University of Leeds

SCHOOL OF COMPUTER STUDIES

RESEARCH REPORT SERIES

Report 2001.21

Tracking Multiple Vehicles using Foreground, Background and Motion Models [1]

by

D R Magee

December 2001

[1] Submitted to European Conference on Computer Vision, May 2002

Abstract

In this paper a vehicle tracking algorithm is presented based on the combination of a per pixel background model (an extension of work by Stauffer and Grimson [12]) and a set of single hypothesis foreground models based on a general model of object size, position, velocity, and colour distribution. Each pixel in the scene is thus 'explained' as either background, belonging to one of the foreground objects or as noise. Calibrated ground-plane information is used within the foreground model to strengthen the object size and velocity consistency assumptions. A learned *a priori* model of typical road travel direction and speed is used to provide a prior estimate of object velocity which used to initialise the velocity model of each of the foreground object models. This model is typically an Extended Kalman filter but other models are possible within the algorithm. The system runs at near video frame rate (>20fps) on modest hardware and is robust assuming sufficient image resolution is available and vehicle sizes do not greatly exceed the priors on object size used in object initialisation.

1 Introduction

In recent years there has been much work on the tracking of moving objects within a scene. Systems developed for such tasks as people tracking [16, 1, 8, 6], face tracking [4, 9] and vehicle tracking [3, 13, 5] have come in many shapes or sizes, but may be broadly divided into explicit model based methods (where a fixed, detailed model of object characteristics is built e.g. [4]) or implicit model based methods (where more general object, or scene, characteristics are modelled, often dynamically e.g. [9]). There is no such thing as a 'model free' object tracker as all systems make assumptions about the object or scene which form the basis of a tracking model.

Our particular interest is in the analysis of traffic scenes with multiple (often large numbers) of vehicles interacting. A frame from a typical sequence is shown in figure 1.

Figure 1.1: Image of a Typical Traffic Scene

Figure 1 illustrates that the vehicles of interest vary widely in appearance and an explicit, detailed model is not necessarily suitable. Beymer *et al.* [3] came to a similar conclusion in a similar scenario. Ferryman *et al.* [5] (based on earlier work by Sullivan [13]) demonstrate that explicit models may be made to work in a vehicle mounted camera scenario, however results are only presented for 12 seconds of video, containing a limited number of vehicles. The former method has been demonstrated to work (to a degree of accuracy[2]) on 44 hours of video.

In this paper a more general modelling strategy is taken (a la Beymer *et al.* [3] rather than Ferryman/Sullivan *et al.* [5, 13]), including both a background and foreground model in our system. Our scheme uses a modified version of the Stauffer and Grimson [12] background model combined with a novel foreground model that borrows conceptually from this background model. Our foreground model is based on modelling vehicle invariants size, colour distribution and velocity (which is assumed to be locally invariant in time) for a particular vehicle. Foreground pixels (identified by the background model) rather than blobs (as used by Wren *et al.* [16]) or corner features / regions (as used by Beymer *et al.* [3]) are compared with various instances of our foreground model to determine to which model they belong (if any). A velocity model (one example of which is a Kalman filter) is used to propagate these models over time (predictive tracking).

A prior model of road velocity over the scene of interest is built from an initial (prior free) version of the tracker and used to initialise velocity and direction of travel estimates in the final implementation. This leads to 'lock' being achieved faster and fewer lost vehicles.

[2]The Baymer system achieves 74-94% accuracy depending on the road scenario

2 Modelling the Background Using a Multi-modal Statistical Model

There have been many methods proposed for the modelling of backgrounds. The simplest perhaps is to take a single frame of a scene containing no objects of interest and subtract the greylevel or colour values of this image from a frame containing objects of interest. Any non zero pixel values are thus classified as part of a foreground object. This method does not work in most practical situations due to noise in the intensity, caused by imaging and lighting effects, and spatial noise, caused by camera jitter. Practical methods attempt to model the distribution of this noise using statistical techniques.

Baumberg and Hogg [2] use a median filter at each pixel to construct a background model. A thresholded absolute image difference is used to identify foreground pixels. Haritaoglu *et al.* [6] model background pixel intensities using minimum and maximum values in addition to a maximum difference between frames. If a pixel intensity falls outside this model it is classified as foreground. Ridder *et al.* [10] use a Kalman filter at each pixel to model pixel intensities and predict a single value background model, however the foreground detection is performed using a thresholded absolute image difference. Wren *et al.* [16] model pixel colour as a full covariance Gaussian distribution in YUV colour-space. Similarly McKenna *et al.* [8] use a diagonal covariance Gaussian in RGB colour-space. Our work however is based on that of Stauffer and Grimson [12] in which pixel colour value histories in RGB [3] space are modelled as mixtures of Gaussians.

Mixtures of Gaussians allow the colour distribution of a given pixel to be multi-modal, which is essential if there is significant camera jitter as in our application. Figure 1 shows the distribution of pixel colour over time at an edge pixel in the presence of camera jitter.

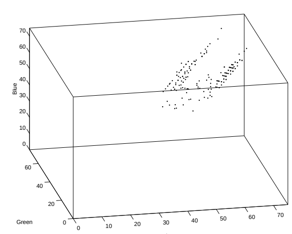

Figure 2.1: Colour Values of an Edge Pixel Over Time

In Stauffer and Grimson's method [12] a fixed number of Gaussians (typically 3-5), with diagonal covariances, are used. The variance of the red, green and blue terms is constrained to be equal. The parameters of the mixture (weights, Gaussian means and covariances) are updated dynamically over time using equations 1, 2 and 3.

[3] It is claimed this choice of colour-space is arbitrary and the method works as well in any colourspace

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

Where α is the learning rate and $M_{k,t}$ is 1 if the current RGB input matches Gaussian k (i.e. k is the closest Gaussian and within n standard deviations of the mean) and 0 otherwise. The weights are re-normalised after this update. If a Gaussian is matched its mean (μ) and variance (σ) terms are updated with reference to the current RGB input (X).

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t \tag{2}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{3}$$

where:

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k) \tag{4}$$

$$\eta(X|\mu,\sigma) = e^{-\frac{1}{2}(X-\mu)^T\sigma^{-1}(X-\mu)} \tag{5}$$

If the current RGB input matches no Gaussian the Gaussian with the lowest weight is replaced by one with its mean at the current input value and a large variance. To remove transients (i.e. foreground pixels) from the distribution only a subset of the mixtures are used as the background model. The subset is chosen by ordering the Gaussians in descending weight order and selecting the first B distributions where:

$$B = argmin_b(\sum_{k=1}^{b} \omega_k > T) \tag{6}$$

Where T is "a measure of the minimum portion of the data that should be accounted for by background".

2.1 Improvements to the Stauffer-Grimson Background Model

The system we are developing is to be used in the UK where the weather is variable. In particular the sun often disappears behind clouds for lengths of time before re-appearing rapidly. This can lead to fairly large intensity changes over short periods of time. Uni-modal background modelling schemes must model this as as single elongated distribution in colourspace (reducing the specicifity of the model) or use an intensity normalised colourspace. Our experiments suggest that the latter approach decreases the signal to noise ratio in an image stream as compression methods compress colour information rather more than intensity information in keeping with human perception. The alternative approach is to use a multi-modal representation such as Stauffer and Grimsons Gaussian mixture models [12]. We take this latter approach as our basis, however even this has its limitations.

The Stauffer/Grimson method adapts over time and, as such, encodes a finite time history of events. The learning rate controls the scale of this history (a faster rate encodes a shorter history) with a trade off necessary between being fast enough to adapt to novel changes and being slow enough to store a useful temporal history. At fast adaptation rates the distribution quickly becomes dominated by a single Gaussian (and thus uni-modal). We have modified the Stauffer/Grimson method to improve the temporal history storage while running at reasonably high adaptation rates (adapting in a novel background within a seconds or two rather than 20+ seconds as in the Stauffer/Grimson setup). These modifications are:

2.1.1 Variable Adaptation Framerate

Model distributions are updated with a variable framerate per pixel (a integer division of the incoming image stream framerate). Stable pixels (i.e. pixels that fit the background distribution over a number of frames) are updated with a lower framerate than pixels that have been recently classified as foreground. The framerate is given by:

$$R_{update} = \begin{cases} R_{input}/N_{bgd}, & \text{where } N_{bgd} < N_{max} \\ R_{input}/N_{max}, & \text{otherwise} \end{cases} \tag{7}$$

Where:
R_{update} = Update rate
R_{input} = Input Framerate
N_{bgd} = No. of consecutive background frames
N_{bgd} = Threshold on frame rate division (typically 25 frames / 1 sec)

This addition also serves to reduce the computational expense of the method considerably. The reader may note that the scheme described does not differentiate between changes in the background and foreground models entering the scene. The result is that a high adaptation rate is used at pixels containing a foreground object. As a result of this slow moving vehicles may become included in the background model. If a method is available to differentiate between foreground object pixels and pixels where colour change is simply a result of an environmental change the adaptation rate of the pixels relating to foreground objects may be reduced. Fortunately the foreground modelling method (described later) provides us with exactly this information. It is important to note that the adaptation rate should never be reduced to zero as, if an area of pixels are falsely classified as resulting from a foreground object, the environmental change that caused this would never be incorporated in the background model and the false hypotheses would be propagated through time. Fortunately in our case there are constraints on what is classified as part of a foreground model (see later) including a non zero speed requirement in the initialisation phase and this is not really a problem.

2.1.2 Maximum Weight Limit for Update

To further prevent the distribution becoming effectively uni-modal an upper limit is put on the weight of any one Gaussian component (typically 0.5). On update if a sample matches a Gaussian with a weight above this threshold the mixture components are left unchanged. If the distribution of a pixel has low modality (i.e is fairly stable) this can lead to unmodified Gaussians or Gaussians relating to temporary transients being selected as background. To counter this Gaussians are labeled on initialisation as 'unmodified' and only Gaussians subsequently labeled as 'modified' are used in the background model. In addition a lower limit is put on the weight of any Gaussian to exclude transients.

2.1.3 Modelling in RGB space as an 'Intensity Cylinder'

The Stauffer/Grimson method effectively defines a background colour distribution as lying within a set of spheres in colourspace. Figure 2 illustrates that (for a non-edge, theoretically uni-modal) pixel the distribution is not modelled well by a sphere.

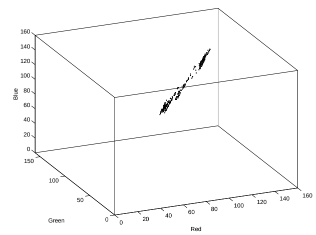

Figure 2.2: Colour Values of a Non-Edge Pixel Over Time

It can be seen from figure 2 that (for a non edge pixel) intensity values approximately lie along a section of a straight line in RGB colourspace that, if extended, approximately intersects the origin. The Stauffer/Grimson method models this uni-modal distribution as a multi-modal set of spheres in RGB space. If an intensity normalised colourspace was used any point along the complete line would be equivalent, allowing a a single sphere to be used, however the discrimination power of the space would be reduced (especially at low intensities). Our approach is to model the variation along (and perpendicular to) an approximation to this line. The approximation used is the line between the origin and the current mean of the data (or uni-modal subset of the data). We model the variation along this line as a single (1D) Gaussian centered on the mean, and the variation perpendicular to the line with a fixed noise threshold. Mixtures of this 'cylindrical' representation are used in exactly the same way as in the Stauffer/Grimson method to define a per-pixel observation history. This may be thought of as equivalent to using a mixture of Eigenspaces [15], however without the computational overhead of actually performing Principal Components Analysis.

2.2 Modelling Very Low and Very High Intensities

At low and high intensities the assumption that uni-modal data lies along a straight line breaks down due to reduction in signal to noise ratio (at low intensities) and colour saturation (at high intensities). In such circumstances the 3D colour mixture is simply converted into a 1D intensity mixture and this used as the basis of background determination.

3 Modelling Foreground Objects

Stauffer and Grimson [12] state they do not use a foreground model; their approach is to associate 'blobs' extracted using connected components analysis using a Kalman filter. A similar approach is taken by Beymer *et al.* [3] who associate regions containing corner features over time using a Kalman filter. We conjecture that such methods do contain a foreground model which is based on the Kalman filter assumptions (e.g. constant velocity). Such methods are essentially a simplification of the strategy employed by Wren *et al.* [16] in which prior spatial and temporal information is included in addition to motion assumptions. All these methods work by classifying extracted features (corners, blobs / connected regions) as coming from one of a number of processes (vehicles, body parts).

In our scenario (see figure 1) features such as corners are un-reliable due to the relative size of the objects of interest. Connected components analysis (based on a foreground extracted using a per-pixel background model) is also a poor tool as the similarity of objects to background in some cases can result in a highly fragmented foreground with many lone pixels. This is illustrated in figure 1.

a)
b)

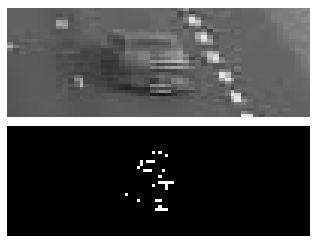

Figure 3.1: Fragmented Foreground Object Produced Using the (original) Stauffer/Grimson Background Method

It is clear from figure 1 that performing connected components analysis adds little information and, if small regions are discarded, may significantly reduce the information present. For this reason we choose to associate pixels, rather than blobs or regions, with object models. This has the added advantage of a computational saving. Our foreground model is inspired by the Stauffer/Grimson background model and also (in terms of the use of colour) by the work of McKenna *et al.* [8]. The model consists of representations for:

1. **Position:** A single 2D point is used to represent the centroid of a given object on the ground plane.

2. **Size:** 1D Gaussians are used to represent the object size as a variation (in ground plane co-ordinates) in the direction of travel and perpendicular to the direction of travel (relative to the position).

3. **Velocity:** Single values are used to represent a rolling average of the velocity vector (in ground plane co-ordinates). An alternative implementation uses an Extended Kalman filter to estimate these values.

4. **Colour Distribution:** A Gaussian mixture is used to represent colour over the entire object in exactly the same way as the background model.

4 Tracking Using Prediction

Tracking is performed by predicting forward position from the previous frame into the current frame (using the velocity estimate obtained by taking a rolling average of the position differential or from a Kalman filter[4]) and associating each pixel with a foreground model. This is performed by defining a distance measure based primarily on the Position/Size of objects for each measure. This is given as the magnitude of the manhalnobis distances of the data from the model in the two size distributions as in equation 8.

$$D^2 = \frac{\Delta_{in}^2}{V_{in}} + \frac{\Delta_{per}^2}{V_{per}} \tag{8}$$

Where:
Δ_{in} = Distance of data from model mean in dir. of travel
Δ_{per} = Distance of data from model perpendicular to dir. of travel
V_{in} = Model variance in dir. of travel
V_{per} = Model variance perpendicular to dir. of travel

A pixel is associated with the model with the lowest distance (D) if this distance is below a specified threshold (typically 2.5). If this distance is above the threshold but below a second threshold (typically 4 or 5) the colour value of the pixel is compared with the colour Gaussian mixture in the model by taking the minimum mahalanobis distance of the colour value from any Gaussian mixture mean. If this distance is less than a specified threshold (typically 2.5) the pixel is accepted as resulting from this model, otherwise it is rejected. This scheme allows the object size hypotheses to be enlarged over time from the initial hypotheses if (and only if) the image colour information supports this. Figure 1 illustrates how pixel/model grouping is performed.

If a pixel is not classified as resulting from one of the current foreground objects a new model is initialised centred at this pixel with pre-specified size parameters (based on the typical size of a car). The colour distribution is initially uninitialised as it is built up over the first few frames.

Model parameters are updated at each frame from associated pixels in a similar manner to the background model parameters. Position is calculated as a weighted centroid of associated pixels, with the predicted position (μ_{pred}), and size variance ($\sigma_{in}, \sigma_{per}$) providing the weights as in equation 9.

$$\mu_{new} = \frac{\sum X_n \eta(X_n|\mu_{pred}, \sigma_{in})\eta(X_n|\mu_{pred}, \sigma_{per})}{\sum \eta(X_n|\mu_{pred}, \sigma_{in})\eta(X_n|\mu_{pred}, \sigma_{per})} \tag{9}$$

Size variances are calculated by taking an unweighted mean of the square of the distances from associated pixel locations to the predicted centroid in the direction of

[4] The Kalman filter is observed to give a better estimate of velocity

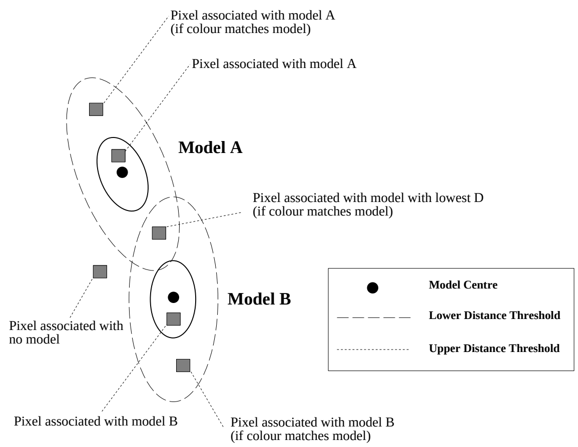

Figure 4.1: Association of Foreground Pixels with Foreground Models

travel and perpendicular to the direction of travel as given in equations 10 and 11.

$$\sigma_{in_{new}} = \frac{\sum (X_n - \mu) \cdot \triangledown}{N} \tag{10}$$

$$\sigma_{per_{new}} = \frac{\sum (X_n - \mu) \cdot \triangledown \perp r}{N} \tag{11}$$

Where $\triangledown$ and $\triangledown \perp r$ are unit vectors in the direction of travel and perpendicular to the direction of travel respectively. Colour distributions are updated in the same way as for the background model (see section 2), with pixels from different spatial, as well as temporal, locations updating the model.

As with the background model an update factor (α) is used to control the update. The colour model is updated as described previously (section 2) and the position and size variance models updated using a weighted sum as given in equations 12 to 14.

$$\mu_{n+1} = \alpha \mu_{new} + (1 - \alpha) \mu_n \tag{12}$$

$$\sigma_{in_{n+1}} = \alpha \sigma_{in_{new}} + (1 - \alpha) \sigma_{in_n} \tag{13}$$

$$\sigma_{per_{n+1}} = \alpha \sigma_{per_{new}} + (1 - \alpha) \sigma_{per_n} \tag{14}$$

The update factor (α) in initially set high to allow rapid adaptation to a novel object. After a number of frames (typically 5-10) α is reduced to stabilise the model, locking it on to its target.

5 Incorporating A-priori Road Information

In the initial implementation no knowledge of road direction or typical speeds was included. This forced initial velocity and direction of travel estimates to zero. Section 4 describes how velocity and direction of travel information are integral in the tracking process. Poor initial values for these can lead to failure of a model to 'lock onto' a vehicle (especially in the far-ground). If an initial zero velocity hypothesis is used a model mean can lag behind the centroid of the moving vehicle until the velocity estimate becomes more realistic. If image support is poor the lag can become great enough that the vehicle is lost. The direction of travel effects the perceived aspect ratio of the vehicle, this can also lead to poor tracking.

The solution to this problem is to build a prior model of typical road travel direction and speed. As cars typically travel in the same direction, within a limited speed range, on the same stretch of road (due to road traffic regulations) any prior based on previous observation should be a reasonable estimate of the actual direction and speed of a newly observed vehicle. Our prior model is trained on the output of the initial implementation of the tracker, on the basis that, in general, this gives accurate tracking results and false and missing tracks are rare enough to be treated as outliers in any statistical model.

The position output of the tracker is quantised using the vector quantiser proposed by Johnson and Hogg [7]. Positional (ground plane) space is then represented as a set of 2D Gaussian mixtures centred on these VQ prototypes using the adaptive Kernel method [11] [5]. This method places Gaussian Kernels with higher variance in areas of lower prototype density. In this way the 'contribution' of the i'th point in space to the j'th prototype may be calculated as in equation 15.

$$p_{ij} = \frac{w_j G(\mathbf{x}_i|\mu_j, \sigma_j)}{\sum_{j=1}^{m} G(\mathbf{x}_i|\mu_j, \sigma_j)} \tag{15}$$

Given the velocity output of the tracker, a direction vector ($\bigtriangledown$) and speed can be calculated for each vehicle at each timestep. We use the contribution of each of these to each prototype to calculate a weighted mean direction and speed as in equations 16 and 17.

$$\bar{\bigtriangledown}_j = \frac{\sum_{i=1}^{N} p_{ij} \bigtriangledown_i}{\sum_{i=1}^{N} p_{ij}} \tag{16}$$

$$\bar{S}_j = \frac{\sum_{i=1}^{N} p_{ij} S_i}{\sum i = 1^N p_{ij}} \tag{17}$$

Figure 1 shows a typical ground plane velocity map learned.

The velocity map is used to calculate an initial estimate of velocity and direction (unit normalised velocity) when new object instances are initialised using equation 18.

$$V_{xy} = \frac{\sum_{j=1} M \bar{S}_j \bar{\bigtriangledown}_j G(x, y|\mu_j, \sigma_j)}{\sum_{j=1} M G(x, y|\mu_j, \sigma_j)} \tag{18}$$

[5] Optimal performance was achieved using a window width around 10 times smaller than the value suggested by Silverman [11]

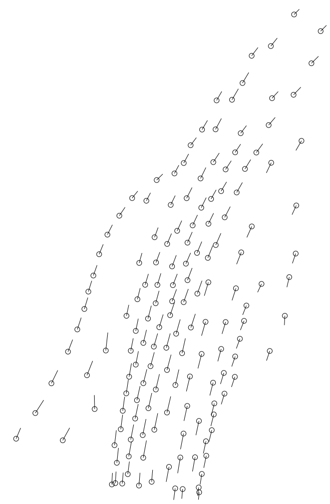

Figure 5.1: Ground Plane Velocity Map Learned From Tracked Data

6 System Evaluation

The vehicle tracker was evaluated by drawing a box around all vehicles in a scene using an interactive tool. The results of this for a single frame is shown in figure 1.

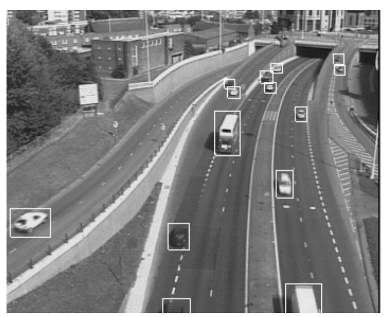

Figure 6.1: Hand Fitted Boxes Used as 'Ground truth' for Evaluation

This interactive fit was performed at ten frame intervals on a 1 minute / 25fps sequence (i.e. 150 frames were labeled up). The sequences contains 102 separate objects with 12-28 being present in any one frame. This gives a total of 2563 object instances to evaluate. The sequence was carefully chosen to contain several types of vehicles (cars, vans, lorries) and different sorts of flow (congested and relatively free flow). Statistics for how well the tracker matches this artificial 'ground truth' were then gathered. Figure 2 shows results for the complete tracker, in addition to two incomplete trackers which do not use road direction information or colour information in modelling the foreground objects respectively.

Tracker	Objects Tracked	Prop. Frames Tracked*
Full Tracker	100%	80.7 % [20.2 %]
No Road	100%	78.6 % [20.7 %]
No Colour	100%	79.3 % [21.9 %]

* Values given are mean and std. dev. over all objects

Figure 6.2: Evaluation of Tracking Results

The results presented in figure 2 are comparable with those presented by Baymer *et al.* [3] (although evaluated on a completely different sequence). Presenting the results as we have done however does not tell the entire story as the tracker performs much better in he foreground than it does in the background, as objects are significantly smaller in the background than the foreground due to foreshortening. To demonstrate this the input image is divided evenly into three regions as illustrated in figure 3.

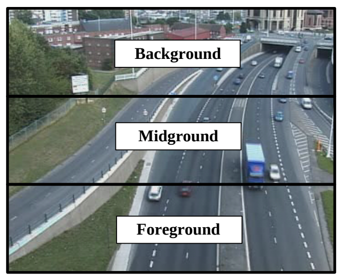

Figure 6.3: Definition of the three evaluation areas

Typical sizes of objects in these regions are 4x3, 8x7 and 16x11 pixels at a resolution of 180x144. This resolution was chosen as it is possible to run the tracker at near frame rate (20-23Hz depending on number of objects) on a conventional PC (PIII 1GHz) at this resolution. Figure 4 shows the results for the full tracker presented divided into these three regions.

Section	Objects Tracked	Prop. Frames Tracked
Foreground	100%	95.9 % [12.0 %]
Midground	99%	95.7 % [15.9 %]
Background	99%	71.4 % [27.5 %]

Figure 6.4: Evaluation of Tracking Results for different Regions

It can be seen that the tracker performs well in the foreground and midground, however performance is poorer in the background. An additional consideration when evaluating an object tracker is multiple object hypotheses being associated with different models. This is very rarely a problem when tracking cars (typically less than 1% of cars are associated with multiple hypothesis), however for larger vehicles such as lorries this is a frequent problem. These large vehicles represent 5-10% of the vehicles observed.

7 Discussion, Conclusions and Future Work

We have presented a vehicle tracking system that has potential for use in an online road monitoring scenario. The system is based on the combination of a mixtures of Gaussians colour background model and a set of foreground models each of which propagates a single estimate of an object position, velocity, size and colour distribution. An *a priori* model of typical road travel direction and speed and priors on object size are used in the initialisation phase. This system has been evaluated offline against a hand labeled sequence and found to be robust if a) There is sufficient resolution in the image for the background model to distinguish an object from sensor noise (typically the object must be at least 6x6 pixels in the image plane) and b) the prior on the size of the vehicle is accurate.

Issue a) is easily addressed by zooming the camera in further (and using multiple cameras if necessary), however issue b) is not so straightforward. The initialisation scheme described in this paper is rather basic and not ideal in terms of its initial position and size estimates (however it is extremely computationally efficient). Pixels that don't match a current object hypothesis are used to initialise new object instances in raster scan order. This leads to new hypotheses being located at the top left of a new object (rather than at the centroid. Figure 1 illustrates this works adequately for vehicles of similar (or smaller) size to the initial size hypothesis, however for larger vehicles multiple hypothesis may be initialised.

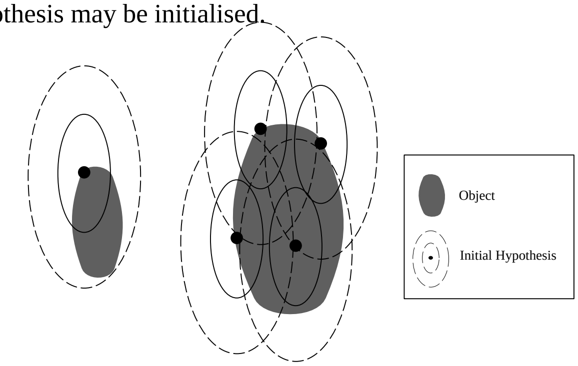

Figure 7.1: Multiple Initialisation Problem for Large Vehicles

Increasing the initial size hypotheses is not a solution to this problem as this would tend to associate a hypotheses with multiple vehicles when these vehicles are in close proximity. This is currently not a great problem as the initial size hypothesis values are chosen to relate to the smallest observable vehicles (cars) and, even when in close proximity, the colour model of one generally dominates and excludes pixels relating to the other from the update stage. It can be seen from figure 1 that the location of the hypothesis are more of a problem than the (initially) fixed size. Future work will investigate initialisation schemes that cluster pixels un-associated with current vehicle hypothesis to allow initial hypotheses to be better located near the centroid of vehicles.

Developing methods that perform this task robustly at frame rate is a far from trivial task.

Ferryman *et al.* [5] present a method (based on an earlier algorithm by Tan *et al.* [14]) for identifying vehicle type (car, van or lorry) by examining image gradient profiles and comparing them to histograms calculated offline for the three vehicle types. A similar method could be used within the context of our tracker to select between a set of size priors.

In conclusion we have presented an vehicle/object tracking scheme based on general assumptions about scene and object characteristics. These assumptions (colour consistency for the background and local position, size, colour and velocity consistency for moving objects) generally hold true and we have identified previously where these assumptions break down. This is in contrast to more detailed models where assumptions (such as shape consistency over objects) are at best approximations to the truth. Priors on object characteristics (such as information on expected size and velocity) are easily incorporated into this scheme. The tracker could be extended to include shape priors (if appropriate) which may model objects better than the current Gaussian assumption. It is however not clear that any single shape prior would be generally applicable due to the wide range of observed object shapes..

The combination of background and foreground models serves to completely 'explain' the observed scene and is an approach that is now feasible for online ('real time') systems due to the increased computational power available. There is much scope for future research into methods that fully model (explain) an observed scene as the result of a set of underlying processes.

References

[1] A. Baumberg and D. Hogg. An efficient method for contour tracking using active shape models. In *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 194–199, 1994.

[2] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In *European Conference on Computer Vision*, pages 299–308. Springer Verlag, 1994.

[3] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *Proc. CVPR*, pages 495–501, 1997.

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *Proc. European Conference on Computer Vision*, volume 2, pages 484–498, 1998.

[5] J. Ferryman, S. Maybank, and A. Worrall. Visual surveillance for moving vehicles. *International Journal of Computer Vision*, 37(2):187–197, 2000.

[6] I. Haritaoglui, D. Harwood, and L. Davis. W4: Who? when? where? a real time system for detecting and tracking people. In *Proc. IEEE Conference on Automatic Face and Gesture Recognition*, pages 222–227, 1998.

[7] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14:609–615, 1996.

[8] S. McKenna, S. Jabri, Z. Duric, and H. Wechsler. Tracking interacting people. In *Proc. International Conference on Automatic Face and Gesture Recognition*, pages 348–353, 2000.

[9] S. McKenna, Y. Raja, and S. Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17(3-4):225–231, 1999.

[10] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. In *Proc. International Conference on Recent Advances in Mechatronics*, pages 193–199, 1995.

[11] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[12] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. CVPR*, pages 246–252, 1999.

[13] G.D. Sullivan. Model-based vision for traffic scenes using the ground-plane constraint. *Real-time Computer Vision*, 1994. D. Terzopoulos and C. Brown (Eds.). CUP.

[14] T.N. Tan, G.D. Sullivan, and K.D. Baker. Efficient image gradient-based localisation and recognition. In *Proc. CVPR*, pages 397–402, 1996.

[15] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society B*, 12(3):611–622, 1999.

[16] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on PAMI*, 19(7):780–785, 1997.