Vision-based Traffic Surveys in Urban Environments

Zezhi Chen[a], Tim Ellis[a] * and Sergio A Velastin[b]

[a] Digital Information Research Centre, Kingston University, UK
[b] Applied Artificial Intelligence Research Group, University Carlos III, Madrid, Spain
* Email: t.ellis@kingston.ac.uk

Abstract — This paper presents a state-of-the-art vision-based vehicle detection and type classification to perform traffic surveys from a roadside CCTV camera. Vehicles are detected using background subtraction based on a Gaussian Mixture Model (GMM) that can cope with vehicles that become stationary over a significant period of time. Vehicle silhouettes are described using a combination of shape and appearance features using an intensity-based pyramid HOG (histogram of orientation gradients). Classification is performed using SVM (support vector machine), which is trained on a small set of hand-labeled silhouette exemplars. These exemplars are identified using a model-based pre-classifier that utilizes calibrated images mapped by Google Earth to provide accurately-surveyed scene geometry matched to visible image landmarks. Kalman filters track the vehicles to enable classification by majority voting over several consecutive frames. The system counts vehicles and separates them into four categories: car, van, bus and motorcycle (including bicycles). Experiments with real-world data have been undertaken to evaluate system performance and vehicle detection rates of 96.45% and classification accuracy of 95.70% have been achieved on this data.

Index Terms—Background subtraction, Gaussian mixture model, Vehicle detection, Vehicle classification, Vehicle tracking.

I. INTRODUCTION

With the increasing levels of traffic congestion on urban road network accurate estimates of traffic type and flow analysis are important elements for effective traffic management. A variety of technologies can be used to acquire such data – inductive loop sensors, ANPR, audio, radar speed measurement, satellites, and increasingly, wireless communications technology such as cellular radio, RFID and bluetooth and CCTV. Each offer different benefits and drawbacks, though the richness of information associated with CCTV can enable a much broader understanding of traffic patterns, but at the cost of a more demanding image analysis and interpretation task.

Roadside CCTV is widely deployed in many countries to provide remote observation of traffic volumes and flow to a traffic control centre. Current analysis of the video is mostly undertaken manually by experienced operators who employ the information as one source of data to achieve efficient traffic control.

Automated analysis of the video data is seen as an increasingly important component of an intelligent transportation system (ITS), providing the capability to fully exploit the real-time information available from the live video stream. Several problems have to be solved in recovering measurements of individual vehicles, ranging from low and middle level vision tasks, such as the detection and tracking of multiple moving objects in a scene, to high level analyses, like vehicle classification [1]. Vehicle classification is particularly useful for gathering traffic statistics, re-identification in multi-sensor networks and anomalous event detection as well as more common applications of traffic flow analysis and unobtrusive path tracing.

Identifying moving objects in a video sequence is a fundamental and critical task in video surveillance, traffic monitoring and analysis, human detection and tracking, as well as other visual tracking tasks, such as gesture recognition in the human-machine interface. A static camera observing a scene is a common case of a surveillance and monitoring system. Background modeling is widely used to estimate the background and then detect the moving objects in the scene. The general theory is that the background model is built from the data and objects are detected if they appear significantly different from the background. The foreground pixels are further processed for object detection, tracking and classification. The principal challenges are how to correctly and efficiently model and update the background model and how to deal with shadows. A robust system should be independent of the scene, and robust to lighting effects and changeable weather conditions. It should be capable of dealing with movement through cluttered areas, objects overlapping in the visual field, gradual illumination changes (e.g. time of day, evening and night), sudden illumination changes (e.g. when street lighting is switched on and off, headlights, clouds moving in front of the sun), camera automatic gain control (e.g. white balance and auto-iris are often applied to optimally map the amount of reflected light to the digitizer dynamic range), moving background (e. g. camera vibration, swaying trees, snow or rain), slow-moving objects and ones that become stationary, cast shadows (trees, buildings, etc. cast on the road surface) and geometric deformation of foreground objects.

Other challenges for the video analysis include the problem if detecting vehicles subject to varying levels of occlusion, which is increasingly likely as traffic density increases and in the presence of large vehicles (e.g. buses and large trucks).

Vehicle type classification is important for determining the proportion of vehicle classes over given periods of time. Such information is traditionally collected manually in periodic surveys of road usage, but provides only a limited snapshot of traffic distributions that will have a diminishing currency over time. An automated system offers a more accurate, lower cost solution that can provide continuous, real-time output. Compared with object recognition from still images, analysis of video sequences simplifies the recognition task, because moving objects can be more easily separated from a static background using background modeling and subtraction, so problems of clutter are reduced.

This paper presents an Automatic Vehicle Detection and Classification (AutoVDC) system that classifies and counts vehicles as they move through a fixed detection zone in a camera's field of view. Vehicles are classified into one of four primary categories: motorcycle (including bicycles), car (car and taxi), van (van, minivan, minibus and limousine) and bus (single/double decker). A fifth category of vehicle that covers large commercial vehicles (truck) was excluded in the current work as insufficient examples were available in the video dataset that was used for evaluation. Counts are then produced for each category. One aim is to collect traffic census data for statistical analysis.

Figure 1 depicts the video analysis system partitioned into four modules. The first involves learning the background. A common challenge for many background separation algorithms occurs when moving objects become stationary in the scene (an all-too common occurrence for urban traffic) and are 'absorbed' into the background model. When they finally move, there is a lag in detection and "ghost" effects, resulting in poor segmentation. We address this transient stop-start behavior with a new self-adaptive Gaussian Mixture Model (GMM) that remembers the original background, invoking this model and reacting faster when the traffic begins to move again. The second stage detects foreground objects moving against the background, suppressing shadows and 'mending' holes in the binary silhouette using a morphological operation. A manually-positioned detection zone is used so that vehicles are reliably counted only once, even when the traffic becomes stationary. Finally, silhouettes are classified into one of four classes (car, van, bus and motorcycle/bicycle). Defining stable class types is challenging because some instances of a class (e.g. MPVs (multi-purpose vehicles) and small vans) may be very similar in appearance (especially from particular viewpoints) and the classification may be ambiguous, resulting in high error rates. We combine tracking with classification and use a majority voting over several frames to improve classification performance. Training classifiers typically require large quantities of labeled data to reliably model intra-class variation. Labels are normally generated by manually annotating a large number of samples, but this ground-truthing task can be time-consuming and laborious and would need to be repeated for different camera positions. We propose a semi-automated approach to ground-truthing that requires a small number of annotations and demonstrate that this can achieve classification error rates comparable to those from a large manually-annotated dataset.

This paper focuses on a systematic evaluation of the performance of the system, identifying the contribution made by each of the analysis stages and determining the set values and sensitivity of the various parameters that control the performance of each algorithmic stage. It considers the performance of the Self-Adaptive background GMM (SAGMM) and shadow suppression in coping with changing illumination and stationary vehicles; representation of the resulting vehicle silhouettes using shape and appearance-based features; an efficient method for the annotation of vehicles; and a combination of tracking with classification to achieve a high classification performance.

The next section reviews the literature on vehicle detection and classification. Section III describes our background learning and foreground extraction; vehicle detection is described in Section IV; feature extraction, vehicle type classification and counting are presented in Section V; experimental results are presented in Section VI and discussed in Section VII. Finally, conclusions and future perspectives are given in Section VIII.

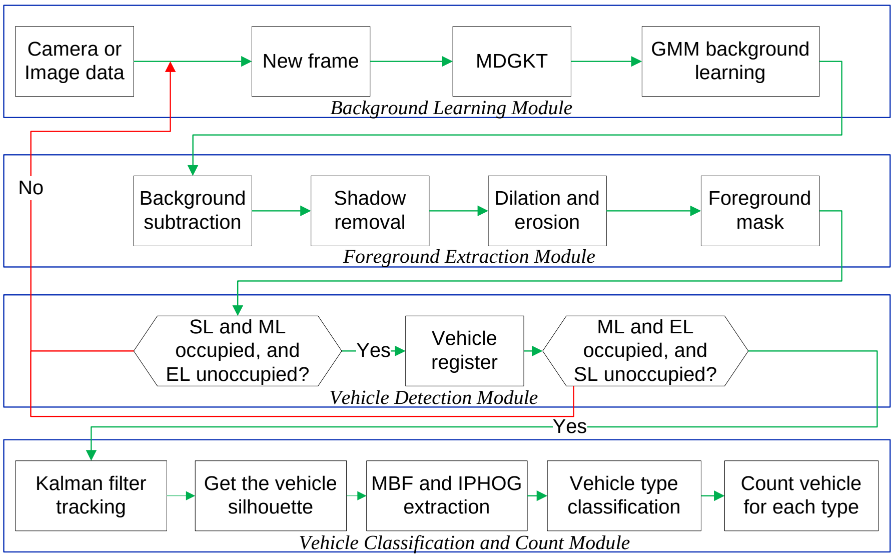

Camera or Image data
New frame
MDGKT
GMM background learning
Background Learning Module
No
Background subtraction
Shadow removal
Dilation and erosion
Foreground mask
Foreground Extraction Module
SL and ML occupied, and EL unoccupied?
Yes
Vehicle register
ML and EL occupied, and SL unoccupied?
Vehicle Detection Module
Yes
Kalman filter tracking
Get the vehicle silhouette
MBF and IPHOG extraction
Vehicle type classification
Count vehicle for each type
Vehicle Classification and Count Module

Fig.1. Flow chart of the AutoVDC system.

II. LITERATURE REVIEW

A. Gaussian mixture model and foreground segmentation

At the heart of any background subtraction algorithm is the construction of a statistical model that describes the state of each background pixel. Many algorithms have been developed and the most recent surveys can be found in [2][3]. A widely-used approach that models the multi-modal background distribution with a Gaussian mixture model (GMM) was originally proposed by Stauffer and Grimson [4]. The algorithm relies on assumptions that the background is visible more frequently than the foreground and that the model has a relatively narrow variance. This approach has been found to cope reliably with slow lighting changes, repetitive motions from clutter, and long-term scene changes.

However, GMMs have drawbacks. Firstly, they are computationally intensive and the parameters require careful tuning. Second, they are sensitive to sudden changes in global illumination. If a scene component remains stationary for a long period of time, the variance of the background components may become very small. A sudden change in global illumination can then turn the entire frame into foreground; a low learning rate can result in a very wide and inaccurate model that will have low detection sensitivity. On the other hand, for a high learning rate, the model updates too quickly, and slow moving objects will be absorbed into the background model, resulting in a high false negative rate.

Many adaptive GMM models have been proposed to improve the original method. Power and Schoonees [5] used a hysteresis threshold to extend the GMM model. They introduced a faster and more logical application of the fundamental approximation than that used in [4]. The standard GMM update equations were extended to improve the speed and adaptation rate of the model [6]. Greggio et al. [7] recently proposed self-adaptive Gaussian mixture models for real-time background subtraction. All these GMMs use a fixed number of components. More recently, Zivkovic and Heijden [8][9] presented an improved GMM model using a recursive computation to constantly update the parameters of a GMM, which adaptively selects the appropriate number of Gaussians to model each pixel on-line, from a Bayesian perspective.

Martel-Brisson and Zaccarin [10] proposed a novel statistical model based on a GMM to cope with scenes with complex and time-varying illumination, including regions that are highly color-saturated, whilst suppressing false detection in regions where shadows cannot be detected. Zhao and Lee [11] extended the background GMM to spatial relations, where the joint colors of each pixel-pair are modeled by a GMM to suppress the effects of illumination changes. Unfortunately, none of the existing background models can achieve robust performance to sudden changes in global illumination. Moreover, the parameters of a GMM may vary for different scenarios. Existing work either openly admits to setting blending and thresholding parameters by hand, or more commonly, does not mention how they are set.

Another challenge in the application of background subtraction is identifying shadows cast by objects that also move with them through the scene. Shadows cause serious problems while segmenting and extracting moving objects due to the misclassification of shadow points as foreground. Prati et al. [12] presented a comprehensive survey of moving shadow detection approaches. Cucchiara et al. [13] proposed the detection of moving objects, ghosts and shadows in HSV color space and gave a comparison of different background subtraction methods.

Wu and Juang [14] have presented an adaptive vehicle detector approach for complex environments. In their approach, histogram extension was used to suppress the effects of weather and illumination variation and a gray-level differential value method was used to extract moving objects. A big reported advantage of this method is its high processing speed, yielding an average of 76 frames per second and average detection and false alarm rates of 93.7% and 3.7%, respectively.

B. Vehicle classification

Approaches to vehicle classification can be ascribed to: i) 3D vehicle modelling, ii) shape-based recognition, iii) appearance-based and iv) categorising different vehicle types. Early research by Tan et al. [15] used 3D wireframe models to track and classify vehicles. They utilized camera calibration and the ground-plane constraint to match image edge features to a set of wireframe models (hatchback, saloon and lorry) for both tracking and classification. More recently, Unzueta et al. [16] have used background subtraction and 3D models to detect, track, count and classify vehicles by integration of temporal information and model priors within a Markov Chain Monte Carlo method. The detected vehicle was classified as either a two wheeled, light or heavy vehicle. Messelodi et al. [1] also used 3D models and low level image features to classify vehicles into seven categories (bicycle, motorcycle, car, van, lorry, urban bus and extra-urban bus) in an urban environment, whilst Buch et al. applied a 3D model [17] and 3DHOG [18] to detect and classify vehicles into four categories (bus/lorry, van, car/taxi and motorbike/bicycle).

Hasegawa and Kanade [19] described a vision system to recognize moving targets such as vehicle type and pedestrians on a public street using an 11-dimensional vector of image features. They used features of the object’s bounding box, including, the width, height and area; first, second and third image moments and the centroid coordinate. Ma and Grimson [20] used edge-based features and modified scale invariant feature transform (SIFT) descriptors. They were able to distinguish objects at a more detailed level, discriminating between vans, minivans, sedans and taxis. An appearance learning-based method is presented by Zhang and Avery et al. [21] that can distinguish between moving objects, such as cars, vans, trucks, people and bikes using multi-block local binary patterns. Morris and Trivedi [22] presented a tracking system with the ability to classify vehicles into three classes. They constructed a 10-feature measurement vector and then applied either principal component analysis (PCA) or Fisher's linear discriminant analysis (LDA) to manage the size of the data, followed by a weighted k-nearest neighbour (wkNN) classifier and fuzzy C-means clustering methods.

Zhang and Li et al. [23] proposed a length-based vehicle classification ITS using un-calibrated video cameras. Chen et al. [24] classify road vehicle type (car, van and HGV – heavy goods vehicle) with a combination of size, shape and color using a support vector machine. The feature vector to describe the vehicle silhouette encodes size, aspect ratio, width, solidity and 3D color histograms. Mithun et al. [25] used multiple time-spatial images (TSIs) to identify latent occlusions among the vehicles and reduce the dependencies of the pixel intensities between the still and moving objects to increase the accuracy of detection and classification performance. A two-step *k*NN classifier used shape and texture-based features to identify vehicle class. The detected vehicle was initially classified into two, three, four and six-wheel vehicle categories, and then specific types of vehicles were identified.

III. Background Learning and Foreground Extraction

This section describes a GMM model (SAGMM) for foreground extraction that uses a dynamic learning rate and a global illumination change factor to deal with sudden illumination changes and also incorporates a mechanism for shadow suppression; this is fully described in [26]. It has been modified to cope with objects that become stationary for relatively long periods of time (i.e. hundreds of frames), which is a common occurrence in urban traffic streams, associated with traffic light stops, pedestrian crossing, queueing at junctions and slow queueing rush-hour scenarios.

A. Multi-dimensional Gaussian kernel density transform (MDGKT)

A multivariate kernel is used to perform spatio-temporal smoothing in the input color image to reduce image noise. It is defined as the product of two radially symmetric kernels and the Euclidean metric allows a single bandwidth parameter for each domain:

$$K_{h_t,h_s}(x)=\frac{L}{h_s h_t}k\left(\left\|\frac{x^s}{h_s}\right\|^2\right)k\left(\left\|\frac{x^t}{h_t}\right\|^2\right) \tag{1}$$

where x^s is the spatial part and x^t is the temporal part of the feature vector *x*. $k(x)$ is a common kernel profile used in both spatial and temporal domains, and *L* is the corresponding normalization constant. MDGKT requires a pair of bandwidth parameters (h_s, h_t) to control the size of the kernel, as described in more detail in [27].

B. Self-adaptive Gaussian mixture model (SAGMM)

An illumination-invariant change detection model (IICDM) is a process of identifying illumination variation over time. Changing image illumination causes problems for many computer vision applications operating in unconstrained environments and especially affects background subtraction methods. The most trivial approach for IICDM is the subtraction of intensities of two sequential video frames. The main disadvantage of such a simple method is its sensitivity to noise. Alternative intensity estimation methods were compared in [28]. Usability was evaluated with background classification. An accurate non-iterative estimate of the apparent gain factor is reported by experimentally comparing six algorithms. According to simulation results many algorithms performed well, with the best performance demonstrated for the Median of Quotient (MofQ), both with and without outlier removal.

For a pixel I_c in the current image, the MofQ global illumination change factor g between the current image I_c and the reference image I_r is defined as:

$$g = median\left(\frac{I_r}{I_c}\right) \tag{2}$$

Our SAGMM algorithm is based on the GMM presented by Zivkovic and Heijden [8] [9], herein referred to as ZHGMM. The advantage of ZHGMM over other GMM variants is that recursive equations are used to constantly update the parameters of the GMM and to simultaneously select the appropriate number of components for each pixel.

SAGMM adds an adaptive schedule to the recursive ZHGMM learning procedure. This method uses the global illumination change MofQ factor g between the learnt background I_r and the current input image I_c, a dynamic learning rate α , and a counter c for each Gaussian component in the mixture model. The factor g keeps track of how the global illumination changes and the counter c keeps track of how many data points have contributed to the parameter estimation of the Gaussian.

A squared Mahalanobis distance (*SMD*) from the current pixel to the *m*th Gaussian component is calculated as:

$$D_m^2(I_c) = \hat{\delta}_m^T \sum_m \hat{\delta}_m \tag{3}$$

where $\hat{\delta}_m = g \cdot I_c - \mu_m$, μ_m is the estimated mean value of the *m*th GMM component, and $\sum_m$ is the estimated covariance matrix. A sample is "close" to a component if the *SMD* is less than a threshold (T_{SMD}), resulting in updated values for the estimated weight, mean, variance and count; otherwise a new component is generated with initial values. The recursive update formulae can be found in [26].

These modifications provide two significant advantages for the algorithm: i) if an object becomes stationary and then part of the background, it does not destroy the existing (underlying) model of that background - the original background values remain in the GMM. If the object then moves, the distribution describing the previous background still exists with the same estimated mean and variance. This is also useful, for instance, for detecting a large vehicle with uniform surface color, where although the object may be moving, the pixel value does not change because of the uniformity of the object surface; ii) the dynamic learning rate and *SMD* calculation (Eq.(3) and factor g) compensate for global illumination changes. Experimental results are presented to demonstrate that SAGMM is stable to sudden illumination changes and copes with vehicles that become stationary. We note that it is also possible to deal with the typical ghosts left behind when a vehicle that had become stationary finally moves on, using an edge-based method as reported in [48], but that only deals with that phenomenon and not the background model as a whole and also we wanted to minimize the number of variables in the evaluation of performance.

C. Shadow removal

RGB color information is useful for suppressing shadows and highlights, by separating color information from brightness information. Without shadow and highlight suppression, the size and shape of foreground objects (i.e. vehicles) can be significantly distorted, which may lead to detection and classification errors. Horprasert et al. [29] describe the deviation between the expected RGB values of a pixel and the measured RGB values as a distortion, such as could be caused by the shadow cast by a foreground object onto a true background pixel. They decompose this distortion measurement in RGB space into two components, brightness distortion and chromaticity distortion. It should be noted that this approach primarily addresses the detection of shadow and highlight pixels that are associated with foreground detection, and is not applied to other shadows regions in the image frame that result from shadows cast by buildings or trees etc. and form part of the modeled background.

The observed color vector is projected onto the expected color vector, and the *i*th pixel's brightness distortion b_i is a scalar value (less than unity for a shadow) describing the fraction of remaining 'brightness'. This may be obtained by minimizing

$$\phi(b_i) = (I_i - b_i E_i)^2 \tag{4}$$

where $I_i = [I_{Ri}, I_{Gi}, I_{Bi}]$ denotes the *i*th pixel value of current image in RGB space, $E_i = [\mu_{Ri}, \mu_{Gi}, \mu_{Bi}]$ represents the *i*th pixel's expected (mean) RGB value in the SAGMM background. The solution to equation (4) is a b_i value equal to the inner product of I_i and E_i, divided by the square of the Euclidean norm of E_i.

Balancing the color bands by rescaling the color values by the pixel standard deviation $\sigma_i = [\sigma_{Ri}, \sigma_{Gi}, \sigma_{Bi}]$, the brightness and chromaticity distortions are computed from:

$$b_i = \frac{g\sum_{\kappa\in[R,G,B]} I_{\kappa i}\mu_{\kappa i} / \sigma_{\kappa i}^2}{\sum_{\kappa\in[R,G,B]} [\mu_{\kappa i} / \sigma_{\kappa i}]^2} \tag{5}$$

$$CD_i = \sqrt{\sum_{\kappa\in[R,G,B]} (gI_{\kappa i} - b_i\mu_{\kappa i})^2 / \sigma_{\kappa i}^2} \tag{6}$$

Then a pixel in the foreground segmentation is classified as either a shadow or highlight reflection on the true background as follows:

$$\begin{cases} Shadow & CD_i < q_1 \; and \; q_2 < b_i < 1 \\ Highlight & CD_i < q_1 \; and \; \; b_i > q_3 \end{cases} \tag{7}$$

q_1 is a threshold value used to determine the similarities of the chromaticity between the SAGMM and the current observed image. If there is a case where a pixel from a moving object in the current image contains a very low RGB value, then this dark pixel will always be misclassified as a shadow, because the value of the dark pixel is close to the origin in RGB space and all chromaticity lines in RGB space meet at the origin. Thus a dark color point is always considered to be close or similar to any chromaticity line. The threshold q_2 is introduced for the normalized brightness distortion to avoid this problem. The threshold q_3 is introduced to detect highlight reflections. The automatic threshold selection method of Horprasert et al. [29] was used to select appropriate values for q_1 and q_2 ($0 < q_1, q_2 < 1.0$, $q_3 > 1.0$).

IV. Vehicle Detection

There are several key considerations when implementing a vehicle detection algorithm, and they vary depending on the specific task. For traffic flow statistics, it is essential to count each vehicle only once. To ensure this condition, a detection zone and detection gate are employed. The gating logic is comprised of three fiducial lines: StartLine (SL), MiddleLine (ML) and EndLine (EL) [30]. Line detectors are sensitive to miss-detection as a consequence of the ragged edge of a noisy vehicle silhouette boundary. To minimize this effect, the detection lines have a finite thickness (D_T) to ensure a stable detection of the vehicle when it intersects the line. The separation D_s between detector lines depends on the traffic speed, video capture rate and size of the smallest detected object. The detector operates bidirectionally to accommodate two-way traffic flow. They are placed at locations where vehicles are most clearly visible with minimal occlusion, i.e. usually closest to the camera. The traffic motion direction is obtained using a Kalman tracker. A separate detector is allocated to each lane to handle the measurements for each traffic stream.

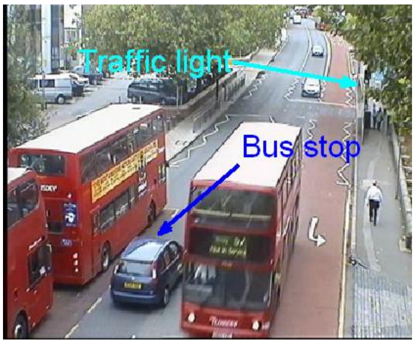
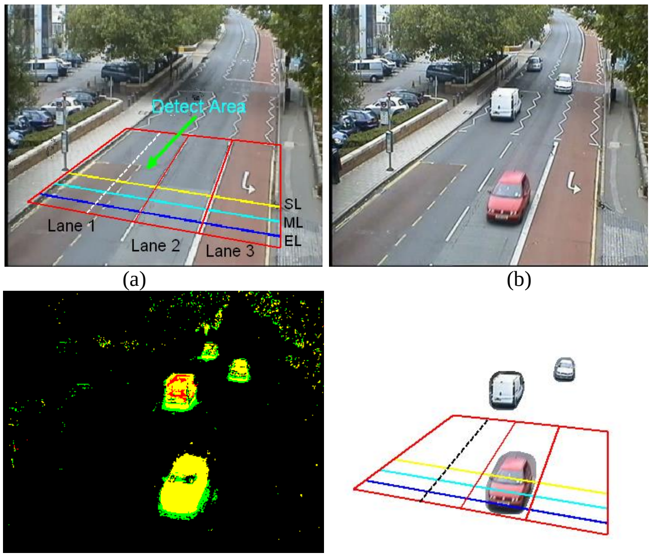


Fig. 2. Detection environment.



(a) (b)

(c) (d)

Fig. 3. Vehicle detection: (a) modeled GMM background image and detection zone gates (b) current input image. (c) background subtraction results: black pixels represent the modeled background, foreground object (yellow), shadow (green) or reflection highlights (red); (d) foreground image created by extracting the pixels from the original frame using the final foreground object mask, morphologically dilated.

Figure 2 shows the complexity of the scene. There is a bus stop in the detection zone, and a pedestrian crossing near the detection zone. If a bus stops, the background changes dramatically, and following vehicles will change their lane in the detection area to overtake. Partial or full occlusion commonly occurs during such conditions. Similarly, if pedestrians are crossing the road, there may be a long vehicle queue in the detection zone. Such situations are common in urban environments, and present a significant challenge for robust vehicle detection and classification.

Figure 3 illustrates the object detection procedure. Shadows, road reflection and reflection highlights are suppressed, followed by a post-processing binary morphological opening using a diamond-shaped structuring element that removes noise and fills holes to create a binary foreground object detection mask for each vehicle. Small blobs with an area significantly below the minimum size of the smallest object class (i.e. A_{min}, a motorcycle/ bicycle) are discarded from further consideration.

A valid vehicle detection requires that a detector line (SL, ML and EL) is occupied only when the proportion of pixels intersecting the detection line is above a threshold (D_L), otherwise it is deemed as unoccupied. This threshold is chosen as a tradeoff between detecting small vehicles (such as bicycles and motorbikes) whilst being insensitive to small blobs associated with noise. To ensure that vehicles are only counted once, the detector considers a vehicle to be "present" only when both SL and ML are occupied and EL is unoccupied (for traffic moving towards the camera, i.e. lanes 2 and 3). For these lanes, a vehicle is said to be "leaving" the zone when ML and EL are occupied and SL is unoccupied. A vehicle is counted only when it changes from the "present" state to the "leaving" state. This is reasonable in congested situations and for stationary traffic so that the detector does not over-count. It is only necessary to swap SL and EL to account for vehicles in the traffic stream moving away from the camera (e.g. lane 1 in Figure 3(a)). To detect vehicles overtaking a stationary bus at the stop, lane 1 is split into two detection lanes.

V. Vehicle Type Classification

Reliable classification requires distinctive and stable features that can be robustly extracted from the image data. In this study, vehicles are classified using size and shape features measured from the silhouette, combined with a multi-scale representation of the internal structure. Classification uses SVM with a polynomial kernel, constructing a multi-class classifier using a one-vs-all strategy, in which the *m*th classifier constructs a hyperplane between class m and the *N*-1 other classes.

The SVM classifier is first trained on a set of labelled exemplars for which the vehicle silhouette has been manually delineated and labelled. An effective training set typically requires a large quantity of training data that is representative of the object classes under consideration. However, manual annotation of the object silhouette is slow and laborious, so we have developed an approach that eliminates the need to manually extract many silhouettes, minimising the number of annotations required [31].

A. Feature Extraction

1) Measurement-based features (MBF)

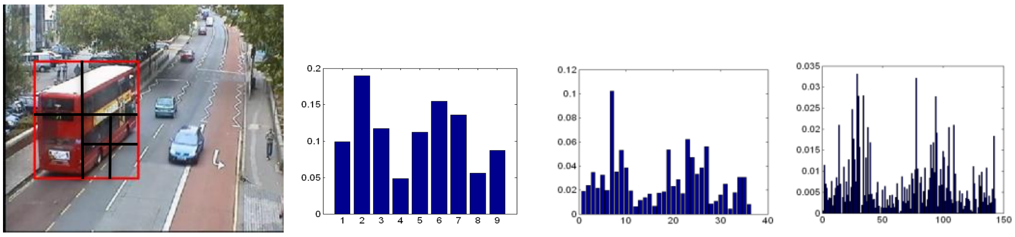
Each vehicle silhouette is represented by a 13-component feature vector, comprising measures of size and shape from the binary silhouette and encompassing bounding box (width, height, perimeter and area), circularity (dispersedness, equivdiameter), ellipticity (length of major and minor axis, eccentricity), and shape-filling measures (filled area, convex area, extent, solidity).

2) Structural features

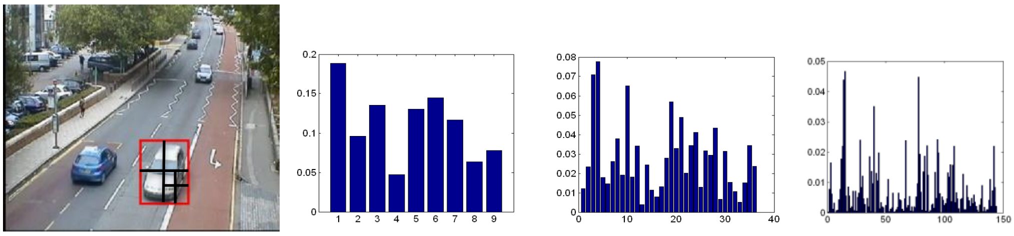
An important feature of vehicles is the geometric detail of their design. To capture this detail we use a pyramid of histogram of gradient orientations (PHOG). HOG represents local object appearance and shape by the distribution of local intensity gradients or edge directions, without precise knowledge of their relative position. This is implemented by dividing the image window into small spatial regions (cells), and for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over all the pixels of the cell. The combined histogram entries form the representation [32].

PHOG was first proposed by Bosch et al. [33] and has been successfully applied to object recognition, human expression recognition and image classification [34] [35]. As a spatial shape descriptor, it can represent the statistical information of global shape and local shape (in a sub-region), which is effective for object recognition. The local shape is captured by the distribution over edge orientations within a region, and the spatial layout by tiling the image into regions at multiple resolutions. The descriptor consists of a histogram of orientation gradients over each image sub-region at each resolution level of the detection bounding-box, normalized (summed to 1) for each level. For local shape representation there are two alternative computations of PHOG: edge-based PHOG (EPHOG) is computed using the Canny algorithm and represented by a histogram of edge orientations within an image region and its sub-region; intensity-based PHOG (IPHOG) is represented by the distribution of local intensity gradients, without precise knowledge of the corresponding edge point. For vehicle classification IPHOG is found to give a marginally better classification performance [36] and is used in the current work.

For vehicle classification, we compute HOG at 3 spatial levels and 9 orientation bins, evenly spaced over 0º - 360º, normalizing them for each level. The best number of levels and orientations were found empirically. The resulting vector is constructed by concatenating these histograms into a 189 element vector (9+4×9+4×4×9). Figure 4 shows an input image with the cells and the shape spatial pyramid



bus level 0 level 1 level 2



car level 0 level 1 level 2

Fig. 4. An input image and the shape spatial pyramid representation of IPHOG for a bus and a car over three spatial scales: 9 features (level 0), 36 features (level 1), 144 features (level 2).

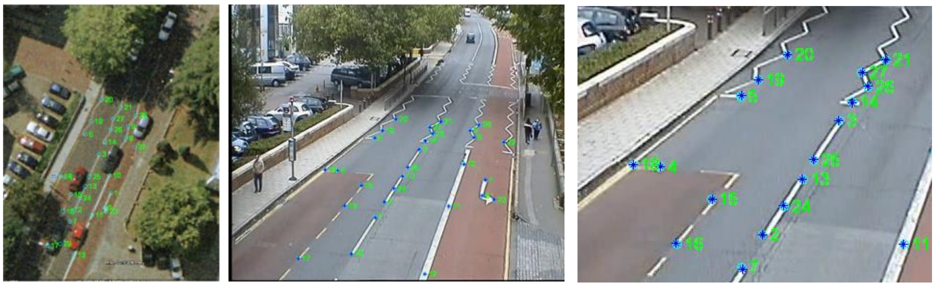
The final feature vector that represents each vehicle silhouette is formed by concatenating the MBF and IPHOG feature vectors.

B. Semi-automatic annotation of ground truth

This section describes the generation of the ground-truth that is used to train the final vehicle classifier and for performance evaluation. The conventional approach to creating the ground-truth data would be to manually trace the silhouette of a set of vehicles, assigning the appropriate class label to each silhouette traced. Object features extracted from the traced silhouettes would then be used to train a classifier for vehicle type classification. High classification performance necessitates a comprehensive set of training examples.

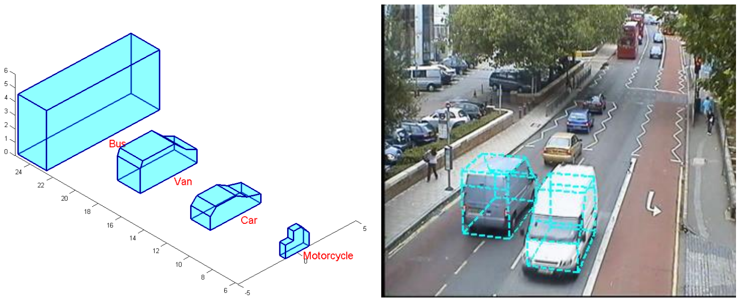
Our alternative uses two classification steps that eliminate the need to trace the silhouettes, and requires a minimal number of class labels to be manually assigned. In the first step, a wireframe model of each vehicle class (see fig. 6a) is projected onto the road plane to create a synthetic silhouette of a vehicle instance using parameters derived from camera calibration. To realistically represent the variety and range of real vehicle silhouettes, a random perturbation is added to the vertices of the wireframe model, which is then projected onto the near ground-plane at an orientation aligned with the lane direction (see examples in Section VI.D, Fig. 9). Each silhouette is represented by the MBF values and this synthetic silhouette dataset is used to train an initial classifier using the multi-class SVM [30, 37].

In the second step, the classifier trained using only MBF is applied to predict the label of real vehicle silhouettes that have been automatically detected by SAGMM. Only a small proportion of resulting labeled data, located in the low confidence region of the classifier (i.e. the margin) need to be manually corrected. The magnitude of the classification score for each sample is used as a measure of confidence in the initial labeling, since examples far from the separating hyperplane are presumed more likely to be classified correctly. Data in the low confidence (most informative) region, which are close to the decision boundary, may need their annotation to be manually corrected. Since only the support vectors are used by SVM to determine the decision boundary between each pair of classes, these low confidence samples are the most likely candidates to be used as the support vectors. The resulting labeled data is then used as the training set for the final classifier design. Results reported in the next section demonstrate that this provides a significant reduction in the effort to create a labeled dataset for training purposes [38].



(a) (b) (c)

Fig. 5. a) Plan view image; b) calibration reference image (middle) and c) zoomed portion of b). Cyan circles and index numbers indicate the corresponding points and the blue asterisks indicate the re-projected points.



(a) (b)

Fig.6. 3D wireframe models [17] with true size (a) and samples of their projections on the ground plane (b).

The camera calibration parameters used in step1 above are generated using a novel approach that manually matches ground-plane features visible in the image data with the same features visible in calibrated images available from Google Earth (see fig. 5). The resulting parameters are found to be sufficiently accurate for the purpose of re-projecting the wireframe models to create synthetic silhouettes, with an average re-projection error of 0.97 pixels (see fig. 5 and [37]). One potential drawback with the method is that the ground-plane features (in our case, the endpoints of white lines painted onto the road for traffic control purposes) in the Google Earth view may have changed over time. However, this was not an issue for the results presented here.

VI. EXPERIMENTS

This section quantifies the impact of each stage of the video analysis system depicted in figure 1. The performance of each component is measured by the vehicle detection rate (DR) and the classification rate (CR), and includes false positive (FP) and false negative (FN) detection rates

A. Parameter settings

The first set three experiments investigate the set values for various parameters and their sensitivity, examining the impact of the spatio-temporal filtering, the learning rate of background estimation and the parameters of the detection zone logic. These experiments are performed using a leave-one-out protocol, where each parameter is separately tuned by assessing its impact on the overall performance of the system to robustly detect and classify vehicles. The assessments were performed on 50,000 video frames (approx. 33 minutes) recorded from a single pole-mounted roadside camera in daytime on a busy road in a local town center using a capture rate of 25 frames per second and an image size was 352× 288. The sequence contained over 7449 vehicles that were manually identified (by location and type) to generate the ground truth. As mentioned earlier and in **Error! Reference source not found.**, it is unfortunate that in this field there are virtually no public datasets of urban traffic of sufficient length and variety of traffic and atmospheric conditions, nor public source or executable code to compare like with like. Nevertheless, the results reported here compare favorably with those in similar work [47]. We think that the data used here (that may be made available for research purposes by contacting the corresponding author, subject to data privacy restrictions) is a positive step toward investigating robustness. Based on the promising results here we are currently discussing with the local town the feasibility of an extended dataset of 24-hour videos for a wide range of conditions (daylight, darkness, fog, rain, harsh sunshine, etc.). There are important challenges such as privacy issues and the significant manual effort in annotating such an extensive dataset.

The first experiment considers the efficacy of the spatio-temporal filtering, comparing it with conventional spatial filtering. Table I presents detection and classification rates for the 50,000 video frames with the following: i) no filtering, ii) a 3x3 spatial (Gaussian) filter, iii) a 5x5 spatial (Gaussian) filter and iv) a 3x3x3 spatio-temporal (Gaussian) filter (MDGKT).

TABLE I. IMPACT OF PRE-FILTERING ON VEHICLE DETECTION AND CLASSIFICATION PERFORMANCE.

	DR	FP	FN	CR
No filter	0.9117	0.0470	**0.0883**	0.8912

3x3 spatial filter	0.8928	0.0585	0.1072	0.8438
5x5 spatial filter	0.8928	0.0585	0.1072	0.8438
3x3x3 spatiotemporal filter	**0.9801**	**0.0114**	0.1099	**0.9025**

The two spatial filters produce identical performance for both detection and classification, and are both worse than no filter on all four measures. The best performance is achieved using a 3x3x3 spatio-temporal filter, which had a detection error rate of only 2% and a vehicle classification rate of 90.25%, though the lowest false negative rate is found when filtering is not applied.

The next experiment considers the learning rate parameter, which determines how quickly the background model will adapt to change. Table II shows the results of varying the learning rate parameter over periods ranging from 2-20 seconds (at 25 fps). The results show the strongest performance for a value of 100 frames (i.e. 4 seconds).

TABLE II. IMPACT OF LEARNING RATE ON VEHICLE DETECTION AND CLASSIFICATION PERFORMANCE.

Learning rate (frames)	DR	FP	FN	CR
50	0.8886	0.0627	0.1114	0.8430
100	**0.9801**	**0.0114**	**0.0199**	**0.9025**
250	0.8760	0.0404	0.1240	0.8535
500	0.8552	0.0148	0.1448	0.8564

The final experiments consider the performance of the detection zone, which is critical to ensuring vehicles are correctly counted. Table III considers the sensitivity of the threshold applied to detect valid vehicles entering the detection zone, rejecting noisy and partial detections. The results indicate that a rejection criteria based on the lower threshold value (1/10) results in the best detection and classification performance and the lowest FP and FN rates.

TABLE III. IMPACT OF TARGET/LANE WIDTH RATIO ON VEHICLE DETECTION AND CLASSIFICATION PERFORMANCE.

Lane width ratio	DR	FP	FN	CR
1/10	**0.9801**	**0.0114**	**0.0199**	**0.9025**
2/10	0.8928	0.0306	0.1072	0.8906
3/10	0.9095	0.0265	0.0905	0.8742

Table IV shows how the detection rate (DR) performance changes across various settings of the detection line width and the separation distance between each detection line. The combination of a line width of 5 pixels and a line separation of 16 pixels results in the best detection rate.

TABLE IV. IMPACT OF DETECTOR LINE SEPARATION AGAINST DETECTOR LINE WIDTH ON VEHICLE DETECTION RATE

Separation / Width	1	5	10
8	0.8760	0.8705	0.8691
16	0.8928	**0.9801**	0.8816
24	**0.9248**	0.9220	**0.9192**

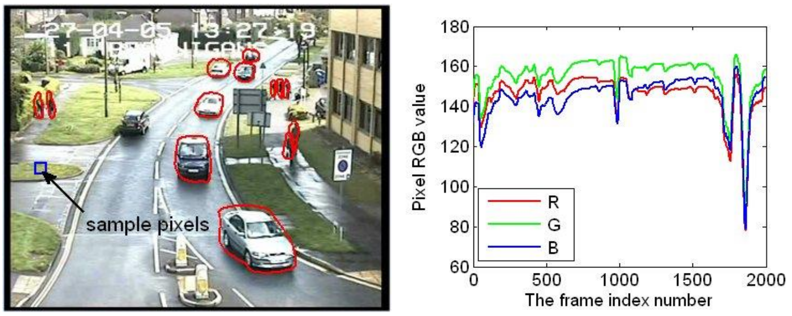
Finally in this section, the results of using a gating logic that employs only two detection lines was assessed. In this case, although the detection rate is high (0.9610), the false positive rate is also very high, indicating that many vehicles are counted more than once. The final implementation used 3 detection lines achieving the highest detection rate with the lowest false positive and negative rates.

TABLE V. IMPACT OF NUMBER OF DETECTION LINES ON VEHICLE DETECTION RATE

Detection lines	DR	FP	FN
2	0.9610	0.9554	0.0390
3	**0.9801**	**0.0114**	**0.0199**

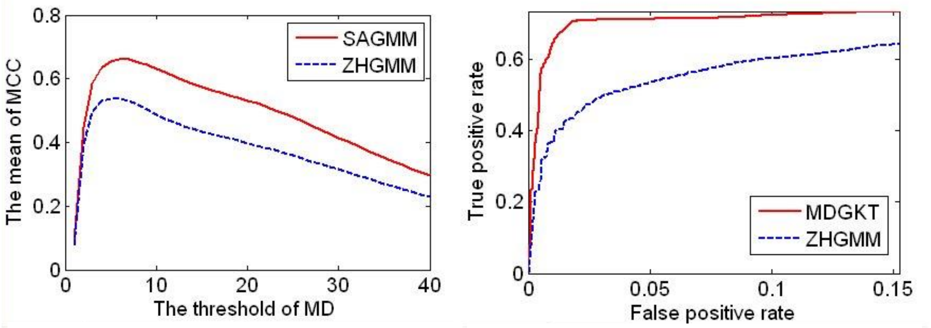
B. Evaluation of segmentation performance

The aim of this experiment is to evaluate the performance of segmentation results obtained from the online dynamical learning rate and global illumination background model adaptation of SAGMM. A comparison is given between the performance of SAGMM and the original algorithm (ZHGMM) as an example of the state-of-the-art. In this experiment, the input data is a traffic video captured under broken cloud conditions after a rain shower, resulting in significant variations in the illumination levels and strong road surface reflection (taken from the iLids dataset [40]). A total of 2000 frames are analyzed from this video. An annotated sample image from the test video is shown in Figure 7(a). The average values of red, green and blue channel in the blue box, where no intruding object appears over time, is shown in Figure 7(b). It can be seen that the intensity variation is large, ranging from values 79-165 for R, G and B, over a period of 80 seconds. The foreground ground-truth objects (red silhouettes) were created using Viper [41]. The car in the left lane remains stationary throughout the sequence.



(a) (b)

Fig. 7. (a) Example image, (b) the variation of average value of red, green, blue channel sampled from region indicated by the blue box over 2000 frames.



(a) (b)

Fig. 8. (a) The variation of MCC corresponding to different thresholds of SMD. (b) ROC curve.

Three performance metrics are used to evaluate the segmentation performance of the SAGMM algorithm and compare it with that of ZHGMM: the True Positive Rate (TPR$_{bg}$), the False Positive Rate (FPR$_{bg}$) and the Matthews Correlation Coefficient (MCC$_{bg}$), computed by:

$$MCC_{bg} = \frac{TP_{bg} \times TN_{bg} - FP_{bg} \times FN_{bg}}{(TP_{bg} + FP_{bg})(TP_{bg} + FN_{bg})(TN_{bg} + FP_{bg})(TN_{bg} + FN_{bg})} \quad (8)$$

The receiver operating characteristic (ROC) is used to analyze the model's performance. In our case, the two parameters of most significance are the threshold *SMD* (Eq.(3)) and the learning rate α. For a typical fixed α =0.001, an object will be stationary for a maximum of 105 frames before it is merged into the background. MCC$_{bg}$ is used to explicitly select the probability tradeoff between the TPR$_{bg}$ and FPR$_{bg}$, where a larger value is better. The variation of MCC$_{bg}$ corresponding to different *SMD* thresholds is given in Figure 8(a). It shows that the SAGMM yields higher values of MCC$_{bg}$ than ZHGMM for all thresholds *SMD* $\in$ [1, 44], (the increment of *SMD* is 1). Each MCC$_{bg}$ value is the average over the entire video sequence (of which only 492 frames include the moving vehicle). Using the optimal threshold value of 6, Figure 8(b) shows the ROC curve for the learning rate α_{frame} varying from 2 to 200 frames. The α_{frame} is converted to α by the formula $\alpha = 1 - e^{\ln 0.9 / \alpha_{frame}}$. The area under the ROC curve for SAGMM and ZHGMM are 0.731 and 0.632, respectively, indicating that the overall learning rate for SAGMM is better than for ZHGMM.

C. Evaluation of vehicle detection

The experimental data is taken from 5 hours of video recorded from a single pole-mounted roadside camera in daytime on a busy road in the local town centre. The capture rate is 25 frames per second and the image size was 352×288. Weather conditions ranged from dry and overcast (approximately 1 hour), to periods of light and heavy rain, after which the road surface was wet and shiny. A total of 7456 vehicles were manually observed over this period (car: 6208, van: 725, bus: 330, motorcycle: 186; lorry: 7). Automatic detection using SAGMM found a total of 7191 vehicles (that from the ground truth correspond to car: 6035, van: 690, bus: 313, motorcycle: 153). The threshold used to select the background model (T_{SMD}) and the thresholds for shadow and highlight detection used empirically-derived values (see [45]) with T_{SMD}=16 and q_1=0.5, q_2=0.01, q_3=1.2.

The evaluation results are given in Table VI, which shows counts of true positive, false positive and false negative and the associated rates. The overall detection rate (DR) is 7191/7456 = 96.45%, an overall FPR = 123/7456 = 1.65%, and FNR = 258/7456 = 3.46%. In good weather (dry, overcast), the DR rises to 97.30%.

TABLE VI. EVALUATION OF VEHICLE DETECTION RESULTS ON 5 HOUR SEQUENCE.

	Car	Van	Bus	Motorcycle
Total number	6208	725	330	186
Detected vehicles	6035	690	313	153
FP	75	18	19	11
FN	173	35	17	33
FPR	0.0121	0.0248	0.0576	0.0591
FNR	0.0279	0.0483	0.0515	0.1774
DR	0.9721	0.9517	0.9485	0.8226

The output of the vehicle detection results from SAGMM is a binary object mask (blob). The blob centroid is tracked using a constant velocity Kalman filter [42]. The state of the filter is the centroid location and velocity, $s = [c_x, c_y, v_x, v_y]^T$, and the measurement is an estimate of this entire state, $y = \hat{s} = [\hat{c}_x, \hat{c}_y, \hat{v}_x, \hat{v}_y]^T$. The data association problem between multiple blobs is solved by comparison of the predicted centroid location with the centroids of the detections in the current frame. The blob with its centroid closest to the predicted location is chosen as the best match for the track.

D. Semi-automatic data annotation

A total of 3600 synthetic silhouette samples (car: 1482, van: 927, bus: 878, motorcycle: 313) were created using a random variation determined by zero mean Gaussian noise with a standard deviation of 5.0. Figure 9 shows examples of synthesized silhouettes that are generated by randomly perturbing the vertices of the four wireframe models (shown in Fig. 6d). This creates variation in the size, shape, position and orientation of the projected vehicle wireframe and hence the resulting vehicle silhouette. A closed convex polygon around their extremal boundary is constructed to create the synthetic MBF. MBF features are computed for each silhouette and used to train the first stage of the classifier. For stage 2, the stage 1 classifier was applied to 8875 real vehicle silhouettes (1000 cars, 516 vans, 135 buses and 124 motorcycles tracked over five consecutive image frames) detected using the method described in Section IV. A 202-dimensional feature vector is constructed for each silhouette, comprising the 13 MBF and 189 IPHOG features, and these data are used to train a multi-class SVM (parameter setting for the polynomial kernel used with SVM was d=3, and the penalty constant C=500).

The classifiers were evaluated using 10-fold cross-validation. The means of the ACC over the four classes is 97.57%. Of the 8875 training samples, a total of 443 required manual re-labeling, representing only 5% of the full training set and a significant saving in the time required for manual annotation. More importantly, the classifier is trained with real rather than manually-traced silhouettes, and hence is more representative of the real video data.

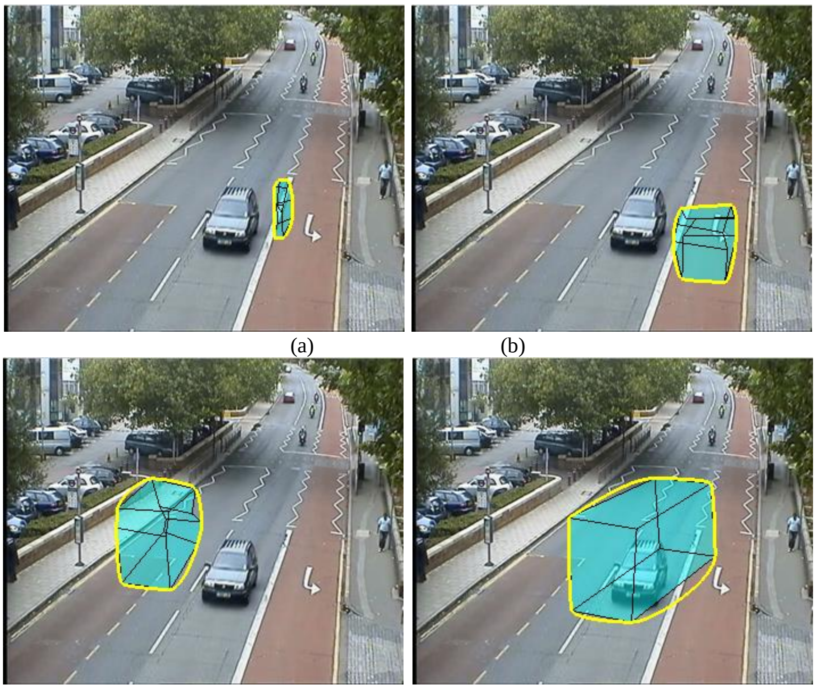


Figure 9. Synthetic vehicle data set for training SVM classifier, yellow curve shows the convexhull silhouette. (a) motorcycle, (b) car, (c) van, (d) bus.

E. Evaluation of vehicle classification

A binary morphological opening (diamond-shaped structuring element, 4 pixels in extent) is used to smooth the boundary and fill holes in the detected silhouettes. Small blobs with an area of less than 100 pixels are discarded. The training model obtained in the previous section (VI.C) was used to classify each foreground blob detected by SAGMM. The final class label is assigned by a majority vote over five separate detections of the tracked vehicle from five consecutive image frames. A snapshot of the classification output is shown in Figure 10.

The track class label is computed at each frame, but the final label is assigned by a majority voting scheme that considers the entire track to make a decision on class type, rather than employing a single frame that could be corrupted by different noise sources. This provides multiple instances of the same vehicle, each of which is independently classified. There are insufficient samples in the lorry category to consider these for classification.

The experimental results show that the majority voting scheme over five consecutive frames improves the classification accuracy. If the confidence in the voting procedure is defined as *l*/5, where *l* is the number of majority labels over 5 consecutive frames, some 14.34% of vehicles were detected with a confidence less than 1.0, indicating that the labeling was inconsistent. For only one observation the ACC = 94.51%, rising to 95.23% when the best out of three are chosen, and then to 95.70% for the best out of five. Tow snapshots of GUI of the system are given in Figure 11.

The confusion matrix is given the table VII. The classification accuracy (ACC) is 95.70%. In good weather (dry, overcast), the ACC rise to 96.38%.

TABLE VII. CONFUSION MATRIX.

	Car	Van	Bus	Motorcycle
Car	**5825**	76	116	18
Van	26	**608**	55	1
Bus	5	3	**302**	3
Motorcycle	4	0	2	**147**

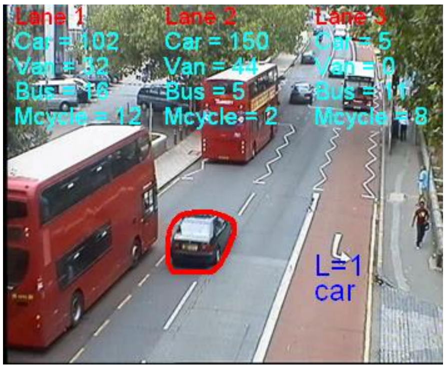


Fig. 10. A snapshot GUI of AutoVDC showing occurrence of vehicle types associated with each traffic lane. Note that Lane 3 is primarily allocated for bus and taxi usage, though bicycles will typically be located next to the kerb.

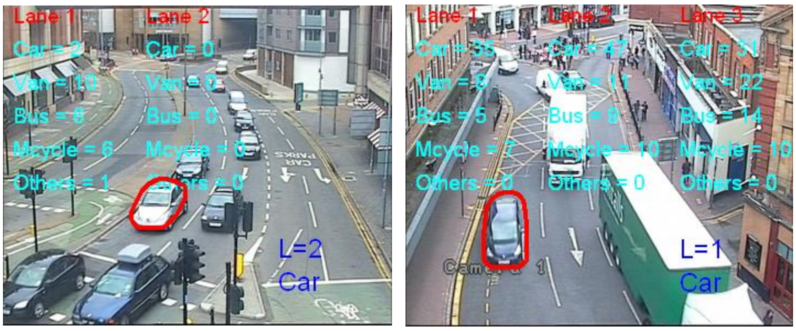


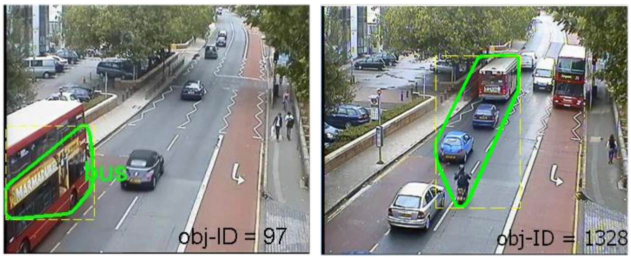
Fig. 11. Snapshot GUI from two other camera locations.

Manual analysis of the detection failures of the systems were primarily identified with the following conditions:

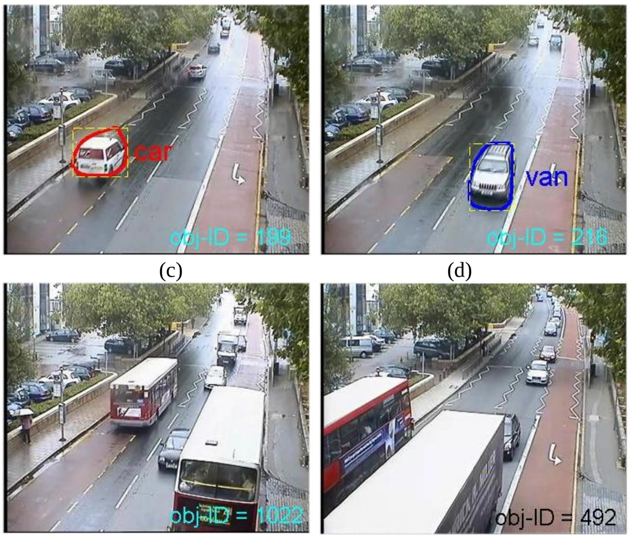
1. Under-segmentation of light-coloured vehicles from road surface under bright illumination;
2. Stationary vehicles queuing in the detection zone;
3. Stationary buses at the bus stop occupy the detection lines in the detection zone, interfering with the detection gate logic that expects only single vehicles per lane;
4. Reflections from the road surface when the road is wet.

Similarly, errors in vehicle classification could be mainly attributed to the following:

5. Vehicles in close proximity (e.g. queuing and dense traffic) in the detection zone, resulted in two vehicles being detected and classified as one;
6. Over-segmentation of small vehicles due to illumination and wet roads



(a) (b)



(c) (d)

(e) (f)

Fig. 12. Examples of failures: a-b) in detection, c-d) in classification and e-f) complete misses due to heavy occlusion. The bold curve is silhouette of the detected blob in different categories (bus: green, car: red and van: blue). The yellow rectangle is the corresponding bounding box.

Figure 12 presents examples of the failure of the detection and classification modules. Figure 12(a) shows where a car passes a stopped bus that has been stationary for a long time (more than 100 frames), and fully absorbed into the modeled GMM. In this case, the reflection of the car on the side of the bus has been detected by the background subtraction. Although the ground truth is a car, car is undetected and the detection result is the bus. Figure 12(b) illustrates another failure of the detector: where an extended blob is detected due to low inter-vehicle separation and reflections from the wet road surface. This large blob has been classified as a bus. These two examples illustrate classification errors created from errors in the detection module rather than the classification module. Figure 12(c) and (d) show misclassification errors between a car and a van. Because cars and vans have very similar features from this viewpoint they are more likely to be confused by the classifier. Finally, figures 12 (e) and (f) show the failure of the foreground extraction module, where one vehicle is occluded by another vehicle, and hence is not detected.

VII. DISCUSSION

The average vehicle count for this 0.55 hour sequence is 6778 vehicles per hour per lane (except bus lane) from a total of 7456 vehicles, a relatively low vehicle density. However, vehicle flow rates are impacted by the pedestrian crossing, which operates on a sporadic basis, causing traffic queuing into the detection zone and so the distribution of traffic densities and speeds will be heavily influenced by this, rather than from the average estimate. It will also be influenced by the bus stop, which also causes significant change to a normal flow behaviour.

The detection failures in Fig. 12(a) and (b) might benefit from a different approach to segmentation, such as the mean-shift algorithm [43], level set methods [44] or model-based algorithm using mixtures of multiscale deformable part models [45]. For instance, the method proposed in [43] could be used to detect partially occluded vehicles depicted in Fig 12(e). We have experimented with such algorithms to further process background subtraction results, but they are computationally expensive or have slow convergence characteristics and hence are inappropriate for a real-time system. Alternatively, the failures shown in Fig. 12(e-f), could be solved using a multi-camera system [46], but this would require much higher densities of deployed CCTV cameras.

Comparing our classification results with the published results of other researchers is unproductive. Of the papers reviewed in Section II, few perform both automatic detection from video and classification of the resulting blobs. In addition they use different categories or classes of vehicles. The lack of a common video dataset further complicates the task of trying to directly compare our results with those from other researchers, captured under different environments and conditions. For instance, in [47] Mithun et al report an average classification performance of 90% for seven vehicle types classified according the number of ground wheels. However, the headline performance of our classifier (~95%) is comparable to the best of those reported.

VIII. Conclusions and Future Work

Acquisition of reliable vehicle counts and classification data is necessary to establish an enriched information platform and improve the quality of ITS. The approach proposed in this paper is a hybrid algorithm, employing modules for background subtraction, foreground extraction and vehicle detection. The improved background subtraction method is demonstrated to alleviate the negative impacts from camera vibration, shadow and reflection highlights and both sudden and gradual changes in the illumination. In the vehicle classification module, a Kalman tracker and SVM were combined to yield improved accuracy. Experimental results show that the highest performance resulted from using SVM applied to the MBF+IPHOG features, classifying the foreground blobs using a majority vote over 5 consecutive frames. The results demonstrate a vehicle detection rate of 96.45% and classification accuracy of 95.70% under varying illumination and weather conditions.

Further work will be aimed at identifying features to minimize this ambiguity, and to extend the approach to operate in a view independent way, dense traffic scenarios and apply the best classifier design to assess its performance when applied to automatically segmented binary silhouettes of vehicles detected from video sequences.

Acknowledgements

The authors gratefully acknowledge the Royal Borough of Kingston for providing the video data. The authors also thank the anonymous reviewers for their detailed comments. S.A. Velastin is grateful to funding received from the Universidad Carlos III de Madrid, the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement nº 600371, el Ministerio de Economía y Competitividad (COFUND2013-51509) and Banco Santander

References

[1] S. Messelodi, C. M. Modena, and M. Zanin, "A computer vision system for the detection and classification of vehicles at urban road intersections," *Pattern Analysis & Applications*, 2005, 8(1-2):17–31.

[2] R.J. Radke, S. Andra, O. Al-Kofahi and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Transactions on Image Processing*. 14(3): 294-307, 2005.

[3] T. Bouwmans, F. E. Baf, B. Vachon, "Background modeling using mixture of Gaussian for foreground detection: A survey," *Recent Patents on Computer Science*, 1(3):219-237, 2008.

[4] C. Stauffer, W. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8): 747-757, 2000.

[5] P. W. Power, J. A. Schoonees, "Understanding background mixture models for foreground segmentation," *Proceedings of Image and Vision Computing*, New Zealand, November, 2002, pp. 267-271.

[6] D-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5): 827-832, 2005.

[7] N. Greggio, A. Bernardino, C. Laschi, P. Dario, J. Santos-Victor, "Self-adaptive Gaussian mixture models for real-time video segmentation and background subtraction," *10th International Conference on Intelligent Systems Design and Applications (ISDA)*, pp.983-989, 2010.

[8] Z. Zivkovic and F. van der Heijden, "Recursive unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5): 651-656, 2004.

[9] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, 27(7): 773-780, 2006.

[10] N. Martel-Brisson and A. Zaccarin, "Learning and removing cast shadows through a multidistribution approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1133-1146, 2007.

[11] S. L. Zhao, H. J. Lee, "A spatial-extended background model for moving blob extraction in indoor environments," *Journal of Information Science and Engineering*, 25, 1819-1837, 2009.

[12] A. Prati, I. Mikic, M. M. Trivedi, R. Cucchiara, "Detecting moving shadows: algorithms and evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7), pp. 918-923, 2003.

[13] R. Cucchiara, M. Piccardi, A. Prati, "Detecting moving objects, ghosts and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 25(10), pp. 1337-1342, 2003.

[14] B. F. Wu and J. H. Juang, "Adaptive vehicle detector approach for complex environments," *IEEE Transactions on Intelligent Transportation Systems*, 13(2): 817-827, 2012.

[15] T. N. Tan, G. D. Sullivan and K. D. Baker, "Model-based localization and recognition of road vehicles," *Int J of Comp Vision*, 27(1): 5-25, 1998.

[16] L. Unzueta, M. Nieto, A. Cortes, J. Barandiaran, O. Otaegui and P. Sanchez, "Adaptive multicue background subtraction for robust vehicle counting and classification," *IEEE Transactions on Intelligent Transportation Systems*, 13(2): 527-540, 2012.

[17] N. Buch, J. Orwell and S. A. Velastin, "Urban road user detection and classification using 3D wire frame models," *IET Computer Vision*, pp. 105-116, 2010.

[18] N. Buch, J. Orwell, S. A. Velastin, "3D Extended Histogram of Oriented Gradients (3DHOG) for Classification of Vehicles and Pedestrians in Urban Scenes", *British Machine Vision Conference, London,* 2009

[19] O. Hasegawa and T. Kanade, "Type classification, color estimation, and specific target detection of moving targets on public streets," *Machine Vision Applications*, 16: 116-121, 2005.

[20] X. Ma and W. E. L. Grimson, "Edge-based rich representation for vehicle classification," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, vol. 2, pp. 1185–1192, 2005.

[21] G. Zhang, R. P. Avery and Y. Wang, "A video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras," *Transportation Research Record: Journal of the Transportation Research Board*, 1993: 138-147, 2007.

[22] B. Morris and M. M. Trivedi, "Learning, modelling and classification of vehicle track patterns from live video," *IEEE Transactions on Intelligent Transport Systems*, 9(3): 425-437, 2008.

[23] Z. Zhang, M. Li, K. Huang and T. Tan, "Boosting local feature descriptors for automatic objects classification in traffic scene surveillance," *19th International Conference on Pattern Recognition*, 1-4, 2008.

[24] Z. Chen, N. E. Pears, M. Freeman and J. Austin, "Road vehicle classification using support vector machines," in *Proceedings of IEEE International Conference on Intelligent Computing and Intelligent System*, Shanghai, China, 214-218, 2009.

[25] N. C. Mithun, N. U. Rashid and S. M. Rahman, "Detection and classification of vehicles from video using multiple time-spatial images", *IEEE Transactions on Intelligent Transportation Systems*, 13(3): 1215-1225, 2012.

[26] Z. Chen, T. Ellis, "Self-adaptive Gaussian mixture model for urban traffic monitoring system," *The 11th IEEE International Workshop on Visual Surveillance (ICCV'2011 Workshop)*, Barcelona, Spain, pp. 1769-1776, 2011.

[27] Z. Chen, T. Ellis, "A self-adaptive Gaussian mixture model", Computer Vision and Image Understanding, 122, pp. 35-46, 2014.

[28] P. J. Withagen, K. Schutte and F. C. A. Groen, "Global intensity correction in dynamic scenes," *International Journal of Computer Vision*, 86: 33-47, 2010.

[29] T. Horprasert, D. Harwood, L. S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," *Proceedings of IEEE ICCV'99 Frame rate workshop*, pp. 1-19, 1999.

[30] Z. Chen, T. Ellis and S. A. Velastin, "Vehicle detection, tracking and classification in urban traffic," *15th IEEE Annual Conference on Intelligent Transportation Systems,* pp. 951-956, 2012.

[31] Z. Chen, T. Ellis and S. A. Velastin, "Confidence based active learning for vehicle classification in urban traffic," the 4th Chile Workshop on Pattern Recognition (CWPR'2012), 12-16th Nov. 2012, Valparaiso, Chile.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in: Proceedings of Computer Vision and Pattern Recognition, Vol. 1, 886-893, 2005.

[33] A. Bosch, A. Zisserman, X. Munoz, "Representing shape with a spatial pyramid kernel," in: *Proceedings of the ACM International Conference on Image and Video Retrieval*, 2007.

[34] A. Bosch, A. Zisserman, X. Munoz, "Image classification using random forests and ferns," in *Proceedings of the IEEE 11th International Conference on Computer Vision*, 1-8, 2007.

[35] X. Han, Y. Chen, X. Ruan, "Image recognition by learned linear subspace of combined bag-of-features and low-level features," in: *Proceedings of IEEE International Conference on Image Processing*, 1049-1052, 2010.

[36] Z. Chen, T. Ellis, "Multi-shape descriptor vehicle classification for urban traffic," *International Conference on Digital Image Computing: Techniques and Applications*, pp. 456-461, 2011.

[37] Z. Chen, T. Ellis and S. A. Velastin, "Vehicle type categorization: A comparison of classification schemes," *IEEE 14th International Conference on Intelligent Transportation Systems*, pp. 74-79, Oct. 2011.

[38] Z. Chen, T. Ellis, "Efficient annotation of video for vehicle type classification", 16th International IEEE Conference on Intelligent Transportation Systems-(ITSC), pp. 59-64, 2013

[39] N. Buch, S. A. Velastin and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Transaction on Intelligent Transportation Systems*, 12(3), pp. 920-939, 2011.

[40] iLIDS, Home office scientific development branch. Imagery Library for Intelligent Detection Systems (iLIDS), accessed 4 September 2009 (http://www.ilids.co.uk/).

[41] D. Doermann and D. Mihalcik, "Tools and techniques for video performances evaluation," *ICPR*, pages 167-170, 2000 (http://viper-toolkit.sourceforge.net/

[42] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *SIGGRAPH 2001 course 8. In Computer Graphics, Annual Conference on Computer Graphics & Interactive Techniques*. ACM Press, Addison-Wesley,

Los Angeles, CA, USA (August 12–17), SIGGRAPH course pack edition, 2001.

[43] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), pp. 603–619, 2002.

[44] Z. Chen, A. M Wallace, “Active segmentation and adaptive tracking using level sets,” *British Machine Vision Conference 2007*, University of Warwick, pp. 920-929, September 2007.

[45] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 32(9), 1627-1645, 2010.

[46] R. Rios-Cabera, T. Tuytelaars and L. V. Gool, “Efficient multi-camera vehicle detection, tracking, and identification in a tunnel surveillance application,” *Computer Vision and Image Understanding*, 116(6), 742-753, 2012.

[47] Niluthpol Chowdhury Mithun, Nafi Ur Rashid, and S. M. Mahbubur Rahman, “Detection and Classification of Vehicles From Video Using Multiple Time-Spatial Images”, IEEE Trans. ITS, **13**(3), 1215-1225, 2012

[48] F. Yin, D. Makris, and S.A. Velastin. “Time efficient ghost removal for motion detection in visual surveillance systems.” *Electronics Letters* 44.23 (2008): 1351-1353.

Zezhi Chen received the M.Sc degree in mathematics from Northwest University, Xi’an, China, in 1994. He was awarded the first Ph.D degree in communication and information engineering from Xidian University, Xi’an, China, in 2002 and the second Ph.D degree in image processing from Kingston University, London, UK, in 2012. His research interests include computer vision, image processing and machine learning. He is currently a visiting researcher in the School of Computer Science and Mathematics, Faculty of Science, Engineering and Computing, Kingston University, London, U.K., where he has worked on traffic surveillance systems using image analysis. He is the author or a co-author of more than 60 publications in international journals and conferences.

Tim Ellis received his first degree in physics from the University of Kent, Canterbury, Kent, U.K., in 1974 and the Ph.D. degree in biophysics from London University, London, U.K., in 1981. He joined City University in 1979. In 2003, he moved to Kingston University, London, where he was previously Dean of the Faculty of Computing, Information Systems and Mathematics. His research interests include visual surveillance, industrial inspection, color image analysis, and vision systems hardware. He co-chaired the 2004 British Machine Vision conference and a previous chairman of the British Machine Vision Association.

Sergio A. Velastin received the B.Sc. degree in electronics, the M.Sc. (Research) degree in digital image processing, and the Ph.D. degree from the University of Manchester, Manchester, U.K., in 1978, 1979, and 1982, respectively. He became Professor of Applied Computer Vision at Kingston University where he also led its Digital Imaging Research Centre. He is currently UC3M-Conex Marie Curie Research Professor at the University Carlos III in Madrid. His current research interests include computer vision for pedestrian and traffic monitoring and human action recognition. He is a Fellow of the IET and a Senior Member of the IEEE.

Captions of the figures

Fig.1. Flow chart of the AutoVDC system.

Fig. 2. Detection environment.

Fig. 3. Vehicle detection: (a) modeled GMM background image and detection zone gates (b) current input image. (c) background subtraction results: black pixels represent the modeled background, foreground object (yellow), shadow (green) or reflection highlights (red); (d) foreground image created by extracting the pixels from the original frame using the final foreground object mask, morphologically dilated.

Fig. 4. An input image and the shape spatial pyramid representation of IPHOG for a bus and a car over three spatial scales: 9 features (level 0), 36 features (level 1), 144 features (level 2).

Fig. 5. a) Plan view image; b) calibration reference image (middle) and c) zoomed portion of b). Cyan circles and index numbers indicate the corresponding points and the blue asterisks indicate the re-projected points.

Fig.6. 3D wireframe models [17] with true size (a) and samples of their projections on the ground plane (b).

Fig. 7. (a) Example image, (b) the variation of average value of red, green, blue channel sampled from region indicated by the blue box over 2000 frames.

Fig. 8. (a) The variation of MCC corresponding to different thresholds of SMD. (b) ROC curve.

Figure 9. Synthetic vehicle data set for training SVM classifier, yellow curve shows the convexhull silhouette. (a) motorcycle, (b) car, (c) van, (d) bus.

Fig. 10. A snapshot GUI of AutoVDC showing occurrence of vehicle types associated with each traffic lane. Note that Lane 3 is primarily allocated for bus and taxi usage, though bicycles will typically be located next to the kerb.

Fig. 11. Snapshot GUI from two other camera locations.

Fig. 12. Examples of failures: a-b) in detection, c-d) in classification and e-f) complete misses due to heavy occlusion. The bold curve is silhouette of the detected blob in different categories (bus: green, car: red and van: blue). The yellow rectangle is the corresponding bounding box.