# Automatic Vehicle Detection Using Statistical Approach

Chi-Chen Raxle Wang and Jenn-Jier James Lien

Robotics Laboratory, Dept. of Computer Science and Information Engineering,
National Cheng Kung University, No. 1, Ta-Hsueh Road, Tainan, Taiwan
{raxle, jjlien}@csie.ncku.edu.tw
http://robotics.csie.ncku.edu.tw

**Abstract.** This study develops a statistical approach to the automatic detection of vehicles. Compared to traditional approaches, which consider the entire 2-dimensional vehicle region, this study uses three meaningful local features for each vehicle to perform vehicle detection. The proposed approach has a superior tolerance toward wider viewing angles and partial occlusions. Four possible models for vehicle detection are evaluated in the current training and testing processes. For the process of the best model, each local subregion projects into corresponding eigenspace and residual independent basis space with subregion position information. We further simplify the procedure steps of computing the independent component analysis (ICA) in residual space without constructing residual images in order to reduce the computational time. Then the joint probability of projection weight vectors and coefficient vectors of local subregions and positions of local subregions, is used to model the vehicle. Finally, we introduce vector quantization with a new classification method to accelerate the posterior probability calculation.

## 1 Introduction

Automatic license plate identification tools are invaluable for applications such as parking lot access control, easy-pass toll collection, stolen vehicle recovery, etc. Vehicle detection is an essential and integral part of vehicle plate identification. In [4], stereo is used to detect moving targets. In this approach, the target vehicle is identifiable if its grayvalues and the edges of the target region exhibit left-right symmetry. However, this method suffers when partial occlusion occurs since this results in unsymmetrical regions. Posterior probability [5] can be applied to detect moving vehicles based on their edge information. However, using edge information alone renders the vehicle detection process liable to noise and illumination effects. Furthermore, the success of the posterior probability approach relies strongly on the probability of the vehicle appearance falling within a limited range during the training process. For real-time vehicle detection applications, background subtraction method is used in the initial segmentation process of the foreground moving objects and background scenes [3]. Subsequently, vehicle templates and edge information are applied to carry out vehicle detection.

Some detection methods capture the global feature information associated with vehicle images, while others simply capture the local feature information. Although global feature approaches such as [9] and [10] perform reasonably well, in real life, individuals are able to recognize a vehicle from its local features without needing to

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 171 – 182, 2006.

see the entire vehicle. The studies of [7] and [12] have demonstrated that local feature approaches yield better detection results than their global feature counterparts. Local feature approaches such as those proposed in [13], [14], [15], and [16] and part-based approaches, e.g. [1], [6], and [8], have been successfully implemented for object, face, and vehicle detection systems. The latter approaches incorporate an interest-points detector to locate a specified region and to describe their corresponding position in the vehicle image.

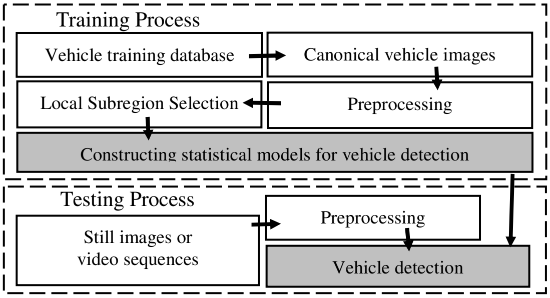
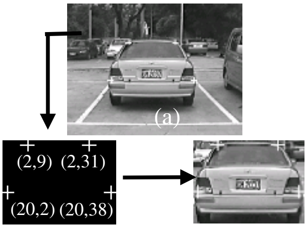

Training Process
Vehicle training database
Canonical vehicle images
Local Subregion Selection
Preprocessing
Constructing statistical models for vehicle detection
Testing Process
Still images or video sequences
Preprocessing
Vehicle detection

**Fig. 1.** Workflow diagram of the proposed vehicle detection system



(b) Vehicle template region: $N_r$*$N_c$ pixels

(c) Canonical vehicle image

**Fig. 2.** Geometrical normalization and canonical vehicle image creation

Posterior probability estimation, including both vehicle and non-vehicle information, is used in [14] for the robust detection of vehicles, faces and objects of different sizes and poses. In this approach, the likelihood evaluation operation is based on the estimated joint probability of wavelet coefficients and their corresponding positions within a given region. The same authors used a similar approach in detecting vehicles and faces of different sizes and poses [15]. Their study adopted principal component analysis (PCA) and considered the position information of a 16*16-pixel subregion in the joint probability estimation procedure. Unfortunately the detection process involved in this kind of studies were rather time-consuming.

To overcome the weaknesses of the methods reported in the literature, this study utilizes posterior probability with both vehicle and non-vehicle information to conduct automatic vehicle detection. The joint probability for the maximum-likelihood estimation procedure considers both meaningful local features and their corresponding positions. This study combines the PCA space and the ICA in residual space to model the vehicle. The performance of the proposed approach is verified through a series of experimental studies. Moreover, in order to reduce the computational time required for the detection process, a vector quantization method with a new classification approach is applied to classify the training vehicle and non-vehicle images into several clusters. However, we accelerate the detection process but it won't decrease the system performance.

## 2 Vehicle Detection System

The vehicle detection system consists of a training process and a testing process, as shown in Figure 1. Current study considers the case of a surveillance system such as that used for access control in schools, which detects only the rear and frontal views of passing small vehicles, such as saloon and SUV. Therefore, the case of detecting passing vehicles from side perspectives, etc. is specifically excluded from the current scope. The vehicle template and canonical vehicle images in the present study are *Nr*(=32) * *Nc*(=41) (rows*columns) pixels.

To create a canonical rear-viewed vehicle image, four corner points are manually selected on the vehicle in the original training image, as shown in Figure 2.a. The vehicle image is then normalized and cropped by using the vehicle template shown in Figure 2.b and applying a process of affine transformation to the four corner points. The resulting canonical vehicle image is shown in Figure 2.c. The same procedure is adopted to generate the canonical frontal-viewed vehicle image.

## 2.1 Local Subregion Selection

The present system considers only local features rather than the entire vehicle in the detection process since in real life, individuals are easily able to recognize a vehicle from its local features, i.e. they do not need to see the vehicle in its entirety. Furthermore, this approach can reduce the alignment error by accommodating geometric distortions of the vehicle appearance (texture or grayvalue) to a certain extent [15]. The proposed approach also increases the detection tolerance in the event of unbalanced targets caused by uneven road surfaces or unstable input sources such as handheld video cameras. Finally, considering a local subregion can improve the overall system performance by reducing the computation of time.

Generally, the significant local features in the rear- and frontal-viewed vehicle images contain high texture components such as roofs, windshields, tail-lights (or head-lights), license plates, rear-viewed mirrors, and the wheels [12]. These features exhibit high variances in the spatial domain. However, these subregions may not always be visible in the vehicle image. For example, the rear-viewed mirrors and wheels may disappear in some situations. Moreover, the subregions around the license plate and the windshield areas are sensitive to different locations and illumination, respectively. Therefore, as shown in Figure 3, the current study opts specifically to ignore these particular significant features, and chooses instead the subregions around the roof and the tail-lights (or head-lights).

## 2.2 Vehicle Detection Using Posterior Probability Function

This study detects both rear- and frontal-viewed vehicles from an input image, *I*, by shifting a window, $I_T$, measuring $N_r$*$N_c$ pixels, pixel by pixel over the entire image. The vehicle is detected if it is found within the window.

The following posterior probability function is used in the vehicle detection procedure:

$$P(vehicle \mid I_T) = \frac{P(I_T \mid vehicle)P(vehicle)}{P(I_T \mid vehicle)P(vehicle) + P(I_T \mid non-vehicle)P(non-vehicle)} \tag{1}$$

It is assumed that the prior probability is uniformly distributed, i.e. *P*(*vehicle*) = *P*(*non-vehicle*) = 0.5. It is also assumed that the likelihood probabilities $P(I_T|vehicle)$ and $P(I_T|non\text{-}vehicle)$ conform to a multivariate Gaussian distribution, i.e.

$$P(I_T \mid C) = \frac{\exp\left[-\frac{1}{2}(I_T - \bar{I}_{C,T})^T \Sigma^{-1} (I_T - \bar{I}_{C,T})\right]}{(2\pi)^{N/2}|\Sigma|^{1/2}} \tag{2}$$

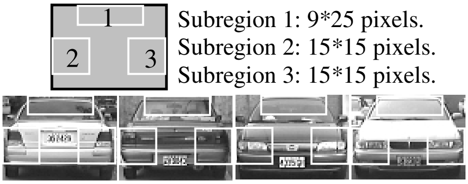

1
2
3
Subregion 1: 9*25 pixels.
Subregion 2: 15*15 pixels.
Subregion 3: 15*15 pixels.

**Fig. 3.** There are three local feature subregions for each canonical vehicle image

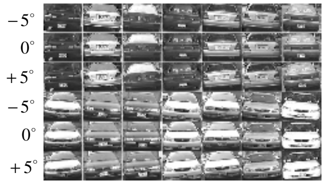


**Fig. 4.** Original ($0^0$) and synthetic canonical vehicle images (-$5^0$ and +$5^0$)

where $C$ is either the vehicle class or non-vehicle class; $\bar{I}_{C,T}$ and $\Sigma$ are the mean vector and the covariance matrix of all canonical training image vectors for class $C$, respectively; and $N$ is the total number of vector dimensions. From equation (2), the Mahalanobis distance $d(I_T)$ [13] is given by:

$$d(I_T) = \tilde{I}_T^{\,T} \Sigma^{-1} \tilde{I}_T = \tilde{I}_T^{\,T} \left[UW^{-1}U^T\right]\tilde{I}_T \approx \tilde{I}_T^{\,T} \left[U_k W_k^{-1} U_k^{\,T}\right]\tilde{I}_T = y_k^T W_k^{-1} y_k \qquad (3)$$

where $\tilde{I}_T = I_T - \bar{I}_{C,T}$, $y_k = U_k^{\,T}\tilde{I}_T$, and $W_k$ and $U_k$ are the first $k$ principal components of the eigenvalue matrix $W$ and its corresponding eigenvector matrix, $U$, of the covariance matrix $\Sigma$, respectively. The input window, $I_T$, is projected into the eigenspace $U_k$ to generate a weight vector $y_k$. Therefore, the Mahalanobis distance can be represented as:

$$d(I_T) = \sum_{i=1}^{k} \frac{y_i^2}{\lambda_i} \qquad (4)$$

Hence, the likelihood probability in equation (2) becomes:

$$P(I_T \mid C) = \frac{Exp\left[-\frac{1}{2}\sum_{i=1}^{k}\frac{y_i^2}{\lambda_i}\right]}{(2\pi)^{k/2}\prod_{i=1}^{k}\lambda_i^{1/2}} \qquad (5)$$

### 2.3 Different Detection Models in the Training and Testing Processes

A separated detection process is employed for rear- and frontal-viewed vehicles. The current training database includes 275 canonical rear-viewed vehicle images and 262 frontal-viewed vehicle images. In order to develop the capability of detecting vehicles moving on uneven roads or shot by handheld video cameras, two additional in-plane roll-rotation image views, i.e. (-$5^0$) and (+$5^0$), are generated synthetically from the original canonical vehicle images ($0^0$).

Therefore, as shown in Figure 4, each canonical vehicle in the training database actually has three associated images. Furthermore, three subregions are defined within each image. In other words, the training database contains a total of 2475 canonical rear-viewed vehicle images (275 * 3(rotations) * 3(subregions)), and a total of 2358 canonical frontal-viewed vehicle images (262 * 3(rotations) * 3(subregions). The images in the database are preprocessed by affine lighting correction and histogram equalization [7]. The intensity over the entire canonical image is then normalized to zero mean and unit variance [15].

Previous studies [7], [11], [14], and [15] have shown that the use of subregions in face detection or recognition applications yields excellent results. Hence, the detection system proposed in this study operates on the basis of three independent subregions rather than over the entire vehicle region. Hence, the likelihood probability is given by:

$$P(I_T \mid C) = \prod_{i=1}^{3} P(subregion_i \mid C) \tag{6}$$

Unfortunately, this approach is computationally expensive when applying posterior probability since it involves a very high-dimensional image vector. Therefore, this study evaluates the performance of four different detection models in the current training and testing processes. Previous studies have confirmed that the PCA method employed in this study has excellent properties. First, the correlation of the neighborhood pixels remains high. Second, a larger eigenvalue implies more significant variance among the original unbiased image vectors. Third, each original image vector can be reconstructed by the linear combination of the major eigenvectors without losing significant characteristics. Furthermore, the ICA applied to the residual subregion spaces in this study also has excellent characteristics [2] and [11]. First, the ICA can capture high-order statistical information. Second, it is suitable for the modeling of non-Gaussian distributed data sets, such as those associated with the residual subregion spaces in the present study. Third, the ICA applied in the residual spaces is robust to illumination and pose variations. The following sections describe in detail the application of the four proposed detection models to rear-viewed vehicle images. However, it is noted that these models are equally applicable to the detection of frontal-viewed vehicle images.

### 2.3.1 1st Model: All Subregions Are Projected into One Single Eigenspace Without Position Information

In the training process, one eigenspace is generated from the 2475 subregions of the total set of canonical rear-viewed vehicle images. The first 32 principal components are captured since the accumulated eigenvalue percentage curve has a turning point at *k*=32. All of the canonical vehicle or non-vehicle subregions are then projected into this eigenspace, which consists of 32 major eigenvectors and hence reduces its dimensions from 225 to 32. Equation (6) becomes:

$$\prod_{i=1}^{3} P(subregion_i \mid C) = \prod_{i=1}^{3} P(projection_i \mid C) \tag{7}$$

where *projection*$_i$ represents a 32-dimensional weight vector for subregion *i*. Figure 5 presents the eigenvectors of this eigenspace. It can be seen that the 1st, 4th and 6th eigenvectors fall mainly within subregion 1, while the 2nd、3rd and 5th eigenvectors fall inside subregions 2 or 3. Finally, the posterior probability equation for the 1st model is given by:

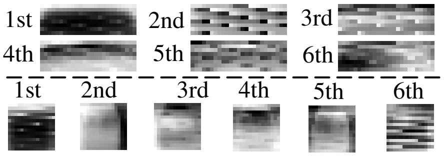


**Fig. 5.** The first six eigenvectors of the 1st model

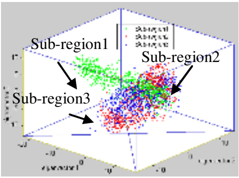


**Fig. 6.** Weight-vector distributions corresponding to canonical vehicle subregions in the 2nd model

$$P(vehicle \mid I_T) = \frac{\prod_{i=1}^{3} P(projection_i \mid vehicle)}{\prod_{i=1}^{3} P(projection_i \mid vehicle) + \prod_{i=1}^{3} P(projection_i \mid non-vehicle)} \tag{8}$$

In the testing process, any input window ($N_r$*$N_c$ pixels), $I_T$, consists of three subregions. Each subregion is projected into the eigenspace to generate a corresponding weight vector. Each of the three input subregions is then compared with the total set of canonical training subregions. Based on equations (5) and (8), it is possible to detect the existence of a vehicle inside this input window by means of the following criterion:

$$\begin{array}{ll} if & (P(vehicle \mid I_T) \geq threshold) \\ then & (vehicle\ \ exists) \\ otherwise & (vehicle\ \ doesn't\ exist) \end{array} \tag{9}$$

### 2.3.2 2nd Model: All Subregions Are Projected into One Single Eigenspace with Position Information

In the training process, this model uses the same eigenspace as that described in the 1st model, above. However, this model also takes into account the feature position of each subregion. The complete set of canonical vehicle subregions are classified in accordance with their positions into three separated groups of weight vectors in a 32-dimensional eigenspace. As can be seen in Figure 6, the distributions of subregions 2 and 3 are overlapped since they are symmetric in the canonical vehicle images. However, the distribution of subregion 1 is very different as a result of the apparent texture differences between itself and subregions 2 and 3. The same classification process is also applied to each of three canonical non-vehicle subregions. Taking the additional position information into account, equation (6) becomes:

$$\prod_{i=1}^{3} P(subregion_i \mid C) = \prod_{i=1}^{3} P(projection_i, pos_i \mid C) \tag{10}$$

where $pos_i$ is the position of subregion $i$ of the given vehicle template region. The posterior probability equation for the 2nd model becomes:

$$P(vehicle \mid I_T) = \frac{\prod_{i=1}^{3} P(projection_i, pos_i \mid vehicle)}{\prod_{i=1}^{3} P(projection_i, pos_i \mid vehicle) + \prod_{i=1}^{3} P(projection_i, pos_i \mid non-vehicle)} \tag{11}$$

In the testing process, each subregion of the input window, $I_T$, is compared with the canonical training subregions located at the corresponding position. The posterior probability of the input window can be evaluated from equations (5) and (11). The existence of vehicles can then be determined by assigning different threshold values in equation (9).

### 2.3.3 3rd Model: Each Subregion Is Projected into Corresponding Eigenspace with Position Information

In addition to taking into account the position of the subregions, this model also generates an eigenspace for each group of canonical vehicle subregions. Hence, three eigenspaces exist for the three subregion groups with different positions. For reasons of consistency, each eigenspace has 32 major eigenvectors, i.e. as in the two models presented above. Figure 7 shows the first three eigenvectors for each subregion eigenspace. Finally, each canonical vehicle or non-vehicle subregion is projected into the corresponding eigenspace to generate a weight vector. Therefore, equation (6) becomes:

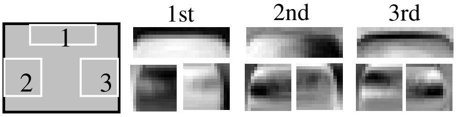


**Fig. 7.** The first three eigenvectors for each subregion eigenspace in the 3rd model

$$\prod_{i=1}^{3} P(subregion_i \mid C) = \prod_{i=1}^{3} P\left(projection_i^i, pos_i \mid C\right) \tag{12}$$

where $projection_i^i$ is the weight vector of the subregion $i$ projected into the corresponding eigenspace $i$. The posterior probability equation for the 3rd model becomes:

$$P(vehicle \mid I_T) = \frac{\prod_{i=1}^{3} P(projection_i^i, pos_i \mid vehicle)}{\prod_{i=1}^{3} P(projection_i^i, pos_i \mid vehicle) + \prod_{i=1}^{3} P(projection_i^i, pos_i \mid non-vehicle)} \tag{13}$$

In the testing process, each subregion of the input vehicle template window, $I_T$, is projected into the corresponding eigenspace to generate a 32-dimensional weight vector. Hence, three weight vectors exist for each input window. The posterior probability of the input window, $I_T$, is calculated from equations (5) and (13). The existence of vehicles can then be determined by assigning different threshold values in equation (9).

### 2.3.4 4th Model: Each Subregion Is Projected into Corresponding Eigenspace and Residual Independent Basis Space with Position Information

This model applies the ICA in the residual spaces to detect the vehicle. The similar work in face recognition [11] performs well in its result. The authors construct ICA in residual space after computing the residual images by subtracting the reconstructed images from the original images. We further derive equations that simplify the procedure steps of computing the ICA in residual space without constructing residual images, and then apply Bayesian theory to detect vehicles. The equations we developed require less complicated calculations.

The independent components, which form non-orthogonal axes, describe the residual subregion spaces of the three subregion groups with different positions. The residual subregion spaces (see Figure 8.c) represent the difference between the original subregion images (see Figure 8.a) and the PCA reconstructed subregion images (see Figure 8.b). It is found that the PCA reconstructed subregions are similar to low-pass filtered versions. The residual subregion images, which contain high frequency components, are less sensitive to illumination variations.

ICA is applied in the residual subregion spaces since these spaces are non-Gaussian distributions. Therefore, to achieve a detection operation, which is robust to illumination and pose variation effects, each residual subregion image is represented by a linear combination of independent components.

Each residual subregion image, $\triangle subregion$, can be obtained by equation 14:

$$\Delta subregion = subregion - subregion' \tag{14}$$

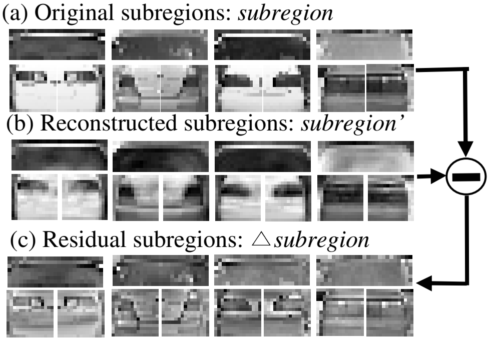
where *subregion* is the original subregion image, and *subregion*' is the PCA reconstructed subregion image. They are given by:

$$subregion = UU^T * subregion = \begin{bmatrix} U_{k'} & U_h \end{bmatrix} \begin{bmatrix} U_{k'}^T \\ U_h^T \end{bmatrix} * sugregion \tag{15.a}$$

and

$$subregion' = U_{k'}U_{k'}^{T} * subregion \tag{15.b}$$

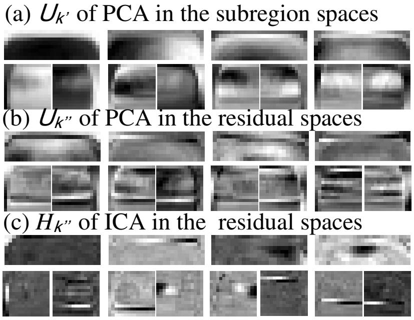
where $U_{k'}$ (see Figure 9.a) is the first $k'$ principal components in eigenvector matrix $U$, $U_h$ is the $h$ residual principal components, and $N$ is $k' + h$. The first $k'$ ($k'$=7) components are chosen based on the Gaussian axes assumption and the $h$ residual principal components are based on non-Gaussian axes assumption. Therefore, △*subregion* can be rewritten by using following equation, i.e.


(a) Original subregions: *subregion*

(b) Reconstructed subregions: *subregion'*

(c) Residual subregions: △*subregion*

**Fig. 8.** The process for the residual subregion images


(a) $U_{k'}$ of PCA in the subregion spaces

(b) $U_{k''}$ of PCA in the residual spaces

(c) $H_{k''}$ of ICA in the residual spaces

**Fig. 9.** First row is the first $k'$ principle components in $U$. Second row is the remaining $k''$ residual principle components $U_{k''}$. Third row is the independent basis $H_{k''}$ in the residual spaces.

$$\begin{aligned} \Delta subregion &= UU^{T} * subregion - U_{k'}U_{k'}^{T} * subregion \\ &= (U_{k'}U_{k'}^{T} * subregion + U_{h}U_{h}^{T} * subregion) - U_{k'}U_{k'}^{T} * subregion \\ &= U_{h}U_{h}^{T} * subregion \approx U_{k''}U_{k''}^{T} * subregion \end{aligned} \tag{16}$$

where $U_{k''}$ (see Figure 9.b) is the first $k''$ ($k''$=29) principal components in $U_h$. As a result, the residual subregion weight vector can be calculated by $U_{k''}^{T} * subregion$ . In addition, by applying ICA to $U_{k''}$, statistically independent basis images $H_{k''}$ with dimensions $k''$ can be generated. $H_{k''}$ (see Figure 9.c) is represented by

$$H_{k''}^{T} = T_{k''}U_{k''}^{T} \tag{17}$$

where $T_{k''}$ is the weight matrix. Bell and Sejnowski's algorithm [2] is used to estimate $T_{k''}$, which is an invertible matrix. Thus, the residual subregions image can be reconstructed by:

$$\begin{aligned} \Delta subregion &= U_{k''}U_{k''}^{T} * subregion = H_{k''}(T_{k''}^{-1})^{T} U_{k''}^{T} * subregion \\ &= H_{k''}(U_{k''}T_{k''}^{-1})^{T} * subregion = H_{k''}B \end{aligned} \tag{18}$$

Therefore, △*subregion* consists of $(U_{k''}T_{k''}^{-1})^{T} * subregion = B$ , which are linear combination coefficients of the independent basis images, $H_{k''}$. Here, the ICA transformation matrix is denoted as $ICA\_TranM_{k''}$ and is computed by:

$$ICA\_TranM_{k''} = (U_{k''}T_{k''}^{-1}) \tag{19}$$

From equation (3), the Mahalanobis distance $d(I_T)$ becomes:

$$\begin{aligned} d(I_T) &= subregion^T * \Sigma^{-1} * subregion = subregion^T * \left[UW^{-1}U^T\right] * subregion \\ &\approx subregion^T * \left[U_{k'}W_{k'}^{-1}U_{k'}^T + U_{k''}W_{k''}^{-1}U_{k''}^T\right] * subregion \\ &= y_{k'}^T W_{k'}^{-1} y_{k'} + \left(U_{k''}^T * subregion\right)^T W_{k''}^{-1}\left(U_{k''}^T * subregion\right) \end{aligned} \tag{20}$$

where $y_{k'} = U_{k'}^T * subregion$ is the weight vector based on eigenvectors $U_{k'}$. The residual subregion weight vector is then transformed to linear combination coefficients of $H_{k''}$ by means of equation (18), i.e., $B = (U_{k''}T_{k''}^{-1})^T * subregion$. Therefore, equation (20) for the Mahalanobis distance can be represented as:

$$d(I_T) = \sum_{i=1}^{k'} \frac{y_i^2}{\lambda_i} + \sum_{\substack{i=k'+1 \\ j=1}}^{\substack{i<=(k'+k'') \\ j<=k''}} \frac{B_j^2}{\lambda_i} \tag{21}$$

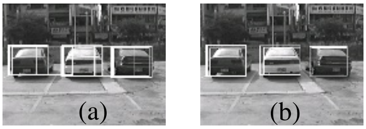


**Fig. 10.** Vehicle detection without and with position information of the subregions, as show in (a) and (b), respectively

**Table 1.** The performances of different models evaluated by testing database (a). (PC: P4 3G Hz. 'FA': False Alarm. 'SF': Seconds/Frame.)

| | 1st M | 2nd M | 3rd M | 4th M | 4th M+VQ |
|---|---|---|---|---|---|
| **R:DR %** | 87.0% | 87.6% | 86.6% | 92.8% | 91.5% |
| **F:DR %** | 89.1% | 89.5% | 88.4% | 94.0% | 93.4% |
| **R:FA** | 73 | 53 | 47 | 37 | 46 |
| **F:FA** | 59 | 43 | 41 | 28 | 44 |
| **SF** | 4.856 | 1.643 | 2.455 | 3.455 | 0.28 |

So, the likelihood probability in equation (5) becomes:

$$P(I_T \mid C) = \frac{Exp\left[-\frac{1}{2}(y_{k'}^T W_{k'}^{-1} y_{k'} + B^T W^{-1} B)\right]}{(2\pi)^{(k'+k'')/2} \prod_{i=1}^{(k'+k'')} \lambda_i^{1/2}} = \frac{Exp\left[-\frac{1}{2}\sum_{i=1}^{k'} \frac{y_i^2}{\lambda_i}\right]}{(2\pi)^{k'/2} \prod_{i=1}^{k'} \lambda_i^{1/2}} * \frac{Exp\left[-\frac{1}{2}\sum_{\substack{i=k'+1 \\ j=1}}^{\substack{i<=(k'+k'') \\ j<=k''}} \frac{B_j^2}{\lambda_i}\right]}{(2\pi)^{k''/2} \prod_{i=k'+1}^{(k'+k'')} \lambda_i^{1/2}} \tag{22}$$

The posterior probability equation for the 4th model becomes:

$$P(vehicle \mid I_T) = \frac{\prod_{i=1}^{3} P(projection_i^i, ICACoeff_i^i, pos_i \mid vehicle)}{\prod_{i=1}^{3} P(projection_i^i, ICACoeff_i^i, pos_i \mid vehicle) + \prod_{i=1}^{3} P(projection_i^i, ICACoeff_i^i, pos_i \mid non-vehicle)} \tag{23}$$

where $projection_i^i$ is weight vector of the subregion $i$ projected into the corresponding eigenspace $i$, and $ICACoeff_i^i$ is the ICA coefficient vector of the subregion $i$ projected into the corresponding independent basis $i$.

In the testing process, each subregion of the input window, $I_T$, is projected into the corresponding eigenspace and the corresponding independent basis space to generate a $k'$-dimensional weight vector and a $k''$-dimensional ICA coefficient vector, respectively. Hence, three weight vectors and three ICA coefficient vectors exist for each input window. The posterior probability of the input window, $I_T$, is calculated from equations (22) and (23). The existence of vehicles can then be determined by assigning different threshold values in equation (9).

## 3 Experiment Results

A testing database of 457 vehicle images was compiled from the internet and from images captured using handheld video cameras. In total, the database contained 303 rear-viewed vehicle images and 154 frontal-viewed vehicle images. The vehicles in the testing images displayed a wide variety of size and orientation. Moreover, the images featured various background sceneries, lighting conditions and degree of occlusion. In addition, we also tested the following published vehicle databases: MIT CBCL Group 187 rear- and 252 frontal-viewed vehicles images and Caltech Vision Group 526 rear-viewed vehicles images.

Initially, the input image was processed by applying a low-pass filter to remove noises. This image was then down-sampled from original resolution of 240*320 pixels (level 0) to 32*43 pixels (level 15) by a factor of 7/8. In the searching window extraction process, searching window $I_T$ of 32*41 pixels, which is exactly the same size as the vehicle template region, was employed to conduct vehicle detection by shifting this window pixel by pixel at each level.

The non-vehicle information was extracted from the false acceptance subregions by applying the vehicle detection process to the original training vehicle images. The actual vehicle subregion inside the false acceptance vehicle region is not qualified as non-vehicle information. We collected about 10000 images of rear-viewed non-vehicle and 9800 images of frontal-viewed non-vehicle. A similar collection method has been used in [8], [14] and [15].

Figure 10 illustrates the effect of including feature position information in the vehicle detection process. Figure 10.a shows the result of vehicle detection when the feature position information is not considered (1st model). Ignoring this information causes false acceptances between two neighboring vehicles, since subregion 1 encloses the top edge profile of the wall, which resembles the roof profile of a vehicle. In Figure 10.b, the individual vehicles are correctly detected by including feature position information in the detection process (2nd model). The 3rd and 4th models also solve above problem by considering position information.

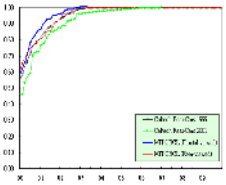


**Fig. 11.** ROC curves (x-axis is false detection rate and y-axis is detection rate) for vehicle detection using the 4th model.

**Table 2.** Detection rate comparison using the Caltech rear-viewed vehicle database

| | Our 4th model | Fergus, et al. [6] |
|---|---|---|
| **Detection Rate** | 92% | 84.8% |

The four models described in the previous sections were applied to our testing image database. The corresponding experimental results are listed in Table 1. It can be seen that the 4th model yields the best performance, while the 3rd model yields the poorest results. Therefore, the 4th model represents the best approach for vehicle detection. It has the lowest false detection rate and the highest detection rate. The 4th model was then applied to each of the MIT CBCL group and Caltech vision group 1999 and 2001 testing databases. The resulting ROC curves, as shown in Figure 11,

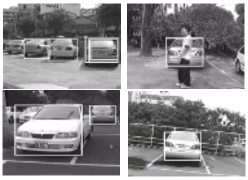
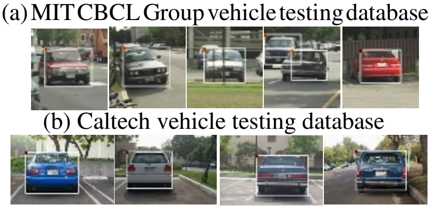


**Fig. 12.** Tolerances of our vehicle detection



(a) MIT CBCL Group vehicle testing database

(b) Caltech vehicle testing database

**Fig. 13.** Detection example of using the 4th model

also show consistent and promising performance. Table 2 demonstrates that the proposed system of using the 4th model provides better results for rear-viewed vehicle detection than the method proposed in [6]. In addition, the current vehicle detection system is tolerant to pan and roll rotations, scaling, and partial occlusions, as demonstrated in Figure 12. Some experimental results are shown in Figure 13.

## 4 Speedup by Vector Quantization

In order to find the maximum posterior probability, it is necessary to compare each weight vector with all the canonical subregions. It is very time consuming (see [15]) because the number of training subregions is huge. To speed up computation, we use vector quantization to classify all the training vehicle and non-vehicle weight vectors into clusters (explained later). Now the comparison occurs between the input weight vector and each of the clustering weight-vector centers.

The training vehicle and non-vehicle weight vectors create two codebooks independently by using the likelihood probability in equation (22) for the measure of the nearest neighbor rule. The initial classification process is only for those weight vectors, whose likelihood probabilities pass the threshold (0.8). Next, the same process and threshold apply on remaining weight vectors started from the center of remaining weight vectors. We repeat the same process until the remaining weight vectors belong to the same cluster or the total cluster numbers do not change. The computational time and result are show in Table 1.

## 5 Discussions and Conclusion

This study has developed an automatic vehicle detection system based on a statistical approach. Meaningful local features are considered in this detection process. Four possible models for vehicle detection have been proposed in order to overcome the problem of inefficiency associated with traditional methods, and to determine the factors affecting successful vehicle detection. The current experiments have shown that the false alarm rate is directly influenced by the feature position information of the subregions. The 2nd, 3rd and 4th models have lower false alarm rate since they consider the position information of the subregions. The 1st model has the highest false alarm rate because it does not consider the position information of the subregions. It is also found that the detection rate is directly affected by the correlation of the neighborhood pixels, which is a feature of the PCA method. The 1st and 2nd models exhibit similar detection rates because they share the same eigenspace. Meanwhile, the 3rd model yields an inferior detection rate because it uses three individual eigenspaces with wider distribution variances, particularly in subregion 1. This model is sensitive to variations in lighting conditions and vehicle orientation.

The 4th model represents the promising result for vehicle detection. It has the lowest false detection rate and the highest detection rate because the 4th model models parts of each local subregion eigenspace as a Gaussian distribution, while it models residual space as a non-Gaussian distribution. That is, it not only models low frequency information by PCA, but also models high frequency information by ICA applied in the residual space, which can overcome the drawbacks caused by the sensitivity to lighting conditions and vehicle orientation in the 3rd model. Therefore, the 4th model is tolerant of limited pan and roll rotations, and partial occlusion.

## References

1. S. Agarwal and D. Roth, "Learning to Detect Objects in Images via a Sparse, Part-based Representation ", IEEE Tran. on PAMI, Vol. 26, No. 11, pp. 1475-1490, 2004.
2. M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski, “Face Recognition by ICA”, IEEE Tran. on Neural Networks, Vol. 13, No. 6, pp. 1450- 1464, November 2002.
3. M. Betke, E. Haritaoglu, and L.S. Davis, “Multiple Vehicle Detection and Tracking in Hard Real Time”, IEEE Intelligent Vehicles Symposium, pp. 351-356, 1996.
4. A. Broggi, “Visual Perception of Obstacles and Vehicles for Platooning”, IEEE Tran. on ITS, Vol. 1, No. 3, pp. 164-176, 2000.
5. F. Dellaert, “CANSS: A Candidate Selection and Search Algorithm to Initialize Car Tracking”, CMU-RI-TR-97-34,1997.
6. R. Fergus, P. Perona, and A. Zisserman, “Object Class Recognition by Unsupervised Scale-Invariant Learning”, IEEE Con. on CVPR, Vol. 2, pp. 264-271, 2003.
7. B. Heisele, P. Ho, J. Wu, and T. Poggio, “Face Recognition: Component-based versus Global Approaches”, Computer Vision and Image Understanding, Vol. 91, No. 1/2, pp. 6-21, 2003.
8. F. Jurie and C. Schmid, "Scale-Invariant Shape Features for Recognition of Object Categories", IEEE Con. on CVPR, pp. 90-96, 2004.
9. M. Kagesawa, S. Ueno, K. Ikeuchi, and H. Kashiwagi, “Recognizing Vehicles in Infrared Images Using IMAP Parallel Vision Board”, IEEE Tran. on ITS, Vol. 2, pp. 10-17, 2001.
10. T. Kato, Y. Ninomiya, and I. Masaki, “Preceding Vehicle Recognition Based on Learning From Sample Images”, IEEE Tran. on ITS, Vol. 3, No. 4, pp. 252-260, 2002.
11. T.K. Kim, H. Kim, W. Hwang, S.C. Kee, and J. Kittler, “Independent Component Analysis in a Facial Local Residual Space”, IEEE Tran. on PR, 37, pp. 1873-1885, 2004.
12. B. Leung, “Component-based Car Detection in Street Scene Images”, Master Thesis, Department of Electrical Engineering and Computer Science, MIT, May 2004.
13. B. Moghaddam and A. Pentland, “Probabilistic Visual Learning for Object Representation”, IEEE Tran. on PAMI, Vol. 19, No. 7, pp. 696–710, 1997..
14. H. Schneiderman and T. Kanade, “Object Detection Using the Statistic of Parts”, Int.l Journal of Computer Vision, Vol. 56, pp. 151-177, February 2003.
15. H. Schneiderman and T. Kanade, “Probabilistic Modelig of Local Appearance and Spatial Relationships for Object Recognition”, IEEE Con. on CVPR, pp. 45-51, 1998.
16. Z. Sun, G. Bebis, and R. Miller, "Object Detection Using Feature Subset Selection", Pattern Recognition Letter, Vol. 37, pp. 2165-2176, 2004.