Monocular Precrash Vehicle Detection: Features and Classifiers

Zehang Sun, George Bebis, and Ronald Miller

***Abstract*—Robust and reliable vehicle detection from images acquired by a moving vehicle (i.e., on-road vehicle detection) is an important problem with applications to driver assistance systems and autonomous, self-guided vehicles. The focus of this work is on the issues of feature extraction and classification for rear-view vehicle detection. Specifically, by treating the problem of vehicle detection as a two-class classification problem, we have investigated several different feature extraction methods such as principal component analysis, wavelets, and Gabor filters. To evaluate the extracted features, we have experimented with two popular classifiers, neural networks and support vector machines (*SVMs*). Based on our evaluation results, we have developed an on-board real-time monocular vehicle detection system that is capable of acquiring grey-scale images, using Ford's proprietary low-light camera, achieving an average detection rate of 10 Hz. Our vehicle detection algorithm consists of two main steps: a multiscale driven hypothesis generation step and an appearance-based hypothesis verification step. During the hypothesis generation step, image locations where vehicles might be present are extracted. This step uses multiscale techniques not only to speed up detection, but also to improve system robustness. The appearance-based hypothesis verification step verifies the hypotheses using Gabor features and *SVMs*. The system has been tested in Ford's concept vehicle under different traffic conditions (e.g., structured highway, complex urban streets, and varying weather conditions), illustrating good performance.**

***Index Terms*—Gabor filters, neural networks (NNs), principal component analysis (PCA), support vector machines (SVMs), vehicle detection, wavelets.**

I. Introduction

EVERY minute, on average, at least one person dies in a vehicle crash. Auto accidents also injure at least ten million people each year, and two or three million of them seriously. The hospital bill, damaged property, and other costs will add up to 1%–3% of the world's gross domestic product [1]. Each year in the United States, motor vehicle crashes account for about 40,000 deaths, more than three million injuries, and over $130 billion in financial losses [2]. The loss is too startling to be ignored. With the aim of reducing injury and accident severity, precrash sensing is becoming an area of active research among automotive manufacturers, suppliers and universities. Several national and international projects have been launched over the past several years to investigate new technologies for improving safety and accident prevention.

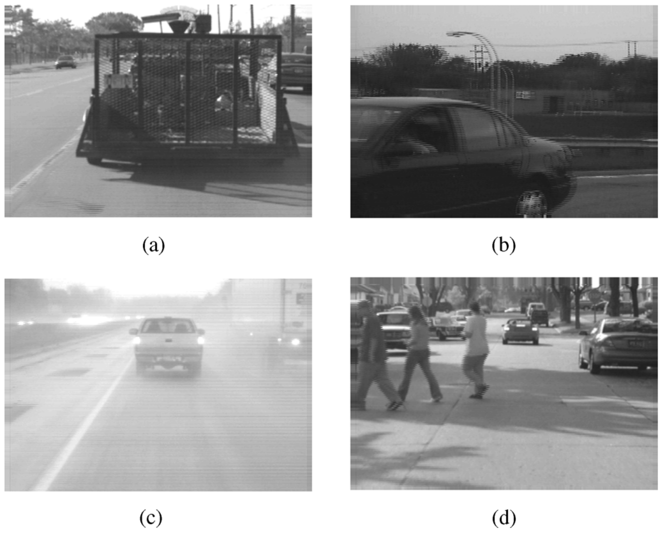


Fig. 1. Variety of vehicle appearances poses a big challenge for vehicle detection.

Vehicle accident statistics disclose that the main threats a driver is facing are from other vehicles. Consequently, an on-board automotive driver assistance system aiming to alert a driver about driving environments, and possible collision with other vehicles has attracted a lot of attention. In these systems, robust and reliable vehicle detection is the first step—a successful vehicle detection algorithm will pave the way for vehicle recognition, vehicle tracking, and collision avoidance. The focus of this paper is on the problem of optical sensor based vehicle detection. A comprehensive review on on-road vehicle detection systems can be found in [3], while more general overviews of intelligent driver assistance systems can be found in [4], [5], [2].

Vehicle detection based on optical sensors is very challenging due to huge within-class variabilities. For example, vehicles may vary in shape [Fig. 1(a)], size, and color. Also, vehicle appearance depends on its pose [Fig. 1(b)] and is affected by nearby objects. Complex outdoor environments, e.g., illumination conditions [Fig. 1(c)], cluttered background, and unpredictable interactions between traffic participants [Fig. 1(d)] are difficult to control. Using on-board moving cameras makes some well established techniques, such as background subtraction, unsuitable. Moreover, on-board vehicle

Manuscript received November 29, 2004; revised October 14, 2005. This work was supported in part by the Ford Motor Company under Grant 2001332R; in part by the University of Nevada, Reno, under an Applied Research Initiative (ARI) Grant; and in part by the National Science Foundation under CRCD Grant 0088086. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zoltan Kato.

Z. Sun was with the Computer Vision Laboratory, University of Nevada, Reno, NV 89557 USA. He is now with eTreppid Technologies, LLC, Reno, NV 89521 USA (e-mail: zehang@etreppid.com).

G. Bebis is with the Computer Vision Laboratory, Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557 USA (e-mail: bebis@cse.unr.edu).

R. Miller is with the Vehicle Design R&A Department, Ford Motor Company, Dearborn, MI 48126-2798 USA (e-mail: rmille47@ford.com).

Digital Object Identifier 10.1109/TIP.2006.877062

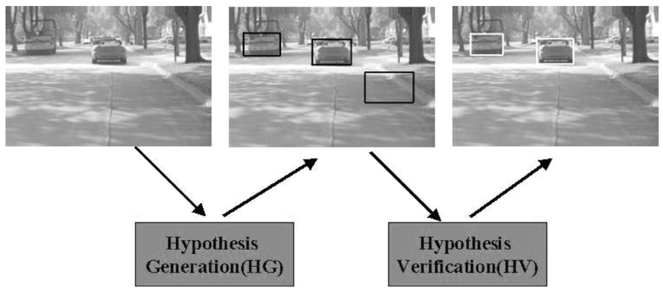


Fig. 2. Illustration of the two-step vehicle detection strategy.

detection systems have high computational requirements. They need to be able to process acquired images at real-time or close to real-time to save more time for driver reaction.

A. Two-Step Scheme

Although it is a challenging task to be accomplished, many optical sensor based vehicle detection algorithms and systems have been proposed and implemented. The majority of them follow two basic steps: 1) hypothesis generation (HG) where the locations of possible vehicles in an image are hypothesized, and 2) hypothesis verification (HV), where tests are performed to verify the presence of vehicles in an image (see Fig. 2).

1) Hypothesis Generation: The objective of the HG step is to provide some candidate locations quickly for further exploration. Methods reported in the literatures fall in one of the following three basic categories: knowledge-based, stereo-vision-based, and motion-based.

Knowledge-based methods employ a priori knowledge to hypothesize vehicle locations in an image. We review below some approaches using information about symmetry [6], color [7], shadow [8], horizontal/vertical edges [9], and texture [10].

Stereo-based approaches take advantage of the inverse perspective mapping (IMP) [11] to estimate the locations of vehicles and obstacles in images. Bertozzi *et al.* [12] computed the IMP both from the left and right cameras. By comparing the two IMPs, they were able to find objects that were not on the ground plane. Using this information, they determined the free space in front of the vehicle. In [13], the IPM was used to wrap the left image to the right image. Knoeppel *et al.* [14] developed a stereo-system detecting vehicles up to 150 m. The main problem with stereo-based methods is that they are sensitive to the recovered camera parameters. Accurate and robust methods are required to recover these parameters because of vehicle vibrations due to vehicle motion or windy conditions [15].

Motion-based methods detect vehicles and obstacles using optical flow. Generating a displacement vector for each pixel (continuous approach), however, is time-consuming and also impractical for a real-time system. In contrast to continuous methods, discrete methods reported better results using image features, such as color blobs [16], lines [17], or local intensity minima and maxima [18]. In particular, a real-time system was reported in [17], implemented on a general-purpose image processor, achieving a processing speed of 10 to 50 ms/frame which is among the fastest reported in the literature although the system does not perform verification. The main idea was testing whether three horizontal lines on the vehicle satisfy the motion constraint of the ground plane or that of the surface plane of the vehicle.

2) Hypothesis Verification: The input to the HV step is the set of hypothesized locations from the HG step. During HV, tests are performed to verify the correctness of a hypothesis. HV approaches can be classified into two main categories: 1) template-based and 2) appearance-based. Template-based methods use predefined patterns of the vehicle class and perform correlation between an input image and the template. Betke *et al.* [19] proposed a multiple-vehicle detection approach using deformable grayscale template matching. In [20], a deformable model was formed from manually sampled data using *PCA*. Both the structure and pose of a vehicle were recovered by fitting the *PCA* model to the image.

Appearance-based methods learn the characteristics of the vehicle class from a set of training images which capture the variability in vehicle appearance. Usually, the variability of the nonvehicle class is also modelled to improve performance. First, each training image is represented by a set of local or global features. Then, the decision boundary between the vehicle and nonvehicle classes is learned either by training a classifier [e.g., neural network (*NN*)] or by modeling the probability distribution of the features in each class (e.g., using the Bayes rule assuming Gaussian distributions). In [21], *PCA* was used for feature extraction and NNs for classification. Goerick *et al.* [9] used a method called local orientation coding (*LOC*) to extract edge information. The histogram of *LOC* within the area of interest was then provided to a *NN* for classification.

A statistical model for vehicle detection was investigated by Schneiderman *et al.* [22], [23]. A view-based approach based on multiple detectors was used to cope with viewpoint variations. The statistics of both object and "nonobject" appearance were represented using the product of two histograms with each histogram representing the joint statistics of a subset of *PCA* features in [22] or Haar wavelet features in [23] and their position on the object. A different statistical model was investigated by Weber *et al.* [24]. They represented each vehicle image as a constellation of local features and used the expectation-maximization (EM) algorithm to learn the parameters of the probability distribution of the constellations. An interest operator, followed by clustering, was used to identify important local features in vehicle images. Papageorgiou *et al.* [25] have proposed using the Haar wavelet transform for feature extraction and support vector machines (*SVMs*) for classification. In the past, Haar-like features (i.e., gradient features) have been used for obstacle detection in a number of studies [26], [27].

B. Main Contributions

The focus of this work is on feature extraction and classification methods for on-road vehicle detection. Different feature extraction methods determine different subspaces within the original image space either in a linear or nonlinear way. These subspaces are essentially the feature spaces, where the original images are represented and interpreted differently. "Powerful" features with high degree of separability are desirable for any pattern classification system. Generally speaking, it is hard to say which feature set is more powerful. The discrimination power of a feature set is usually application dependent. In this

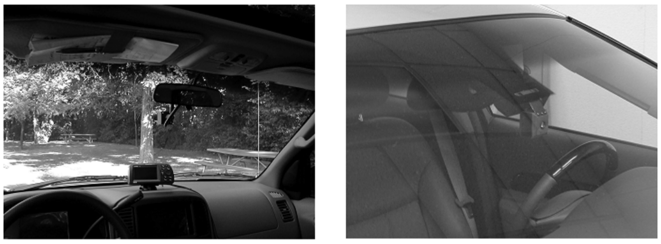


Fig. 3. Low-light camera in the prototype vehicle.

paper, we have investigated six different feature extraction methods (PCA features, wavelet features, truncated/quantized wavelet features, Gabor features, and combined wavelet and Gabor features) in the context of vehicle detection. Some of these features, such as PCA and Wavelet features, have been investigated before for vehicle detection, while others, such as quantized/truncated wavelet and Gabor features, have not been fully explored. To evaluate the extracted features for vehicle detection, we performed experiments using two powerful classifiers: *NNs* and *SVMs*.

The evaluation results have guided us to develop a real-time, rear-view vehicle-detection system from gray scale images using Ford's proprietary low-light camera. A forward facing camera has been installed inside Ford's prototype vehicle which is connected to a frame-grabber of an embedded system (see Fig. 3). Camera images are digitally captured and processed in real-time enabling vehicle detection on timescales on the order of 10 Hz. Our detection system consists of two steps: a multiscale driven hypothesis generation step and an appearance-based hypothesis verification step. Multiscale analysis in HG provides not only robust hypothesis generation but also speeds up the detection process. In appearance-based hypothesis verification, Gabor filters are used for feature extraction and SVMs for classification.

The rest of the paper is organized as follows. In Section II, we provide a brief overview of the system developed. A description of the multiscale driven hypothesis generation step is given in Section III. Various features and classifiers are detailed in Section IV. Comparisons of various *HV* approaches are presented in Section VI. The final real-time system and its performances are presented in Section VII. Our conclusions and directions for future research are given in Section VIII.

II. Monocular Precrash Vehicle Detection System Overview

Precrash sensing is an active research area with the aim of reducing injury and accident severity. The ability to process sporadic sensing data from multiple sources (radar, camera, and wireless communication) and to determine the appropriate actions (belt-pretensioning, airbag deployment, and brake-assist) is essential in the development of active and passive safety systems. To this end, Ford Research Laboratory has developed several prototype vehicles that include in-vehicle precrash sensing technologies such as millimeter wavelength radar, wireless vehicle-to-vehicle communication, and a low-light Ford proprietary optical system suitable for image recognition. An embedded and distributed architecture is used in the vehicle to

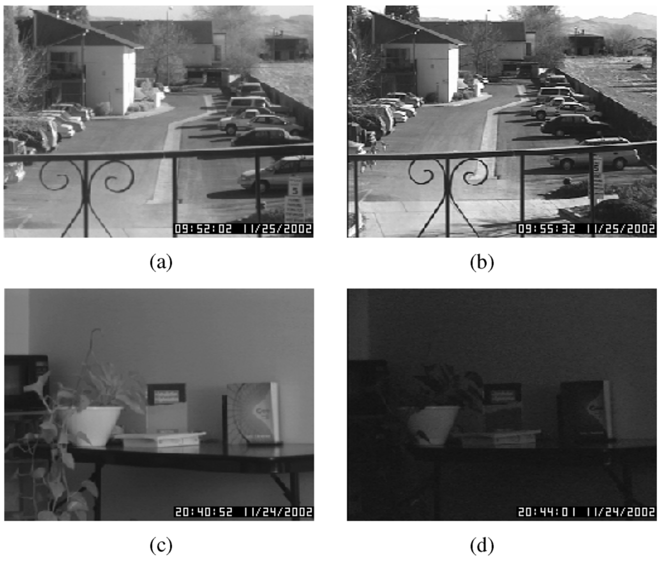


Fig. 4. Low-light camera versus normal camera. (a) Low-light camera daytime image. (b) Same scene caught using normal camera. (c) Low-light camera nighttime image. (d) Same nighttime scene caught suing normal camera.

process the sensing data, determine the likelihood of an accident, and when to warn the driver. This Smart Information Management System (SIMS) forms the cornerstone to Ford's intelligent vehicle system design and is responsible for determining the driver safety warnings. Depending on the situation, SIMS activates an audible or voice alert, visual warnings, and/or a belt-pretensioning system. Extensive human factor studies are underway to determine the appropriate combination of precrash warning technologies, as well as the development of new threat assessment algorithms that are robust in an environment of heterogeneous sensing technologies and vehicles on the roadway.

The optical system represents a principal component in precrash sensing and, with the introduction of inexpensive camera systems, can form a ubiquitous sensing tool for all vehicles. The vehicle prototypes have forward and rearward facing cameras enabling a nearly 360° field of view. Fig. 3 shows the orientation of the forward facing camera in the vehicle prototypes. Forward facing cameras are also mounted in the side-mirror housings and are used for pedestrian and bicycle detection as well as to see around large vehicles. The Ford proprietary camera system was developed jointly between Ford Research Laboratory and Sentech. The board level camera uses a Sony x-view CCD with specifically designed electronic profiles to enhance the camera's dynamic range, thereby enabling daytime and nighttime operation without blooming. Fig. 4(a) and (c) shows the dynamic range of the low-light camera, while Fig. 4(b) and (d) shows the same scene images caught under same illumination conditions by using a normal camera. Obviously, the low-light camera provides much wider dynamic range.

III. Multiscale Driven Hypothesis Generation

To hypothesize possible vehicle locations in an image, prior knowledge about rear vehicle view appearance could be used. For example, rear vehicle views contain lots of horizontal and vertical structures, such as rear-window, fascia, and bumpers. Based on this observation, the following procedure could be

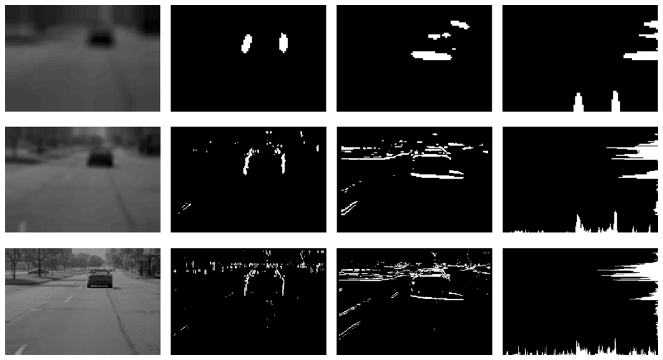


Fig. 5. Multiscale hypothesis generation. The size of the images in the first row are: 90×62; second row: 180×124; and third row: 360×248. The images in the first column have been obtained by applying low pass filtering at different scales; second column: vertical edge maps; third column: horizontal edge maps; and fourth column: vertical and horizontal profiles. Note that all images have been scaled back to 360×248 for illustration purposes.

applied to hypothesize candidate vehicle locations. First, interesting horizontal and vertical structures could be identified by applying horizontal and vertical edge detectors. To pick the most promising horizontal and vertical structures, further analysis would be required, for example, extracting the horizontal and vertical profiles of the edge images and perform some analysis to identify the strongest peaks (e.g., last row of Fig. 5).

Although this method could be very effective, it depends on a number of parameters that affect system performance and robustness. For example, we need to decide the thresholds for the edge detection step, the thresholds for choosing the most important vertical and horizontal edges, and the thresholds for choosing the best maxima (i.e., peaks) in the profile images. A set of parameter values might work well under certain conditions, however, they might fail in other situations. The problem is even more severe for on-road vehicle detection since the dynamic range of the acquired images is much bigger than that of an indoor vision system.

To deal with this issue, we have developed a multiscale approach which combines subsampling with smoothing to hypothesize possible vehicle locations more robustly. Assuming that the input image is f, let set $f^{(K)} = f$. The representation of $f^{(K)}$ at a coarser level $f^{(K-1)}$ is defined by a reduction operator. For simplicity, let us assume that the smoothing filter is separable, and that the number of filter coefficients along one dimension is odd. Then it is sufficient to study the one-dimensional (1-D) case

$$\begin{aligned} f^{K-1} &= \mathrm{REDUCE}(f^K) \\ f^{K-1}(x) &= \Sigma_{n=-N}^{N} c(n) f^K(2x-n) \end{aligned} \tag{1}$$

where the REDUCE operator performs down-sampling and $c(n)$ are the coefficients of a low pass (i.e., Gaussian) filter.

The size of the input images from our video capturing card is 360×248. We use three levels of detail: $f^K(360 \times 248)$, $f^{K-1}(180 \times 124)$, and $f^{K-2}(90 \times 62)$. At each level, we process the image by applying the following steps: 1) low pass filtering (e.g., first column of Fig. 5); 2) vertical edge detection (e.g., second column of Fig. 5), vertical profile computation of the edge image (e.g., last column of Fig. 5), and profile filtering using a low pass filter; 3) horizontal edge detection (e.g., third column of Fig. 5), horizontal profile computation of the edge image (e.g., last column of Fig. 5), and profile filtering using a low pass filter; 4) local maxima and minima detection (e.g., peaks and valleys) of the two profiles. The peaks and valleys of the profiles provide strong information about the presence of a vehicle in the image.

Starting from the coarsest level of detail (f^{K-2}), first, we find all the local maxima at that level. Although the resulted low resolution images have lost fine details, important vertical and horizontal structures are mostly preserved (e.g., first row of Fig. 5). Once we have found the maxima at the coarsest level, we trace them down to the next finer level f^{K-1}. The results from f^{K-1} are finally traced down to level f^K, where the final hypotheses are generated. Candidate vehicles generate two peaks in the vertical profile (i.e., due to the left and right sides of the vehicle) and at least one peak in the horizontal profile (i.e., due to the bottom of the vehicle). It should be noted that due to the complexity of the scenes, some false peaks are expected to be found. We use some heuristics and constraints to get rid of as many of them as possible, for example, the ratio of successive maxima and minima, the absolute value of a maximum, and perspective projection constraints under the assumption of flat surface (i.e., road). To generate the hypotheses, we consider the two vertical peaks with each of the horizontal peaks left. Each triplet of peaks allows us to define a rectangular area that encloses the vehicle. The rectangular area is defined by the intersection of a vertical stripe, defined by the two vertical peaks, and a horizontal stripe, defined by the horizontal peak and heuristics on the aspect ratio of vehicles. Hypotheses that enclose the vehicle poorly are eventually rejected by the verification step. These rules are applied at each level of detail.

The proposed multiscale approach improves system robustness by making the hypothesis generation step less sensitive to the choice of parameters. Forming the first hypotheses at the lowest level of detail is very useful since this level contains only the most salient structural features. Besides improving robustness, the multiscale scheme speeds-up the whole process since the low resolution images have much simpler structure as illustrated in Fig. 5 (i.e., candidate vehicle locations can be found faster and easier). Several examples are provided in Fig. 6 (left column).

IV. Appearance-Based Hypothesis Verification

Verifying a hypothesis is essentially a two-class pattern classification problem (i.e., vehicle versus nonvehicle). Building a pattern classification system requires finding an optimum decision boundary among the classes to be categorized. In most cases, pattern classification involves “concepts” having huge within-class variability (e.g., vehicles), rather than specific objects. As a result, there is no easy way to come up with a decision boundary to separate certain“conceptual objects” against others. A feasible approach is to learn the decision boundary from a set of training examples.

The majority of real-world pattern classification problems require supervised learning where each training instance is associated with a class label. Building a pattern classification system under this scenario involves two main steps: 1) extracting a

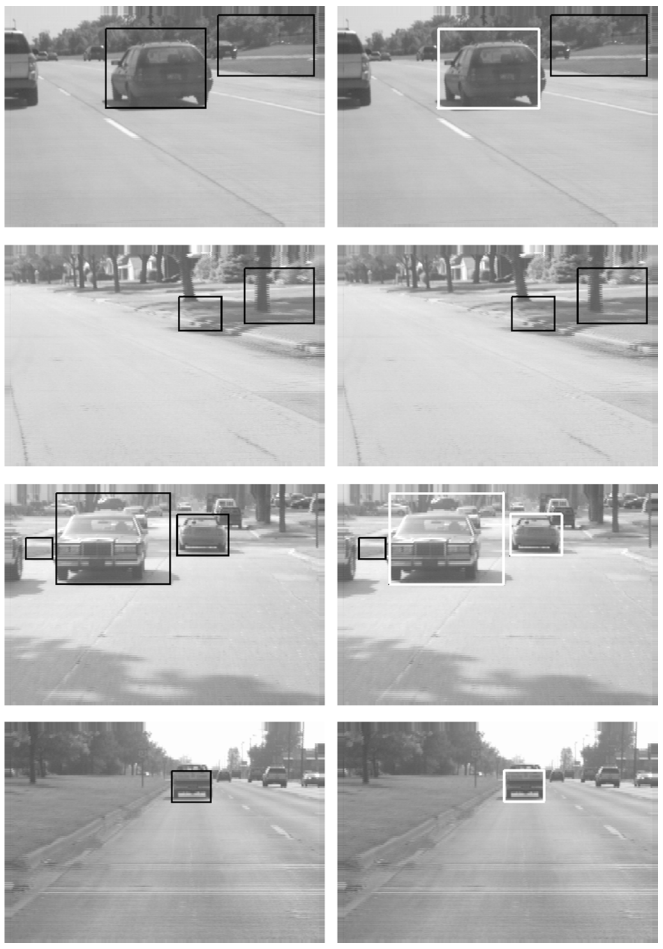


Fig. 6. Examples of the (left column) HG and (right column) HV steps: The black boxes indicate the hypothesized locations while the white boxes are the ones verified by the HV step.

number of features and 2) training a classifier using the extracted features to distinguish among different class instances. The ultimate goal of any pattern classification system is to achieve the best possible classification performance, a task that is highly dependent on the features and classifier employed.

In most cases, relevant features are often unknown *a priori*. The goal of feature extraction is to determine an appropriate subspace of dimensionality m in the original feature space of dimensionality d where m is less than or equal to d [28]. Depending on the nature of the task at hand, the features can be extracted either manually or automatically by applying transformations. The transformations used for feature extraction perform dimensionality reduction which could be either linear or nonlinear. Transformation-based methods have the potential of generating better features than the manual ones, however, the new features may not have clear physical meanings.

A. Feature Extraction

1) PCA Features: Eigenspace representations of images use PCA [29] to linearly project an image in a low-dimensional space. This space is spanned by the principal components (i.e., eigenvectors corresponding to the largest eigenvalues) of the distribution of the training images. After an image has been projected in the eigenspace, a feature vector containing the coefficients of the projection is used to represent the image. Here, we just summarize the main ideas [29].

Representing each image $I(x,y)$ as a $N \times N$ vector Γ_i, first the average face Ψ is computed

$$\Psi = \frac{1}{R}\sum_{i=1}^{R}\Gamma_i \tag{2}$$

where R is the number of faces in the training set. Next, the difference Φ of each face from the average face is computed: $\Phi_i = \Gamma_i - \Psi$. Then the covariance matrix is estimated by

$$C = \frac{1}{R}\sum_{i=1}^{R}\Phi_i\Phi_i^T = AA^T \tag{3}$$

where, $A = [\Phi_1\Phi_2\ldots\Phi_R]$. The eigenspace can then be defined by computing the eigenvectors μ_i of C. Since C is very large ($N \times N$), computing its eigenvector will be very expensive. Instead, we can compute ν_i, the eigenvectors of A^TA, an $R \times R$ matrix. Then μ_i can be computed from ν_i, as follows:

$$\mu_i = \sum_{j=1}^{R}\nu_{ij}\Phi_j, j = 1\ldots R. \tag{4}$$

Usually, we only need to keep a smaller number of eigenvectors R_k corresponding to the largest eigenvalues. Given a new image Γ, we subtract the mean ($\Phi = \Gamma - \Psi$) and compute the projection

$$\widetilde{\Phi} = \sum_{i=1}^{R_k} w_i\mu_i. \tag{5}$$

where $w_i = \mu_i^T\Gamma$ are the coefficients of the projection. We refer to $\{w_i\}$ as eigen-features.

2) Gabor Features: Gabor filters provide a mechanism for obtaining some degree of invariance to intensity due to global illumination, selectivity in scale, as well as selectivity in orientation. Basically, they are orientation and scale tunable edge and line detectors. Vehicles do contain strong edges and lines at different orientation and scales, thus, the statistics of these features could be very powerful for vehicle verification.

The general function $g(x,y)$ of the two-dimensional (2-D) Gabor filter family can be represented as a Gaussian function modulated by an oriented complex sinusoidal signal

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{\tilde{x}^2}{\sigma_x^2} + \frac{\tilde{y}^2}{\sigma_y^2}\right)\right]\exp[2\pi jW\tilde{x}] \tag{6}$$

$$\tilde{x} = x\cos\theta + y\sin\theta \text{ and } \tilde{y} = -x\sin\theta + y\cos\theta \tag{7}$$

where σ_x and σ_y are the scaling parameters of the filter, W is the center frequency, and θ determines the orientation of the filter, and its Fourier transform $G(u,v)$ is given by

$$G(u,v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\}. \tag{8}$$

Gabor filters act as local bandpass filters. Fig. 7(a) and (b) shows the power spectra of two Gabor filter banks (the bright areas indicate spatial frequencies and wave orientation).

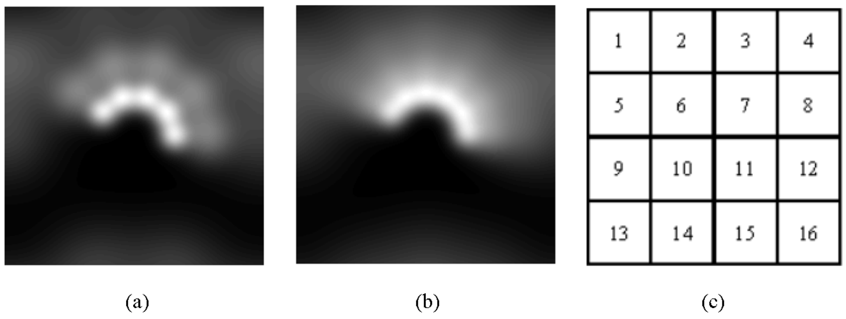


Fig. 7. (a) Gabor filter bank with three scales and five orientations. (b) Gabor filter bank with four scales and six orientations. (c) Feature extraction subwindows.

In this paper, we use the design strategy described in [30]. Given an input image $I(x,y)$, Gabor feature extraction is performed by convolving $I(x,y)$ with a Gabor filter bank. Although the raw responses of the Gabor filters could be used directly as features, some kind of postprocessing is usually applied (e.g., Gabor-energy features, thresholded Gabor features, and moments based on Gabor features [31]). We use Gabor features based on moments, extracted from several subwindows of the input image.

In particular, each hypothesized subimage is scaled to a fixed size of 32×32. Then, it is subdivided into nine overlapping 16×16 subwindows. Assuming that each subimage consists of 16 8×8 patches [see Fig. 7(c)], patches 1, 2, 5, and 6 comprise the first 16×16 subwindow, 2, 3, 6, and 7 the second, 5, 6, 9, and 10 the fourth, and so on. The Gabor filters are then applied on each subwindow separately. The motivation for extracting—possibly redundant—Gabor features from several overlapping subwindows is to compensate for errors in the hypothesis generation step (e.g., subimages containing partially extracted vehicles or background information), making feature extraction more robust.

The magnitudes of the Gabor filter responses are collected from each subwindow and represented by three moments: the mean μ_{ij}, the standard deviation σ_{ij}, and the skewness κ_{ij} (i.e., i corresponds to the ith filter and j to the jth subwindow). Using moments implies that only the statistical properties of a group of pixels is taken into consideration, while position information is essentially discarded. This is particularly useful to compensate for errors in the hypothesis generation step (i.e., errors in the extraction of the subimages). Suppose we are using $S = 2$ scales and $K = 3$ orientations (i.e., $S \times K$ filters). Applying the filter bank on each of the nine subwindows yields a feature vector of size 162, having the following form:

$$[\mu_{11}\sigma_{11}\kappa_{11}, \mu_{12}\sigma_{12}\kappa_{12}, \dots \mu_{69}\sigma_{69}\kappa_{69}]. \tag{9}$$

We have experimented with using the first two moments only, however, much worst results were obtained which implies that the skewness information is very important for our problem. Although we believe that the fourth moment (kurtosis, a measure of normality) would also be very helpful, we do not use it since it is computationally expensive.

3) Wavelet Features: Wavelets are a essentially a multiresolution function approximation method that allow for the hierarchical decomposition of a signal or image. Several reasons make these features attractive for vehicle detection. First, they form a compact representation. Second, they encode edge information, an important feature to represent the general shape of vehicles as a class. Third, they capture information from multiple resolution levels. Finally, there exist fast algorithms, especially in the case of Haar wavelets, for computing them.

Any given decomposition of a signal into wavelets involves just a pair of waveforms (mother wavelet and scaling function). The two shapes are translated and scaled to produce wavelets (wavelet basis) at different locations (positions) and on different scales (durations). We formulate the basic requirement of multiresolution analysis by requiring a nesting of the spanned spaces as

$$\cdots V_{-1} \subset V_0 \subset V_1 \cdots \subset L^2. \tag{10}$$

In space V_{j+1}, we can describe finer details than in space V_j. In order to construct a multiresolution analysis, a scaling function ϕ is necessary, together with a dilated and translated version of it

$$\phi_i^j(x) = 2^{j/2}\phi(2^j x - i). \quad i = 0, \dots, 2^j - 1. \tag{11}$$

The important features of a signal can be better described or parameterized, not by using $\phi_i^j(x)$ and increasing j to increase the size of the subspace spanned by the scaling function, but by defining a slightly different set of function $\psi_i^j(x)$ that span the difference between the spaces spanned by various scales of the scale function. These functions are the wavelets, which spanned the wavelet space W_j such that $V_{j+1} = V_j \bigoplus W_j$, and can be described as

$$\psi_i^j(x) = 2^{j/2}\psi(2^j x - i). \quad i = 0, \dots, 2^j - 1. \tag{12}$$

Different scaling functions $\phi_i^j(x)$ and wavelets $\psi_i^j(x)$ determine various wavelet transforms. In this paper, we use the Haar wavelet which is the simplest to implement and computationally the least demanding. Furthermore, since Haar basis forms an orthogonal basis, the transform provides a nonredundant representation of the input images. The Haar scaling function is

$$\phi(x) = \begin{cases} 1, & \text{for } 0 \le x < 1 \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

and the Haar wavelet is defined as

$$\psi(x) = \begin{cases} 1, & \text{for } 0 \le x < \frac{1}{2} \\ -1, & \text{for } \frac{1}{2} \le x < 1 \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

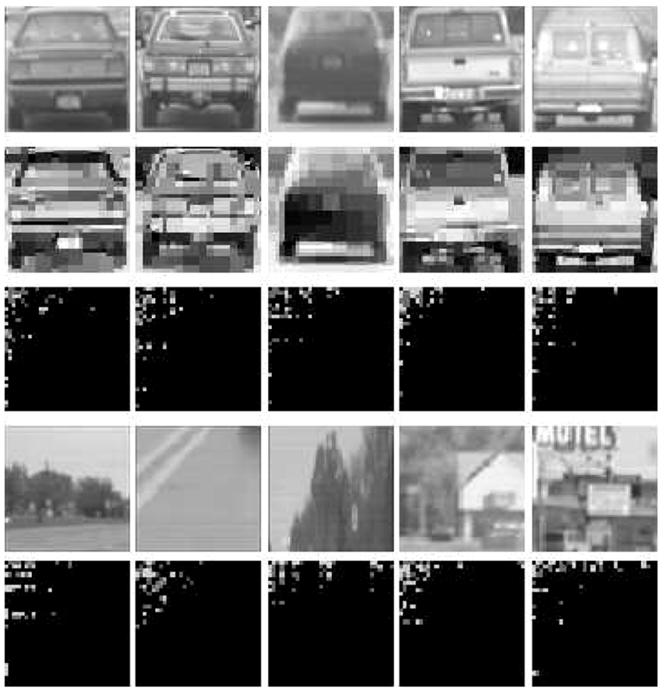


Fig. 8. First row: Vehicle subimages used for training. Second row: Reconstructed subimages using the 50 largest coefficients. Third row: Illustration of the 50 quantized largest coefficients. Fourth and Fifth rows: Similar results for some nonvehicle subimages.

Wavelets capture visually plausible features of the shape and interior structure of objects. Features at different scales capture different levels of detail. Coarse scale features encode large regions while fine scale features describe smaller, local regions. All these features together disclose the structure of an object in different resolutions.

We use the wavelet decomposition coefficients as our features directly. We do not keep the coefficients in the *HH* subband of the first level since they encode mostly fine details and noise [23], which is not helpful at all given we aim to model the general shape of the vehicle class.

4) Truncated and Quantized Wavelet Features: For a $N \times N$ image, there are N^2 wavelet coefficients. Given that many of them are pretty small, rather than using all of them, it is preferable to "truncate" them by discarding those coefficients having small magnitude. This is essentially a form of subset feature selection. The motivation is keeping as much information as possible while rejecting coefficients that are likely to encode fine details or noise that might not be essential for vehicle detection. Fig. 8 (second row) shows examples of reconstructed vehicle images using only the 50 largest coefficients. It should be clear from Fig. 8 that these coefficients convey important shape information, a very important feature for vehicle detection, while unimportant details have been removed.

We go one step further here by quantizing the truncated coefficients based on an observation—the actual values of the wavelet coefficients might not be very important since we are interested in the general shape of vehicles only. In fact, the magnitudes indicate local oriented intensity differences, information that could be very different even for the same vehicle under different lighting conditions. Therefore, the actual coefficient values might be less important or less reliable compared to the simple presence or absence of those coefficients. Similar observations have been made in [32] in the context of an image retrieval application. We use three quantization levels: -1, 0, and $+1$ (i.e., -1 representing large negative coefficients, $+1$ representing large positive coefficients, and 0 representing everything else). The images in the third row of Fig. 8 illustrate the quantized wavelet coefficients of the vehicle images shown in the first row. For comparison purposes, the last row of Fig. 8 shows the quantized wavelet coefficients of the nonvehicle images shown in the fourth row.

5) Combined Wavelet and Gabor Features: Careful examination of our results using wavelet or Gabor features revealed that the detection methods based on these two types of features yield different misclassifications. This observation suggests that wavelet features and Gabor features offer complementary information about the pattern to be classified, which could be used to improve the overall detection performance. This led us to the idea of combining the wavelet and Gabor features for improving performance.

As in Section IV-A3, we use the wavelet decomposition coefficients as our features directly. Performing the wavelet transform on the 32×32 images and throwing out the coefficients in the HH subband of the first level, yields a vector of 768 features. A filter bank consisting of four scales and six orientations is used here as it has demonstrated better performance (see Section VI-B). The combined feature set contains 1416 features. Since the values of Gabor and wavelet features are within different ranges, we normalize them in the range $[-1\ 1]$ before combining them in a single vector.

B. Classifiers

1) Multilyer Feed Forward NN: NNs have been very popular over the last 15 years, therefore, we provide only a brief review. In general, NNs implement a nonlinear mapping of the from $u = G(x)$. The mapping function G is established during a training phase where the network learns to correctly associate input patterns x to output patterns u (i.e., supervised learning). During training, their free parameters (i.e., weights and biases) are adjusted in a systematic way so as to minimize a cost function. Typically, the cost function is defined on the basis of the mean-square error between a desired network response (i.e., u) and the actual network output. In the context of classification, NNs can learn highly nonlinear decision boundaries, without explicitly estimating the probability distribution of the data.

There have been a number of important theoretical results illustrating the powerful computational capabilities of two-layer feed-forward NNs. Specifically, it has been shown that a single hidden layer feed-forward network with arbitrary sigmoid hidden layer activation functions can approximate arbitrarily well an arbitrary mapping from one finite-dimensional space to another [33]. This means that feed-forward networks can approximate virtually any function of interest to any desired degree of accuracy, provided sufficiently many hidden units are available.

In practice, determining the number of hidden nodes is not straightforward and can affect generalization performance [34]. Unlike the number of input and output nodes, which are determined by the dimensionality of the input vectors and the number of classes, the number of hidden nodes is not simply related to any obvious properties of the data. Although several methods have been proposed for choosing a proper number of hidden nodes, reaching the optimal point is still an open problem [35]. Other issues affecting NN performance include the choice of the activation functions, initial weights, and learning rates. In this paper, we consider a two-layer feed-forward NN with sigmoidal activation functions, trained by back-propagation, a popular learning algorithm that uses gradient descent to adjust the network weights and biases [35].

2) SVMs: SVMs are primarily two-class classifiers that have been shown to be an attractive and more systematic approach to learning linear or nonlinear decision boundaries [36], [37]. Their key characteristic is their mathematical tractability and geometric interpretation. This has facilitated a rapid growth of interest in SVMs over the last few years. SVMs have demonstrated remarkable success in fields as diverse as text categorization, bioinformatics, and computer vision [38].

Given a set of points, which belong to either of two classes, *SVM* finds the hyper-plane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyper-plane. This is equivalent to performing structural risk minimization to achieve good generalization [36], [37]. Assuming l examples from two classes

$$(x_1, y_1)(x_2, y_2)\dots(x_l, y_l),\ x_i \in R^N, y_i \in \{-1, +1\} \quad (15)$$

finding the optimal hyper-plane implies solving a constrained optimization problem using quadratic programming. The optimization criterion is the width of the margin between the classes. The discriminate hyper-plane is defined as

$$f(x) = \sum_{i=1}^{l} y_i a_i k(x, x_i) + b \quad (16)$$

where $k(x, x_i)$ is a kernel function and the sign of $f(x)$ indicates the membership of x. Constructing the optimal hyper-plane is equivalent to finding all the nonzero a_i. Any data point x_i corresponding to a nonzero a_i is a support vector of the optimal hyper-plane.

Suitable kernel functions can be expressed as a dot product in some space and satisfy the Mercer's condition [36]. By using different kernels, *SVMs* implement a variety of learning machines (e.g., a sigmoidal kernel corresponding to a two-layer sigmoidal neural network while a Gaussian kernel corresponding to a radial basis function (*RBF*) neural network). The Gaussian radial basis kernel is given by

$$k(x, x_i) = \exp\left(-\frac{\| x - x_i \|^2}{2\delta^2}\right). \quad (17)$$

The Gaussian kernel is used in this study (i.e., our experiments have shown that the Gaussian kernel outperforms other kernels in the context of our application).

3) Empirical Data Modelling: NN Versus SVM: Pattern classification problems root in the old topic—empirical data modelling. In empirical data modelling, a process of induction is used to build up a model of the system. The ultimate goal is to deduce the response of the system on unseen data. In general, system performance depends on the quantity and quality of the data used to build the model. In practice, the data obtained is finite and nonuniformly sampled. If the problem is high dimensional, then the sampled data will only form a sparse distribution in the input space which will yield the problem ill-defined.

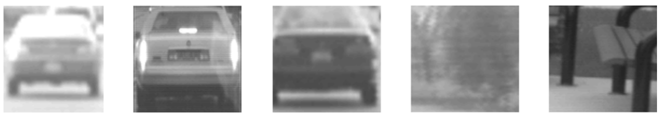


Fig. 9. Subimages for training.

Traditional NN approaches have suffered difficulties with generalization. This is a consequence of the optimization algorithms used for parameter selection and the statistical measures used to select the "best" model [36], [37]. Conventional NNs employ the traditional empirical risk minimization (ERM) principle, where the error on the training data is minimized. Different from those methods, the formulation of SVMs embodies the structural risk minimization (SRM) principle, which has been shown to be superior to ERM principle. SRM minimizes an upper bound on the expected risk. It is this difference that equips SVM with greater ability to generalize nicely—the goal in statistical learning. Roughly speaking, for a given learning task, with a given finite amount of training data, the best generalization performance will be achieved if the right balance is achieved between the accuracy attained on that particular training set, and the capacity of the machine, that is, the ability of the machine to learn any training set without error.

A key characteristic of SVMs is their mathematical tractability and geometric interpretation. In contrast, although there is a body of solid mathematical results today about NNs, many of them are asymptotic or based on assumptions. These results are hard to verify when applied to finite data sets which means that applying the NN theory in practice is not straightforward. SVMs contain only a small number of tunable parameters, while training a SVM (i.e., solving a convex quadratic programming problem) leads to solutions that are global and usually unique [39]. In contrast, NNs involve many more parameters and suffer from local minima.

V. Dataset

The images used for training were collected in two different sessions, one in Summer 2001 and one in Fall 2001, using Ford's proprietary low-light camera. To ensure a good variety of data in each session, the images were taken on different days and times, as well as on five different highways. The training sets contain subimages of rear vehicle views and nonvehicles which were extracted manually from the Fall 2001 data set. A total of 1051 vehicle subimages and 1051 nonvehicle subimages were extracted by several students in our lab. There is some variability in the way the subimages were extracted; for example, certain subimages cover the whole vehicle, others cover the vehicle partially, while some contain the vehicle and some background (see Fig. 9). In [25], the subimages were aligned by warping the bumpers to approximately the same position. We have not attempted to align the data in our case since alignment requires detecting certain features on the vehicle accurately. Moreover, we believe that some variability in the extraction of the subimages could actually improve performance. Each subimage in the training and test sets was scaled to 32×32 and preprocessed to account for different lighting conditions and contrast [40]. First, a linear function was fit to the intensity of the image. The result was subtracted out from the original image to correct for lighting differences.

To evaluate the performance of the proposed approach, the average error (*ER*), false positives (*FPs*), and false negatives (*FNs*), were recorded using a three-fold cross-validation procedure. Specifically, we split the training dataset randomly three times (*Set1*, *Set2*, and *Set3*) by keeping 80% of the vehicle subimages and 80% of the nonvehicle subimages (i.e., 841 vehicle subimages and 841 nonvehicle subimages) for training. The remaining 20% of the data was used for validation. For testing, we used a fixed set of 231 vehicle and nonvehicle subimages which were extracted from the Summer 2001 data set.

VI. Experimental Comparison of Various HV Approaches

In this section, we present experimental results of the HV approaches using the data set described in Section V and two classifiers: SVMs and NNs. Both of these classifiers require proper parameter values for good performance. Here, we evaluate each classifier by varying its parameters.

In the case of SVMs, we employ the Gaussian kernel, where σ is the only parameter for the kernel function. Our experiments indicate that SVM performance is not sensitive to σ, as long as it is within a certain range. We have found that performance differences are negligible when $0.005 < \sigma < 0.5$. In the following experiments, we have set $\sigma = 0.1$. Besides σ, another important parameter for SVMs is the regularization parameter C. A larger C corresponds to assigning a higher penalty to miss-classifications. We have experimented with C values in the range of 10 to 100 without noticing important performance differences. In the following experiments, we use $C = 10$.

In the case of NNs, we have experimented with one hidden layer, assuming different numbers of hidden units and different random initial weights. As we discussed in Section IV-B1, choosing the right number of hidden nodes is very important. If we choose a large number of hidden nodes, the training error can become very small, however, the NN would become tuned to the particular training set (i.e., overfitting). Consequently, the test error will be very high. On the other hand, if we use too few hidden nodes, the NN will not be able to fit the training data well, and again the test error will be high. In our experiments, we varied the number of hidden nodes (i.e., 15, 20, 25, 30, and 35) and used cross validation to terminate training. For each of these five different architectures, we trained the NN five times, using different random initial weights each time. We report the average performance on the test sets.

A. HV Using PCA Features

From our literature review in Section I, PCA features have been used quite extensively for vehicle detection. These features can be regarded as global features since changes in even

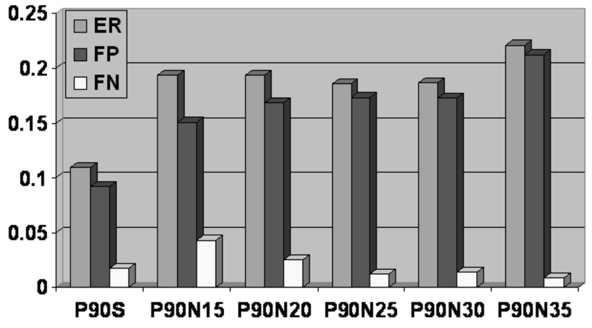


Fig. 10. HV using P90–PCA feature set preserving 90% information.

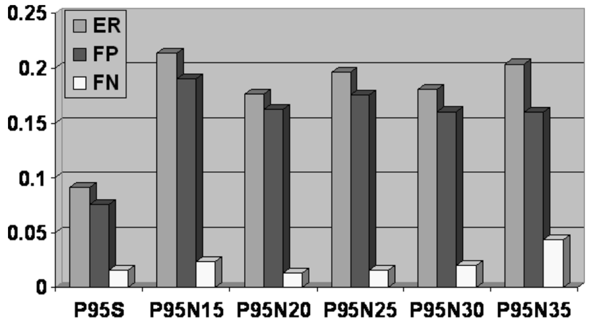


Fig. 11. HV using P95–PCA feature set preserving 95% information.

one pixel value of the image affect all the features. Two sets of *PCA* features have been used here, one preserving 90% information(*P90*) and one preserving 95% of the information (*P95*). For comparison purposes, we evaluated the performance of these feature sets using both *NNs* and *SVMs*. First, we used PCA features to train NN classifiers, referred to as *P90N* and *P95N*. Then, we tried the same *PCA* feature sets using SVMs (*P90S* and *P95S*).

Figs. 10 and 11 show the performances of the *PCA* feature sets in terms of error rate and FP/FN. When utilizing the feature set *P90*, the best performance was yielded by a NN with 25 hidden nodes *P90N25*—an average error rate of 18.6%, an average *FP* rate of 17.34% and an average *FN* rate of 1.26%. Slightly better than the feature set *P90*, the error rate, *FP* and *FN* using *P95* were 17.61%, 16.25%, and 1.36% with 20 hidden nodes. Compared to the *NN* classifier, the *SVM* classifier performed much better. *P95S* achieved an average error rate of 9.09%, which is almost 9% lower compared to NN's lowest error (i.e., *P95N20*). *P90S* achieved an error rate of 10.97%, which is almost 8% lower than *P90N25*. Obviously, *SVM* outperformed *NN* in this vehicle detection experiment using *PCA* features.

B. HV Using Gabor Features

In contrast to *PCA* features, Gabor features can be considered as local features. Two different Gabor feature sets were investigated in this paper. The first was extracted using a filter bank with 4 scales and 6 orientations [Fig. 7(b)], referred to as *G46*. The second one was extracted using a filter bank with three scales and five orientations (*G35*), illustrated in Fig. 7(a). First, we evaluated the performance of the two feature sets using *NNs*

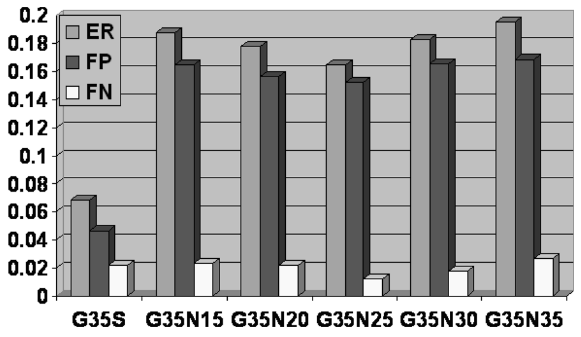


Fig. 12. HV using Gabor features *G35*.

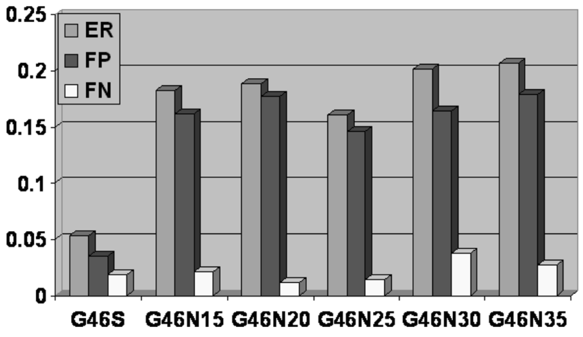


Fig. 13. HV using Gabor feature set *G46*.

with different architectures, as we did in the previous subsection. We refer to these those methods as *G46N15*, *G35N15* etc. Fig. 12 shows the performance using *G35*, while Fig. 13 shows the performance using *G46*. The best performance in the case of *G35* was achieved by a NN with 25 hidden nodes – an average error rate of 16.467%, an average FP rate of 15.23% and FN rate of 1.237%. Slightly better than *G35*, the lowest error rate using *G46N* was 16.04%, yielded by the NN with 25 hidden nodes. Then, we applied SVMs on these two feature sets. We refer to them as *G46S* and *G35S*. Figs. 12 and 13 illustrate that SVMs performed much better than NNs. In particular, the average error rate of *G46S* was 5.33%, the FP rate was 3.46% and FN rate was 1.88%. The error rates, FP and FN for *G35S* were 6.78%, 4.62%, and 2.16% correspondingly.

Fig. 14 shows some successful detection examples using *G46S*. The results illustrate several strong points of this method(*G46S*). Fig. 14(a) shows a case where only the general shape of the vehicle is available (i.e., no details) due to its distance from the camera. The method seems to discard irrelevant details, leading to improved robustness. In Fig. 14(b), the vehicle was detected successfully from its front view, although we did not use any front views in the training set. This demonstrates good generalization properties. Also, the method can tolerate some illumination changes as can be seen in Fig. 14(c)–(d).

C. HV Using Wavelet Features

Wavelet features can also be considered as local features. As described before, each of the images was scaled to 32×32 and then a five level *Haar* wavelet decomposition was performed on it, yielding 1024 coefficients. The final set contained 768 features after getting rid of the coefficients in the *HH* subband in the first level of the decomposition. We refer to this feature set as *W*. Experimental results are graphicly shown in Fig. 15.

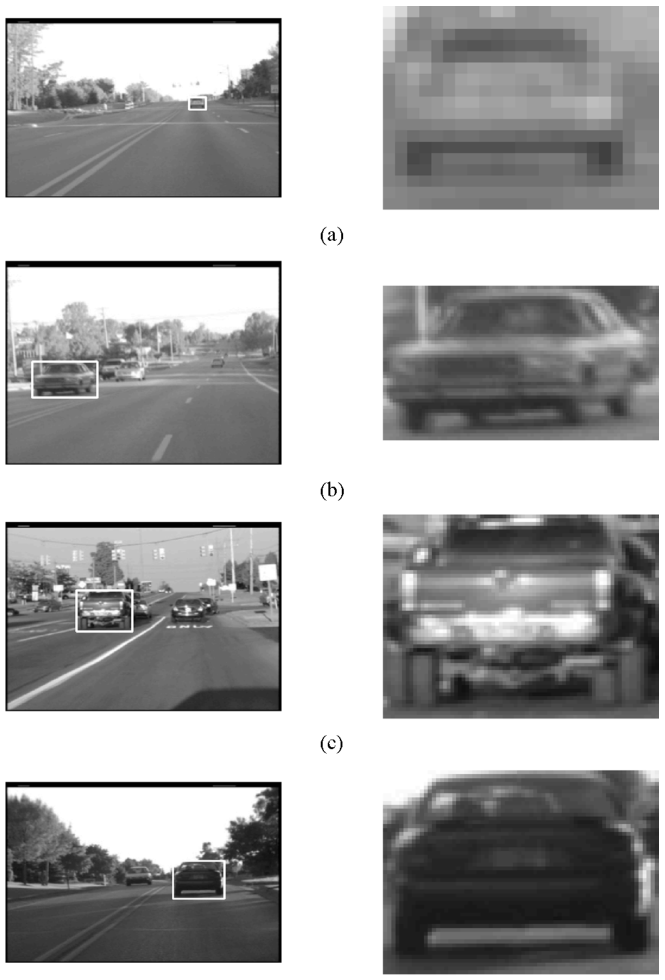


Fig. 14. Some examples of successful detection using Haar wavelet features (or Gabor features).

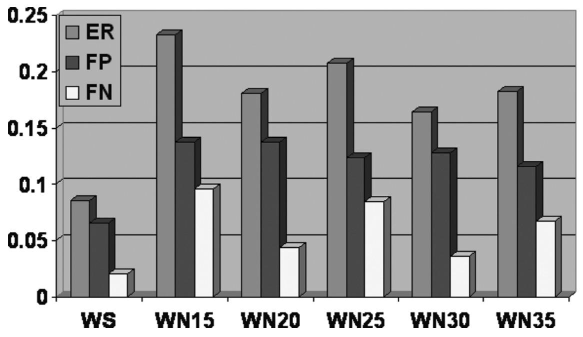


Fig. 15. HV using wavelet features.

Using *SVMs*, the average error rate was 8.52%, the average FP rate was 6.50%, and the average FN rate was 2.02%. Next we evaluated the performance of wavelet features using *NNs*, referred to as *WN* (see Fig. 15). Following the same evaluation methodology as before, the lowest error rate was 16.4% (FP 12.81% and FN 3.59% and was achieved by a NN with 30 hidden nodes. Similar to the observations made previously, *SVMs* performed better than *NN* using wavelet features).

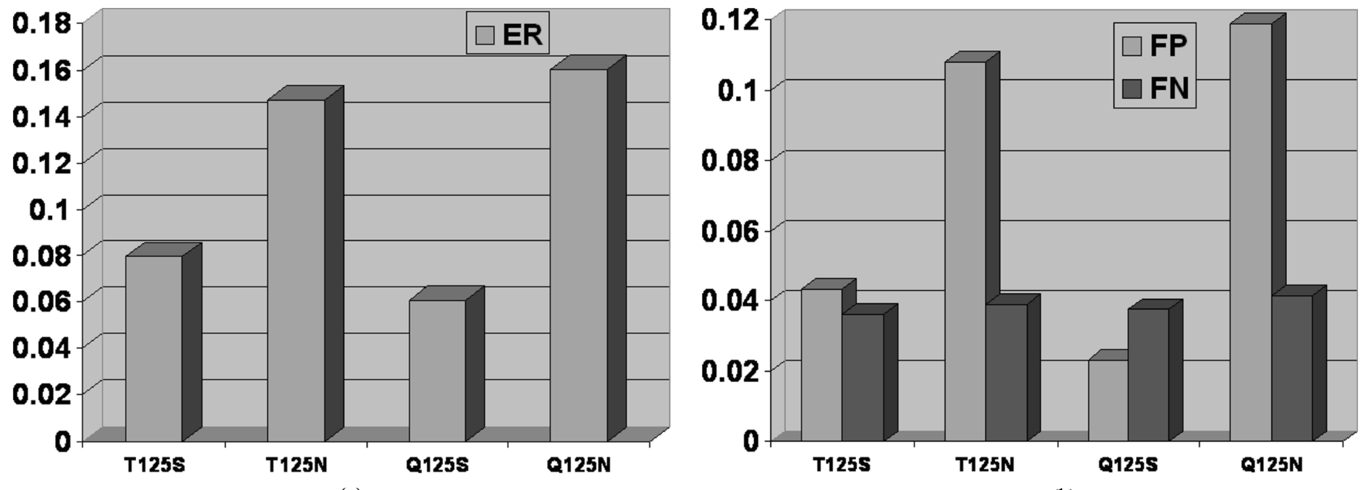


Fig. 16. HV using quantized/truncated wavelet features (a) error rate and (b) FPs and FNs.

Fig. 14 shows some successful detection examples using wavelet features and *SVMs* (i.e., same examples presented earlier to demonstrate Gabor features). Wavelet features seem to have similar properties to Gabor features – modelling general shape information [Fig. 14(a)], having good generalization properties [Fig. 14(b)], and demonstrating some degree of insensitivity to illumination changes [Fig. 14(c)–(d)].

D. HV Using Truncated Quantized Wavelet Features

The main argument for using the truncated quantized wavelet coefficients is that fine details of the training vehicle examples are not helpful. In order to eliminate the fine details, we truncated the wavelet coefficients by keeping only the ones having large magnitude. Using *SVMs*, we ran several experiments by keeping the largest 25, 50, 100, 125, 150, and 200 coefficients, setting the rest zero. The best results were obtained in the case of keeping 125 coefficients [see Fig. 17(a)–(c) for the performances]. Specifically, the average error rate of *T125S* was 7.94%, the average *FP* rate was 4.33%, and the average *FN* rate was 3.61%.

Then, we quantized the truncated coefficients to either "-1" or "$+1$" and trained *SVMs* using the quantized coefficients. We ran several experiments again by quantizing the largest 25, 50, 100, 125, 150, and 200 coefficients as described in Section IV-A4. Fig. 17(a)–(c) shows the error rate, FP, and FN rates obtained in this case. The best results were obtained again using 125 coefficients (see Fig. 16). The error rate obtained by *Q125S* was 6.06%, the average *FP* rate was 2.31%, and the average *FN* rate was 3.75%. As can be observed from Fig. 17(a), the *QSVM* approach demonstrated lower error than the *TSVM* approach in all cases. In terms of *FPs*, the performance of the *QSVM* approach was consistently better or equal to the performance of the *TSVM* approach when keeping 100 coefficients or more [see Fig. 17(b)]. In terms of *FNs*, the performance of the *QSVM* approach was consistently better or equal to that of the *TSVM* approach when keeping 25 coefficients or more [see Fig. 17(c)].

The superiority of the SVM classifier has been demonstrated over the PCA features (*P90* and *P95*), Gabor features (*G35* and *G46*), as well as the standard wavelet features (*W32*). Hereafter, only the best performance yielded by NN classifiers is reported. Overall, feature sets *Q125* and *T123* demonstrated the best performance using *SVMs*. For comparison purposes, we tested these two feature sets using *NNs*. The average error rate of *T125N* was 14.78%, while that of *Q125N* was 16.02%. Once again, *SVMs* yielded better performance.

E. HV Using Combined Wavelet and Gabor Features

A careful analysis of our results using wavelet and Gabor features revealed that, many times, the two approaches would make different classification errors. This observation motivated us to consider a simple feature fusion approach by simply combining wavelet features with Gabor features, referred to as *GW*. In particular, we chose Gabor features extracted using a filter bank with four scales and six orientations on 32×32 images, and the original wavelet features described in Section VI-C. Fig. 18(a) and (b) shows the results using the combined features. Using *SVMs*, the average error rate obtained in this case was 3.89%, the average *FP* rate was 2.29%, and the average *FN* rate was 1.6%. It should be reminded that the Gabor feature alone (i.e., *G46S*) yielded an error rate of 5.33%, while using wavelet features alone (i.e., *WS*) yielded an error rate of 8.52%. Using the combined feature set and *NNs* (*WGN*), the error rate achieved was 11.54%, which was lower than *G46N25* (i.e., 16.04%) or *WN30* (i.e., 16.4%).

Fig. 19 shows some examples that were classified correctly by the *GWS* approach, however, neither *G46S* nor *WS* were able to perform correct classification in all cases. Fig. 19(a), for example, shows a case that was classified correctly by the *G46S* approach but incorrectly by the *WS* approach. Fig. 19(c) shows another case which was classified incorrectly by the *G46S* but correctly by the *WS* approach. Neither *G46S* nor *WS* were able to classify correctly the case shown in Fig. 19(b). Obviously, feature fusion is a promising direction that requires further investigation.

F. Overall Evaluation

Several interesting observations can be made from analyzing the above experimental results. First, the local features con-

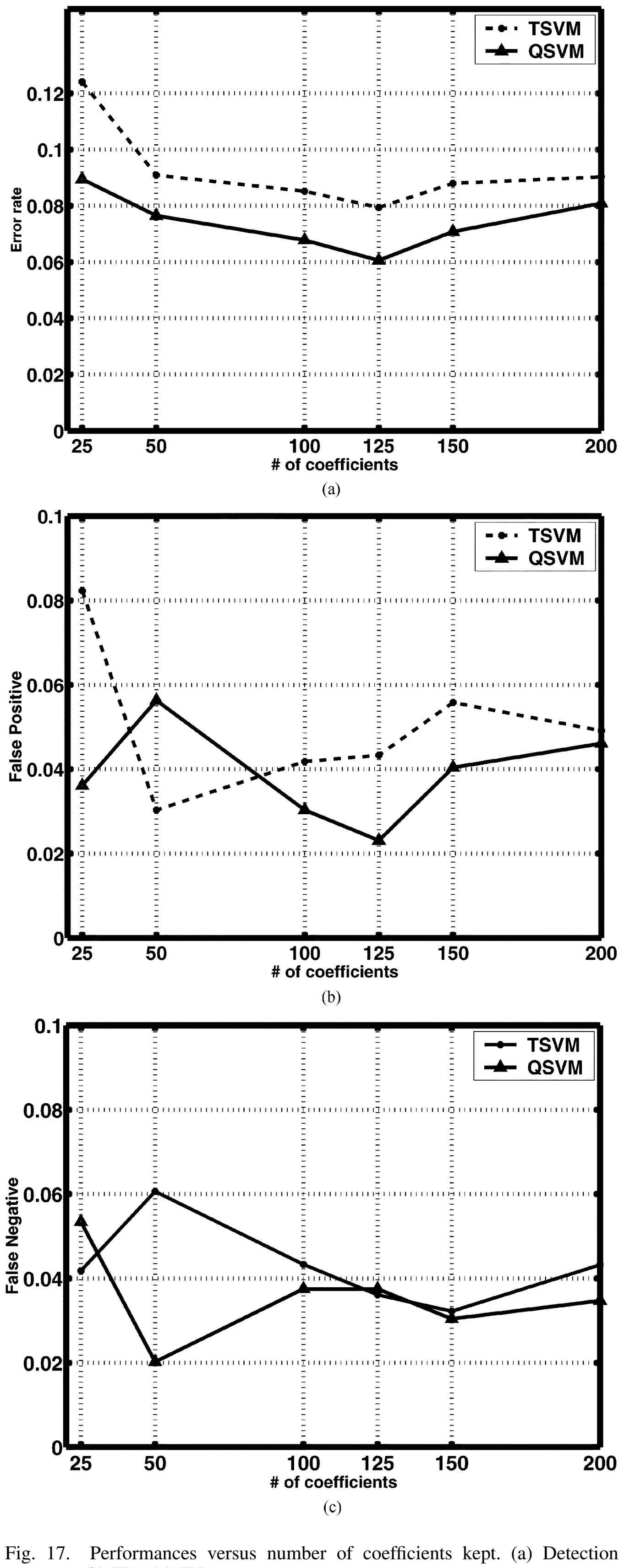


Fig. 17. Performances versus number of coefficients kept. (a) Detection accuracy. (b) FPs. (c) FNs.

sidered in this study (i.e., Gabor and wavelet features) outperformed the global ones (i.e., PCA features)—the lowest error rate using PCA features was 9.09%, (i.e, *P95S*), while the lowest error rate using wavelet features was 6.06% (i.e, *Q125S*), 5.33% using Gabor features (i.e., *G46S*), and 3.89% using the combined feature set (i.e., *WGS*). A possible reason for this is that the relative location of vehicles within the hypothesized windows is not fixed. Since we do not employ any normalization step prior to hypothesis verification, PCA features lack robustness. In contrast, local features, such as wavelet and Gabor features, can tolerate these "drifts" better.

Second, in the context of vehicle detection, *SVMs* yielded much better results than *NNs*. For instance, using the same PCA features, *SVMs* yielded an error rate of about 8% lower than *NNs*. Similar observations can be made using the other features. Due to the huge within-class variability, it is very difficult to obtain a perfect training data set for on-road vehicle detection. *SVMs* are capable of maximizing the generalization error on novel data by performing structural risk minimization, while *NN* can only minimize the empirical risk. This might be the main reason why *NNs* did not work as well as *SVMs*.

Third, the choice of features is an important issue. For example, using the same classifier (i.e., *SVMs*), the combined wavelet-Gabor feature set yielded an average error rate of 3.89%, while PCA features yielded an error rate of 9.09%. For vehicle detection, we would like features capturing general information of vehicle shape. Fine details are not preferred, for they might be present in specific vehicles only. The feature set should also be robust enough to cope with the uncertainty introduced by the HG step (i.e., "drift").

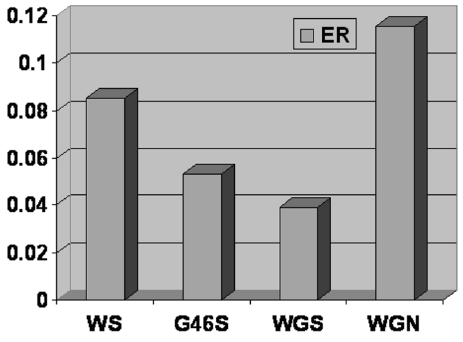
Fourth, feature selection is an area for further exploration. The quantized wavelet features yielded an average error rate of 6.06%, while the original wavelet features yielded an error rate of 8.52%. By varying the number of coefficients kept (i.e., some form of subset feature selection), truncated/quantized feature based methods demonstrated different performances. This implies that by ignoring or paying less attention to certain features, better performance can be obtained. However, the issue of selecting an optimum subset of features is still an open problem. We are currently investigating the problem of feature selection using genetic algorithms [41], [42].

Fifth, feature fusion can help improve detection. By simply concatenating the wavelet and Gabor features together, the detection error rate went down to 3.89% from 5.33% using Gabor features and 8.52% using wavelet features. Obviously, feature fusion is a subject that requires further investigation.

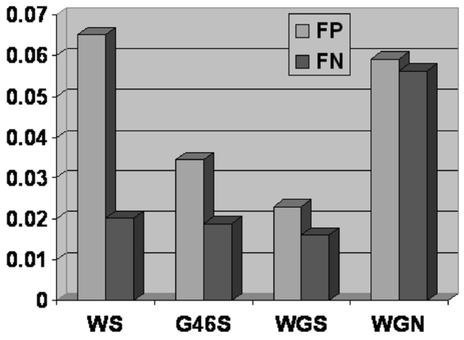
In terms of accuracy, the combined wavelet and Gabor features yielded the best results. Limited by real-time constraints, however, it is not realistic to use the *WGS* approach because of higher computational requirements (i.e., requires computing both wavelet and Gabor features). The performance of *G46S* (i.e., using Gabor feature only) was slightly worse than the *WGS*. Thus, our real-time system was based on the *G46S* approach.

Standard deviations can provide a measure of confidence. In our experiments, the standard deviations for accuracy obtained based on our cross-validation strategy were in the range of 10^{-2} to 10^{-3}. NNs were more unstable, especially when considering different number of layers and nodes per layer.

In terms of time requirements, we were not able to draw any useful conclusions. There are many different factors that affect the speed of a certain combination including the number

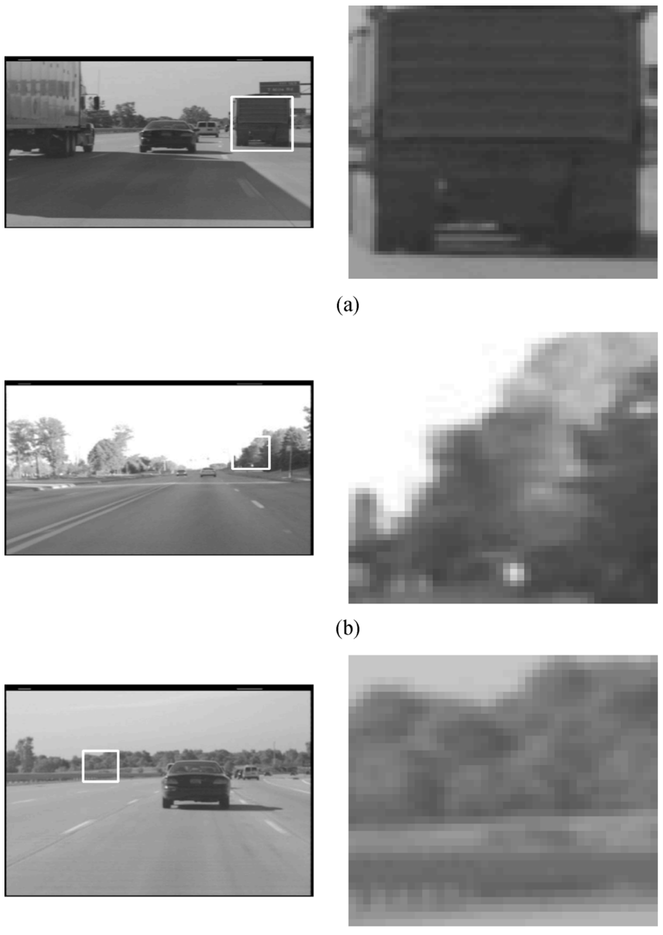


(a)



(b)

Fig. 18. HV using combined wavelet and Gabor features. (a) Error rate. (b) FPs and FNs.

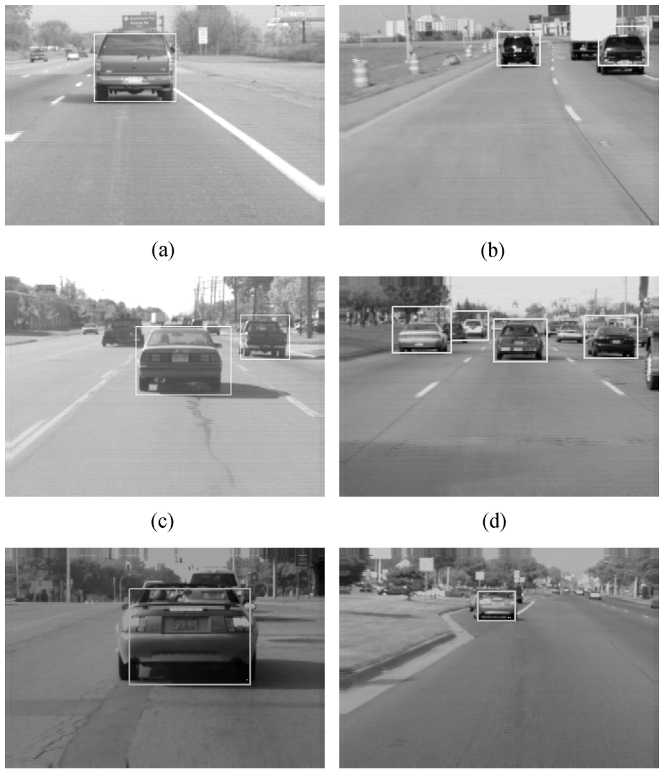


(a)

(b)

(c)

Fig. 19. Cases where either the *G46S* approach or the *WS* approach had failed to perform correct classification (all cases were classified correctly by the *GWS* approach).



(a) (b)

(c) (d)

(e) (f)

Fig. 20. Vehicle detection examples in rather simple scenes.

of layers and nodes in the NN, the number of support vectors in the SVM, the number of scales and orientations in the Gabor filters, and the number of levels in the wavelet transform. Generally speaking, we did not notice significant time differences for each of the combinations considered in this study.

VII. Real-Time System

In order to evaluate the performance of the two-step vehicle detection system, tests were carried out under different driving conditions. Figs. 20 and 21 show some representative detection results. The bounding boxes superimposed on the original images indicate the final detections. All the results shown in this section were generated by driving Ford's concept car around in the Detroit area. Fig. 20 shows some detection results assuming rather simple scenes like national highways. This is the easiest traffic scenario for any vision-based on-road vehicle detection system. Our system worked very well under this scenario. Detection under an urban area is much more difficult because vehicles are closer to each other, while buildings or trees might cast shadows both on the road and the vehicles. Fig. 21(a)–(f) shows some detection results under this scenario, where our system worked quite satisfactory. The performance of the system degraded when we drove the prototype vehicle under some abnormal conditions, such as, rain, little contrast between cars and

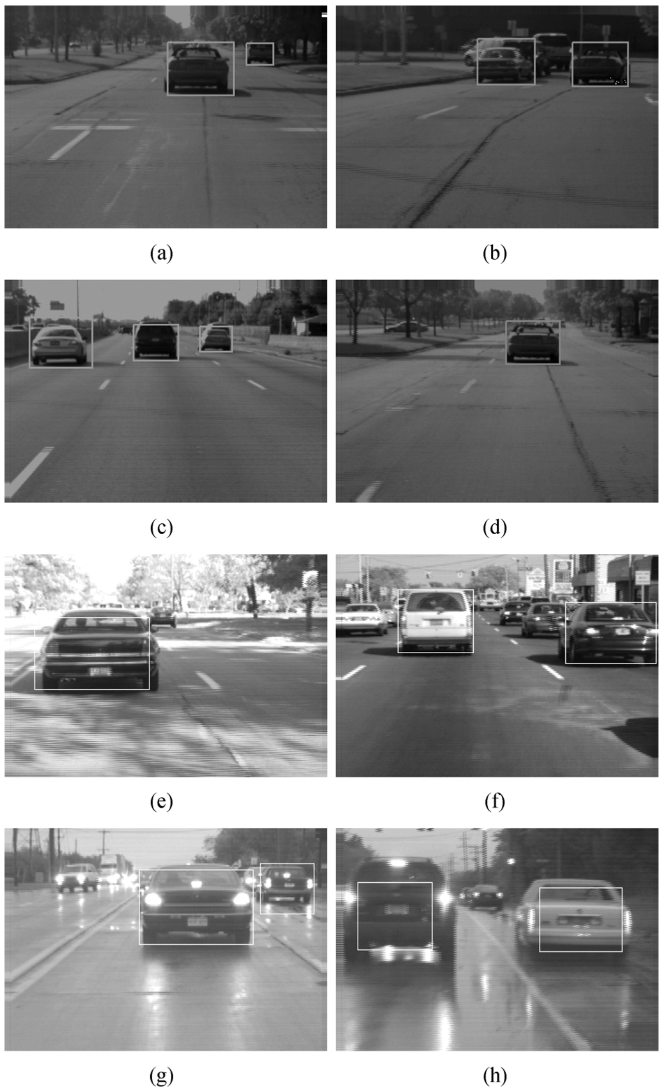


Fig. 21. Vehicle detection examples in complex scenes.

background, heavy congested traffic, etc. Fig. 21(g)–(h) shows two successful examples under this scenario.

The system demonstrated some tolerance to detecting vehicles from slightly nonrear views [e.g., see Fig. 21(b)], however, we have not analyzed the sensitivity of our system with respect to this factor since the objective of this study was different. There are several ways to improve the sensitivity of our method to small angle changes, for example, by training the classifier on data from slightly different angles or building a two-stage classification scheme in the spirit of [43]. In this approach, the second stage contains classifiers trained on different aspects while the first stage contains a classifier trained to assign a given input to a particular aspect.

We have achieved a detection rate of approximately 10 fps (NTSC: processing on the average every third frame) using a standard PC machine (Pentium III 1133 MHz), without making particular efforts to optimize our code. This is an average performance since some times images can be processed much faster than others (i.e., when there is only one vehicle present). It should be mentioned that vehicle detection for precrash sensing requires three-dimensional (3-D) interpretation (i.e., 3-D distance) for precise precrash actions. In general, 3-D information can be obtained using a stereo camera or an active sensor (e.g., radar or lidar). Alternatively, given the intrinsic camera parameters and assuming that the road is flat and vehicle size is approximately known (e.g., through classifying vehicles in different classes such as sedan, truck, bus, etc.) 3-D information can be inferred from a single camera [44].

Vehicle detection for precrash sensing requires, under certain circumstances, a higher sampling rate in order to provide a satisfactory solution. Our solution, presently, has a 10-Hz sampling rate. If the vehicle's speed is about 70 mph, 10-Hz corresponds to a 3-m interval. For many situations, this level of resolution is sufficient. We are currently working to increase the temporal resolution to 20 Hz, enabling side-impact collision avoidance and mitigation.

VIII. Conclusion and Future Work

Robust and reliable vehicle detection in images acquired by a moving vehicle is an important problem with applications to driver assistance systems or autonomous, self-guided vehicles. On-road vehicle detection is essentially a two-class pattern classification problem (i.e., vehicle versus nonvehicle). The focus of this paper is on feature extraction and classification for vehicle detection. We have investigated six different feature extraction methods (i.e., PCA features, wavelet features, truncated/quantized wavelet features, Gabor features, and combined wavelet and Gabor features) in the context of vehicle detection. For evaluation purposes, we considered two popular classifiers: *NNs* and *SVMs*.

A real-time monocular precrash vehicle detection system using Ford's proprietary low-light camera has been developed based on our evaluations. The vehicle detection algorithm includes two main steps: a multiscale driven hypothesis generation step and an appearance-based hypothesis verification step. The multiscale driven hypothesis generation step forms possible hypotheses at a coarse level of detail first. Then, it traces them down to the finer resolution. This scheme not only provides robustness but also speeds-up the whole process. The hypothesis verification is based on vehicle appearance. Specifically, we used statistical Gabor features extracted using a filter bank with four scales and six orientations, and SVMs (*G46S*).

We have evaluated the system using Ford's concept vehicle under different traffic scenario: simple scenes, complex urban scenes, and scenes assuming varying weather conditions. Our system worked very well on structured highways, provided good results in urban streets under normal conditions, and degraded gracefully under some adverse conditions, such as inclement weather and heavy congested traffic.

For future work, we plan to enhance the proposed vehicle detection scheme by exploiting temporal continuity. This can be achieved by employing a tracking mechanism to hypothesize the location of vehicles in future frames [19]. Tracking takes advantage of the fact that it is very unlikely for a vehicle to show up only in one frame. Therefore, vehicle location can be hypothesized using past history and a prediction mechanism. When tracking performance drops, the proposed hypothesis generation scheme can be deployed to maintain performance levels.

The majority of existing on-road vehicle detection and tracking systems use a *detect-then-track* approach (i.e., vehicles are first detected and then turned over to the tracker). This approach aims to resolve detection and tracking sequentially and separately. We envision a different strategy (i.e., *detect-and-track*), where detection and tracking are addressed simultaneously in a unified framework (i.e., detection results trigger tracking, and tracking re-enforces detection by accumulating temporal information through some probabilistic models). Approaches following this framework would have better chances to filter out false detections in subsequent frames. In addition, tracking template updates would be achieved through repeated detection verifications.

It should be mentioned that the number of negative examples used for training in this study was equal to the number of positive examples. This might not sound quite reasonable given that the nonvehicle class is much larger. Our motivation to use a relatively small number of negative examples was in order to keep training time within reasonable limits. However, although the set of negative examples used in this study was smaller than typically, the examples were of much better quality. This is because all negative examples were extracted in a controlled manner from a set of realistic images. Specifically, the negative examples in this study were extracted manually by a group of students using images of traffic scenes. This is in contrast to other studies where, although they used much larger sets of negative examples, the negative examples were selected randomly from a set of random images (e.g., in [45], the nonface examples were selected randomly from a set of scenery images). This might yield quite redundant (i.e., similar examples repeated over and over again) and/or irrelevant examples (i.e., might not have a great effect on determining the final solution).

We plan to improve classification performance using a "bootstrapping" strategy like in [45]. Bootstrapping allows to choose a more representative set of training examples, especially when the nontarget class (i.e., nonvehicles) has huge variability. Moreover, we plan to select the training examples based on the outputs of the hypothesis generation step (i.e., hypotheses classified incorrectly) instead of choosing them manually.

REFERENCES

[1] W. Jones, "Keeping cars from crashing," *IEEE Spectrum*, vol. 38, no. 9, pp. 40–45, Sep. 2001.

[2] ——, "Building safer cars," *IEEE Spectrum*, vol. 39, no. 1, pp. 82–85, Jan. 2002.

[3] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[4] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artifical vision in road vehicles," *Proc. IEEE*, vol. 90, no. 7, pp. 1258–1271, Jul. 2002.

[5] E. Dickmanns, "The development of machine vision for road vehicles in the last decade," presented at the IEEE Intelligent Vehicle Symp., 2002.

[6] A. Kuehnle, "Symmetry-based recognition for vehicle rears," *Pattern Recognit. Lett.*, vol. 12, pp. 249–258, 1991.

[7] S. D. Buluswar and B. A. Draper, "Color machine vision for autonomous vehicles," *Int. J. Eng. Appl. Artif. Intell.*, vol. 1, no. 2, pp. 245–256, 1998.

[8] E. Dickmanns *et al.*, "The seeing passenger car 'vamors-p'," in *Proc. Int. Symp. Intelligent Vehicles*, 1994, pp. 24–26.

[9] C. Goerick, N. Detlev, and M. Werner, "Artificial neural networks in real-time car detection and tracking applications," *Pattern Recognit. Lett.*, vol. 17, pp. 335–343, 1996.

[10] T. Kalinke, C. Tzomakas, and W. v. Seelen, "A texture-based object detection and an adaptive model-based classification," in *Proc. IEEE Int. Conf. Intelligent Vehicles*, 1998, pp. 143–148.

[11] H. Mallot, H. Bulthoff, J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biol. Cybern.*, vol. 64, no. 3, pp. 177–185, 1991.

[12] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Process.*, vol. 7, no. 1, pp. 62–81, Jan. 1998.

[13] G. Zhao and Y. Shini'chi, "Obstacle detection by vision system for autonomous vehicle," in *Proc. IEEE Intelligent Vehicle Symp.*, 1993, pp. 31–36.

[14] C. Knoeppel, A. Schanz, and B. Michaelis, "Robust vehicle detection at large distance using low resolution cameras," in *Proc. IEEE Intelligent Vehicle Symp.*, 2000, pp. 267–172.

[15] M. Suwa, "A stereo-based vehicle detection method under windy conditions," in *Proc. IEEE Intelligent Vehicle Symp.*, 2000, pp. 246–249.

[16] B. Heisele and W. Ritter, "Obstacle detection based on color blob flow," in *Proc. IEEE Intelligent Vehicles Symp.*, 1995, pp. 282–286.

[17] R. Okada, Y. Taniguchi, K. Furukawa, and K. Onoguchi, "Obstacle detection using projective invariant and vanishing lines," presented at the IEEE Int. Conf. Computer Vision, 2003.

[18] D. Koller, N. Heinze, and H. Nagel, "Algorithm characterization of vehicle trajectories from image sequences by motion verbs," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1991, pp. 90–95.

[19] M. Betke, E. Haritaglu, and L. Davis, "Multiple vehicle detection and tracking in hard real time," in *Proc. IEEE Intelligent Vehicles Symp.*, 1996, pp. 351–356.

[20] J. Ferryman, A. Worrall, G. Sullivan, and K. Baker, "A generic deformable model for vehicle recognition," in *Proc. Brit. Machine Vision Conf.*, 1995, pp. 127–136.

[21] N. Matthews, P. An, D. Charnley, and C. Harris, "Vehicle detection and recognition in greyscale imagery," *Control Eng. Practice*, vol. 4, pp. 473–479, 1996.

[22] H. Schneiderman and T. Kanade, "Probabilistic modeling of local appearance and spatial relationships for object recognition," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 1998, pp. 45–51.

[23] H. Schneiderman, "A Statistical Approach to 3D Object Detection Applied to Faces and Cars," CMU-RI-TR-00-06, 2000.

[24] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," in *Proc. Eur. Conf. Comptuer Vision*, 2000, pp. 18–32.

[25] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *Int. J. Comput. Vis.*, vol. 38, no. 1, pp. 15–33, 2000.

[26] E. Dickmanns, B. Mysliwetz, and T. Christians, "n inte-grated spatio-temporal approach to automatic visual guidance of autonomous vehicles," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 6, pp. 1273–1284, Nov./Dec. 1988.

[27] F. Thomanek, E. Dickmanns, and D. Dickmanns, "Multiple object recognition and scene interpretation for autonomous road vehicle guidance," in *Proc. IEEE Intelligent Vehicles Symp.*, 1994, pp. 231–236.

[28] A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[29] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, pp. 71–86, 1991.

[30] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996.

[31] S. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on Gabor filters," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1160–1167, Oct. 2002.

[32] C. Jacobs, A. Finkelstein, and D. Salesin, "Fast multiresolution image querying," in *Proc. SIGGRAPH*, 1995, pp. 277–286.

[33] K. Hornik, M. Stinchombe, and H. White, "Multilayer feed-forward networks are universal approximators," *Neural Netw.*, vol. 2, no. 3, pp. 359–366, 1989.

[34] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks: Why network size is so important," *IEEE Potentials*, no. 4, pp. 27–31, Oct./Nov. 1994.

[35] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York: Wiley, 2001.

[36] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[37] C. Burges, "Tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Disc.*, vol. 2, no. 2, pp. 955–974, 1998.

[38] N. Cristianini, C. Campbell, and C. Burges, "Kernel methods: Current research and future directions," *Mach. Learn.*, vol. 46, pp. 5–9, 2002.

[39] C. Burges and D. Crisp, "Uniqueness of the SVM solution," *NIPS*, vol. 12, 2000.

[40] G. Bebis, S. Uthiram, and M. Georgiopoulos, "Face detection and verification using genetic search," *Int. J. Artif. Intell. Tools*, vol. 9, no. 2, pp. 225–246, 2000.

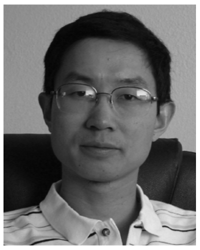
[41] Z. Sun, G. Bebis, and R. Miller, "Boosting object detection using feature selection," presented at the IEEE Int. Conf. Advanced Video and Signal Based Surveillance, 2003.

[42] ——, "Evolutionary Gabor filter optimization with application to vehicle detection," presented at the IEEE Int. Conf. Data Mining, 2003.

[43] H. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," presented at the IEEE Conf. Computer Vision and Pattern Recognition, 1998.

[44] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miche, S. Mousset, and G. Moulminet, "A cooperative approach to vision-based vehicle detection," presented at the IEEE Int. Conf. Intelligent Transportation Systems, 2001, pp. 209–214.

[45] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23–38, Jan. 1998.



Zehang Sun received the B.Eng. degree in telecommunications and the M.Eng. degree in digital signal processing from Northern Jiaotong University, China, in 1994 and 1997, respectively, the M.Eng. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 1999, and the Ph.D. degree in computer science and engineering from the University of Nevada, Reno, in 2003.

He joined eTreppid Technologies, LLC, Reno, immediately upon his graduation. His expertise is in the area of real-time computer vision systems, statistical pattern recognition, artificial intelligence, digital signal processing, and embedded systems.

George Bebis received the B.S. degree in mathematics and the M.S. degree in computer science from the University of Crete, Crete, Greece, in 1987 and 1991, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Central Florida, Orlando, in 1996.

Currently, he is an Associate Professor with the Department of Computer Science and Engineering at the University of Nevada, Reno, and Director of the UNR Computer Vision Laboratory. His research interests include computer vision, image processing, pattern recognition, machine learning, and evolutionary computing. His research is currently funded by the National Science Foundation, the National Aeronautics and Space Administration, the Office of Naval Research, the and Ford Motor Company.

Dr. Bebis is a member of the IAPR Educational Committee. He has served on the program committees of various national and international conferences and has organized and chaired several conference sessions. In 2002, he received the Lemelson Award for Innovation and Entrepreneurship. He is an Associate Editor of the *Machine Vision and Applications Journal* and serves on the Editorial Board of the *Pattern Recognition Journal* and the *International Journal on Artificial Intelligence Tools*.

Ronald Miller received the B.S. degree in physics from the University of Massachusetts, Amherst, in 1983, and the Ph.D. degree in physics from the Massachusetts Institute of Technology, Cambridge, in 1988.

His research has ranged from computational modeling of plasma and ionospheric instabilities to automotive safety applications. He heads a research program at Ford Motor Company, Dearborn, MI, in intelligent vehicle technologies focusing on advanced RF communication, radar, and optical-sensing systems for accident avoidance and telematics.