Real-time Multiple Objects Tracking with Occlusion Handling in Dynamic Scenes

Tao Yang[1], Stan Z.Li[2], Quan Pan[1], Jing Li[1]
[1]*College of Automatic Control, Northwestern Polytechnical University, Xi'an, China , 710072*
[2]*National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, 100080*
yangtaonwpu@163.com , szli@nlpr.ia.ac.cn , quanpan@nwpu.edu.cn , jinglinwpu@163.com

Abstract

This work presents a real-time system for multiple objects tracking in dynamic scenes. A unique characteristic of the system is its ability to cope with long-duration and complete occlusion without a prior knowledge about the shape or motion of objects. The system produces good segment and tracking results at a frame rate of 15-20 fps for image size of 320x240, as demonstrated by extensive experiments performed using video sequences under different conditions indoor and outdoor with long-duration and complete occlusions in changing background.

1[1]. Introduction

Object tracking is an essential component of an intelligent video surveillance system. Accurate and real-time object tracking will greatly improve the performance of object recognition, activity analysis and high-level event understanding [1,2,3,4,5].

Methods to solve the occlusion problem in multiple interacting objects tracking have been previously presented. Shiloh [6], Chang [7] and Dockstader [8] overcame occlusion in multiple objects tracking by used fusing multiple camera inputs. Cucchiara [9] proposed probabilistic masks and appearance models to cope with frequent shape changes and large occlusions. Eng [10] developed a Bayesian segmentation approach that fused a region-based background subtraction and a human shape model for people tracking under occlusion. Wu [11] proposed a dynamic Bayesian network which accommodates an extra hidden process for partial occlusion handling. Andrew [12,13], Siebel [14], Hieu [15] and Alper [16] used appearance models to track occluded objects. Tao [17] presented a dynamic background layer model and model each moving object as a foreground layer, together with the foreground ordering, the complete information necessary for reliably tracking objects through occlusion is included.

Although many algorithms have been proposed in the literature, the problem of multiple interacting objects tracking in complex scene is still far from being completely solved. Multiple camera based tracking methods [6,7,8] cannot handle complete occlusion. Precise model based algorithms [12,13] are sensitive to background clutter, and they are at the cost of computationally more expensive schemes because model estimation for the number of model parameters is usually large. Moreover, many of those algorithms are designed to deal with short-duration partial occlusion, and fail at severe occlusions and when a partial occlusion lasts for a long time. Probabilistic approaches like Monte Carlo filter is useful in dealing with the problem of background clutter as it allows for the tracking of multiple hypotheses [18,19,20]. However, the measure of object has to be detected by an independent technique that may not be acquired in heavy occlusion. Several methods using motion model to perform robust tracking can deal with some instances of occlusion. These methods require precise motion modeling [21] and fail at the non-linear motion of interacting objects.

To deal with multiple objects tracking in dynamic scenes, we separate the object state into three parts: Before, during and after occlusion. Considering occlusion often cased by touching objects, we suppose that during the occlusion, the trajectory of each individual object is similar to the entire group, fortunately, this is a valid assumption in real surveillance scene. If we could keep on tracking and labeling each individual object correctly before and after the occlusion, and tracking the entire group during the occlusion, the integrate trajectory of each object will be recovered. In this paper, we present a real-time system for multiple objects tracking in complex real world. The system consists of two parts (shown in Figure 1): (1) object segmentation, and (2) merging and splitting detection, and feature correspondence. In part one, a fast algorithm is presented for background

[1] * The work presented in this paper was sponsored by the Foundation of National Laboratory of Pattern Recognition and National Natural Science Foundation of China (60172037)

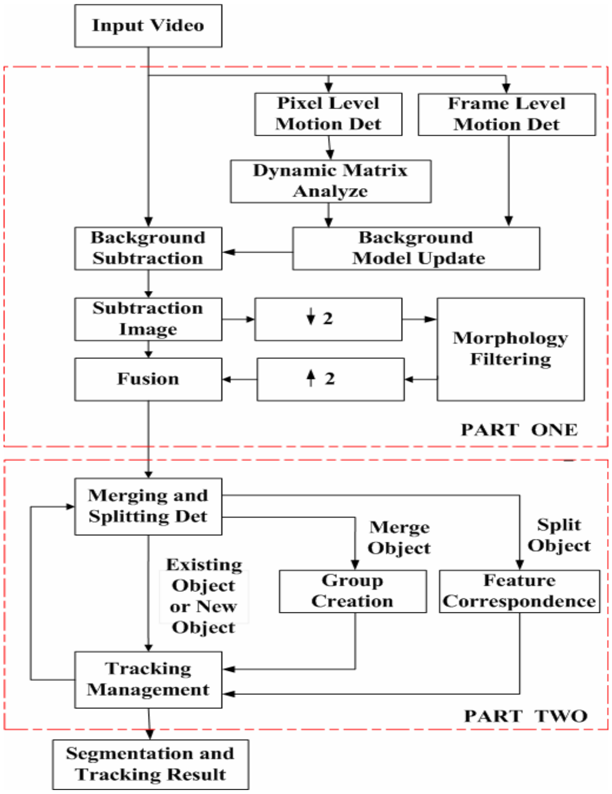


Figure 1. Tracking System Diagram. Part 1: Background Maintenance and Moving Object Segmentation. Part 2: Object Tracking and Occlusion Handling.

maintenance to handle various scene changes, including ghosts and illumination changes, running at 20 fps. The input video is used to estimate a background model based on a two level pixel motion analyze algorithm, which is then used to perform background subtraction image. After connected area analysis, small blobs will be removed and the resulting foreground regions will be saved. To reduce the large scale noises caused by background clutter, the tracking management module of the second part associates foreground regions in consecutive frames to construct hypothesized tracks, only those blobs which have been correctly corresponded for several frames will be considered as a valid target.

In part two, a combination mechanism is embedded to detect merging and splitting events, using object tracking and segmentation result. In the merging and splitting detection module, the detected object is divided into four classes: existing object, new object, merge object and split object. The first two class objects will be directly used to update the tracker in the tracking management module. For the merge object, a group will be created which contains the trajectory and color feature of the objects in it. For the split object, the feature correspondence module are employed to assign a correct label to each split object based on Kullback-Leibler (KL) distance. In the following, we explain details of the system.

The paper is structured as follows: Section 2 presents the moving object segmentation algorithm. Section 3 explains the merging and splitting detection method. Section 4 discusses feature correspondence method. Section 5 presents extensive results.

2. Moving Object Segmentation

We adopt the background subtraction approach. However, rather than based on mixture Gaussian models [5,22] for background modeling or relying upon the distribution of the pixel value, we present a two level (pixel level and frame level) background maintenance algorithm for real-time segmentation and background updating. This is to avoid problems (high computational costs and slow adaptation to a new background model) associated with mixture Gaussian background modeling.

The basic idea of the pixel level background updating is based on an assumption that the pixel value in the moving object's position changes faster than those in the real background. Fortunately, this is a valid assumption in most application fields. Under this assumption, we can distinguish the foreground and background accurately by a simple frame-to-frame difference method, which could detect the fast changes of pixel. However, this method will fail when the inside color of object is uniform. In this situation, pixel values do not vary within the object. To deal with this problem, we present a dynamic matrix $D(k)$ to analyzing the changes detection result of the frame-to-frame difference method, where the motion state of each pixel is stored in the matrix. Only those pixels whose values do not change much can be updated into the background.

Let $I(k)$ denote the input frame at time k, and the subscript i,j of $I_{i,j}(k)$ represent the pixel position. The frame-to-frame difference image $F(k)$ and the dynamic matrix $D(k)$ at time k are defined as follows:

$$F_{i,j}(k) = \begin{cases} 0 & \left|I_{i,j}(k) - I_{i,j}(k-\gamma)\right| \le Tf \\ 1 & otherwise \end{cases} \qquad (1)$$

$$D_{i,j}(k) = \begin{cases} D_{i,j}(k-1)-1 & F_{i,j}(t)=0, D_{i,j}(k-1)\neq 0 \\ \lambda & F_{i,j}(t)\neq 0 \end{cases} \qquad (2)$$

where γ represent the interval time between the current frame and the old one, Tf is the threshold to make a decision whether the pixel is changing at time k or not, and λ is the time threshold of the pixel's stable times in consecutive frames. Once $D_{i,j}(k)$ equates to zero, the pixel will be updated into the background with a linear model (3):

$$B_{i,j}(k) = \alpha \cdot I_{i,j}(k) + (1-\alpha)\cdot B_{i,j}(k-1) \qquad (3)$$

where $B(k)$ is the background image at time k and α is the weight of input frame.

Although the pixel level background update method could deal with many serious problems mentioned above, it still has a drawback in that it only considers each individual pixel while ignoring the motion information contained in the frame.

The frame level updating is used to solve this problem. The mechanism utilizes the moving character of the whole image v (4) to achieve fast background update under the situation of the abrupt scene changing such as camera shaking, illumination changing and new left object in the scene.

$$v = \frac{\sum_{j=1}^{n}\sum_{i=1}^{m} F_{i,j}(k)}{m \times n} \quad (4)$$

where m, n represent the width and height of the image. Once v is less than a threshold, we will make a decision that no moving object in the current image and update all the stable pixels in the current frame to the background immediately using (3).

By fusing the detection result at both pixel and frame levels, the background update procedure maintains a suitable background model under different conditions. In background subtraction step, each video frame is compared with the reference background model, pixels in the current frame that deviate significantly from the background will be detected. After that a size filter is used to remove small components and the moving object positions will be gained and transformed to the original subtraction image to get the accurate final segmentation results. Considering the background clutter and the similarity of the foreground region and the background, noises blobs with large size are still exist after the morphology filtering. To solve this problem, we analyze the correct correspondence times of each blob in consecutive frames, thus uncertain noises will be removed.

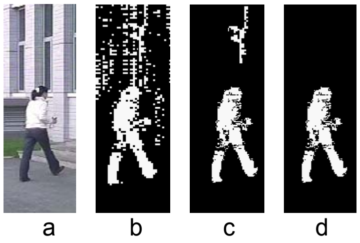


Figure 2. An example of object segmentation. a) Input video. b) Subtraction Image. c) Morphology filtering result. d) Final result fused with spatio-temporal information.

3. Merging and Splitting Detection

This module includes two main steps: (1) correspondence between foreground region and track, and (2) merging and splitting detection.

As in most of the tracking approaches, the correspondence process attempts to associate the foreground regions with one of the existing tracks. Let $T(k) = \{T_1(k), T_2(k), ..., T_m(k)\}$ denote the existing tracks and $M(k) = \{M_1(k), M_2(k), ..., M_n(k)\}$ denote the foreground region measures at time k. This process starts with the construction of a distance matrix D_E^k between the active tracks $T(k)$ and the each of the foreground region measure $M(k)$. The distance matrix D_E^k (rows correspond to existing tracks and columns to foreground regions in the current frame) is based on the Euclidean distance (5).

$$\begin{aligned} D_E^k(i,j) &= \left\| T_i(k) - M_j(k) \right\| \\ &= \sqrt{(T_{ix}(k) - M_{jx}(k))^2 + (T_{iy}(k) - M_{jy}(k))^2} \end{aligned} \quad (5)$$

where $T_{ix}, T_{iy}, M_{jx}, M_{jy}$ represent the center positions of the bounding box of T_i and M_j , $i = 1, \ldots, m$, $j = 1, \ldots, n$.

Considering the similarity between the tracker and measure, if their distance is larger than a threshold, they will not be associated and the relative element in matrix D_E^k will be set to infinitude. Based on analyzing the matrix D_E^k , a correspondence matrix C_E^k at time k is constructed to assign the foreground region measure to the track. The following is the details of construction.

1. Firstly, all the elements of matrix C_E^k are set to zero.
2. Find the position of the minimal elements in every row $\alpha = \{\alpha_1, \ldots, \alpha_m\}$ and column $\beta = \{\beta_1, \ldots, \beta_n\}$ of D_E^k through the following equations:

$$\begin{cases} D_E^k(i, \alpha_i) = \min(D_E^k(i,j)), j = 1, \cdots, n \\ D_E^k(\beta_j, j) = \min(D_E^k(i,j)), i = 1, \cdots, m \end{cases} \quad (6)$$

3. Finally, add one to the correspond element in matrix C_E^k .

$$C_E^k(i, \alpha_i) = C_E^k(i, \alpha_i) + 1, i = 1, ..., m \quad (7)$$

$$C_E^k(\beta_j, j) = C_E^k(\beta_j, j) + 1, j = 1, ..., n \quad (8)$$

Three possible values may found in the element of matrix C_E^k : Zero, one and two. Zero means no selection. One represents one selection happens. Two means the track and the measure selects each other both. Five possible results can arise in the matrix C_E^k :

- A track is not associated to any measure (All the elements in a row are zero).
- A measure is not associated to any track (All the elements in a column are zero).
- A track is associated to more than one measure (More than one element in a row are larger than zero).
- A measure is associated to more than one track (More than one element in a row are larger than zero).
- A measure is associated to a track (The element value is two).

In this paper, if an element value in matrix C_E^k equals to two, the measure will assign to the track, and all the elements in the same row and column of the distance matrix D_E^k are updated to infinitude. After that, a new correspond matrix C_E^k is constructed from the updated distance matrix D_E^k. This process will keep on looping until none of the elements value of matrix C_E^k equals to two. Finally, the foreground measures and existing tracks are classified into three parts: Non-matched track, non-matched measure, matched track and measure.

The above association method assigns one measure to one track and can not handle merging and splitting event, in which one measure may assign to multiple tracks and one track may assign to multiple measures. To solve this problem, we develop a merging and splitting detection procedure based on the obtained classification results.

For those non-matched track, a merging detection algorithm is used to decide whether the track is merged by another measure or is missed. If a merging happens, a new group is generated. If the track is missed, the confidence of the track will be decreased, once it drops below a specific threshold, the track will be deleted. For those non-matched measures, a splitting detection module is developed to decide whether the measure is split from an active track or it is a new target. When a splitting event is confirmed, a feature correspondence module (see section 4) is performed to labeling each object correctly.

Merging might occur due to a non-matched track overlapped with a measure. This judgment is based on the assumption that there must be overlapped area between the initial merging bounding box and the merged object (Figure 3, first row).This is a valid assumption when the segmentation process is fast enough, as soon as object touches with each other at time k+1, a large bounding box contains all the merged objects will be created and it has large overlapping areas with the merged objects at time k. Fortunately the moving object segmentation method mentioned in section 2 achieves 20fps in the surveillance system, fast enough to detect merging event even with high-speed objects. Similar to the merging method above, splitting is detected due to a non-matched measure overlapped with a track (Figure 3, second row). When a group splits, each split object will be labeled correctly with a feature correspondence method (See section 4).

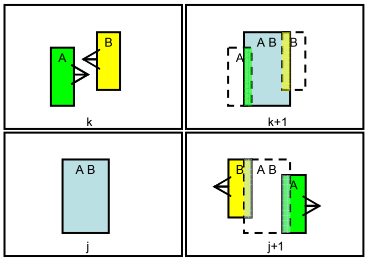


Figure 3. A scenario of blob merging and splitting detection . The first row contains the blob merging events and the overlapping areas. The second row contains the splitting events.

4. Feature Correspondence

When a merging event has been detected, the information of the occluded track will be added into the group, for instance its trajectory and certain feature. After that, the entire group will be tracked as one target. When it splits, the feature information of the occluded object in the group will be used for correspondence.

An important point is how to select the suitable feature. To reduce the complexity of the tracking system, we use 2D feature of the object. During the last two decades, three classes of features have been widely considered for tracking purpose: motion, appearance and color. The motion feature based methods smooth the position and motion of the object only and the object has to be detected by an independent technique. Once occlusion happens, the measure of the filter cannot be acquired and the confidence of tracking result is decided by the occluded object's motion character and the maximal duration of the occlusion, and it will be decreased according to object's non-linear motion and long time occlusion. Appearance model has got much attention recently [10,11,13] and it is powerful in dealing with short time partial occlusion. However the object appearance can change a lot after long time occlusion, moreover, it can not handle complete occlusion which is quite often in complex real world surveillance. Considering the complex real world surveillance scene, in our system, we use color feature to measure similarity. Let $O^q(k)=\{O_1^q(k),O_2^q(k),...,O_u^q(k)\}$ denote the occluded objects of group q and $S^q(k)=\{S_1^q(k),S_2^q(k),...,S_v^q(k)\}$ denote the split objects from the group q at time k. PO_i^q denote the color distribution of the *ith* occluded object $O_i^q(k)$ and

PS_j^q represents the color distribution of the jth split object $S_j^q(k)$ of group q. Since in the experiments, we achieved the same qualitative correspondence results with RGB and HSV color space, we chose the RGB space and computing the color distribution with $Nr \cdot Ng \cdot Nb$ bins. In addition, instead of computing the color histogram of the bounding box, we use a strategy of combing motion segmentation to provide more accurate color feature. The color distance matrix $D_{C_q}^k$ (rows correspond to occluded objects and columns to split foreground region) is measured using the Kullback-Leibler (KL) distance (9) between the two color distributions and use the association algorithm mentioned in section 3 to assign each split object.

$$D_{C_q}^k(i,j) = \sum_{l=1}^{Nr\cdot Ng\cdot Nb} PO_i^q(l) \cdot \log(PO_i^q(l) / PS_j^q(l)) \quad (9)$$

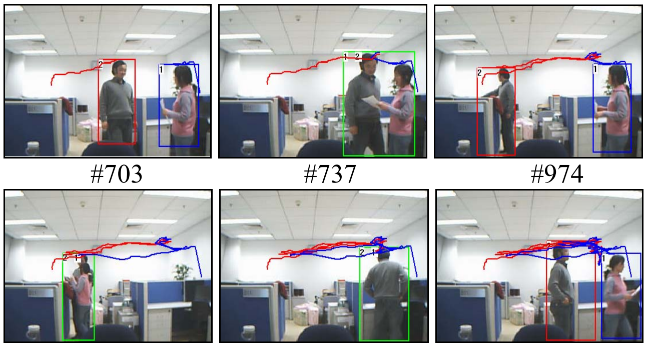
where $i = 1,\ldots,u$, $j = 1,\ldots,v$.

Considering the number of split targets is less than the total number of the occluded objects in the group, after the correspondence step, the split foreground regions and existing occluded objects are classified into two parts: Non-matched occluded object, matched object and measure. If the number of non-matched occluded object equals to zero, the group will be terminated.

5. Experimental Results

The system is implemented on standard PC hardware (Pentium IV at 3.0GHz) and works at 15-20fps. The video image size is 320x240 (24 bits per pixel) captured by Sony DCR9E at 25fps. The system is tested in typical indoor and outdoor environments for handling ghost situation, background modeling and occlusion. In the system we use color histogram in RGB color space with 10x10x10 bins for feature correspondence. We deliberately selected clips taken under difficult conditions. The following presents the results.

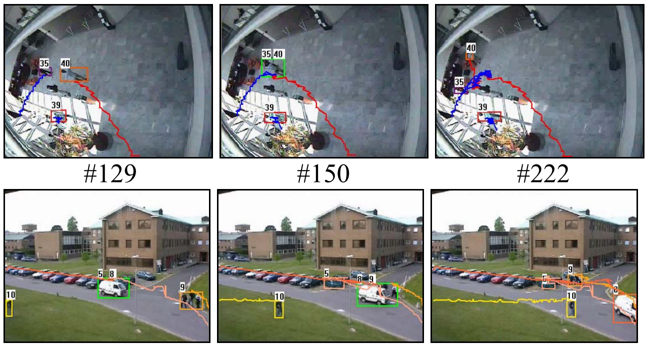
Figure 4 shows an example of tracking two interacting persons in an indoor environment. The red and blue box shows the position of the person without occlusion. Green box shows the position of the group in which people are occluded with each other. Their trajectories are shown with red and blue lines. In this sequence, the target No.2 (Target ID is labeling at the top left corner of the bounding box) changes his motion direction suddenly (Figure 4, frame #974, red line), the motion model based tracking approach is always failed at this situation. At frame #1176, occlusion happens again and after that the two persons are tracked as a whole group until frame #1665. During this process, the target No.1 is completed occluded by target No.2 (Figure 4, frame #1562) for several frames and the occlusion lasts for 183 frames (From frame #1482 to frame #1665). It is hard for those template matching methods or appearance models to handle this situation. Once the end of occlusion has been determined (Figure 4, frame #1665), the people can be recaptured and correct labeled.



#703 #737 #974

#1176 #1562 #1665

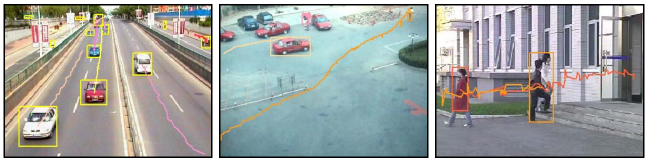
Figure 4. A sequence of two interacting persons tracking with heavy occlusion in an indoor environment . Note that long-duration complete occlusion (Frame #1562) is correctly handled.



#129 #150 #222

#996 #1078 #1208

Figure 5. Tracking results of CAVIAR and PETS 2001 sequence. The first row contains the tracking result of two people fighting of CAVIAR dataset. The second row contains the tracking result of PETS 2001 dataset.



a b c

Figure 6. Real-time multiple objects detection and tracking results in our real-time surveillance system. a) Road surveillance. b) Entrance of parking lot. c) Entrance of the building.

We report results of the system on the most recent dataset of EC Funded CAVIAR project [23] and PETS 2001 dataset in Figure 5. In the CAVIAR dataset (Figure 5, first row), two people meet, fight and run away. Heavy occlusion happens during the fighting (Figure 5, first row, frame #150) and the system correctly tracks each person before, during and after fighting. Two occlusions happen in PETS sequence (Figure 5, second row, frame #996, frame #1078) and objects are tracked correctly before and after occlusions. Images in Figure 6, selected from the real-time surveillance system, represent typical scenes in a residential area, including traffic road surveillance, entrance of parking lot and entrance of the building.

6. Conclusion

A real-time multiple objects tracking system is presented. Experiments on complex indoor and outdoor environments show that the system can deal with difficult situations such as ghosts and background changes. Moreover, it can track multiple objects with long-duration and complete occlusion. While the system is highly computationally cost effective and accurate, future work includes developing a real-time high-level events understanding system.

References

[1]A. Amer, E. Dubois, and A. Mitiche, Real-time system for high-level video representation: application to video surveillance, *in Proc. SPIE Int. Symposium on Electronic Imaging, Conf. on Visual Communication and Image Processing (VCIP)*, Santa Clara, USA, vol. 5022, pp. 530-541, Jan. 2003.

[2]Collins et al. A System for Video Surveillance and Monitoring.VSAM Final Report, *Technical report CMU-RI-TR-00-12*, Carnegie Mellon University, May, 2000.

[3]I.Haritaoglu, D.Harwood, and L.S.Davis, W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol: 22 , Issue: 8 , pp.809-830,Aug. 2000.

[4]Fengjun Lv, Jinman Kang, Ram Nevatia, Isaac Cohen, and Gérard Medioni, Automatic Tracking and Labeling of Human Activities in a Video Sequence, *Proceedings of the 6th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS04)*, Prague, Czech Republic, May, 2004.

[5]C. Stauffer and W. Grimson , Learning Patterns of Activity Using Real Time Tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pages 747-767, Aug. 2000.

[6]Dockstader et al, Multiple camera tracking of interacting and occluded human motion, *Proceedings of the IEEE* , Vol: 89 , Issue: 10 , pp.1441-1455, Oct. 2001.

[7]Ting-Hsun Chang, Shaogang Gong, and Eng-Jong, Tracking multiple people under occlusion using multiple cameras. In Proc.11th British Machine Vision Conference, 2000.

[8]S.L. Dockstader and A.M. Tekalp, Multiple camera fusion for multi-object tracking. *In Proc. IEEE Workshop on Multi-Object Tracking*, 2001. pp. 95-102.

[9]R.Cucchiara, C.Grana, G.Tardini, R.Vezzani, Probabilistic people tracking for occlusion handling, *Proceedings of the 17th International Conference on ICPR 2004*, Vol:1, pp. 132 – 135 Aug. 23-26, 2004.

[10]How-Lung Eng, et al. A bayesian framework for robust human detection and occlusion handling using human shape model. *Proceedings of the 17th International Conference on ICPR 2004*, Vol: 2 , pp. 257-260, August 23-26, 2004.

[11]Ying Wu, Ting Yu, Gang Hua, Tracking appearances with occlusions, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003*. Vol: 1, pp. 789-795, June 2003.

[12]A. Senior, et al. Appearance Models for Occlusion Handling. *Proc. 2nd IEEE Int. Workshop on PETS*, Kauai,Hawaii,USA, December 9,2001.

[13]A. Senior, Tracking with Probabilistic Appearance Models, *Proc. ECCV workshop on Performance Evaluation of Tracking and Surveillance Systems*, pp. 48-55, 1 June 2002.

[14]N. T. Siebel, S. Maybank, Fusion of Multiple Tracking Algorithms for Robust People Tracking, *7^{th} European Conf. on Computer Vision*, Denmark, Vol.IV, pp. 373-387, May 2002.

[15]Hieu T. Nguyen and Arnold W.M. Smeulders, Fast Occluded Object Tracking by a Robust Appearance Filter, *IEEE Transactions on Pattern Analysis and Machine Intelligence.*Vol.26, No.8, pp. 1099-1104, August 2004.

[16]A. Yilmaz and M. Shah, Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 11, November, 2004.

[17]H. Tao, H. S. Sawhney, and R. Kumar, Dynamic Layer Representation with Applications to Tracking. *Proc.Computer Vision and Pattern Recognition*, Vol.2, pp. 134-141,2000.

[18]M. Isard and A. Blake, CONDENSATION – Conditional Density Propagation for Visual Tracking, *International Journal on Computer Vision* 1(29),1998.

[19]M. Isard and A. Blake, Contour Tracking by Stochastic Propagation of Conditional Density, *In ECCV '96*, pp. 343-35,1996.

[20]A. Doucet, N. Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer. 2001.

[21]R. Rosales and S. Sclaroff, Improved Tracking of Multiple Humans with Trajectory Prediction and Occlusion Modeling, *Proc.IEEE Conf. on Computer Vision and Pattern Recognition*. Workshop on the Interpretation of Visual Motion, Santa Barbara,CA,1998.

[22]P. KawTraKulPong , R. Bowden, An improved adaptive background mixture model for real-time tracking with shadow detection, *In Second European Workshop on Advanced Video-based Surveillance Systems*,2001.

[23]*http://homepages.inf.ed.ac.uk/ rbf/CAVIAR/*. 2004.