A Tempo-Topographical Model Inference of a Camera Network for Video Surveillance

Khalid Al-Shalfan and M. Elarbi-Boudihir

***Abstract*—A method is proposed to automatically construct a tempo-topographical model of a wireless IP-multi-camera network using some observations of known moving targets. The method is entirely unsupervised with no calibration of the cameras. This enables to extend the activity analysis across the entire camera network, constructing transition model between non-overlapping camera views, and also enabling the tracking across the blind-regions of the network. Based only on camera spatial and temporal information, this approach eliminates redundancy in the paths linking any pair of cameras by using the weighted graph technique to each node. The experimental results on real-time video streams show the feasibility of our system and its effectiveness in human activity tracking.**

***Index Terms*—Abnormal behavior, real-time detection, multi-camera tracking, video surveillance.**

I. Introduction

Modern video surveillance systems gained attention in the wider community of computer vision more than a decade ago. Today, the issue receives more intense pursuit from the narrower but more focused visual surveillance community [1]. Automated video surveillance systems constitute a network of cameras observing people as well as other moving and interacting objects in a given environment for patterns of normal/abnormal activities, interesting events, and other domain-specific goals [2]. Given multiple camera views, activity can be further categorized into global activity and regional activity. A regional activity refers to an activity that takes place locally in a single region of a camera view [3]. A global activity, on the other hand, is an activity that involves correlated partial observations of multiple regional activities across multiple cameras. To establish reliable reasoning on observed activity in a distributed camera network, it is critical and necessary to learn a global visual context that defines the spatial, temporal, and dependency relationships between partial observations across camera views. Activity models that are based on trajectory observations are proposed in [4], [5]. Although these activity models are appropriate for either single camera views or for overlapped camera views, where targets can be continuously tracked, they are inappropriate for multi-camera systems with occluded regions.

Manuscript received December 9, 2012; revised March 7, 2013. This work was supported in part by the King Abdula-Aziz City for Science and Technology, under Grant AT-29-314. The authors would like to thank the King Abdul-Aziz City for Science and Technology (KACST) for financing and supporting this work.

K. Al-Shalfan is with the Computer Science Department, Imam Med Bin Saud University, Riyadh. Kingdom of Saudi Arabia (email: Kshalfan@gmail.com).

M. Elarbi-Boudihir is with the Applied Mathematics Department, Imam Med Bin Saud University, Riyadh. Kingdom of Saudi (e-mail: elarbi-boudihir@lycos.com).

II. Multi-Camera Configuration

A multiple-camera system is capable of covering a wider area of a complex scene. Importantly, it has the potential of providing a complete record of an object's activity in a complex scene, allowing a global interpretation of the object's underlying behavior. Objects moving across camera views often experience drastic variations in their visual appearances owing to different illumination conditions, camera orientations and changes in object pose. All these factors increase the uncertainties in activity understanding [6]. Apart from inevitable visual variations across views, unknown and arbitrary inter-camera gap between disjoint cameras is another factor that leads to uncertainty in activity understanding. In particular, the unknown and often large separation of cameras in space causes temporal discontinuity in visual observations [7]-[10] (See Fig.1).

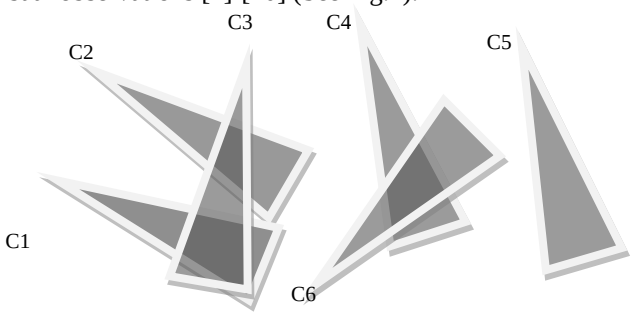


Fig. 1. The most common case involving overlapping

Furthermore, two widely separated camera views may include arbitrary number of entry/exit locations in the gap, where existing objects can disappear and new objects can appear, causing uncertainty in understanding and correlating activities in both camera views [11].

Our approach is to model time and space linking multi-camera activities by tackling three problems in multi-camera activity understanding:

1) Estimating the spatial topology and importantly the temporal topology of a camera network.
2) Facilitating more robust and accurate person re-identification between different camera views, by resolving ambiguities and uncertainties that arise due to large and unknown separation between cameras both spatially and temporally.
3) Performing activity-based temporal segmentation by linking visual evidence collected from different camera views.

DOI: 10.7763/IJCEE.2013.V5.746

III. System Architecture

Video surveillance is increasingly found in academic institutions [12]. It is used to oversee the safety of faculty members, staff and students, as well as to protect assets from vandalism and theft. Moreover, the campuses may be extensive, especially in the case of universities, and be comprised of several buildings, accesses and parking lots to monitor. Since educational institutions often have an IP network infrastructure, it is beneficial to set up digital video surveillance systems [13]. Due to the above reasons, we have implemented our IVSS in our University for testing. Basically, the system is composed of a set of wireless IP cameras plugged directly in the local network hub. The main advantage of such architecture is its flexibility. It enables a single human operator to monitor activities over a broad area using a distributed network of wireless IP-cameras. The architecture of our proposed system focuses on a reliable link between image processing and video content analysis as seen on Fig 2. Hence, integration of image processing within the digital video networked surveillance system itself is inevitable. The proposed IVSS system contains all the modules (video capture, image analysis, image understanding, event generator and field experience). Moreover, it contains an auto-learning module and another module about video retrieval.

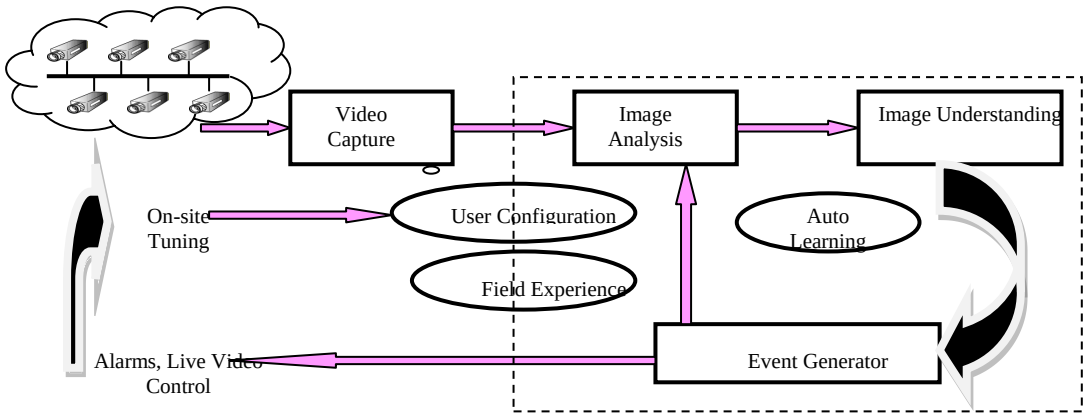


Fig. 2. Video system architecture

The video capture module is responsible of managing the video input data from different IP-cameras over a LAN where each camera can be accessed by its IP address. Accordingly, this module generates report about failures in the video capture process or in the network itself. Moreover, the image understanding module represent the master piece of the IVSS, it includes all AI techniques to figure out the meaning of the scene. Among its tasks: detecting abnormal behavior of human and other moving objects in the scene. The abnormal behavior is forwarded to the event generator module, which generates an alarm for the user and helps the image analysis module to tune the image processing tasks to enhance the behavior for easier perception and monitoring. The detected events based on abnormal behaviors can be modeled and stored in the field experience module for easier access and future detection [14]. This requires temporal and spatial coordination between the cameras to ensure that the same object is being tracked in each, as well as to merge statistical information about the tracked object into a coherent framework.

IV. Network Topology Inference

The aim of tracking description in multiple-camera configuration is to make a link between the tracking and the analysis processes. It is then important to establish correspondences between the objects in different image sequences taken by different cameras. The success of the correlation process depends on how well data is represented, how reliable and adequate the model of data used and how accurate and applicable prior knowledge is. Fig. 5.1 shows the environment wherein our IVSS was implemented. For fixed cameras, a 2D context can be defined by the system administrator identifying areas in the image such as input/output regions, zones to ignore, etc.

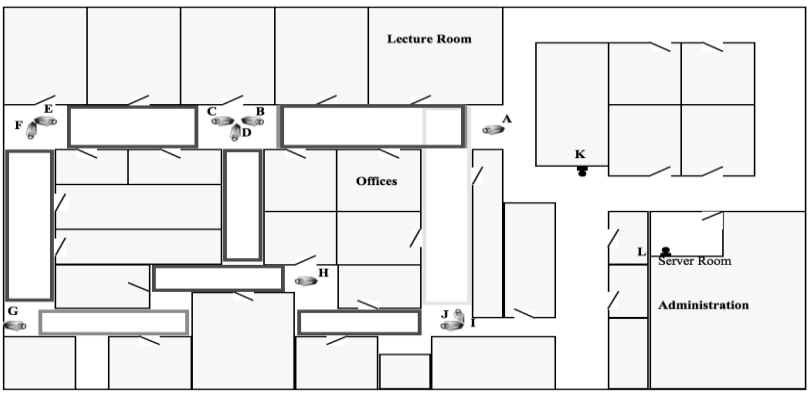


Fig. 3. IP-Camera network in the college building (Environment View)

The interface used in our IVSS is shown by Fig. 4 thereby the operator can have a general view of what happening in the area under surveillance. The two cameras of type CIVS-IPC-3431 (denoted camera K and L) were installed in the server room and just in the nearby corridor for the purpose of identifying persons accessing the server room and checking for access rights. While the ten cameras of type CIVS-IPC-4300 have been installed in the corridors of the first floor of the department to cover a wide closed area where students move and access the lecture rooms, faculty offices, administration offices and toilets. The ten cameras were denoted A, B, C, D, E, F, G, H, I and J as illustrated by Fig.4 It was necessary to create an interface with a mosaic of all available cameras and when clicking on an image, we can see it in a bigger size or in full screen mode. While designing the interface, we had several choices depending on the development language we are going to choose. With Java language, we have the Swing library for developing desktop application. With C++, we have MFC, GTK or QT Framework, which offer all a complete SDK for developing portable cross-platform application especially GTK or QT. Accordingly, Java being a higher-level language, we prefer C++ for an intensive resource consuming application like video processing.

The tracking analysis is a process that generates predefined patterns like objects entering from a defined zone of the image and exiting by another one, or objects which have exceeded a certain speed limit, or also stopped objects for a minimum time which stem from another mobile object. After detecting the motion and tracking the object from frame to frame, it would be interesting to know in which camera the moving object will probably appear after it has disappeared from a given one. This will make the object tracking process easy for the operator in a multi-camera surveillance system. A weighted-graph is best suited for this representation as illustrated in the following Fig. 5.

Fig. 4. Visual interface of the IVS showing 12 cameras video streams

The links between the cameras are assigned one of the three different weights 1, 2 or 3 depending on the relation between their fields of view (FOV) and the spatial configuration of the overlapping area as the following:

1) If there is no overlapping area, but the target after exiting from the FOV of the first camera may appear in the second one
2) If the FOV on camera is incident on the FOV of the other (the target appears on both cameras with different shapes only if it enters the overlapping area.)
3) if the FOV are incident on each other (any target appearing on Camera X should appear on Camera Y with different shape.)

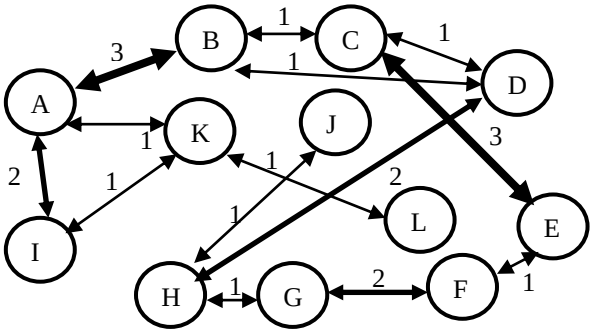
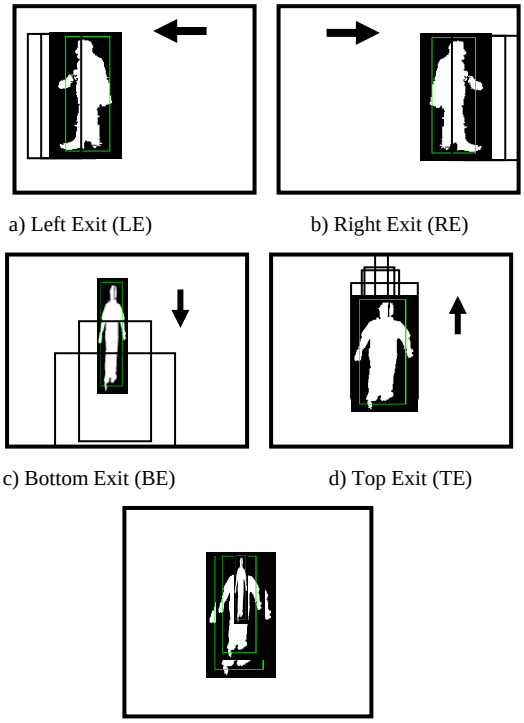


Fig. 5. The topology of a 12-camera network, which can be represented as a weighted graph

We notice that the exit of any target from the FOV of any camera may be done in several ways which then determine the most probable camera wherein the target will reappear. This will enable to assign the correct weight for each link in the graph. In fact, in each view the target object will exit from the scene in four different ways: from the left (Left Exit: LE) (Fig. 6.a), from the right (Right Exit: RE) (Fig. 6.b), from the bottom (Bottom Exit: BE) (Fig. 6.c), from the top (Top Exit: TE) (Fig.6.d), and vanishing point (Vanishing point Exit: VE), (Fig.6.e). Note that, the top exit (TE) and the vanishing point exit (VE) provide the same conclusion. In fact, the top exit takes place when the camera is pointing downward from the horizontal axis. If the camera viewing axis is set horizontal, the TE and VE would be identical. The reason for pointing a bit downward the viewing camera is to cover more details of the near field of view.



a) Left Exit (LE) b) Right Exit (RE)

c) Bottom Exit (BE) d) Top Exit (TE)

e) Vanishing Point Exit (VE)

Fig. 6. Different exit ways of the moving frame

Any moving object exiting from a camera field of view is very likely to appear in another one, if that object does not

leave the global area under surveillance. It is then essential for tracking analysis to establish a link between the different zones. The tracking process will make use of this link to easily find the tracked object when it exits an area and enters another one.

Our system relies on a 12-camera network, trajectory data was derived by motion tracking modules, running for 10-hour period. The dataset consists of 80 trajectories for each camera (the dataset used is collected during the whole day when students are present in the department). Each trajectory is classified for deriving automatically the entry and exit zones in each camera. The detected zones are the nodes of the activity network and are numbered to illustrate the real spatial relationship of the zones.

Moreover, for each exit type (LE, RE, BE, TE or VE) it is possible to build a link table as shown by table 1 for a left exit from any camera in the network (the symbol ⇑ means exit from camera X (column) and appears probably in camera Y (row)). The left, bottom, top and vanishing point exits tracking analysis may be derived easily. The table below has been developed in the IVS tracking analysis process.

TABLE I: CAMERA LINK TABLE FOR TRACKING ANALYSIS BASED ON A LEFT EXIT (LE) ACTIVITY

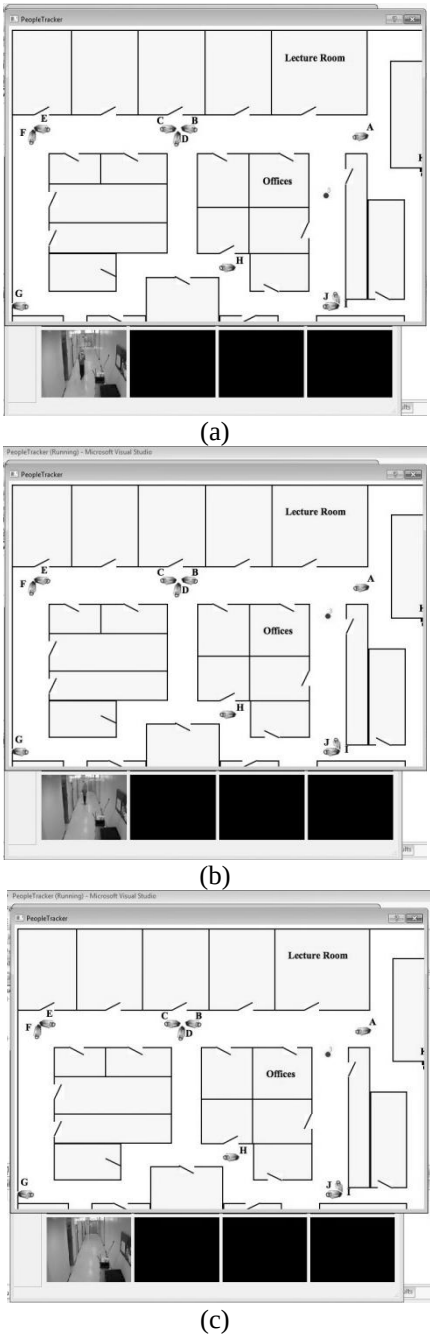
	A	B	C	D	E	F	G	H	I	J	K	L
A												
B										⇑		
C												
D							⇑					
E												
F					⇑							
G					⇑							
H										⇑		
I				⇑				⇑				
J	⇑				⇑							
K									⇑	⇑		
L												

The above table shows the different possibilities of reappearance of a target who exited from left of any camera FOV in the network. The temporal characteristics of the link can be expressed by the period of time taken by the target to move between the two views. On the other hand, the spatial characteristics can be derived from the weighted graph where the weight of the link enhances the probability of seeing the target in the next node. During the learning phase, the link weights are established by letting a known target moving along the network from node to node and changing the direction at each blind area non-covered by the consecutive nodes (camera) in the network. In the case of both the spatial and temporal characteristics these should be represented by a probabilistic model, since it is important to explicitly capture the variability associated with different targets. Targets disappear from the node X and appear at the node Y. A third virtual Similarly, new targets may enter node Y without having passed through node Y. The transition probability is estimated by the formula [15]:

$$T^{p}_{xy}(\tau) = C^{c}_{xy}(\tau) / p_n(1-p_n)$$

where $T^{p}_{xy}(\tau)$ is the probability of any target transiting from camera X to camera Y, $C^{c}_{xy}(\tau)$ is the covariance matrix of the event in the link between camera X and Y and $p_n=E(r_x(t))$ where $r_x(t)$ is the target disappearance rate from node X.

After detecting blobs on the monitored area, the next step was to represent the evolution of this blobs on a Map in real-time. Once the camera scale is fixed, each camera tracker is given the coordinates of the blobs detected and it will be displayed on the map as shown on Fig. 7.



(a)

(b)

(c)

Fig. 7. Display on the map (red point) of a tracked person (shown at bottom left) from one of the camera

V. CONCLUSION

This paper presents a method for inferring the topology of a wireless IP-multi-camera network given initial observations of activity in the monitored region. The initial phase is a learning phase for the system where a well known target moves across all the nodes of the network and performing all the possible scenarios in the blind areas of the

network. It is entirely unsupervised and based only on spatial and temporal information derived by the cameras. The application of the weighted-graph technique to each node of the inter-connected network has eliminated the redundancy of activity paths that are generated by the algorithm. Results from both simulations and experiments have shown the ability of our algorithm to generate accurate results.

REFERENCES

[1] I. S. Kim, H. S. Choi, Y. K. Moo, C. J. Young, and S. G. Kong, "Intelligent visual surveillance — a survey," *International Journal of Control, Automation, and Systems*, vol. 8, pp. 926–939, 2010.

[2] H. M. Dee and S. A. Velastin, "How close are we to solving the problem of automated visual surveillance? A review of real-world surveillance, scientific progress and evaluative mechanisms," *Machine Vision and Applications*, vol. 19, pp. 329–343, September 2008.

[3] D. Vallejo *et al.*, "A cognitive surveillance system for detecting incorrect traffic behaviors," *Elsevier. Expert Systems with Applications*, 2009.

[4] D. Makris and T. J. Ellis, "Automatic Learning of an Activity-Based Semantic Scene Model," *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Miami, FL, USA, pp. 183-188, July 2003.

[5] S. Amari, "Information geometry on hierarchical decomposition of stochastic interactions," *IEEE Trans. on Information Theory*, vol. 47, no. 5, pp. 1701-1711, July 2001.

[6] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "View-invariant motion trajectory-based activity classification and recognition," *Multimedia Systems*, vol. 12, no. 1, pp. 45–54, 2006.

[7] SAGEM *et al.* (2007). Integrated surveillance of crowded areas for public security. [Online]. Available: http://www.iscaps.reading.ac.uk/about.htm.

[8] V. Gouaillier and A. E. Fleurant, "Intelligent video surveillance: Promises and challenges technological and commercial intelligence report," Technical report, CRIM and Technopole Defence and Security, 2009.

[9] D. Duque, H. Santos, and P. Cortez, "Prediction of abnormal behaviors for intelligent video surveillance systems," in *IEEE Symposium on Computational Intelligence and Data Mining*, CIDM 2007, pp. 362–367, April 2007.

[10] H. Zhou and D. Kimber, "Unusual Event Detection via Multi-camera Video Mining," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 1161–1166, 2006.

[11] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.

[12] P. L. Venetianer and H. Deng, "Performance evaluation of an intelligent video surveillance system - a case study," *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1292–1302, 2010.

[13] H. Dee and D. Hogg, "Detecting inexplicable behavior," in *Proc. British Machine Vision Conference*, vol. 477, pp. 486, 2004.

[14] Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti, "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking," *IEEE Signal Processing Mag*., vol. 22, no. 2, pp. 38–51, Mar. 2005.

[15] D. Makris, T. Ellis, and J. Black, "Bridging the Gaps between Cameras" in *Proc. the 2004 IEEE computer society conference on Computer vision and pattern recognition* , pp. 205-210

K. A. Shalfan is currently an associate professor at the department of computer science and information systems at Imam University. He received his M.Sc and Ph.D. from the University of Bradford, England, in 1997 and 2001 respectively. His research interests include: Computer vision, Image processing and analysis, Image rectification, 3-D reconstruction, Target detection and tracking, Image coding and compression, Outdoor/Indoor scene interpretation for remote inspection and surveillance, Robotics, Programmable logic design, System and application programming, IT security and cyber crime.

M. Elarbi-Boudihir is currently a professor in Computer Science, Department of Applied Mathematics at Imam University. He received the M.S. degree, and the Ph.D. degree in computer vision and robotics from the computer science and engineering college at the University of Nancy in 1989, and 1992, respectively. His research interests include machine vision, mobile robot navigation, obstacle avoidance, real-time control, multisensor integration, and computer interfacing and integration