Multi-Camera Target Tracking in Blind Regions of Cameras with Non-overlapping Fields of View

Amit Chilgunde*, Pankaj Kumar†, Surendra Ranganath*, Huang WeiMin†
*Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117576.
†Institute for InfoComm Research, 21 Heng Mui Keng Terrace, Singapore 119613.
*{eng01277,elesr@nus.edu.sg} †{kumar,wmhuang@i2r.a-star.edu.sg}

Abstract

In this paper, we propose a real time system for tracking targets across blind regions of multiple cameras with non-overlapping fields of views (FOVs) using camera topology, and targets' motion and shape information. Kalman filters are used to robustly track each target's shape and motion in each camera view and the common ground plane view composed of all camera views. The target's track in the blind region between cameras is obtained using Kalman filter predictions. For multi-camera correspondence matching we compute the Gaussian distributions of the tracking parameters across cameras for the target motion and position in the ground plane view. Matching of targets across camera views uses a graph based track initialization scheme, which accumulates information from occurrences of target in several consecutive frames of the video. Probabilistic matching is carried out by using the track parameters for new tracks obtained from the graph in a camera view with the parameters of the terminated tracks learnt by Kalman filters in the other camera views and ground plane view. We obtain 85% accuracy for corresponding matching while tracking vehicles observed from two cameras monitoring a highway.

1 Introduction

Tracking objects in multiple cameras is of interest for wide area video surveillance systems. Multi-camera tracking with non-overlapping fields of view (FOV) involves the tracking of targets in the blind region and the correspondence matching of targets across cameras. We consider these problems in this paper.

In the blind regions between the cameras, we track the targets in a common ground plane view which has the different camera views mapped onto it using homography. For homography to work it is required that the surfaces being mapped be planar, or it should at least be possible to decompose a non-planar surface which is to be mapped into several

BMVC 2004 doi:10.5244/C.18.42

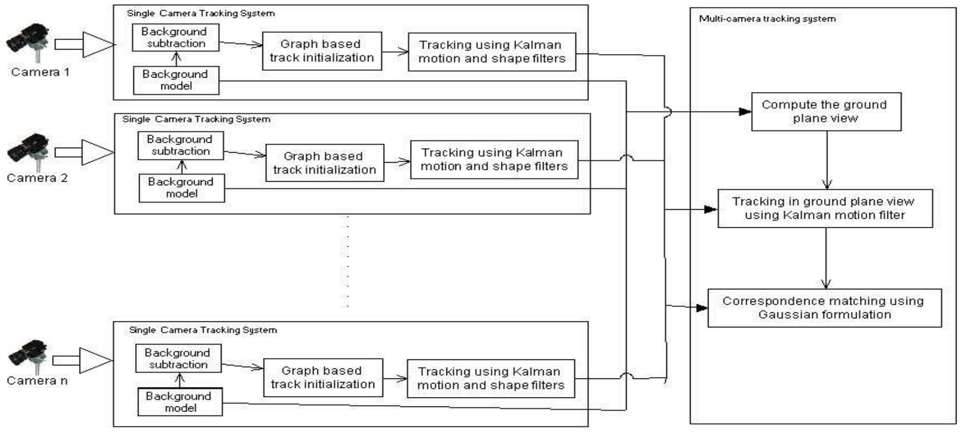

Single Camera Tracking System
Background subtraction
Background model
Graph based track initialization
Tracking using Kalman motion and shape filters
Camera 1
Camera 2
Camera n
Multi-camera tracking system
Compute the ground plane view
Tracking in ground plane view using Kalman motion filter
Correspondence matching using Gaussian formulation

Figure 1: Schematic block diagram of the multi-camera tracking system.

planar patches. Even in the latter case, the Kalman filters used for tracking and correspondence matching gives good results. When targets are not in the FOV of any of the cameras, the Kalman filter continues to track the most likely path of the target in the ground plane view. When the target reemerges in the FOV of another camera, a new track for this target is constructed using a graph theoretic approach. Correspondence matching for cameras with non-overlapping FOVs is done using cues like location of the cameras, the topography, and information of the targets such as shape and motion. Figure 1 shows an overview of our system. The parameters of new tracks in each camera view are compared with the parameters of the terminated tracks learnt in the previous camera views. We use Gaussian distributions to model the error in position and shape parameters of the targets across camera views, so that the probability of match for two targets in different views can be calculated. The parameters of the Gaussian distributions are obtained using a training data set.

The rest of the paper is organized as follows: In Section 2, we review some prior works in multi-camera tracking for cameras with non-overlapping FOVs. Next, in Section 3 for completeness, we briefly discuss the single camera tracking system which is a pre-requisite for our system and the graph algorithm for new track initialization in a single camera view. Section 4 describes our multi-camera tracking and correspondence matching scheme for targets across camera views. Section 5 shows the results of a two camera tracking system and finally, we conclude the paper in Section 6.

2 Related Work

Most of the work on multi-camera surveillance assumes overlapping camera views and the focus is on using color information of the objects and camera calibration. Kettnaker *et al.* [6] make use of the corridor topology and usual walking speed of people to form expectations about the time windows and the locations in which people will reappear next. They also make use of color information of the targets to achieve correspondence matching by histogram comparisons after normalizing for lighting. Javed *et al.* [3] automatically estimate the line where the feet of pedestrians should appear in a secondary camera when they exit the reference camera. They make use of the relationship between the camera FOV boundaries to establish correspondence between views of the same object in multiple cameras. In [4] the authors make use of Parzen windows to estimate the inter-camera space-time probabilities from the training data i.e. probability of an object entering a certain camera at a certain time given the location, time and velocity of its exit from the other cameras.

Kettnaker *et al.* in [5] use a Bayesian formalism for the correspondence task, where the optimal solution is the set of object paths with highest posterior probability given the observed data. Huang *et al.* [2] also use a probabilistic model for finding correspondence between vehicles on a highway in which the transition times of the objects between two cameras are modeled as Gaussian distributions. They make use of the velocity, location, size and color information of the vehicles to achieve correspondence matching. In [10], Porikli *et al.* make use of the camera topography. The correlation between camera layout and likelihood of the objects appearing in a certain camera after they exit from another one is formulated by using a probabilistic Bayesian network where cameras form the nodes and the edges represent the transition probabilities i.e. the likelihood of a person moving from one camera to another. Color calibration between cameras is done by forming a matrix of histogram bin distances. In [7], Khan *et al.* design a system that discovers spatial relationships between the camera FOVs and use this information to make the correspondence between different perspective views of the same person. However, this system assumes overlapping camera FOVs. In [9], Markis *et al.* automatically learn the camera topology and the entry/exit zones of a network of non-calibrated cameras with overlapping as well as non-overlapping FOVs.

3 Single Camera Tracking

Firstly, a robust real-time single camera target tracking system is implemented, which uses background subtraction to detect moving foreground objects as segmented patches (*SPs*). The *SPs* are approximated by fitting ellipses around them. Kalman filters are employed to track the position and motion of targets as in [8]. This system is robust to merges, splits and occlusions. Each target has the following ellipse parameters for representation: the major axis a, minor axis b, and the centroid (X_c, Y_c). Data association is done using shape and location information of the targets. We define a match measure D_s for data association between target 1 in the previous frame and target 2 in the current frame as:

$$D_s = c_1 \times |\hat{a_1} - a_2| + c_2 \times |\hat{b_1} - b_2| + c_3 \times |\hat{X_{1c}} - X_{2c}| + c_4 \times |\hat{Y_{1c}} - Y_{2c}| \tag{1}$$

where $\hat{a_1}$, $\hat{b_1}$, $\hat{X_{1c}}$ and $\hat{Y_{1c}}$ are the Kalman predicted values of the major axis, minor axis, and centroid, respectively, of target 1 for the current frame while a_2, b_2, X_{2c} and Y_{2c} are the corresponding measured values of target 2 in the current frame. $c_1 ... c_4$ are constants that determine the weight of each component in the match measure. In our experiments we used c_1 = c_2 = 0.8 and c_3 = c_4 = 1. Less weight is given to the shape parameters since they are noisy due to foreground segmentation errors.

A Kalman filter is used to track the shape parameters of a target. For shape tracking, the measurement vector of the *SP* for the k^{th} target in the n^{th} frame can be written as:

$$\mathbf{SP}_k^n = [a_k^n, b_k^n, S_{a_k}^n, S_{b_k}^n]$$

where a and b are the major and minor axes of the ellipse and $S_{a_k}^n$, $S_{b_k}^n$ are the parameters for shape change which are obtained as:

$$S_{a_k}^n = \frac{a_k^n}{a_k^{n-1}}, \qquad S_{b_k}^n = \frac{b_k^n}{b_k^{n-1}} \tag{2}$$

3.1 Track Initialization

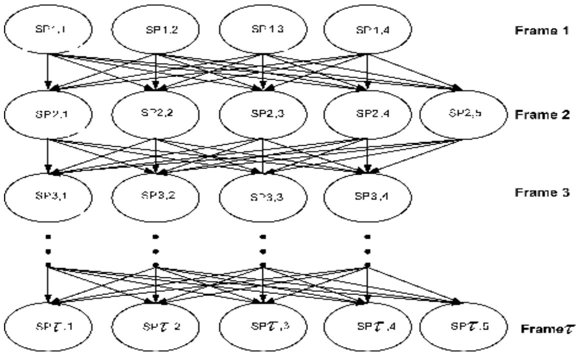


Figure 2: Structure of the attributed graph used for track initialization.

Robust track initialization is achieved using a graph based method similar to that in [8, 12]. New targets are initialized for tracking only when a target's measurements are reliably available in the past τ frames. This makes the initialization accurate and the tracker stable.

Automatic initialization of target tracks is done by using an attributed graph of the *SPs* in τ frames, as shown in Figure 2. The attributes of each node in the graph are: 1. the frame number, 2. the centroid, 3. shape parameters, 4. parent id and child id. Edges are present between nodes whose frame number differ by 1 as shown in Figure 2. The weights of these edges is the value of match measure D_s between the nodes. The nodes with frame number 1 are considered as source nodes and nodes with frame number τ are considered as destination nodes. From all source nodes the shortest paths to all destination nodes are computed using Dijkstra's algorithm. Amongst these shortest paths, different paths have different sum-of-weights. Of these paths the one with smallest sum-of-weights is chosen as a valid target track and is called the path of least sum-of-weights. The nodes of this path are removed from the graph to give rise to a new graph with reduced number of nodes and edges. The same process is repeated until there is no node in any one of the τ frames in the graph or the path of least sum-of-weights amongst the computed shortest paths at any iteration is greater than a heuristic threshold. For new targets which enter the FOV when tracking of other targets is in progress, another attributed graph for *SPs* which have no match with the targets being tracked is maintained. The path of least sum-of-weights amongst all the shortest paths from the source nodes to the destination nodes is computed as described earlier. The source nodes are from the first layer formed by the unmatched *SPs* in frame $(n-\tau+1)$, where n is the current frame number. The destination nodes are the unmatched *SPs* in frame n. A track for a new target is confirmed by appearance of least sum-of-weights path amongst all the shortest paths possible from the source nodes to the destination nodes.

4 Multi-camera Target Tracking

The task of multi-camera tracking is to reconstruct the paths taken by the targets that appear in the FOVs of multiple cameras and also to find the correspondence between targets in the different FOVs. To track objects in the blind region of the cameras we transform the views from each camera to a common ground plane view containing all the camera views using homography and also use a Kalman filter in this common ground plane view to track the targets. To compute the homography transform matrix H at least 4 corresponding points between each camera view and ground plane is required. We use more than four point correspondences, obtained manually, to compute H using least squares. To find the H matrix, we make actual measurements to find the co-ordinates of points on the ground. This H matrix then gives the transformation of the points in the FOV of the camera to the points on the horizontal ground plane. H can be seen as a transformation from the camera image to a planar image of the ground as seen from a camera high above the ground (see Figure 3) with the blind region shown as black.

Thus if there are n cameras in the multi-camera setup, there will be n transformation matrices H_1, $H_2 ... H_n$ where the i^{th} transformation matrix H_i gives the homography transformation between the FOV of the i^{th} camera and the ground plane view.

4.1 Tracking in blind regions between cameras

A Kalman filter for target motion maintains a state vector which consists of the co-ordinates of the centroid of the target and its velocity in the common ground plane view. The state vector of the Kalman filter for the k^{th} target in frame n is written as:

$$\mathbf{S}_k^n = [X_{c_k}^n, Y_{c_k}^n, V_{x_k}^n, V_{y_k}^n]^T \tag{3}$$

where X_c and Y_c are the co-ordinates of the centroid and V_x and V_y are the x and y components of the target velocity in the ground plane view. We assume a constant velocity model for the targets with the state equation:

$$\mathbf{S}_k^{n+1} = \mathbf{A}\,\mathbf{S}_k^n + \omega_k \tag{4}$$

where $\mathbf{A}$ is a 4×4 identity matrix and ω_k is 4×1 zero mean process noise vector.

The position and motion measurement $\mathbf{Z}_k^n$ is given by the measurement equation:

$$\mathbf{Z}_k^n = \mathbf{N}\,\mathbf{S}_k^n + \delta_k \tag{5}$$

where $\mathbf{N}$ is a 4×4 identity matrix and δ_k is a 4×1 zero-mean measurement noise vector.

Suppose the centroid of the target in the FOV of camera 1 are (x_c, y_c). Then the co-ordinates of the corresponding point in the ground plane view is given as:

$$\begin{aligned}
\begin{bmatrix} X_c' \\ Y_c' \\ \lambda_c' \end{bmatrix} &= H_1 \times \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \\
X_c &= X_c'/\lambda_c' \\
Y_c &= Y_c'/\lambda_c'
\end{aligned} \tag{6}$$

In every frame the target centroid (X_c, Y_c) is found and the Kalman filter associated with that target is updated using these measurements. Thus the filter learns the motion of the target in the ground plane view. If the target is present in the FOV of camera 2, H_2 is used in (6).

When the target exits the FOV of a camera, the Kalman filter continues to predict the movement of the target in the inter-camera region in the ground plane view without any further measurements, yielding the most likely path of the target. Now if a target enters the FOV of another camera, the problem of correspondence matching is to determine if this target is the same as a target which exited from the FOV of camera 1. We present a Gaussian formulation for this correspondence matching problem.

4.2 Gaussian formulation for correspondence matching

When a new target enters the FOV of a camera the parameters of the target track are obtained using the graph algorithm discussed in Section 3.1. The track parameters are: X_c, Y_c, the target centroid; V_x, V_y, the Kalman filter estimates of the velocity of the target; W, the width of the target; L, the length of the target; and t, the time of observation in frame numbers. The width and length of the target are the major and minor axes of the fitting ellipse obtained from each camera view and not the ground plane view. For any target in the FOV of a camera, we define an observation vector **O** using target measurements, as:

$$\mathbf{O} = [X, Y, V_x, V_y, W, L, t] \tag{7}$$

Here X, Y, V_x, V_y, W, and L are treated as independent random variables for obtaining a probabilistic match measure for target identity.

Let ΔX be the displacement of the target from its original position during time Δt. Since we assume a constant velocity model, ΔX can be written as:

$$\Delta X = V_x \times \Delta t \tag{8}$$

However, the velocity of the target may vary a little over the distance in which case (8) will not be exactly satisfied. We model ΔX as Gaussian distributed random variable with mean equal to $V_x \times \Delta t$ and a variance of Var_x obtained using training data.

Thus

$$\Delta X \sim N(V_x \times \Delta t, Var_x) \tag{9}$$

Or

$$\Delta X - V_x \times \Delta t \sim N(0, Var_x) \tag{10}$$

Similarly we model ΔY as a Gaussian distributed random variable with mean equal to $V_y \times \Delta t$ and variance of Var_y so that:

$$\Delta Y - V_y \times \Delta t \sim N(0, Var_y) \tag{11}$$

Suppose a target a is observed in the FOV of camera 1 and some time later a target b is observed in the FOV of camera 2. These give rise to observation vectors $\mathbf{O}_a{}^1$ and $\mathbf{O}_b{}^2$:

$$\mathbf{O}_a{}^1 = [X_a{}^1, Y_a{}^1, V^1_{x_a}, V^1_{y_a}, W_a{}^1, L_a{}^1, t_a{}^1] \tag{12}$$

$$\mathbf{O}_b{}^2 = [X_b{}^2, Y_b{}^2, V^2_{x_b}, V^2_{y_b}, W_b{}^2, L_b{}^2, t_b{}^2] \tag{13}$$

The changes in width and length of the targets as seen from the individual camera FOVs are also modeled with Gaussian distributions. The observed width and length of the target in one camera may be different from the width and length of the same target in the other camera. Assuming fixed location and orientation of the cameras, the change in length and width of the target from camera 1 to camera 2 can be written as:

$$W_b{}^2 = C_{w_{12}} \times W_a{}^1 \tag{14}$$

$$L_b{}^2 = C_{l_{12}} \times L_a{}^1 \tag{15}$$

where $C_{w_{12}}$ and $C_{l_{12}}$ are constants for camera pair 1 and 2. These constants are determined from training data for the multi-camera setup. Since the location of the cameras are fixed, when an object enters the FOV of one camera after exiting from the FOV of another camera, its width and length in the second camera can be related to its width and length in the first camera by constants of proportionality, which are $C_{w_{12}}$ and $C_{l_{12}}$, respectively.

Equation (14) and (15) will not always hold exactly due to noise. This noise can be due to the measurement noise and the noise from foreground segmentation errors. We model $W_b{}^2$ as Gaussian distributed with mean $C_{w_{12}} \times W_a{}^1$ and variance Var_w^{12}, and $L_b{}^2$ as Gaussian distributed with mean $C_{l_{12}} \times L_a{}^1$ and variance Var_l^{12}, or equivalently:

$$\Delta W_{ab}^{12} = (W_b{}^2 - C_{w_{12}} \times W_a{}^1) \sim N(0, Var_w{}^{12}) \tag{16}$$

$$\Delta L_{ab}^{12} = (L_b{}^2 - C_{l_{12}} \times L_a{}^1) \sim N(0, Var_l^{12}) \tag{17}$$

We define a vector $\mathbf{O}_{ab}^{12}$ for a target a which exits the FOV of camera 1 at t_a and has newly appeared as target b in camera 2 at time t_b as

$$\mathbf{O}_{ab}^{12} \triangleq [\Delta X_{ab}^{12}, \Delta Y_{ab}^{12}, \Delta W_{ab}^{12}, \Delta L_{ab}^{12}] \tag{18}$$

where

$$\Delta X_{ab}{}^{12} \triangleq X_b{}^2 - X_a{}^1 \tag{19}$$

$$\Delta Y_{ab}{}^{12} \triangleq Y_b{}^2 - Y_a{}^1 \tag{20}$$

We define a match measure M in terms of likelihoods to determine if a vector $\mathbf{O}_{ab}^{12}$ from targets a and b in different camera views are of the same target as follows:

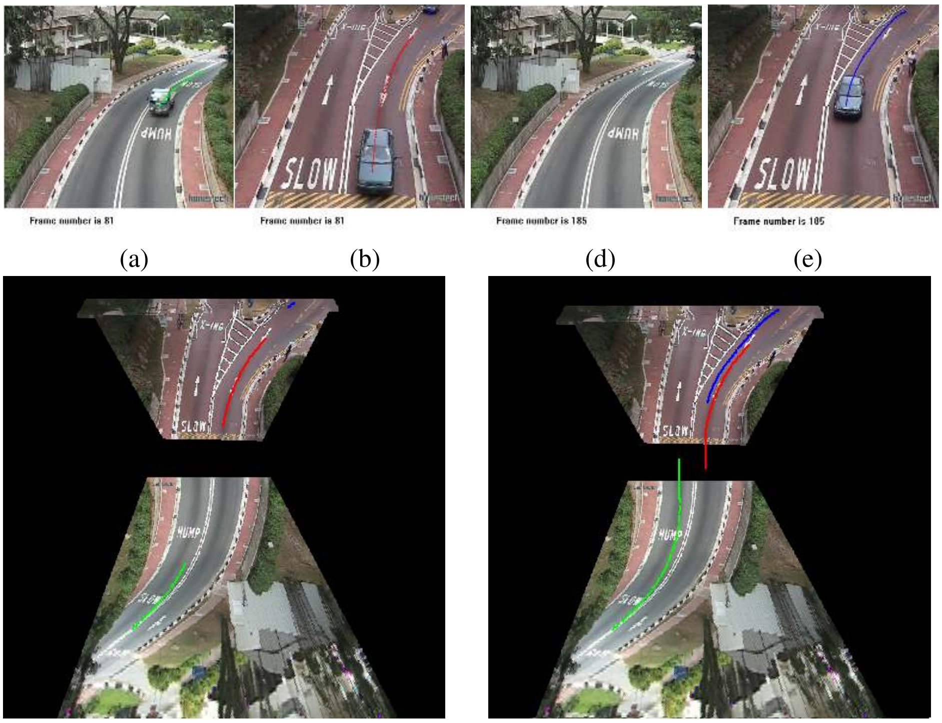
$$M(a = b/\mathbf{O}_{ab}^{12}) \triangleq |log_2\{\mathscr{L}_x(\Delta X_{ab}^{12})\mathscr{L}_y(\Delta Y_{ab}^{12})\mathscr{L}_w(\Delta W_{ab}^{12})\mathscr{L}_l(\Delta L_{ab}^{12})\}| \tag{21}$$

where $\Delta X_{ab}^{12} \sim N(V_{x_a} \times (t_b - t_a), Var_{x_{ab}}^{12})$, $\Delta Y_{ab}^{12} \sim N(V_{y_a} \times (t_b - t_a), Var_{y_{ab}}^{12})$, and $\Delta W_{ab}{}^{12}$ and $\Delta L_{ab}{}^{12}$ are distributed as in (16) and (17), respectively.

Our interest is to determine if the target entering the FOV of one camera is the same as the one that exited the FOV of another camera. Thus we calculate M for a new object that enters the FOV of any camera in the ground plane view. We use the observations of the targets at the entry/exit locations to get the observation vectors. By using a suitable threshold on M, the correspondence matching problem between the cameras can be solved. The number of correct matches in the multi-camera tracking system largely depends on the threshold selected for the value of M in (21), obtained using training data. In our system we have used M = 350.

5 Tracking results

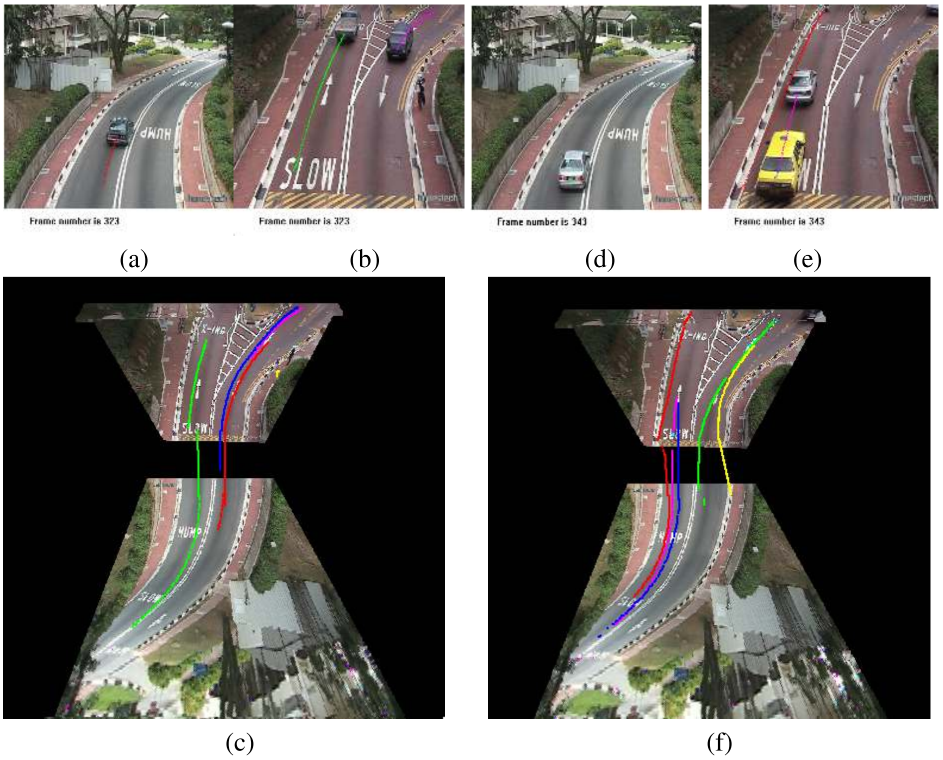
The proposed system is implemented and tested for a two-camera system monitoring traffic from two sides of an over-bridge. Figure 3 shows the FOVs of two cameras in (a), (b) and (d), (e) and the derived ground plane views in (c) and (f). The latter also show the tracking results. Figure 4 shows another example of target tracking results. The track for carrier van in Figure 3(a) is shown. It has been correctly tracked in the blind region across the camera views and correct correspondence matching has been achieved in spite of significant change in target size and color information due to change in camera angle. The figures show successful tracking of another car and van. Even though the roads in Figures 3(a) and 3(b) are sloping downwards the tracking results are still good. Our system is a real time system as it processes 12 frames/second (for each camera, frame size is 352 × 288) for a two camera setup on a Pentium 4 3.06 GHz machine. The success rate of our system for correspondence matching is 85%.



(a) (b) (d) (e)

(c) (f)

Figure 3: (a) and (d) are the views from camera 1. (b) and (e) are the views from camera 2. (c) and (f) shows the ground plane view constructed using the homography transform. The vehicles in the FOVs of the cameras are being tracked in the ground plane view (The vehicles are not shown in the ground plane view since it is computationally intensive to calculate the ground plane view for every frame). The trajectories of the car and truck are assigned tracks as seen in (c). When the car and truck exit the FOVs of the cameras, their path in the blind region is tracked as seen in (f).



(a) (b) (d) (e)

(c) (f)

Figure 4: (c) shows the tracking results after the correspondence matching is done using the proposed Gaussian formulation for the car and truck in Figure 3. The Kalman filter tracks the vehicles in the blind region between the cameras. (f) shows some more tracking results.

6 Conclusion

In this paper we have proposed a solution for multi-camera correspondence matching and tracking of targets in cameras with non-overlapping FOVs. For this, a robust graph based single camera target track initialization algorithm was proposed which gathers information over multiple frames before matching the targets across cameras. The parameters for correspondence matching were obtained by tracking the shape of the target in the camera view and motion of the targets in the common ground plane view. The common ground plane view was obtained by computing the homography of each camera view with the ground plane. Shape and motion Kalman filters were used to track the targets in the individual camera FOVs while a motion Kalman filter was used to track the targets in the ground plane view. The change in position and shape parameters of the same target across cameras were modeled as Gaussian distributions and we used the latter to compute the likelihoods for correspondence matching. When the target exits the FOV of a camera, the motion Kalman filter continues to track the target in the blind region. In [2] the solution is specifically for a system of two widely separated cameras on a highway. Their main goal was to achieve the correspondence matching of the vehicles on the highway and the movement of the vehicles in the blind regions between the cameras was not of interest. Target tracking in the blind region has applications in traffic surveillance systems to predict the most likely position of a vehicle in the blind region of the surveillance cameras. Also, the complete path of a vehicle, including its motion in the blind regions can be visually represented for a network of non-overlapping cameras.

The computations in target matching across cameras can be significantly reduced by using the knowledge of camera topology. Camera topology can be computed using methods as in [1, 11]. The robustness of correspondence matching across camera views can be further improved by using color calibration of the cameras and then using the color information along with shape and motion to match the targets across cameras.

References

[1] T.J. Ellis, D. Makris, and J. Black. Learning a multicamera topology. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 165–171, Nice, France, October 2003.

[2] T. Huang and S. Russell. Object identification: A bayesian analysis with application to traffic surveillance. *Artificial Intelligence*, 103:1–17, 1998.

[3] O. Javed, Z. Rasheed, O. Alatas, and M. Shah. Knight: A real time surveillance system for multiple overlapping and non-overlapping cameras. In *The fourth International Conference on Multimedia and Expo*, Baltimore, Maryland, 2003.

[4] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. In *The Ninth IEEE International Conference on Computer Vision*, Nice, France, 2003.

[5] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. In *Computer Vision and Pattern Recognition-Volume 2*, pages 2253–2258, Fort Collins, Colorado, June 1999.

[6] V. Kettnaker and R. Zabih. Counting people from multiple cameras. In *IEEE International Conference on Multimedia Computing and Systems Volume II-Volume 2*, pages 267–272, Florence, Italy, June 1999.

[7] S. Khan, O. Javed, Z. Rasheed, and M. Shah. Human tracking in multiple cameras. In *The Eighth IEEE International Conference on Computer Vision*, Vancouver, Canada, July 2001.

[8] P. Kumar, S. Ranganath, K. Sengupta, and H. Weimin. Co-operative multi-target tracking and classification. In *Proceedings of European Conference on Computer Vision, Prague*, Prague, Czech Republic, May 2004.

[9] D. Makris, T.J. Ellis, and J. Black. Bridging the gaps between cameras. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2004*, Washington DC, USA, June 2004.

[10] F. Porikli and A. Divakaran. Multi-camera calibration, object tracking and query generation. In *The fourth International Conference on Multimedia and Expo*, Baltimore, Maryland, 2003.

[11] C. Stauffer and K. Tieu. Automated multi-camera planar tracking correspondence modeling. In *Proc. Computer Vision and Pattern Recognition*, pages 259–266, July 2003.

[12] J.K. Wolf, A.M. Viterbi, and G.S. Dixon. Finding the best set of k paths through a trellis with application to multi-target tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 287–295, 1989.