Multi-Camera Visual Surveillance for Motion Detection, Occlusion Handling, Tracking and Event Recognition

Oytun Akman, A. Aydin Alatan, Tolga Çiloglu

▶ **To cite this version:**

Oytun Akman, A. Aydin Alatan, Tolga Çiloglu. Multi-Camera Visual Surveillance for Motion Detection, Occlusion Handling, Tracking and Event Recognition. Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications - M2SFA2 2008, Andrea Cavallaro and Hamid Aghajan, Oct 2008, Marseille, France. inria-00326780

HAL Id: inria-00326780

https://inria.hal.science/inria-00326780

Submitted on 5 Oct 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Camera Visual Surveillance for Motion Detection, Occlusion Handling, Tracking and Event Recognition[1]

Oytun Akman, A. Aydın Alatan, and Tolga Çiloğlu

Department of Electrical and Electronics Engineering, M.E.T.U., Ankara, Turkey
{oytun, alatan}@eee.metu.edu.tr

Abstract. This paper presents novel approaches for background modeling, occlusion handling and event recognition by using multi-camera configurations that can be used to overcome the limitations of the single camera configurations. The main novelty in proposed background modeling approach is building multivariate Gaussians background model for each pixel of the reference camera by utilizing homography-related positions. Also, occlusion handling is achieved by generation of the top-view via trifocal tensors, as a result of matching over-segmented regions instead of pixels. The resulting graph is segmented into objects after determining the minimum spanning tree of this graph. Tracking of multi-view data is obtained by utilizing measurements across the views in case of occlusions. The last contribution is the classification of the resulting trajectories by GM-HMMs, yielding better results for using together all different view trajectories of the same object. Hence, multi-camera sensing is fully exploited from motion detection to event modeling.

1 Introduction

The field of machine (computer) vision is concerned with problems that involve interfacing computers with their surrounding environment through visual means. One such problem, surveillance, has an objective to monitor a given environment and report the information about the observed activity that is of significant interest. With the decrease in costs of off-the-shelf hardware for sensing and computing, and the increase in the processor speeds, surveillance systems have become commercially available, and applied to a number of different applications, such as traffic monitoring, airport and bank security, etc. However, state-of-the-art visual surveillance algorithms that automatically analyze the scene, especially with a single camera, (e.g., [1–3]) are still severely affected by many shortcomings, such as occlusions, shadows, illumination differences, complex movements, etc. In this respect, multi-camera visual networks are becoming more popular due to the exploitation of 3D information and the presence of larger field of view due to observation from different angles for better performance.

[1] This work was supported by TÜBİTAK under COST 292 and EC IST 6th Framework 3DTV NoE.

In general, a typical surveillance process can be decomposed into 3 main tasks, as *moving object detection*, *object tracking* and *event recognition*. In this research effort, three novel methods for background modeling, occlusion handling and event recognition for multi-camera configurations are presented. A common (joint) background model for two-views is constructed by using mixture of multivariate Gaussians. However, instead of constructing the observation vector by color and depth information, which is relatively difficult to obtain, RGB values of homography-related pixels between views are utilized. Occlusion handling is achieved by top-view generation via trifocal tensors. Over-segmented foreground regions are matched between different views and transferred to the top-view. Next, transferred points are segmented into objects via graph-based clustering. In the proposed event recognition method, trajectories of the object are extracted for all views during tracking and these trajectories are concatenated to generate a multi-view trajectory. Instead of single-view trajectories, these multi-view trajectories are used as observations for training a GM-HMM. Then, incoming object trajectories are tested against this GM-HMM to classify their motion as normal or abnormal, in order to detect incidents in the scene.

2 Moving Object Detection

Foreground object detection via background subtraction has been used extensively in video surveillance applications due to its satisfactory performance and computational effectiveness [1, 3–6]. However, in case of a single camera, it is relatively difficult to deal with erroneous segmentation results due to the dynamic scenes and shadows. These false segmentations usually result in performance degradation in subsequent actions, such as tracking and event recognition.

Utilization of multiple cameras leads to better handling of dynamic scenes, shadows and illumination changes due to the exploitation of 3D information, compared to a single camera. However, multi-camera methods usually have heavier computational load than single camera methods have. Most of the methods employing multi-camera systems, which are discussed in the following section, use stereo cameras and depth information to model their background. However, stereo cameras are not available in most of the surveillance applications, and systems consisting of individual cameras with intersecting field of views (FOVs) are generally preferred, due to their wide area coverage. In this section, multi-camera background modeling method, which is comparatively less demanding in terms of computation, is proposed. This method uses the information coming from two separate cameras with intersecting FOVs, which is obtained by relating the input images by the homography matrix. The incoming information from the cameras is concatenated by using a simple fusion method and therefore, it achieves a real-time performance.

2.1 Related Work on Multi-Camera Object Detection

Many methods have attempted to solve some of the background modeling problems by using multi-camera configurations [5, 6]. Most of these approaches use

the depth information from stereo cameras (narrow-baseline) to segment the foreground regions [6]. In [5], 3D-geometry of a static scene is reconstructed by using images captured from a number of calibrated cameras. However, these methods are relatively difficult to apply for wide-baseline camera configurations.

On the other hand, Harville et al. [3] propose a multimodal system, which adapts Gaussian mixture models of the background appearance, to the combined image feature space of depth and luminance-normalized color. They use spatially-registered, time-synchronized pairs of color and depth images that are obtained by static cameras. However, the complexity of the algorithm is relatively high and the depth information for each pixel is, again, difficult to robustly estimate for any wide-baseline camera configuration. Therefore, instead of using color and depth information together, we propose a method in which the color values of homography-related pixels between views can be used.

2.2 Mixture of Multivariate Gaussians Background Model

Multivariate Gaussians can be thought of as a generalization to higher dimensions of the one-dimensional Gaussians. Therefore, mixture of multivariate Gaussians background modeling can be thought of as a generalization of its single camera counterpart [4]. Each dimension of this multivariate model is obtained by using the information from one of the cameras and the simplest way of relating different views is fusing the available images via *homography* [7] with the assumption that the observed scene is approximately planar. Each pixel in the main camera view and intersecting FOVs unified background model is constructed by a mixture of K multivariate Gaussian distributions. The pixels in the non-overlapping field of the main camera view are modeled by using mixture of univariate Gaussians, which is similar to the single camera case. Then, the probability of observing a pixel value X_N in the common FOV at time N is

$$\rho(X_N) = \sum_{k=1}^{K} w_k \frac{1}{(2\pi)^{D/2} |\sum_k|^{1/2}} e^{-\frac{1}{2}(X_N-\mu_k)^T \sum_k^{-1} (X_N-\mu_k)} \tag{1}$$

where $X_N = \begin{bmatrix} x_N \\ x'_N \end{bmatrix}$, $\mu_k = \begin{bmatrix} \mu_{k_1} \\ \mu_{k_2} \end{bmatrix}$, $\sum_k = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2 \\ \sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$, w_k is the weight of k^{th} Gaussian and $D = 2$ for two camera case. In the formulation above, x_N refers to a pixel in the main camera's intersecting FOV, while x'_N is the homographic projection of onto the auxiliary camera's intersecting FOV. Therefore, x_N, x'_N pairs satisfy the $x'_N = H_{12}x_N$ relation. Finally, for each observation vector X_N, there are 3 K-mixture multivariate Gaussian models, each corresponding to a color channel, R, G or B. For other pixels, formulation that is explained in [4] is used.

Every new pixel value, x_t, is compared against the resulting K Gaussian distributions, until a *match* is found. A match is defined as a pixel value within 2.5 standard deviations of a distribution. If none of the K distributions match the current pixel value, the least probable distribution is replaced with the current

value as its mean value, an initially high variance, and low prior weight. The prior weight of the k^{th} Gaussian at time t is adjusted as follows [4]

$$w_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha M_{k,t} \tag{2}$$

where α is the learning rate and $M_{k,t}$ is 1 for matched models, and 0 for remaining models. The μ and σ for unmatched distributions remain same, while the matched ones are updated as follows [4]

$$\mu_t = (1 - \varphi)\mu_{t-1} + \varphi X_t \text{ and } \sum_t = (1 - \varphi)\sum_{t-1} + \varphi(X_t - \mu_t)(X_t - \mu_t)^T \tag{3}$$

where $\varphi = \alpha\eta(X_t|\mu_k, \sigma_k)$.

2.3 Simulations on Multi-camera Object Detection

Various multi-camera surveillance sequences are tested by the proposed approach. The mixture of K ($K = 5$) multivariate Gaussian densities are estimated via online EM-algorithm, as described in Section 2.2, approximately by using 30 frames. The required homography relation is obtained by the help of calibration markers on the scene. A typical result is given in Fig.1 for both univariate and multivariate mixture of Gaussian cases. The proposed approach has higher robustness against erroneous segmentation in one of the views, as it can be observed from Fig.1. Apart from this fact, the objects, which are occluded by static background in one of the views, can be correctly segmented as foreground, if the object is observed in the other views. Unfortunately, the assumption of planar objects might yield undesired masks, especially for tall vehicles. Therefore, the method is more suitable for the cases in which the cameras are placed on higher locations, where the height of the object becomes negligible with respect to the camera position. Moreover, the presented algorithm achieves real time performance (on a Pentium-IV 3 GHz PC) due to the low computational load of homographic projections.

3 Occlusion Handling via Multi-Camera

Occlusions of moving objects are one of the major problems in any surveillance system. During the moving object detection process, occlusions cause moving objects to be either segmented in an erroneous shape or became completely lost. These false segmentation results might cause subsequent actions to fail or a decrease in their performance. Utilization of multiple cameras should lead to better handling of occlusions compared to a single camera, due to the presence of different views of the same scene.

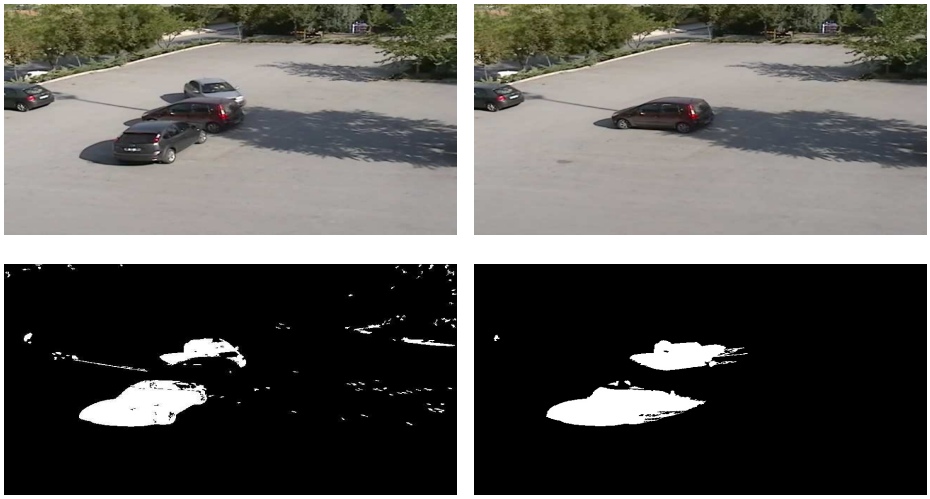

Fig. 1: Simulation results of Multivariate MOG from left to right and top to bottom, input frame, background model, foreground mask due to single camera MOG, and foreground mask due to multivariate MOG

3.1 Related Work on Occlusion Handling

There are various methods for occlusion handling in mono-camera surveillance [1, 2], all of which have limited performance, especially in case of severe occlusions. All these methods have the inevitable drawback of handling (tracking) objects that might be initially occluded. Apart from the aforementioned methods, some feature-based trackers are also used to handle occlusions where only partial occlusions are handled [8]. However, segmentation of these features into individual objects generally fails during dynamic occlusions. On the other hand, multi-camera surveillance systems have improved occlusion handling capabilities with respect to their single-camera counterparts. M2Tracker [9] uses a region-based stereo matching algorithm to determine 3D points on an object, and utilize Bayesian pixel classification with occlusion analysis to segment people occluded in different levels of crowd density. The complexity of this algorithm is relatively high due to pixel-wise classification. Moreover, in [10], Mittal and Davis describe an algorithm for detecting and tracking multiple people in a cluttered scene by using multiple synchronized cameras located far away from each other.

3.2 Occlusion Handling from Multi-view Video

The most promising way of handling occlusion problem is to generate (virtual) view(s) of a scene from the available multi-view data, so that this novel view(s) is free of occlusions. However, the rendering process is a computationally demanding procedure for all pixels; hence, such an approach should be applied

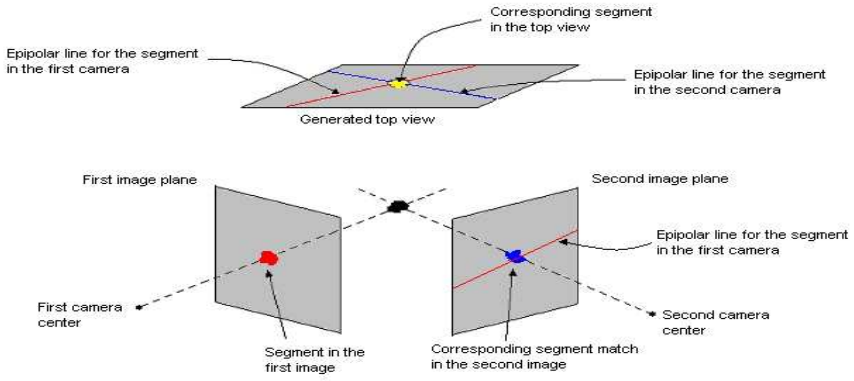

Fig. 2: Point transfer by using trifocal tensor

in a simpler manner by utilizing small regions, instead of pixels [10]. Following the moving object detection step, foreground regions are segmented into homogeneous regions based on their color content. These small segments are used to replace the pixel-based view rendering process, during the estimation of the epipolar geometry between views. To this aim, all the center of masses for these segments are transferred to an imaginary top-view via trifocal tensor, as shown in Fig.2, by intersecting the epipolar lines of two matching points on the top-view [7]. The points in the resulting top-view must be segmented in order to cluster them into individual objects; the only available information are the positions and the (average) color values of these points. Point clustering is performed by using *minimal spanning tree* [11] of the graph resulting from these points, where top-view points are defined as nodes (or vertices) of this undirected graph. Affinity-based graph theoretic clustering is avoided due to its computational burden. In order to further decrease the computational cost, only one node is chosen in $N \times N$ neighborhood of that node. Assuming that the segments of the same object are close to each other and in most of the cases they have similar color values, the weight of a link or an edge between nodes *nodes-i* and *-j* could be defined as

$$W_{ij} = \alpha H_{diff} + (1 - \alpha) D_{diff} \tag{4}$$

where H_{diff} is the difference between the hue values of the *nodes-i* and *-j*, D_{diff} is the Euclidian distance between these nodes, whereas α is the weight between these measures. The edges, whose weights are larger than a certain threshold value in the minimal spanning tree, are cut and minimal spanning forests are generated.

3.3 Simulation Results for Occlusion Handling

The data described in Section 2.3 are utilized during these tests. Typical results are given in Fig.3. Clustered minimal spanning forests with different colors in

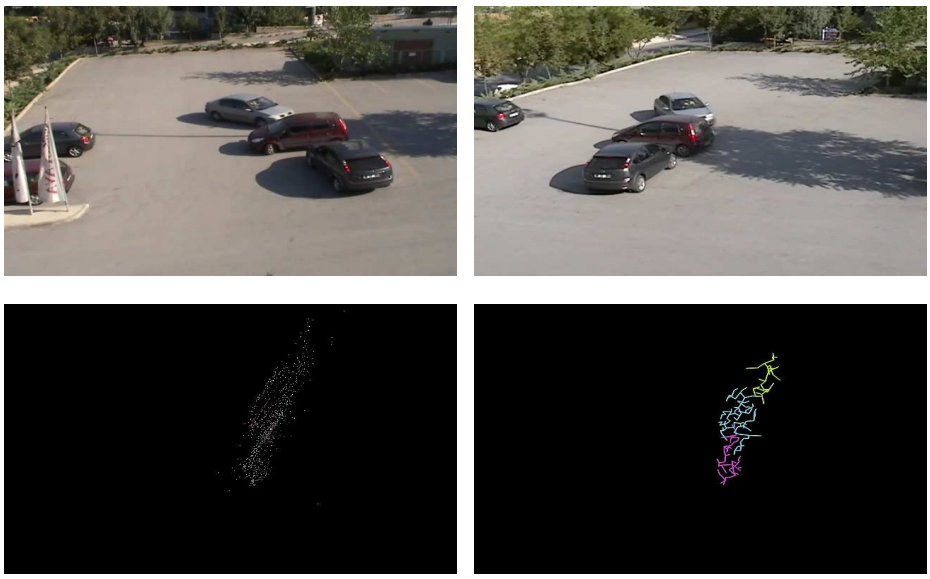

Fig. 3: Simulation results of occlusion handling from left to right and top to bottom, first input image, second input image, from the top-view transferred segment centers, clustered minimal spanning forest

the bottom-right image represent the segmented objects. The performance of the overall system is quite acceptable for segmenting partially occluded objects. However, under strong occlusions, epipolar matching cannot be performed accurately and the generated top view might not represent the foreground objects. Moreover, when the objects in the mask have similar color values and they are close to each other, the foreground mask is under-segmented, since the transferred points might not be distinguishable.

4 Tracking and Event Recognition from Multi-Camera

Tracking of objects in a scene is another imperative requirement for any video surveillance system. The main causes of tracking failure in mono-camera cases are static and dynamic occlusions. Considering a minimal amount of occlusions, a single camera might be sufficient for tracking. However, as the densities of static and dynamic occlusions are increased, multi-camera configurations with a larger field of view should be capable of resolving static, as well as dynamic, occlusions better than single-camera configurations. It should be noted that in multi-camera systems, occlusions might occur at different time instants for separate views, and the overall system should be able to track occluded objects successfully after exploiting the available track information in different views. By the help of such a multi-camera tracking method, the resulting trajectories of the object can be

used for event recognition. Since multi-camera information is being utilized, these trajectories lead to better interpretations of activity type that is of significant interest.

4.1 Related Work on Tracking and Event Detection

The previous research efforts indicate that the multi-camera employment gives promising results also during tracking. Especially for the occluded scenes, collaboration of multi-camera configurations leads to better handling of track losses [12]. Among different methods, the approach by Black et al. [12] is pursued for developing a tracking algorithm due to the promising results for their relatively simple algorithm.

For event analysis, many methods have been proposed to identify unusual activities in the scene [13, 14]. Bashir et al. [15] employ Hidden Markov Models (HMMs) to recognize the segmented trajectories, rather than GMMs, and those hidden states in HMMs are represented by GMs. Porikli and Li [16] propose a traffic congestion estimation algorithm that employs Gaussian Mixture Hidden Markov Models (GM-HMM), while they extract the congestion features in the compressed domain by less accurate motion vectors.

4.2 Tracking

As stated in [12], moving object segmentation is performed by background subtraction and Kalman filter is used to track segmented objects in each of the views. In the proposed approach, in case of no measurement associated with the track due to an occlusion, the other view(s) are used to generate measurement. In order to fuse the information coming from different cameras, a relation between these two tracks should be defined. The simplest way of relating different views is point transferring via homography, since epipolar matching, by using epipolar lines, is computationally expensive and difficult to perform for wide-baseline configurations.

In the proposed approach, tracking is performed in each view and trackers in different views are related to each other via homography. The object states are tracked in 2D by using separate Kalman filters. The object state in 2D Kalman filter includes the image location of object as well as its velocity in pixels and constant speed assumption for object velocity is used. When an object is observed for the first time, a separate Kalman filter is initiated for this object. Then, for next incoming frame, background subtraction is performed and a set of foreground moving objects are obtained. Among these objects, the nearest moving object to the predicted state of the tracked object is labeled as the next position of that object and the position is used to update the corresponding Kalman filter. Hence, object association between the consecutive frames is achieved via Kalman predictions. If the distance between the nearest moving object and the predicted state of tracked object is larger than some certain threshold, or there is no moving object in the foreground mask, then this situation is denoted as *no measurement case*. When no measurement case occurs in one of the views,

mostly due to occlusions, if there is a measurement in the other view for tracked object, the position information projected from the other view is used as a measurement to update for the corresponding Kalman filter. In order to utilize this information, a match of the object in the other camera view must be determined. Given a set of detected moving objects in each camera view, a match between a correspondence pair is defined when the following *transfer error condition* [12] is satisfied:

$$(x' - Hx)^2 + (x - H^{-1}x')^2 < \tau \tag{5}$$

where x and x' are image coordinates in the first and second camera views, respectively. This constraint is applied to determine correspondence between the moving objects that are already detected in each camera view. The representative coordinate of an object is assumed to be the closest point of its foreground mask to the ground plane (which is generally the rear end of the mask), since its location minimizes the projection errors. When no measurement case occurs in both of the views associated with the tracked object, updates of the Kalman filters are calculated separately by using corresponding predicted state values. The aforementioned steps are repeated until the tracked object leaves the field of view (FOV). When the object leaves the common FOV, its trajectories for both of the cameras are extracted for event recognition.

4.3 Simulation Results on Tracking

The discussed method is tested by using various video sequences in each of which a vehicle runs in front of a camera. Also, individual performances of trackers (without other view measurement assistance) are tested by using the same video sequences and a typical result is given in the Fig.4 where trajectories of tracked objects are shown in red. The performance of the proposed method is strongly dependent on the correct matching performance of the tracked object between the two views. Therefore, the object must be segmented correctly in the initial frames, as soon as it enters the common FOV. Moreover, a couple of correct measurements are needed to initiate the Kalman filters. The small variations in the object trajectories are caused by erroneous foreground masks. During background subtraction step, the foreground mask of the object differs slightly in size (especially along its borders) between consecutive frames. Therefore, the location of the object, which is the rear-end point of its foreground mask, slightly vibrates between frames. This method has an obvious advantage compared to single-camera tracking, when tracked object stops behind an obstacle. As long as one of the cameras continues to observe the object, it can be tracked correctly along the frames. However, in the single-camera case, Kalman tracker for the occluded object would fail and object becomes lost. As shown in the simulation results, Kalman trackers (without assistance) failed and lost the track of object when it passed behind an obstacle. Moreover, new Kalman trackers are erroneously initiated, since occluded objects are incorrectly identified as if they are appearing for the first time.

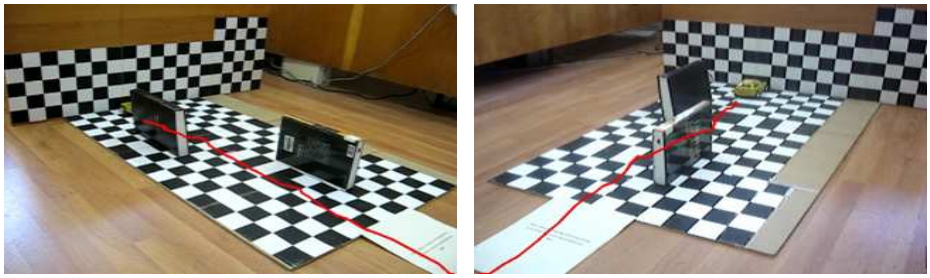

Fig. 4: Simulation results of multi-camera tracking algorithm, (left) other view measurements are not used in 'no measurement case', (right) both view measurements are utilized

4.4 Event Recognition via Multi-view Trajectories

A GM-HMM based method is utilized to recognize the spatio-temporal events that occur in the observed scene. The events are classified into two sets, as normal and abnormal due to their spatio-temporal behaviors that are stored in their trajectories. *Normality* is simply defined as existing in the training set. The extracted object trajectories, which are accepted to be normal, are used to train a GM-HMM and new trajectories are classified by using this model. For the single camera case, the measurements can be obtained by using only one sensor. Therefore, the extracted data are limited, compared to the multi-camera case. In the proposed multi-camera (two cameras) event recognition method, three different trajectory matrices are defined for each object, such that the first one is the image locations of the object in the first camera view, whereas the second one is the image locations of its matched pair in the second camera view. The last trajectory matrix is obtained by simple concatenation of the first and the second trajectories. The positions of the i^{th} object in the k^{th} frame of first and second camera are shown as the position vectors;

$$p^i_{k_1} = \left[\, x^i_{k_1} \; y^i_{k_1} \,\right] \text{ and } p^i_{k_2} = \left[\, x^i_{k_2} \; y^i_{k_2} \,\right] \tag{6}$$

where x and y are the image coordinates in the corresponding camera views. Then, the first and second trajectory matrices consist of

$$tr^1(i) = \begin{bmatrix} x^i_{m_1} & y^i_{m_1} \\ x^i_{m_1+1} & y^i_{m_1+1} \\ \vdots & \vdots \\ x^i_{n_1} & y^i_{n_1} \end{bmatrix} \text{ and } tr^2(i) = \begin{bmatrix} x^i_{m_2} & y^i_{m_2} \\ x^i_{m_2+1} & y^i_{m_2+1} \\ \vdots & \vdots \\ x^i_{n_2} & y^i_{n_2} \end{bmatrix} . \tag{7}$$

The third trajectory matrix is

$$tr^3(i) = \begin{bmatrix} x^i_{m_1} & x^i_{m_2} & y^i_{m_1} & y^i_{m_2} \\ x^i_{m_1+1} & x^i_{m_2+1} & y^i_{m_1+1} & y^i_{m_2+1} \\ \vdots & \vdots & \vdots & \vdots \\ x^i_{n_1} & x^i_{n_2} & y^i_{n_1} & y^i_{n_2} \end{bmatrix} \tag{8}$$

where m denotes the starting frame number, in which the object enters the FOV, and n is the end frame number, in which the object leaves the FOV. These three trajectory matrices are used to train 3 different GM-HMMs and the incoming trajectory matrices are tested separately by using these three models in order to check their applicability to these trained models. The simulation results are given in the following section.

4.5 Simulation Results on Event Detection

During the simulations for the proposed multi-camera (two cameras) event recognition method, 3 different trajectory matrices are defined for each object, such that the first set of matrices contain the positions of the objects in the first camera view, whereas those of the second set contain image locations of matched pairs in the second view. The third trajectory matrix is obtained by concatenating the first and the second trajectories. These three trajectory matrices are used to train 3 different GM-HMMs and the incoming trajectory matrices are tested separately by using these three models. Each model involves left-to-right connected 4 states. Each of these 3 GM-HMMs is trained by using 270 trajectory matrices from typical traffic surveillance data. Then, various *abnormal* cases (such as reverse traffic flow or lane crossing) are tested against each of these 3 GM-HMMs separately. As a result of training, the average Viterbi distances of the training objects for these three models are obtained, as, 10.20, 10.06 and 20.04, respectively. Table 1 presents the Viterbi distances of some typical test matrices, which are known to be abnormal, against these 3 models, whereas the ratios of Viterbi distances of these test trajectories to the average Viterbi distances are given in the Table 2. These ratios indicate that utilization of both trajectories for modeling gives a better discrimination against recognition of abnormal events.

Table 1: Test case: Viterbi distances of test objects trajectories to the models

Object ID	Viterbi distance to GM-HMM-1	Viterbi distance to GM-HMM-1	Viterbi distance to GM-HMM-1+2
1	20.4	19.9	45.1
2	21.2	19.7	45.0
3	26.9	24.7	55.2
4	10.7	10.5	21.2
5	10.4	10.5	22.1
6	10.1	9.9	19.7

Table 2: Ratios of the Viterbi distances of test objects trajectories to the average Viterbi distances of training objects trajectories

Object ID	Viterbi distance / Average-1	Viterbi distance / Average-2	Viterbi distance / Average-1+2
1	2.00	1.98	2.25
2	2.08	1.96	2.24
3	2.64	2.45	2.75
4	1.05	1.05	1.05
5	1.02	1.04	1.10
6	0.99	0.99	0.98

References

1. Dockstader, S., Tekalp, A.: Tracking multiple objects in the presence of articulated and occluded motion. Workshop on Human Motion (2000) 88–95
2. Haritaoglu, I., Harwood, D., Davis, L.S.: W4: Real-time surveillance of people and their activities. PAMI **22** (2000) 809–830
3. Harville, M., Gordon, G., Woodfill, J.: Adaptive video background modeling using color and depth. In: ICIP. Volume 3. (2001) 90–93
4. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: CVPR. Volume 02. (1999) 2246
5. Lim, S.N., Mittal, A., Davis, L.S., Paragios, N.: Fast illumination-invariant background subtraction using two views: Error analysis, sensor placement and applications. In: CVPR. Volume 1. (2005) 1071–1078
6. Goldlucke, B., Magnor, M.: Joint 3D-reconstruction and background separation in multiple views using graph cuts. In: CVPR. Volume 1. (2003) 683–688
7. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press (2004)
8. McLauchlan, P., Beymer, D., Coifman, B., Mali, J.: A real-time computer vision system for measuring traffic parameters. In: CVPR. Volume 0. (1997) 495
9. Mittal, A., Davis, L.S.: M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. IJCV (2003) 189–203
10. Mittal, A., Davis, L.: Unified multi-camera detection and tracking using region-matching. In: WOMOT. Volume 0. (2001) 3
11. Eppstein, D.: Spanning trees and spanners. In Sack, J.R., Urrutia, J., eds.: Handbook of Computational Geometry. Elsevier (2000) 425–461
12. Black, J., Ellis, T., Rosin, P.: Multi view image surveillance and tracking. Workshop on Motion and Video Computing (2002) 169–174
13. Brand, M., Oliver, N., Pentland, A.: Coupled hidden Markov models for complex action recognition. In: CVPR. Volume 0. (1997) 994
14. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. PAMI **22** (2000) 747–757
15. Bashir, F., Qu, W., Khokhar, A., Schonfeld, D.: HMM-based motion recognition system using segmented PCA. In: ICIP. Volume 3. (2005) 1288–1291
16. Porikli, F., Li, X.: Traffic congestion estimation using HMM models without vehicle tracking. Intelligent Vehicles Symposium (2004) 188–193