Computer vision techniques for traffic flow computation

Article *in* Pattern Analysis and Applications · December 2004
DOI: 10.1007/s10044-004-0238-x · Source: DBLP

Li Bai
University of Nottingham

Computer Vision Techniques for Traffic Flow Computation

Li Bai[1], William Tompkinson[2], Yan Wang[1]

[1]School of Computer Science and Information Technology,
University of Nottingham, Jubilee Campus,
Wollaton Road, Nottingham, NG8 2BB
{bai, yqw}@cs.nott.ac.uk

[2]Research & Innovation,
Ordnance Survey,
Romsey Road, Southampton, SO16 4GU
William.Tompkinson@ordnancesurvey.co.uk

Abstract. This paper describes an application of computer vision techniques to road surveillance. It reports on a project undertaken in collaboration with the Research & Innovation group at the Ordnance Survey. The project aims to produce a system that detects and tracks vehicles in real traffic scenes to generate meaningful parameters for use in traffic management. The system has now been implemented using two different approaches: a feature-based approach that detects and groups corner features in a scene into potential vehicle objects, and an appearance-based approach that trains a cascade of classifiers to learn the appearances of vehicles as an arrangement of a set of pre-defined simple Haar features. Potential vehicles detected are then tracked through an image sequence using the Kalman filter motion tracker. Experimental results of the algorithms are presented in this paper.

Key words: computer vision, object tracking, road surveillance, Kalman filter.

1 Introduction and Related Work

Detecting and tracking vehicles in traffic scenes using computer vision techniques has been an active research area in recent years. Traffic data such as traffic flow and vehicle velocity is important for traffic control and road engineering in order to improve efficiency of road utilisation [1]. This is a topic that has been of strong interest to cities and communities but has seen little real progress over the past few years. Traffic flow calculation for traffic management constitutes a solution to a 'real' problem and requires the combination of many components within a system.

Several strategies are followed in the literature for vehicle detection and traffic surveillance, including active contour based [2][3], model based [4][5][6][7], feature based [1], appearance based and stereo vision based approaches [8][9]. Koller et al [2] presents an active contour based approach to track multiple cars on the road. The approach employs a background model to perform motion based image segmentation. Active contours are then used to enclose detected blobs (vehicles) segmented from the background. Two Kalman filters are combined to form a motion model for estimating shape and motion of the object contours. Belief network based reasoning is also adopted to cope with vehicle partial occlusion. The working of the system is demonstrated by smooth vehicle motion trajectories tracked through 270 video frames of a four-lane motorway. However, the system is too slow for real time applications - 8 to 10 vehicles are tracked simultaneously at a rate of 7 frames per second on a SunSparc station. Shadows in images pose another problem, due to which two vehicles may be wrongly merged into one by the system.

Sullivan et al [4] proposed a model based approach for vehicle detection and classification that takes into account multiple views of a vehicle. The system operates in three stages: hypothesis generation to predict the existence of vehicles using 1D vehicle templates, hypothesis tracking to follow vehicles through an image sequence, and hypothesis verification using 2D sparse wireframe vehicle templates. The templates are computed according to the expected poses of each of the vehicle models known to the system. The 1D templates used by hypothesis generation are obtained by firstly constructing a vehicle centred object coordinate system whose *x* and *y* axes are coincident with the two major directions of the image, and then by projecting orthographically 3D wireframe vehicle models along the coordinate axes. In hypothesis generation, the gradients of an image region are projected along the same vehicle centred object coordinate axes and correlation analysis is then performed between the model projection and the image projection. Evidence of strong correlation between the two sets of projections indicates the presence of a vehicle in the image. In the hypothesis verification stage, the hypothesized vehicle (image) is matched against pre-defined sparse 2D wireframe vehicle templates. A confidence value for the hypothesis is computed by averaging the projections of the image gradients along appropriate lines of the 2D templates. The approach suffers from heavy computation cost. Its use is therefore constrained by computational resources and the need to operate in real time. It also has to deal with the mismatch problem, e.g., windscreen of the model could be matched to the bonnet of the observed vehicle.

The approaches mentioned above consider only scenarios involving stationary and single cameras. Traffic surveillance systems based on stereo vision have also been developed. A stereo vision based system is described in [8]. The system is designed for vehicle navigation purposes, which requires the detection of road markings and obstacles, and the calculation of distance to an obstacle from the vehicle in which the navigation system is being used. For obstacle detection and distance calculation, the system uses stereo matching techniques. To reduce the complexity of stereo matching, the system 'remaps' stereo images using *inverse perspective mapping* to remove perspective effect from the images. A difference image is then computed from the two remapped stereo images for obstacle detection and obstacle distance computation. However, this approach will only work if the road surface is flat, as the inverse perspective mapping process can only produce remapped perspective free images of the road if the road condition is satisfactory. Stereo vision works well within short distances but it is unsuitable for our application. Computation of stereo matching and inverse perspective mapping is also expensive. It is clear that there is definite a need to research real time vehicle detection and tracking methods that are able to achieve a reasonable trade-off between speed and accuracy. In this paper we describe a prototype system that uses computer vision techniques to detect and track vehicles in real traffic scene to gather information for traffic flow computation. The system can work properly on video frames of 320x240 pixels and at the frame rate of 20Hz on a 1.13GHz laptop. The system is created under the assumption that a stationary camera (CCTV or webcam) is mounted at a position so that a wide road area can be covered and vehicle occlusion is reduced to minimum. A single traffic direction is considered at any one time. The system comprises six major components. These are frame acquisition, road modeling, vehicle detection, vehicle tracking, traffic flow calculation and data storage.

The vehicle detection component implements two approaches, using corner features and Haar features respectively. Corner features are detected from the traffic scene first and are then divided into a number of groups according to two pre-defined heuristic rules. Each corner group represents a vehicle. A cascade of classifiers, previously used for face detection [10], is trained to distinguish vehicles from their background based on the presence of Haar features. Once a vehicle is detected the Kalman filter is used to track it. The Kalman filter algorithm aims to construct a probabilistic model for a system to iteratively estimate the next state of the system based on the current state estimate. Practical issues in using the Kalman filter, such as how to model vehicle motion are investigated.

The outline of this paper is as follows: Section 2 presents a feature-based approach for vehicle detection using corner detection and grouping; Section 3 describes an alternative approach to vehicle detection based on a cascade of classifiers trained to recognize vehicles by the presence of Haar features; Section 4 introduces a method for road modeling using the homographic transform to map image coordinates to world coordinates. The Kalman filtering algorithm and its use for vehicle tracking is given in Section 5. Section 6 reviews a method for computing traffic flow; Section 7 demonstrates experimental results of vehicle detection algorithms implemented in the system. The paper concludes with a brief comparison of different approaches and possible directions for future research work.

2 The Feature-based Approach

Accurate extraction of image features is crucial to the subsequent tracking of objects along the image sequence. Rather than tracking objects as a whole, the feature based approach detects features of a vehicle object such as corners or edges and tracks them. In circumstances this approach is believed to be advantageous over other approaches since it is able to handle partial object occlusion and requires less computation.

To detect corner features, first the spatial image gradient $(G_x, G_y)^T$ is calculated for each image point. Then for each image point p and its neighborhood Q (e.g., a 3×3 block surrounding p), a 2×2 matrix of the sums of gradients over Q in both x and y directions is defined as:

$$C := \begin{pmatrix} \sum G_x^2 & \sum G_x G_y \\ \sum G_x G_y & \sum G_y^2 \end{pmatrix}$$

The eigenvectors of C encode edge orientations and the eigenvalues encode edge strength. A point p is considered a corner point if at p the eigenvalues λ_1 and λ_2 of matrix C are larger than a pre-defined threshold t.

Vehicles on the road often come in clusters and each of them may have several corners. It is therefore critical to group corners that are likely to belong to the same vehicle. Two heuristic grouping rules are defined for this purpose. One dictates that if two corners are distant enough, they should belong to different vehicles. The other dictates that if two corners move at different speeds (i.e., their distance varies in consecutive video frames), it is reasonable to assume that they belong to different vehicles.

To apply the first grouping rule the maximal distance between corners needs to be specified. This depends on the average size of vehicles, which is quite difficult to define and which requires mapping image coordinates to world coordinates, as discussed in Section 4. Generally speaking, most cars are no more than two meters wide and no more than six meters long. Therefore the distance between two corners of the same car would be less likely to exceed this limit. Given a set of corners in image coordinates $P_1, P_2, \ldots, P_n$, and their corresponding world coordinates (X_1, Y_1), $(X_2, Y_2), \ldots, (X_n, Y_n)$, any two corners P_i and P_j in one group should satisfy that $\left|X_i - X_j\right| \le \tau_w$ and $\left|Y_i - Y_j\right| \le \tau_l$. τ_w and τ_l can be set to the maximal width and length of ordinary cars respectively.

However, the first grouping rule may not work if one vehicle is partially occluded by another. Therefore, the second grouping rule is applied to ensure correct grouping. This involves constantly evaluates and records distance variations between corner features within the same group along a sequence of video frames. If the distances from corner from others in the same group change dramatically in two consecutive frames, the corner should be taken out of the group, since the distances between corners in the same group (vehicle) should stay relatively stable. For example, assume a pair of corner features P_1 and P_2 are back-projected to world coordinates via the homographic transform, and are grouped together in light of the first grouping rule at instant t_0. The Euclidean distance between them, $d(P_1, P_2)$, can be obtained in world coordinates. The distance is continuously recorded along the time instants and its variance in two consecutive instants, Δd, is computed. Two corners remain in the same group unless the variance Δd reaches a threshold. In the case of a significant variance Δd, the original group is split into two subgroups.

Figure 1 illustrates the second grouping rule applied to four corners along a sequence of video frames. Suppose that at instant t_k, a remarkable change of distance between P_1 and P_3 is observed which exceeds a predefined threshold. Suppose also that P_1 is chosen as the reference point and the distances between P_1 and others, e.g., P_2 and P_4, are examined. If $\Delta d_{1,2} < \tau$ and $\Delta d_{1,4} > \tau$, P_1 and P_2 will be regrouped together so will P_3 and P_4, resulting in two subgroups, each of which is counted as a vehicle. The same grouping process iterates on each of the subgroups at the subsequent time instant.

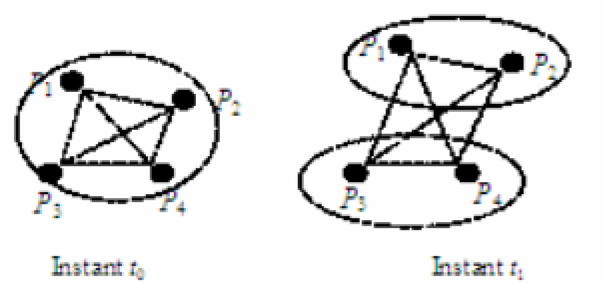

Figure 1: The process of dividing a group into two.

3 The Appearance-based Approach

Corner features do not articulate the shape of vehicles. Hence the feature based approach can only be applied if all moving objects on the road are vehicles. However, most traffic scenes contain non-vehicle objects (e.g. buildings, motorcycles, and people) that also contain corner features. This prompts us to use pattern recognition techniques to classify vehicles and non-vehicles according to their appearances. For this purpose we choose to use the fast AdaBoost classifier training algorithm, as described in [10][11][12][13], to generate a 'strong' classifier from a cascade of 'weak' classifiers. This involves extracting Haar features, named after the Haar wavelets, from positive training sample images (vehicles), and negative training sample images.

As shown in Figure 2, Haar features are computed using a set of pre-defined templates, each of which consists of a number of dark-shaded and lights-shaded rectangles. The templates that are made of two rectangles can be used to detect edges, those made of three rectangles are suitable for detecting lines, and those with four rectangles can be used to detect corners in images. To compute Haar features for an image, a looping procedure can be employed that slides each template through each location in the image. At a specific location in the image, a Haar feature can be calculated by subtracting the sums of intensity values between the image pixels covered by the light rectangles and pixels covered by the dark rectangles of the template. The size of a template also varies so that features at different scales can be computed.

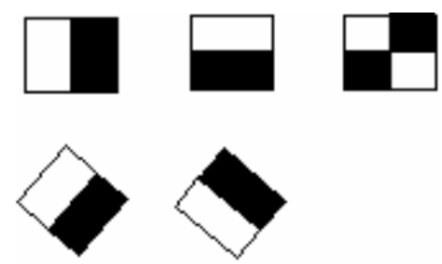

Figure 2: Haar feature types

In the AdaBoost training process, each weak classifier is then trained/boosted to select a few Haar features from the whole set of features obtained from training images, which can best separate positive training samples (vehicles) from negative samples (non-vehicles). Specifically, a weak classifier is designed for determining a threshold for optimal classification as:

$$h(x) = a\delta(x_i > \theta) + b \tag{3.1}$$

where x is a vector consisting of Haar features detected from 24x24 grayscale images, x_i is the ith component of the feature vector x, θ is a threshold, δ is a parity function negative when the inequality is not satisfied, and positive otherwise, and a and b are regression parameters to best fit the training data. At the start of the training process each training sample is weighed equally. A weak classifier is generated to fit the data at each round of the boosting process. The weights are iteratively adjusted for the next round of the process to highlight the classification errors made in the current round. The cascade of boosted classifiers can be illustrated as a degenerate decision tree, as shown in Figure 3. Each is trained to achieve a low

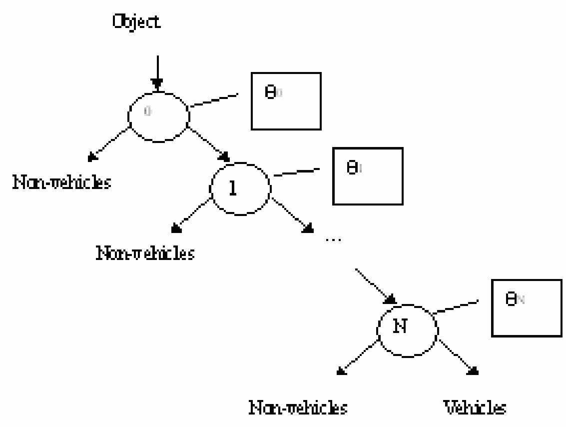

Figure 3: A cascade of classifiers

false negative as well as a low false positive rate by evaluating positive outputs from the previous stage classifier and rejecting negative examples. If the desired minimal hit rate and maximal false positive rate of each classifier are set to α and β respectively, the minimal hit rate and maximal false positive rate of cascade can be bounded by α^n and β^n, where n is the number of stages in training the cascade.

4 Road Modeling

In order to perform corner grouping using real distances in the world coordinate system, we need to map the real coordinates to image coordinates. Instead of the IPM algorithm described in [8] and the camera calibration process, we apply a simple homographic transform to map image coordinates to world coordinates. This involves multiplying the image coordinate with a 3×3 matrix H:

$$(X\ Y\ Z) = H_{3\times3}\ (x\ y\ 1) \tag{4.1}$$

where $(X\ Y\ Z)$ is a world coordinate and $(x\ y\ 1)$ is an image coordinate. In planar homographic transform an assumption is made that all points in the real scene are on a single plane. Once at least four points on the plane and their correspondences in the image plane are specified, the matrix H can then be decided uniquely. If we choose one of those points as the reference point, the world coordinates of the others can be specified in terms of real distance from the reference point. This is obviously not exactly the case for 3D scenes. However it is a reasonable approximation for vehicles on the road, since if observed through a wide view camera, vehicles can be seen moving in parallel with the road plane and are close to the ground. As a result, the modeling process aims to find four points in the real world to define a rectangle according to the shape of the road in the traffic scene.

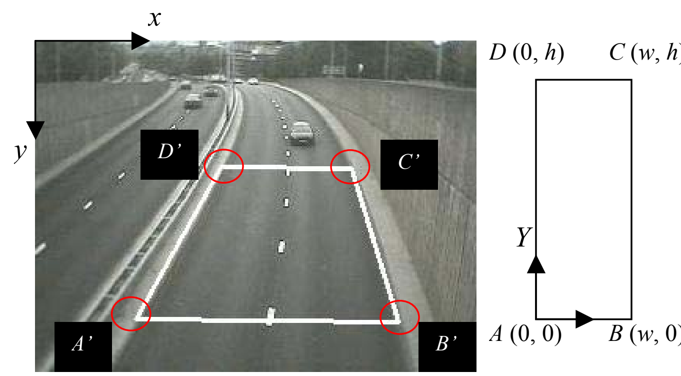

Figure 4: A road section is modeled as a quadrangle in an image. Point *A'*, *B'*, *C'* and *D'* are projections of *A*, *B*, *C* and *D* in world coordinates, respectively.

The question is how to find the four vertices of the quadrangle in an image, which can be back projected to the 3D world as a rectangle in order to compute the homographic matrix. The possible solution is to make use of the shape of the road. Figure 4 illustrates the mapping between a road section (a rectangular region) in a traffic scene and a quadrangle in an image. Note that the line segments *A'D'* and *B'C'* are in accordance with the edges of the road and *A'B'* limits the road section which we are interested in together with *C'D'*. If the length and width of the road section are known, the world coordinate system can be established, regarding one of the vertices of the road section as the origin point (for instance, the bottom left point). Four points can therefore be specified in terms of world coordinates. A field survey is carried out to observe road markings and measure the width and length of a rectangular road section in the traffic scene. The surveillance region is thus represented as a quadrangle in our system and we only need to concentrate on the quadrangle - only vehicles passing through it will be detected and tracked.

5 Tracking Vehicles using Kalman Filter

We constructed a linear Kalman filter [14] to track vehicles detected through a sequence of video frames. The Kalman filter can construct a linear motion model for a moving object and predicts its position in the next image frame, based on the estimates made in previous frames. Assume that at instant t_{k-1} the location of a vehicle in the world coordinate system is $(x_{k-1}, y_{k-1})^T$ and that its velocity is $\mathbf{v}_{k-1} = (v_{x,k-1}, v_{y,k-1})^T$. The state vector of vehicles can be described as $\mathbf{x}_{k-1} = (x_{k-1}, y_{k-1}, v_{x,k-1}, v_{y,k-1})^T$. At the next instant t_k the location and velocity of the vehicle can be described as:

$$(x_k \quad y_k)^T = (x_{k-1} \quad y_{k-1})^T + (v_{x,k-1} \quad v_{y,k-1})^T + \xi_{k-1} \tag{5.2}$$

$$(v_{x,k} \quad v_{y,k})^T = (v_{x,k-1} \quad v_{y,k-1})^T + \eta_{k-1} \tag{5.3}$$

where $\boldsymbol{\xi}_{k-1}$ and $\boldsymbol{\eta}_{k-1}$ are zero-mean Gaussian distribution functions. The two equations can be written in one united form as:

$$x_k = \phi_{k-1} x_{k-1} + \omega_{k-1} \tag{5.4}$$

where the state transition matrix is:

$$\phi_{k-1} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and $\omega_{k-1} = (\xi_{k-1}, \eta_{k-1})^T$. Since we only need measure the location of a vehicle, the measurement z_k is $(x_k \quad y_k)^T$ at instant t_k and can be acquired by:

$$z_k = H_k x_k + \mu_k \tag{5.5}$$

In the above equation, $H_k = \left(\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix}\right)$ is the measurement matrix, and $\boldsymbol{\mu}_k$ is a random additive noise approximating the error of measurement. The initial state $\boldsymbol{e}_0$ of the vehicle to be tracked can be set to $(x_0, y_0, 0, 0)^T$, $(x_0, y_0)^T$ is the world coordinate of the vehicle when it enters the surveillance region and velocity is initially set to $(0, 0)^T$. The validation region is defined by:

$$v_k^T S_k^{-1} v_k < \gamma \tag{5.12}$$

where $\mathbf{v}_k = \boldsymbol{z}_k - H_{k-1} \boldsymbol{e}_k'$ is the innovation, S_k the covariance of the innovation, γ is an arbitrary positive number, and $\boldsymbol{e}_k'$ is the observation. If $\boldsymbol{e}_k'$ lies within this region, it will be accepted as a valid measurement to the current state of the system. In multi-target tracking, it is possible that several locations would appear in that region. Thus the problem is to decide which measurement should be used to update the state of the object to be tracked. The simplest solution is the nearest neighbor method to select the closest one. In other words, the candidate location, which minimizes the left-hand side of equation 5.12, is chosen as the actual position of the vehicle being tracked:

$$\arg\max_l \{P(z_{k,l})\} = \arg\min_l \{v_{k,l}^T S_{k.l}^{-1} v_{k,l}\} \tag{5.13}$$

If no features are found in the predicted region, the object is lost and should be dropped from the tracking process.

6 Computing Traffic Flow

In the context of road surveillance, traffic flow is one of the most important traffic parameters. Traditionally, it has been measured by averaging the number of vehicles that pass a specified point of the road over a sampled period of time. We adopt a more feasible approach recommended in [1], which generates traffic flow over time and space by considering vehicle trajectories passing through the surveillance region. Mathematically, given a surveillance region A with length L and width W, and length of each vehicle's trajectory passing through A is denoted as l_i, over a sampled period of time t, the flow f can be computed as:

$$f = \frac{\sum l_i}{L \times W \times t} \tag{6.1}$$

where L×W denotes area of region A.

7 Experiments

The experiments were conducted on a notebook PC with a 1.13G Intel® Pentium® III processor and 120 Megabytes internal memory. A processing speed of 20 frames per second was achieved, which is fast enough for most real time computer vision applications. We recorded a 15 minute video clip for the experiments with a resolution of 320x240 pixels showing 771 vehicles traveling towards the camera on a four-lane motorway in daytime. The traffic scene contains shadows of vehicles and road markings. The results are measured by three parameters: false positive rate, false negative rate and hit rate.

7.1 Results of Feature-based Approach

The results are presented in Table 1. A screen shot of vehicle tracking is shown in Figure 5. We report here our observations and problems encountered in the experiments using the feature based approach and possible solutions to these problems.

Vehicles	Count	FN	FNR%	FP	FPR%	HR%
771	734	110	14.3	73	9.5	85.7

Table 1: Tracking results using corner detection. FPR: False Positive Rate. FNR: False Negative Rate. HR: Hit Rate.

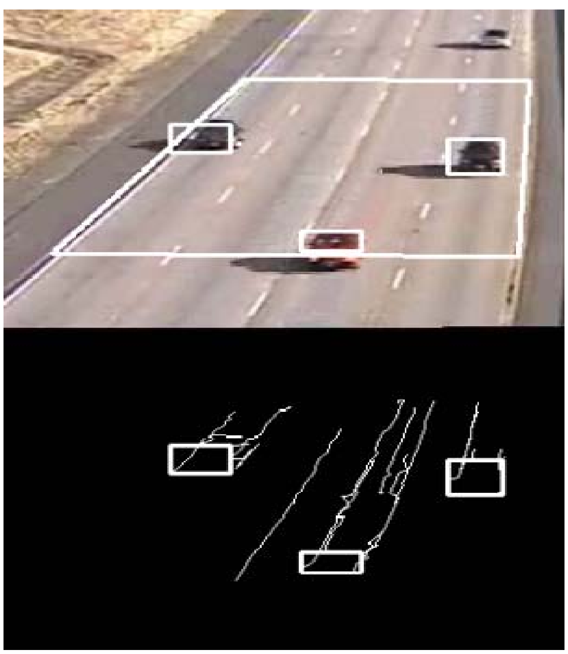

Figure 5: Vehicle detection and tracking using corner features

Firstly, it was observed that some corner features have appeared in the current frame but later on disappeared in the subsequent frame. This will cause sudden perturbation of the tracked trajectory, see Figure 5. Consequently, it may cause difficulty in using the second grouping rule. Corners deviating from a group may be considered as belonging to another group and thus the same vehicle will be wrongly counted twice. One solution to this problem is to enforce the *prediction precision* of the Kalman filter by shrinking the prediction region, at the cost of possibly losing some potentially important features. Another possible solution is to discard the corner group that only contains one corner as it arrives at the end of the surveillance region, assuming that most of vehicles would have several stable corners.

The second major problem is concerned with crowded road conditions. If two cars are moving in parallel and getting very close to each other, their corners may initially be put into one group according to the first grouping rule. Assume further that they move at the same speed and would not deviate from one another until after they are out of the surveillance region. Corners of both cars will remain in the same group and the two cars are wrongly counted as one. This situation is rare but is quite possible in heavy traffic conditions. Currently there is no ideal solution to this problem, although the surveillance region can be defined long enough so that relative movement between two vehicles can be easily observed before they move out of the region.

The third major problem encountered in the experiments is the interference of stationary corners in the background such as those of road markings. To get round this problem we first pre-process the image using morphological operator such as the dilation operator to eliminate, as much as possible, road markings while preserving the contrast of vehicles to the background. An opening function to smooth off jagged objects in the image is conducted twice on the image. The opening function is defined as:

$$A \circ B = Dilate(Erode(A, B), B) \qquad (7.1)$$

where A is the original image, B is a flat structured element. We use a 3x3 rectangle with the anchor point B on its top right corner. The result of the morphological operation is shown in Figure 6.

Figure 6: Morphological operation to remove road markings.

Varying vehicle sizes are another course of concern of the feature based approach. Large-sized vehicles (truck, bus) tend to be taller than cars, resulting in the upper part of these vehicles appearing to be distant from the road surface. This causes the effect that some corners 'float' over the road surface and appear increasingly further away from other corners due to a high degree of perspective. In this case, significant errors may occur when computing world coordinates of these corners using the homographic transform.

7.2 Results of Appearance-based Approach

In this experiment the number of iterations of the AdaBoost training process is set, arbitrarily, to 10. 480 hand-labeled positive sample images that contain vehicles of frontal appearance and 516 negative sample images that do not contain any vehicles are used in the training process. The negative sample images contain the background (e.g., road surface) or other scenes totally irrelevant to vehicles. All the training examples are normalized to 24x24 pixels. Some positive examples are shown in Figure 7. The expected minimal hit rate and maximal false alarm rate are set to 0.99 and 0.6 respectively for each classifier. The hit rate and false positive rate of the cascade is estimated at $0.99^{10} \approx 0.904$ and $0.6^{10} \approx 0.006$.

Figure 7: Some examples of vehicles used for training.

As mentioned, the AdaBoost algorithm trains each classifier in the cascade to select a few features. Totally 22 features are selected by the cascade during the training process. The first and second Haar features selected are depicted in Figure 8.

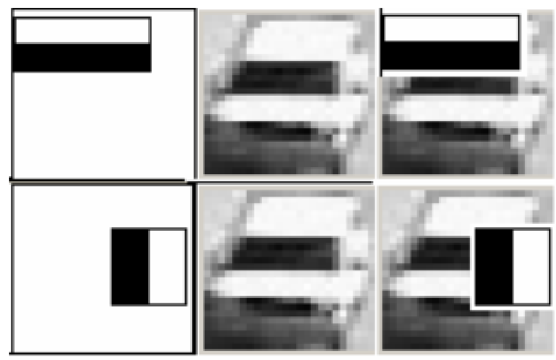

Figure 8: The first and second Haar features selected

The test results of the appearance based approach are presented in Table 2. A screen shot of this approach tracking vehicles in the 15 minute video footage with 771 vehicles is shown in Figure 9. The cascade has achieved a hit rate of 78.7% and false positive rate of 13.7%. This is not the best result. The inaccuracy is largely due to the insufficient amount of training data available. Nevertheless, the results are good enough in terms of the application.

Vehicles	Count	FN	FNR%	FP	FPR%	HR%
771	713	164	21.3	106	13.7	78.7

Table 2: Tracking results based on Haar feature detection. FPR: False Positive Rate. FNR: False Negative Rate. HR: Hit Rate.

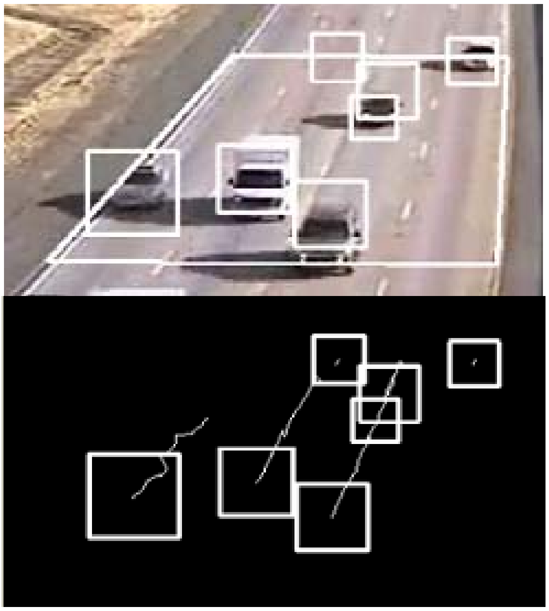

Figure 9: Vehicle detection and tracking using Haar features.

8 Conclusion and Future Work

Comparing the results of the two approaches to vehicle detection and tracking, it appears that, with the particular long video footage used for the experiments, the overall performance of the feature based approach using corner features is better than that of the appearance based approach using Haar features. However, the appearance based approach has outperformed the feature based approach in previous experiments on shorter (3 minutes) video footages. We list the previous results in Table 3 and Table 4 for discussion purposes. One of the shorter videos used before shows 65 vehicles approaching the camera (this footage is named as 'In' in Table 3 and Table 4), the other footage has 30 vehicles moving away from the camera (this footage is named as 'Out'). Table 3 contains the results of feature based approach and Table 4 shows the results of appearance based approach. A cascade of nine classifiers is trained to locate vehicles from the 'In' video sample, using 100 positive and 45 negative samples. Whilst a cascade of only four classifiers is trained for the 'Out' video sample, using 72 positive and 40 negative samples. Again the results are measured by three parameters: false positive rate, false negative rate and hit rate.

Video	Vehicles	Count	FPR%	FNR%	HR%
In	65	66	10.77	10.77	89.23
Out	30	29	16.67	13.33	86.67

Table 3: Tracking results using corner detection (short video)

Video	Vehicles	Count	FPR%	FNR%	HR%
In	65	65	7.69	7.69	92.31
Out	30	29	3.33	6.67	93.33

Table 4: Tracking results using Haar features (short video)

There are several possible reasons for this inconsistency of results, which can be explained by the strength and weakness of each of the approaches. Corner detection does not depend on the shape of a vehicle, which is advantageous in handling vehicle occlusion in that although some corners may be occluded, remaining corners can still be grouped to represent a vehicle. However, corner detection works under the assumption that all moving objects in the traffic scene are vehicles. If non-vehicle objects are present constantly in the scene, the performance will suffer. Some corners may either not be detected or be allocated to a wrong group, resulting the same vehicle being counted more than once or not counted at all.

The performance of the appearance-based approach using Haar features depends on the distribution of training samples in the feature space. As the training data we use in the experiments is from only a few video clips recorded during a short time period in a day, the cascade classifier is less accurate. Heavy road traffic will also render the approach less accurate because partially occluded vehicles will look different from training vehicles. One way to improve the performance of this approach is to include images of various vehicles, of various poses, and under various lighting conditions in the training set. Though this makes classifier training a non-trivial task, the strength of this appearance approach lies in its speed in real time operation. It is faster than the model-based approach, which matches a potential vehicle object against all possible templates at run time. It is also possible to combine the appearance based and feature based approaches to produce a hybrid approach and take advantages of both approaches.

Motion tracking also plays a key role in system performance. Robust tracking algorithms need to be developed to cope with real traffic scenarios. We are currently experimenting with Condensation [15] and ICondensation algorithms [16] as alternatives to the Kalman filter tracker. The Condensation algorithm allows more general representations of probability distributions and the use of non-linear motion models more complex than those commonly used in Kalman filters. This enables it to cope with temporarily ambiguous image evidence and to maintain tracking in the presence of clutter and occlusions. ICondensation improves Condensation by combining it with statistical technique of importance sampling.

This investigation has resulted in methods for real time road surveillance, producing results that could be used in dynamic applications such as web-based traffic flow maps. The evaluation of different approaches could also benefit other researchers.

Acknowledgements

Yan Wang was supported by Ordnance Survey under its Postgraduate Research Programme.

9 References

[1] Coifman B. (1998) A Real time Computer Vision System for Vehicle Tracking and Traffic Surveillance, Institute of Transportation Studies, University of California.

[2] Koller D., Weber J. and Malik J. (1994) Robust Multiple Car Tracking with Occlusion Reasoning, Proceeding of Third European Conference on Computer Vision.

[3] Malik J., Russell S., Weber J., Huang T. and Koller D. A Machine Vision Based Surveillance System for California Roads, PATH Project MOU-83, University of California.

[4] Sullivan G. D., Baker K. D., Worrall A. D., Attwood C. I. and Remagnino P. R. (1996) Model-based vehicle detection and classification using orthographic approximations, 7th British Machine Vision Conference.

[5] Böckert A. (2002) Vehicle Detection and Classification in Video Sequences, Computer Vision Laboratory, Department of Electrical Engineering, Linköping University.

[6] Remagnino P., Baumberg A., Grove T., Hogg D., Tan T., Worrall A., Baker K. (1997), An Integrated Traffic and Pedestrian Model-Based Vision System, Proceedings of British Machine Vision Conference 1997, volume 2.

[7] Worrall A. D., Sullivan G. D., Baker K. D. (1993), Advances in Model-based Traffic Vision, Proc. of the 4th British Machine Vision Conference

[8] Bertozzi M. and Broggi A. (1998) GOLD: A Parallel Real time Stereo Vision System for Generic Obstacle and Lane Detection, IEEE Transaction on Image Processing.

[9] Rabie, T. and Auda, G. (2001), Active Vision based Traffic Surveillance and Control, 3rd International Symposium on Mobile Mapping Technology

[10] Viola P. and Jones M. (2001) Rapid Object Detection using a Boosted Cascade of Simple Features, CVPR.

[11] Oren M., Papageorgiou C., Sinha P., Osuna E. and Poggio T. (1997) Pedestrian Detection Using Wavelet Templates, CBCL and AI Lab, MIT.

[12] Lienhart R. and Maydt J. (2002) An Extended Set of Haar like Features for Rapid Object Detection, IEEE International Conference on Image Processing.

[13] Freund Y. and Schapire R. E. (1996) Experiments with a New Boosting Algorithm, AT&T Research.

[14] Welch G. and Bishop G. (1995) An Introduction to the Kalman Filter. Technical Report TR95-041, University of North Carolina at Chapel Hill.

[15] Isard M. and Blake A. (1998) Condensation Conditional Density Propagation for Visual Tracking, IJCV, Volume 29(1):5-28.

[16] Isard M and Blake A (1998) ICondensation: Unifying low-level and high-level tracking in a stochastic framework, Proc 5th European Conf. Computer Vision, Vol. 1 893-908.