International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:6, 2007

Learning Spatio-Temporal Topology of a Multi-Camera Network by Tracking Multiple People

Yunyoung Nam, Junghun Ryu, Yoo-Joo Choi, and We-Duke Cho

***Abstract*—**This paper presents a novel approach for representing the spatio-temporal topology of the camera network with overlapping and non-overlapping fields of view (FOVs). The topology is determined by tracking moving objects and establishing object correspondence across multiple cameras. To track people successfully in multiple camera views, we used the Merge-Split (MS) approach for object occlusion in a single camera and the grid-based approach for extracting the accurate object feature. In addition, we considered the appearance of people and the transition time between entry and exit zones for tracking objects across blind regions of multiple cameras with non-overlapping FOVs. The main contribution of this paper is to estimate transition times between various entry and exit zones, and to graphically represent the camera topology as an undirected weighted graph using the transition probabilities.

***Keywords*—**Surveillance, multiple camera, people tracking, topology.

I. Introduction

As camera and network technology has improved, the cost of installing a surveillance system has dropped significantly, leading to an exponential increase in the use of security cameras. A typical visual surveillance system is used for observation in wide areas which need security, such as banks, casinos, airports, and military installations. The surveillance system may operate continuously or only as required to monitor a particular event. For example, a traffic monitoring system detects congestion and notice accidents for traffic flow and immediate assistance. Furthermore, increasing use of the surveillance system in public places have been successful at reducing some types of crimes like property crime, for acting as a deterrent in car parks or in other public places.

The visual surveillance system comprises a large network of cameras distributed over wide areas. In the camera network, to discover the relationship between the cameras is a one of the most important issues to develop the intelligent surveillance system. In this paper, we present a conceptual approach for representing the spatio-temporal topology of the camera network with overlapping as well as non-overlapping fields of views (FOVs). In order to determine the spatio-temporal topology, three problems are addressed: (1) detecting the same target object in overlapping FOVs; (2) recognizing occluded object; (3) tracking multiple objects in non-overlapping FOVs.

For object identification, we used the grid-based approach [1] for robustly extracting head and hand regions of moving people in a varying distance from the camera. First, a background subtraction scheme is applied to the sequence of images based on hue and saturation information to classify the foreground and background pixels. The background subtracted image is partitioned into grid patches. The grid patches are classified into background, non-skin foreground and skin foreground classes based on the histogram analysis of patch feature values.

Object occlusion is a challenging problem in the object tracking process. It can be very difficult to deal with by a single camera and cause loss of target in multiple object tracking algorithms. Occlusion can be full or partial, and it can be caused by another foreground object or by a background object. In this paper, we used the merge and split approach which used attributes of atomic blobs and operations such as entry, exit, merge, and split.

A common assumption in multi-camera surveillance systems is that the FOVs of each camera overlap. However, it is not always possible to have overlapping camera views because of monitoring a wide area. In order to overcome this limitation, camera handoff is commonly used to keep track of an object across multiple camera FOVs. Multi-camera tracking with non-overlapping FOVs involves the tracking of targets in the blind region and the correspondence matching of targets across cameras. In this paper, we considered the appearance of people as they move through cameras and the transition time between entry and exit zones. Finally, the systems can automatically learn the topology of an arbitrary network of cameras.

Manuscript received July 31, 2007. This research is supported by the ubiquitous Computing and Network (UCN) Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

Y. Y. Nam was with the Graduate School of Information and Communication, Ajou University, Korea. He is now with Center of Excellence for Ubiquitous System, Ajou University, Suwon, Korea (corresponding author to provide phone: +82-31-219-1693; fax: +82-31-219-1695; e-mail: young022@ gmail.com. cuslab.com).

J. H. Ryu is with the Department of Electrical and Computer Engineering, Ajou University, Suwon, Korea (e-mail: ryujunghun@ gmail.com. ajou.ac.kr).

Y. J. Choi is with the Department of Computer Science and Application, Seoul University of Venture and Information, Seoul, Korea (e-mail: yjchoi@suv.ac.kr. suv.ac.kr).

W. D. Cho is with Center of Excellence for Ubiquitous System, Ajou University, Suwon, Korea, on leave from the Korea Electronics Technology Institute, Sungnam, Korea (e-mail: wdukecho@ gmail.com. cuslab.com).

The rest sections of this paper are organized as follows. Section 2 describes the related work. In Section 3, object identification is described, and Section 4 presents how to learn camera network topology. In Section 5, experimental results are presented. Section 6 concludes this paper.

II. Related Work

Previous work on multiple cameras has dealt with identification, recognition, and tracking of moving objects. Numerous researchers have proposed camera network calibration to achieve robust object identification and tracking from multiple viewpoints.

To indentify objects from cameras, colour is often used in the matching process. Black et al. [2] use a non-uniform quantization of the HSI colour space to improve illumination invariance, while retaining colour detail. KaewTraKulPong and Bowden [3] use a Consensus-Colour Conversion of Munsell colour space (CCCM) as proposed by Sturges et al. [4]. This is a coarse quantization to provide consistent colour representation inter-camera without colour camera calibration.

Object tracking across multiple cameras is usually based on a prior registration of the cameras using common scene features or tracked moving objects. For overlapped cameras, tracking algorithms [5, 6] required camera calibration and a computation of the handoff of tracked objects between cameras. To accomplish this, it needs to share a considerable common FOV with the first.

However, these requirements of overlapped cameras are impractical due to the large number of cameras required and the physical constraint upon their placement. Thus, it needs to be able to deal with non-overlapping region in the system where an object is not visible to any camera. Most single camera tracking algorithms rely on smooth motion using the previously observed velocity to predict the future location using methods such as the Kalman filter [7]. However, motion between cameras is rarely smooth. Thus a number of techniques have been developed to handle the invisible spots and improve object handoff.

Haritaoglu et al. [8] have developed a system which employs a combination of shape analysis and tracking to locate people and their parts (head, hands, feet, torso etc.) and tracks them using appearance models. In [9], they incorporate stereo information into their system. Kettnaker and Zabih [10] presented a Bayesian solution to track objects across multiple cameras where the cameras have non-overlapping FOVs. They used constraints on the motion of the objects between cameras, which are positions, object velocities and transition times. A Bayesian formulation of the problem was used to reconstruct the paths of objects across multiple cameras. They required manual input of the topology of allowable paths of movement and the transition probabilities.

Huang and Russell [11] use a probabilistic approach for tracking cars on a highway. They used a combination of appearance matching and transition times of cars in non-overlapping cameras with known topology. The appearance of the car is evaluated by the mean of the colour and the transition times modeled as Gaussian distributions.

Cai and Aggarwal [12] extend a single-camera tracking system by switching between cameras. They used calibrated cameras with overlapping FOVs. The correspondence between objects was established by matching geometric and appearance features.

Javed et al. [13] present a more general system by learning the camera topology and path probabilities of objects using Parzen windows. Individual tracks are corresponded by maximizing the posterior probability of the spatio-temporal and colour appearance, adapted to account for changes between cameras. The transition probabilities are learnt using a small number of manually labeled trajectories, so this method is based on supervised learning.

Dick et al. [14] use a stochastic transition matrix to describe peoples observed patterns of motion both within and between FOVs. This does not require correspondences, but does need a training phase and does not scale well.

Ellis et al. [15] do not require correspondences or a training phase, instead observing motion over a long period of time and accumulating appearance and disappearance information in a histogram. They used a thresholding technique to look for peaks in the temporal distribution of travel times between entrance-exit pairs; a clear peak suggesting that the cameras are linked.

This approach has been extended by Stauffer [16] and Tieu et al. [17] to include a more rigorous definition of a transition based on statistical significance. They tracked across multiple cameras with both overlapping and non-overlapping FOVs, building a correspondence model for the entire set of cameras. They made an assumption of scene planarity and recovered the inter-camera homographies.

Recently, Gilbert et al. [18] presented an approach to automatically derive the main entry and exit areas in a camera probabilistically using incremental learning, while simultaneously the colour variation inter camera is learnt to accommodate inter-camera colour variations.

III. Object Identification

A. Background Subtraction

Background subtraction is a critical step for moving person detection and tracking. The basic idea is to subtract the current stereo pair from corresponding reference background images to get the foreground.

For background subtraction, it may be used a hue saturation intensity (HSI) colour model has been widely used in color image processing and analysis. We have analyzed the noise characteristics of the HSI colour model and developed an adaptive spatial filtering method to reduce the magnitude of noise and the non-uniformity of noise variance in the HSI colour space.

The hue factor generally represents the unique value of color itself while minimizing the effect of the illumination. The saturation represents the pureness of a color. The intensity is a

(a) Background (b) Silhouette extraction (c) Object detection
Fig. 1 Example of the object detection

measure of the brightness of the colour, which may not always be satisfied to subtract background involving operations over extended time period.

In this paper, we statistically analyze the reference background image in HSI colour space for fifty frames with different illuminations and all pixels of the static background scene image are modeled as Gaussian distribution with respect to the hue and saturation values. After the preprocessing for analyzing the background image, a sequence of images containing a moving human captured from a camera is converted into HSI colour images and subtracted from the reference background image. If the subtraction values are greater than the threshold values which are derived based on the variance values from the background image, those pixels are determined as belonging to the foreground pixels. We classify a pixel into a foreground or background class based on the following equation:

$$R(P_{ij})\begin{cases} 1(foreground), & if\ |H_{ij}-H_{b_{ij}}|>\omega_1\sigma(H_{h_{ij}}) \\ & and\ |S_{ij}-S_{b_{ij}}|>\omega_2\sigma(S_{b_{ij}}), \\ 0(background), & otherwise \end{cases} \quad (1)$$

$H_{ij}: Hue,\ S_{ij}: saturation,\ P_{ij}: currentpixel$

$H_{b_{ij}}: average\ \ Hue,\ S_{b_{ij}}: average\ \ Saturation$

$\sigma(H_{b_{ij}}): Hue\ \ variance,\ \sigma(S_{b_{ij}}): Saturation\ \ variance$

B. Grid-based ROI Extraction

In order to improve object detection, we used the grid-based approach [1] for robustly extracting head and hand regions of a moving human in a varying distance from the camera. The proposed method defines grid images which continuously maintain the foreground and background information in significantly lower resolution than that of original input images. The grid images are adaptively classified into background, non-skin foreground, and skin-foreground classes based on the analysis of the portion of the number of foreground and skin pixels with respect to that of the overall input image pixels. We effectively perform the labeling of skin regions by using the grid image in a low resolution and reduce the unexpected artifacts from the noises.

The result image of background subtraction is partitioned into grid patches of 8 × 8 pixels, and the numbers of foreground pixels and skin pixels are counted for each grid. The ratio of the number of foreground pixels to sixty-four pixels in a grid (F_{ij}) and the ratio of the number of skin pixels to that of foreground pixels in a grid (S_{ij}) are used as patch feature values.

$$F_{ij}=\frac{\#\ of\ \ foreground\ \ pixels\ \ in\ \ Patch(i,j)}{64}$$
$$S_{ij}=\frac{\#\ of\ \ skin\ \ pixels\ \ in\ \ Patch(i,j)}{\#\ of\ \ foreground\ \ pixels\ \ in\ \ Patch(i,j)} \quad (2)$$

Grid patches are classified into a background class, a general non-skin foreground class or a skin foreground class based on two patch feature values, F_{ij} and S_{ij}. That is, we build a grid image which consists of three classes of patches.

$$G_{ij}=\begin{cases} foreground\ \ patch, & if\ \ F_{ij}\ge X_{fg}\ \ and\ \ S_{ij}\langle X_{sk} \\ skin\ \ patch, & if\ \ F_{ij}\ge X_{fg}\ \ and\ \ S_{ij}\ge X_{sk} \\ background\ \ patch, & otherwise \end{cases} \quad (3)$$

$$X_{fg}=\arg\min(f_{fg}(x)),\quad f_{fg}(x)>(1-P_{fg})$$
$$X_{sk}=\arg\min(f_{fg}(x)),\quad f_{sk}(x)>(1-P_{sk})$$
$$where$$
$$f_{fg}(x)=\sum_{i=0}^{x}H_{fg}(i),\quad f_{sk}(x)=\sum_{i=0}^{x}H_{sk}(i) \quad (4)$$
$$P_{fg}=\frac{width(ROI)\times height(ROI)}{width(I)\times height(I)}$$
$$P_{sk}=\frac{\#\ of\ \ skin\ \ pixels}{width(ROI)\times height(ROI)}$$

We first applied the static patch classification using the fixed threshold and performed the proposed adaptive patch classification based on patch histogram analysis. We compared the ROI extraction results of two approaches in Fig. 2. In case of the static patch classification, the small skin region in a long distance image was missed in the skin ROI detection, while the large skin region is robustly detected in a short distance image. In the proposed adaptive patch classification, the small skin regions in the long distance are also detected successfully.

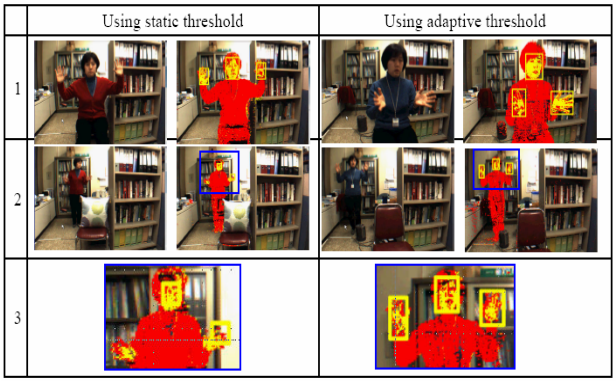


Fig. 2 Background subtraction images in 648×468 resolution. Non-Skin foreground pixels are coloured red and skin foreground pixels are coloured yellow

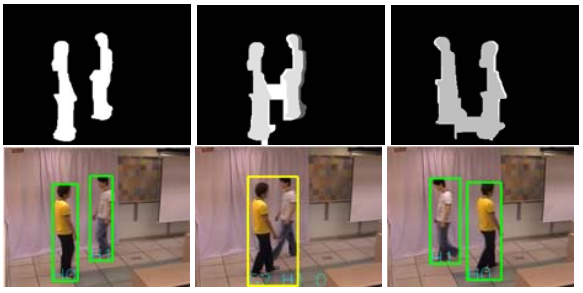
C. Occluded Object Recognition

In order to solve the occlusion problem, we use the HSI colour model until objects come into an occlusion situation and attempt to measure noise according to the change in *blob* surface during merging. The *blob* is defined as being a group of objects such as persons or cars, which acts as a container that can have one or more objects. For each foreground blob, the color model is initialized and a distance measure is used to consistently label the blobs for the rest of the sequence. The model accommodates the presence of partial occlusions, pose variations and illumination changes, through partial updating at every frame.

There are two major approaches for dealing with occlusion using a single camera. The first approach is the Merge-Split (MS) approach that merges the detected occlusion blobs into a single new blob. From that point on, the attribute of original objects is encapsulated into the new blob. The second approach is the Straight-Through (ST) approach that continues to track the individual blobs containing only one object through the occlusion without attempting to merge them. We used the MS approach in this paper.

When a new set of blobs is computed for a frame, an association with the previous frame's set of blobs is required. This association can be an unrestricted. With each new frame, blobs can entry, exit, merge, and split. The system detects that two or more people have merged into a group when the total number of blobs in the frame has decreased and two or more blobs in the previous frame overlap with a single blob in the current frame.

On the other hand, the group can also be split. This event is detected when the total number of blobs in the frame has increased and several blobs in the current frame overlap with a group blob in the previous frame. In order to assign labels after a split, each blob involved in the splitting is segmented as if it was still the group with all the components. Assuming that each person can only be present in one of the blobs involved in the splitting process, it is concluded that a person is present in the blob that contains the largest number of pixels labeled with that person's label. Fig. 3 shows the result of tracking under occlusion. In the frame 662, both human H0 and H1 are seen in a camera. In the frame 669, two humans have merged into a group G2. In the frame 705, the group G2 has split into H0 and H1.



(a) Frame 662 (b) Frame 669 (c) Frame 705

Fig. 3 Tracking results of handling multiple occluded people on single camera

IV. Learning Camera Network Topology

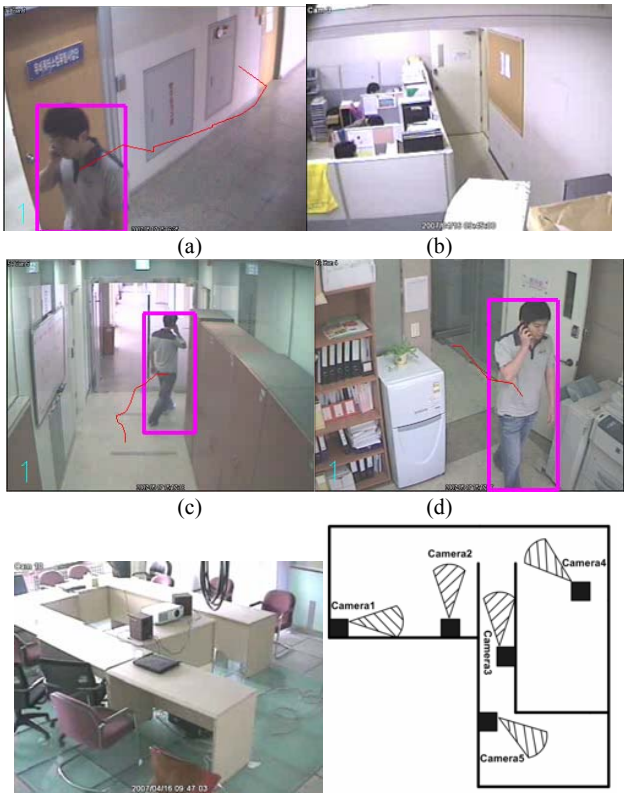
To learn the topology of an arbitrary camera network, we consider the spatio-temporal relationship between cameras, which can be used to support predictive tracking across the camera network. We use the entry-exit and travel-transition time model for spatial relationship and temporal relationship, respectively. The model can be graphically represented as a stochastic state automation such as a graph, where the nodes correspond to camera locations. The links represent possible transitions between connected camera locations and are annotated by a model of the transition time and the probability that an object visible in the first location will become visible next in the second camera.

First of all, the entry-exit model defines entry and exit zones with appearance and disappearance of objects. The entry zones can be new entry zone and re-entry zone according to object identification. The new entry zone is determined when new object is first detected from multiple cameras. On the other hand, the re-entry zone is determined when an object detected from one camera reappear in another camera after disappearance. When an object moves from new entry zone and exit zone, the movement time is called the *travel time* of the object. In addition, an object moves from exit zone and re-entry zone, the movement time is called the *transition time*. Thus, the *travel time* denotes the size of FOV and the transition time represents the camera distance.

In this paper, we define several nodes for camera network topology, which are the *overlapping node*, the *non-overlapping node*, the *virtual node*. In overlapping FOVs, an object entered into one camera and the object can reappear in another camera before disappearance. In this case, the *overlapping node* needs to represent an overlapping edge. In non-overlapping FOVs, the *non-overlapping node* can be an entry node or an exit node whether an object appears or disappears. In particular, an object disappears from one camera and may reappear in the same camera. For this case, we define the *virtual node* to represent the invisible edge.

Finally, we can construct an undirected weighted graph G to represent the camera network. The graph G consists of two sets: a set of vertices $V = \{v_1, v_2, ..., v_n\}$ represents the camera node, the overlapping node, non-overlapping node, and virtual node. The other set of edges $E = \{e_1, e_2, ..., e_n\}$ connects vertices in V which represents edge, overlapping edge, and non-overlapping edge. An edge $e_i \in E$ is represented by the tuple $< v_i, v_j, w_{ij} >$, where $v_i, v_j \in V$. It corresponds to a connection between corresponding two nodes in the camera network. Also, weight w_i is assigned a number representing the normalized distance from v_i to v_j, which is calculated by the inter-camera travel time and the transition time.

$$G =< V, E, W > \quad (5)$$



(a) (b) (c) (d) (e) (f)

Fig. 4 (a)-(e) The tracking environment. (f) The top down layout of the camera system

$$w_{ij} = |T(C_i) - T(C_j)| \tag{6}$$

Typically, a graph can be represented using an adjacency matrix. The adjacency matrix for undirected graph G is a $|V|$ x $|V|$ array, say A, with A(i,j) = A(j,i) = distance of edge(i, j) if there exists an edge between i and j in G. An A(i,j) = 0 if there is no such edge in G.

$$A(i,j) = \begin{cases} w_{ij}, & if \quad (i,j) \in E \\ 0, & if \quad (i,j) \notin E \end{cases} \tag{7}$$

In Fig. 7, if the graph is undirected, every entry is a set of two nodes containing the two ends of the corresponding edge. Thus, its adjacency matrix contains many 0s, which leads to waste of space. Also, the matrix is symmetric for an undirected graph. To improve the space efficiency, we used an adjacency list representation for graph. The graph G is represented by the list $Adj[1...|V|]$ of lists. For each $v \in V$, the list $Adj[v]$ is a linked list of all vertices which is implemented by data structure as shown below.

TABLE I
DATA STRUCTURE FOR LINKED LIST

[0]	[1]	[2]
index of adjacent vertex	weight	reference to next edge node

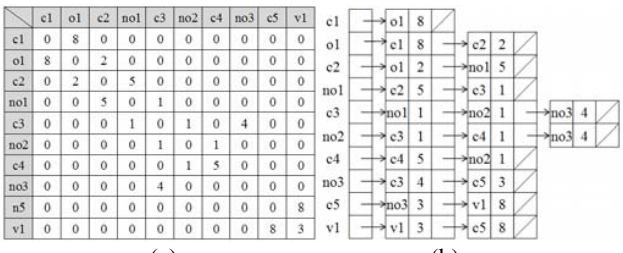
TABLE II
CAMERA ZONE AND TRAVEL-TRANSITION TIME CORRESPONDENCES IDENTIFIED

Time	Camera	New entry	Re-entry	Exit	Travel time	Transition time
09:45:22:12	c1	p1				
09:45:26:87	c4	p2				
09:45:30:34	c2		p1		8	
09:45:31:68	c4			p2	5	
09:45:32:56	c1			p1	10	
09:45:32:81	c3		p2			1
09:45:37:41	c2			p1	7	
09:45:37:94	c3			p2	5	
09:45:38:23	c3		p1			1
09:45:39:18	c3			p1	1	
09:45:40:76	c4		p1			1
09:45:40:56	c5		p2			3
09:45:45:24	c4			p1	5	
09:45:46:74	c3		p1			1
09:45:48:21	c5			p2	8	
09:45:51:81	c3			p1	5	
09:45:54:81	c5		p1			3
09:46:02:64	c5			p1	8	
09:46:03:37	c5	p3				
09:46:05:15	c5		p1			-3
09:46:11:37	c5			p3	8	

V. EXPERIMENTS AND RESULTS

To evaluate the performance of our system, we performed experiments with five CCD cameras (320 x 240 resolutions) and used PCs with Pentium 4 Processors and 1GB of RAM as hardware platform, and Microsoft SQL Server 2000 as underlying DBMS. In addition, we implemented using the C++ and OpenCV library [19] for image processing.

The experiment was done with five cameras in three rooms and a passage as shown in Fig. 4. The training phase lasted for five minutes and the test was run for ten minutes. Table II presents a part of the results that contains entry-exit zones and movement time of three moving objects (p1, p2, p3). As shown in Table II, a p1 loitered in FOV of a camera 4 to start at 09:45:40 and end at 09:45:45. The p1 met a p3 in visible FOV of a camera 5 and the visual system effectively treats partly occluded objects. The p1 disappeared from the camera 5 at 09:46:02 and reappeared in the same camera 5 at 09:46:05, because there is no exit. In this case, the virtual node v1 is created in the topology.



(a) (b)

Fig. 5 (a) An adjacency matrix for graph about a p1. (b) An adjacency list representation

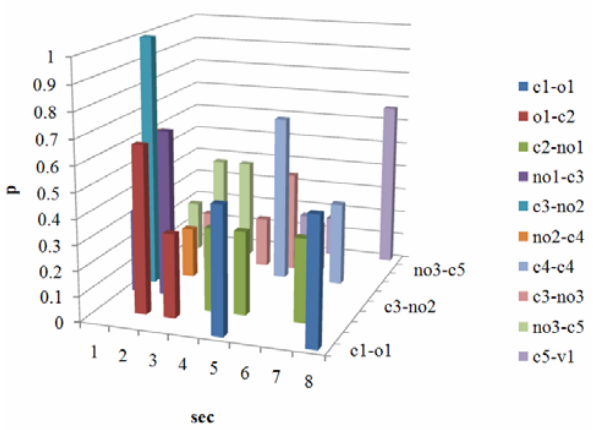


Fig. 6 The probability distribution showing the travel and transition time for ten nodes

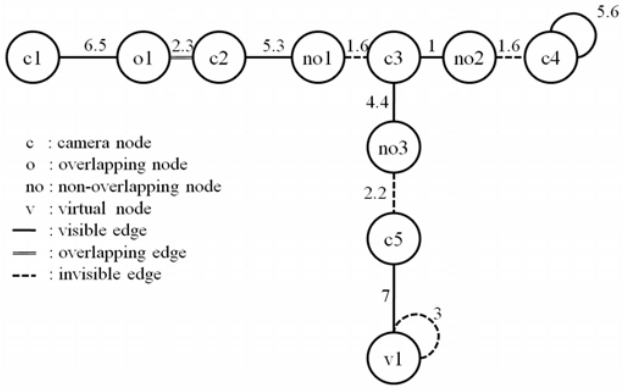


Fig. 7 The camera network topology which is represented by an undirected weighted graph

Fig. 5 presents an adjacency matrix and an adjacency list representation for graph about a p1 based on Table II. As shown in the table, the matrix is symmetric and contains many 0s. Fig. 6 shows the temporal probability distribution of ten edges for five minutes. In conclusion, Fig. 7 shows a camera network topology which is represented by an undirected weighted graph.

VI. Conclusion

In this paper, we have presented a method to automatically construct a spatio-temporal topology of a multiple camera network using the entry-exit and the travel-transition time model. The model is based on an unsupervised learning method from the perspective of statistical modeling, which relies on the matching objects between images from at least two nodes whether overlapping or non-overlapping cameras. In order to represent the spatio-temporal topology, we have created an undirected weighted graph.

For object identification, we used the grid-based approach for robustly and efficiently extracting skin regions of a moving human in a varying distance from the camera. Furthermore, we improved the object identification efficiency using an occluded object recognition that considers both the occluded block and the HSI colour space,

In future work, we plan to extend our spatio-temporal topology to 3-dimensional topology of the building with stairs and elevators. In addition, we will consider compound actions and complex events with the visible FOV using probabilistic methods to recognize different types of object interactions.

References

[1] Y. Choi, K. Kim, W. Cho, "Grid-Based Approach for Detecting Head and Hand Regions," International Conference on Intelligent Computing, Qingdao China, August 21-24, 2007, pp. 1126-1132.
[2] J. Black, T. Ellis, and D. Makris, "Wide Area Surveillance with a Multi-Camera Network," Proc. IDSS-04 Intelligent Distributed Surveillance Systems, 2003, pp. 21–25.
[3] P. KaewTrakulPong and R. Bowden, "A Real-time Adaptive Visual Surveillance System for Tracking Low Resolution Colour Targets in Dynamically Changing Scenes," Journal of Image and Vision Computing. Vol 21, Issue 10, Elsevier Science Ltd, 2003, pp. 913–929.
[4] J. Sturges and T. Whitfield, "Locating Basic Colour in the Munsell Space," Colour Research and Application, 1995, pp. 364-376.
[5] Q. Cai and J. Agrarian, "Tracking Human Motion using Multiple Cameras," Proc. International Conference on Pattern Recognition, 1996, pp. 67–72.
[6] P. Kelly, A. Katkere, D. Kuramura, S. Moezzi, and S. Chatterjee, "An Architecture for Multiple Perspective Interactive Video," Proc. of the 3rd ACE International Conference on Multimedia, 1995, pp. 201–212.
[7] G. Welch and G. Bishop, "An Introduction to the Kalman Kilter," Technical Report 95-041,University of North Carolina at Chapel Hill, 1995.
[8] I. Haritaoglu, D. Harwood, L. Davis, "W4:Who, When, Where, What: A Real Time System for Detecting and Tracking People," Third International Conference on Automatic Face and Gesture, 1998.
[9] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4S: A realtime system for detecting and tracking people in 2 1/2D," 5th European Conference on Computer Vision, Freiburg, Germany, 1998.
[10] V. Kettnaker, R. Zabih, "Counting People from multiple cameras," in IEEE ICMCS, Florence, Italy, 1999, pp. 267-271.
[11] T. Huang and S. Russell, "Object Identification in a Bayesian Context," Proc. International Joint Conference on Artificial Intelligence (IJCAI-97), Nagoya, Japan, 1997, pp. 1276–1283.
[12] Q. Cai and J.K. Aggarwal, "Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized video Streams," 6th International conference on Computer Vision, Bombay, India, 1998, pp. 356-362.
[13] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. "Tracking Across Multiple Cameras with Disjoint Views". Proc. IEEE International Conference on Computer Vision, 2003, pp. 952–957.
[14] A. Dick and M. Brooks, "A Stochastic Approach to Tracking Objects Across Multiple Cameras," Australian Conference on Artificial Intelligence, 2004, pp. 160–170.
[15] T. J. Ellis, D. Makris, and J. Black, "Learning a multi-camera topology," In Joint IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), 2003, pp. 165–171.
[16] C. Stauffer, "Learning to track objects through unobserved regions," In IEEE Computer Society Workshop on Motion and Video Computing, 2005, pp. 96–102.
[17] K. Tieu, G. Dalley, and W. Grimson, "Inference of nonoverlapping camera network topology by measuring statistical dependence," In Proc. IEEE International Conference on Computer Vision, 2005, pp. 1842–1849.
[18] A. Gilbert, R. Bowden, "Tracking Objects Across Cameras by Incrementally Learning Inter-camera Colour Calibration and Patterns of Activity," In Proc European Conference Computer Vision, 2006, pp. 125-136.
[19] Intel Open Source Computer Vision Library, URL http://sourceforge.net/projects/opencvlibrary/