V_5$; (b) $V_6 \rightarrow V_9$; (c) $V_{10} \rightarrow V_{11}$
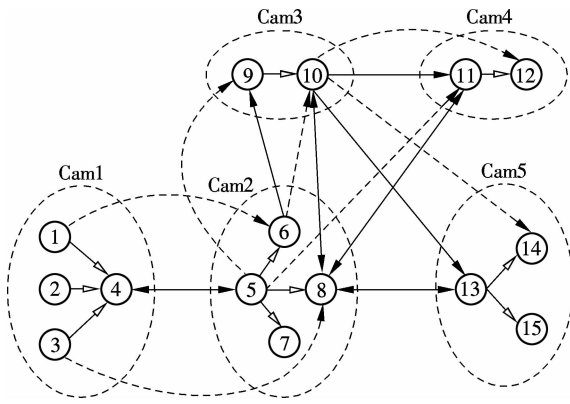

Fig. 6   Curves of our CCF between disconnected nodes. (a) $V_1 \rightarrow V_{14}$; (b) $V_2 \rightarrow V_{12}$; (c) $V_3 \rightarrow V_{15}$
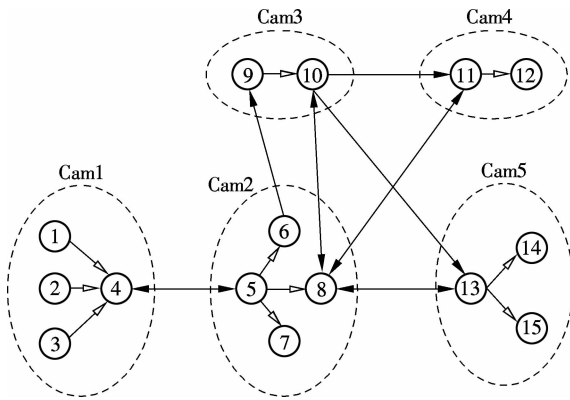
Fig. 7 Curves of the color-based CCF between disconnected nodes. (a) $V_1 \rightarrow V_{14}$; (b) $V_2 \rightarrow V_{12}$; (c) $V_3 \rightarrow V_{15}$

Fig. 8 Curves of the traditional CCF between disconnected nodes. (a) $V_1 \rightarrow V_{14}$; (b) $V_2 \rightarrow V_{12}$; (c) $V_3 \rightarrow V_{15}$

Fig. 9 Learned topology without false link elimination

Fig. 10 Learned topology with false link elimination

As a result, the topology learned by our approach can offer more useful spatio-temporal information than the commonly learned topology.

4 Conclusion

An un-supervised camera network topology learning approach is proposed in this paper. Joint appearance similarity is employed to improve the traditional CCF in edges learning. A false link elimination strategy is proposed to optimize the learned topology. An update strategy is also used due to environmental changes. Comparative experiments in a practical camera network with disjoint views indicate that our approach can learn a more precise topology which is also robust to environmental changes.

References

[1] Javed O, Rasheed Z, Shafique K, et al. Tracking across multiple cameras with disjoint views [C]//*Proceedings of the Ninth IEEE International Conference on Computer Vision*. Nice, France, 2003: 952 – 957.

[2] Javed O, Shafique K, Shah M. Appearance modeling for tracking in multiple non-overlapping cameras [C]//*Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Diego, CA, USA, 2005: 26 – 33.

[3] Makris D, Ellis T, Black J. Bridging the gaps between cameras [C]//*Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA, 2004: 205 – 210.

[4] Niu C, Grimson E. Recovering non-overlapping network topology using far-field vehicle tracking data [C]//*Proceedings of International Conference on Pattern Recognition*. Hong Kong, China, 2006: 944 – 949.

[5] Center for Research in Intelligent Systems. Continuous learning of a multilayered network topology in a video camera network[R]. Berkeley, CA, USA: University of California at Berkeley, 2009.

[6] Nam Y Y, Ryu J H, Choi Y J, et al. Learning spatio-temporal topology of a multi-camera network by tracking multiple people[J]. *International Journal of Computer, Information, Systems and Control Engineering*, 2007, **6**(1): 1523 – 1528.

[7] Piccardi M, Cheng E D. Tracking matching over disjoint camera views based on an incremental major color spectrum histogram [C]//*IEEE Conference on Advanced Video and Signal Based Surveillance*. Como, Italy, 2005: 147 – 152.

[8] Alahi A, Vandergheynst P, Bierlaire M, et al. Cascade

of descriptors to detect and track objects across any net-work of cameras[J]. *Computer Vision and Image Under-standing*, 2010, **114**(6): 624 – 640.

[9] Zhao R, Ouyang W L, Wang X G. Unsupervised sali-ence learning for person re-identification [C]//*Proceed-ings of* 2013 *IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA, 2013: 3586 – 3593.

[10] Viola P, Wells W M III. Alignment by maximization of mutual information[J]. *International Journal of Comput-er Vision*, 1997, **24**(2): 137 – 154.

[11] Liu S H, Lai S M, Zhang M J. A min-cost flow based algorithm for objects association of multiple non-overlap-ping cameras[J]. *ACAT Automatica Sinica*, 2010, **36**(10): 1484 – 1489.

[12] Exner D, Bruns E, Kurz D. Fast and robust CAMShift tracking [C]//2010 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Fran-cisco, CA, USA, 2010: 9 – 16.

无重叠视域摄像机网络拓扑自适应学习方法

杨　彪[1]　　林国余[2]　　张为公[2]

([1] 常州大学信息科学与工程学院, 常州 213164)
([2] 东南大学仪器科学与工程学院, 南京 210096)

摘要:提出一种自适应拓扑学习方法来无监督地学习摄像机网络的拓扑. 利用混合高斯算法建立节点模型, 并通过计算节点对的互关联函数得到该对节点的连通性以及连通节点对的转移时间分布. 利用交互信息计算连通的节点对的转移概率. 对学习到的拓扑结构, 提出虚假连接排除策略以及拓扑更新策略对其进行优化. 为了测试所提出算法的有效性, 搭建了由 5 个不包含重叠视域的摄像机组成的监控系统进行试验. 通过与已有算法的对比, 结果表明该算法可以更准确地学习监控网络的拓扑, 并对环境变化有一定的鲁棒性.

关键词:无重叠视域;交互信息;混合高斯模型;自适应拓扑学习;互关联函数

中图分类号:TP391