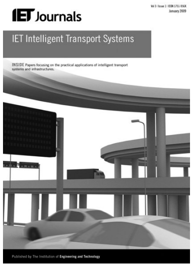
Published in IET Intelligent Transport Systems
Received on 8th February 2012
Revised on 27th October 2012
Accepted on 21st November 2012
doi: 10.1049/iet-its.2012.0104

www.ietdl.org



ISSN 1751-956X

A GMM-SVM Approach to Vehicle Type and Color Classification

Zezhi Chen[a,*], Nick Pears[b], Michael Freeman[b], Jim Austin[b],

[a] *Cybula Ltd, Computer Science Building, Deramore Lane, York, YO10 5DD, UK*
[b] *Department of Computer Science, University of York, York, YO10 5DD, UK*

Abstract

We describe our approach to segmenting moving road vehicles from the color video data supplied by a stationary roadside CCTV camera and classifying those vehicles in terms of type (car, van and HGV - Heavy Goods Vehicle) and dominant color. For the segmentation, we use a recursively updated Gaussian mixture model approach, with a multi-dimensional smoothing transform. We show that this transform improves the segmentation performance, particularly in adverse imaging conditions, such as when there is camera vibration. We then present a comprehensive comparative evaluation of shadow detection approaches, which is an essential component of background subtraction in outdoor scenes. For vehicle classification, a practical and systematic approach using a kernelized support vector machine is developed. The good recognition rates achieved in our experiments indicate that our approach is well suited for pragmatic vehicle classification applications.

Keywords:
Vehicle recognition, Segmentation, Classification, GMM background

[*] Corresponding author
Email address: z.chen@kingston.ac.uk (Zezhi Chen)

Preprint submitted to IET Intelligent Transport Systems *October 27, 2012* subtraction

1. Introduction

Traffic monitoring is an important tool in the development of intelligent transport systems (ITS) involving the detection and categorization of road vehicles. We consider the case of a nominally static camera observing a road scene, such as is the case in many visual surveillance applications and we aim to categorize vehicles into their type (car, van, HGV - Heavy Goods Vehicle) and color. Our system is intended to be used for intelligent surveillance systems for crime detection, security and road charging schemes. For example, it is a common offence to swap a licence plate from a small vehicle (car) to a large vehicle (van) in order to reduce road charges, which are derived using automatic number plate recognition (ANPR). A system such as ours, used in conjunction with ANPR and database connectivity, would be able to determine whether the licence plate belongs the particular vehicle type and color. Given that the vehicle type provides detailed information on the traffic composition, it is also likely to be widely useful from a transportation operation perspective

To achieve our goal, we generate a background/foreground image segmentation. In real applications, cameras are often mounted on metal poles, which can oscillate in the wind, thus making the segmentation problem more difficult. To deal with this kind of problem, a spatio-temporal filtering improvement to Zivkovic's recursively updated Gaussian mixture model (GMM) approach [37] is proposed.

Another main challenge in the application of background subtraction is identifying shadows that objects cast, which also move along with them in the scene. Shadows cause serious problems while segmenting and extracting moving objects due to the misclassification of shadow points as foreground. Thus, we present a comprehensive evaluation of shadow/highlight detection across different color spaces, and quantitative analysis results of our complete foreground/background segmentation system with shadow removal in several real-world scenarios. This is valuable to those developing pragmatic visual surveillance solutions that demand a high quality foreground segmentation.

Central to our system are a set of kernelized support vector machine (SVM) classifiers operating on measurement-based feature (MBF) vectors and color properties of the foreground blob corresponding to the segmented vehicle. MBF vectors encode the size, aspect ratio, width and solidity of this foreground blob. Note that *solidity* is a scalar specifying the proportion of the pixels in the convex hull of the foreground blob that is in the foreground blob itself.

For vehicle segmentation and classification, occlusions must also be properly dealt with to ensure accuracy. Occlusions occur when one vehicle appears next to another and blocks the line of sight either partially or completely. In addition, the image of a vehicle can be occluded by other objects, such as buildings and bridges along the road. Occlusions of this type should be regarded as the result of poor roadside camera placement and, in most cases, should be avoidable. Vehicle occluding vehicle, however, is more problematic and can result from very dense traffic flow. Occlusion reasoning becomes indispensable for vehicle detection under congested scenarios in urban traffic, when vehicle spacing becomes minimal and vehicle occlusions increase drastically. Once vehicles occlude each other, it becomes very difficult to segment them [35]. Many approaches turn to using the vehicle's appearance, thus relying on various models to determine an occluded vehicle's position. Segmentation of several vehicles from a single foreground moving object can be done only through some prior knowledge of the objects' appearance and/or behaviour. Note that, in this paper, we do not present a solution to vehicles occluding each other. Rather, our method is limited to small accidental occlusions.

In the following section, we discuss relevant prior work, we then present our GMM-SVM based approach to vehicle type and color classification. This is followed by an evaluation section and finally conclusions are presented.

2. Related literature

There is a lot of existing literature concerning the application of computer vision techniques to the analysis of urban traffic and a good review is provided by Buch et al. [4], which compares the latest research results. One recent approach by Nieto et al. [22] uses 3D models to detect and classify vehicles by integration of temporal information and model priors within a Markov Chain Monte Carlo (MCMC) method. Our approach is based on Gaussian Mixture Model (GMM) based background modelling for segmentation and Support Vector Machines (SVM) for classification and we present previous literature for each of these approaches in the following two subsections.

2.1. Background modelling and segmentation

To segment moving objects, a background model is built from the static camera image data and objects are segmented if they appear significantly different from this modeled background. A GMM was proposed by Friedman and Russell [12] and it was refined for real-time tracking by Stauffer and Grimson [27]. The algorithm relies on the assumptions that the background is visible more frequently than any foreground regions and that it has models with relatively narrow variances. The system can deal with real-time outdoor scenes with lighting changes, repetitive motions from clutter, and long-term scene changes. Many adaptive GMM model have been proposed to improve the background subtraction method since that original work. Power and Schoonees [24] presented a GMM model employed with a hysteresis threshold. They introduced a faster and more logical application of the fundamental approximation than that used by Stauffer and Grimson [27]. The standard GMM update equations have been extended to improve the speed and adaptation of the model [17]. Martel-Brisson and Zaccarin [20] extend the GMM to deal with shadows. Many researchers have adapted this model for traffic analysis [16, 33, 34]. All these GMMs use a fixed number of components. Zivkovic and Heijden [37] presented an improved GMM model that adaptively chooses the number of Gaussian mixture components for each pixel on-line, according to a Bayesian perspective. We call this method the Zivkovic-Heijden Gaussian mixture model (ZHGMM) in the remainder of this paper and it forms the basis of our improved method.

We also deal with shadows in our work. Prati et al. [25] present a comprehensive survey of moving shadow detection approaches. It is important to recognize the type of features utilized for shadow detection. Some approaches improve performance by using spatial information working at a region level or at a frame level instead of pixel level [10]. Finlayson et al. [11] proposed a method to remove shadows from a still image using illumination invariance. Cucchiara et al. [9] proposed the detection of moving objects, ghosts and shadows in HSV color space and gave a comparison of different background subtraction methods.

2.2. Support vector machines for vehicle type classification

For vehicle classification, prior related work includes that of Baek et al. [2], who presented a vehicle color classification based on the SVM. The implementation results showed 94.92% of success rate for 500 outdoor vehicles with 5 colors. Ambardekar et al. [1] used optical flow and knowledge of camera parameters to detect the pose of a vehicle in the 3D world. This information is used in a model-based vehicle detection and classification technique employed by their traffic surveillance application. Ma and Grimson [19] proposed an approach to vehicle classification under a mid-field surveillance framework. They discriminate features based on edge points and modified SIFT descriptors [18]. Eigenvehicle and PCA-SVM were proposed and implemented to classify vehicle into trucks, passenger cars, van and pick-ups [36].

In our paper, both vehicle type and vehicle color classification are based on kernelized SVMs. The SVM is a nonlinear generalization of the generalized portrait algorithm developed in Russia in the sixties [32][31]. It is firmly grounded in the framework of statistical learning theory, which has been developed over the last three decades by Vapnik [28][29][23]. Intuitively, given a set of points which belong to one of two classes, an SVM finds the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. According to [29][30], this hyperplane minimizes the risk of misclassifying examples of some unseen test set.

The solution of binary classification problems using SVMs is well developed. Multi-class problems (such as object recognition and image classification [6]) have typically been solved by combining independently produced binary classifiers. In the one-vs-all (OVA, or one-vs-rest) method, one constructs k classifiers, one for each class. The mth classifier constructs a hyperplane between class m and the k-1 other classes. If, for example, the classes of interest in an image include car, van and HGV, classification would be effected by classifying car against non-car (i.e. HGV and van) or HGV against non-HGV (i.e. car and van). This method has been used widely in the support vector literature to solve multi-class pattern recognition problems [3][26][21]. Alternatively, one-vs-one (OVO, or all-vs-all) approach involves constructing an SVM for each pair of classes resulting in $k(k-1)/2$ classifiers. For each distinct pair m_1 and m_2 , we run the learning algorithm on a binary problem in which examples labeled $y = m_1$ are considered positive, and those labeled $y = m_2$ are negative. All other examples are simply ignored. When applied to a test point, each classification gives one vote to the winning class and the point is labeled with the class having most votes. This approach can be further modified to give weighting to the voting process.

3. GMM-SVM-based Vehicle Type and Color Classification

In this section, we present our approach to road vehicle type and color classification, which is based on GMMs for segmentation and SVMs for classification. Firstly, we give an overview of our system, to show how all the different parts fit together. After this, we present our spatio-temporally smoothed segmentation process, then we discuss how we can remove shadows in several different color spaces and how we can improve the accuracy of the GMM from a video which is acquired by a shaking camera (these are evaluated later in the paper). Finally we discuss how to represent color for the purposes of color classification.

3.1. System overview

Our system is constructed from three modules: background learning, foreground extraction and vehicle classification. Figure 1 illustrates the flow chart of this system, where MDGKT in the background learning module is our multi-dimensional Gaussian kernel density transform employed for spatio-temporal smoothing. Other stages in this flow chart are either explained later in this paper or are self-evident. Note that *dilation* and *erosion* are standard morphological operations applied to a binary image, where contiguous areas of either ones or zeros are grown (dilation) or shrunk (erosion) by the radius of some so-called structuring element [14]. Morphological opening is an often-used methodology in image processing. Opening of a binary image is erosion followed by a dilation using the same structuring element for both operations.

3.2. Applying spatio-temporal smoothing to ZHGMM-based segmentation

In order to improve the stability and robustness of the ZHGMM background learning algorithm, we have used a *Multi-Dimensional Gaussian Kernel density Transform* (MDGKT) as a pre-process. Thus the basis of our background modeling process is to employ the ZHGMM algorithm, augmented with the MDGKT. Typically, an image is represented as a two-

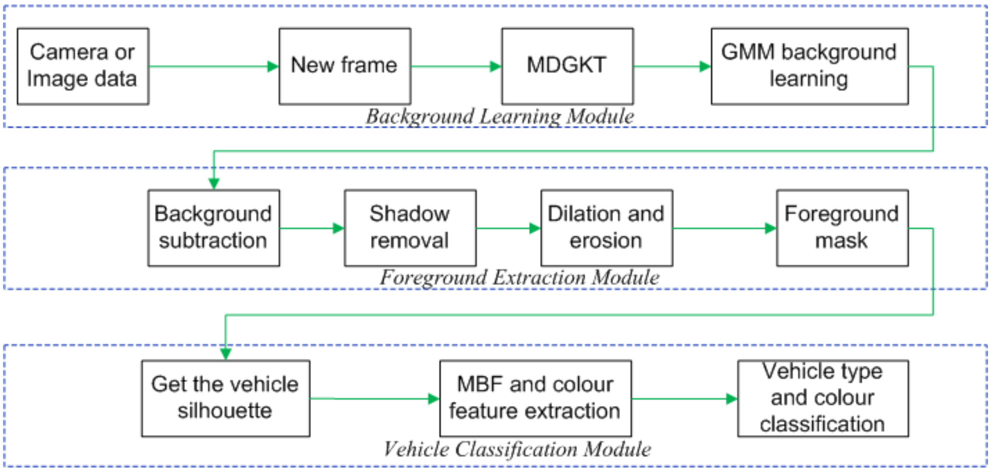

Camera or Image data
New frame
MDGKT
GMM background learning
Background Learning Module
Background subtraction
Shadow removal
Dilation and erosion
Foreground mask
Foreground Extraction Module
Get the vehicle silhouette
MBF and colour feature extraction
Vehicle type and colour classification
Vehicle Classification Module

Figure 1: Flow chart of the overall system. MDGKT is the module employed for spatio-temporal smoothing. Dilation and erosion are standard morphological operations applied to a binary image.

dimensional matrix of p-dimensional vectors, where p=1 in the gray-level case, p=3 for color images, and $p > 3$ for multispectral images. The space of the matrix is known as the spatial domain, while the gray, color or multispectral is known as the *spectral* domain [7, 8]. For algorithms that use image sequences, there is also the temporal domain.

We define $k(x)$ to be a kernel profile (we use Gaussian), then over spatial and temporal domains this is given as:

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

where x^s is the spatial part and x^t is the temporal part of the feature vector, h_s and h_t are the kernel bandwidths, and D is the corresponding normalization constant.

3.3. Shadow removal

The previous section showed promising initial qualitative results for the ZHGMM augmented with MDGKT background subtraction algorithm. However, the algorithm is susceptible to both global and local illumination changes such as shadows and highlight reflections (specularities). These often cause subsequent processes, such as tracking and recognition, to fail. We give a comparison of several different shadow removal methods, working in different color spaces below. For clarity, we distinguish two different foreground segmentations: segmentation **F1**, is the foreground segmentation which includes shadows (raw background subtraction output), while **F2** is the foreground segmentation after we have removed shadows.

3.3.1. Shadow removal in RGB space

RGB color. The observed color vector is projected onto the expected color vector obtained from the background model, and the ith pixel's brightness distortion is a scalar value (less than unity for a shadow) describing the fraction of remaining 'brightness'. This may be obtained by minimizing [13],

$$\Phi\left(\alpha_i\right) = \left(I_i - \alpha_i E_i\right)^2 \tag{2}$$

where $I_i = [I_{Ri}, I_{Gi}, I_{Bi}]$ denotes the ith pixel value in RGB space, $E_i = [\mu_{Ri}, \mu_{Gi}, \mu_{Bi}]$ represents the ith pixel's expected (mean) RGB value in the modeled background. The solution to equation (2) is an alpha value equal to the inner product of I_i and E_i, divided by the square of the Euclidean norm of E_i. Color distortion is defined as the orthogonal distance between the observed color and the expected color vector. Thus, the chromaticity distortion of the ith pixel is $CD_i = \|I_i - \alpha_i E_i\|$. If we balance the color bands by rescaling the color values by the pixel's *std* $s_i = [\sigma_{Ri}, \sigma_{Gi}, \sigma_{Bi}]$, the brightness and chromaticity distortion are

$$\alpha_i = \frac{\sum_{\kappa \in R,G,B} I_{\kappa i} \mu_{\kappa i} / \sigma_{\kappa i}^2}{\sum_{\kappa \in R,G,B} \left[\mu_{\kappa i} / \sigma_{\kappa i}\right]^2} \tag{3}$$

$$CD_i = \sqrt{\sum_{\kappa \in R,G,B} \left(I_{\kappa i} - \alpha_i \mu_{\kappa i}\right)^2 / \sigma_{\kappa i}^2} \tag{4}$$

Then the pixel in the foreground segmentation (**F1**) may be classified as either a shadow or highlight reflection on the true background as follows:

$$\begin{cases} Shadow \quad CD_i < \beta_1 \;\; and \;\; \beta_3 < \alpha_i < 1 \\ Highlight \quad CD_i < \beta_1 \;\; and \;\; \alpha_i > \beta_2 \end{cases} \tag{5}$$

β_1 is a selected threshold value, used to determine the similarities of the chromaticity between the modeled background 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. We introduce a threshold β_3 to avoid this problem. This is defined as: $\beta_3 = 1/(1-\epsilon)$, where ϵ is a lower band for the normalized brightness distortion. We also introduce a threshold β_2 on normalised brightness distortion, in order to detect highlights. An automatic threshold selection method was provided by Horprasert et al. [13].

Using intensity information only. Let the brightness of a pixel value of the modeled background be $s_{bi} = R_{bi} + G_{bi} + B_{bi}$, and assume that this pixel is covered by a shadow in frame t and let s_{ti} be the observed brightness value for this pixel at this frame. Then, the pixel in the foreground segmentation (**F1**) may be classified as either a shadow or highlight reflection on the true background, as follows:

$$\begin{cases} Shadow & \beta_1 < s_{ti}/s_{bi} \leq \beta_2 \\ Highlight & \beta_3 < s_{ti}/s_{bi} \end{cases} \tag{6}$$

where β_1, β_2 and β_3 are selected threshold values used to determine the similarities of the normalized brightness between the background image and the current observed image.

3.3.2. Shadow removal in HSV color space

HSV color space explicitly separates chromaticity and luminosity and has proven easier than RGB space to set a mathematical formulation for shadow detection [9, 25]. For each pixel in **F1**, that initially has been segmented as foreground, we check if it is a shadow on the background according to the following consideration.

$$\begin{cases} Shadow \quad \beta_1 < V_{Ii}/V_{Bi} < \beta_2 \;\; and \;\; |H_{Ii} - H_{Bi}| < \tau_H \\ \qquad\qquad and \;\; |S_{Ii} - S_{Bi}| < \tau_S \\ Highlight \;\; V_{Ii}/V_{Bi} > \beta_3 \;\; and \;\; |H_{Ii} - H_{Bi}| < \tau_H \\ \qquad\qquad and \;\; |S_{Ii} - S_{Bi}| < \tau_S \end{cases} \tag{7}$$

with $0 < \beta_1, \beta_2, \tau_H, \tau_S < 1$ and $\beta_3 > 1$, where H_{Ii}, S_{Ii}, V_{Ii} and H_{Bi}, S_{Bi}, V_{Bi} are H, S, V channels in the current image and modeled background respectively.

3.3.3. Shadow removal in YCbCr and Lab color spaces

We now consider the luminance and chrominance color space (YCbCr) to remove shadows from the results of background subtraction. If a shadow is cast on a background, the shadow darkens a point in the background. The luminance distortion is $\zeta_i = Y_i^I / Y_i^B$, and chrominance channels difference is

$$CH_i = \left(\left| C_{bi}^I - C_{bi}^B \right| + \left| C_{ri}^I - C_{ri}^B \right| \right) /2,$$

where Y_i^I, C_{bi}^I, C_{ri}^I and Y_i^B, C_{bi}^B, C_{ri}^B are Y, Cb, Cr channels in the current image and modeled background respectively. A pixel in the **F1** is classified as follows:

$$\begin{cases} Shadow & \zeta_i < 1 \ \ and \ \ CH_i < \beta_1 \\ Highlight & \zeta_i > \beta_2 \ \ and \ \ CH_i < \beta_1 \end{cases} \tag{8}$$

where $\beta_1 < 1$ and $\beta_2 > 1$. There is a similar criterion for shadow removal in Lab space.

3.4. Vehicle color representation

The use of color histograms leads to very simple and low-level methods for color classification. Since we are dealing with discretely sampled data from color images, we use discrete densities stored as m-bin histograms. The basic idea is that we divide RGB space up into a set of equal-sized cubic regions and map these regions into a 1D histogram that can be populated by the colors of the pixels in the segmented foreground region (F2).

If all possible colors in three channels in a 24-bit image are quantised, there are 256^3 bins. Such a histogram would be sparsely populated. Very fine quantization of the color space is probably unjustified for images in which the illumination may be variable, and there is additional noise on the color video. 16 bins histogram for each color is reasonable, but it still needs a large amount of memory and computation cost, particularly in terms of the embedded hardware (i.e. in camera) implementation that we aimed to develop. Therefore, we use a much coarser quantization of the color space, namely 8 bins along each color axis, giving color histograms with $512 = 8^3$ bins. If we increased the number of bins, the computation time would dramatically increase.

4. Evaluation

In this section, we present a quantitive evaluation of our system. Firstly, we evaluate the segmentation system, which includes background subtraction and shadow removal in the various color spaces previously described. We present experiments that demonstrate how our system alleviates the negative impacts of camera vibration. Finally, we detail the evaluations of vehicle type classification and vehicle color classification.

4.1. Online training using MDGKT

A sample RGB frame is shown in Fig.2(a). The black 'X' in the centre of the red rectangle shows the position of a sample pixel stream over a video (596 frames) in an area where no intruding object appears over time. A Gaussian kernel was chosen as the kernel profile. For this particular pixel, the standard deviation (*std*) of the blue and red channels in the original video in the temporal domain is 1.834 and 1.110 respectively, but the *std* values of the same channels using MDGKT smoothing are only 1.193 and 0.832. Figures 2(b) and (c) show the scatter plots of the original and MDGKT image (red, blue) values of the same pixel in the temporal domain. Fig.2(c) shows that the distribution of MDGKT image is more localized within two Gaussian components of the mixture model, illustrating the effect of the spatio-temporal filtering in the spectral domain. A Gaussian mixture of two components is required to model the blue channel intensity distribution. The ground truth GMM (blue trace) and the estimated GMM distribution (red trace) using the MDGKT of the blue channel of the sample pixel are shown in Fig.2(d).

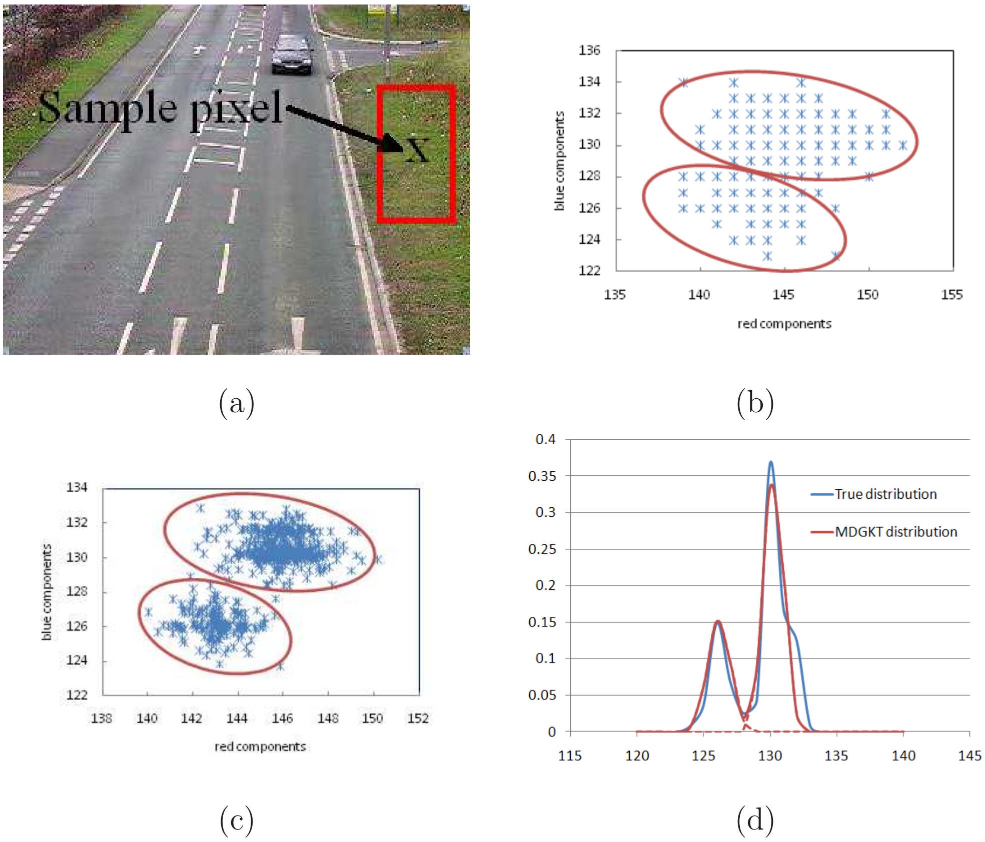


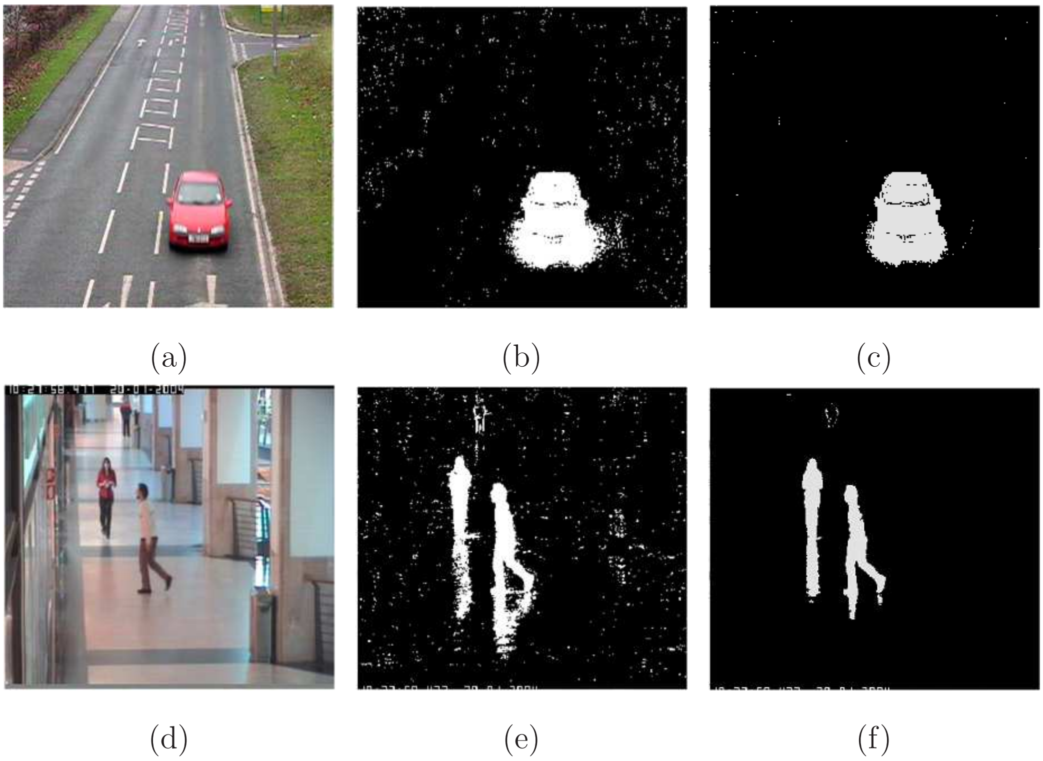
Figure 2: The effect of spatio-temporal filtering. (a) A sample image showing a sample pixel. (b) and (c) the scatter plots of the sample pixel's color (blue and red channels) distribution in the original images and MDGKT images respectively. (d) The modeled GMM component (red trace) and the actual distribution (blue trace) of the blue channel values of the sample pixel in the temporal domain.

The MDGKT algorithm described above allows us to identify the foreground pixels in each new frame while updating the description of each pixel's background model. This procedure is effective in determining the boundary of moving objects, thus moving regions can be characterized not only by their position, but also size, aspect ratio, moments and other shape and color information. These characteristics can be used for later processing and classification, for example, using a support vector machine [15].

To give an initial qualitiative understanding of the performance of the algorithm, we used a dynamic scene to do some preliminary segmentation evaluation. The results are shown in Fig. 3, where (a) and (d) are original images: one is an outside scene, while the other is an indoor scene. Figures 3 (b) and (e) are the results of the basic ZHGMM algorithm, while (c) and (f) are the results of ZHGMM augmented with our MDGKT algorithm. Note that the results shown are pixel-based results without the application of any post-processing. For online training using MDGKT, the speed of this method is real-time (25 frames per second) for a C++ implementation on a standard PC. Also, note that, for a background model learning alpha value of 0.001, any sudden increase in background lighting, or any moving object that becomes stationary (eg a parking car), will have fully merged into the background model after 105 frames (just over 4 seconds at 25fps).

4.2. Evaluation of segmentation with shadow removal

This section demonstrates the performance of the proposed algorithms above on several videos of both indoor and outdoor scenes, using an image size of 320×240. A quantitative comparison of two GMMs (ZHGMM and



(a) (b) (c)

(d) (e) (f)

Figure 3: Comparative results of ZHGMM (b and e) segmentation and ZHGMM augmented with MDGKT (c and f)

ZHGMM augmented with MDGKT) with different shadow removal methods is presented. A set of videos to test the algorithms was chosen and, in order to compute the evaluation metrics, the ground truth for each frame is necessary. We obtained this ground truth by segmenting the images with a manual classification of points as foreground, background and shadow regions. We prepared 41 ground truth frames in a '*walking people*' sequence, and 26 in a '*moving car*' sequence. We did not annotate the frames which have been used for GMM learning. All shadow removal methods in five color spaces using the two GMM methods have been fully implemented. Three metrics for the segmentation evaluation: true positive rate (TPR), true negative rate (TNR) and false positive rate (FPR) are defined as follows:

$$TPR = \frac{TP}{TP + FN} \tag{9}$$

$$TNR = \frac{TN}{FP + TN} \tag{10}$$

$$FPR = \frac{FP}{FP + TN} \tag{11}$$

where TP is true positive pixels, FP is false positive pixels, TN is true negative pixels and FN is false negative pixels. Quantitative results of TPR and TNR are reported in Table 1, which illustrate that the results in RGB color space provide the best segmentation in terms of the combined values of TPR and TNR (the larger values are better).

One may ask why RGB performs better than other color spaces for shadow removal, for the set of parameters that we have used, particularly since other studies have suggested that HSV is a better color space for this. Firstly, for each system, we tried to get the best performance by adjusting each threshold parameter in sequence. This does not mean that a global optimum of performance will be found and, furthermore, it is likely to be easier to tune a system in this way, when the system has fewer parameters (our RGB system has 3 parameters, whilst our HSV system has 5). Secondly, when bounding the variables of a color space with thresholds, we are bounding them with planes that, when mapped into other color spaces become curved surfaces. It is likely that some color spaces will work better than others simply because we can linearly bound the training examples better in one space than another. Our sequential tuning approach worked best in RGB

Table 1: Experimental quantitative results

	ZHGMM		MDGKT	
	TPR	TNR	TPR	TNR
RGB	**0.8548**	**0.9838**	**0.9552**	**0.9853**
Lab	0.7165	0.9828	0.8499	0.9846
YCbCr	0.6183	0.9811	0.6748	0.9811
Intensity	0.6077	0.9628	0.6356	0.9714
HSV	0.5039	0.9671	0.6327	0.9712

space, although future work could give a stronger result by performing a grid search over the full parameter space for each method and by using a larger body of test data.

Fig.4 shows sample frames 9 and 17 of the '*walking people*' video, 3 and 8 of the '*moving car*' video. Each two-by-two block of images refers to the same frame in the original video. The top-left image is the original frame. The bottom-left image is the foreground segmentation (**F1**) results. In this image, all colored pixels are the foreground segmentation output of the ZHGMM augmented with MDGKT algorithm, while the black pixels represent the modeled background. The colored pixels are categorized as foreground object (colored yellow), shadow (colored green) or highlight (colored red) by our shadow removal algorithm operating in RGB color space. Note that the highlight reflections appear to be not very strong, with only a few pixels colored red. This is because the resolution of the image in the figure is very low and the red pixels are hard to see. The shadow and highlight reflection pixels are then removed and this is then followed by a post-processing binary morphology stage of dilatation and erosion to remove sparse noise. This gives the final foreground segmentation, as shown in the bottom right image of each two-by-two block. Finally, the top-right image in each block is a synthetic image, created by using the final foreground segmentation as a mask to extract the foreground object from the original frame, and superimposing this on the background model (mean value of each pixel). Clearly these synthetic images are largely shadow-free. The two videos in Fig.4 are scenes with very strong shadows.

The receiver operating characteristic (ROC curve) is used to analyse our system's performance. This plots TPR against FPR for a binary classifier as the threshold used to discriminate between the two classes is varied. For a GMM, any input distribution can be converted to ROC curves. One may combine multiple ROC plots for different values of some of the algorithm's fixed parameters. In this case, the learning rate α in the ZHGMM is the most significant parameter of interest. To produce a ROC curve, the learning rate α_{frame} is set up as an evenly distributed value ranging from 2 to 200 frames, in increments of 4 frames. The α_{frame} is converted to α value by the formula $\alpha = 1 - e^{ln0.9/\alpha_{frame}}$. For the '*moving car*' sequence, under the optimal threshold of Mahalanobis distance (threshold $= 7$), the ROC curve is shown in Fig.5. The lower rates of FPR towards the left of the graph are more interesting; for example, when FPR $= 0.0074$, the TPR of ZHGMM with MDGKT is 0.7377, but the TPR of ZHGMM only is 0.6639. The ROC curve show that ZHGMM with MDGKT performs much better than ZHGMM alone.

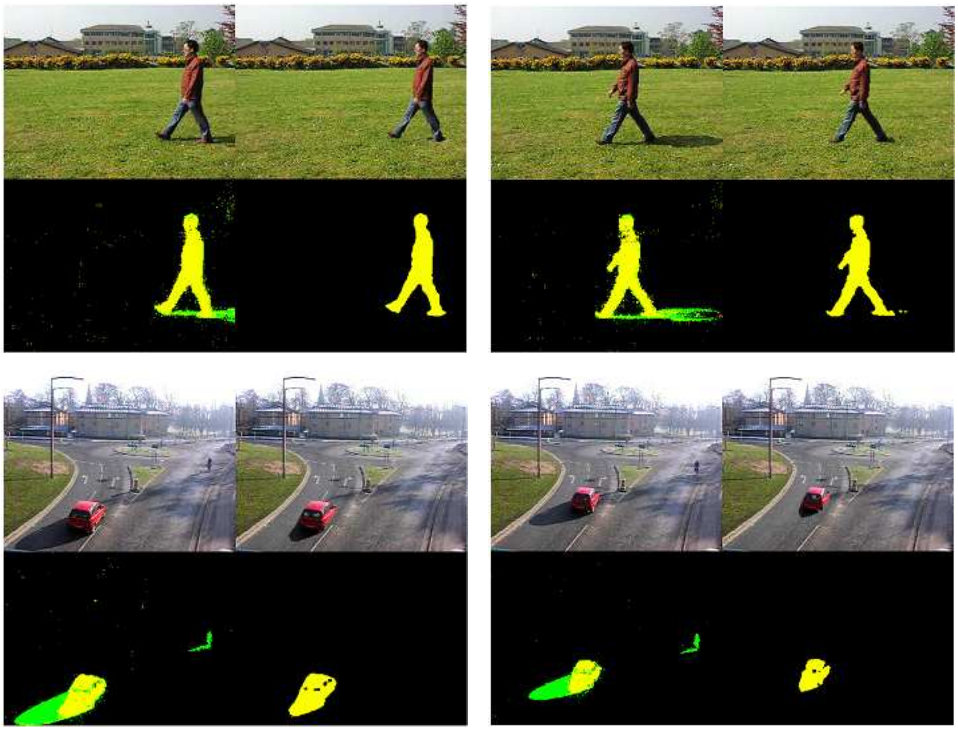


Figure 4: Foreground segmentation results in RGB colour space. Each two-by-two block of images refers to the same frame in the original video. The top-left image is the original frame. The bottom-left image is the foreground segmentation results. The black pixels represent the modeled background. The foreground object is colored yellow, with shadows colored green and highlights colored red. The bottom right image is the final foreground segmentation after shadow and highlight removal. The top-right image is a synthetic shadow free image composed of the foreground object overlayed onto the background model.

4.3. Evaluation of segmentation with camera vibration

In order to illustrate that MDGKT can improve the accuracy of GMM background subtraction in terms of camera vibration in RGB color space, both ZHGMM and ZHGMM augmented with MDGKT are compared using video acquired by a pole mounted road side CCTV camera under very

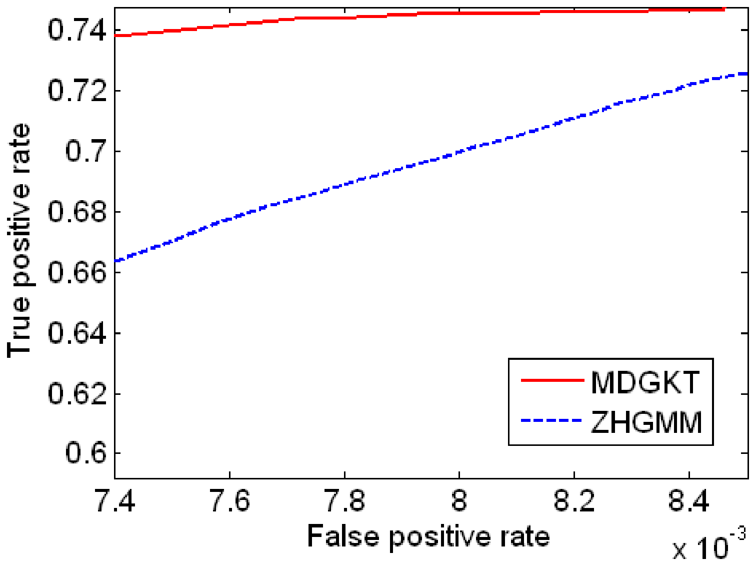
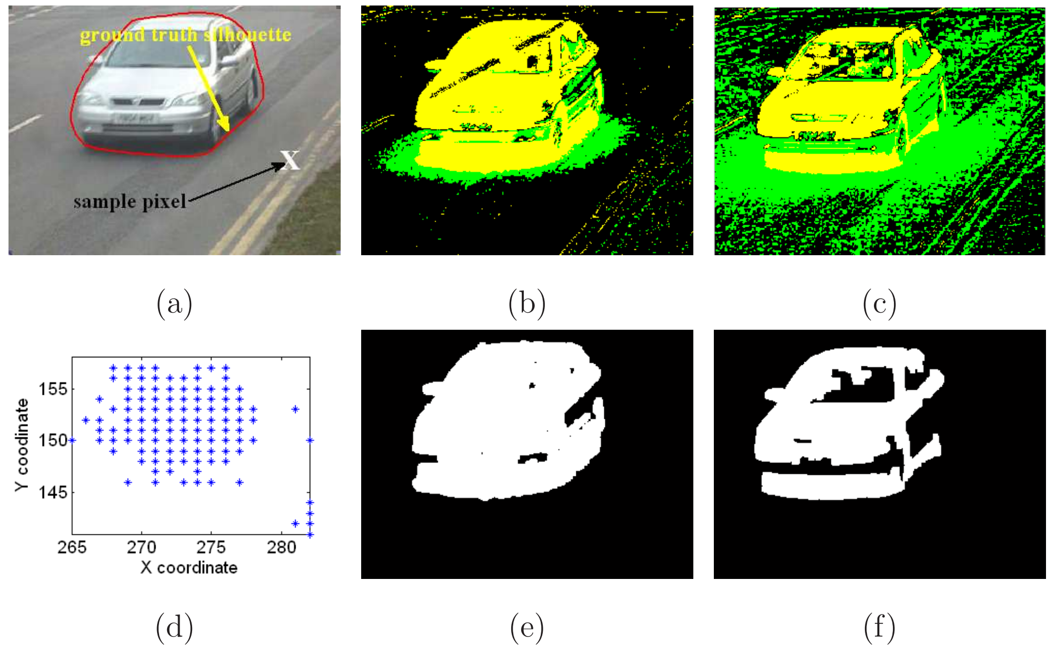


Figure 5: ROC curve of the '*moving car*' sequence. The ROC curve show that ZHGMM with MDGKT performs much better than ZHGMM alone, particularly at lower FPR values.

strong wind weather conditions. 781 frames are included in the video. 113 frames including foreground objects (vehicles) have been manually annotated to evaluate the algorithm. The image size is 320×240 pixels, 25 frames per second. A sample image with annotated ground truth (red silhouette) is given in Fig.6(a). The white 'X' shows the sample point. The ground truth location of this sample pixel stream over all frames has been annotated. The x and y coordinates of the sample point are given in Fig.6(d). The variance of x and y coordinates are 4.918 and 5.115 pixels, respectively. The maximum variance range along x direction is [-8, 9] pixels, along y direction is [-11, 6] pixels. Fig.6(b) and (e) illustrate the background subtraction results from ZHGMM using our MDGKT approach. Fig.6(c) and (f) show the background subtraction result from ZHGMM. (b) and (c) are the raw background subtraction result. Yellow pixels are foreground, green pixels are shadow and black pixels are background. Shadow removal is performed, followed by binary morphological opening (dilation and erosion) to remove noise. A diamond-shaped structuring element with 4 pixels as the distance from the structuring element origin to the points of the diamond is used. However, erosion and dilation alone cannot eliminate the larger noise elements after background subtraction and shadow removal. Small objects with an area of less than 200 pixels have been removed by connected component area calculation. The final foreground objects masks are given in Fig.6(e) and (f). The TPR of the ZHGMM with MDGKT algorithm and the ZHGMM (only) algorithm are 0.7840 and 0.5823, respectively. These experimental results illustrate that our MDGKT can improve the accuracy of GMM background segmentation, when the roadside camera is subject to wind-induced vibration.

4.4. Evaluation of SVM-based vehicle type classification

There is a lot of readily available software that uses SVMs to solve classification problems. In this research, the SVM-KM toolbox [5], which is a library of MATLAB routines, was used to handle multi-class classification problems. Our system classifies different vehicle types, in particular car, van and HGV. The foreground blob containing the vehicle is obtained using the improved background subtraction method based on the MDGKT. The size and width of the foreground blob is determined and normalized using the the known size of the licence plate. The size of the licence plate is easily determined, since it is retroreflective and is shown as a high intensity image region



(a) (b) (c)

(d) (e) (f)

Figure 6: Experimental results for a video is acquired by a vibrated road side CCTV camera. (a) A sample image with annotated ground truth (red silhouette) and the sample pixel (white 'X') used to annotate ground truth location. (d) Scatter plot x and y coordinates of sample pixel stream over the video. (b) and (e) background subtraction result from ZHGMM using MDGKT. (c) and (f) background subtraction result from ZHGMM. The foreground object (colored yellow), shadow (colored green) and background (colored black).

when illuminated using high intensity LEDs from within the ANPR camera. Totally 254 vehicles have been detected (car: 101, van: 92, HGV: 61). We extract a simple silhouette feature vector, whose components include size and width (normalized), aspect ratio, and solidity of the vehicle foreground blob. The observation vectors were normalized to a standard score ($z = (x - \mu)/\sigma$, μ and σ are the mean and standard deviation of the raw vector x; the mean and standard deviation variation of z is from 0 to 1). Note that, if some parts of the vehicle are the same color as the background, then there can be holes or ‘drop outs’ in the foreground segmentation. In practice, this did not appear to be a significant problem, and most of our measures, other than solidity (ie. width, size, aspect ratio) are fairly robust to this.

For our kernelized SVM, our data was partitioned into two halves: one half was used for training and the other half was used for our testing and validation strategy. A process of averaging the classification performance over 10 runs with different random selections of training and test sets was used to evaluate the classification method. Note that it is very difficult to visualize the shape of a separating boundary between two classes of vehicle in a high dimensional space. Therefore, we find the best SVM kernel and associated parameters simply by trying a selection of the most commonly used kernels and by doing a coarse grid search in the associated parameter space. If we assume $\gamma = 1$, there is one parameter, C, which needs to be determined for the Gaussian kernel, and two parameters, C and d, for the polynomial kernel. In order to obtain a good value for C (so that the classifier can accurately predict unknown data), a 2D *grid-search* on C and d values (for example $C = 2^{-5}, 2^{-3}, ..., 2^{15}$, d=1,2,3) was used. We found that he best results for our vehicle type classification were achieved using a Gaussian kernel with C=100.

The sensitivity, specificity and accuracy of classification results of both OVA and OVO are given in Table 2. Note that the accuracy figure is computed using the sum of the true positive and true negative classifications, divided by the total number of classifications. In addition, the associated confusion matrices are given Table 3. From these tables, we know that OVO

Table 2: The comparison of OVO and OVA for vehicle type classification

	OVO			OVA		
	Sensitivity	Specificity	accuracy	Sensitivity	Specificity	accuracy
Car	0.7797	0.9384	0.9016	0.8835	0.9338	0.9134
Van	0.6907	0.8408	0.7835	0.6222	0.7805	0.7244
HGV	0.7347	0.8780	0.8504	0.5574	0.8601	0.7874

Table 3: The confusion matrices for vehicle type classification

	OVO			OVA		
	car	van	HGV	car	van	HGV
Car	0.9010	0.0990	0	0.9109	0.0891	0
Van	0.0978	0.6087	0.2935	0.1304	0.7283	0.1413
HGV	0.0492	0.3934	0.5574	0.0656	0.3443	0.5902

and OVA do not have significant differences in performance, but OVO performance is slightly better than that of OVA on our data, especially for van classification.

4.5. Evaluation of vehicle color classification

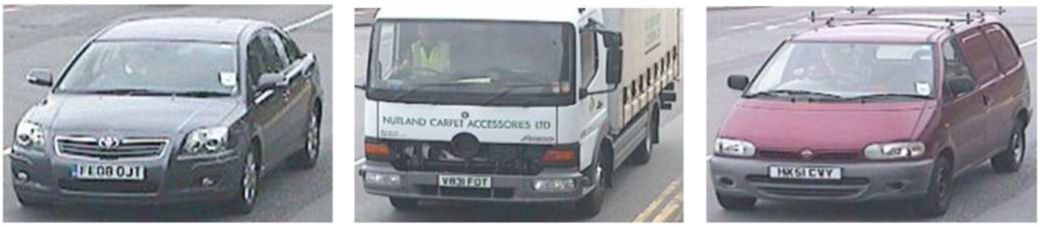


Figure 7: Three different vehicle types and colors (black car, white HGV and red van).

Table 4: Results of vehicle color classification

Color	Number	Sensitivity	Specificity	Accuracy
Black	101	0.8772	0.9963	0.9610
White	220	0.9952	0.9213	0.9610
Red	64	1.0	1.0	1.0

For color classification experiments, selected samples are randomly segregated into two sets, where half of the samples are used for training and half for testing. The observation vector is constructed by 8-bin 3D histogram in RGB color space. A normalized vector (sum of the vector elements is unity) of $8^3 = 512$ components is obtained. Figure 7 shows our own real data, where three different vehicle types and colors are illustrated (black car, white HGV and red van). Table 4 gives experimental results with the associated confusion matrix given in Table 5. The results were obtained using OVA with a polynomial SVM kernel, and parameter settings C=1000 and d=2. From the confusion matrix, we find that red vehicles are unambiguously distinguishable, the classifier has 100% recognition rate on our data set. Black and white classifications are slightly less successful, but the performance still leads to a useable system where suspect vehicles can be flagged for checking by a human operator.

5. Conclusions

Online learning of adaptive GMMs on nonstationary distributions is an important technique for moving object segmentation. This paper has presented an improvement to an existing adaptive Gaussian mixture model,

Table 5: Confusion matrix for vehicle color classification

	Black	White	Red
Black	0.9804	0.0196	0
White	0.0636	0.9364	0
Red	0	0	1.0

using a multi-dimensional spatio-temporal Gaussian kernel smoothing transform for background modelling in moving object segmentation applications. The model update process can robustly deal with slow light changes (from clear to cloud or vice versa), blurred images and camera vibration in very strong wind. The proposed solution has significantly enhanced segmentation results over a commonly used recursive GMM. We have given a comprehensive analysis of performance results in a wide range of environments and using a wide variety of color space representations. The system has been successfully used to segment objects in both indoor and outdoor scenes, with strong shadows, light shadows, and highlight reflections and we have verified our system with rigorous evaluation. We have found that working in standard RGB color space provides the best results.

We have presented a method to perform automatic vehicle classification. It is a two-step algorithm. The first step is type recognition. The type vector components are size, width, aspect ratio and solidity of the foreground (vehicle) blob. The second step is color recognition. In order to save memory space and computation complexity, the system uses an 8-bin 3D color histogram as the vector for SVM classification.

Acknowledgements

We thank the UK Technology Strategy Board (TSB) for funding this work under the CLASSAC project. The authors also would like to thank the anonymous reviewers for their general and detailed valuable comments and suggestions, which helped very much to improve the presentation of the paper.

References

[1] Ambardekar, A., Nicolescu, M., Bebis, G., 2008. Efficient vehicle tracking and classification for an automated traffic surveillance system. Signal and Image Processing.

[2] Baek, N., Park, S., Kim, K., Park, S., 2007. Vehicle color classification based on the support vector machine method. ICIC, CCIS 2, 1133–1139.

[3] Blanz, V., Scholkopf, B., Bulthoff, H., Burges, C., Vapnik, V., Vetter, T., 1996. Comparison of view-based object recognition algorithms using realistic 3d models. In Artificial Neural Networks - ICANN'96 1112, 251–256.

[4] Buch, N., Velastin, S., Orwell, J., 2011. A review of computer vision techniques for the analysis of urban traffic. IEEE Transactions on Intelligent Transportation Systems 99, 1–20.

[5] Canu, S., Grandvalet, Y., Guigue, V., Rakotomamonjy, A., 2005. Svm and kernel methods matlab toolbox. Perception Systems

et Information, INSA de Rouen, Rouen, France. http://asi.insa-rouen.fr/enseignants/ arakoto/toolbox/index.html.

[6] Chapelle, O., Haffner, P., Vapnik, V., 1999. Support vector machines for histogram-based image classification. IEEE Transaction on Neural newworks 10 (5), 1055–1064.

[7] Chen, Z., Husz, Z., Wallace, I., Wallace, A., 2007. Video object tracking based on a chamfer distance transform. In: IEEE International Conference on Image Processing, San Antonio, Texas, USA, 357–360.

[8] Comaniciu, D., Meer, P., 2002. Mean shift: a robust approach toward feature space analysis. IEEE Transaction on Pattern Analysis and Machine Intelligence 24 (5), 603–619.

[9] Cucchiara, R., Piccardi, M., Prati, A., 2003. Detecting moving objects, ghosts and shadows in video streams. IEEE Transaction on Pattern Analysis and Machine Intelligence 25 (10), 1337–1342.

[10] Elgammal, A., Harwood, A., Davis, L., 2000. Non-parametric model for background subtraction. LNCS, In: Proceedings of the 6th European Conference on Computer Vision-Part II 1843, 751–767.

[11] Finlayson, G., Hordley, S., Drew, M., 2002. Removing shadows from images. In: European Conference on Computer Vision, Lecture Notes in Computer Science 2353 (4), 823–836.

[12] Friedman, N., Russell, S., 1997. Image segmentation in video sequences: a probabilistic approach. In: Proc 13th Conf on Uncertainty in Artificial Intelligence, 175–181.

[13] Horprasert, T., Harwood, D., Davis, L., 1999. A statistical approach for real-time robust background subtraction and shadow detection. In: Proceedings of IEEE ICCV'99 Frame rate workshop.

[14] Jain, R., Kasturi, R., Schunck, B., 1995. Machine vision. McGraw-Hill series in Computer Science. McGraw-Hill, Inc., New York, NY.

[15] Joachims, T., 2006. Training linear svms in linear time. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06), Philadelphia, Pennsylvania, USA, 217–226.

[16] Johansson, B., Wiklund, J., Forssen, P. E., Granlund, G., 2009. Combining shadow detection and simulation for estimation of vehicle size and position. Pattern Recognition Letters 30 (8), 751–759.

[17] Lee, D.-S., 2005. Effective gaussian mixture learning for video background subtraction. IEEE Transaction on Pattern Analysis and Machine Intelligence 27 (5), 827–832.

[18] Lowe, D., 2004. Distinctive image features from scale invariant keypoints. International Jornal of Computer Vision 60 (2), 91–110.

[19] Ma, X., Grimson, W., 2005. Edge-based rich representation for vehicle classification. International Conference on Computer Vision, 1185–1192.

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

[21] Melgani, F., Bruzzone, L., 2004. Classification of hyperspectral remote sensing images with support vector machines. IEEE Transaction on Geoscience and Remote Sensing 42 (8), 1778–1790.

[22] Nieto, M., Unzueta, L., Cortes, A., Barandiaran, J., Otaegui, O., Sanchez, P., 2011. Real-time 3d modeling of vehicles in low-cost monocamera systems. In: Proc. Int. Conf. on Computer Vision Theory and Applications VISAPP2011, 459–464.

[23] ortes, C., Vapnik, V., 1995. Support vector network. Machine Learning 20, 1–25.

[24] Power, P., Schoonees, J., 2002. Understanding background mixture models for foreground segmentation. In: Proceedings of Image and Vision Computing, New Zealand.

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

[26] Scholkopf, B., Burges, C., Vapnik, V., 1995. Extracting support data for a given task. In the Proceedings of the First International Conference on Knowledge Discovering and data Mining, 252–257.

[27] Stauffer, C., Grimson, W., 2000. Learning patterns of activity using real-time tracking. IEEE Transaction on Pattern Analysis and Machine Intelligence 22 (8), 747–757.

[28] Vapnik, V., 1982. Estimation of Dependences Based on Empirical Data. Springer, Berlin.

[29] Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer, New York.

[30] Vapnik, V., 1999. An overview of statistical learning theory. IEEE Transaction on neural networks 10 (5), 988–999.

[31] Vapnik, V., Chervonenkis, A., 1964. A note on one class of perceptrons. Automation and Remote Control 25.

[32] Vapnik, V., Lerner, A., 1963. Pattern recognition using generalized portrait method. Automation and Remote Control 24, 774–780.

[33] Wang, J., Ma, Y., L. C., H., W., Liu, J., 2008. Multilevel framework to detect and handle vehcle occlusion. IEEE transaction on Intelligent Transportation Systems 9 (1), 161–174.

[34] Wang, J., Ma, Y., Li, C., Wang, H., Liu, J., 2009. An efficient multi-object tracking method using multiple particle filters. WRI World Congress on Computer science and Information Engineering, 568–572.

[35] Wang, Y., Malinovskiy, Y., Wu, Y., 2008. Occlusion robust and environment insensitive algorithm for vehicle detection and tracking using surveillance video cameras. Technical Report, TransNow Budget No. 61-6020.

[36] Zhang, C., Chen, X., Chen, W., 2006. A pca-based vehicle classification framework. International Conference on Data Engineering Workshops (ICDEW'06).

[37] Zivkovic, Z., Heijden, F., 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters 27 (7), 773–780.