Improving the Performance of On-Road Vehicle Detection by Combining Gabor and Wavelet Features

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

Abstract— **Appearance-based methods represent a promising research direction to the problem of vehicle detection. These methods learn the characteristics of the vehicle class from a set of training images which capture the variability in vehicle appearance. First, training images are represented by a set of features. Then, the decision boundary between the vehicle and non-vehicle classes is computed by modelling the probability distribution of the features in each class or through learning. The purpose of this study is to investigate the effectiveness of two important types of features for vehicle detection based on Haar wavelets and Gabor filters. In both cases, the decision boundary is computed using Support Vector Machines (*SVMs*), a recent development in classification algorithms which performs structural risk minimization to maximize generalization on novel data. Both wavelet and Gabor features have demonstrated good performance in various application domains including face detection and image retrieval. Wavelet features encode edge information, a good feature for vehicle detection. Most importantly, they capture the structure of vehicles at multiple scales. Gabor filters provide a mechanism for obtaining orientation and scale tunable edge and line detectors. Vehicles do contain strong edges and lines at different orientation and scales, thus, this type of features are also very attractive for vehicle detection. Our experimental results and comparisons using real data illustrate the effectiveness of both types of features for vehicle detection, with Gabor features performing better than Haar wavelet features. Careful examination of our results revealed that the two feature sets yield different misclassification errors which led us to the idea of combining them for improving performance. The combined set of features outperformed each feature set alone on completely novel test images, yielding an average error rate of 3.03% compared to 5.19% using Gabor features and 8.52% using Haar wavelet features.**

Keywords— **Vehicle detection, Haar wavelet transform, Gabor filters, Support Vector Machine.**

I. Introduction

Robust and reliable vehicle detection in images acquired by a moving vehicle (on-road vehicle detection) is an important problem with application to driver assistance systems or autonomous, self-guided vehicles. Several factors make on-road vehicle detection very challenging including variability in scale, location, orientation, and pose. Vehicles, for example, come into view with different speeds and may vary in shape, size, and color. Vehicle appearance depends on its pose and is affected by nearby objects. In-class variability, occlusion, and lighting conditions also change the overall appearance of vehicles. Landscape along the road changes continuously while the lighting conditions depend on the time of the day and the weather. Last but not least, real-time processing is required.

In this paper, we consider the problem of rear-view vehicle detection from gray-scale images. A first step of any vehicle detection system is hypothesizing the locations in images where vehicles are present. Then, verification is applied to test the hypotheses. Both steps are equally important and challenging. Approaches to generate the hypothetical locations of vehicles in images include using motion information [1], symmetry [2], shadows [3], and vertical/horizontal edges [4]. Our emphasis here is on improving the performance of the verification step.

Approaches to visual vehicle detection can be classified into three categories: (1) stereo- or motion-based, (2) template-based, and (3) appearance-based. Stereo-based approaches take advantage of the inverse perspective mapping (IMP) [5] to estimate the locations of vehicles and obstacles in images. Bertozzi et al. [6] compute the IMP of both the left and right images and compare them. Based on the comparison, they find objects that are not on the ground plane and use this information to determine the free space in front of a vehicle. In [7], inverse perspective mapping is used to warp the left image to the right image. The main problem with these methods is that they are sensitive to calculated camera parameters. Accurate and robust methods are required to recover these parameters because of vehicle vibrations due to vehicle motion or windy conditions [8]. Template-based methods use predefined patterns of the vehicle class and perform correlation between an input image the the template. Betke et al. [4] proposed a multiple-vehicle detection approach using deformable gray-scale template matching. Improved detection results were claimed by using temporal information. In [9], a deformable model is formed from manually sampled data using Principal Component Analysis (*PCA*). Both the structure and pose of a vehicle can be 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 should capture the variability in vehicle appearance. Usually, the variability of the non-vehicle class is also modelled. First, each training image is represented by a set of local or global features. Then, the decision boundary between the vehicle and non-vehicle class 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 Matthews et al. [10], feature extraction is based on *PCA*. Subwindows containing vehicle candidates were first scaled to a 20x20 subwindow. Each 20x20 subwindow was then divided into 25 4x4 subwindows and each 4x4 subwindow was subjected to *PCA*. The *PCA* features were then fed to a *NN* for classification. Goerick et al. [11] 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. [12] [13]. First, a view-based approach with multiple detectors was used to cope with variation from different viewpoint. Second, a statistical model within each of these detectors was used to account for other variations. The statistics of both object appearance 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 [12] and Haar wavelet features in [13] and their position on the object. A different statistical model was investigated by Weber et al [14]. They represented each vehicle image as a constellation of local features and use the EM algorithm to learn the parameters of the probability distribution of the constellations. An interest operator, followed by clustering, is used to identify important local features in vehicle images. Papageorgiou et al. [15] have proposed using the Haar wavelet transform for feature extraction and Support Vector Machines (*SVMs*) for classification.

The focus of this work is on the problem of feature extraction and classification for rear-view vehicle detection from grayscale images. In particular, we investigate the effectiveness of Haar wavelet features and Gabor features for rear-view vehicle detection, using *SVMs* for classification. Both types of features have shown good performance in various application domains including object detection and image retrieval [13], [15], [16], [17], [18]. Wavelet-based features encode edge information, which is a good feature for vehicle detection. Most importantly, they capture the structure of vehicles at multiple resolution levels. 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. Essentially, they are orientation and scale tunable edge and line detectors. Vehicles do contain strong edges and lines at different orientation and scales, thus, these features could be very powerful for vehicle detection.

Our experimental results and comparisons using real data indicate that both wavelet and Gabor features are powerful for vehicle detection with Gabor features performing better than Haar wavelet features. Careful analysis of our results, however, revealed that many times we would get the correct classification result using wavelet features but not Gabor features and the opposite. Thus, we considered combining the two feature sets. Superior performance was observed both in terms of accuracy and false positives/false negatives using the combined set of features.

The rest of the paper is organized as follows: In Section II, we provide brief overview of Gabor filters, the Haar wavelet transform, and *SVMs*. The feature extraction methodology is described in Section III. A description of the real data set used in our experiments is given in Section IV. Our experimental results and comparisons are presented in Section V. Section VI contains our conclusions and plans for future work.

II. Filters and SVMs Review

A. Gabor Filters

There has been an increased interest in Gabor analysis motivated by biological findings (i.e., the receptive fields of neurons in the visual cortex are known to have shapes that can be approximated by 2-D Gabor filters [19]). Gabor filters have been successfully applied to many image analysis applications including texture analysis [20] [21], handwritten number recognition [22], and image retrieval [18]. An important property of Gabor filters is that they have optimal joint localization both in the spatial and frequency domains [19]. The general functional of the two-dimensional Gabor filter family can be represented as a Gaussian function modulated by an oriented complex sinusoidal signal. Specifically, a two dimensional Gabor filter $g(x, y)$ and its Fourier transform $G(u, v)$ can be written as:

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

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

where $\sigma_u = \frac{1}{2}\pi\sigma_x$, $\sigma_v = \frac{1}{2}\pi\sigma_y$ and

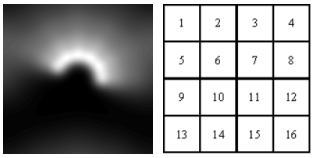
$$\tilde{x} = x\cos\theta + ysin\theta \; and \; \tilde{y} = -x\sin\theta + ycos\theta \quad (3)$$

where σ_u and σ_v are the scaling parameters of the filter (i.e., determine the effective size of the neighborhood of a pixel), W is the center frequency, and θ determines the orientation of the filter (i.e., it will respond stronger to a bar or edge, the normal to which coincides with θ). Gabor filters act as local bandpass filters. Figure (1a) shows the power spectrum of a 4×6 Gabor filter bank (the light areas indicate spatial frequencies and wave orientation).

In this paper, we use the design strategy described in [18]. Given an input image $I(x, y)$, Gabor feature extraction is performed by convolving $I(x, y)$ with a Gabor filter bank:

$$r(x,y) = \int\int I(\xi,\eta)g(x-\xi,y-\eta)d\xi d\eta \quad (4)$$

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 [23]). In this paper, we use Gabor features based on moments, extracted from several subwindows of the input image (see Section III)



(a) (b)

Fig. 1. (a) Gabor filter bank with 4 scales and 6 orientations; (c) feature extraction patches.

B. Haar Wavelet Transform

Wavelets are essentially a type of multiresolution function approximation that allow for the hierarchical decomposition of a signal or image. They have been applied successfully to various problems including object detection [15], [13], face recognition [16] and image retrieval [17]. Any given decomposition of a signal into wavelets involves just a pair of waveforms (mother wavelets). 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:

$$\cdots V_{-1} \subset V_0 \subset V_1 \cdots \subset L^2 \tag{5}$$

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 the dilated and translated version of it:

$$\phi_i^j(x) = 2^{\frac{j}{2}}\phi(2^j x - i). \quad i = 0, \cdots, 2^j - 1. \tag{6}$$

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:

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

Different scaling function $\phi_i^j(x)$ and wavelets $\psi_i^j(x)$ determines different wavelet transform. In this paper, we use Haar wavelet. Haar wavelet 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:

$$\phi(x) = \begin{cases} 1 & for \ \ 0 \le x < 1 \\ 0 & otherwise \end{cases} \tag{8}$$

And the Haar wavelet is defined as:

$$\psi(x) = \begin{cases} 1 & for \ \ 0 \le x < \frac{1}{2} \\ -1 & for \ \ \frac{1}{2} \le x < 1 \\ 0 & otherwise \end{cases} \tag{9}$$

Wavelet features 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.

C. 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 [24] [25]. Given a set of points, which belong to either of two classes, *SVM* finds the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. This is equivalent to performing structural risk minimization to achieve good generalization [24] [25]. Assuming l examples from two classes

$$(x_1, y_1)(x_2, y_2)...(x_l, y_l), \quad x_i \in R^N, y_i \in \{-1, +1\} \tag{10}$$

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 hyperplane is defined as:

$$f(x) = \sum_{i=1}^{l} y_i a_i k(x, x_i) + b \tag{11}$$

where $k(x, x_i)$ is a kernel function and the sign of $f(x)$ indicates the membership of x. Constructing the optimal hyperplane 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 hyperplane.

Suitable kernel functions can be expressed as a dot product in some space and satisfy the Mercer's condition [24]. 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

$$k(x, x_i) = \exp(-\frac{\| x - x_i \|^2}{2\delta^2}) \tag{12}$$

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).

III. Feature Extraction for Vehicle Detection

A. Gabor Filter Features

In this section we describe our Gabor feature extraction procedure. The input to the feature extraction subsystem are the hypothetical vehicle subimages extracted from the input image. As mentioned in the introduction, vehicles contain strong edges and lines at different orientation and scales, information that is also

captured by the Gabor features. The statistics of these features provide a compact and powerful representation for vehicle detection. Instead of extracting these statistics from the whole image, we collect them from several subwindows obtained by subdiving the vehicle subimage. This provides robustness to errors in the hypothesis generation step.

First, each subimage is scaled to a fixed size which is 64×64. Then, it is subdivided into 9 overlapping 32×32 subwindows. Assuming that each subimage consists of 16 16×16 patches (see Figure 1(c)), patches 1,2,5,and 6 comprise the first 32×32 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:

$$[\mu_{11}\sigma_{11}\kappa_{11}, \mu_{12}\sigma_{12}\kappa_{12} \cdots \mu_{69}\sigma_{69}\kappa_{69}] \qquad (13)$$

We have experimented with using the first two moments only, however, much worst results were obtained which implies that skewness information is very important for our problem.

B. Wavelet Features

We use the wavelet decomposition coefficients as our features directly. Performing the wavelet transform on the 64×64 images, yields a vector of 4096 features, which is much bigger than the Gabor feature vector. To compute a feature vector of approximately the same size as the Gabor feature vector, we rescale the images to 32×32, then we perform a 5 level *Haar* wavelet decomposition which yields 1024 coefficients. We do not keep the coefficients in the *HH* subband of the first level since they encode mostly noise [13]. This yielded a vector of 768 features.

C. Feature Combination

The combined feature set contains 1416 features. Since the values of Gabor and wavelet features assume different ranges, first we normalize them in the range [-1 1] before combining them in a single vector.

IV. Dataset

The images used in our experiments were collected in Dearborn, Michigan during 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 caught during different times, different days, and on five different highways. The training set contains 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. Although specific instructions were given to the students, there is some variability in the way the subimages were extracted. For example, certain subimages cover the whole vehicle, others cover the vehicle partially, and others contain the vehicle and some background (see Figure 2). In [15], the subimages were aligned by wrapping 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 can actually improve performance. Each subimage in the training and test sets was scaled to 64×64 and preprocessed to account for different lighting conditions and contrast [26]. 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. Then, histogram equalization was performed to improve contrast.

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 during the training of the neural network classifier which was used for comparison purposes. For testing, we used a fixed set of 231 vehicle and non-vehicle subimages which were extracted from the Summer 2001 data set.

Fig. 2. Subimages for training.

V. Experimental Results and Comparisons

We have performed a number of experiments and comparisons to demonstrate the proposed approach. First, we evaluated the performance of wavelet features using *SVMs*, referred to a *WSVM*. Figure 3(a) shows the error rate of this approach while Figure 3(b) shows the *FP*/*FN* rates. 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 Gabor features using *SVMs*, referred to as *GSVM*.

We used a 4×6 filter bank in these experiments. Figure 3(a) shows the error rate while Figure 3(b) shows the *FP*/*FN* rates. We have also experimented with a 3×5 filter bank without observing much differences in the average error rate except in the number of *FNs* (the 3×5 filter bank yielded lower number of *FNs*). The average error rate using Gabor features was 5.19%, the average *FP* rate was 1.58%, and the average *FN* rate was 3.61%. Obviously, using Gabor features have reduced the error and *FP* rates significantly, however, the *FN* rate was somewhat higher. In terms of *SVM* compactness, the average number of support vectors using Gabor features was 218, 278 less than using wavelet features.

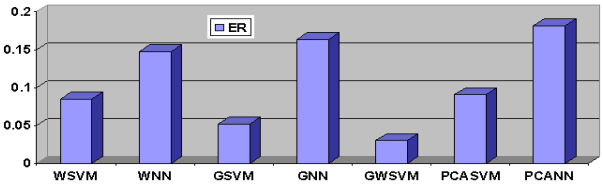
Our literature review in Section I shows that features based on *PCA* and classification using *NNs* have been used quite extensively for vehicle detection. For comparison purposes, we have evaluated the performance of *PCA* features using *SVMs*, referred to as *PCASVM*, and Gabor features using *NNs*, referred to as *GNN*. To extract the *PCA* features, we preserved 95% of the information. The *NN* classifier used was a fully connected, two-layer, feed-forward neural network trained by the back-propagation algorithm. We varied the number of hidden nodes to obtain optimum performance and used cross-validation to stop training. Figure 3(a) shows the performance of the *PCASVM* and the *GNN* approaches. The *PCASVM* approach achieved an average error rate of 9.09%, an average *FP* rate of 7.52%, and average *FN* rate of 1.57%. The *GNN* approach achieved an average error rate of 16.33%, an average *FP* rate of 14.35% and average *FN* rate of 1.98%. Obviously, Gabor features perform much better than *PCA* features while the *SVM* classifier performs better than the *NN* classifier.

Considering the *WSVM* and *GSVM* approaches again, a careful analysis of our results revealed that there were many times where the two approaches would make different classification errors. Thus, we decided to apply a simple fusion approach by simply combining the wavelet and Gabor features, referred to as *GWSVM*. The average error rate obtained in this case was 3.03%, the average *FP* rate was 0.87%, while the average *FN* rate was 2.16% (still slightly higher than using wavelet features alone). Figures 3(a) and (b) show these results. Obviously, feature fusion is a subject that requires further investigation.

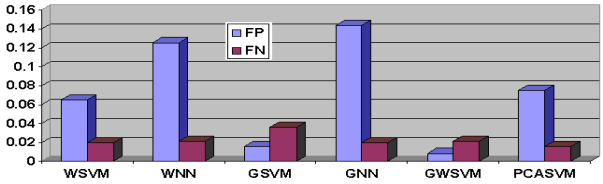
Figure 4 show some of successful detection examples using the *GWSVM* approach. These results illustrate several strong points. Figure 4(a) shows a case where only the general shape of the vehicle is available (i.e., no details) due to its far distance from the camera. The proposed method seems to discard irrelevant details, leading to improved robustness. Figures 4(b) shows an example where the vehicle is detected successfully from its front views. This is despite the fact that we did not include any front views in our training set. Also, the proposed method can tolerate some illumination changes as can be seen from Figures 4(c-d).

The majority of *FNs* were due to the lack of representative examples in the training set or due to some extreme rotations. The *FPs* also were due to the relatively small number of "non-vehicle" examples used for training. Given that the "non-vehicle" class is much larger than the vehicle class, it would make more sense to include much more "non-vehicle" examples in the training sets. Bootstrapping [27] would definitely be very useful in choosing good "non-vehicle" examples.

Figure 5 shows some more examples that have been classified correctly by the *GWSVM* approach, however, either *GSVM* or *WSVM* have failed to perform correct classification in these cases. Figure 5(a), for example, shows a case classified correctly by the *GSVM* approach but incorrectly by the *WSVM* approach. Figure 5(b) 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 Figure 5(d).



(a)



(b)

Fig. 3. Performance of various methods: (a) error rate, (b) false positives and false negatives.

VI. Conclusions and Future Work

We have considered the problem of on-road vehicle detection from rear views of gray-scale images. In particular, we investigated the effectiveness of two types of features using *SVMs* for classification: Haar wavelet features and Gabor features. Our experimental results demonstrated that both types of features are promising for vehicle detection, with Gabor features performing better than Haar wavelet features. We also considered a simple fusion approach by combining wavelet and Gabor features. The combined feature set demonstrated even better performance than either feature set alone. For future work, we plan to investigate more extensively the problem of feature fusion for vehicle detection.

Acknowledgements

This research was supported by Ford Motor Company under grant No.2001332R and in part by NSF under CRCD grant No.0088086. The authors would like to thank Dave Dimeo and Perry MacNeille from Ford Research Lab for their help with the data collection.

References

[1] N. Friedman and S. Russell, "Image segmentation in video sequences: a probabilistic approach," *Proc. Thirteenth Conf. on Uncertainty in Artificial Intelligence (UAI 97)*, 1997.

[2] A. Kuehnle, "Symmetry-based recognition for vehicle rears," *Pattern Recognition Letters*, vol. 12, pp. 249–258, 1991.
[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] M. Betke, E. Haritaglu and L. Davis, "Multiple vehicle detection and tracking in hard real time," *IEEE Intelligent Vehicles Symposium*, pp. 351–356, 1996.
[5] 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.
[6] 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.
[7] G. Zhao and Y. Shini'chi, "Obstacle detection by vision system for autonomous vehicle," *IEEE Intelligent Vehicle Symposium*, pp. 31–36, 1993.
[8] M. Suwa, "A stereo-based vehicle detection method under windy conditions," *IEEE Intelligent Vehicle Symposium*, pp. 246–249, 2000.
[9] 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.
[10] 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.
[11] 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.
[12] 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.
[13] H. Schneiderman, *A statistical approach to 3D object detection applied to faces and cars.* CMU-RI-TR-00-06, 2000.
[14] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," *European Conference on Comptuer vision*, pp. 18–32, 2000.
[15] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.
[16] G. Garcia, G. Zikos, and G. Tziritas, "Wavelet packet analysis for face recognition," *Image and Vision Computing*, vol. 18, pp. 289–297, 2000.
[17] C. Jacobs, A. Finkelstein and D. Salesin, "Fast multiresolution image querying," *Proceedings of SIGGRAPH*, pp. 277–286, 1995.
[18] 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.
[19] J. Daugman, "Complete discrete 2-d gabor transforms by neural network for image analysis and compression," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, 1988.
[20] T. Weldon, W. Higgins and D. Dunn, "Efficient gabor filter design for texture segmentation," *Pattern Recognition*, vol. 29, no. 12, pp. 2005–2015, 1996.
[21] A. Jain and F. Farrokhnia, "Unsupervised texture segementation using gabor filters," *Pattern Recognition*, vol. 23, pp. 1167–1186, 1991.
[22] Y. Hamamoto, S. Uchimura, M. Watanabe, T. Yasuda, Y. Mitani, and S. Tomota, "A gabor filter-based method for recognizing handwritten numerals," *Pattern Recognition*, vol. 31, no.4, pp. 395–400, 1998.
[23] 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.
[24] V. Vapnik, *The Nature of Statistical Learning Theory.* Springer Verlag, 1995.
[25] C. Burges, "Tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 955–974, 1998.
[26] 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.
[27] H. Rowley, S. Baluja and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

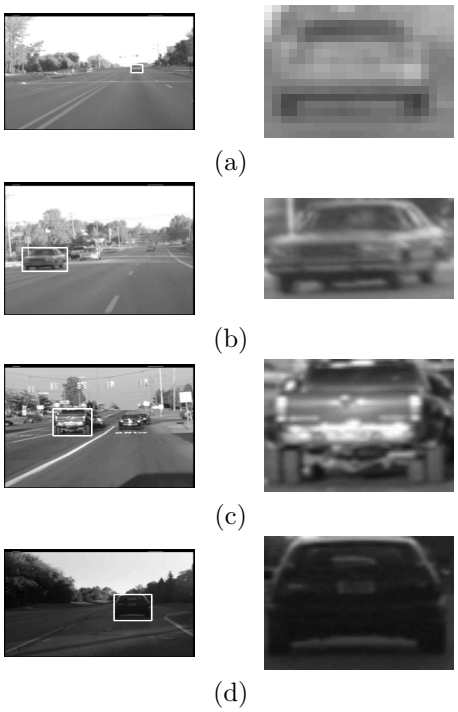


Fig. 4. Successful vehicle detection examples.

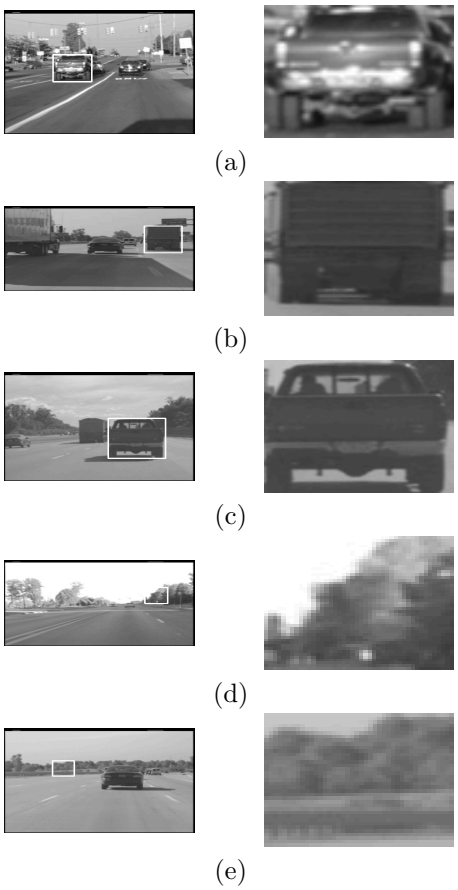


Fig. 5. Cases where either the *GSVM* approach or the *WSVM* approach failed to perform correct classification(all cases are classified correctly by the *GWSVM* approach).