# Co-operative Multi-target Tracking and Classification

Pankaj Kumar[1], Surendra Ranganath[2], Kuntal Sengupta[3], and Huang Weimin[1]

[1] Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613,
kumar@i2r.astar.edu.sg, wmhuang@i2r.a-star.edu.sg,
[2] National University of Singapore, 4 Engineering Drive 3 Singapore 117576,
elsesr@nus.edu.sg
[3] AuthenTec, Inc. Post Office Box 2719, Melbourne, Florida 32902-2719
kuntal.sengupta@authentec.com

**Abstract.** This paper describes a real-time system for multi-target tracking and classification in image sequences from a single stationary camera. Several targets can be tracked simultaneously in spite of splits and merges amongst the foreground objects and presence of clutter in the segmentation results. In results we show tracking of upto 17 targets simultaneously. The algorithm combines Kalman filter-based motion and shape tracking with an efficient pattern matching algorithm. The latter facilitates the use of a dynamic programming strategy to efficiently solve the data association problem in presence of multiple splits and merges. The system is fully automatic and requires no manual input of any kind for initialization of tracking. The initialization for tracking is done using attributed graphs. The algorithm gives stable and noise free track initialization. The image based tracking results are used as inputs to a Bayesian network based classifier to classify the targets into different categories. After classification a simple 3D model for each class is used along with camera calibration to obtain 3D tracking results for the targets. We present results on a large number of real world image sequences, and accurate 3D tracking results compared with the readings from the speedometer of the vehicle. The complete tracking system including segmentation of moving targets works at about 25Hz for 352×288 resolution color images on a 2.8 GHz pentium-4 desktop.

## 1 Introduction

This paper address several problems of tracking and classifying multiple targets in real-time, which can be used for behavior analysis of the moving targets. Several new ideas have been developed to solve the problem of Multi-Target Tracking (MTT) in 3D under the following assumptions: 1. Image sequences are obtained from a single stationary camera looking down into the scene. 2. The targets are moving on a ground plane and some 3D measurements on the ground and their corresponding locations in the image are available for camera calibration. In this paper the problem of MTT is formulated as an optimal feature estimation and data association problem, which has been the usual paradigm

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3021, pp. 376–389, 2004.

for MTT in radar community [1][2]. To obtain good classification results target tracking has to be accurate. The knowledge of object type can improve the tracking results. In this paper these two ideas have been combined to get accurate classification and 3D tracking results. First the issue of efficiently handling splits and merges in the segmentation of the targets so that tracking can continue even when targets split and merge and there are large number of targets in the Field of View (FOV). Second a new target track initialization algorithm is introduced which ensures accurate and stable initialization of the trackers which is usually required by many tracking algorithms. The third contribution is a new co-operative tracking-classifying-tracking algorithm. The results of image based 2D tracking are used to classify the target into different categories using a Bayesian network. This allows using a representative 3D model for each category to compute the 3D tracking results of the targets.

The paper is organized as as follows: Related works and their differences to our work is discussed in Section 2. In Section 3 target modelling and pattern matching algorithm which facilitates the use of a dynamic programming (DP) strategy to efficiently compute the data association of targets in the presence of multiple splits and merges is discussed. Section 4 briefly discusses Kalman filter based motion and shape tracking. The new algorithm for automatic initialization of target tracking is discussed in Section 5. In Section 6 target classification based on Bayesian network is discussed. Section 7 explains the 3D models of the different classes and the camera calibration method used to obtain 3D tracking results from 2D tracking. Finally results and conclusions are presented in Sections 8 and 9.

## 2 Related Work

Paragios and Deriche [3] considered the problem of simultaneously tracking several non-rigid targets. The motion parameters of the targets were estimated using a coupled front propagation model, which integrates boundary and region-based information. McKenna *et al.'s* [4] work on tracking groups of people performs tracking at three levels of abstraction: regions, people, and groups. People are tracked through mutual occlusions as they form groups and separate from one another. Strong use of color information is made to disambiguate occlusion and to provide qualitative estimates of depth ordering and position during occlusion. Javed and Shah [5] presented an automated surveillance system where the objects were tracked and classified into different categories with a new feature, "Recurrent Motion Image"(RMI). The tracking discussed in [5] is based on region correspondence matching which may fail when there are large number of similar targets undergoing merges and splits. Haritaoglu *et al.* [6][7] proposed a system that combines shape analysis and statistical techniques to track people and their parts in an outdoor environment. To handle interactions amongst the tracked people, they used a generic human model tuned to each target's specific details to resolve the ambiguities. To track objects in $2\frac{1}{2}$D they used stereo camera. In our approach 3D tracking results have been obtained using a single camera.

Medioni *et.al.* [8] proposed an approach similar to Reid's Multiple Hypothesis Tracking (MHT) [9][10]. They use attributed graph matching for creating new target tracks and tracking them. This approach is more advanced than MHT as it can handle cases where a target gives rise to multiple measurements because of splitting. The solution for MTT proposed in this paper can simultaneously handle both splitting and merging of targets in colored images. Tao *et al.* in [11] proposed dynamic motion layer based approach for tracking persons and vehicles in image sequences. Initialization of the tracker relies on blob detection. The system runs at 5 Hz for four moving objects in the scene. They show tracking results for 4 to 5 objects in the FOV. We show tracking results for 10-17 targets in the FOV at 25 Hz.

## 3 Feature Extraction and Pattern Matching

We use our active background modelling and foreground segmentation scheme to segment moving foreground objects in the Field of View (FOV) [12]. Another foreground segmentation technique which can be used is [13]. The foreground regions are enclosed within their convex hulls to remove concavities. If there are small connected regions lying within the convex hull of a larger connected region then the smaller regions are ignored and only the larger region is considered. The convex hulls are approximated by an ellipse using the algorithm given in [14]. The measurements obtained for each foreground region in a frame is called a Segmented Patch ($SP$) and its features are:

1. Centroid of the ellipse, $Xc$.
2. $J$ angularly equidistant control points $X1, X2, ..., XJ$ on the ellipse.
3. The normalized, $I$ bin histograms of the $Y, Cr, Cb$ channels of the $SP$, $H1, H2, ..., HI$.

The $o^{th}$ $SP$ of a frame is represented as:

$$C^o = c^o_{Xc}, c^o_{X1}, c^o_{X2}, ..., c^o_{XJ}, c^o_{H1}, c^o_{H2}, ..., c^o_{HI}. \tag{1}$$

The targets being tracked by the Kalman filter have same representation as the measurements with some extra features like velocity of the centroid $b^n_{Vc}$ and a parameter to measure change of shape $b^n_s$ . The $n^{th}$ target and its features are represented as:

$$B^n = b^n_{Xc}, b^n_{X1}, b^n_{X2}, ..., b^n_{XJ}, b^n_{H1}, b^n_{H2}, ..., b^n_{HI}, b^n_{Vc}, b^n_s \tag{2}$$

### 3.1 Match Measures

Three match measures $D_S$, $D_X$ and $D_H$ are discussed here for matching targets with $SPs$ based on shape and color information. The control points of the $n^{th}$ target $B^n$ are $b^n_{X1}, b^n_{X2}, ..., b^n_{XJ}$. These control points form polygon $Poly_{B^n}$ and enclose area $A_{B^n}$. Similarly, the control points of the $o^{th}$ $SP$, $C^O$ form polygon $Poly_{C^o}$ and enclose area $A_{C^o}$. $(A_{B^n} \bigcap A_{C^o})$ is the common area between

the polygons $Poly_{B^n}$ and $Poly_{C^o}$. The match measures $D_S$ and $D_X$ used for matching shape of $SP\ C^o$ with target $B^n$ are defined as:

$$D_S(C^o, B^n) \triangleq \frac{\sum_{j=1}^{J} d_s(c^o_{Xj}, Poly_{B^n})^2}{\{A_{B^n} + A_{C^o}\}} \tag{3}$$

$$d_s(c^o_{Xj}, Poly_{B^n}) \triangleq \textit{Shortest distance of } c^o_{Xj} \textit{ from polygon } Poly_{B^n}.$$

The sum of area term in the denominator is to normalize the match measure with respect to area of the patterns.

$$D_X(C^o, B^n) \triangleq \frac{\sum_{j=1}^{J} d_x(c^o_{Xj}, Poly_{B^n})^2}{\{A_{B^n} + A_{C^o}\} \times F} \tag{4}$$

$$d_x \triangleq \begin{cases} 0 & \textit{If } c^o_{Xj} \textit{ is within polygon } Poly_{B^n} \textit{ otherwise} \\ \textit{shortest distance of } c^o_{Xj} \textit{ from the polygon } Poly_{B^n}. \end{cases}$$

$$F \triangleq \begin{cases} 0 & \textit{If } A_{B^n} \bigcap A_{C^o} = 0. \\ 1 & \textit{If } A_{B^n} \bigcap A_{C^o} > 0. \end{cases}$$

The computation of $d_s$ and $d_x$ is explained in Figure 1. $D_S$ is a simple shape matching measure mentioned here for the purpose of comparing the matching results with the new match measure $D_X$. Some of the properties of $D_X$ are:

**1.** Non-negative: $D_X(C^o, B^n) \geq 0$.
**2.** Non-symmetric: $D_X(C^o, B^n) \neq D_X(B^n, C^o)$.
**3.** If $D_X(A, B) = 0$ & $D_X(B, C) = 0$ $=> D_X(A, C) = 0$.
**4.** But $D_X(A, C) = 0$ & $D_X(B, C) = 0$ $\neq> D_X(A, B) = 0$.

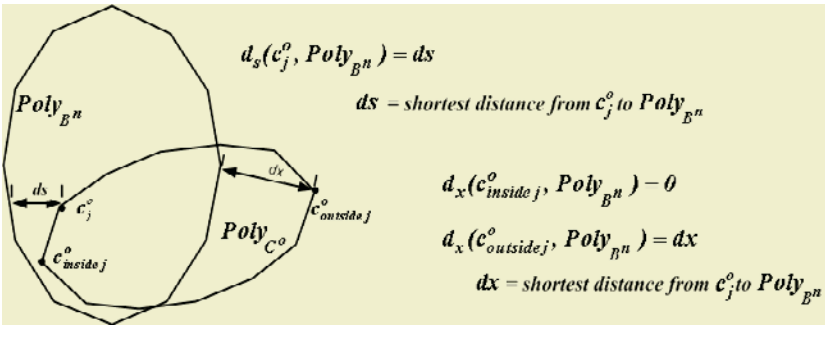



**Fig. 1.** This figure explains the computation of distance $d_x$ and $d_s$ of a control point on $Poly_{C^o}$ with $Poly_{B^n}$.

A $SP$, $C^o$ is a match with target $B^n$ when the match measure $D_X(C^o, B^n)$ is equal to zero. This happens when $C^o$ is spatially coincident with target $B^n$ or $C^o$ lies entirely within $B^n$. However, in practice a $C^o$ would be considered a match with $B^n$ when $D_X(C^o, B^n)$ is smaller than a threshold, which is not critical. The presence of $F$ term in the denominator of (4) ensures that the two contours of $C^o$ and $B^n$ match only when there is overlap between them.

The match measure $D_H$ is defined as

$$D_H(C^o, B^n) \triangleq \sum_{i=1}^{I} |c^o_{Hi} - b^n_{Hi}| \tag{5}$$

for matching two patterns based on intensity and color information. Each bin $Hi$ has three sub-bins corresponding to $Y, C_r$, and $C_b$ channels and $|c^o_{Hi} - b^n_{Hi}|$ is the sum of the absolute differences of the sub-bins in bin $Hi$.

### 3.2 Pattern Matching

Here we consider matching targets with their $SP$ when there is merging and splitting. Let the set of targets being tracked be represented by $B^1, B^2, \ldots, B^n, \ldots, B^N$. A frame consist of several $SPs$, which are $C^1, C^2, \ldots, C^o, \ldots, C^O$. A $SP$ $C^o$ can be from: 1. single target, 2. multiple targets merging together, 3. part of a target, which has split into multiple $SPs$, and 4. part of a target which has simultaneously merged with other targets and undergone split to give the $SP$. We focus on solving data association in cases 2 and 3 optimally and 4 is solved sub-optimally.

The merging of two or more targets is expressed with operator $\oplus$, i.e. $B^1 \oplus B^2 \oplus B^3 \oplus B^4$ denotes the merging of the 4 targets $B^1, B^2, B^3$, and $B^4$. The synthesized-pattern $\overline{B}$ formed by merging targets $B^1, B^2, B^3$, and $B^4$ will have a new convex hull. This convex hull of $\overline{B}$ is obtained from the points on the convex hull of $B^1, B^2, B^3$, and $B^4$. Given $N$ targets, the total number of different ways in which a new synthesized-pattern can be formed is $2^N - 1$. For example, for $N = 4$ the different possibilities for $\overline{B}$ are $\{B^1, B^2, B^3, B^4, B^1 \oplus B^2, B^1 \oplus B^3, B^1 \oplus B^4, B^2 \oplus B^3, B^2 \oplus B^4, B^3 \oplus B^4, B^1 \oplus B^2 \oplus B^3, B^1 \oplus B^2 \oplus B^4, B^1 \oplus B^3 \oplus B^4, B^2 \oplus B^3 \oplus B^4, B^1 \oplus B^2 \oplus B^3 \oplus B^4\}$. Notationally any possible synthesized-pattern $\overline{B}$ formed from merges can be written as

$$\overline{B} = B^{n(1)} \oplus B^{n(2)} \oplus \ldots \oplus B^{n(p)} \oplus \ldots \oplus B^{n(P)}. \tag{6}$$

Where $P$ of the available targets have merged and $n(p)$ denotes the index of the targets used in synthesis of $\overline{B}$. For each $\overline{B}$ the match measure $D_S(\overline{B}, C^o)$ can be computed to find the best match such that

$$\hat{P}, \hat{n}(p) = arg \min_{P, n(1), \ldots, n(p)} [D_S(\overline{B}, C^o)]. \tag{7}$$

This formulation considers all the possible merges of targets and the one which gives minimum match measure is finally chosen, thus giving an optimal solution. However, the minimization problem expressed in (7) is computationally prohibitive to solve in real time. The order of computation to find optimal match when targets merge is $O((2^N - 1) \times O)$. For $N = 12$ *to* 17 this would be quite a large number.

The optimal solution to the problem of splitting can be addressed by merging the $SPs$, $C^o$ in all possible combinations and computing their match measure

with the different targets. The new synthesized pattern formed by merging different $SPs$ is $\overline{C} = C^{o(1)} \oplus C^{o(2)} \oplus \ldots \oplus C^{o(p)} \oplus \ldots \oplus C^{o(P)}$. The problem is to compute all possible $\overline{C}$ by changing $P$, the number of $SP$ and $o(p)$ the indexes of $SP$. Thus the problem can be optimally solved as in the case of merges with a computational complexity of $O((2^O - 1) \times N)$. Some times the total number of $SP$ can be quite large due to the presence of clutter, which would make $2^O$ a very large number. The total complexity to handle both splits and merges is $O((2^N - 1) \times O) + O((2^O - 1) \times N)$. Next we show how by using the new match measure $D_X$ and dynamic programming the same problem can be solved in $O(N \times O)$.

### 3.3 Dynamic Programming Strategy for Efficient Pattern Matching

Dynamic Programming (DP) is a powerful nonlinear optimization technique, and is used here to solve the pattern matching problem by optimizing a function which evaluates the match between targets and $SPs$. The use of DP in solving a problem requires that the problem be divided into sub-problems and optimal solution of these sub-problems can be combined together to obtain optimal solution for the main problem. The properties of the distance function $D_X$ facilitates the use of the DP strategy. In Figure 2 four targets $B^{n(1)}, B^{n(2)}, B^{n(3)}$, and $B^{n(4)}$ at time instance $k$ are being tracked with a Kalman filter based tracker. The predictions for their shape and position are available for the next frame $k+1$ and are denoted as, $\hat{B}^{n(1)}, \hat{B}^{n(2)}, \hat{B}^{n(3)}$, and $\hat{B}^{n(4)}$. These targets merge to give

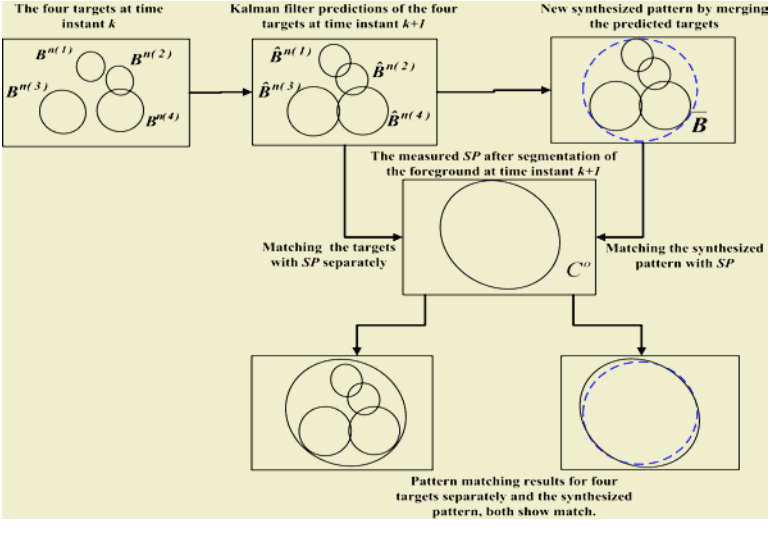



**Fig. 2.** This figure explains the pattern matching principle which enables use of dynamic programming strategy to speed up computations. A synthesized pattern $\overline{B} = \hat{B}^{n(1)} \oplus \hat{B}^{n(2)} \oplus \hat{B}^{n(3)} \oplus \hat{B}^{n(4)}$ is a match with $SP$ $C^o$ by the distance measure $D_X$ when $\hat{B}^{n(1)}$, $\hat{B}^{n(2)}, \hat{B}^{n(3)}$, and $\hat{B}^{n(4)}$ all match with $C^o$ separately.

rise to a new synthesized pattern $\overline{B}$ in frame $k+1$. If a $SP$ $C^o$ is due to the merger of these four targets in frame $k+1$ then $\overline{B}$ will coincide with $C^o$ and hence $D_X(\overline{B}, C^o)$ will be equal to zero. Now if $D_X$ of $\hat{B}^{n(1)}, \hat{B}^{n(2)}, \hat{B}^{n(3)}$, and $\hat{B}^{n(4)}$ is computed with $C^o$ separately then each one of them will be equal to zero. Because all the control points of these targets lie within the polygon formed by the control points of $\overline{B}$ which is a match with $C^o$. Therefore the problem of pattern matching when targets undergo merge to form a new synthesized pattern $\overline{B}$ is sub-divided to the problem of finding all targets which match $SP$, $C^o$ separately by the match measure $D_X$. If the predictions of all the targets $B^{n(1)}, B^{n(2)}, \ldots, B^{n(P)}$ match with $C^o$ separately then $\overline{B}$ formed by merging these targets is optimal match for $C^o$.

When the targets undergo splitting to give rise to more $SPs$ then the targets in the scene then the same DP strategy as above can be applied by reversing the roles of $B^n$ and $C^o$.

## 4 Kalman Filter Based Tracking

The system uses two Kalman filters to track each target. Theoretically our system can be classified as a multiple model system [15]. The motion an position of a target is tracked by tracking the centroid of the ellipse modelling the target. Assuming that video rate sampling of 25 frame/sec is fast enough we model the motion of the targets with a constant velocity model. The shape of the target is tracked by tracking the $J$ control points representing the shape of the target. The motion of these control point are approximated by affine motion model, which has been widely used in computer vision for segmentation and tracking [16][17]. The change of shape of the targets as they move away or towards the camera is accounted by the parameter $b_s^n$ of the $n^{th}$ target. The details of the Kalman filter equations and their derivation can be obtained from [18].

From the discussion in Section 3 it can said that there are three types of matches possible. Each of these and their different methods for updating the filter parameters are:

1. The targets which have not under gone merge or splits, match their corresponding $SP$ with match measures $D_H$ and $D_X$. The motion, position, and shape attributes of these targets are updated by the Kalman Filter *estimates*.
2. For matches where a target has split into multiple $SPs$ the shape feature of the target is updated by Kalman Filter *predictions* but position and motion are updated by Kalman Filter *estimates*. In the latter case the new measurement of the centroid is the mean of all centroids of the different $SPs$, which match the target.
3. The shape, position and motion features of targets, which have merged or have undergone simultaneous merges and splits are updated by their Kalman Filter *predictions* for all attributes motion, position, and shape.

The result of tracking as a co-operative effort between pattern matching and Kalman filter based tracking are shown on a test image sequence in Figure 3.

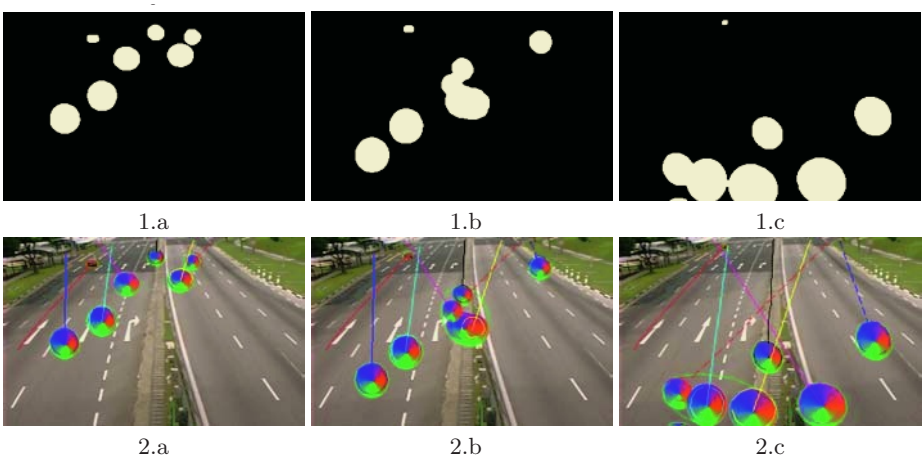



**Fig. 3.** Images 1.a,b,c,show the segmentation results for eight targets in the FOV and images 2.a,b,c show the tracking results. Images 1.b and 2.b show a case where four targets were merged into one $SP$. Here each of the eight targets were properly tracked even as they underwent multiple merges and splits. Please note here that all the targets are similarly colored so a correspondence based tracking is likely to fail.

These images show the algorithm's ability to handle multiple merges. The video was made by overlaying artificial targets on a real video. In spite of many instances of merges the targets position and shape have been quite accurately tracked.

## 5 Track Initialization

Accurate initialization of the position, motion, and shape parameters of a target is an important step, which must be accomplished in the presence of clutter. There are two types of track initializations that needs to be handled: 1. the initial bootstrapping and 2. when the tracking is in progress.

Attributed graphs are very useful for initializing Multi-Target Tracking (MTT) systems as it provides a technique for incorporating both spatial and temporal information of the targets in decision making. Graphs for target track initialization and tracking have been used in [19] and [8], respectively. Our system initializes a new target for tracking only when a target's reliable measurements are available in the past $\tau$ frames. This property makes the initialization accurate and the tracker stable.

Automatic initialization of target tracks is done by using an attributed graph of the $SPs$ in $\tau$ frames, as shown in Figure 4. The attributes of each node in the graph are: 1. the frame number, 2. the centroid, 4. shape parameters, 5. color histogram of the $SP$, 6. parent id and child id. Edges are present between nodes whose frame number differ by 1 as shown in Figure 4. The weights of these

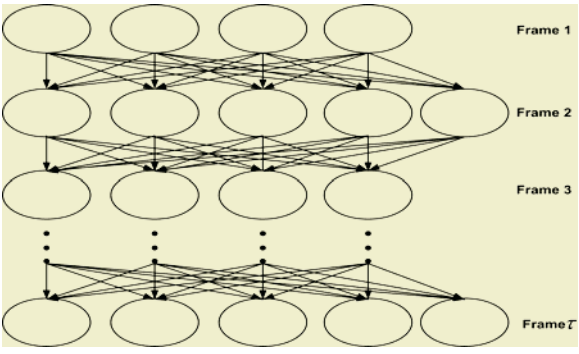


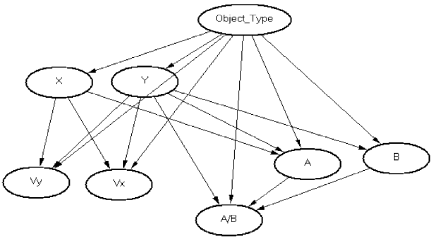

**Fig. 4.** This figure shows the structure of the attributed graph used for initialization of tracking.

edges is the weighted sum of match measures $D_H$ and $D_S$ between the nodes. The nodes with frame number 1 are considered as source nodes and nodes with frame number $\tau$ are considered as destination nodes. For all source nodes the shortest path 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 path with smallest sum-of-weights is chosen and is called the path of least sum-of-weights. This path is considered a valid target track. The nodes of this path is removed from the graph giving rise to a new graph with reduced number of nodes and edges. The same process is repeated for the new graph until there is no node from any one of the $\tau$ 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.

Another problem is initialization of tracking for new targets, which enter the FOV or appear in the FOV due to resolution of occlusion, when tracking of other targets are in progress. To solve this problem an attributed graph of $SPs$ 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 $(k - \tau + 1)$, where $k$ is the current frame number. The destination nodes are the unmatched $SPs$ of frame $k$. A new target is confirmed by appearance of least sum-of-weights path amongst all the shortest path possible from the source nodes to the destination nodes. All the nodes of this path are removed from the attributed graph.

## 6 Bayesian Network Based Classifier

Bayesian network classifiers provide a probabilistic framework, which allow the power of statistical inference and learning to be combined with the temporal and contextual knowledge of the problem [20]. We have used Bayesian network for classification of targets in image sequences from a stationary camera. The




(a)

| Object | Total Instances | Correct Classification |
|---|---|---|
| Pedestrian | 63 | 86.7% |
| Motor bike | 105 | 92.5% |
| Cars | 695 | 96.3% |
| Truck | 92 | 93.3% |
| Heavy Truck | 45 | 93.6% |
| Clutter | 163 | 88.6% |

(b)

**Fig. 5.** (a) Bayesian Network structure used for target classification and (b) the classification results obtained from this classifier.

different classes of targets considered are pedestrians, motorcycles, cars/vans, trucks/buses, heavy trucks, and clutter. In general it is difficult to model a deterministic relationship between the size, shape, position, and motion features to the object class due to perspective effects. For example a car close to the camera may be of the same size as a truck far from the camera; similarly a pedestrian passing close to the camera may show motion in image space which is similar to the motion of a fast moving car far from the camera. Furthermore there are internal dependencies amongst the features. For example, the speed and size of an object is dependent upon its position. Thus, to establish a relationship between the various image features of a target and its type, and to model the conditional dependencies amongst the features we use a Bayesian Network based classifier. The use of motion and position parameters of a target from tracking module makes the classification more robust. For example the size and shape of a moving motorcycle and a pedestrian may be similar but their motion and position are usually different.

Figure 5(a) shows the proposed Bayesian Network model. Each node is a variable and the object node is the root node. The seven measurement nodes are $X$, $Y$ (the $x, y$ co-ordinates of the target in image space), $V_x$, $V_y$ (the $x, y$ components of the target velocity in image space), $A$, $B$, the major and minor axis of the ellipse modelling the target's shape and $A/B$ the aspect ratio of the ellipse. Figure 5(b) shows the classification results from several image sequences. In each of these case the ground truth was manually obtained.

## 7 Camera Calibration and 3D Tracking

To convert the tracking results in image co-ordinate space to world co-ordinate space we need to know the perspective transformation matrix $P$. We use the technique similar to that of [21] to compute $P$. The $XY$ plane of the world co-ordinate system is aligned with the ground plane of the scene and the $Z$ axis is perpendicular to the ground plane. The image co-ordinates are related to the world co-ordinate as follows:

$$\begin{bmatrix} x_i \\ y_i \\ \lambda \end{bmatrix} = \begin{bmatrix} p_{11}\ p_{12}\ p_{13}\ p_{14} \\ p_{21}\ p_{22}\ p_{23}\ p_{24} \\ p_{31}\ p_{32}\ p_{33}\ p_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{8}$$

From (8) it can be easily shown that if the height of a point $Z_w$ in the world co-ordinates is known along with its image co-ordinates, then the corresponding world co-ordinate location $X_w,\ Y_w$ of the point can be obtained as:

$$X_w = \frac{(p_{12} - p_{32}x_i)(p_{23} - p_{33}y_i) + (p_{13} - p_{33}y_i)(p_{32}y_i - p_{22})}{(p_{32}y_i - p_{22})(p_{31}x_i - p_{11}) - (p_{12} - p_{32}x_i)(p_{21} - p_{31}y_i)} Z_w \tag{9}$$
$$+ \frac{(p_{12} - p_{32}x_i)(p_{24} - p_{34}y_i) + (p_{24} - p_{34}y_i)(p_{32}y_i - p_{22})}{(p_{32}y_i - p_{22})(p_{31}x_i - p_{11}) - (p_{12} - p_{32}x_i)(p_{21} - p_{31}y_i)}$$

$$Y_w = \frac{(p_{21} - p_{31}y_i)X_w}{(p_{32}y_i - p_{22})} + \frac{(p_{23} - p_{33}y_i)Z_w}{(p_{32}y_i - p_{22})} + \frac{(p_{24} - p_{34}y_i)}{(p_{32}y_i - p_{22})} \tag{10}$$

After the targets are classified, a model height of targets in different categories as shown in Table 1 is used to estimate the 3D position and motion of the targets. The model heights are a rough guides to the height of a 3D point on top of the target. In simulations it was found that an error of $\pm 0.5$ *meters* in the height estimate translates to about $\pm 10\%$ error in speed estimate. To get an accurate estimate of the world speed of a target, a point which is on top of the target is selected using heuristics based on the camera view.

**Table 1.** Model height values for the different classes of targets obtained by averaging the different heights of objects in a class.

| Pedestrian | Motorcycles | Cars/Vans | Trucks/Buses | Heavy Truck/Double-Decker |
|---|---|---|---|---|
| 1.7m | 1.5m | 1.7m | 3.0m | 4.0m |

## 8 Results

We show the results of tracking for both articulated and non-articulated objects. In all the results the green ellipses are used to show the measurements obtained in every frame. The targets and their tracks are shown with same color, which is different for different targets as far as possible. In Figure 6 we show the robust tracking results for a real traffic scene and one of the image sequence of PETS data set. Complete tracking video for traffic scene can be seen in 'video.avi' in the supplementary files. Here simultaneous tracking of upto 17 targets can be seen. There are instances here when the targets were completely occluded, a target splitted in to more than one *SPs* and some targets merged to give one *SP* and there were lot of clutter. In all these cases tracking continued without errors and there was no wrong initialization.

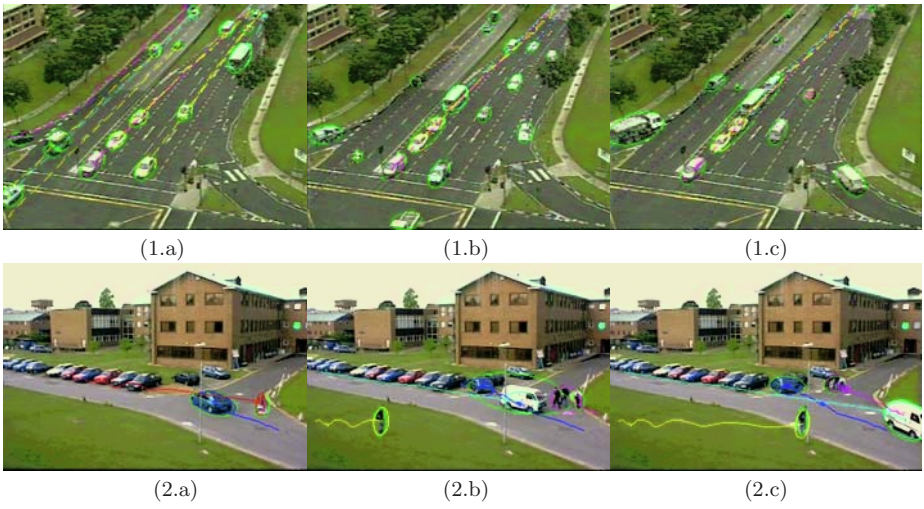



**Fig. 6.** Images 1.a,b,c show the tracking results for a traffic scene. (a) shows a case where the $2^{nd}$ target from bottom left split into two measurements, (b) shows a frame where 17 targets are being tracked simultaneously, and (c) shows a case where four vehicles merged into one measurement. Images 2. a,b,c shows the tracking results on an image sequence from PETS2001 data set. Image 2.b shows a case where an instance of simultaneous merging and splitting has been handled properly.

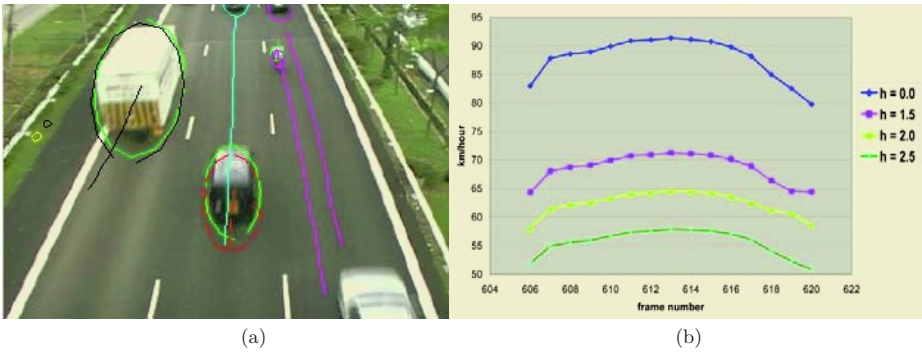



**Fig. 7.** Image (a) shows 2D tracking results of a frame in a test image sequence. The vans in the center of image (a) was moving with a constant speed of 65 km/hour as read from its speedometer. Plot (b) shows the computed speed of the vehicle for different height estimates denoted by the parameter 'h' and expressed in meters.

Figure 7 shows the results of 3D tracking algorithm proposed in the paper. Figure 7(a) shows the 2D tracking results and 7(b) shows the plot of the computed speed of the black van, being tracked in the center of the image 7(a), at different estimates of height. When the estimated height of the vehicle is taken to be zero then there is significant error in the speed estimates. The speed esti-

mates are in the range of 80-92 km/hour when the actual speed is 65km/hour as read from the speedometer of the vehicle. The speed estimates for other values of height, such as 1.5 meters, 2 meters, and 2.5 meters are close to the actual speed of 65km/hour. The actual height of the van is 2 meters. This accurate estimation of the speed of the target allows for detecting a vehicle's acceleration as well.

## 9 Conclusions

We have addressed several problems for robust and reliable tracking and and classification of multiple targets in image sequences from a stationary camera. A new efficient algorithm based on DP strategy for pattern matching was proposed, which can handle data association during complex splitting and merging of the targets. When this technique is combined with Kalman filter based tracking, it is possible to preserve the labels of the targets even when they cross each other, or get completely or partially occluded by background or foreground objects. An attributed graph based technique was proposed to initialize the tracks. Using a Bayesian network based classification and a simple camera calibration we have obtained accurate 3D tracking results for vehicles. Results have been shown where the tracker can handle up to 17 targets simultaneously. At present this system is being used to detect potential accident behavior between pedestrians and vehicles in traffic videos.

## References

1. Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Transactions on Automatic Control*, vol. Vol. AC-23, No. 4, pp. 618–626, August 1978.
2. S. Blackman, *Multiple-Target Tracking with Radar Application*. Artech House, 1986.
3. N. Paragios and R. Deriche, "Geodesic active regions for motion estimation and tracking," *IEEE Transactins on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 266–280, March 2000.
4. S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," *Computer Vision and Image Understanding*, vol. 80, pp. 42–56, 2000.
5. O. Javed and M. Shah, "Tracking and object classification for automated surveillance," in *European conference on Computer Vision*, 2002.
6. I. Haritaoglu, D. Harwood, and L. Davis, "W4: Who, when, where, what: A real time system for detecting and tracking people," in *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition (FG'98)*, pp. 222–227, April 1998.
7. I. Haritaoglu, D. Harwood, and L. Davis, "W4s: A real time system for detecting and tracking people in 2.5d," *Fifth European Conference on Computer Vision*, vol. June, pp. 877–892, 1998.
8. G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event detection and analysis from video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 873–889, August 2001.

9. D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. Vol. AC-24, No. 6, pp. 843–854, December 1979.
10. I. cox and M. Miller, "On finding rank assignments with application to multitarget tracking and motion correspondence," *Aerosys*, vol. Vol. 32, pp. 486–489, Jan. 1996.
11. H. Tao, H. S. Sawhney, and R. Kumar, "Object tracking with bayesian estimation of dynamic layer representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24 No. 1, pp. 75–89, 2002.
12. P. Kumar, S. Ranganath, and W. Huang, "Queue based fast background modelling and fast hysteresis thresholding for better foreground segmentation," in *Proccedings of The 2003 Joint Conference of the Fourth ICICS and PCM*, (Singapore), p. 2A2.5, December 2003.
13. L. Li, W. Huang, Y. G. Irene, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proccedings of ACM Multimedia*, pp. 2–10, November 2003.
14. A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476–480, May 1999.
15. E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey.," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 4, pp. 103–123, 1998.
16. R. Cipolla and A. Blake, "Surface orientation and time to contact from image divergence and deformation," in *Proccedings of Second European Conference on Computer Vision* (G. Sandini, ed.), (S. Margherita, Ligure, Italy), pp. 187–202, Springer-Verlag, Berlin, Heidelberg, New York, May 1992.
17. Q. Zheng and R. Chellappa, "Automatic feature point extraction and tracking in image sequences for unknown camera motion," in *Proccedings of International Conference on Computer Vision*, (Berlin, Germany), pp. 335–339, May 1993.
18. P. Kumar, "Multi-body tracking and behavior analysis." Ph.D. Thesis, National University of Singapre.
19. J. K. Wolf, A. M. Viterbi, and G. S. Dixon, "Finding the best set of k-paths through a trellis with application to multitarget tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 287–295, 1989.
20. A. Mittal and C. L. Fah, "Characterizing content using perceptual level features and context cues through dynamic bayesian framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. Special Issue on Conceptual and Dynamical Aspects of Multimedia Content Description., under submission.
21. K. Matsui, M. Iwase, M. Agata, T. Tanaka, and N. Onishi, "Soccer image sequence computed by a virtual camera," in *Proc. Conf. on Computer Vision and Pattern Recognition*, pp. 860–865, 1998.