Monocular Pre-crash Vehicle Detection: Features and Classifiers

Zehang Sun[1], George Bebis[1] and Ronald Miller[2]
[1]Computer Vision Lab. Department of Computer Science, University of Nevada, Reno
[2]e-Technology Department, Ford Motor Company, Dearborn, MI
(zehang,bebis)@cs.unr.edu, rmille47@ford.com

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 (*PCA*), Wavelets, and Gabor filters. To evaluate the extracted features, we have experimented with two popular classifiers, Neural Networks(*NNs*) and Support Vector Machines(*SVMs*). Based our evaluation results, we have developed an on-board real-time monocular pre-crash 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 multi-scale 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 multi-scale techniques 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, varying weather conditions), illustrating good performance.**

Keywords— **Vehicle detection, Principal Component Analysis, Wavelets, Gabor filters, Neural Networks, Support Vector Machines.**

I. Introduction

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. The statistics is similar in European Union (42,500 death, 3.5 million injuries and $160 billion euro loss). The loss is too startling to be ignored. Recognizing that vehicle safety is a primary concern for many motorists, many national and international companies have lunched multi-year research projects to investigate new technologies for improving safety and accident prevention [1]. With the aim of reducing injury and accident severity, pre-crash sensing is becoming an area of active research among automotive manufacturers, suppliers and Universities. Vehicle accident statistics disclose that the main threats drivers are facing when driving a vehicle are from other vehicles. Consequently, on-board automotive driver assistance systems aiming to alert a driver about driving environments, possible collision with other vehicles, or take control of the vehicle to enable collision avoidance and mitigation, have attracted more and more attention lately. In these systems, robust and reliable vehicle detection is a critical first step.

The most common approach to vehicle detection is using active sensors such as lasers or millimiter-wave radars. Prototype vehicles employing active sensors have shown promising results, however, active sensors have several drawbacks such as low resolution, may interfere with each other, and are rather expensive. Passive sensors on the other hand, such as cameras, offer a more affordable solution and can be used to track, more effectively, cars entering a curve or moving from one side of the road to another. Moreover, visual information can be very important in a number or related applications such as lane detection, traffic sign recognition, or object identification (e.g., pedestrians, obstacles).

Vehicle detection using passive optical sensors involves several challenges. For example, vehicles may vary in shape, size, and color. The appearance of a specific vehicle depends on its pose and is affected by nearby objects. Complex outdoor environments(e.g., illumination conditions, unpredictable interaction between traffic participants, cluttered background) cannot be controlled. On-board moving cameras make some well established techniques, such as background subtraction, quite unsuitable. Furthermore, on-board vehicle detection systems have strict constraints on computational cost. They should be able to process acquired images in real-time or close to real-time, in order to save more time for driver's reaction.

The majority of vehicle detection algorithms in the literature consist of two basic steps: (1) Hypothesis Generation (HG) which hypothesizes the locations in images, where vehicles might be present, and (2) Hypothesis Verification(HV) which verifies the hypotheses. HG approaches can be classified into one of the following three categories: (1) knowledge-based, (2) stereo-based, and (3) motion-based. Knowledge-based methods employ knowledge about vehicle shape and color as well as general information about streets, roads, and freeways. Tzomakas et al. [2], [3] for example, have modelled the intensity of the road and shadows under the vehicles to estimate the possible presence of vehicles. Symmetry detection approaches using the intensity or edge map have also been exploited based on the observation that vehicles are symmetric about the vertical axis [4], [5].

Stereo-based approaches take advantage of the Inverse

Perspective Mapping (IMP) [6] to estimate the locations of vehicles and obstacles in images. Bertozzi et al. [7] 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 [8], the IPM was used to wrap the left image to the right image. Knoeppel et al. [9] developed a stereo-system detecting vehicles up to 150m. 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 [10].

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 [11] or local intensity minima and maxima [12].

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. [13] proposed a multiple-vehicle detection approach using deformable gray-scale template matching. In [14], 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 non-vehicle 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 non-vehicle classes is learned either by training a classifier (e.g., Neural Network (*NN*)) or by modelling the probability distribution of the features in each class (e.g., using the Bayes rule assuming Gaussian distributions). In [15], *PCA* was used for feature extraction and NNs for classification. Goerick et al. [16] used a method called Local Orientation Coding (*LOC*) to extract edge information. The histogram of *LOC* within the area of interest was then fed to a *NN* for classification. A statistical model for vehicle detection was investigated by Schneiderman et al. [17] [18]. A view-based approach based on multiple detectors was used to cope with viewpoint variations. The statistics of both object and "non-object" appearance were represented using the product of two histograms with each histogram representing the joint statistics of a subset of *PCA* features in [17] or Haar wavelet features in [18] and their position on the object. A different statistical model was investigated by Weber et al. [19]. 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. [20] have proposed using the Haar wavelet transform for feature extraction and Support Vector Machines (*SVMs*) for classification. Gabor features, quantized wavelet features, and fusion of Gabor and wavelet features have been explored in our previous studies [21], [22], [23].

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 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 of our study 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 framegrabber of a normal PC (see Fig.1). The PC is sitting inside the vehicle and is powered up by a converter in the car. Camera images are digitally captured and processed in nearly real-time enabling vehicle detection on timescales on the order of 10Hz. Our detection system consists of two steps: a multi-scale driven hypothesis generation step and an appearance-based hypothesis verification step. Multi-scale 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.

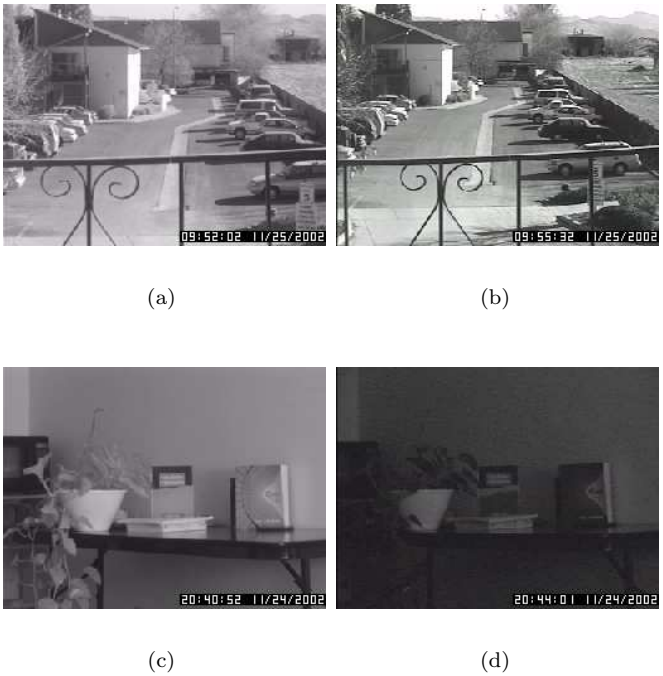
The rest of the paper is organized as follows: In Section II, we provide a brief overview of the system developed. A description of the multi-scale driven hypothesis generation step is given in Section III. Various features and classifiers are detailed in IV. Comparisons of various *HV* approaches are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

II. monocular pre-crash vehicle detection system overview

Pre-crash 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, 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 pre-crash 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 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 pre-crash 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.

Fig. 1. Low light camera in the prototype vehicle

The optical system represents a principal component in pre-crash 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 camera enabling a nearly 360 field of view. Fig. 1 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. 2.a and Fig. 2.c shows the dynamics range of the low light camera , while Fig. 2.b and Fig. 2.d show the same scene images caught under same illumination conditions by using a normal camera. Obviously, the low light camera provides much wider dynamic range.



(a) (b)

(c) (d)

Fig. 2. Low light camera v.s. normal camera. (a) Lowlight camera daytime image, (b) Same scene caught using normal camera, (c) Lowlight camera nighttime image, (d) Same nighttime scene caught suing normal camera

III. multi-scale 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 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. 3).

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 multi-scale

approach which combines sub-sampling 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 case:

$$f^{K-1} = REDUCE(f^K) \qquad (1)$$
$$f^{K-1}(x) = \Sigma_{n=-N}^{N} c(n) f^K(2x - n)$$

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

Fig. 3. Multi-scale 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; fourth column: vertical and horizontal profiles. Note that all images have been scaled back to 360×248 for illustration purposes.

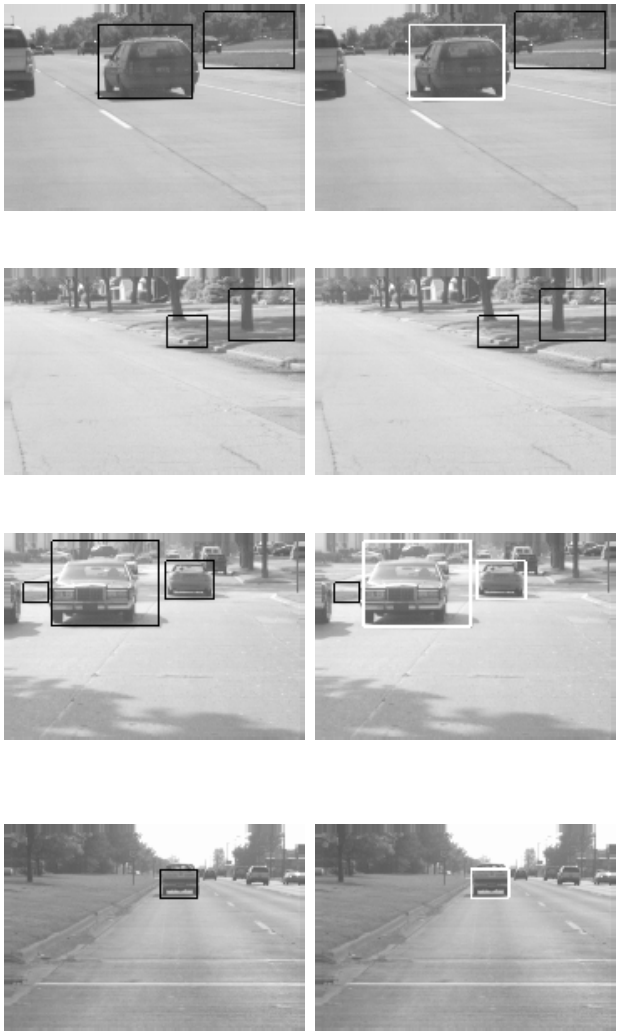


Fig. 4. Examples of the HG (left column)and HV (right column) steps: the black boxes indicate the hypothesized locations while the white boxes are the ones verified by the HV step.

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. 3) (2) vertical edge detection (e.g., second column of Fig. 3), vertical profile computation of the edge image (e.g., last column of Fig. 3), and profile filtering using a low pass filter, (3) horizontal edge detection (e.g., third column of Fig. 3), horizontal profile computation of the edge image (e.g., last column of Fig. 3), 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. 3). 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. 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 them, 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). These rules are applied at each level of detail.

The proposed multi-scale 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 multi-scale scheme speeds-up the whole process since the low resolution images have

much simpler structure as illustrated in Fig. 3 (i.e., candidate vehicle locations can be found faster and easier). Several examples are provided in Fig. 4 (left column).

IV. appearance-based hypothesis verification

Verifying a hypothesis is essentially a two-class pattern classification problem (i.e., vehicle versus non-vehicle). 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: (i) extracting a number of features and (ii) 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 *apriori*. 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 [24]. Depending on the nature of the task at hand, the features can be extracted either manually or automatically by applying transformations on hand-picked features or the original raw pixel values of the image (i.e., primitive features). The transformations used for feature extraction perform dimensionality reduction which could be either linear or non-linear. Transformation-based methods have the potential of generating better features than the original ones, however, the new features may not have a physical meaning since they are combinations of the original ones.

In this paper, we investigate five different feature extraction methods (linear/nonlinear, global/local). To evaluate the extracted features for vehicle detection, we present experiments using two powerful classifiers: NNs and SVMs.

A. Feature Extraction

A.1 PCA Features

Eigenspace representations of images use PCA [25] 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 [25]:

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, $A = [\Phi_1\Phi_2 \dots \Phi_R]$. 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:

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

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. In this paper, $\{w_i\}$ are our eigen-features.

The projection coefficients allow us to represent images as linear combinations of the eigenvectors. It is well known that the projection coefficients define a compact image representation and that a given image can be reconstructed from its projection coefficients and the eigenvectors (i.e., basis). The eigenspace representation of images has been used in various applications such as image compression and face recognition, as well as vehicle detection[14], [15], [18].

A.2 Wavelet Features

Wavelets are a essentially a multiresolution function approximation method that allow for the hierarchical decomposition of a signal or image. They have been applied successfully to various problems including object detection [20], [18], face recognition [26], image retrieval [27], and vehicle detection [18] [20]. 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:

$$\cdots V_{-1} \subset V_0 \subset V_1 \cdots \subset L^2 \tag{6}$$

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:

$$\phi_i^j(x) = 2^{\frac{j}{2}}\phi(2^j x - i). \quad i = 0, \cdots, 2^j - 1. \tag{7}$$

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 functions, 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:

$$\psi_i^j(x) = 2^{\frac{j}{2}}\psi(2^j x - i). \quad i = 0, \cdots, 2^j - 1. \tag{8}$$

Different scaling functions $\phi_i^j(x)$ and wavelets $\psi_i^j(x)$ determine various wavelet transforms. In this paper, we use 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 non-redundant representation of the input images. The Haar scaling function is:

$$\phi(x) \quad = \quad \begin{cases} 1 & for \;\; 0 \leq x < 1 \\ 0 & otherwise \end{cases} \tag{9}$$

And the Haar wavelet is defined as:

$$\psi(x) \quad = \quad \begin{cases} 1 & for \;\; 0 \leq x < \frac{1}{2} \\ -1 & for \;\; \frac{1}{2} \leq x < 1 \\ 0 & otherwise \end{cases} \tag{10}$$

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 [18], which is not helpful at all given we aim to model the general shape of the vehicle class.

A.3 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. 5 (2nd row) shows examples of reconstructed vehicle images using only the 50 largest coefficients. It should be clear from Fig. 5 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 [27] 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. 5 illustrate the quantized wavelet coefficients of the vehicle images shown in the first row. For comparison purposes, the last row of Fig. 5 shows the quantized wavelet coefficients of the non-vehicle images shown in the fourth row.

Fig. 5. 1st row: vehicle sub-images used for training; 2nd row: reconstructed sub-images using the 50 largest coefficients; 3rd row: illustration of the 50 quantized largest coefficients; 4th and 5th rows: similar results for some non-vehicle sub-images.

A.4 Gabor Features

Gabor features have been used successfully in image compression [28] texture analysis [29], [30] face recognition [31] and image retrieval [32]. We believe that these features are quite appropriate for our application. 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 Gabor filter family can be represented as a Gaussian function modulated by an oriented complex sinusoidal signal:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp[-\frac{1}{2}(\frac{\tilde{x}^2}{\sigma_x^2}+\frac{\tilde{y}^2}{\sigma_y^2})]\exp[2\pi jW\tilde{x}] \quad (11)$$

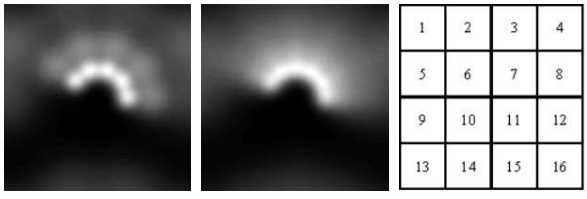
$$\tilde{x} = x\cos\theta + ysin\theta \; and \; \tilde{y} = -x\sin\theta + ycos\theta \quad (12)$$

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:

$$G(u,v) = exp\{-\frac{1}{2}[\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}]\} \quad (13)$$

Gabor filters act as local bandpass filters. Fig. (6.(a)) and (6.(b)) show the power spectra of two Gabor filter banks (the light areas indicate spatial frequencies and wave orientation).

In this paper, we use the design strategy described in [32]. 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 post-processing is usually applied (e.g., Gabor-energy features, thresholded Gabor features, and moments based on Gabor features [33]). We use Gabor features based on moments, extracted from several subwindows of the input image.



(a) (b) (c)

Fig. 6. (a) Gabor filter bank with 3 scales and 5 orientations; (b) Gabor filter bank with 4 scales and 6 orientations; (c) Feature extraction subwindows.

In particular, each hypothesized subimage is scaled to a fixed size of 32×32. Then, it is subdivided into 9 overlapping 16×16 subwindows. Assuming that each subimage consists of 16 8×8 patches (see Fig. 6.(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 forth. 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 i-th filter and j to the j-th subwindow). Using moments implies that only the statistical properties of a group 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 9 subwindows, yields a feature vector of size 162, having the following form:

$$[\mu_{11}\sigma_{11}\kappa_{11}, \mu_{12}\sigma_{12}\kappa_{12}, \cdots \mu_{69}\sigma_{69}\kappa_{69}] \quad (14)$$

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 to compute.

A.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-A.2, 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 HH subband of the first level, yields a vector of 768 features. A filter bank consisting of 4 scales and 6 orientations is used here as it has demonstrated better performance (see V-D). 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

B.1 Back-Propagation Neural Network

Various neural network models have been utilized in the vehicle detection literature. In our experiments, we used a two-layer perceptron NN with sigmoidal activation functions, trained by the back-propagation algorithm. Cybenko has shown that a two layer network (i.e., one hidden and one output layers) is sufficient to approximate any mapping to arbitrary precision, assuming enough hidden nodes [34].

Back-propagation neural networks can directly construct highly non-linear decision boundaries, without estimating the probability distribution of the data.

B.2 SVMs

SVMs are primarily two-class classifiers that have been shown to be an attractive and more systematic approach to learning linear or non-linear decision boundaries [35] [36]. 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 [35] [36]. Assuming l examples from two classes

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

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:

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

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 find 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 [35]. 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

$$k(x, x_i) = \exp(-\frac{\parallel x - x_i \parallel^2}{2\delta^2}) \tag{17}$$

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).

C. Dataset

The images used for training were collected in two different sessions, one in the Summer of 2001 and one in the Fall of 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 non-vehicles which were extracted manually from the Fall 2001 data set. A total of 1051 vehicle subimages and 1051 non-vehicle 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. 7). In [20], 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 [37]. 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 non-vehicle subimages (i.e., 841 vehicle subimages and 841 non-vehicle subimages) for training. The rest 20% of the data was used for validation. For testing, we used a fixed set of 231 vehicle and non-vehicle subimages which were extracted from the Summer 2001 data set.

Fig. 7. Subimages for training.

V. Experimental Comparison of Various HV Approaches

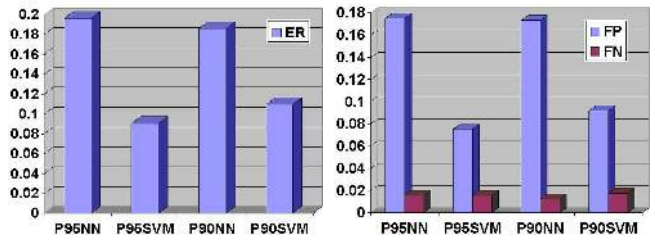
Experimental results of various HV approaches using the data set described in IV-C are carried out in this section.

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 the pixel values of the image affect all the features. Two sets of *PCA* features have been used here, one preserving 90% information(*P90N*) and one preserving 95% of the information (*P95N*). For comparison purposes, we evaluated the performance of these feature sets using both *NNs* and *SVMs*. First, we used PCA features to train a NN classifier, referred to as *P90NN* and *P95NN*. In order to obtain optimum performance, we varied the number of hidden nodes and used cross-validation to terminate training. Then, we tried the same *PCA* feature sets using SVMs (*P90SVM* and *P95SVM*).

Fig. 8 shows the performances of the *PCA* feature sets in terms of error rate and FP/FN. The *P95NN* approach achieved an average error rate of 18.98%, an average *FP* rate of 17.56% and an average *FN* rate of 1.42%. Slightly better than *P95NN*, the error rate, *FP* and *FN* using

P90NN were 18.19%, 17.32% and 0.87% respectively. Compared to the *NN* classifier, the *SVM* classifier performed much better. *P95SVM* achieved an average error rate of 9.09%, which is almost 10% lower than applying *NN* on the same feature set (*P95NN*). *P90SVM* achieved an error rate of 10.97%, which is 7% lower than *P90NN*. Obviously, *SVM* outperformed *NN* in this vehicle detection experiment using *PCA* features.

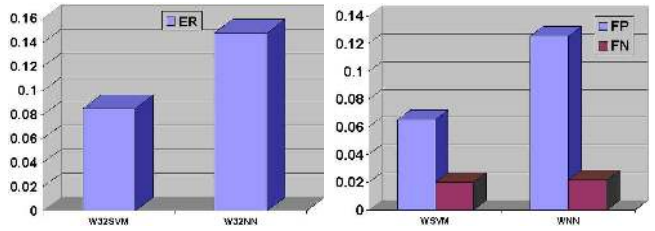


(a) (b)

Fig. 8. HV using PCA features(a) Error rate; (b) FP and FN.

B. HV using wavelet features

In contrast to *PCA* features, wavelet features can 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 *W32*. Experimental results are graphicly shown in Fig. 9. Using *SVMs*, the average error rate was 8.52%, the average

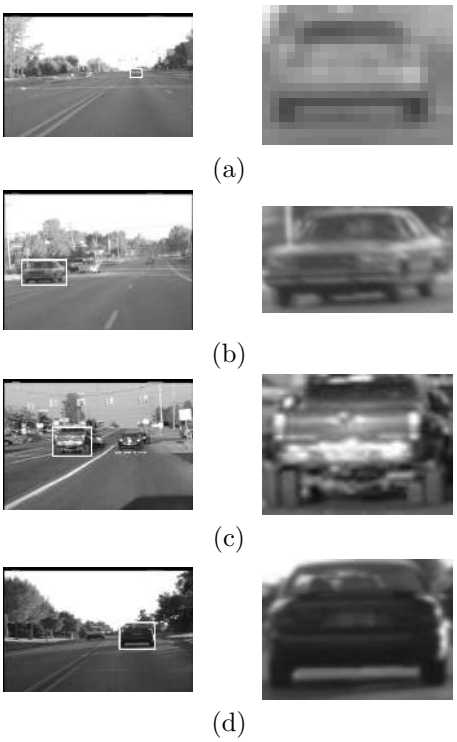


(a) (b)

Fig. 9. HV using original wavelet features (a) Error rate; (b) FP and FN.

FP rate was 6.50%, and the average FN rate was 2.02%. Next we evaluated the performance of wavelet features using *NN*, referred to as *W32NN*. The error rate, FP and FN of the *W32NN* approach were 14.81%, 12.55% and 2.16% correspondingly. Similarly to the observation made in Section V-A, *SVMs* performed better than *NN* using wavelet features.

Fig. 10 shows some successful detection examples using wavelet features and *SVMs*. The results illustrate several



(a) (b) (c) (d)

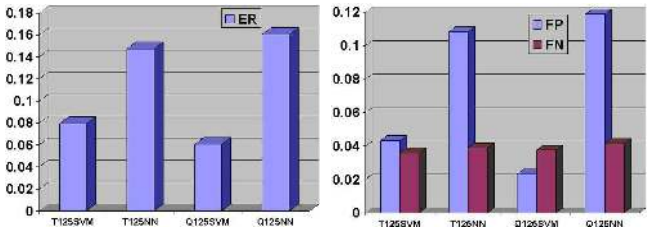
Fig. 10. Some examples of successful detection using Haar wavelet features.

strong points of this method(*W32SVM*). Fig. 10.(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. 10.(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 Figures 10.(c-d).

C. HV using Truncated Quantized Wavelet Features

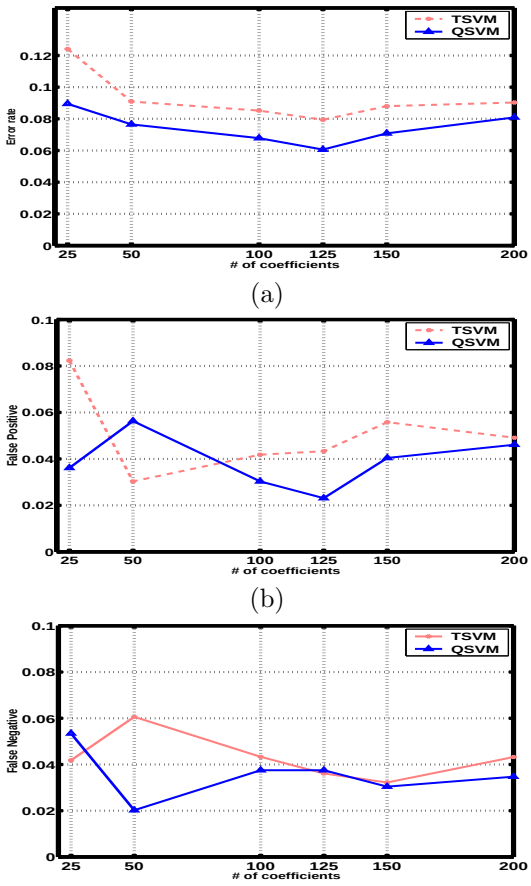
The main argument for using the truncated quantized wavelet coefficients is that fine details of the training vehicle examples are not helpful. In order to eliminate the fine details, we truncated the wavelet coefficients by keeping only the ones having large magnitude. Using *SVMs*, we ran several experiments 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. 12.(a-c) for the performances). Specifically, the average error rate 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-A.3. Fig. 12.(a-c) show the error rate, FP, and FN rates obtained in this case. The best results were obtained again using 125 coefficients (see Fig. 11). The error rate obtained in this case was 6.06%, the average *FP* rate was

2.31%, and the average *FN* rate was 3.75%. As can be observed from Fig. 12.(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. 12.(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. 12.(c)). Overall, feature sets *Q125* and *T123* demonstrated best performance when using *SVMs*. For comparison purposes, we tested these two feature sets using *NNs*. The average error rate of *T125NN* was 14.78%, while that of *Q125NN* was 16.02%. Once again, *SVMs* yielded better performance.



(a) (b)

Fig. 11. HV using quantized/truncated wavelet features (a) Error rate; (b) FPs and FNs.



(a) (b) (c)

Fig. 12. Performances v.s. number of coefficients kept. (a). Detection accuracy. (b). FPs. (c). FNs

D. HV using Gabor Features

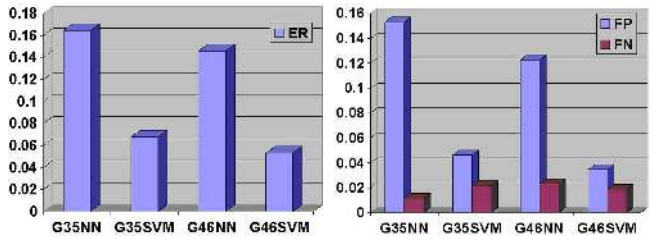
Two different Gabor feature sets were investigated in this paper. The first was extracted using a filter bank with 4 scales and 6 orientations(Fig. 6.b), referred to as *G46*. The second one was extracted using a filter bank with 3 scales and 5 orientations (*G35*), illustrated in Fig. 6.a. First, we evaluated the performance of the two feature sets using *NNs*. We call these two methods *G46NN* and *G35NN*. Fig. 13.a shows the error rates while Fig. 13.b shows the FP/FN rates. *G46NN* achieved an average error rate of 14.57%, an average FP rate of 12.27% and FN rate of 2.31%. Slightly worse than *G46*, the error rate of *G35N* was 16.45%. Then, we applied SVMs on these two feature sets. We refer to them as *G46SVM* and *G35SVM*. Fig. 13 illustrates that SVMs performed much better than NNs. In particular, the average error rate of *G46SVM* was 5.33%, the FP rate was 3.46% and FN rate is 1.88%. The error rates, FP and FN of *G35SVM* were 6.78%, 4.62% and 2.16% correspondingly.

Fig. 10 shows some successful detection examples using *G46SVM* (the same examples were presented earlier using wavelet features). Gabor features seem to have similar properties to wavelet features - model general shape information (Fig. 10.(a)), have good generalization properties(Fig. 10.(b)), and demonstrate some degree of insensitivity to illumination changes (Fig. 10.(c-d)).

E. HV using Combined Wavelet and Gabor Features

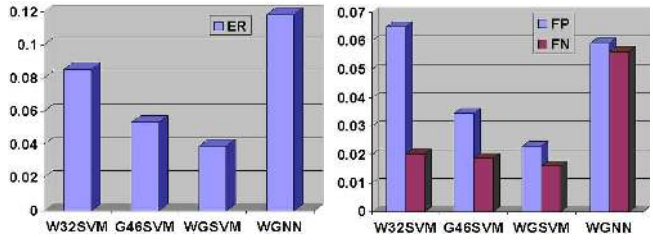
A careful analysis of our results using wavelet and Gabor features revealed that, many times, the two approaches would make different classification errors. This observation motivated us to consider a simple feature fusion approach by simply combining wavelet features with Gabor features, referred to as *GWSVM*. In particular, we chose Gabor features extracted using a filter bank with 4 scales and 6 orientations on 32×32 images, and the original wavelet features described in Section V-B. Fig. 14(a) and (b) show 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 Gabor feature alone (i.e., *G46SVM*) yielded an error rate of 5.33%, while using wavelet features alone (i.e., *W32SVM*) yielded an error rate of 8.52%. Using the combined feature set and *NNs*, the error rate achieved was 11.54%, which was lower than *G46NN* (i.e., 14.57%) or *W32NN* (i.e., 14.81%).

Fig. 15 shows some examples that were classified correctly by the *GWSVM* approach, however, neither *GSVM* nor *WSVM* were able to perform correct classification in all cases. Fig. 15(a), for example, shows a case that was classified correctly by the *GSVM* approach but incorrectly by



(a) (b)

Fig. 13. HV using Gabor features and 32×32 images (a) Error rate; (b) FP and FN.



(a) (b)

Fig. 14. HV using combined wavelet and Gabor features(a) Error rate; (b) FPs and FNs.

the *WSVM* approach. Fig. 15(c) shows another case which was classified incorrectly by the *GSVM* but correctly by the *WSVM* approach. Neither *GSVM* nor *WSVM* were able to classify correctly the case shown in Fig. 15(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 considered 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, *P95SVM*), while the lowest error rate using wavelet features was 6.06% (i.e, *Q125SVM*), 5.33% using Gabor features (i.e., *G46SVM*), and 3.89% using the combined feature set (i.e., *WGSVM*). 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 that *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 yield 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., a form of subset feature selection), truncated/quantized feature based methods demonstrated different performance. 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 [38][39][40].

Fifth, feature fusion can help to 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 difficult to use the *WGSVM* approach because of higher computational requirements (i.e., requires computing both wavelet and Gabor features). The performance of *G46SVM* (i.e., using Gabor feature only) was slightly worse than the *WGSVM*. Thus, our real time system was based on the *G46SVM* approach.

VI. Real time system

In order to evaluate the performance of the two-step vehicle detection system, tests were carried out under different driving condition. Fig. 16 and Fig. 17 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. 16 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. 17(a-f) shows some detection results under this scenario, where our

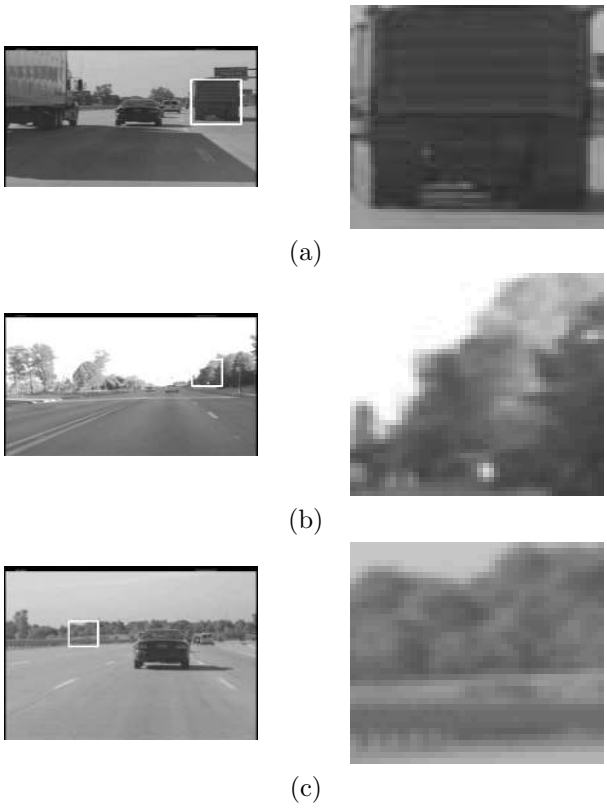


Fig. 15. Cases where either the *GSVM* approach or the *WSVM* approach had failed to perform correct classification (all cases were classified correctly by the *GWSVM* approach).

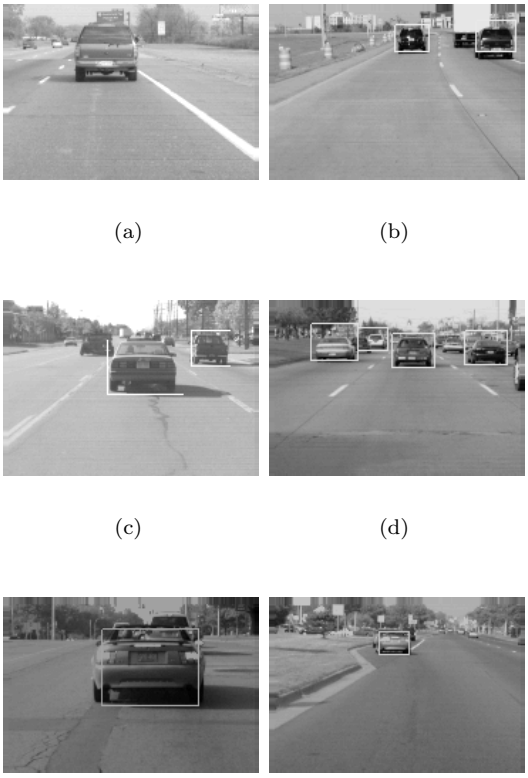


Fig. 16. Vehicle detection examples in rather simple scenes

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 background, heavy congested traffic, etc. Fig. 17(g-h) shows two successful examples under this scenario.

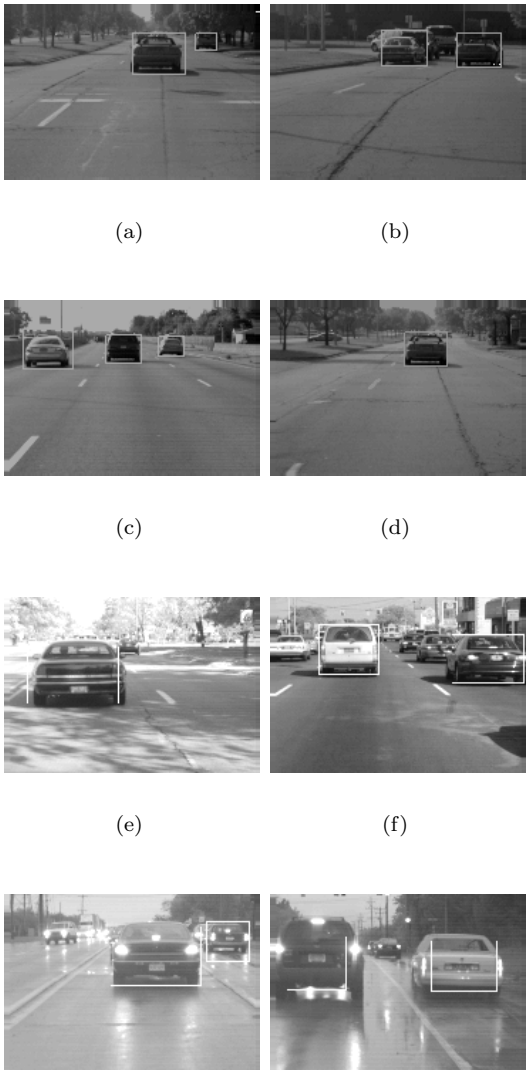
We have achieved a detection rate of approximately 10 frame per second (NTSC: processing on the average every third frame) using a standard PC machine (Pentium III 1133 MHz), without making particular efforts to optimize our code. This is an average performance since some times images can be processed much faster than others (i.e., when there is only one vehicle present). It should be mentioned that vehicle detection for pre-crash 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 70mph, 10Hz corresponds to a 3 meter 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.

VII. Conclusions 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 v.s. non-vehicle). The focus of this paper is feature extraction and classification for vehicle detection. We have investigated five different feature extraction methods (i.e., PCA features, wavelet features, Truncated/Quantized Wavelet Features, Gabor Features, 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 multi-scale driven hypothesis generation step and an appearance-based hypothesis verification step. The multi-scale 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 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 4 scales and 6 orientations, and SVMs (*G46SVM*).

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



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 17. Vehicle detection examples in complex scenes

ditions, such as inclement weather and heavy congested traffic.

Acknowledgements

This research was supported by Ford Motor Company under grant No.2001332R, the University of Nevada, Reno under an Applied Research Initiative (ARI) grant, and in part by NSF under CRCD grant No.0088086.

References

[1] W. Jones, "Keeping cars from crashing," *IEEE Spectrum*, pp. 40–45, September, 2001.

[2] C. Tzomakas and W. V. Seelen, "Vehicle detection in traffic secens using shadows," *Internal Report 98-06, Institut fur Neuroinformatik*, 1998.

[3] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. von Seelen, "An image processing system for driver assistance," *Image and Vision Computing*, vol. 18, pp. 367–376, 2000.

[4] A. Kuehnle, "Symmetry-based recognition for vehicle rears," *Pattern Recognition Letters*, vol. 12, pp. 249–258, 1991.

[5] T. Zielke, M. Brauckmann and W. V. Seelen, "Intensity and edge-based symmetry detection with an application to car-following," *CVGIP:Image Understanding*, vol. 58, pp. 177–190, 1993.

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

[7] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. on Image Processing*, vol. 7, pp. 62–81, 1998.

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

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

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

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

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

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

[14] J. Ferryman, A. Worrall, G. Sullivan, and K. Baker, "A generic deformable model for vehicle recognition," *Proceedings of British Machine Vision Conference*, pp. 127–136, 1995.

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

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

[17] H. Schneiderman and T. Kanade, "Probabilistic modeling of local appearance and spatial relationships for object recognition," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 45–51, 1998.

[18] H. Schneiderman, *A statistical approach to 3D object detection applied to faces and cars*. CMU-RI-TR-00-06, 2000.

[19] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," *European Conference on Comptuer vision*, pp. 18–32, 2000.

[20] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.

[21] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using gabor filters and support vector machines," *International Conference on Digital Signal Processing*, July, 2002,Greece.

[22] Z. Sun, G. Bebis, and R. Miller, "Quantized wavelet features and support vector machines for on-road vehicle detection," *Seventh International Conference on Control, Automation, Robotics and Vision*, December, 2002, Singapore.

[23] Z. Sun, G. Bebis, and R. Miller, "Improving the performance of on-road vehicle detection by combining gabor and wavelet features," *IEEE Fifth International Conference on Intelligent Transportation Systems* , September, 2002, Singapore.

[24] A. Jain, R. Duin and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[25] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991.

[26] G. Garcia, G. Zikos, and G. Tziritas, "Wavelet packet analysis for face recognition," *Image and Vision Computing*, vol. 18, pp. 289–297, 2000.

[27] C. Jacobs, A. Finkelstein and D. Salesin, "Fast multiresolution image querying," *Proceedings of SIGGRAPH*, pp. 277–286, 1995.

[28] J. Daugman, "Complete discrete 2-d gabor transforms by neural network for image analysis and compression," *IEEE Transac-*

tions on Acoustics, Speech, and Signal Processing, vol. 36, no. 7, pp. 1169–1179, 1988.

[29] T. Weldon, W. Higgins and D. Dunn, "Efficient gabor filter design for texture segmentation," *Pattern Recognition*, vol. 29, no. 12, pp. 2005–2015, 1996.

[30] A. Jain and F. Farrokhnia, "Unsupervised texture segementation using gabor filters," *Pattern Recognition*, vol. 23, pp. 1167–1186, 1991.

[31] R. Wurtz, "Object recognition robust under translations, deformations, and changes in background," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 769–775, 1997.

[32] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[33] P. Kuizinga, N. Petkov and S. Grigorescu, "Comparison of texture features based on gabor filters," *Proceedings of the 10th International Conference on Image Analysis and Processing*, pp. 142–147, 1999.

[34] G. Cybenko, "Approximation by superposition of sigmoidal function," *Mathematics of Control, Singals, and Systems*, vol. Chap. 2, pp. 303–314, 1989.

[35] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[36] C. Burges, "Tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 955–974, 1998.

[37] G. Bebis, S. Uthiram, and M. Georgiopoulos, "Face detection and verification using genetic search," *International Journal on Artificial Intelligence Tools*, vol. 9, no. 2, pp. 225–246, 2000.

[38] Z. Sun, X. Yuan, G. Bebis, and S. Louis, "Neural-network-based gender classification using genetic eigen-feature extraction,," *IEEE International Joint Conference on Neural Networks*, May, 2002.

[39] Z. Sun, G. Bebis, X. Yuan, and S. Louis, "Genetic feature subset selection for gender classification: A comparison study," *IEEE International Workshop on Application of Computer Vision*, Dec., 2002.

[40] Z. Sun, G. Bebis, and R. Miller, "Boosting object detection using feature selection," *IEEE International Conference on Advanced Video and Signal Based Surveillance*, July, 2003.