International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.3 (2016), pp.239-248
http://dx.doi.org/10.14257/ijmue.2016.11.3.23

Wide Area Object Tracking using Multiple Cameras with Mixed Configuration of Over-lapping and Non-Overlapping FOV

Daw-Tung Lin[1*] and Kuan-Yu Wu[2]

Department of Computer Science and Information Engineering
National Taipei University, Taiwan
Email: [1]dalton@mail.ntpu.edu.tw, [2]garywuntpu@gmail.com*

Abstract

Seamless object tracking in wide area is an important and yet challenging issue of intelligent visual surveillance multi-camera collaboration tracking approach has been becoming attractive. In this work, a flexible and integrated system is constructed for object tracking using multiple cameras either with overlapping fields of views (FOVs), non-overlapping FOVs or mixed configurations for wide areas surveillance. The Kalman filter is applied for object tracking in a single camera and object motion prediction is implemented across blind regions for non-overlapping FOVs. Furthermore, a client and server architecture is constituted using TCP/IP network. The FOV lines are used to guide camera switching. In addition, we integrate various features for object matching. Finally, this system can track objects in a wide area no matter the FOVs are overlapping, non-overlapping or mixed situations.

***Keywords**: Wide area surveillance, seamless object tracking, collaborative multiple camera tracking, overlapping FOV, non-overlapping FOV*

1. Introduction

Seamless object tracking in wide area is an important and yet challenging issue of intelligent visual surveillance [1]. Multi-camera collaboration tracking approach has been becoming attractive [2-6]. The benefit of multi-camera tracking is that it can adapt to variant view angles in the wide environment [7-9]. The categories of multi-camera tracking problems can be classified into installation, calibration, cameras switching and object matching [10]. Tracking objects in non-overlapping fields of camera views is the most critical problem of multiple cameras tracking [11]. Zhu *et al*. use the distance between FOV line and object feet to predict location in camera view. They also adopt the color histogram for identifying the objects [10]. Javed *et al*. adjusts the color disparity in two cameras and uses the color information and object space-time features to integrate the object in non-overlapping area [12]. Chilgunde *et al*. use Kalman filter to robustly track each target motion in each camera view and the common ground plane view [13]. In this work, we integrate several single camera tracking system (client) and fusing mechanism (server) to communicate with each clients and server. For single camera tracking, we utilize Kalman filter and Mean Shift algorithms to track the moving objects and tackle the occlusion problem. Here, we estimate the cameras hand-off timing based on whether the object leaved the view or not. Finally, we analyze the overlapping and non-overlapping FOVs issue by predicting the object moving position in the blind region.

The rest of this paper is organized as follows. Section 2 elucidates the system architecture of the proposed wide area object tracking approach. Section 3 illustrates further issues about multiple cameras tracking. The main task of this chapter is to integrate multiple cameras and keep track for the same object in different views, including overlapping and non-overlapping FOV situations. The system comprises calibration, camera switching, object motion prediction and object matching. Finally, we try to deal

ISSN: 1975-0080 IJMUE

with this work in both conditions. Section 4 describes the experimental environment and demonstrates the simulation results. The objects in overlapping and non-overlapping case are tracked by miscellaneous features. Finally, conclusion is drawn in Section 5.

2. System Architecture

In this paper, the multiple cameras object tracking system requires more than two cameras to work as network cooperatively and synchronously to seamlessly track objects. This system is composed of two components including clients and server [14]. Each client executes object tracking algorithm for a single channel video input. Next, client sends objects' features including width, height, position and color histogram to the server. Furthermore, the server not only integrates the features, but also manages the critical section of camera hand-off. The proposed system architecture is shown in Figure 1. The major difference between the proposed system and that of Lin and Huang's [14] is that our system can process multiple cameras object tracking for both overlapping FOV case and non-overlapping FOV case at the same time. To achieve this goal, the server calculates the corresponding view of each camera by homography transformation. Furthermore, the server can obtain the correspondence between FOV of each camera and common ground plane view [9]. Based on the FOV lines, the system can determine the location of objects, and switches to the proper camera for tracking. Therefore, this system can seamlessly track objects in different situations including overlapping, non-overlapping or mixed overlapping and non-overlapping FOVs. Finally, the system provides the conspicuous outcome for multiple cameras object tracking on the common ground plane using homography transformation. The installation configuration of multiple cameras tracking consists of overlapping and non-overlapping FOVs as shown in Figure 2.

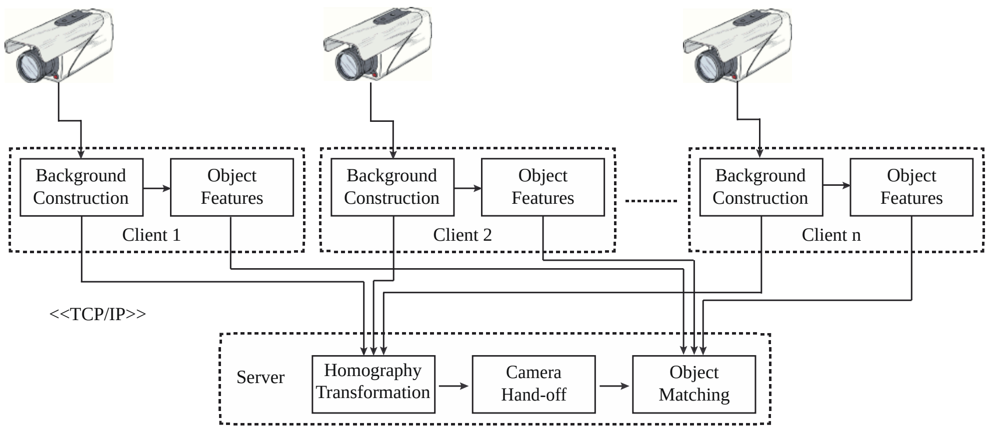

Figure 1. System Architecture of Proposed Multi-Camera Collaborative Tracking System

3. Multiple Camera Tracking

Multiple camera configuration is essential for object tracking in a wide region. Nevertheless, the object appears in one camera view is often very different from that perceived in other cameras [15]. As the different object sizes, it is impossible to use the feature of height and width to match with each other. It is because the objects are constructed from 2-D camera images, in fact the object is 3-D model in real world. Therefore, we must adjust the space of 3-D axes for each camera. Furthermore, there are still some problems in multiple cameras tracking system such as calibration, cameras hand-off, object matching. Also, to track object in the non-overlapping area Chilgunde *et al*. trans- form the views from each camera to a common ground plane view containing all the camera views using homography and also use a Kalman filter in this common ground view to track the targets [13].

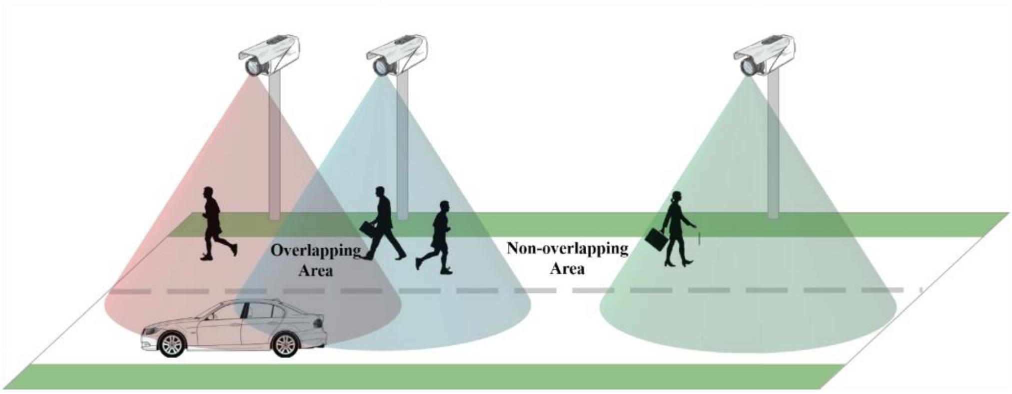

Figure 2. Installation Configuration of Multiple Camera Tracking in Wide Area

3.1. Camera Hand-Off

In order to track objects across multiple cameras, the central server must perform hand-off between the corresponding cameras and keep tracking of the objects. Khan *et al*. proposed FOV lines for camera switching for overlapping views [16]. The significant landmarks of FOV are the boundaries of camera view, transformed by homography method.

Algorithm 1: Camera Hand-off

---

1. Associate two cameras C_i and C_j, let $p_{center}(C_i)$ and $p_{center}(C_j)$ represents the center coordination of cameras C_i and C_j, respectively;
2. For each moving object O_k, compute distances $D_i(O_k)$ and $D_j(O_k)$, where $D_i(O_k)$ stands for the distance between object O_k and $p_{center}(C_i)$ and $D_j(O_k)$ stands for the distance between object O_k and $p_{center}(C_j)$;
3. Initially set object O_k status flag $s(O_k)= 0$;
4. ◇ case: object O_k moves from C_i to C_j
5.   while system detects object O_k do
6.   if O_k is in C_i then
7. $s(O_k)= 1$;
8. O_k is tracked by C_i;
9.   end if
10.   if (O_k is in C_i) and (O_k is in C_j) then
11. $s(O_k)= 2$;
12.   if $D_i(O_k) \leq D_j(O_k)$ then
13. O_k is tracked by C_i;
14.   else
15. O_k is tracked by C_j;
16.   end if
17.   end if
18.   if O_k is in C_j then
19. $s(O_k)= 1$;
20. O_k is tracked by C_j;
21.   end if
22.   end while;

◇ case: object O_k moves from C_j to C_i
while system detects object O_k do
if O_k is in C_j then
$s(O_k)= 1$;
O_k is tracked by C_j;
end if
if (O_k is in C_j) and (O_k is in C_i) then
$s(O_k)= 2$;
if $D_j(O_k) \leq D_i(O_k)$ then
O_k is tracked by C_j;
else
O_k is tracked by C_i;
end if
end if
if O_k is in C_i then
$s(O_k)= 1$;
O_k is tracked by C_i;
end if
end while;

For example, if the object O_k is located on plane *i* of camera C_i, then we define the object by local label $L_i(O_k)$ with status $s(O_k)$=1, and this object is labeled as well by global label $L_{common}(O_k)$ by the system. We assume the object O_k would arrive in the overlapping area between FOV lines of cameras C_i and C_j. When the object enters the overlapping area with status $s(O_k)$=2, this object appears in the camera C_j as $O_{k'}$. These two objects O_k and $O_{k'}$ are the same object in the real world. Therefore, the system needs to initiate the identification function for these two objects in different cameras and clients. To achieve this goal, we compare the feature histogram of object with another and search for the most similar object. For instance, if objects O_k and $O_{k'}$ are the same, the global label $L_{common}(O_{k'})$ will be replaced by $L_{common}(O_k)$. Finally, the system switches to the nearest plane and keep tracking of the object on the common ground plane. Algorithm 1 describes the camera hand-off process. In conclusion, the camera hand-off not only helps us to understand the statuses of objects, but also identify objects consistently.

3.2. Tracking in the Blind Region

In Algorithm 1, we have not yet resolved with the condition when object Ok enters into the non-overlapping FOV area or blind region. To track objects in the blind region, the system must reconstruct the object moving path. Chilgunde *et al*. applied Kalman filter to robustly track objects in the blind region [13]. The system transformed the centroid coordinates of each object and velocity to the ground plane view. When the object steps into the blind region or blocked by the background item, Kalman filter is applied to predict the coordinates of the centroid of each object. Furthermore, to reduce the motion vector noise which caused by false foreground segmentation, the system retains the significant motion of objects by filtering out the unstable motion vector. Besides, if the object goes to the opposite direction, the system should reconstruct the prediction model. We average the current vector and the previous vector to obtain the main motion vector when the angle between the current vector and the previous vector is smaller than sixty degrees, otherwise, the system updates the current motion vector. Chilgunde *et al*. utilized a state vector which consists of the co-ordinates of the centroid of the target and its velocity in the common ground plane view [13]. The Kalman filter of target motion can maintain its state vector. The state vector of the Kalman filter for the object O_k in frame n is defined as $S^n(O_k)= [X^n(O_k), Y^n(O_k), V^n_x(O_k), V^n_y(O_k)]^T$, where $X^n(O_k)$ and $Y^n(O_k)$ are the centroid coordinates of object O_k, and $V^n{}_x(O_k)$ and $V^n{}_y(O_k)$ denote the x and y components of the motion velocity of object O_k in the common ground view, respectively. The state equation of a constant velocity model is shown as below:

$$S^{n+1}(O_k) = AS^n(O_k) + \omega_k, \quad (1)$$

Where A is a 4×4 identity matrix and ω_k is a 4×1 zero mean process noise vector. The position and motion measurement $Z^n(O_k)$ is given by the measurement equation:

$$Z^n(O_k) = NS^n(O_k) + \delta_k, \quad (2)$$

Where N is a 4×4 identity matrix and δ_k is a 4×1 zero mean measurement noise vector.

To obtain the stable motion vector $V^n(O_k)$, the measurement equation is given as:

$$V^n(O_k) = (V^n{}_x(O_k),\ V^n{}_y(O_k)), \quad (3)$$

Where

$$V^n(O_k) = \begin{cases} \left(\dfrac{V_x^{n\text{-}1}(O_k) + V_x^{n\text{-}2}(O_k)}{2}, \dfrac{V_y^{n\text{-}1}(O_k) + V_y^{n\text{-}2}(O_k)}{2}\right), & 0.5 \le \dfrac{\overrightarrow{V^{n\text{-}1}}(O_k)\overrightarrow{V^{n\text{-}2}}(O_k)}{\left|\overrightarrow{V^{n\text{-}1}}(O_k)\right|\left|\overrightarrow{V^{n\text{-}2}}(O_k)\right|} \le 1 \\ V^{n\text{-}1}(O_k), & otherwise \end{cases} \quad (4)$$

In conclusion, the system predicts the object moving location using the above-mentioned prediction model when object enters the blind region. Moreover, we can retrieve the path of object in the non-overlapping area. The prediction process is described as follows:

1. Calculate the angle between the current vector and the previous vector.
2. Update the motion vector and the state vector of the Kalman filter.
3. Predict the object position in next frame.

8. Experimental Results

In this section, we present the experimental results of the proposed system on object matching in cases of overlapping FOV, non-overlapping FOV, and mixed situations. The proposed system has been implemented in Visual C++ 7.0 under a Windows 7 platform. For a 320×240 resolution video sequence, our system operates at about 10 frames per second on Intel Core i5-2410M 2.30GHz processor with 4GB RAM. Several video sequences taken in campus were applied to demonstrate the system performance, containing different size of objects, object occlusion scenarios, and pedestrians with different moving directions. The most distinctive characteristics between the proposed system and the other multiple camera object tracking systems is that the proposed system can track objects in various situations using any numbers of cameras. Fujimura *et al*. developed a pedestrian tracking system using a camera network with overlapping FOVs [17]. Chen *et al*. proposed an object tracking system using an adaptive learning method with non-overlapping FOVs [18-19]. Lin *et al*. developed an object tracking system with overlapping FOVs or non-overlapping FOVs [14]. Chu *et al*. proposed a self-organized and scalable multiple-camera tracking system with non-overlapping views [11]. Liem and Gavrila present a multiple persons tracking system with overlapping views by integrating appearance cues and temporal information [20]. The comparison of various approached is shown in Table 1. Table 2 and Table 3 present the performance of multiple cameras tracking with overlapping FOV and non-overlapping FOV, respectively. The proposed system performs a satisfactory tracking accuracy. The snapshots of seamless object tracking results with camera overlapping FOV and non-overlapping FOV scenarios are shown in Figure 3 and Figure 4, respectively.

Table 1. System Configuration Comparison of Various Approaches

Approach	Overlapping FOV	Non-overlapping FOV	Mixed Configuration
Fujimura *et al*. [17]	Yes	No	No
Chen *et al*. [18,19]	No	Yes	No
Lin *et al*. [14]	Yes	Yes	No
Chu *et al*. [11]	No	Yes	No
Liem and Gavrila [20]	Yes	No	No
Proposed	Yes	Yes	Yes

Table 2. Experimental Results with Overlapping FOV Cases

Video	Video Length	Missing Tracking	False Tracking	Detection Lag
Video 1	11'10"	1/29	1/29	0 sec.
Video 2	16'45"	0/30	1/30	0 sec.

Table 3. Experimental Results with Non-Overlapping FOV Cases

Video	Video Length	Missing Tracking	False Tracking	Detection Lag
Video 3	22'15"	3/30	8/30	0.9 sec.
Video 4	8'22"	3/20	6/20	1.1 sec.

Table 4. Experimental Results of Mixed Configuration of Overlapping and Non-Overlapping FOV Cases

Video	Video Length	Missing Tracking	False Tracking	Detection Lag
Video 5	22'25"	1/16	1/16	0.9 sec.
Video 6	27'39"	3/11	1/11	2.0 sec.

Furthermore, we demonstrate the performance of multiple cameras tracking with mix of overlapping and non-overlapping FOVs on the outdoor test video sequences. We present the performance for all situations in Table 4. In Figure 5, Figure 6 and Figure 7, we show the multiple tracking sample with different distances. In Figure 5, object #8 appears in the plane of camera 1, and it will enter the blind region. The system gives the same tag to the new object which was detected by camera 2 after object matching procedure, as show in Figure 6. Finally, object #8 enters the overlapping region, and the system activates the “camera hand-off” procedure to keep tracking of the same object, as show in Figure 7.

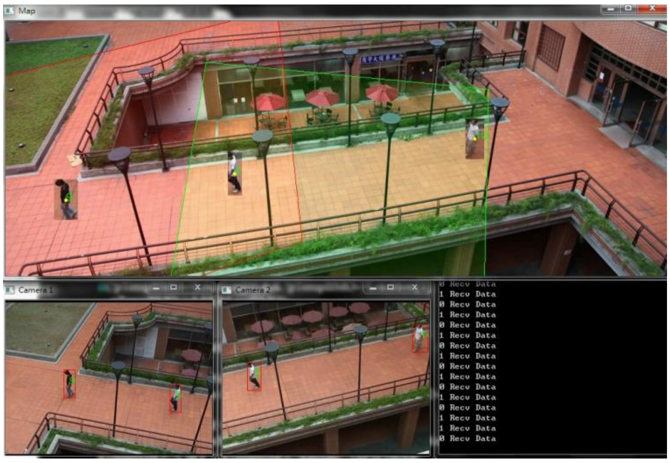

Figure 3. Snapshot of Seamless Object Tracking with Camera Overlapping FOV Scenario

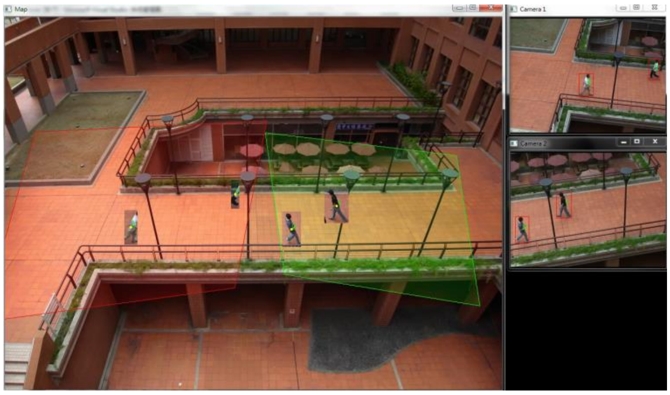

Figure 4. Snapshot of Seamless Object Tracking Result with Camera Non-Overlapping FOV Scenario

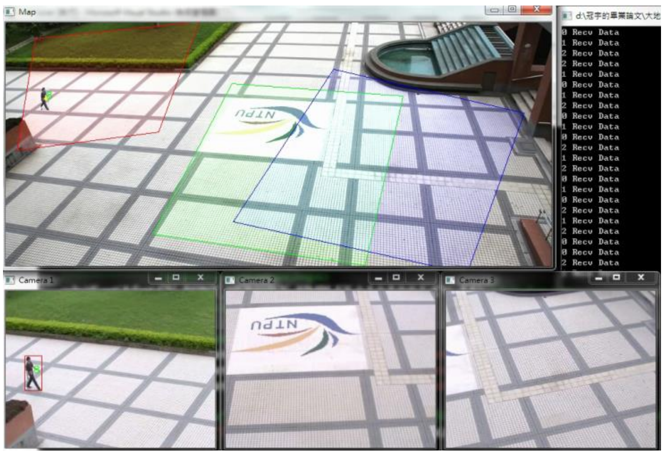

Figure 5. Object Tracking with Mixed Configuration of Overlapping and Non-Overlapping FOV Scenario: Object #8 Steps into Camera 1

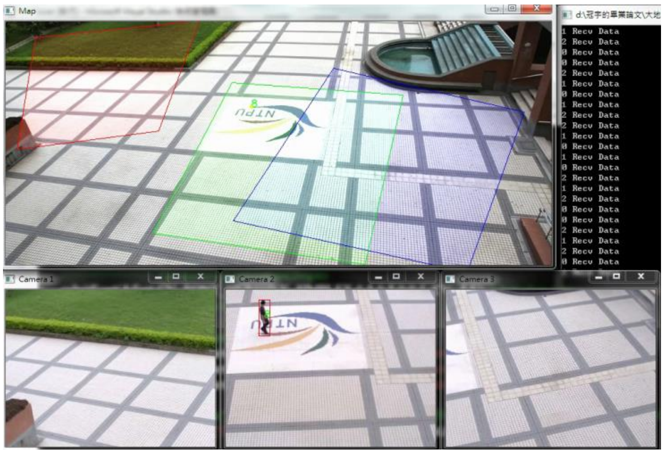

Figure 6. Object Tracking with Mixed Configuration of Overlapping and Non-Overlapping FOV Scenario: Object #8 Passes through the Blind Region between Camera 1 and Camera 2

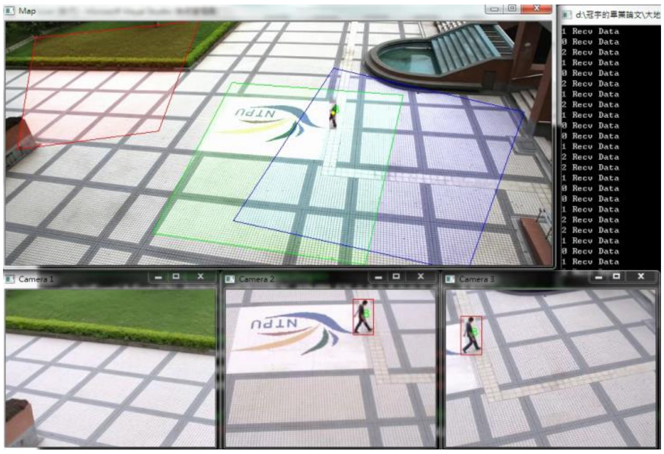

Figure 7. Object Tracking with Mixed Configuration of Overlapping and Non-Overlapping FOV Scenario: Object #8 Enters the Overlapping Area between Camera 2 and Camera 3

9. Conclusion

We have developed a flexible multi-camera tracking system for both overlapping and non-overlapping FOVs configurations and mixed configurations for wide area surveillance seamless object tracking. This system deals with the object matching from different FOVs. The server builds the corresponding FOV lines on the common ground map. When the object steps into any FOV, the system assigns a tag to the object and tracks it continuously. If the object enters the overlapping or non-overlapping area, the system will activate the "camera hand-off" procedure. In the overlapping case, the system finds the highest similarity probability of object and switches to corresponding camera and tracks it. For non-overlapping case, the system predicts the object moving position and reconstructs the path in the blind region. When the object located in the blind region and steps into a common plane, the system will re-activate the "object matching" procedure again to search for the highest object similarity. Finally, the tracking performance of the proposed system was satisfactory for all cases of overlapping FOV, non-overlapping FOV and mixed configurations.

Acknowledgment

This work was supported in part by the National Science Council, Taiwan, grant NSC 99-2221-E-305 -012 -MY2.

References

[1] E. Strigel, D. Meissner and K. Dietmayer, "Vehicle Detection and Tracking at Intersections by Fusing Multiple Camera Views", Proceedings of the IEEE Intelligent Vehicles Symposium (IV), **(2013)**, pp. 882-887.

[2] L.-J. Zhu, J.-N. Hwang and H.-Y. Cheng, "Tracking of Multiple Objects Across Multiple Cameras with Overlapping and Non-Overlapping Views", Proceedings of the IEEE International Symposium on Circuits and Systems, **(2009)**, pp. 1056-1060.

[3] M. Song, T. Dachent and S. J. Maybank, "Sparse Camera Network for Visual Surveillance--A Comprehensive Survey", arXiv preprint arXiv: 1302.0446, **(2013)**.

[4] Q. Cai, A. Sankaranarayanan, Q. Zhang, Z. Zhang and Z. Liu, "Real Time Tead Pose Tracking from Multiple Cameras with a Generic Model", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), **(2010)**, pp. 25-32.

[5] Q. Zhang and K. N. Ngan, "Segmentation and Tracking Multiple Objects Under Occlusion from Multiview Video", IEEE Transactions on Image Processing, vol. 20, no.11, **(2011)**, pp. 3308-3313.

[6] X. Wang, "Intelligent Multi-Camera Video Surveillance: A Review", Pattern Recognition Letters, vol. 34, no.1, **(2013)**, pp. 3-19.

[7] Y. Raja and S. Gong, "Scalable Multi-Camera Tracking in a Metropolis", Person Re-Identification, Springer London, **(2014)**, pp. 413-438.

[8] R. Eshel and Y. Moses, "Tracking in a Dense Crowd Using Multiple Cameras", International Journal of Computer Vision, vol. 88, no. 1, **(2010)**, pp. 129-143.

[9] M. Hassan, A. Mehmood and M. F. Khan, "An Efficient Method of Tracking Across Multiple Cameras", Proceedings of the IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (Trust Com), **(2011)**, pp. 1477-1481.

[10] W. Hu, T. Tan, L. Wang and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors", IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 34, no. 3, **(2004)**, pp. 334–352.

[11] C.-T. Chu, K.-H. Lee and J.-N. Hwang, "Self-Organized and Scalable Camera Networks for Systematic Human Tracking Across Non-overlapping Cameras", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), **(2013)**, pp. 2322-2326.

[12] O. Javed, K. Shafique and M. Shah, "Appearance Modeling for Tracking in Multiple Non-overlapping Cameras", Proceedings of the IEEE Computer Vision and Pattern Recognition, vol. 2, **(2005)**, pp. 26–33.

[13] A. Chilgunde, P. Kumar, S. Ranganath and H. WeiMin, "Multi-Camera Target Tracking in Blind Regions of Cameras with Non-overlapping Fields of View", Proceedings of the BMVC04, **(2004)**, pp. 1-10.

[14] D. T. Lin and K. Y Huang, "Collaborative Pedestrian Tracking and Data Fusion with Multiple Cameras", IEEE Transactions on Information Forensics and Security, vol. 6, no. 4, **(2011)**, pp. 1432–1444.

[15] N. T. Pham, K. Leman, R. Chang, J. Zhang and H. L. Wang, "Fusing Appearance and Spatio-temporal Features for Multiple Camera Tracking", Multimedia Modeling, Springer International Publishing, **(2014)**.

[16] S. Khan and M. Shah, "Consistent Labeling of Tracked Objects in Multiple Cameras with Overlapping Fields of View", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 10, **(2003)**, pp. 1355– 1360.

[17] K. Fujimura, Y. Hyodo and S. Kamijo, "Pedestrian Tracking Across Panning Camera Network", International Journal of Intelligent Transportation Systems Research, vol. 8, no. 1, **(2010)**, pp. 10-25.

[18] K. W. Chen, C. C. Lai, Y. P. Hung and C. S. Chen, "An Adaptive Learning Method for Target Tracking Across Multiple Cameras", Proceedings of the IEEE Computer Vision and Pattern Recognition, **(2008)**, pp. 1-8.

[19] K. W. Chen, C. C. Lai, P. J. Lee, C. S. Chen and Y. P. Hung, "Adaptive Learning for Target Tracking and True Linking Discovering Across Multiple Non-overlapping Cameras", IEEE Transactions on Multimedia, vol. 13, no. 4, **(2011)**, pp. 625-638.

[20] M. C. Liem and D. M. Gavrila, "A Comparative Study on Multi-Person Tracking Using Overlapping Cameras", Computer Vision Systems, Springer Berlin Heidelberg, **(2013)**, pp. 203-212.

Authors

Daw-Tung, Lin, He received the B.S. degree in control engineering from National Chiao-Tung University, Hsin Chu, Taiwan, in 1985, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, MD, U.S.A., in 1990 and 1994, respectively. He is currently the Executive Vice President for Academics and Research of National Taipei University, Taiwan. From 2011 to 2015, he served as the Dean of Academic Affairs of National Taipei University, Taiwan. Previously he served as the Director of the Computer Center with Chung Hua University from 2001 to 2003, the Dean of the College of Engineering with Chung Hua University from 2003 to 2005, the Chairperson of the Department of Computer Science and Information Engineering with National Taipei University from 2006 to 2009, the Director of the Graduate Institute of Communication Engineering with National Taipei University from 2010 to 2011, and the Dean of Academic Affairs of National Taipei University from 2011 to 2015. From 1995 to 2004, he was an Associate Professor of the Department of Computer Science and Information Engineering at the Chung Hua University, Taiwan. He has been with the National Taipei University since 2005, where he became a tenured Professor of computer science and information engineering in 2009. He is a senior member of IEEE. He has been an active contributor to the computer vision and image processing international research societies and published more than 112 papers. His research interests include image processing, computer vision, pattern recognition, and intelligent surveillance.

Kuan-Yu Wu, He received the master degree in computer science and information engineering from National Taipei University, New Taipei City, Taiwan (R.O.C). His research interests include computer vision and embedded systems. His specialty is hetrogenious system integration.

