中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-09-007

A Multi-Camera Tracking System That Can always Select A Better View to Perform Tracking

**Shih-Wei Sun, Hsing-Yuan Lo, Hong-Ju Lin,
Yong-Sheng Chen, Fay Huang, Hong-Yuan Mark Liao**

July 1, 2009 || Technical Report No. TR-IIS-09-007

http://www.iis.sinica.edu.tw/page/library/LIB/TechReport/tr2009/tr09.html

A Multi-Camera Tracking System That Can always Select A Better View to Perform Tracking

Shih-Wei Sun*, Hsing-Yuan Lo†*, Hong-Ju Lin‡*, Yong-Sheng Chen†, Fay Huang‡, and Hong-Yuan Mark Liao*†‡
* Institute of Information Science, Academia Sinica, Taipei, Taiwan
†Dept. Computer Science and Information Engineering, National Chiao Tung Univ., Hsin-Chu, Taiwan
‡Institute of Computer Science and Information Engineering, National Ilan Univ., I-Lan, Taiwan
E-mail: swsun@iis.sinica.edu.tw, hylo@iis.sinica.edu.tw, hjlin@iis.sinica.edu.tw,
yschen@cs.nctu.edu.tw, fay@niu.edu.tw, liao@iis.sinica.edu.tw

***Abstract*—In this paper, we propose a new multiple-camera people tracking system that is equipped with the following functions: (1) can handle long-term occlusions, complete occlusions, and unpredictable motions; (2) can detect arbitrary sized foreground objects; (3) can detect objects with much faster speed. The main contribution of our method is twofold: 1) An M-to-one relationship with only point homography matching for occlusion detection can achieve efficiency; 2) A view-hopping technique based on object motion probability (OMP) is proposed to automatically select an appropriate observation view for tracking a human subject.**

I. Introduction

Tracking multiple people using multi-camera is a challenging issue in recent years. When a suspicious human subject walks in an environment monitored by a multi-camera surveillance system, the cooperation among different cameras becomes very important. According to the literature [1], [2], [3], multi-camera tracking techniques have shifted from the monocular approaches [4], [5], [6], [7], [8], [9] toward the multi-camera approaches [1], [2], [3]. The tracking approaches using monocular camera aim to track people by a single camera. Most of the existing systems adopted blob-based [4], [5], [6] and color-based [7], [8], [9] approaches to perform tracking. A set of features extracted from a human subject is updated sequentially in both above mentioned approaches. However, the major drawback of the above systems is that when a human subject is occluded, there is no way to keep updating the changes across time. Under these circumstances, once a human subject is suddenly occluded and then re-appears in the field of view, the tracking system may not be able to catch him/her due to a significant change of pose, shape, or illumination condition. Some approaches have been proposed for solving the occlusion problem. For example, Kalman filtering [10], [11] and particle filtering [12], [13] are proposed to predict motions when occlusion occurs. However, no matter Kalman filtering or particle filtering is applied, they can only deal with a short-term occlusion problem due to their prediction-based nature. To handle a long-term occlusion problem, some other approaches need to be proposed. Among different potential solutions, utilizing multiple cameras to work together as a team is one of the best solutions to this problem. There are a number of difficult issues associated with a multi-camera surveillance system. These issues include: fusion of data extracted from multiple cameras, illumination difference at different locations, camera placement problem, etc.

In recent years, homography mapping [14], [15] has been applied to the problem of multiple-camera-based video surveillance. This technique can be used to match the corresponding points among different camera views. Hu et al. [1] proposed a principal axis-based correspondence checking among multiple cameras. For the same human subject detected by different cameras, the correspondences are matched based on homography mapping. However, people tracking in each view is still based on Kalman filtering. Under the circumstances, the unreliable motion prediction process would degrade the performance of a developed system. Fleuret et al. [2] proposed to use a probabilistic occupancy map which is built by fusing the extracted data from multiple cameras to perform homography mapping. For each decided position, the average human height and width (a rectangle) are given. This rectangle is used to represent a person's foreground area. Hence, a person who is much shorter (a child) than the average height would still be assigned with the default size. This kind of inflexible design is inappropriate to the occlusion case. Khan and Shah [3] proposed a multiple occluding people tracking method by localizing on multiple scene planes. A planar homography occupancy constraint and the foreground likelihood information extracted from different views are combined to tackle the occlusion problem. Nevertheless, fusion of information from different views and multiple planes (10-20 planes) would be very time consuming and it is not tolerable for a real-time surveillance system.

In this paper, we propose a new multiple-camera people tracking system providing the following functions: (1) can deal with objects with occlusions for a long-term, complete occlusions from other objects, and objects with faster motions; (2) can detect foreground objects with arbitrary sizes; (3) can efficiently detect objects. Our method has two main contributions: 1) An occlusion detection function based on an M-to-one relationship with only point homography matching can achieve high computational efficiency; 2) An object motion

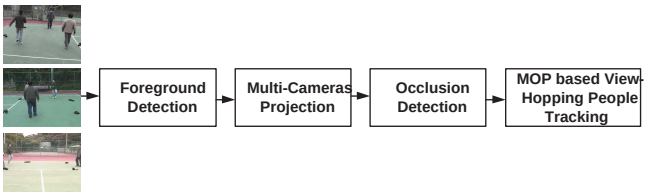

Fig. 1. Proposed People Tracking System Architecture.

probability (OMP) based metric is proposed to automatically select an appropriate observation view in our view-hopping mechanism. The rest of the paper is organized as follows. The proposed system architecture is highlighted in Sec. II. Next, the proposed techniques are described in Sec. III. Subsequently, the experimental results are demonstrated in Sec. IV. Finally, conclusions are drawn in Sec. V.

II. Proposed System Architecture

Fig. 1 shows the proposed system architecture. The left-most part is the input of the people tracking system. In our implementation, we used three video camcoders to capture video data. We tried to synchronize the three input camcorders and then analyzed the videos frame by frame. The first step of our proposed people tracking system is foreground detection. We used a simplified Gaussian Mixture Model (GMM) [16] ($K = 1$) for background modeling. This model can achieve more effective background reconstruction results than adaptive GMM. Next, we used the foreground objects detected in one view to match the corresponding objects in other views. The homography technique [14], [15] was adopted to calculate the correspondences among different views. For the occlusion problem, we propose a multiple-points-to-one-region (M-to-one) relation to deal with it. When an occlusion event is detected, our system will respond with a hopping action. That is, to hop from an occluded view to other (non-occluded) views. A strategy based on object motion probability (OMP) is proposed to select an appropriate view to hop. The details about how view-hopping is implemented will be discussed in the next section.

III. Proposed People Tracking System

A. Foreground Detection

GMM [1], [3] has been extensively applied to perform background modeling and foreground detection in the past few years. However, for real world applications, a GMM may not be suitable for real-time extraction of the foreground objects due to its costly re-computation on the GMM distributions. In a multi-camera tracking system, a near real-time requirement is necessary. Most of the time, the system should notify the administrator the runaway direction of a suspicious human subject in seconds. As a result, a simplified GMM-based background modeling scheme is proposed in this work.

Let I_t be an image frame acquired from one of the multiple cameras at time t, and k be one half length of the search window (previous k frames and subsequent k frames). The frame difference I_t^d at time t can be calculated as:

$$I_t^d = \left| I_t - \frac{1}{2k}(\Sigma_{i=t-k}^{t-1} I_i + \Sigma_{i=t+1}^{t+k} I_i) \right|, \quad (1)$$

where i is the frame index. In other words, based on Eq. (1), the difference image I_t^d can be generated according to the difference computed from the current frame to the mean of all $2k$ frames in the search window at time t.

B. Multi-camera Projection

In our proposed system, the homography [14], [15] technique plays the role of matching correspondence between different views. For a detected human subject, the positions of his/her feet represent where his/her location is in the scene. The correspondence of a same human subject in different views can be calculated by a homography transformation. A 3×3 homography matrix can be expressed as follows:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}. \quad (2)$$

Let the detected foot point in a view be $f_p = [f_x, f_y]$, and its corresponding point in another view be $f'_p = [f'_x, f'_y]$. The corresponding foot point can be calculated from f_p and H as:

$$[(f'_p)^T; 1] = H[(f_p)^T; 1]. \quad (3)$$

However, the ground plane and corresponding points of landmarks should be provided by user at the initial state.

C. Occlusion Detection: Multiple Points to One Region Relationship (M-to-One)

When a human subject is detected in the field of view of a surveillance camera, his/her foot touching the ground should be at the bottom of the line segment that links the head and that foot. This is because we assume a human subject should maintain his/her body vertical when walking. On the other hand, it is reasonable to assume the center of a walking human subject is the intersection of the above mentioned vertical line segment and the line segment linking the two hands of the human subject. The upper right part of Fig. 2 shows how human subjects are detected by our method. The regions bounded by blue boxes are the detected human subjects. The red, blue, and green squares at the bottom indicate the IDs of different people. From the detection results shown in the upper right of Fig. 2 , it is obvious that the detected foot locations are quite close to the real locations.

Suppose I_t^f is a foreground object detected at time t. There are two descriptors used to represent a detected human subject. They are, the foot position $f_p = [f_x, f_y]$, and the object region OR, respectively. The definition of OR is as follows:

$$OR = \{ I_t^f(x, y) = I_t(x, y) : B_r > x > B_l, B_t > y > B_b. \} \quad (4)$$

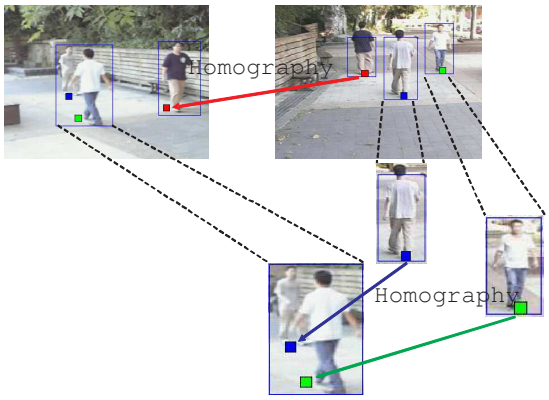

Fig. 2. Occlusion detection: upper part: no occlusion (red square person), lower part: occlusion event (green and blue persons). Red, green, and blue squares in the right view: foot points of each person. The squares in the left view: corresponding foot points from the right view to the left view according to homography.

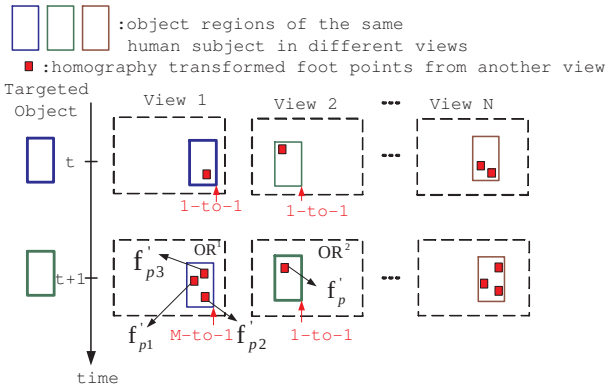

Fig. 3. View-hopping example: The targeted object is hopped from $View$ 1 to $View$ 2 due to an occlusion event.

Here B_r and B_l represent the right and left bounds in the x-direction, and B_t and B_b are the top and bottom bounds in the y-direction. The rectangular boxes formed by these bounds are shown at the bottom-right of Fig. 2. The foot position descriptor and the object region descriptor can work together to easily identify an occlusion event. Unlike conventional occlusion detection based on single camera, we propose to detect an occlusion event by fusing the information grabbed from different views. In our approach, if an occlusion event happened, the foot position descriptor will detect more than one foot point falling into the same object region. The upper-left part of Fig. 2 indicates an occlusion event is happening because the blue and the green squares that belong to two different human subjects fall into a same object region. However, from the view observed by another camera (upper-right of Fig. 2), the two corresponding human subjects do not occlude each other.

In what follows, we shall describe how to use the homography transform to judge whether an occlusion event is happening or not. Let f_{p1} and f_{p2} be two foot points detected by one camera. Their corresponding points viewed by another camera can be computed by the homography transform, H, as follows:

$$[(f'_{p1})^T;1] = H[(f_{p1})^T;1], \text{ and } [(f'_{p2})^T;1] = H[(f_{p2})^T;1]. \quad (5)$$

If both f'_{p1} and f'_{p2} fall into the same region, OR, i.e.,

$$f'_{p1} \in OR, \text{ and } f'_{p2} \in OR, \quad (6)$$

an occlusion event can be detected. It is reasonable to extend the relation from the case of two-points-to-one-region (2-to-one) to that of multiple-points-to-one-region (M-to-one). For the case of M points falling into the same region simultaneously, i.e.,

$$f'_{p1} \in OR, \ f'_{p2} \in OR, \cdots f'_{pM} \in OR, \quad (7)$$

an occlusion event can also be verified by checking the region OR. Therefore, the M-to-one relation can be utilized to detect an occlusion event by fusing the information grabbed from many cameras.

D. View Hopping-based People Tracking

Suppose there are N cameras simultaneously and separately mounted in a video surveillance system to monitor a same scene. In the scene, it is allowed to have multiple human subjects moving or standing still in it. For a targeted human subject, the corresponding object regions in different views are defined as: $\{OR^1, OR^2, \cdots, OR^N\}$, as shown by the solid rectangles with different colors in the most upper part of Fig. 3. The rectangles with dashed lines in the same rows represent the video frames captured from different views at the same time. The small red squares bounded in object regions are the homograpy transformed foot points which are transformed from other views. These red spots are used to determine whether there is an occlusion event occurring. From the system administrator's point of view, at a certain time instant, he/she can only focus on several of the views from the N cameras. Therefore, we propose a view-hopping strategy to automatically select an appropriate view for the administrator to monitor. Using Fig. 3 as an example, assume at time t, there is no occlusion event detected in both $View$ 1 and $View$ 2. In other words, the one-to-one relations are detected in both of these views. Therefore, we can randomly select one of the object regions for the administrator to monitor. Suppose $View$ 1 is randomly chosen. The object region in $View$ 1 is shown at the left of the upper row in Fig. 3. However, when the time proceeds to $t+1$, there is an occlusion event occurred in $View$ 1, because there are three foot points detected. However, $View$ 2 still has only one foot point detected in the object region (second row, $View$ 2 in Fig. 3). That means the system has to excute an automatic view hopping from $View$ 1 to $View$ 2 to avoid the occlusion case. Therefore, the targeted object TO_{t+1} has to be hopped to $View$ 2, i.e.,

$$TO_{t+1} = \left\{OR^2 : f'_{p1}, f'_{p2}, f'_{p3} \in OR^1 \ ; \ \exists! f'_p \in OR^2\right\}, \quad (8)$$

where f'_{p1}, f'_{p2}, and f'_{p3} are the homography transformed foot points derived from another view, and the object region OR^2 only contains one homography transformed foot point f'_p. When Eq. (8) holds, a view-hopping action is triggered.

It is reasonable to extend the relation from the case of three points to M points. By integrating Eq. (7) and Eq. (8), we can derive:

$$TO_t = \left\{ OR^v : f'_{p1}, f'_{p2} \cdots f'_{pM} \in OR^u \ ; \exists! f'_p \in OR^v \right\}, \tag{9}$$

where u and v are the view indices among N cameras. Eq. (9) represents when an occlusion event occurred in the u-th view but there is no occlusion identified in the v-th view, the observation view should be hopped to the v-th view.

E. Object Motion Probability (OMP) for View Hopping

In a multi-camera environment, a human subject may be occluded by other people in the view of one camera, but he/she may not be occluded in other camera views. Because a front view contains the most information of a human subject, we hope our developed system can automatically hop to that view. In general if a person walks from far to near in the field of view of a camera, its corresponding y-axis component in a 2-D image plane should be from top to bottom, as shown by the red arrow in Fig. 4 (a). In other words, the object motion [17] in y-direction can be used to judge whether a human subject is approaching (can see his/her front) or leaving (can see his/her back) the camera. Therefore, we shall make use of the object motion to judge whether a walking human subject is in front view or not.

As shown in Fig. 4 (a) and Fig. 4 (b), the left human subject is walking in a constant speed. The object motion in the far location (red segment in Fig. 4 (a)) is much smaller than that in the near location (red segment in Fig. 4 (b)). Since the movement of a human subject in the distance may result in a smaller object motion in comparison with an object motion happens nearby, the object motion has to be normalized based on its distance to the viewer. Therefore, we have

$$NOM_{y_t} = \frac{f_{p_t}(f_y) - f_{p_{t-1}}(f_y)}{max\left(f_{p_t}(f_y), f_{p_{t-1}}(f_y)\right)}, \tag{10}$$

where NOM_{y_t} is the normalized object motion in y-direction detected at time t, and $f_{p_t}(f_y)$ is the foot point position in y-direction at time t. For the convenience of representation, we use NOM as the abbreviation of NOM_{y_t}. Fig. 5 shows the relationship of the items in Eq. (10). The black arrow shows the original object motion. The y direction projection is illustrated as the blue arrow (the numerator in Eq. (10)). The denominator in Eq. (10) is represented by the unit object motion, as shown by the red segment in Fig. 5. As a result, the NOM can be obtained as shown by the red arrow in Fig. 5. Note that the NOM is calculated for the corresponding human subject in the same view. Since the computation of foot points can be easily affected by noises, we only consider those significant object motions. That is, only for those object motions that have larger magnitudes can be accepted as valid object motions.

Fig. 4. The movement of a human subject (left one) from a far location to a near location: (a) walking at distance, (b) walking at a near location.

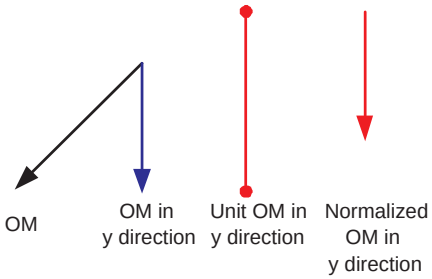

Fig. 5. The relationship among OM, OM in y-direction, unit OM in y direction, and NOM in y direction.

Motivated by the concept of probability updating for appearance model [18], the approaching/leaving probability of object motion for a human subject can be updated by checking the corresponding object motion, i.e.,

$$OMP_{a_t}(NOM_{y_t}) = \begin{cases} OMP_{a_{t-1}} \cdot \lambda + (1-\lambda), & \text{if } NOM_{y_t} \geq Th_{NOM}; \\ OMP_{a_{t-1}} \cdot \lambda, & \text{otherwise}, \end{cases} \tag{11}$$

and

$$OMP_{l_t}(NOM_{y_t}) = \begin{cases} OMP_{l_{t-1}} \cdot \lambda + (1-\lambda), & \text{if } NOM_{y_t} < -Th_{NOM}; \\ OMP_{l_{t-1}} \cdot \lambda, & \text{otherwise}, \end{cases} \tag{12}$$

where NOM_{y_t} is the y-direction object motion of a corresponding human subject at time t, λ is an update factor, set to 0.95 [18], and Th_{NOM} is a threshold for a valid normalized object motion. For the convenience of representation, we use **approaching OMP** to represent OMP_{a_t}, and **leaving OMP** to represent OMP_{l_t}. In our experiments, the **approaching/leaving OMPs** for a human subject are initially set to 0.5 because he/she has equal probability of approaching or leaving a camera without any prior knowledge.

An example of updating **approaching/leaving OMPs** is shown in Fig. 6. At first, the foreground objects can be detected from all three views. The labels with different colors represent different human subjects, as shown in Fig. 6 (a). Next, the proposed occlusion detection algorithm is used to detect the occlusion region. In this example, $View3$ in Fig. 6 (a) was

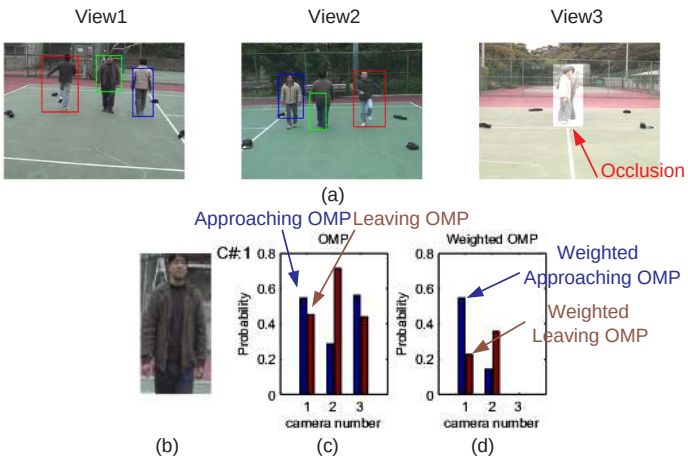

Fig. 6. **OMP** example: (a) detected foreground objects in three views (b) target object (bounded by green rectangle in (a)), (c) **OMPs** in three views, and (d) the weighted **OMPs** in three views.

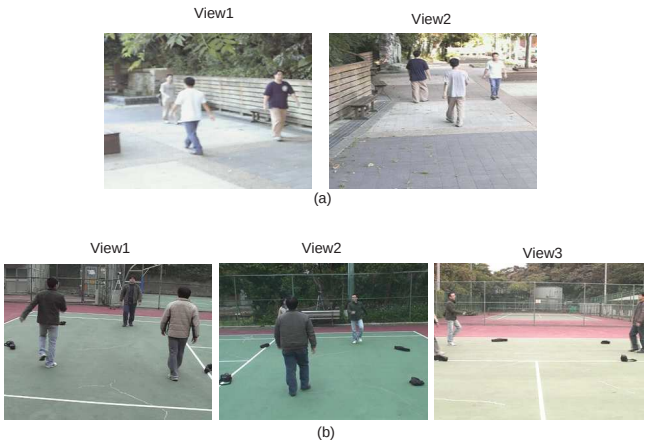

Fig. 7. Test Videos: (a) Two-camera scenario, (b) Three-camera scenario: $Tennis\ Court$.

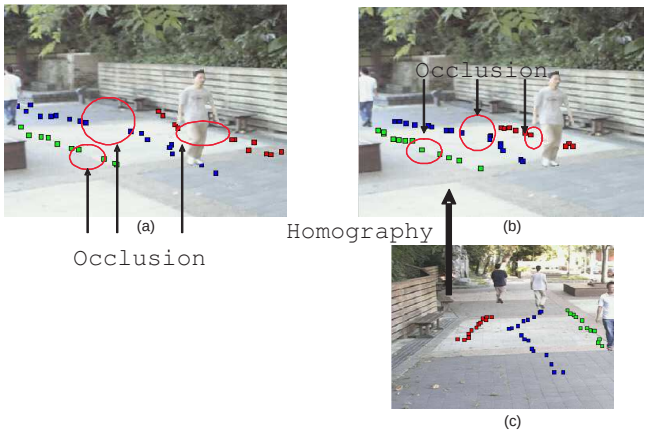

Fig. 8. Trajectories calculated from the two-camera sequence: (a) trajectories of three human subjects detected in $View1$, with occlusions in red circles, (b) trajectories of homography mapping from $View2$ to $View1$, with occlusions in red circles, and (c) trajectories of three human subjects detected in $View2$.

identified as an occlusion event. Therefore, we have to check the **OMP** status of $View1$ and $View2$. Let's use the human subject bounded by the green rectangle as an example. In $View1$, the targeted human subject was detected with a higher **approaching OMP** and a lower **leaving OMP** (the left most blue bar and brown bar, respectively). On the other hand, the **approaching OMP** in $View2$ is much smaller than the **leaving OMP**. Since the targeted human subject is occluded in $View3$, the best view to hop to is $View1$ in this case. The selection of the best view is determined by computing the weighted **OMPs** under different conditions, i.e.,

$$V_t = \arg\max_v \{W^v \cdot OMP_t^v\}, \tag{13}$$

$$W^v = \begin{cases} 1, & \textbf{approaching OMPs} \geq Th_C; \\ 0.5, & \textbf{leaving/approaching OMPs} < Th_C; \\ 0, & \text{disappeared or occluded;} \end{cases} \tag{14}$$

where W^v is the weight for the v-th view, and $OMP_t^v = \{$**approaching OMP, leaving OMP**$\}$. When the **OMPs** fit the situations described in Eq. (14), its corresponding weight W^v would be generated. Under these circumstances, the distribution probability among different views can be calculated (e.g. Fig. 6 (d)). Finally, an appropriate view can be determined by exhaustively searching the view with the maximum weighted **approaching/leaving OMPs** as expressed by Eq. (13).

IV. Experimental Results

To test the effectiveness of the proposed method, we used two scenarios to capture videos at two distinct time spots of a same day. Fig. 7 shows some snapshots of two scenarios: the two-camera scenario is adopted to verity whether the homography transformation can be appropriately used to perform occlusion detection, and the three-camera scenario is used to check whether the proposed scheme can be applied to real-world problems. Fig. 8 shows the trajectories of three human subjects walking in the two-camera surveillance system. The trajectories of $View1$ and $View2$ in Fig. 7 (a) are shown, respectively in Fig. 8 (a) and Fig. 8 (c). The detected foot points of $View2$ can be transformed via homography to $View1$, as indicated in Fig. 8 (b). By comparing the detected trajectories (Fig. 8 (a)) and the homography transformed trajectories (Fig. 8 (b)), we can verify that the results are quite close to each other, showing that the correspondence established from different views by homography transformation are quite accurate.

The tracking results of a real-world case are shown in Fig. 9. The columns represent the video frames captured from different views, and the rows represent the frames captured at different time instants. For different human subjects, we used rectangles with different colors to represent them. For example, in Fig. 9 (c), the human subjects bounded by red rectangles in three views are identical. This indicates that the applied homography transformation can help match the correspondence. On the other hand, $View1$ in Fig. 9 (c), $View2$

Fig. 9. Video test sequence in $Tennis\ Court$, the frames in the first, second, and third columns are from $View1$, $View2$, and $View3$, respectively. The rows show different time instants at (a) frame number 3050 (b) frame number 3150, and (c) frame number 3250.

in Fig. 9 (a), and $View3$ in Fig. 9 (b) all contained detected occlusion events. This outcome shows that our proposed M-to-one occlusion detection mechanism could successfully detect the occlusion regions in frames. As to the best view selection issue, the results are shown in Figs. 10-11. In these figures, the targeted human subject is shown in the left of each row. In Fig. 10, the targeted human subject are viewed from camera 2. This is because the weighted **approaching OMPs** were the largest at all three time instants. In Fig. 11, the targeted human subject are viewed from camera 1 at the first two instants because the weighted **approaching OMPs** were bigger than that of other views. However, the view was forced to hop to camera 2 due to occlusions. In Fig. 11 (c), only camera 2 was associated with the values of weighted **OMPs**. Camera 1 and Camera 3 did not have any weighted **OMPs** value, it means the targeted human subject may be occluded or may be outside the range of that camera. Under these circumstances, we are forced to hop to Camera 2, though it is only a back view.

V. Conclusions

In this paper, a novel multiple-camera people tracking system is proposed to supply the following functions: (1) long-term occlusions, complete occlusions, and unpredictable motions could be handled; (2) an object could be detected according to its corresponding foreground size, avoiding the unreasonable size given problem; (3) objects could be effectively detected. The main contribution of our method was twofold: 1) An M-to-one relationship matched by point homography for occlusion detection could achieve high efficiency; 2) An object motion probability (OMP) based view-hopping technique was proposed to automatically select an appropriate observation view for people tracking. However, the limitation of our system is that when a tracked human subject is occluded in all views, our system cannot identify the occlusion events, furthermore, the view-hopping result could be falsely given.

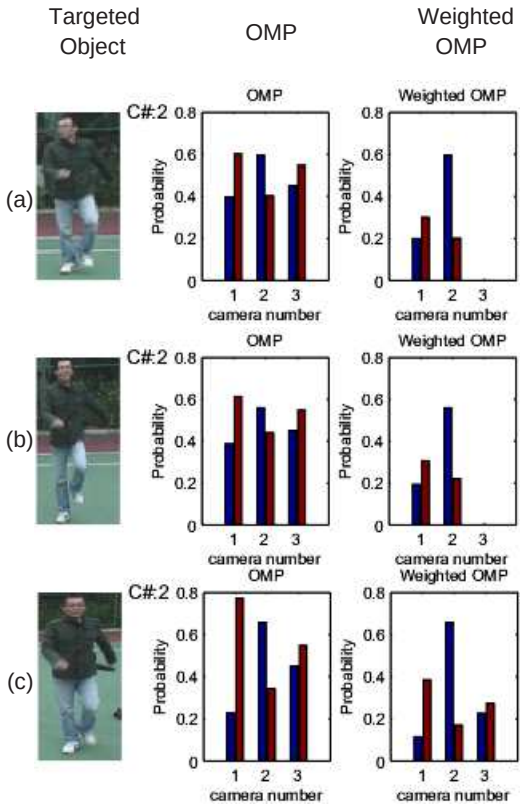

Fig. 10. Experiment # 1 of best view selection. From left to right, the targeted human subject, **OMPs** (blue bars: **approaching OMPs**; brown bars: **leaving OMPs**), and weighted **OMPs**. From top to bottom, frame grabbed at different time instants: (a) frame number 3050 (b) frame number 3150, and (c) frame number 3250.

References

[1] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank, "Principal Axis-Based Correspondence between Multiple Cameras for People Tracking," *IEEE Trans. Patt. Analy. Mach. Intell.*, Vol. 28, No. 4, pp. 663-671, Apr. 2006.

[2] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera People Tracking with a Probablistic Occupancy Map," *IEEE Trans. Patt. Analy. Mach. Intell.*, Vol. 30, No. 2, pp. 267-282, Feb. 2008.

[3] S.M. Khan and M. Shah, "Tracking Multiple Occluding People by Localizing on Multiple Scene Planes," *IEEE Trans. Patt. Analy. Mach. Intell.*, Vol. 31, No. 3, pp. 505-519, Mar. 2009.

[4] I. Haritaoglu, D. Harwood, and L. Davis, "Who, When, Where, What: A Real-Time System for Detecting and Tracking People," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 222-227, 1998.

[5] R.T. Collins, "Mean-Shift Blob Tracking through Scale Space," *Proc. Conf. IEEE Computer Vision and Pattern Recognition*, Vol. 2, pp. 234-240, 2003.

[6] M. Han, W. Xu, H. Tao, and Y. Gong, "An Algorithm for Multiple Object Trajectory Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Vol. 1, pp. 864-871, June 2004.

[7] D. Comaniciu, V. Ramesh, and P. Meer "Real-Time Tracking of Non-Rigid Objects Using Mean Shift," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Vol.2, pp. 142-149, 2000.

[8] S. Khan and M. Shah, "Tracking People in Presence of Occlusion," *Proc. Asian Conf. Computer Vision* , 2000.

[9] Q. Cai and J. Aggarwal, "Automatic tracking of human motion in in door scenes across multiple synchronized video streams," *IEEE Intl. Conf. Computer Vision*, pp. 356-362, 1998.

[10] I. Mikic, S. Santini, and R. Jain, "Video Processing and Integration from Multiple Cameras," *Proc. Image Understanding Workshop*, 1998.

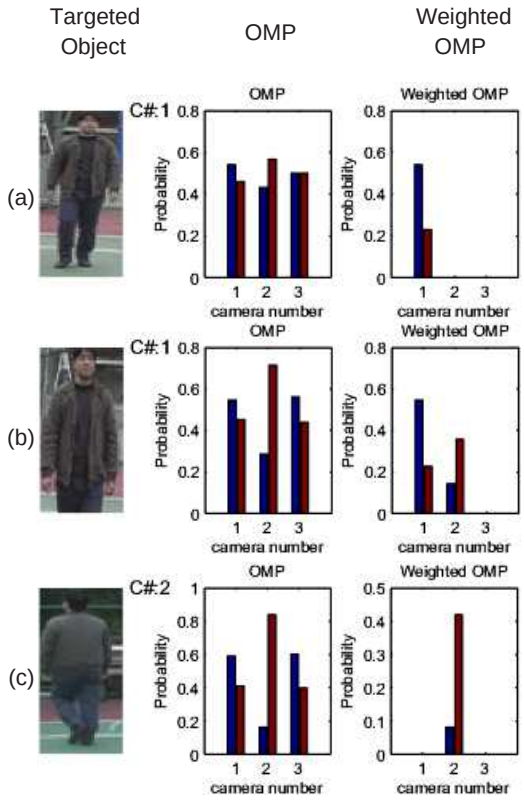

Fig. 11. Experiment # 2 of best view selection. From left to right, the targeted human subject, **OMPs** (blue bars: **approaching OMPs**; brown bars: **leaving OMPs**), and weighted **OMPs**. From top to bottom, frame grabbed at different time instants: (a) frame number 3050 (b) frame number 3150, and (c) frame number 3250.

[11] J. Black, T. Ellis, and P. Rosin, "Multi-view Image Surveillance and Tracking," *Proc. IEEE Workshop on Motion and Video Computing,* pp. 169-174, 2002.

[12] J. Giebel, D. Gavrila, and C. Schnorr, "A Baysian Framework for Multi-cue 3D Object Tracking," *Lecture Notes in Computer Science (Proc. European Conf. Computer Vision),*, Vol. 3024, pp. 241-252, 2004.

[13] K. Smith, d. Gatica-Perez, and J.-M Odobez, "Using Particles to Track Varying Numbers of Interacting People," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Vol. 1, pp. 962-969, 2005.

[14] K.J. Bradshow, L.D. Reid, and D.W. Murray, "The Active Recovery of 3D Motion Trajectories and Tieir Use in Prediction," *IEEE Trans. Patt. Analy. Mach. Intell.*, Vol. 19, No. 3, pp. 219-234, Mar. 1997.

[15] L. Lee, R. Romano, and G. Stein, "Monitoring Activities from Multiple Video Streams: Estabilishing a Common Coordinate Frame," *IEEE Trans. Patt. Analy. Mach. Intell.*, Vol. 22, No. 8, pp. 758-767, Aug. 2000.

[16] C. Stauffer and W. Grimson, "Learinig Patterns of Activity Using Real Time Tracking," *IEEE Trans. Patt. Analy. Mach. Intell.*, Vol. 22, No. 8, pp. 747-767, 2000.

[17] 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 (Applications and Reviews),* Vol. 34, No. 3, pp. 334-352, 2004.

[18] A. Senior, "Tracking People with Probabilistic Appearance Models," *Proc. IEEE Performance Evaluation of Tracking and Surveillance,* pp. 48-55, 2002.