Bayesian Network based Computer Vision Algorithm for Traffic Monitoring using Video

Pankaj Kumar[+] Surendra Ranganath[†] Huang Weimin[+]

E-mail: kumar@i2r.a-star.edu.sg E-mail: elesr@nus.edu.sg E-mail: wmhuang@i2r.a-star.edu.sg

[+] Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613

[†] Department of Electrical and Computer Engineering
National University of Singapore
4 Engineering Drive 3
Singapore 117576

Abstract

This paper presents a novel approach to estimating the 3D velocity of vehicles from video. Here we propose using a Bayesian Network to classify objects into pedestrians and different types of vehicles, using 2D features extracted from the video taken from a stationary camera. The classification allows us to estimate an approximate 3D model for the different classes. The height information is then used with the image co-ordinates of the object and the camera's perspective projection matrix to estimate the objects 3D world co-ordinates and hence its 3D velocity. Accurate velocity and acceleration estimates are both very useful parameters in traffic monitoring systems. We show results of highly accurate classification and measurement of vehicle's motion from real life traffic video streams.

1 Introduction

Traffic management and information systems at present rely on the technology of magnetic loop detectors for estimation of real time traffic parameters. Video based machine understanding provides an alternate, more economical and informative system for traffic monitoring. A video based traffic system provides a larger set of traffic parameters and furthermore cameras are easier to install than loop detectors. Recently there has been a significant amount of research for understanding activities of humans and vehicles in video imagery [6, 12, 19, 16, 17, 22]. Inherent in the problems of activity understanding and behavior analysis are the problems of target detection, tracking, and classification. Video traffic monitoring systems require robust and reliable classification of pedestrians and different types of vehicles, like motor-bike, cars, buses and trucks, and precise estimates of their 3D motion parameters. In this work our focus is on robust and reliable target classification and using knowledge of the target class to get associated 3D information, e.g. target height. The facilitates accurate estimation of the target's 3D motion.

Lipton *et al.* in [15] used a simple Maximum Likelihood Estimation (MLE) criterion and temporal consistency to classify targets into three classes: humans, vehicles, and others. The features used for classification were area and dispersedness of the target. Later, using the same features the classification algorithm was implemented using Multi-Layer Perceptron neural network in [11]. Medioni *et al.* in [17] classified objects into humans, vehicles, and noise based on image features like length, width, speed, and motion direction. In the paper by Javed and Shah [9] the classification of objects was based on Recurrence Motion Image (RMI) there classes were humans, groups of humans, vehicles, and others. We present a Bayesian Network (BN) based classification algorithm which classifies the objects into pedestrians, motor-bikes, cars, buses or trucks, heavy trucks, and noise based on six different parameters. Very high accuracy of classification has been shown by the new algorithm in different real life video streams. Knowledge of the object's class allows its height to be represented by an average value determined for that class. Using this height in conjunction with the object's 2D location allows inferring its 3D location and velocity, which is very useful for analyzing traffic behaviors.

The classification and 3D motion estimation system proposed in this paper has five main stages of processing as outlined in Figure 1. In the first stage the interesting foreground is segmented out from the background using a background substraction algorithm. The features of the segmented patches are used for tracking with a linear Kalman filter in the 2D image space. Above two stages are briefly discussed in Sections 2 and 3. Using the position, shape, and motion features of the target from the feature extraction and tracking algorithm the object is classified into different classes using a BN. The classification algorithm and the camera calibration technique are discussed in detail in Sections 4 and 5, respectively. In Section 6 we show the results of classification and 3D motion measurements and finally in Section 7 we give the conclusions for our work.

2 Background Foreground Segmentation

There have been primarily three techniques for segmentation of the moving foreground objects in video streams.

0-7803-8125-4/03/$17.00©2003 IEEE

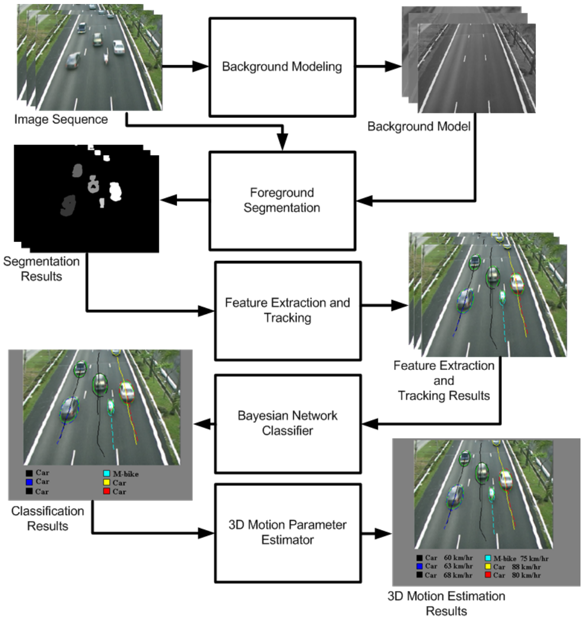


Figure 1. The different modules are shown along with the flow of data and information amongst the modules. The background modelling module models the background from the image sequence and foreground segmentation modules segments the moving targets from the images and feeds its results to the feature extraction and tracking module. The output of the feature extraction and tracking module is used by the classification module. The outputs of the classification and tracking module along with the results of camera calibration are used to compute the motion parameters of the objects in the world co-ordinate space.

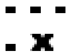


Figure 2. The dots are the 8-connected causal neighbors of the pixel **'x'** for a left to right and top to bottom raster scan.

Frame differencing as used in [2, 15], background subtraction as used in [6, 7, 9, 16] and optical flow as used in [5, 17]. Frame differencing do not yield good results when the objects are not sufficiently textured, and optical flow computations are very intensive and difficult to realize in real time. We propose a background substraction technique to segment the moving objects in image sequences. The technique is capable of modelling the background even in the presence of foreground objects, and update the model as new frames of the image sequence are obtained. We describe such a scheme where the background pixels are modelled with a single Gaussian distribution. This can be easily extended to model the background pixels with a mixture of Gaussians, if desired.

Let N frames of a color image sequence be used for modelling the background ($N = 200$). We use $YCrCb$ color space for background modelling because empirical results of [14] show $YCrCb$ to be the best for foreground segmentation and shadow suppression amongst the various standard color spaces. Let a pixel at image co-ordinate i, j and frame k be $\mathbf{p}_{ijk}$. Since each pixel $\mathbf{p}_{ijk}$ has three components $YCrCb$, their histograms are modelled by three Gaussians. We find the histograms $H_{ij_Y}(u)$ of the pixels of frames $k = \{1...N\}$, at each spatial location i, j and each color channel. The subscript Y is used to indicate that this histogram is of the intensity channel Y. The peak of each histogram is the intensity or chrominance value most frequently found at the corresponding pixel location in the corresponding channel and is thus expected to be the background. Using a window of width $2W$ centered on the mode for each histogram, we compute the mean and variance of the Gaussian distribution using the following equations:

$$\mu_{ij_Y} = \frac{1}{\sum_{u=(u^{max}_{ij_Y})-W}^{(u^{max}_{ij_Y})+W} H_{ij_Y}(u)} \sum_{u=(u^{max}_{ij_Y})-W}^{(u^{max}_{ij_Y})+W} u \times H_{ij_Y}(u) \tag{1}$$

$$\sigma_{ij_Y} = \frac{1}{\sum_{u=(u^{max}_{ij_Y})-W}^{(u^{max}_{ij_Y})+W} H_{ij_Y}(u)} \sum_{u=(u^{max}_{ij_Y})-W}^{(u^{max}_{ij_Y})+W} (u - \mu_{ij_Y})^2 \times H_{ij_Y}(u) \tag{2}$$

In our computations we use $W = 5$.

We use hysteresis thresholding to classify each pixel as being foreground, shadow, or background. The classification rule is as follows:

if *(any of the causal 8 connected neighbors of* $\mathbf{p}_{ijk}$ *as shown in Figure 2 is foreground)*
then *use the lower threshold in classification of the pixel.*
else *use the higher threshold.*

Each channel of each pixel $\mathbf{p}_{ijk}$ has its own thresholds obtained as a product of the corresponding standard deviation with a constant factor. The multiplying factor γ has a lower value for lower threshold and higher value for higher threshold for use in hysteresis thresholding. The thresholds are different for Y, and Cr, Cb, channels and are denoted as $\gamma_{Y_{bg}}$ and $\gamma_{C_{bg}}$ respectively. The work of Prati in [20] on shadow detection in HSV color space, have shown that luminance values of the shadow pixels are always less than the mean and they lie in a band with a lower and higher threshold. There is relatively less difference or negligible change in the chromacity channels Cr, Cb due to shadows. Therefore in our algorithm we use two thresholds $\gamma_{Y_{bg}}$ and $\gamma_{Y_{sh}}$ to detect the shadow pixels.

if $((|p_{ijk_Y} - \mu_{ij_Y}| < \gamma_{Y_{bg}} \times \sigma_{ij_Y})\&$
$(|p_{ijk_{Cb}} - \mu_{ij_{Cb}}| < \gamma_{C_{bg}} \times \sigma_{ij_{Cb}})$ &
$(|p_{ijk_{Cr}} - \mu_{ij_{Cr}}| < \gamma_{C_{bg}} \times \sigma_{ij_{Cr}}))$
then $\mathbf{p}_{ijk}$ *is background*
elseif $((\mu_{ij_y} - p_{ijk_y} > \gamma_{y_{bg}} \times \sigma_{ij_y})\&$
$(\mu_{ij_y} - p_{ijk_y} < \gamma_{y_{sh}} \times \sigma_{ij_y})$ &

```
$(|p_{ijk_{Cb}} - \mu_{ij_{Cb}}| < \gamma_{C_{bg}} \times \sigma_{ij_{Cb}})$ &
$(|p_{ijk_{Cr}} - \mu_{ij_{Cr}}| < \gamma_{C_{bg}} \times \sigma_{ij_{Cr}}))$
then $\mathbf{p}_{ijk}$ is shadow
else $\mathbf{p}_{ijk}$ is foreground.
```

3 Feature Extraction and Tracking

The foreground pixels are labelled into different regions using a M-connected component analysis. The convex hull of the M-connected foreground regions are approximated by an ellipse using the ellipse fitting algorithm of [3]. The centroid of the ellipse and twelve angularly equidistant control points on the perimeter of the ellipse are tracked using a linear Kalman Filter. Details of this algorithm can be found in [13]. Tracking the centroid of the ellipse gives the motion parameters of the object in the image co-ordinates. Tracking the control points on the ellipse, which is used to approximately model the foreground object, gives the size and shape features of the object in terms of the major and minor axis of the ellipse and the aspect ratio of the ellipse. Size, shape, position, and motion features of the target in the 2D image co-ordinate space is used to classify the object into different categories using a BN.

4 Bayesian Network based Classifier

BNs are useful for combining evidence in vision problems particularly where the information is diverse, dependent, both causal and diagnostic (deductive and abductive), and where the inference procedure is best posed in probabilistic terms [10, 21]. BN has been used in many applications such as audio-visual speaker detection [1], content based image and video indexing [18]. Huang *et al.* [8] used BN for automatic traffic scene analysis. We present a BN for classification of objects in video streams from a fixed camera. The different classes of objects are pedestrians, motor-bike (m-bike), cars, trucks/buses, heavy trucks, and noise. The camera is usually placed above the road and looking downward onto the traffic. In this situation, when there is perspective foreshortening, it is difficult to build a one to one relationship between the size, shape, position, and motion features to the object class. For example a car close to the camera may be of the same size as a truck further from the camera. Similarly a pedestrian passing by, 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 in the features themselves. For example the speed and size measure of an object is dependent upon its position. So to establish a relationship between the various image features of a target and its type and also to incorporate the conditional dependencies of the features within themselves we propose a new BN based classifier for inferring object type from measurements from each frame of the video streams.

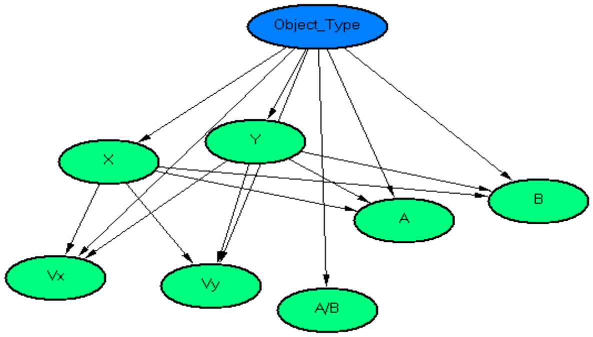


Figure 3. The network structure used for object classification. Here the velocity variable V_x and V_y and the size measures A and B are dependent on both object type and position of the object. The aspect ratio of the object is dependent only on the object type and not on the position of the object.

Figure 3 shows the BN used for the object classification problem. Each node is a variable and the object node is the hidden node. Here we use a supervised training approach, in which the classification node is identified and learning is optimized for classification performance. The seven measurement nodes are the X, Y (the xy coordinates of the object in the image space), V_x and V_y (the xy components of the targets motion in image space), A and B (the major and minor axis of the ellipse modelling the target) and A/B (the aspect ratio of the ellipse). An efficient inference algorithm is used to compute distribution over the object node given the measurements [10]. The arcs between the nodes are parameterized by conditional probability distributions that model dependencies between variables. The absence of arcs between nodes means that the variables are being treated independently. The network structure in Figure 3 has been manually specified using the knowledge of pin-hole camera model. The velocity measure of the object V_x and V_y is dependent upon both the object type variable and the image position of the object X and Y. Similarly the size of the object represented by A and B and is made dependent upon the position of the object and its type. It is not clear whether the aspect ratio, A/B, measured for a target be dependent on its position so we empirically compare the results of two BN structures, in one the variable A/B is dependent only on the object type and in the other, A/B is dependent both on object type and the position of the object. A better approach than this empirical verification would be to learn the network structure automatically from the data [4]. Structure learning algorithms accomplish this by searching over the space of network structures to find the structure, which is best supported by the data. This requires a scoring function for candidate structure and an

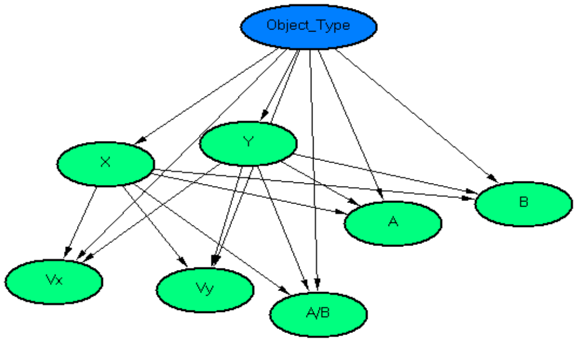


Figure 4. The network structure used for object classification. Here the velocity variable V_x and V_y and the size measures A and B are dependent on both object type and position of the object. The aspect ratio of the object is also dependent both the object type and object position.

efficient search procedure, since the space of all topologies is intractably large for even a small number of nodes. In future development of our work we will look into network structure learning algorithms for better classification of the targets. For the present we use the classification results from the two networks shown in Figures 3 and 4.

5 Camera Calibration

To translate the measurements in image co-ordinates to world co-ordinates we need to know the perspective transformation matrix (3). The 3×4 matrix P is obtained by using manually selected points in the image and their corresponding measurements in the world co-ordinate system. The world co-ordinate is chosen so that the XY plane is aligned to the ground plane of the scene and Z axis is perpendicular to the ground plane.

$$\begin{bmatrix} x_i \\ y_i \\ \lambda \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3}$$

To find P we need a minimum of six corresponding points in the image and the world co-ordinate space. We pick more than the minimum number of points and use a least squares estimate to solve the over-constrained linear equations and filter out the noise due to errors in measurements.

From (3) it can be easily shown that if the 3D height of a point is known along with its image co-ordinates, then its unique 3D location can be computed as

$$X_w = \frac{(p_{32}y - p_{22})}{\{(p_{31}x - p_{11})(p_{32}y - p_{22}) - (p_{12} - p_{32}x)(p_{21} - p_{31}y)\}} \tag{4}$$

$$\left\{\frac{p_{12} - p_{32}x)(p_{23} - p_{33}y)Z_w}{(p_{32}y - p_{22})} + \frac{(p_{12} - p_{32}x)(p_{24} - p_{34}y)}{(p_{32}y - p_{22})}\right\}$$

$$Y_w = \frac{(p_{21} - p_{31}y)X_w}{(p_{32}y - p_{22})} + \frac{(p_{23} - p_{33}y)Z_w}{(p_{32}y - p_{22})} + \frac{(p_{24} - p_{34}y)}{(p_{32}y - p_{22})} \tag{5}$$

The classification of the object into one of the following classes *pedestrian, motorbike, cars, trucks, heavy trucks, and noise*, by BN allows representing the object's height by the values shown in Table 1. These values were obtained by measuring typical heights of the objects in different classes. A point which lies on top of the target will have its Z_w co-ordinate almost equal to the height of the target as we have initially aligned the XY of the world-coordinate with the ground plane of the scene. To ensure that the point we chose for tracking in the segmented foreground region is almost at the top of the target, the following constraints are applied:

- For pedestrians and m-bikes we select the point, which lies on the major axis of the ellipse and is 10% into the perimeter of the ellipse. Here the implicit assumption is that the objects appears upright in the frames.
- For cars the point selected lies on the major axis. It is mid way between the centroid of the ellipse and the point where the major axis intersects the ellipse perimeter in the direction of motion.
- For trucks and heavy trucks the point we choose is on the major axis of the ellipse approximating the target and is 10% inside the ellipse boundary in the direction of motion.

Pedestrian	M-bike	Car	Truck	Heavy Truck
1.7m	1.7m	2.0m	3.0m	3.2m

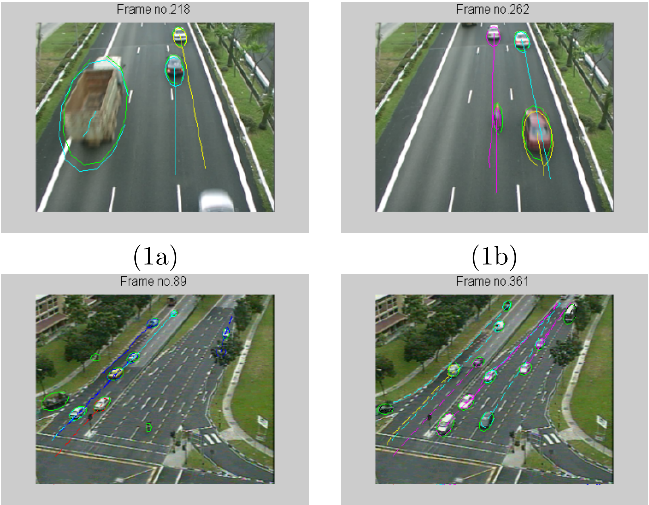
Table 1. This table shows the standard height values for the different classes used in the system

Using this technique of translating measurements from image co-ordinates to world co-ordinates we are able to detect vehicle speeds within an error range of ±5%. This range is obtained by comparing the speed measurement from the speedometer of the vehicle as ground truth and the speed measured from the tracking system as observation. This relatively high accuracy of speed detection makes it possible to detect the acceleration and deceleration of the targets.

6 Results

The system has been applied to several videos of traffic scenes in which pedestrian and other vehicles appeared. Over one thousand occurrences of different objects had been identified and target tracking performed for each one of them. Figure 5 shows a few images of our tracking results.

Tables 2 and 3 shows the classification results of the algorithm applied to these video streams. Very high recognition results have been obtained for cars and pedestrians. Lipton *at el.* [15] showed recognition results of 86.8% and 82.8% for vehicles and humans for classification using Mahalanobis clustering. Here we have higher correct classification rate even though the number of classes is six as compared to three in [15].



(1a) (1b)

(2a) (2b)

Figure 5. Tracking results of our system on two different real life video streams.

Object	Total	Correct Classification
Pedestrian	63	84.7%
Motor bike	105	90.4%
Cars	695	95.7%
Truck	92	90.3%
Heavy Truck	45	92.9%
noise	163	81.4%

Table 2. This table shows the classification results for the BN shown in Figure 3.

In Figure 6 we show the results of the 3D estimation algorithm proposed in the paper. When the estimated height of the vehicle is taken to be zero then there is significant error in the speed estimates. The speed estimates are in the range of 80-92 km/hour when the actual speed is 65km/hour. 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 initial frames when the object just enters the FOV show larger errors in speed estimates because the Kalman filter parameters take some time to settle. Later when the target starts moving move out of the FOV of the camera then the errors in speed estimate may be attributed to the error in choosing the point which lies on top of the target. However the speed estimates are quite accurate. This accurate measurement of speed allows detecting whether a vehicle is accelerating or decelerating. The observation of deceleration by the visual sensors is well correlated with observation of red lights at the back of the vehicle due to application of brakes.

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

Table 3. This table shows the classification results for the BN shown in Figure 4.

7 Conclusions

We have proposed a new approach to object classification in traffic video streams using BN. The network is capable of modelling the internal dependencies of the measured image features of the targets and hence classification results are more robust and reliable for different vehicle types and pedestrians. Another novelty of our work is that we have used the motion parameters obtained by tracking the object in 2D image space for classifying the object type. This makes the discrimination amongst the different classes more distinct. Because different object have different characteristic motion parameters like pedestrian will usually be slow than a motor bike. Furthermore motion is good indicator of objects position in perspective projects; and objects position determines the size of the object imaged in perspective projection. Another novelty of our work is that it effectively combines domain knowledge about the objects in the different classes with a computer vision algorithm to compute the world co-ordinate motion measurements of the targets. Future work involves computing an optimal structure of the BN using some structuring algorithms so that the unwanted bias introduced by the manual design can be removed and a more optimal inference network can be formulated.

References

[1] T. Choudhury, J. M. Rehg, V. Pavlovic, and A. Pentland. Boosting and structure learning in dynamic baysian networks for audio-visual speaker detection. In *Proccedings of International conf. on Pattern Recognition*, pages 789–794, Quebec City, Canada, August 2002.

[2] M.-P. Dubuisson and A. K. Jain. Contour extraction of moving objects in complex outdoor scenes. *International Journal of Computer Vision*, 14:83–105, 1995.

[3] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, May 1999.

[4] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. In *In Proc. Conf. on Uncertainity in AI*, Madison, WI, 1998.

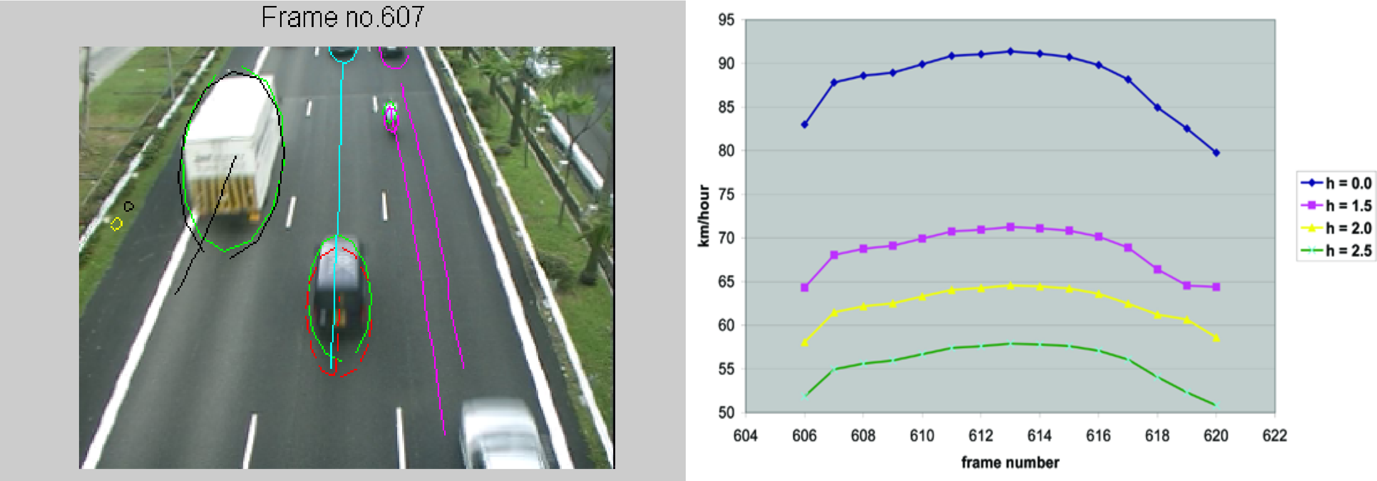


Figure 6. The tracking results of frame 607 shows a black van being tracked with a red ellipse. The experiment is so arranged that the van was moving with a constant speed of 65 km/hour as read from its speedometer. The plot in this figure shows the measured speed of the vehicle for different height estimates denoted by the parameter 'h'and expressed in meters. The estimated speeds from the proposed system, which uses h = 2 m, are 58.1-64.5 km/hour in frames 606-620, as shown by the yellow curve. This is quite accurate considering the errors in measurements from a video camera.

[5] M. C. A. Giachetti and V. Torre. The use of optical flow for road navigation. *IEEE Transactions on Robotics and Automation*, 14(1):34–49, Feb 1998.

[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)*, pages 222–227, April 1998.

[7] I. Haritaoglu, D. Harwood, and L. Davis. A fast background scene modeling and maintenance for outdoor surveillance. In *International Conference of Pattern Recognition*, pages 179–183, September 2000.

[8] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J. Weber. Automatic symbolic traffic scene analysis using belief networks. In *In Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, WA, 1994.

[9] O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *European conference on Computer Vision*, 2002.

[10] F. V. Jensen. *Lecture notes on Bayesian Networks and Influence Diagrams*. 1999.

[11] T. Kanade, R. Collins, A. Lipton, P. Burt, and L. Wixson. Advances in cooperative multi-sensor video surveillance. In *Darpa Image Understanding Workshop*, pages 3–24. Morgan Kaufmann, November 1998.

[12] D. Koller, J. Weber, T. Huang, J. Malik, G.Ogasawara, B. Rao, and S.Russell. Towards robust automatic traffic scene analysis in real-time. In *Proccedings of International Conference on Pattern Recognition*, Israel, November 1994.

[13] P. Kumar, S. Ranganath, and K. Sengupta. An efficient scheme for robust multi-body tracking. NUS TECH. Report, February 2003.

[14] P. Kumar, K. Sengupta, A. Lee, and S. Ranganath. A comparative study of different color spaces for foreground and shadow detection for traffic monitoring system. In *Proccedings of The IEEE 5th International Conference on Intelligent Transportation Systems*, pages 100–105, Singapore, September 2002.

[15] A. Lipton, H. Fujiyoshi, and R. S. Patil. Moving target detection and classification from real-time video. In *Proccedings of IEEE Workshop Application of Computer Vision, 1998*, pages 8–14, 1998.

[16] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80:42–56, 2000.

[17] 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*, 23(8):873–889, August 2001.

[18] A. Mittal and C. L. Fah. Addressing the problems of bayesian network classification of video using high-dimensional features. *IEEE Transactions on Knowledge and Data Engineering*, 2001.

[19] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrain detection using wavelet templates. In *Proccedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 193–199, San Juan, 1997.

[20] A. Prati, I. Mikic, C. Grana, and M. M. Trivedi. Shadow detection algorithms for traffic flow analysis: a comparative study. In *Proccedings of the 4th IEEE Intl Conf. On Intelligent Transportation Systems*, Oakland, California, August 2001.

[21] A. Stassopoulou and T. Caelli. Building detection using bayesian networks. *Intl Journal of Pattern Recognition and Artificial Intelligence*, 14(6):715–734, 2000.

[22] 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*, 24 No. 1:75–89, 2002.