Moving Vehicle Detection for Automatic Traffic Monitoring

Jie Zhou, *Senior Member, IEEE*, Dashan Gao, and David Zhang, *Senior Member, IEEE*

***Abstract*—A video-based traffic monitoring system must be capable of working in various weather and illumination conditions. In this paper, we will propose an example-based algorithm for moving vehicle detection. Different from previous works, this algorithm learns from examples and does not rely on any *a priori* model for vehicles. First, a novel scheme for adaptive background estimation is introduced. Then, the image is divided into many small nonoverlapped blocks. The candidates of the vehicle part can be found from the blocks if there is some change in gray level between the current image and the background. A low-dimensional feature is produced by applying principal component analysis to two histograms of each candidate, and a classifier based on a support vector machine is designed to classify it as a part of a real vehicle or not. Finally, all classified results are combined, and a parallelogram is built to represent the shape of each vehicle. Experimental results show that our algorithm has a satisfying performance under varied conditions, which can robustly and effectively eliminate the influence of casting shadows, headlights, or bad illumination.**

***Index Terms*—Principal component analysis (PCA), statistical learning, support vector machine (SVM), video-based traffic monitoring.**

I. Introduction

In AN intelligent transportation system, traffic data may come from different sensors such as loop detectors, ultrasonic sensors, or cameras. The use of video cameras (many of which are already installed to survey road networks), coupled with computer vision techniques, offers an attractive alternative to other sensors. Video-based camera systems are more sophisticated and powerful because the information content associated with image sequences allows precise vehicle tracking and classification. In contrast, spot sensors have limited capabilities and are often both costly and disruptive to install [1]–[3].

Successful video-based systems for urban traffic monitoring must be adaptive to different weather and illumination conditions. The main difficulties come from cast shadows, vehicle headlights, and noise. Under sunlight, for example, cast shadows always accompany the moving vehicles. Thereby, it may be easily regarded as a part of the vehicle and result in incorrect segmentation. At night, vehicle headlights and bad illumination (which may cause strong noise) can also cause many difficulties for the detection task. Therefore, moving vehicle detection under such situations is always an important but challenging work.

There are many published works concerning moving object detection, including vehicle detection [1]–[10] and human detection [11]–[16]. Among them, there is some literature addressing casting shadow elimination [4], [6]–[10], [16]. In [4], a model of a cast shadow was established on the assumption of knowing the occurrence time, the location of the light source, the scene's geometry, and the shape of the moving object. The authors in [6] and [7] apply an edge detector to separate shadow from vehicles with the knowledge that a shadow is edge-less, and a threshold is chosen to eliminate the shadow. In [8], two simple methods called the landmark-based method and the BS-Edge method are proposed for shadow rejection. In [9], cast shadows are classified from vehicles using the model of the pixels' color appearance. Similarly, the authors in [10] proposed a deterministic approach using the chrominance information, which distinguishes a cast shadow from moving objects in a hue-saturation-value color space on the knowledge, e.g., the shadow has similar chromaticity but lower brightness than that of the same pixel in the background image. Its parameters are chosen empirically, and it is difficult for the tasks to be competent in various illumination changes. In [16], the authors established four criteria for background and light source (e.g., shadows have a penumbra and a soft luminance transition at the contour of the shadow, the background is textured, and etc.), which are designated for laboratory conditions, but these criteria are rarely satisfied for outdoor scenes. In real practice, shadows usually have sharp edges, and the background is often nontextured.

In the published literature, only [2] addressed how to detect the moving vehicles at night. The center of the headlight was treated as a salient feature and extracted by morphological analysis. Thereby, a vehicle was represented with only two points, and there was no other information. As we know, there are streetlights in most urban districts. By the aid of the streetlights' illumination, it is very feasible for us to obtain much more precise information from the traffic image sequences at night.

In pattern recognition tasks, example-based classifiers are widely used because they have no model (or criteria) establishment problems. In this paper, we will propose an algorithm for moving vehicle detection. Different from previous works, this algorithm learns from the known examples and does not rely on the prior model of vehicles, lighting, shadows, or headlights.

Manuscript received August 2, 2002; revised June 13, 2003, September 23, 2005 and January 19, 2006. This work was supported in part by the Natural Science Foundation of China (NSFC) under Grants 60573062, 60205002, and 60332010, the Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP), and the Natural Science Foundation of Beijing (NSFB) under Grant 4042020. The review of this paper was coordinated by Prof. D. Lovell.

J. Zhou and D. Gao are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: jzhou@tsinghua.edu.cn; dgao@ucsd.edu).

D. Zhang is with the Center for Multimedia Signal Processing and Biometrics Technology, Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: csdzhang@comp.polyu.edu.hk).

Digital Object Identifier 10.1109/TVT.2006.883735

First, the background of the scene is estimated adaptively. Then, the image is divided into many small nonoverlapped blocks. By subtracting the current image from the background, the blocks with an intensity change can be found as the candidates for vehicle parts. After that, a low-dimensional feature vector is extracted from each candidate by applying principal component analysis (PCA) to two histograms of the candidate. Since the support vector machine (SVM) approach [17], [18] has a high generalization performance, we use it to design a classifier to classify each vector as part of a real vehicle or others (such as casting shadow, headlight, or strong noise). Finally, all classified results are combined, and a parallelogram is used to represent the vehicle's shape. Experimental results proved that our algorithm has a robust performance under varied conditions.

This paper is organized as follows. In Section II, the description of our algorithm is introduced. Experimental results are presented in Section III. We finish with conclusions and a discussion in Section IV.

II. Algorithm Description

The whole algorithm includes the following steps: background estimation, block division, candidates' selection, feature extraction, SVM-based classification, and shape representation. The flowchart of the algorithm is depicted in Fig. 1. In this paper, we only study the processing of gray-level image sequences instead of color image sequences because the algorithm based on gray-level image sequences has a better generalization and can be more widely utilized in real applications.

A. Background Estimation

In order to detect moving vehicles, we need to estimate the background of the scene first. Some widely used monitoring systems employ algorithms based on interframe difference or the subtraction of an input image from a reference image. However, most researchers have abandoned nonadaptive methods of background estimation because of the need for manual initialization and the limitation of short-term monitoring applications. More and more researches are aiming at adaptive background estimation. A widely used adaptive background estimation algorithm is essentially low-pass filters, which estimate the gray-level intensity of the background from a sequence of input images by running mode or running average [19]. Running mode is reputed to be quite accurate, but its computing speed is so slow that it cannot be considered for real-time application. On the other hand, running average is fast, but its estimation accuracy is rather poor. In [20]–[22], the values of a particular pixel are modeled as a mixture of Gaussian distributions and use an online approximation to update the model. Each pixel that can be represented effectively by the mixtures of Gaussians is considered as part of the background. This algorithm can be adapted to deal with lighting changes and especially repetitive motions of scene elements. In this section, we will propose an improved adaptive background extraction algorithm based on Kalman filtering, which has a much simpler computational complexity and is more suitable for real-time image processing tasks such as automatic traffic monitoring.

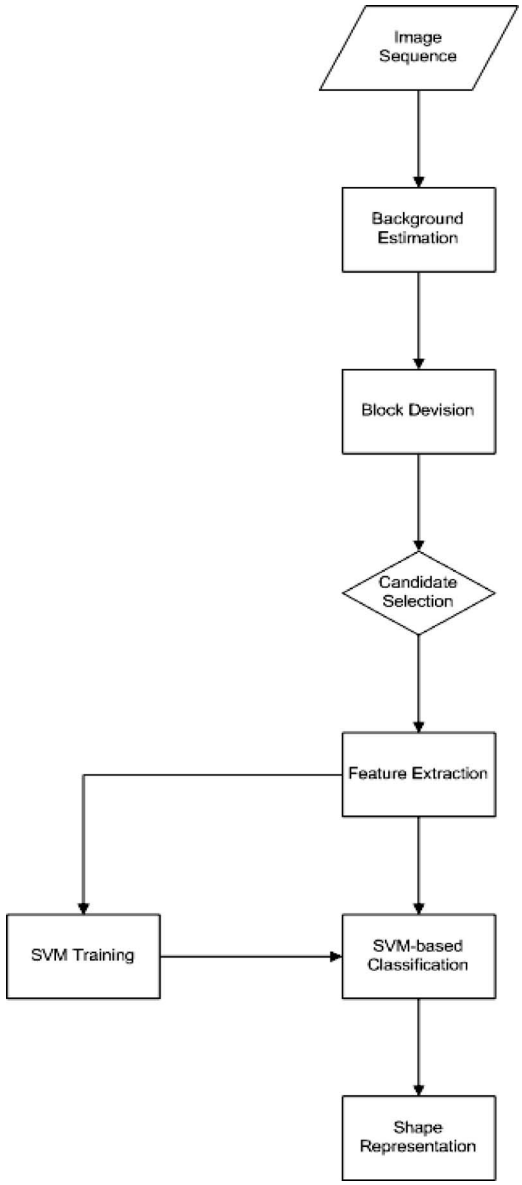

Fig. 1. Flowchart of our algorithm.

In [15], a simple algorithm of background estimation was proposed, based on Kalman filtering. The algorithm assumes that the evolution of the background pixel intensity can be described by a finite-dimension dynamic system and then employs a Kalman filter to update the background image, but the algorithm takes little account of the change of illumination condition in the long term. In [23], the authors proposed an improved Kalman filtering model called the augmented state dynamic model. The algorithm introduces a new nonzero variable component for each pixel to represent the change of illumination intensity. In order to compute the variances of the system and input noise, it is assumed that they are constant to long-term interval, but, in fact, the variances of the noises are varied with the illumination condition. For example, the variance of the noise at night will be much stronger than that in the daytime. Based on the above analysis, we will improve the above model and set up a more reasonable model for a background image for varied situations.

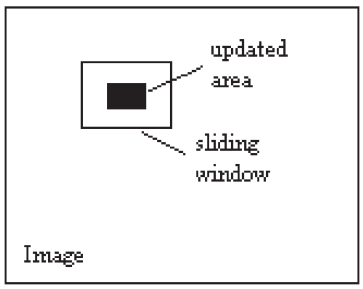

Fig. 2. Illustration of sliding window used in background estimation.

The background on the pixel p of the $(n+1)$th frame can be defined as

$$B(n+1,p) = B(n,p) + \beta(n,p) + \mu(n,p) \tag{1}$$

where $B(n,p)$ is the background on the pixel p of the nth frame, $\beta(n,p)$ is an undefined function describing the changes of illumination intensity along with the time, and $\mu(n,p)$ is a white Gaussian noise with a zero mean and a variance $Q(n,p)$ representing the system error. The input image intensity is described as

$$I(n,p) = B(n,p) + \eta(n,p) \tag{2}$$

where $\eta(n,p)$ is a white Gaussian noise with a zero mean and a variance $R(n,p)$ representing the measurement noise. The difference between our model and that of the study in [23] is as follows: For any pixel p, $Q(n,p)$ and $R(n,p)$ may vary with the different frame (or time). By combining (1) with (2), we have

$$I(n+1,p) = B(n,p) + \omega(n+1,p) \tag{3}$$

where $\omega(n+1,p) = \beta(n,p) + \mu(n,p) + \eta(n+1,p)$. It is easy to know that $\omega(n,p)$ has a Gaussian distribution. Denote $m(n,p)$ and $s(n,p)$ as the mean value and the variance of $\omega(n,p)$, respectively. Obviously, the values of $m(n,p)$ and $s(n,p)$ are not constant in the temporal-spatial space. How to compute them is the key to background estimation.

For traffic monitoring, it is reasonable to suppose that the illumination changes and the distribution of noise are nearly identical in a small local region. That is to say, the values of $m(n,p)$ and $s(n,p)$ are independent of pixel position in the local region; at the same time, therefore, they can also be denoted as $m(n)$ and $s(n)$ in that region, respectively. Then, for a local region, a histogram can be computed from the difference between $\{I(n+1,p)\}$ and $\{B(n,p)\}$. To estimate the value of $m(n)$ and $s(n)$ from the histogram, we use a recursive-least-square (RLS) adaptive filter [33] that can solve the linear filtering problem without invoking assumptions on the statistics of the input.

To update the background of the whole image, we can use a sliding window whose size is chosen as 30×30 in our experiments. To avoid the "mosaic phenomenon," the noise distribution is computed in the larger window, but only a small area at the center of the window is updated, as shown in Fig. 2. The updating method is same with the study in [23]. In practice, we apply two four-order RLS filters to predict the value of $m(n)$ and $s(n)$, respectively.

B. Block Division and Candidates Selection

The image will be divided into nonoverlapped blocks, and each block has the same size in a same image. The block's size is chosen according to different imaging resolution and other parameters such as the height, angle, and zoom factor of the camera. In our experiments, its range is between 8×8 and 30×30.

Now, we will find out the blocks with a gray-level change. The current image will be subtracted from the background to get the difference image, and we compute its mean value for each block. In order to reduce the computational cost, only the blocks with a mean value larger than a predefined threshold will be regarded as candidates containing moving objects. In our experiments, this threshold is chosen as five. Actually, these candidates include real vehicles, casting shadows, headlights, or noise.

C. Feature Extraction

As a simple representation of images, a histogram is invariant to translation and rotation. Compared with the original image, it also has the advantage of small dimension. Therefore, it has been widely used in various applications such as lane detection [23], face recognition [24], [25], and texture retrieval [26]. In this paper, we also use the histogram for feature extraction.

Suppose that the range of an image's gray level is $[0,T]$ (T is 255 in this paper since the images we processed have 256 gray levels). For a difference image $\{D(i,j)\}$, the range of its gray level will be $[-T,T]$. Denote the histogram of $\{D(i,j)\}$ as $\mathrm{ha}(r)$, where $-T \le r \le T$, and it can be easily computed. For convenience, we offset the range of r from $[-T,T]$ to $[0,2T]$. Another histogram, which is denoted as $\mathrm{hb}(r)$, $0 \le r \le T$, can be computed from the image $\{E(i,j)\}$, which is an edge map of $\{D(i,j)\}$ computed by

$$E(i,j) = \frac{1}{2}\{|D(i+1,j+1) - D(i,j)| + |D(i+1,j) - D(i,j+1)|\}. \tag{4}$$

Combine the above two histograms to form a new vector with a dimension of $3T+2$:

$$\mathrm{hc}(r) = \begin{cases} \mathrm{ha}(r), & 0 \le r \le 2T \\ \mathrm{hb}(r-2T-1), & 2T+1 \le r \le 3T+2. \end{cases} \tag{5}$$

In our algorithm, the vector should be normalized in order to adapt to different block division by

$$h(r) = \frac{\mathrm{hc}(r)}{S} \tag{6}$$

where S is the size of the block.

In order to remove the influence of noises from the feature and further reduce dimension, we apply PCA, which is an optimal orthonormal decomposition, to compress the vector $h(r)$. By collecting some sample vectors, a scatter matrix is obtained as

$$S = \frac{1}{N}\sum_{i=1}^{N}(h_i - m)(h_i - m)^{\mathrm{t}} \tag{7}$$

where N is the number of vectors in the training set, h_i is the ith vector, m is the average of all these vectors, and $(h_i - m)^{\mathrm{t}}$ is the transpose of the vector of $(h_i - m)$. For $S \in R^{(3T+2)\times(3T+2)}$, it is easy to obtain its eigenvalues $A = \mathrm{diag}[\lambda_1, \ldots, \lambda_{3T+2}]$, $\lambda_1 \geq \cdots \geq \lambda_{3T+2}$ and its eigen-vectors $V = [\nu_1, \ldots, \nu_{3T+2}]$. Then, a new compressed vector with a dimension p can be computed from the original vector $h(r)$:

$$g_p = [\nu_1, \ldots, \nu_p]^{\mathrm{t}}\, h, \qquad \text{when} \quad p \leq 3T + 2. \tag{8}$$

Using compressed vectors as features, our experiments show that the classification result is better than directly using the original histograms. Moreover, the computation cost can also be reduced.

D. SVM-Based Classification

SVM is based on the principle of structural risk minimization, which states that better generalization capabilities are achieved through a minimization of the bound on the generalization error [17]. It has been widely used in face detection and other recognition tasks [27]–[32].

Let $\{(x_i, y_i)\}_{1\leq i\leq N}$ be a set of training samples, where $x_i \in R^n$, belonging to a class labeled by $y_i \in \{+1, -1\}$, and n is the dimension of the input space. The aim is to define a hyperplane that divides the set of examples such that all the points with the same label are on the same side of the hyperplane [18]. The optimal separating function can be expressed as

$$f(x) = \mathrm{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right) \tag{9}$$

where $K(x, y)$ is a positive symmetric function called the kernel function, and b is a bias estimated on the training set. The parameters $\{\alpha_i\}$ can be achieved by maximizing

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{10}$$

under constraints $\sum_{i=1}^{N} y_i \alpha_i = 0$ and $0 \leq \alpha_i \leq C$. In the nonseparable case, the constant C must be set to a given value, which can be chosen as an arbitrary value by adopting an empirical approach. In our experiments, it is chosen as 1000.

The nature of the decision surface is mainly defined by the kernel functions $K(x, y)$, which should satisfy Mercer's conditions. The commonly used kernels include polynomial kernels $K(x, y) = (x^t y + 1)^d$, where d is a positive integer to define the degree of a polynomial decision surface, and Gaussian kernels $K(x, y) = e^{-g\|x-y\|^2}$.

In applications, we can do the following steps to choose sample data from real traffic video captured under different illumination and weather conditions: Find out the candidates from the videos, and manually divide them into two sets, i.e., with or without real moving vehicle. For all candidates, their feature vectors are extracted. Then, they are put into the SVMs for training, and as a result, the parameters of the SVM-based classifier can be obtained.

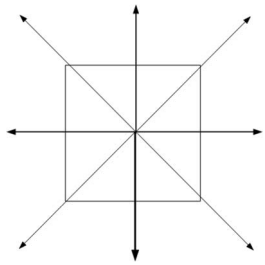

Fig. 3. Eight directions used for eight-connectivity.

On the detection stage, the input image is divided into many blocks. When a block is found as a candidate, a feature vector is generated by PCA from two kinds of histograms of the different image. Then, it is put into the well-trained classifier to judge whether this region is covered with a vehicle or others (such as shadow, headlight, or noise).

E. Shape Representation

A traffic surveillance system should have such functions as vehicle counting, sorting, tracking, and speed estimation. As a basis, the shape information of vehicles needs to be extracted first. That is to say, we need to combine the classified results to approximate the shape of each moving vehicle using a concise but accurate geometrical form. In order to finish the task of shape representation, we assume that the road information, including the direction and position of each lane line in the monitoring area, has been known in advance (in fact, the user can obtain this information by simply marking on the screen at the startup of the system).

After applying the SVM-based classifier on all candidates, those blocks classified as nonvehicles can be discarded, and the others will be transferred to the following steps. At the same time, some parts of real vehicles may also be misclassified as nonvehicles. For example, some dark parts of the vehicles with uniform color could be classified as shadow and be discarded. At night, the case is even worse. Then, we need to take steps to group the remaining blocks and compensate for the misclassification.

We will do the following steps in different lanes. In each lane, the blocks classified as vehicles can be grouped by their eight-connectivity, i.e., all blocks connected with each other along eight directions (see Fig. 3) will belong to a same vehicle. For use at night, the misclassification could cause a single vehicle to be separated into several separate parts. Then, we can set a threshold and measure the distance between each of two adjacent blocks in each lane, and the blocks with a distance smaller than the predefined threshold will be regarded as connective. An optimal threshold is not only related to the moving vehicle itself but also relies on the distance between the other vehicles adjacent to it. Therefore, it can be defined by the average speed and length of the vehicle on this road at night. For simplicity, the threshold is simply set as twice the length of a block's diagonal in our experiments and systems. That is to say, if the distance between the centers of two blocks is smaller than this threshold, these two blocks are thought as connective. Then, all connected blocks will be merged together. We need

to take some steps to compensate for them. To compensate the losing parts in the process of misclassification, we choose to apply the algorithm proposed by the study in [34] which is designed to compute a convex hull from a known set of points. For each vehicle (corresponding to a connected set), the algorithm is taken to derive a convex polygon to enclose all the points in the connected set (see [34] for details).

In traffic monitoring tasks, a thorough delineated contour fitting is not necessary because we only care for statistical information of vehicles such as its length, width, and speed; therefore, it is better for us to use a parallelogram for the approximation of the vehicle's shape. Another advantage of using a parallelogram instead of a convex polygon is that the former will possess a better robustness due to its simple representation. Since we know the information of each lane such as its direction, a parallelogram will be easy to derive from the convex hull to represent the shape of vehicles. Denote the parallelogram as $\overline{P_1P_2P_3P_4}$, where the lines $\overline{P_1P_2}$ and $\overline{P_3P_4}$ should be parallel to the direction of the lane and lines. The two lines $\overline{P_1P_4}$ and $\overline{P_2P_3}$ are parallel too, and their direction can be set in advance, according to the imaging angle (in most cases in our experiments, their direction is chosen to be parallel to the horizontal axis in the image plane). Since the directions of these four lines are known, what we need to do is compute their offsets. For a convex polygon, a natural way is to find the smallest parallelogram to enclose the whole polygon, which can be easily done by projecting the convex polygon along the direction of $\overline{P_1P_2}$ and $\overline{P_2P_3}$, respectively, and finding their beginning and ending points. This kind of parallelogram can roughly describe the shape of vehicles, but since some outer blocks may contain a nonvehicle part, it is better to make the above parallelogram shrink a little to approximate the real shape. That is to say, $\overline{P_1P_2}$, $\overline{P_2P_3}$, $\overline{P_3P_4}$, and $\overline{P_1P_4}$ need to have a shift to four directions, i.e., right, down, up, and left, respectively. The shift is restricted beyond a single block, and it is computed by analyzing the projected distribution of the convex polygon along the directions $\overline{P_1P_2}$ and $\overline{P_2P_3}$, respectively. Take the projection along the lane as an example. Assume the width of a block as a unit length. We can compute the accumulation in each unit extent from left to right. Denote L_1 and L_2 as the accumulation of the first and second unit extent (see Fig. 4). The shift S_{left} of the left margin of the parallelogram $\overline{P_1P_2}$ is defined by

$$S_{\text{left}} = 1 - \min\left(1, \frac{L_1}{L_2}\right). \tag{11}$$

It should be noted that, in most cases, L_1 is smaller than L_2 due to the convexity of the polygon. Similarly, the shift of the other three margins can also be computed.

In the shape representation stage, the overlapped vehicles will be judged as a single one, but they can be easily separated in the tracking scheme (which is not included in this paper). It also should be noted that all training phases in PCA and SVM can be done offline. Therefore, the computation cost of the online stage is rather small, which guarantees the practical use of our algorithm.

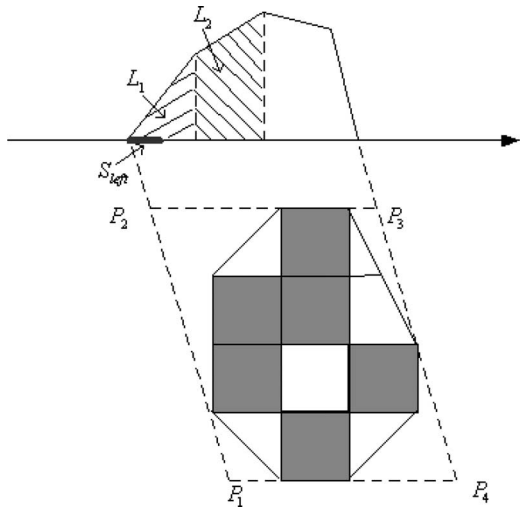

Fig. 4. Illustration of shape representation. The gray blocks are the detected vehicle parts by the SVM classifier, which are grouped by a convex polygon. The parallelogram $\overline{P_1P_2P_3P_4}$ is the smallest parallelogram to enclose the polygon. After projection along the lane, L_1 and L_2 are the accumulation of the first two unit extents (assuming the width of a block as a unit length), respectively, and S_{left} is the shift of $\overline{P_1P_2}$ from left to right when the original parallelogram shrinks.

III. Experimental Results

We collected some real traffic video captured at different times, under different weather conditions. Following the steps stated in Section II, the candidates are extracted from the videos. Out of them, a total of 2009 samples are chosen, in which 1218 samples belong to the vehicles, and the others are nonvehicles (including shadows, headlight, and strong noise). All these samples are divided into two parts: The training set includes 263 vehicles and 246 nonvehicles, which is used to train the SVM-based classifiers, and the others act as the testing set to examine the performance of classifiers. The scatter matrix for PCA is computed using the training samples.

The classification accuracy on the testing set by using different features and different SVM kernel functions is reported in Table I. Out of these features, we choose a 20-dimensional PCA vector (i.e., $p = 20$) for practical use due to its overall performance. The classifiers produced by different kernel functions have a similar accuracy, but SVM with a simpler kernel always has a better generalization; therefore, we decide to apply a polynomial kernel function with two dimensions for the real tasks.

Our algorithm has been implemented into a real traffic monitoring system on a Pentium III 500-MHz PC. Our algorithm can process more than 15 frames a second when the image size is 384 $\times$ 288 pixels. We have compared it with manual counting or loop detectors, a more than 98% accuracy rate of vehicle counting, tracking, classification, or speed estimation are reported in the daytime and more than 90% at night in the outdoor running. It is a rather robust and satisfying result for real applications. Table II presents some results of vehicle counting by applying our algorithm in real traffic scenes. In these experiments, the monitoring region covers two to six lanes, so the vehicles' cast shadow or headlight may influence the neighboring lane frequently. It shows that the system can

TABLE I
CLASSIFICATION ACCURACY ON THE TESTING SET BY USING DIFFERENT FEATURES AND DIFFERENT SVM KERNEL FUNCTIONS

Features used	Polynomial Kernels				Gaussian Kernels		
	d = 2	d = 3	d = 4	d = 5	g = 2	g = 3	g = 6
Original Histogram	93.07%	93.63%	94.36%	94.37%	94.18%	94.49%	93.72%
20-dimensional PCA Vector	95.58%	95.82%	96.03%	96.09%	96.21%	96.45%	96.37%
15-dimensional PCA Vector	94.96%	94.88%	95.37%	95.31%	95.11%	95.43%	95.36%
10-dimensional PCA Vector	92.80%	93.01%	93.37%	93.66%	93.82%	94.01%	94.53%

TABLE II
SOME RESULTS OF VEHICLE COUNTING BY USING OUR ALGORITHM UNDER DIFFERENT CONDITIONS

Weather and Illumination Condition	Number of Vehicles Passing by the Area (by manual counting)	Counted Number of Our Algorithm	Accuracy (%)
Afternoon sunlight with long, heavy shadow	255	256	99.6%
Morning sunlight with long, light shadow	104	104	100%
A rainy day with reflections on the side of vehicles	327	329	99.4%
Nightfall with bright vehicle headlights but no streetlight	98	106	91.8%
At night under streetlight with shadows and headlights	91	96	94.5%

deal with shadows perfectly in the daytime (the accuracy is better than 99%). During nightfall (without streetlight) and at night (with streetlight), our system can still distinguish shadows, vehicle headlights, and strong noise from vehicles with a small error rate (the accuracy is 91.8% and 94.5%, respectively).

To illustrate its effectiveness and robustness, our algorithm is also compared with the algorithms proposed in [10] and [16]. Because the algorithms in [10] and [16] are designed for color images, we randomly choose some color image sequences as the testing sets from real traffic videos. For our algorithm, they are converted into gray image sequences before the further processing. For fairness, these video sequences are different from those used in SVM training. These videos are captured in daytime with thick shadows, in daytime with light shadows, at night under streetlight illumination, and in the dark at nightfall with no streetlight, respectively. Since the authors of [10] did not provide any parameters, the choice of them in this experiment is done manually on the image sequences in order to have the best performance in the daytime and with thick shadows. When implementing the algorithms proposed in [16], the parameters used are the same as those preferred by the authors (see [16, p. 72]).

Some comparison results are shown in Fig. 5. The first column (a1)–(e1) shows the detection results of our algorithm, where white boxes indicate the detected parts of the vehicles. The detection results of the study in [10] and [16] are shown in the second and third column, respectively, where black pixels represent the background, white pixels are those classified as vehicles, and gray pixels are shadow. In order to compare the robustness of each algorithm, the three algorithms are tested using the same data captured under different illumination and weather conditions. In Fig. 5, the first two rows [(a) and (b)] are obtained in the daytime. There are thick shadows, as well as some dark cars in the images. The third row [row (c)] is obtained from a daytime scene with light moving shadows, while the fourth row [row (d)] is at night under streetlight illumination with shadows and considerable noise. Another detect result of our algorithm is shown in the last row [row (e)], which is obtained in the dark at nightfall with no streetlight but strong vehicle headlights in it. The illuminating condition is so poor that vehicles are difficult to detect, and the vehicle headlights are also hard to distinguish from vehicles. From the results in Fig. 5, it is clear that our algorithm can detect most parts of vehicles and eliminates shadows correctly and robustly. Although some dark areas of the vehicles are lost, the detected parts can be easily combined together and represented by parallelograms (see Fig. 6). The algorithm of the study in [10] can detect most parts of the moving vehicles from the cast shadow in (a2) and (b2), which means that the used parameters are suitable for the cases in the daytime and with heavy shadows. In (c2), most of the shadow can be distinguished, but the edges of the shadow are classified as vehicle. That means these parameters cannot work well in the circumstance of light shadow. In (d2), the algorithm has difficulty suppressing shadows correctly. As in (e2), the algorithm proposed in [10] fails to eliminate the disturbance from vehicle light. All the above results show that the algorithm in [10], with a fixed parameter set, is not suitable for use in various illumination and weather conditions, and an adaptive selection of parameters is needed. From the last column [(a3) to (e3)], we can see the results of the algorithm proposed in [16]. In them, large parts of the vehicles are classified as shadows. It shows that this algorithm has a rather poor performance because the criteria used in this algorithm is established to deal with indoor usage and is not suitable for the task of automatic traffic monitoring.

Fig. 6 shows some results of vehicle representation, where the detected parts of each vehicle are combined and represented by a parallelogram. The images used here are the same as those in Fig. 5. From these results, we can see that the parallelograms can effectively describe the vehicles' shape in various cases.

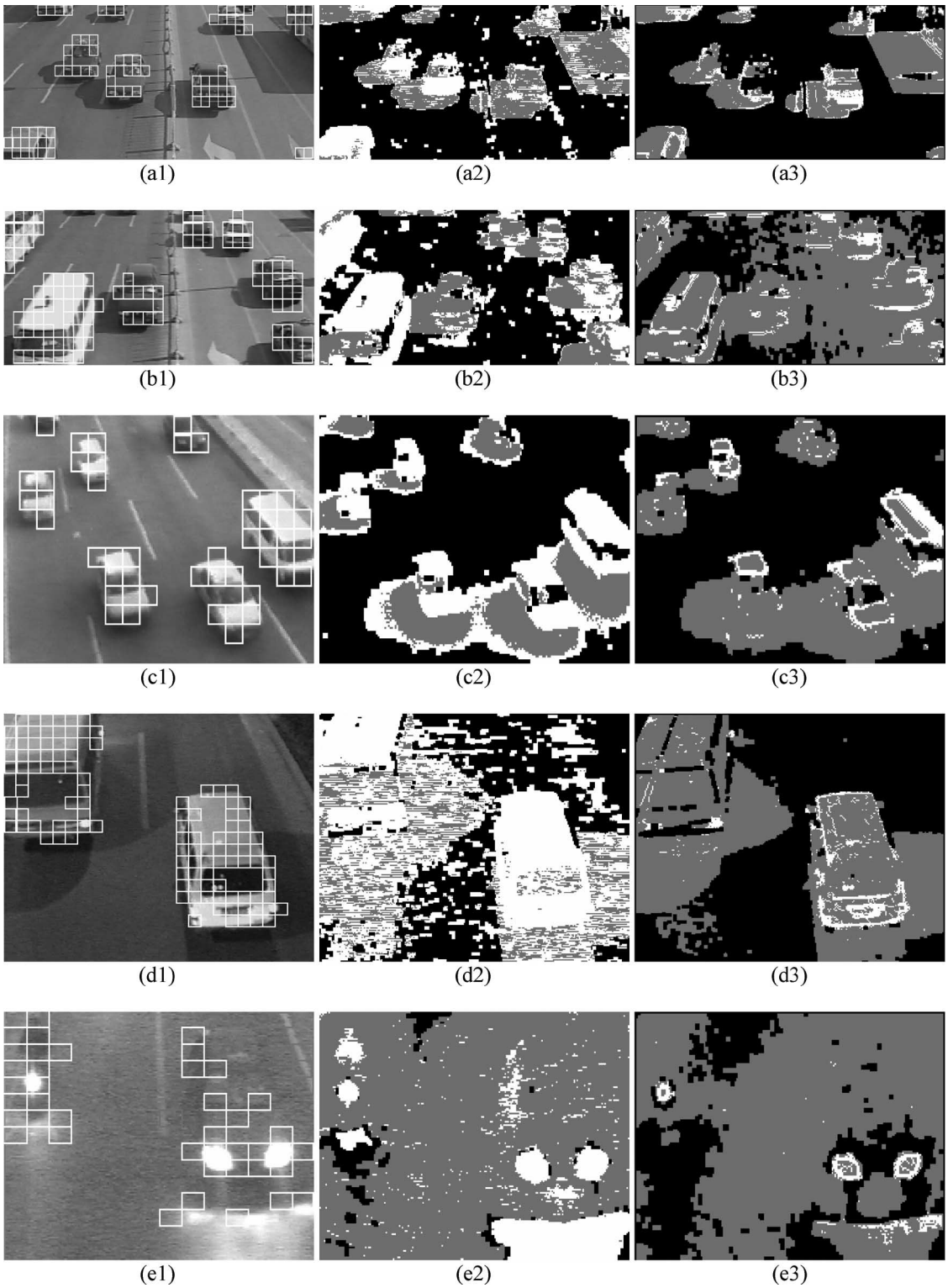

Fig. 5. Some comparison results of moving vehicle detection in real traffic video captured under different illumination and weather conditions. (a1)–(e1) The detection results of our algorithm (white boxes indicate the detected parts of vehicles). (a2)–(e2) and (a3)–(e3) Detection results of the study in [10] and [16], respectively (black pixels represent the background, white pixels are those classified as moving vehicles, and gray ones are shadow).

IV. Conclusion and Discussions

In this paper, we have studied how to detect moving vehicles robustly using an example-based strategy. First, an improved algorithm for adaptive background estimation is proposed. Then, the image is divided into many small nonoverlapped blocks. The candidates of the vehicle's part can be found from the blocks if there is some intensity change between the current image and the background. PCA is used to extract a low-dimensional vector from two histograms of each candidate. The vector will be classified as belonging to the vehicle or not by an

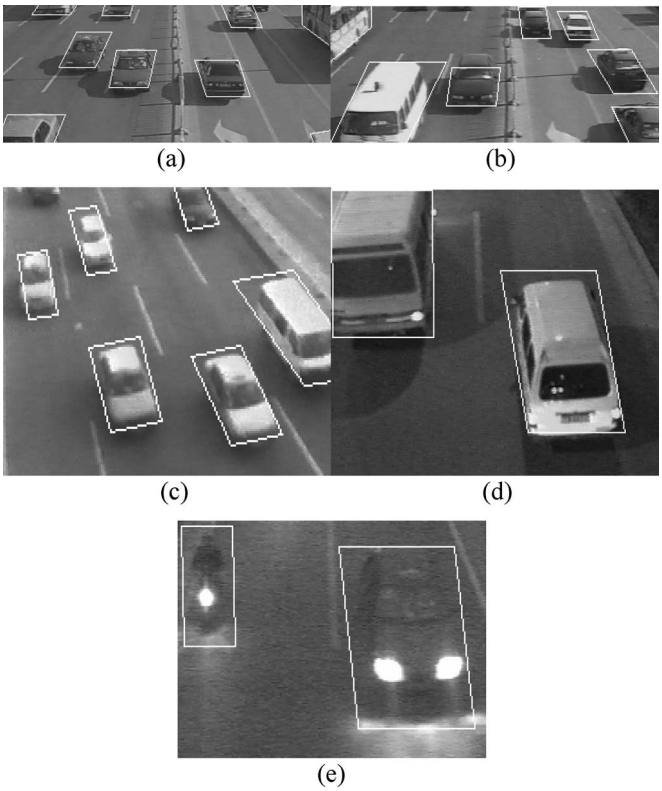

Fig. 6. Some examples of shape representation, which correspond to the images listed in Fig. 5, respectively.

SVM-based classifier that is trained on the examples. Finally, all classified results are combined, and a parallelogram is built to represent the shape of each vehicle. Compared with previous works, this algorithm has no modelling or criteria establishment problems. The experimental results on real traffic video data show that our vehicle detector has strong abilities to deal with different weather and illuminating conditions.

For vehicle detection research the precision on a pixel level other than an object level is desired. In the future, we will improve our algorithm to be more precise. For example, we can apply more information such as color into the algorithm, and some other classifiers can also be considered for future study.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] M. Fathy and M. Y. Siyal, "A window-based image processing technique for quantitative and qualitative analysis of road traffic parameters," *IEEE Trans. Veh. Technol.*, vol. 47, no. 4, pp. 1342–1349, Nov. 1998.

[2] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 119–130, Jun. 2000.

[3] P. G. Michalopoulos, "Vehicle detection through image processing: The autoscope system," *IEEE Trans. Veh. Technol.*, vol. 40, no. 1, pp. 21–29, Feb. 1991.

[4] D. Koller, K. Danilidis, and H. H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, 1993.

[5] E. L. Dagless, A. T. Ali, and J. B. Cruz, "Visual road traffic monitoring and data collection," in *Proc. IEEE-IEE VNIS*, 1993, pp. 146–149.

[6] M. Kilger, "A shadow handler in a video-based real-time traffic monitoring system," in *Proc. IEEE Workshop Appl. Comput. Vis.*, 1992, pp. 11–18.

[7] Z. Zhu and G. Xu, "VISATRAM: A real-time vision system for automatic traffic monitoring," *Image Vis. Comput.*, vol. 18, no. 10, pp. 781–794, 2000.

[8] M. Yu and Y. D. Kim, "Vision based vehicle detection and traffic parameter extraction," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E84A, no. 6, pp. 1461–1470, 2001.

[9] I. Mikic, P. C. Cosman, G. T. Kogut, and M. M. Trevedi, "Moving shadow and object detection in traffic scenes," in *Proc. 15th Int. Conf. Pattern Recog.*, 2000, vol. 1, pp. 321–324.

[10] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, "Improving shadow suppression in moving object detection with HSV color information," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Oakland, CA, 2001, pp. 334–339.

[11] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," *Comput. Vis. Graph. Image Process.*, vol. 80, no. 1, pp. 42–56, 2000.

[12] K. Yoshinari and M. Michihito, "Human motion estimation method using 3-successive video frames," in *Proc. Int. Conf. Virtual Syst. and Multimedia*, 1996, pp. 135–140.

[13] A. Shio and J. Sklansky, "Segmentation of people in motion," in *Proc. IEEE Workshop Visual Motion*, 1991, pp. 325–332.

[14] N. Rota and M. Thonnat, "Video sequence interpretation for visual surveillance," in *Proc. 3rd IEEE Int. Workshop Visual Surveillance*, 2000, pp. 59–68.

[15] K. Karmann, A. Brandt, and R. Gerl, "Moving object segmentation based on adaptive reference images," in *Proc. Eur. Signal Process. Conf.*, 1990, vol. 5, pp. 951–954.

[16] J. Stauder, R. Mech, and J. Ostermann, "Detection of moving cast shadows for object segmentation," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 65–76, Mar. 1999.

[17] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[18] E. Osuna, R. Freund, and F. Girosi, "Support vector machines: Training and applications," Artificial Intelligence Lab. Mass. Inst. Technol., Cambridge, MA, Tech. Rep. 1602, 1997.

[19] A. Lai and N. Yung, "A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence," in *Proc. IEEE Int. Symp. Circuits and Syst.*, 1998, vol. 4, pp. 241–244.

[20] C. Stauffer and W. Grimson, "Adaptive background mixture model for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. and Pattern Recognit.*, 1999, vol. 2, pp. 246–252.

[21] C. Stauffer, W. Eric, and L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[22] A. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric model for background subtraction," in *Proc. ICCV FRAME-RATE Workshop*, 1999, pp. 246–252.

[23] M. Boninsegna and A. Bozzoli, "A tunable algorithm to update a reference image," *Signal Process. Image Commun.*, vol. 16, no. 4, pp. 353–365, 2000.

[24] J. Gonzalez and U. Ozguner, "Lane detection using histogram-based segmentation and decision trees," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2000, pp. 346–351.

[25] R. Brunelli and O. Mich, "Histograms analysis for image retrieval," *Pattern Recognit.*, vol. 34, no. 8, pp. 1625–1637, 2001.

[26] J. Zhou, L. Xin, and D. Zhang, "Scale-orientation histogram for texture image retrieval," *Pattern Recognit.*, vol. 36, no. 4, pp. 1061–1063, 2003.

[27] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proc. IEEE Int. Conf. Comput. Vision*, 1998, pp. 555–562.

[28] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin tags for multiclass classification," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2000.

[29] M. Pontil and A. Verri, "Support vector machines for 3-D object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 6, pp. 637–646, Jun. 1998.

[30] K. I. Kim, J. H. Kim, and K. Jung, "Face recognition using support vector machines with local correlation kernels," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 16, no. 1, pp. 97–111, 2002.

[31] J. Hsieh and Y. Huang, "Multiple-person tracking system for content analysis," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 16, no. 4, pp. 447–462, 2002.

[32] J. Ng and S. G. Gong, "Composite support vector machines for detection of faces across views and pose estimation," *Image Vis. Comput.*, vol. 20, no. 5, pp. 359–368, 2002.

[33] H. Simon, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[34] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.

Jie Zhou (M'01–SM'04) was born on November 1968. He received the B.S. and M.S. degrees, both from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995.

From 1995 to 1997, he served as a Postdoctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China. Currently, he is a Full Professor and Assistant Chair of the Department of Automation, Tsinghua University. His research area includes pattern recognition, image processing, computer vision, and information fusion. In recent years, he has authored more than ten papers in international journals and more than 40 papers in international conferences.

Dr. Zhou is an Associate Editor for the *International Journal of Robotics and Automation*. He received the Best Doctoral Thesis Award from HUST, the First Class Science and Technology Progress Award from the Ministry of Education, China, and the Excellent Young Faculty Award from Tsinghua University in 1995, 1998, and 2003, respectively.

Dashan Gao was born in Ningxia, China, in 1976. He received the B.S. and M.S. degrees from the Department of Automation, Tsinghua University, Beijing, China, in 1999 and 2002, respectively. He is currently working toward the Ph.D. degree at the Department of Electrical and Computer Engineering, University of California, San Diego, CA.

David Zhang (SM'95) received the degree in computer science from Peking University, Beijing, China, in 1974, the M.Sc. and Ph.D. degrees in computer science and engineering from Harbin Institute of Technology, Harbin, China, in 1983 and 1985, respectively, and the second Ph.D. degree in electrical and computer engineering from University of Waterloo, Waterloo, ON, Canada, in 1994.

From 1986 to 1988, he was a Postdoctoral Fellow with Tsinghua University and became an Associate Professor with the Academia Sinica, Beijing, Beijing. Currently, he is a Professor with the Hong Kong Polytechnic University, Hong Kong. His research interests include automated biometrics-based identification, neural systems and applications, and parallel computing for image processing and pattern recognition. So far, he has published over 170 papers, including seven books in his research areas.

Dr. Zhang is the Founder and Director of Biometrics Technology Centre, supported by UGC/CRC, Hong Kong Government. He is the Founder and Editor-in-Chief of the *International Journal of Image and Graphics* and an Associate Editor of *Pattern Recognition* and other five international journals.