On-Road Vehicle Detection: A Review

Zehang Sun, *Member*, *IEEE*, George Bebis, *Member*, *IEEE*, and Ronald Miller

Abstract—Developing on-board automotive driver assistance systems aiming to alert drivers about driving environments, and possible collision with other vehicles has attracted a lot of attention lately. In these systems, robust and reliable vehicle detection is a critical step. This paper presents a review of recent vision-based on-road vehicle detection systems. Our focus is on systems where the camera is mounted on the vehicle rather than being fixed such as in traffic/driveway monitoring systems. First, we discuss the problem of on-road vehicle detection using optical sensors followed by a brief review of intelligent vehicle research worldwide. Then, we discuss active and passive sensors to set the stage for vision-based vehicle detection. Methods aiming to quickly hypothesize the location of vehicles in an image as well as to verify the hypothesized locations are reviewed next. Integrating detection with tracking is also reviewed to illustrate the benefits of exploiting temporal continuity for vehicle detection. Finally, we present a critical overview of the methods discussed, we assess their potential for future deployment, and we present directions for future research.

Index Terms—Vehicle detection, computer vision, intelligent vehicles.

1 Introduction

Every minute, on average, at least one person dies in a vehicle crash. Auto accidents also injure at least 10 million people each year, two or three million of them seriously. It is predicted that the hospital bill, damaged property, and other costs will add up to 1-3 percent of the world's gross domestic product [1], [2]. With the aim of reducing injury and accident severity, precrash sensing is becoming an area of active research among automotive manufacturers, suppliers and universities. Several national and international projects have been launched over the past several years to investigate new technologies for improving safety and accident prevention (see Section 2).

Vehicle accident statistics disclose that the main threats a driver is facing are from other vehicles. Consequently, developing on-board automotive driver assistance systems aiming to alert a driver about driving environments and possible collision with other vehicles has attracted a lot of attention. In these systems, robust and reliable vehicle detection is the first step. Vehicle detection—and tracking—has many applications including platooning (i.e., vehicles traveling in high speed and close distance in highways), stop and go (vehicles traveling in low speeds and close distance in cities), and autonomous driving.

This paper presents a review of recent vision-based on-road vehicle detection systems where the camera is mounted on the vehicle rather than being fixed such as in traffic/driveway monitoring systems. Vehicle detection using optical sensors is very challenging due to huge within class variabilities in vehicle appearance. Vehicles may vary in shape (Fig. 1a), size, and color. The appearance of a vehicle depends on its pose (Fig. 1b) and is affected by nearby objects. Complex outdoor environments (e.g., illumination conditions (Fig. 1c), unpredictable interaction between traffic participants, cluttered background (Fig. 1d) are difficult to control. On-road vehicle detection also requires faster processing than other applications since the vehicle speed is bounded by the processing rate. Another key issue is robustness to vehicle's movements and drifts.

More general overviews on various aspects of intelligent transportation systems (e.g., infrastructure-based approaches such as sensors detecting the field emitted by permanent magnetic markers or electric wires buried in the road) as well as vision-based intelligent transportation systems (e.g., driver monitoring, pedestrian detection, sign recognition, etc.) can be found in [2], [3], [4], [5], [6], [7]. Several special issues have also focused on computer vision applications in intelligent transportation systems [8], [9], [10], [11].

This paper is organized as follows: In Section 2, we present a brief introduction of vision-based intelligent vehicle research worldwide. A brief review of active and passive sensors is presented in Section 3. Detailed reviews of Hypothesis Generation (HG) and Hypothesis Verification (HV) methods are presented in Sections 5 and 6 while exploring temporal continuity by integrating detection with tracking is discussed in Section 7. In Section 8, we provide a critical overview of the HG and HV methods reviewed. Challenges and future research directions are presented in Section 9. Finally, our conclusions are given in Section 10.

2 Vision-Based Intelligent Vehicle Research Worldwide

Vision-based vehicle detection for driver assistance has received considerable attention over the last 15 years. There are at least three reasons for the blooming research in this field: 1) the startling losses both in human lives and finance caused by vehicle accidents, 2) the availability of feasible technologies accumulated within the last 30 years of computer vision research, and 3) the exponential growth in processor speeds have paved the way for running

- *Z. Sun is with eTreppid Technologies, LLC., 755 Trademark Drive, Reno, NV 89521. E-mail: zehang@etreppid.com.*
- *G. Bebis is with the Computer Vision Laboratory, Department of Computer Science and Engineering, University of Nevada, 1664 North Virginia Street, Reno, NV 89557. E-mail: bebis@cse.unr.edu.*
- *R. Miller is with the Vehicle Design R & A Department, Ford Motor Company, 1 American Road, Dearborn, MI 48126-2798. E-mail: rmille47@ford.com.*

Manuscript received 11 Aug. 2004; revised 15 Aug. 2005; accepted 22 Aug. 2005; published online 13 Mar. 2006.
Recommended for acceptance by H. Sawhney.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0422-0804.

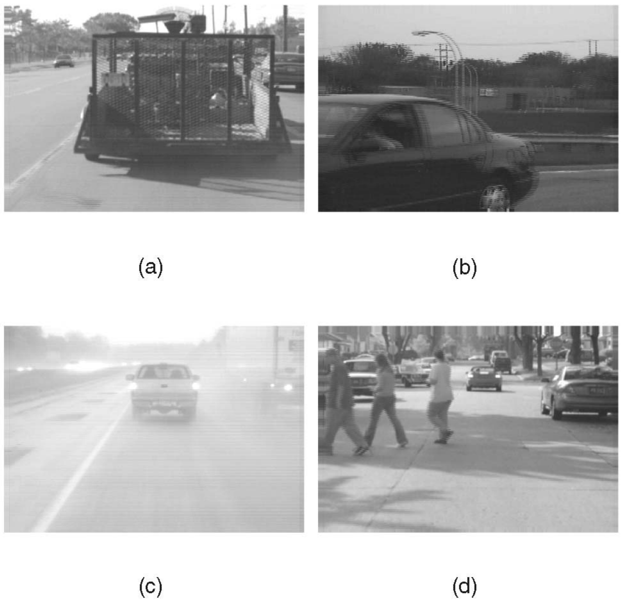

Fig. 1. The variety of vehicle appearances poses a big challenge for vehicle detection.

computation-intensive video-processing algorithms even on a low-end PC in realtime.

With the ultimate goal of building autonomous vehicles, many government institutions, automotive manufacturers and suppliers, and R&D companies have launched various projects worldwide, involving a large number of research units working cooperatively. These efforts have produced several prototypes and solutions, based on rather different approaches [5], [12], [13], [6]. Looking at research on intelligent vehicles worldwide, Europe pioneers the research, followed by Japan and United States.

In Europe, the *PROMETHEUS* project (Program for European Traffic with Highest Efficiency and Unprecedented Safety) started this exploration in 1986. More than 13 vehicle manufactures and research institutes from 19 European countries were involved. Several prototype vehicles and systems were designed and demonstrated as a result of *PROMETHEUS*. In 1987, the UBM (Universitaet der Bundeswehr Munich) test vehicle **VaMoRs** demonstrated the capability of fully autonomous longitudinal and lateral vehicle guidance by computer vision on a 20 km free section of highway at speed up to 96 km/h. Vision was used to provide input for both lateral and longitudinal control. That was considered as the first milestone.

Further development of this work has been in collaboration with von Seelen's group [14] and Daimler-Benz **VITA** project (VIsion Technology Application) [15]. Long range autonomous driving has been demonstrated by the **VaMP** of UBM in 1995. The trip was from Munich to Odense, Denmark, more than 1,600 km. About 95 percent of the distance was driven without intervention of the safety driver [3]. Another experimental vehicle, mobile laboratory (MOB-LAB), was also part of the PROMETHEUS project [16]. It was equipped with four cameras, several computers, monitors, and a control-panel to give a visual feedback and warnings to the driver. One of the most promising subsystems in the MOB-LAB was the Generic Obstacle and Lane Detection (GOLD) system. The GOLD system, utilizing a stereo rig in the MOB-LAB, addressed both lane and obstacle detection at the same time. The lane detection was based on a pattern matching technique, while the obstacle detection was reduced to the determination of the free-space in front of the vehicle using the stereo image pairs without 3D reconstruction. The GOLD system has been ported on ARGO, a Lancia Thema passenger car with automatic steering capabilities [17].

Although the first research efforts on developing intelligent vehicles were seen in Japan in the 1970s, significant research activities have been triggered by prototype vehicles built in Europe in the late-1980s and early-1990s. MITI, Nissan, and Fujitsu pioneered the research in this area by joining forces in the project "Personal Vehicle System" [18], a project with deep influence on Japan. In 1996, the *Advanced Cruise-Assist Highway System Research Association* (AHSRA) was established among automobile industries and a large number of research centers in Japan [5]. The Japanese Smartway concept car will implement some driver aid features, such as lane keeping, intersection collision avoidance, and pedestrian detection. A model deployment project was planned to be operational by 2003 and national deployment in 2015 [6].

In the United States, a number of initiatives have been launched to address this problem. In 1995, the US government established the *National Automated Highway System Consortium* (NAHSC) [19], and launched the *Intelligent Vehicle Initiative* (IVI) in 1997. Several promising prototype vehicles/systems have been investigated and demonstrated within the last 15 years [20]. The Navlab group at Carnegie Mellon University has a long history of development of automated vehicles and intelligent systems for driver assistance. The group has produced a series of 11 vehicles, Navlab 1 through Navlab 11. Their applications have included off-road scouting, automated highways, run-off-road collision prevention, and driver assistance for maneuvering in crowded city environments. In 1995, NavLab5 demonstrated long range partially autonomous driving (i.e., automatic lateral control) on highways from the east coast to the west. With a more than 5,000 km trip, 98 percent of the distance was driven without intervention of the human safety driver [21]. The latest model in the Navlab family is the Navlab 11, a robot Jeep Wrangler equipped with a wide variety of sensors for short-range and midrange obstacle detection [22], [23], [20].

Major motor companies including Ford and GM have poured great effort into this research and already demonstrated several promising concept vehicles. US government agencies are very supportive of intelligent vehicle research. Recently, the *US Department of Transportation* (USDOT) has launched a five year, 35 million dollar project with GM to develop and test preproduction rear-end collision avoidance system [6]. In March 2004, the whole world was stimulated by the "grand challenge" organized by The *US Defense Advanced Research Projects Agency* (DARPA) [24]. In this competition, 15 fully autonomous vehicles attempted to independently navigate a 250-mile (400 km) desert course within a fixed time period, all with no human intervention whatsoever—no driver, no remote-control, just pure computer-processing and navigation horsepower, competing for a 1 million cash prize. Although, even the best vehicle (i.e., "Red Team" from Carnegie Mellon) made only seven miles, it was a very big step towards building autonomous vehicles in the future.

3 Active versus Passive Sensors

The most common approach to vehicle detection is using active sensors [25] such as radar-based (i.e., millimeter-wave) [26], laser-based (i.e., lidar) [27], [28], and acoustic-based [29]. In radar, radio waves are transmitted into the atmosphere, which scatters some of the power back to the radar's receiver. A Lidar (i.e., "Light Detection and Ranging") also transmits and receives electromagnetic radiation, but at a higher frequency; it operates in the ultraviolet, visible, and infrared region of the electromagnetic spectrum.

The reason that these sensors are called active is because they detect the distance of objects by measuring the travel time of a signal emitted by the sensors and reflected by the objects. Their main advantage is that they can measure certain quantities (e.g., distance) directly without requiring powerful computing resources. Radar-based systems can "see" at least 150 meters ahead in fog or rain, where average drivers can see through only 10 meters or less. Lidar is less expensive to produce and easier to package than radar; however, with the exception of more recent systems, lidar does not perform as well as radar in rain and snow. Laser-based systems are more accurate than radars, however, their applications are limited by their relatively higher costs. Prototype vehicles employing active sensors have shown promising results. However, when a large number of vehicles move simultaneously in the same direction, interference among sensors of the same type poses a big problem. Moreover, active sensors have, in general, several drawbacks, such as low spatial resolution and slow scanning speed. This is not the case with more recent laser scanners, such as SICK [27], which can gather high spatial resolution data at high scanning speeds.

Optical sensors, such as normal cameras, are usually referred to as passive sensors [25] because they acquire data in a nonintrusive way. One advantage of passive sensors over active sensors is cost. With the introduction of inexpensive cameras, we could have both forward and rearward facing cameras on a vehicle, enabling a nearly 360 degree field of view. Optical sensors can be used to track more effectively cars entering a curve or moving from one side of the road to another. Also, visual information can be very important in a number of related applications, such as lane detection, traffic sign recognition, or object identification (e.g., pedestrians and obstacles), without requiring any modifications to road infrastructures. Several systems presented in [5] demonstrate the principal feasibility of vision-based driver assistance systems.

4 The Two Steps of Vehicle Detection

On-board vehicle detection systems have high computational requirements as they need to process the acquired images at real-time or close to real-time to save time for driver reaction. Searching the whole image to locate potential vehicle locations is prohibitive for real-time applications. The majority of methods reported in the literature follow two basic steps: 1) HG where the locations of possible vehicles in an image are hypothesized and 2) HV where tests are performed to verify the presence of vehicles in an image (see Fig. 2). Although there is some overlap in the methods employed for each step, this taxonomy provides a good framework for discussion throughout this survey.

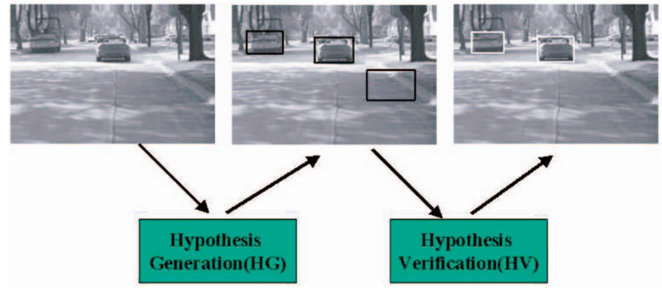

Fig. 2. Illustration of the two-step vehicle detection strategy.

5 HG Methods

Various HG approaches have been proposed in the literature, which can be classified into one of the following three categories: 1) *knowledge-based*, 2) *stereo-based*, and 3) *motion-based*. The objective of the HG step is to find candidate vehicle locations in an image quickly for further exploration. Knowledge-based methods employ a priori knowledge to hypothesize vehicle locations in an image. Stereo-based approaches take advantage of the Inverse Perspective Mapping (IPM) [30] to estimate the locations of vehicles and obstacles in images. Motion-based methods detect vehicles and obstacles using optical flow. The hypothesized locations from the HG step form the input to the HV step, where tests are performed to verify the correctness of the hypotheses.

5.1 Knowledge-Based Methods

Knowledge-based methods employ a priori knowledge to hypothesize vehicle locations in an image. We review below some representative approaches using information about symmetry, color, shadow, geometrical features (e.g., corners, horizontal/vertical edges), texture, and vehicle lights.

5.1.1 Symmetry

As one of the main signatures of man-made objects, symmetry has been used often for object detection and recognition in computer vision [31]. Images of vehicles observed from rear or frontal views are in general symmetrical in the horizontal and vertical directions. This observation has been used as a cue for vehicle detection in several studies [32], [33]. An important issue that arises when computing symmetry from intensity, however, is the presence of homogeneous areas. In these areas, symmetry estimations are sensitive to noise. In [4], information about edges was included in the symmetry estimation to filter out homogeneous areas (see Fig. 3). In a different study, Seelen et al. [34] formulated symmetry detection as an optimization problem which was solved using Neural Networks (NNs).

5.1.2 Color

Although few existing systems use color information to its full extent for HG, it is a very useful cue for obstacle detection, lane/road following, etc. Several prototype systems have investigated the use of color information as a cue to follow lanes/roads [35], or segment vehicles from background [36], [37]. Crisman et al. [35] used two closely positioned cameras to extend the dynamic range of a single camera. One camera was set to capture the shadowed area by opening its iris, and the other the sunny area by using a closed iris. Combining color information (i.e., red, green, and blue) from the two images, he formed a six-dimensional color space. A Gaussian

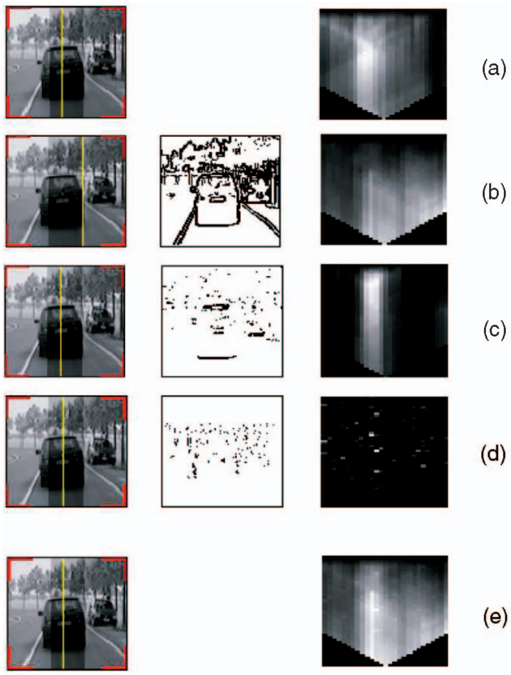

Fig. 3. Computing the symmetry: (a) gray-level symmetry, (b) edge symmetry, (c) horizontal edges symmetry, (d) vertical edges symmetry, and (e) total symmetry (from [70]).

distribution was fit to this color space and each pixel was classified as either road or nonroad pixel.

Buluswar and Draper [36] used a nonparametric learning-based approach for object segmentation and recognition. A multivariate decision tree was utilized to model the object in the RGB color space from a number of training examples. Among various color spaces, the RGB color space ensures that there is no distortion in the initial color information, however, color features are highly correlated—it is difficult to evaluate the difference of two colors from their distance in RGB color space. In [37], Guo et al. chose the L*a*b color space instead. The L*a*b color space has the property that it maps equally distinct color differences into equal Euclidean distances. An incremental region fitting method was investigated in the L*a*b color space for road segmentation [37].

5.1.3 Shadow

Using shadow information as a sign pattern for vehicle detection was initially discussed in [38]. By investigating image intensity, it was found that the area underneath a vehicle is distinctly darker than any other areas on an asphalt paved road. A first attempt to deploy this observation can be found in [39], although there was no systematic way to choose appropriate threshold values. The intensity of the shadow depends on the illumination of the image, which in turn depends on weather conditions. Therefore, the thresholds cannot be, by any means, fixed. To segment the shadow area, a low and a high threshold are required. However, it is obvious that it is hard to find a low threshold for a shadow area. The high threshold can be estimated by analyzing the gray level of the "free driving space"—the road right in front of the prototype vehicle.

Tzomakas and Seelen [40] followed the same idea and proposed a method to determine the threshold values. Specifically, a normal distribution was assumed for the

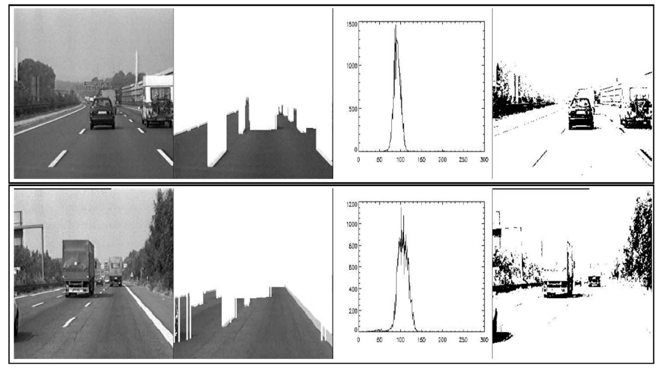

Fig. 4. Free driving spaces, the corresponding gray-value histograms and the thresholded images (from [40]).

intensity of the free driving space. The mean and variance of the distribution were estimated using Maximum Likelihood (ML) estimation. The high threshold of the shadow area was defined as the limit where the distribution of the road gray values declined to zero on the left of the mean, which was approximated by $m - 3\sigma$, where m is the mean and σ is the standard deviation. This algorithm is depicted in Fig. 4. It should be noted that the assumption about the distribution of the road pixels might not always hold true.

5.1.4 Corners

Exploiting the fact that vehicles in general have a rectangular shape with four corners (upper-left, upper-right, lower-left, and lower-right), Bertozzi et al. proposed a corner-based method to hypothesize vehicle locations [41]. Four templates, each of them corresponding to one of the four corners, were used to detect all the corners in an image, followed by a search method to find the matching corners (i.e., a valid upper-left corner should have a matched lower-right corner).

5.1.5 Vertical/Horizontal Edges

Different views of a vehicle, especially rear/frontal views, contain many horizontal and vertical structures, such as rear-window, bumper, etc. Using constellations of vertical and horizontal edges has shown to be a strong cue for hypothesizing vehicle presence. In an effort to find pronounced vertical structures in an image, Matthews et al. [42] used edge detection to find strong vertical edges. To localize left and right position of a vehicle, they computed the vertical profile of the edge image (i.e., by summing the pixels in each column) followed by smoothing using a triangular filter. By finding the local maximum peaks of the vertical profile, they claimed that they could find the left and right position of a vehicle. A shadow method, similar to that in [40], was used to find the bottom of the vehicle. Because there were no consistent cues associated with the top of a vehicle, they detected it by assuming that the aspect ratio of any vehicle was one.

Goerick et al. [43] proposed a method called Local Orientation Coding (LOC) to extract edge information. An image obtained by this method consists of strings of binary code representing the directional gray-level variation in the pixel's neighborhood. These codes carry essentially edge information. Handmann et al. [44] also used LOC, together with shadow information, for vehicle detection. Parodi and Piccioli [45] proposed to extract the general structure of a traffic scene by first segmenting an image into four regions: pavement, sky, and two lateral regions using edge grouping.

Fig. 5. Multiscale hypothesis generation—size of the images: 90×62 (first row), 180×124 (second row), and 360×248 (third row). 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. All images have been scaled back to 360×248 for illustration purposes (from [49]).

Groups of horizontal edges on the detected pavement were then considered for hypothesizing the presence of vehicles.

Betke et al. [46] utilized edge information to detect distant cars. They proposed a coarse-to-fine search method looking for rectangular objects. The coarse search checked the whole image to see if a refined search was necessary, and a refined search was activated only for small regions of the image, suggested by the coarse search. The coarse search looked through the whole edge maps for prominent edges, such as long uninterrupted edges. Whenever such edges were found, the refined search process was started in that region.

In [47], vertical and horizontal edges were extracted separately using the Sobel operator. Then, two edge-based constraint filters (i.e., rank filter and attached line edge filter) were applied on those edges to segment vehicles from background. The edge-based constraint filters were derived from prior knowledge about vehicles. Assuming that lanes have been successfully detected, Bucher et al. [48] hypothesized vehicle presence by scanning each lane starting from the bottom to a certain vertical position, corresponding to a predefined maximum distance in the real world. Potential candidates were obtained if a strong horizontal segment delimited by the lane borders had been found. A multiscale approach which combines subsampling with smoothing to hypothesize possible vehicle locations more robustly was proposed in [49] to address the above problems.

Three levels of detail were used: (360×248), (180×124), and (90×62). At each level, the image was processed by applying the following steps:

1. low pass filtering (e.g., first column of Fig. 5);
2. vertical edge detection (e.g., second column of Fig. 5), vertical profile computation of the edge image (e.g., last column of Fig. 5), and profile filtering using a low pass filter;
3. horizontal edge detection (e.g., third column of Fig. 5), horizontal profile computation of the edge image (e.g., last column of Fig. 5), and profile filtering using a low pass filter; and
4. local maxima and minima detection (e.g., peaks and valleys) of the two profiles.

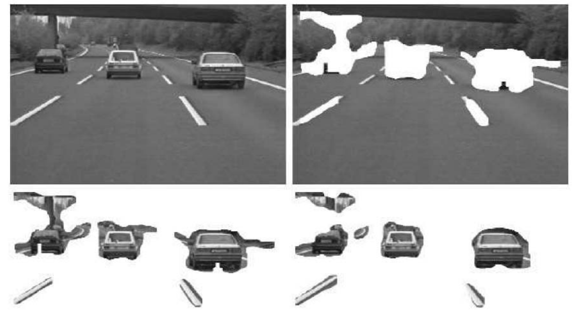

Fig. 6. Image, free driving space and image segmentation-based local image entropy and co-occurrence-based image segmentation (from [50]).

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, all the local maxima at that level are found first. Although the resulted low resolution images have lost fine details, important vertical and horizontal structures are mostly preserved (e.g., first row of Fig. 5). Once the maxima at the coarsest level have been found, they are traced down to the next finer level. The results from this level are finally traced down to the finest level where the final hypotheses are generated.

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

5.1.6 *Texture*

The presence of vehicles in an image causes local intensity changes. Due to general similarities among all vehicles, the intensity changes follow a certain texture pattern [50]. This texture information can be used as a cue to narrow down the search area for vehicle detection. Entropy was first used as a measure for texture detection. For each image pixel, a small window was chosen around it, and the entropy of that window was considered as the entropy of the pixel. Only regions with high entropy were considered for further processing.

Another texture-based segmentation method suggested in [50] uses co-occurrence matrices introduced in [51]. The co-occurrence matrix contains estimates of the probabilities of co-occurrences of pixel pairs under predefined geometrical and intensity constraints. Fourteen statistical features were computed from the co-occurrence matrices [51]. For typical textures of geometrical structures, like trucks and cars, four measurements out of the 14 were found to be critical for object detection (i.e., energy, contrast, entropy, and correlation) [50]. Using co-occurrence matrices for texture detection is more accurate in general than using the entropy method mentioned earlier since co-occurrence matrices employ second order statistics as opposed to histogram information employed by the entropy method (see Fig. 6). However, computing the co-occurrence matrices is expensive.

5.1.7 Vehicle Lights

Most of the cues discussed above are not helpful for nighttime vehicle detection—it would be difficult or impossible to detect shadows, horizontal/vertical edges, or corners in images obtained at night conditions. A salient visual feature during night time is the vehicle lights. Cucchiara and Piccardi [52] have used morphological analysis to detect vehicle light pairs in a narrow inspection area. The morphological operator also considered the shape, size, and minimal distance between vehicles to provide hypotheses.

5.2 Stereo-Vision-Based Methods

There are two types of methods that use the stereo information for vehicle detection. One uses disparity map, while the other uses an antiperspective transformation—Inverse Perspective Mapping (IPM). We assume that camera parameters have already been computed through calibration.

5.2.1 Disparity Map

The difference in the left and right images between corresponding pixels is called disparity. The disparities of all the image points form the disparity-map. If the parameters of the stereo rig are known, the disparity map can be converted into a 3D map of the viewed scene. Computing the disparity map is very time consuming due to the requirement of solving the correspondence problem for every pixel; however, it is possible to do it in real-time using a Pentium class processor or embedded hardware [53]. Once the disparity map is available, all the pixels within a depth of interest according to a disparity interval are determined and accumulated in a disparity histogram. If an obstacle is present within the depth of interest, then a peak will occur at the corresponding histogram bin (i.e., similar idea to the Hough transform).

In [54], it was argued that, to solve the correspondence problem, area-based approaches were too computationally expensive, and disparity maps from feature-based methods were not dense enough. A local feature extractor (i.e., "structure classification") was proposed to solve the correspondence problem faster. According to this approach, each pixel was classified into various categories (e.g., vertical edge pixels, horizontal edge pixels, corner edge pixels, etc.) based on the intensity differences between the pixel and its four direct neighbors. To simplify finding pixel correspondences, the optical axes of the stereo-rig were aligned in parallel (i.e., corresponding points were on the same row in each image). Accordingly, their search for corresponding pixels was reduced to a simple test (i.e., whether two pixels belong to the same category or not). Obviously, there are cases where this approach does not yield unique correspondences. To address this problem, they further classified the pixels by their associated disparities into several bins by constructing a disparity histogram. The number of significant peaks in the histogram indicated how many possible objects were present in the images.

5.2.2 Inverse Perspective Mapping

The term "Inverse Perspective Mapping" does not correspond to an actual inversion of perspective mapping [30], which is mathematically impossible. Rather, it denotes an inversion under the additional constraint that inversely mapped points lie on the horizontal plane. If we consider a point p in the 3D space, perspective mapping implies a line passing through this point and the center of projection N, see Fig. 7. To find the image of the point, we intersect the line with the image plane. IPM is defined by the following procedure: For a point $p_I{}'$ in the image, we trace the associated ray through N towards the horizontal plane. The intersection of the ray with the horizontal plane is the result of the inverse perspective mapping applied to the image point $p_I{}'$. If we compose both perspective and inverse perspective, the horizontal plane is mapped onto itself, while elevated parts of the scene appear distorted.

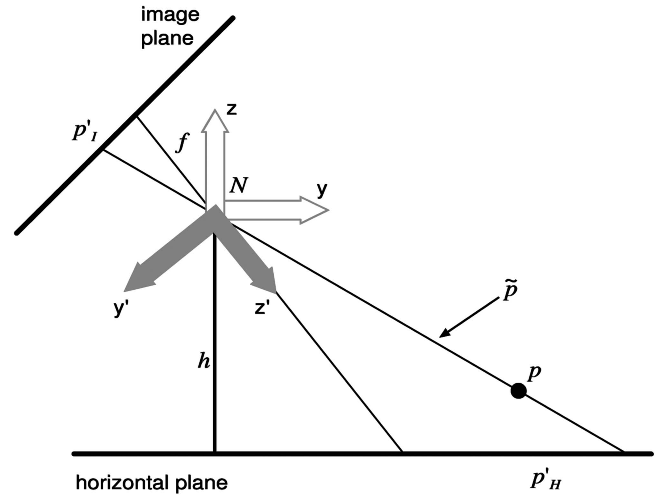

Fig. 7. Geometry of perspective mapping.

Assuming a flat road, Zhao and Yuta [55] used stereo vision to predict the image seen from the right camera, given the left image, using IPM. Specifically, they used IPM to transform every point in the left image to world coordinates, and reprojected them back onto the right image, which were then compared against the actual right image. In this way, they were able to find contours of objects above the ground plane. Instead of warping the right image onto the left image, Bertozzi and Broggi [16], [56] computed the IPM of both the right and left images. Then, they took the difference between the two remapped left and right images. Due to the flat-road assumption, anything elevating out from the road was detected by looking for large clusters of nonzero pixels in the difference image. In the ideal case, the difference image contains two triangles for each obstacle that correspond to the left and right boundaries of the obstacle (see Fig. 8e). This is because, except for those pixels on the left and right boundaries of the obstacle, all other pixels are the same in the left and right remapped images. Locating those triangles, however, was very difficult due to texture, irregular shape, and nonhomogeneous brightness of obstacles. To deal with these issues, they used a polar histogram to detect the triangles. Given a point on the road plane, the polar histogram was computed by scanning the difference image and counting the number of over threshold pixels for every straight line originating from that point. Knoeppel et al. [57] clustered the elevated 3D points based on their distance from the ground plane to generate hypotheses. Each hypothesis was tracked over time and further verified using Kalman filters. This system assumed that the dynamic behavior of the host vehicle was known, and the path information was stored in a dynamic map. The system was able to detect vehicles up to 150 m under normal daytime weather conditions.

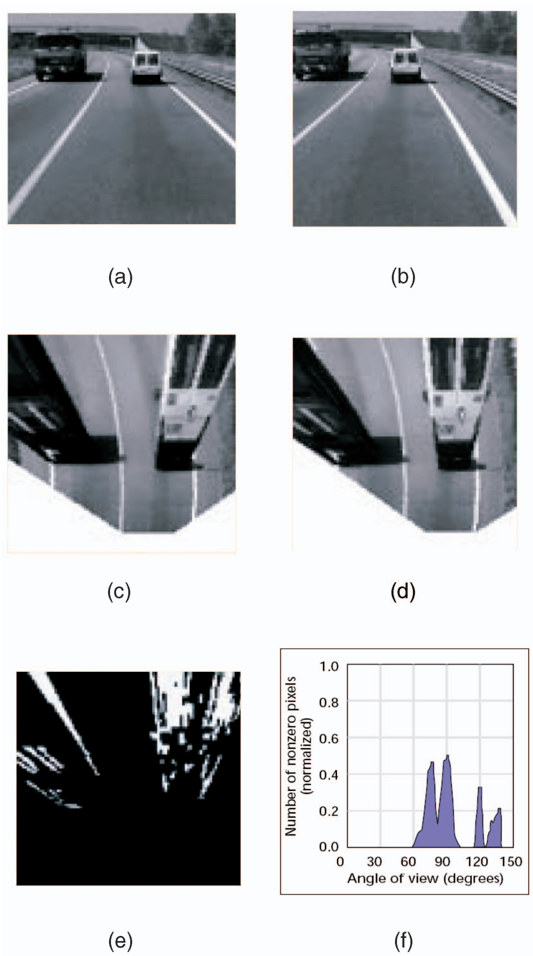

Fig. 8. Obstacle detection: (a) left and (b) right stereo images, (c) and (d) the remapped images, (e) the difference image, and (f) corresponding polar histogram (from [56]).

Although only two cameras are required to find the range and elevated pixels in an image, there are several advantages to use more than two cameras [58]: 1) repeating texture can confuse a two cameras system by causing matching ambiguities, which can be eliminated when additional cameras are present and 2) shorter baseline systems are less prone to matching errors while longer baseline systems are more accurate. The combination is better than either one alone. Williamson and Thorpe [59] investigated a trinocular system. The trinocular rig was mounted on top of a vehicle with the longest baseline being 1.2 meters. The third camera was displaced 50 cm horizontally and 30 cm vertically to provide a short baseline. The system reported a capacity of detecting objects as small as 14 cm at range in excess of 100 m. Due to the additional computational costs, however, binocular system is more preferred in the driver assistance system.

5.3 Motion-Based Methods

All the cues discussed so far use spatial features to distinguish between vehicles and background. Another cue that can be employed is relative motion obtained via the calculation of optical flow. Let us represent image intensity at location (x, y) at time t by $E(x, y, t)$. Pixels on the images appear to be moving due to the relative motion between the sensor and the scene. The vector field $\vec{o}(x, y)$ of this motion is referred to as optical flow.

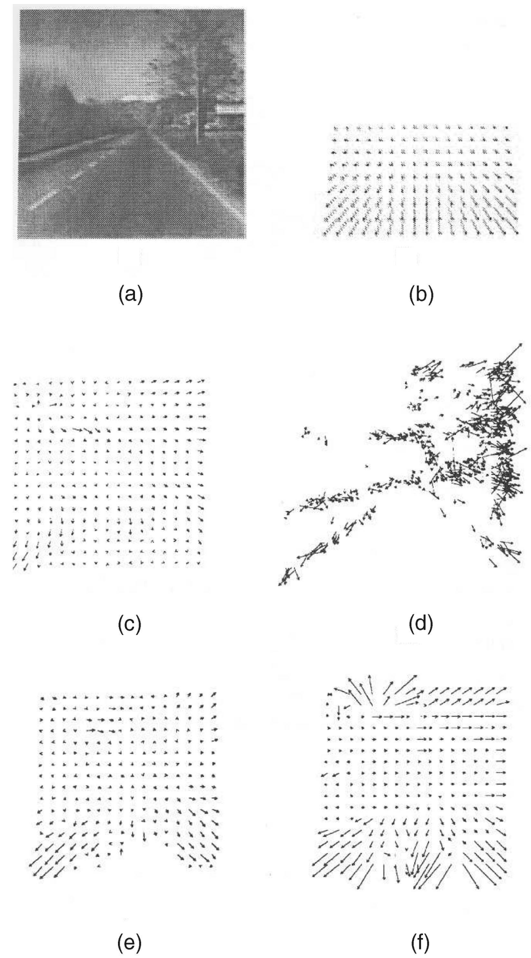

Fig. 9. Comparison of optical flows computed with different algorithms: (a) a frame of image sequence, (b) the theoretical optical flow expected from a pure translation over a flat surface, (c) optical flow from first-order derivative method, (d) optical flow using second-order derivatives the remapped images, (e) optical flow using multiscale differential technique, and (f) optical flow computed with correlation technique (from [60]).

Optical flow can provide strong information for HG. Approaching vehicles at an opposite direction produce a diverging flow, which can be quantitatively distinguished from the flow caused by the car ego-motion [60]. On the other hand, departing or overtaking vehicles produce a converging flow. To take advantage of these observations in obstacle detection, the image is first subdivided into small subimages and an average speed is estimated in every subimage. Subimages with a large speed difference from the global speed estimation are labeled as possible obstacles.

The performance of several methods for recovering optical flow $\vec{o}(x, y)$ from the intensity $E(x, y, t)$ have been compared in [61] using some selected image sequences from (mostly fixed) cameras (see Fig. 9). Most of these methods compute temporal and spatial derivatives of the intensity profiles and, therefore, are referred to as differential techniques. Getting a reliable dense optical flow estimate under a moving-camera scenario is not an easy task. Giachetti et al. [60] developed some of the best first-order and second-order differential methods in the literature and applied them to a typical image sequence taken from a moving vehicle along a flat and straight road. In particular, they managed to remap the corresponding points between two consecutive frames, by minimizing the following distance measure:

$$\sum_{i=-n}^{n}\sum_{j=-n}^{n}\Big[E(i+x',j+y',t')-E(i+x,j+y,t)\Big]^2, \qquad (1)$$

where (x', y') and (x, y) are two corresponding points at time t' and t. The size of the search window was $n \times n$. Since adjusting the corresponding pairs for each of the points was quite expensive, they employed a less dense grid to reduce computational cost.

Kruger et al. [62] estimated optical flow from spatio-temporal derivatives of the gray value images using a local approach. They further clustered the estimated optical flow to eliminate outliers. Assuming a calibrated camera and known ego-motion, they detected both moving and stationary objects. Generating a displacement vector for each pixel (i.e., dense optical flow) is time consuming and also impractical for a real-time system. In contrast to dense optical flow, "sparse optical flow" is less time consuming by utilizing image features, such as corners [63], [64], local minima and maxima [65], or "Color Blobs" [66]. Although it can only produce a sparse flow, feature based methods can provide sufficient information for HG. Moreover, in contrast to pixel-based optical flow estimation methods where pixels are processed independently, feature-based methods utilize high-level information. Consequently, they are less sensitive to noise.

6 HV METHODS

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. Approaches to HV can be classified mainly into two categories: 1) *template-based* and 2) *appearance-based*. Template-based methods use predefined patterns from the vehicle class and perform correlation. Appearance-based methods, on the other hand, 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 nonvehicle class is also modeled to improve the performance. Each training image is represented by a set of local or global features. Then, the decision boundary between the vehicle and nonvehicle classes is learned either by training a classifier (e.g., NNs, Support Vector Machines (SVMs)) or by modeling the probability distribution of the features in each class (e.g., using the Bayes rule assuming a Gaussian distribution).

6.1 Template-Based Methods

Template-based methods use predefined patterns of the vehicle class and perform correlation between the image and the template. Some of the templates reported in the literature represent the vehicle class "loosely," while others are more detailed. It should be mentioned that, due to the nature of the template matching methods, most papers in the literature do not report quantitative results and demonstrate performance through examples.

Parodi and Piccioli [45] proposed a hypothesis verification scheme based on the presence of license plates and rear windows. This can be considered as a loose template of the vehicle class. No quantitative performance was include in the paper. Handmann et al. [44] proposed a template based on the observation that the rear/frontal view of a vehicle has a "**U**" shape (i.e., one horizontal edge, two vertical edges, and two corners connecting the horizontal and vertical edges). During verification, they considered a vehicle to be present in the image if they could find the "**U**" shape.

Ito et al. [67] used a very loose template to recognize vehicles. Using active sensors for HG, they checked whether or not pronounced vertical/horizontal edges and symmetry existed. Due to the simplicity of the template, they did not expect very accurate results, which was the main reason for employing active sensors for HG. Regensburger et al. [68] utilized a template similar to [67]. They argued that the visual appearance of an object depends on its distance from the camera. Consequently, they used two slightly different generic object (vehicle) models, one for nearby objects and another for distant objects. This method, however, raises the question of what model to use in a specific location. Instead of working with different generic models, distance-dependent subsampling was performed before the verification step in [69].

A template, called "moving edge closure," was used in [52] which was fit to groups of moving points. To get the moving edge closure, they performed edge detection on the area covered by the detected moving points, followed by the external edge connection. If the size of the moving edge closure was within a predefined range, they claimed vehicle detected. Nighttime vehicle detection was also addressed in this work [52]. Basically, pairs of headlights were considered as templates for vehicle detection.

A rather loose template was also used in [70], where hypotheses were generated on the basis of road position and perspective constraints. The template contained a priori knowledge about vehicles: "A vehicle is generally symmetric, characterized by a rectangular bounding box which satisfies specific aspect ratio constraints." The model matching worked as follows: Initially, the hypothesized region was checked for the presence of two corners representing the bottom of the bounding box, similar to the "**U**" shape idea in [44]. The presence of corners was validated using perspective and size constraints. Then they detected the top part of the bounding box in a specific region determined, once again, by perspective and size constraints. Once the bounding box was detected successfully, they claimed vehicle presence in that region. This template could be very fast, however, it introduces some uncertainties, given that there might be other objects on the road satisfying those constraints (e.g., distant buildings).

6.2 Appearance Methods

HV using appearance models is treated as a two-class pattern classification problem: vehicle versus nonvehicle. Building a robust pattern classification system involves searching for an optimum decision boundary between the classes to be categorized. Given the huge within-class variabilities of the vehicle class, we can imagine that this is not an easy task. One feasible approach is to learn the decision boundary based on training a classifier using the feature sets extracted from a training set.

Appearance-based methods learn the characteristics of vehicle appearance from a set of training images which capture the variability in the vehicle class. Usually, the variability of the nonvehicle class is also modeled to improve performance. First, a large number of training images is collected and each training image is represented by a set of local or global features. Then, the decision boundary between the vehicle and nonvehicle classes is learned either by training a classifier or by modeling the probability distribution of the features in each class.

Various feature extraction methods have been investigated in the context of vehicle detection. Based on the method used, the features extracted can be classified as either local or global. Global features are obtained by considering all the pixels in an image. Wu and Zhang [71] used standard Principal Components Analysis (PCA) for feature extraction, together with a nearest-neighbor classifier, reporting an 89 percent accuracy. However, their training database was quite small (93 vehicle images and 134 nonvehicle images), which makes it difficult to draw any useful conclusions.

An inherent problem with global feature extraction approaches is that they are sensitive to local or global image variations (e.g., pose changes, illumination changes, and partial occlusion). Local feature extraction methods on the other hand are less sensitive to these effects. Moreover, geometric information and constraints in the configuration of different local features can be utilized either explicitly or implicitly.

Different from [71], in [42], PCA was used for feature extraction and Neural Networks (NNs) for classification. First, each subimage containing vehicle candidates was scaled to 20×20, then it was subdivided into 25 4×4 subwindows. PCA was applied on every subwindow (i.e., "local PCA") and the output was provided to a NN to verify the hypothesis.

Goerick et al. [43] and Noli et al. [72] used the (*LOC*) method (see Section 5.1.5) to extract edge information. The histogram of *LOC* within the area of interest was then provided to a *NN* classifier, a Bayes classifier and combination of both for classification. For *NN*, the number of nodes in the first layer was between 350-450 while the number of hidden nodes was 10-40. They used 2,000 examples for training and the whole system ran in real-time. The performance of the neural net classifier was 94.7 percent, which is slightly better than their Bayes classifier (94.4 percent), also very close to the combined classifier (95.7 percent).

Kalinke et al. [50] designed two models for vehicle detection: one for sedans and the other for trucks. Two different model generation methods were used. The first one was designed manually, while the second one was based on a statistical algorithm using about 50 typical trucks and sedans. Classification was performed using NNs. The input to the NNs was the Hausdorrf distances between the hypothesized vehicles and the models, both represented in terms of the LOC. The NN classified every input into three classes: sedans, trucks, or background. Similar to [43], Handmann et al. [44] utilized the histogram of LOC, together with a NN, for vehicle detection. The Hausdorrf distance was used for the classification of trucks and cars such as in [50]. No quantitative performance was reported in [44] or [50].

A statistical model of vehicle appearance was investigated by Schneiderman and Kanade [73]. A view-based approach employing multiple detectors was used to cope with viewpoint variations. The statistics of both object and "nonobject" appearance were represented using the product of two histograms with each histogram representing the joint statistics of a subset of Haar wavelet features and their position on the object. A three-level wavelet transform was used to capture the space, frequency, and orientation information. This three-level decomposition produced 10 subbands and 17 subsets of quantized wavelet coefficients were used. Bootstrapping was used to gather the statistics of the nonvehicle class. The best performance reported in [73] was 92 percent. A different statistical model was investigated by Weber et al. [74]. 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. They used 200 images for training and reported an 87 percent accuracy.

An overcomplete dictionary of Haar wavelet features was utilized in [75] for vehicle detection. They argued that this representation provided a richer model and spatial resolution and that it was more suitable for capturing complex patterns. The overcomplete Haar wavelet features were derived from a set of redundant functions, where the wavelets at level n was $1/4 \times 2^n$ instead of 2^n. They referred it to as quadruple density dictionary. A total of 1,032 positive training patterns and 5,166 negative training patterns were used for training and the ROC showed that the false positive rate was close to 1 percent when the detection rate approached to 100 percent.

Sun et al. [76], [49] went one step further by arguing that the actual values of the wavelet coefficients are not very important for vehicle detection. In fact, coefficient magnitudes indicate local oriented intensity differences, information that could be very different even for the same vehicle under different lighting conditions. Following this observation, they proposed using quantized coefficients to improve detection performance. The quantized wavelet features yielded a detection rate of 93.94 percent compared to 91.49 percent using the original wavelet features.

Using Gabor filters for vehicle feature extraction was investigated in [77]. Gabor filters provide a mechanism for obtaining orientation and scale tunable edge and line detectors. Vehicles contain strong edges and lines at different orientation and scales; thus, this type of features are very effective for vehicle detection. The hypothesized vehicle subimages were subdivided into nine overlapping subwindows. Gabor filters were then applied on each subwindow separately. The magnitudes of the responses of the Gabor filters were collected from each subwindow and represented by three moments: the mean μ, the standard deviation σ, and the skewness κ. Classification was performed using Support Vector Machines (SVMs) yielding an accuracy of 94.81 percent.

A "vocabulary" of information-rich vehicle parts was constructed automatically by applying the Forstner interest operator onto a set of representative images, together with a clustering method in [78]. Each image was represented in terms of parts from this vocabulary to form a feature vector, which was used to train a classifier to verify hypotheses. Some successful detections were reported under high degree of clutter and occlusion, and an overall 90.5 percent accuracy was achieved. Following the same idea (i.e., detection using components), Leung [79] investigated a different vehicle detection method. Instead of using the Forstner interest operator, differences of Gaussians were applied onto images in scale space, and maxima and minima were selected as the key-points. At each of the key-points, the Scale Invariant Feature Transform (SIFT) [80] was utilized to form a feature vector, which was used to train a SVM Classifier. Leung tested his algorithm on the UIUC data [78], showing slightly better performance.

7 Integrating Detection with Tracking

Vehicle detection can be improved considerably, both in terms of accuracy and time, by taking advantage of the temporal continuity present in the data. This can be achieved by employing a tracking mechanism to hypothesize the location of vehicles in future frames. Tracking takes advantage of the fact that it is very unlikely for a vehicle to show up only in one frame. Therefore, vehicle location can be hypothesized using past history and a prediction mechanism. When tracking performance drops, common hypothesis generation techniques can be deployed to maintain performance levels.

By examining the reported vehicle detection and tracking algorithms/systems at the structural level, many similarities can be found. Specifically, the majority of existing on-road vehicle detection and tracking systems use a *detect-then-track* approach (i.e., vehicles are first detected and then turned over to the tracker). This approach aims to resolve detection and tracking sequentially and separately. There are many examples in the literature following this strategy. In Ferryman et al. [81], vehicle detection is based on template matching [82] while tracking uses dynamic filtering. In that work, high order statistics were used for detection and a Euclidean-distance-based correlation was employed for tracking. In [47], vehicles were tracked using multiple cues such as intensity and edge data. To increase sensor range for vehicle tracking, Clady et al. employed an additional P/T/Z camera [83]. In [84], close to real time performance was reported (i.e., 14 frames per second) by integrating detection with tracking based on deformable models. This approach has several drawbacks. First, false detections will be passed to the tracker without a chance of rectification. Second, tracking templates from imperfect detections will jeopardize the reliability of trackers. Most importantly, this type of approaches do not exploit temporal information in detection.

There exist several exceptions, where temporal information has been incorporated into detection. Betke et al. [85], [46] have realized that reliable detection from one or two images is very difficult and it only works robustly under cooperative conditions. Therefore, they used a refined search within the tracking window to re-enforce the detections (i.e., a car template was created online every 10th frame and was correlated with the object in the tracking windows). Similar to [46], temporal tracking was used to suppress false detections in [86], where only two successive frames were employed. Similar observations were made by Hoffman et al. [87] (i.e., detection quality was improved by accumulating feature information over time).

Temporal information has not been fully exploited yet in the literature. Several efforts have been reported in [88] and more recently in [89]. We envision a different strategy (i.e., *detect-and-track*), where detection and tracking are addressed simultaneously in a unified framework (i.e., detection results trigger tracking, and tracking reenforces detection by accumulating temporal information through some probabilistic models). Approaches following this framework would have better chances to filter out false detections in subsequent frames. In addition, tracking template updates would be achieved through repeated detection verifications.

8 DISCUSSION

On-road vehicle detection is so challenging, that none of the methods reviewed can solve it alone completely. Different methods need to be undertaken and selected based on the prevailed conditions faced by the system [44], [90]. Complementary sensors and algorithms should be used to improve overall robustness and reliability. In general, surrounding vehicles can be classified into three categories according to their relative position to the host vehicle: 1) overtaking vehicles, 2) midrange/distant vehicles, and 3) close-by vehicles (see Fig. 10). In close-by regions (A1), we may only see part of the vehicle. In this case, there is no free space in the captured images, which makes the shadow/symmetry/edge-based methods inappropriate. In the overtaking regions (A2), only the side view of the vehicle is visible while appearance changes fast. Methods detecting vehicles in these regions might be better to employ motion information or dramatic intensity changes [85], [46]. Detecting vehicles in the midrange/distant region (A3) is relatively easier since the full view of a vehicle is available and appearance is more stable.

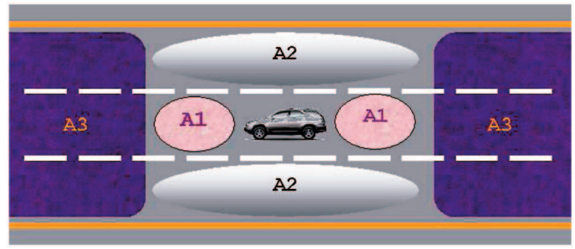

Fig. 10. Detecting vehicles in different regions requires different methods. A1: Close by regions. A2: Overtaking regions. A3: Midrange/distant regions.

Next, we provide a critique of the HG and HV methods reviewed in the previous sections. Our purpose is to emphasize their main strengths and weaknesses as well as to present potential solutions reported in the literature for enhancing their performance for deployment in real settings. The emphasis is on making these methods more reliable and robust to deal with the challenging conditions encountered in traffic scenes. Additional issues are discussed in Section 9.

8.1 Critique of Knowledge-Based HG Methods

Systems employing local symmetry, corners, or texture information for HG are most effective in relatively simple environments with no or little clutter. Employing these cues in complex environments (e.g., when driving downtown where the background contains many buildings and different textures), would introduce many false positives. In the case of symmetry, it is also imperative to have a rough estimate of the vehicle's location in the image for fast and accurate symmetry computations. Even when utilizing both intensity and edge information, symmetry is quite prone to false detections, such as symmetrical background objects, or partly occluded vehicles.

Color information has not been deployed extensively for HG due to the inherent difficulties of color-based object detection in outdoor settings. In general, the color of an object depends on illumination, reflectance properties of the object, viewing geometry, and sensor parameters. Consequently, the apparent color of an object can be quite different during different times of the day, under different weather conditions, and under different poses.

Employing shadow information and vehicle lights for HG have been exploited in a limited number of studies. Under perfect weather conditions, HG using shadow information can be very successful. However, bad weather conditions (i.e., rain, snow, etc.) or bad illumination conditions make road pixels quite dark, causing this method to fail. Vehicle lights is a ubiquitous vehicle feature at night; however, it could be confused with traffic lights and background lights. We believe that these cues have limited employability.

Utilizing horizontal and vertical edges for HG is probably the most promising, knowledge-based, approach reported in the literature. Our experience with using edge information in realistic experiments has been very positive [49]. Although this method can suffer from false positives, many of them can be rejected quickly using simple tests (e.g., aspect ratio). From a practical point of view, there are fast implementations of edge detection in hardware making this approach even more attractive. The main problem with this approach is that it depends on a number of parameters that could 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. We have described in Section 5 a multiresolution scheme addressing these issues.

8.2 Critique of Stereo-Based HG Methods

Stereo-based methods have been employed extensively for HG, however, traditional implementations are time consuming and work well only if the camera parameters have been estimated accurately. As a result, their performance is significantly impaired. Using stereo vision to hypothesize vehicle location, dense disparity maps are necessary to guarantee that all regions are searched for potential vehicles. Naive approaches to stereo computation are not suitable for dynamic object detection at reasonable vehicle speed due to the high complexity (i.e., $O(dm^2n^2)$, where d is the number of shifts over which the correspondence search is performed, m is the size of the support window, and n is the size of the images). There have been several approaches to overcome this problem, such as, computing sparse disparity maps [90], [13], employing multiresolution schemes [90], [13], or using prior knowledge about the environment to limit the search for correspondences [53].

Estimating the stereo parameters accurately is also hard to guarantee in an on-road scenario. Since the stereo rig is on a moving vehicle, vibrations from the vehicle's motion and windy conditions might shift the cameras, while the height of the cameras keeps changing due to the vehicle's suspension. Suwa et al. [91] proposed a method to update the parameters and compensate for errors caused by camera movements. The 3D measurements of a stereo-based system are calculated using:

$$M_w = R_c M_c + t_c, \tag{2}$$

where M_w and M_c represent vectors in the world coordinate and camera coordinate systems, R_c is a rotation matrix, and t_c is a translation vector. A two-parameter sway model was used in [91]: the sway direction angle and the sway range. Incorporating the effect of sway parameters leads to a modified model:

$$M_w = R_\beta R_\alpha (R_{-\beta} R_\gamma M_c + t_c) + t_\epsilon, \tag{3}$$

where $\alpha = -2\sin^{-1}(\frac{d}{2H})$, d is the sway range, H is the height of the camera, β denotes the sway direction angle, and γ the set up tilt angle. The two sway parameters were estimated from corresponding pairs with sway and without sway. The image data without sway was assumed to have been obtained from the no sway image while the sway data was obtained by using correlation. Estimation was done using least squares.

Bertozzi et al. [92] have also analyzed the parameter drifts and argued that vibrations affect, mostly, extrinsic parameters, and not the intrinsic parameters. A fast self-calibration method was considered to deal with this issue. Eight carefully designed markers were put on the hood, four for each of the two cameras. Since the world coordinates of the markers were known, the determination of their image coordinates was sufficient to compute the position and orientation of the cameras in the same reference system.

8.3 Critique of Motion-Based HG Methods

In general, motion-based methods can detect objects based on relative motion information. Obviously, this is a major limitation, for example, this method cannot be used to detect static obstacles, which can represent a big threat. Despite this fact, employing motion information for HG has shown promising results; however, it is computationally intensive while its performance is affected by several factors. Generating a displacement vector for each pixel (continuous approach) is time-consuming and impractical for a real-time system. In contrast, discrete methods based on image features such as color blobs [66] or local intensity minima and maxima [65] has shown good performance while being faster. There have been also attempts to speed up motion-based computations using multiresolution schemes [90]. Several factors affect the computation of motion information [60] including:

- *Displacements between consecutive frames*. Fast movement of the host vehicles causes significant pixel displacements. Points in the image can move by more than five pixels, when the car moves at a speed faster than 30 km/h. Consequently, aliasing in the computation of the temporal derivatives introduces errors into the computation of optical flow.
- *Lack of textures*. Large portions in the images represent the road bed, where gray-level variations are quite small, especially when driving the vehicle in a country road. Significant instability can be introduced to the computation of the spatial derivatives due to texture insufficiency.
- *Shocks and vibrations*. Image motion is the sum of a smooth component due to the car ego-motion and a high frequency component due to the camera shocks and vibrations. In the presence of shocks and vibrations, caused by mechanical instability of the camera, a high frequency noise is introduced to the intensity profile. This noise gets greatly amplified during the computation of the temporal derivatives. In general, error introduced by shocks and vibrations is small if the camera is mounted on high quality antivibrating platforms and the vehicle is moving along usual roads. However, if the camera is mounted less carefully or the vehicle is driven on a bumpy road, the error can be 10 times larger.

Among these factors, camera movement is the main reason that traditional differential methods fail. If we can counterbalance camera movements, then these methods could become very useful. This is the objective of another research direction, called "image stabilization" [93], [94]. Image stabilization is based on frame-to-frame registration. Taking the first frame of the image sequence as a reference, the stabilization method registers this frame to the next frame, computes the motion parameters from the current frame to the reference frame. Then, it uses the estimated parameters to warp the current frame to get the stabilized image, which can be considered as taken from a stationary camera. The motion model employed in [93] contains four parameters: two for translation, one for rotation, and one for scaling:

$$\begin{pmatrix} X_2 \\ Y_2 \end{pmatrix} = s \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} X_1 \\ Y_1 \end{pmatrix} + \begin{pmatrix} \Delta X_2 \\ \Delta Y_2 \end{pmatrix}, \tag{4}$$

where $\begin{pmatrix} X_i \\ Y_i \end{pmatrix}$ are the image frame coordinates at time t_i, $\begin{pmatrix} \Delta X_2 \\ \Delta Y_2 \end{pmatrix}$ is the translation measured in the image coordinate system of a frame at t_2, θ is the rotation angle between the two frames, and s is a scaling factor.

This model was appropriate for image sequence of distant scenes, where perspective distortion could be neglected. The motion parameters were computed by matching a small number of feature points between two frames. The extraction of feature points was done via searching a predefined small window on the very top region of the image using correlation. By constraining the selection of feature points to the very top region, features that were too close to the camera could be avoided and, consequently, less distortion was introduced to the model. It should be mentioned that image stabilization methods would fail when an image contains close scenes—a common scenario when driving a vehicle in downtown or during vehicle turns.

8.4 Critique of HV Methods

Constructing explicit models of the objects to be recognized is very difficult when object appearance varies a lot. In general, appearance-based methods are more accurate than template-based methods, however, they are more costly due to classifier training. Nevertheless, appearance-based methods are becoming more and more popular due to the exponential growth in processor speed. Analyzing the pros and cons of various appearance-based methods proposed in the literature is not simple. Most studies have been performed using different data sets and performance measures making a fair evaluation of different feature extraction methods and classification schemes difficult if not impossible. In a recent study, experimental results were reported using several feature extraction methods (i.e., PCA features, wavelet features, and Gabor features) and classifiers (i.e., Neural Networks (NN) and Support Vector Machines (SVMs)) [95]). Testing was performed using a common data set obtained by driving Ford's concept vehicle under different traffic conditions (e.g., structured highway, complex urban streets, and varying weather conditions). The best approach in terms of accuracy was found to be Gabor features with SVMs, yielding an error rate of 5.33 percent with a false positives (FP) rate of 3.46 percent and a false negatives (FN) rate of 1.88 percent. Combining Gabor and Harr wavelet features yielded a slightly better performance (i.e., an error rate of 3.89 percent with a FP rate of 2.29 percent and a FN rate of 1.6 percent) at the expense of higher time requirements. It should be mentioned that the error rate using Haar wavelet features with SVMs was 8.52 percent, the FP rate was 6.50 percent and the FN rate was 2.02 percent. More systematic evaluations of various feature extraction methods and classification schemes are required in order to assess the performance of HV methods. In order for these comparisons to become more meaningful, it is imperative to develop first representative datasets (i.e., benchmarks) and carefully designed evaluation procedures (i.e., see Section 9.6).

9 Challenges Ahead

An important issue in the realization of successful driver assistance applications is the design of vehicle detection and tracking systems that yield a maximum level of reliability and robustness in real-time. Although many efforts have been put into this research area, many algorithms and systems have already been reported, many prototype vehicles have already been demonstrated, a highly robust, reliable, real-time system is yet to be demonstrated. Achieving these objectives requires addressing several challenges and solving quite different problems.

From a technical point of view, the success of an on-road vehicle detection system will depend on the number of correct detections versus the number of false alarms that it produces, assuming a certain processing rate and a processor platform. Determining the desired level of accuracy for vehicle detection is not easy and depends on the nature of the application. For example, if vehicle detection is part of a warning system, then higher false positive rates can be tolerated. In contrast, systems involving active vehicle control need to be more conservative in terms of false alarms. There are a number of ways to significantly reduce the number of false positives while keeping high accuracy including improved algorithmic solutions (e.g., using multiple cues, advanced statistical, and learning models), sensor fusion (e.g., visible, IR, and radar), and telematics (e.g., vehicle-to-vehicle communication and GPS-based localization). We elaborate more on these issues next.

9.1 Algorithmic Advances

The design of computer vision algorithms that operate robustly and reliably in complex and wide varying environments (e.g., rain, fog, night, etc.) is a major challenge. Using on-board cameras makes some well-established computer vision techniques unsuitable (e.g., background subtraction is not appropriate due to fast background changes caused by camera motion) or not directly applicable unless making certain assumptions or adding enhancements (e.g., stereo-based systems require frequent recalibration to account for camera movements caused by shocks and vibrations). Efficient implementations should also be considered (e.g., fast motion-based estimation) in order to meet real-time performance requirements.

Developing more powerful algorithms to deal with a variety of issues is thus essential. In doing so, it is important to understand first the requirements of on-road vehicle detection and design customized algorithmic solutions that meet the requirements while taking advantage of domain knowledge and inherent constraints (e.g., exploiting temporal continuity to improve accuracy and robustness or assuming a flat road to simplify the mapping between image pixels and word coordinates). We have presented in Section 8 a number of issues associated with HG approaches and potential algorithmic solutions to deal with these issues effectively. More efforts are clearly required in this direction.

In HV, most research efforts have focused on feature extraction and classification based on learning and statistical models. Efforts in this direction should continue while capitalizing on recent advances in the statistical and machine learning areas. For example, one issue that has not been given enough attention in the vehicle detection literature is the issue of selecting a good set of features. In most cases, a large number of features is employed to compensate for the fact that relevant features are unknown a priori. However, without employing some kind of feature selection strategy, many of them would be either redundant or even irrelevant which could affect classification accuracy seriously. In general, it is highly desirable to use only those features that have great separability power while ignoring or paying less attention to the rest. For instance, to allow a vehicle detector to generalize nicely, it would be nice to exclude features encoding fine details which might be present in some vehicles only. Finding out what feature to use for classification/recognition is referred to as feature selection. Recently, a few efforts have been reported in the literature addressing this issue in the context of vehicle detection [76], [96], [97], [98]. Several efforts have even been reported to improve tracking through feature selection [99]. We believe that more efforts are required in this direction along with efforts to develop more powerful feature extraction and classification schemes. Recent advances in machine learning and statistics (e.g., kernel methods [100] should be leveraged in this respect).

Combining multiple cues should also be explored more actively as a viable means to develop more reliable and robust systems. The main motivation is that the use of a single cue suitable for all conceivable scenarios seems to be impossible. Combining different cues has produced promising results (e.g., combining LOC, entropy, and shadow [44], shape, symmetry, and shadow [101], color and shape [102], and motion with appearance [103]). Effective fusion mechanisms as well as cues that are fast and easy to compute are important research issues.

9.2 Sensor Advances

Employing more powerful sensors in vehicle detection applications can influence system performance considerably. Specific objectives include improving dynamic range, spectral sensitivity, spatial resolution, and incorporating computational capabilities.

Traditional CCD cameras lack the dynamic range necessary to operate in traffic under adverse lighting conditions. Cameras with enhanced dynamic range are needed to enable daytime and nighttime operation without blooming. An example is Ford's proprietary low-light camera which has been developed jointly between Ford Research Laboratory and SENTECH. It uses a Sony x-view CCD array with specifically designed electronic profiles to enhance the camera's dynamic range. Fig. 11a and Fig. 11c show the dynamic range of the low light camera, while Fig. 11b and Fig. 11d show the same scene images caught under same illumination conditions by using a normal camera. The low-light camera has been employed in a number of studies including [77], [76], [104], [49], [96], [97]. Recently, several efforts have focused on using CMOS technology to design cameras with improved dynamic range.

Low-light cameras do not extend visual capabilities beyond the visible spectrum. In contrast, Infrared (IR) sensors allow us to sense important information in the nonvisible spectrum. IR-based systems are less sensitive to adverse weather or illumination changes—day and night snapshots of the same scene are more similar to each other. Several studies have been carried out to evaluate the feasibility and advantages of using IR for driver assistance system [105], [106], [107], [108]. An interesting example is the miniaturized optical range camera developed in the project *MINORA* [109], [110]. It works in the near-IR, it is cheap, fast, and capable of providing 3D information with high accuracy. However, it has certain limitations such as low resolution and narrow field of view. Fusing several sensors together could offer considerable performance improvements (see Section 9.3).

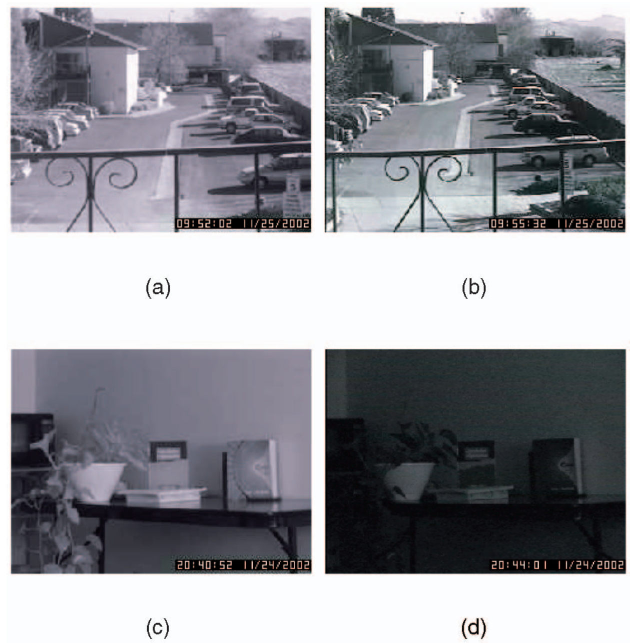

Fig. 11. Low light camera versus normal camera. (a) Low-light camera daytime image. (b) Same scene caught using normal camera. (c) Low-light camera nighttime scene. (d) Same nighttime scene caught using normal camera.

Improving camera resolution can offer significant benefits too. Over the last few years, the resolution of sensors has been drastically enhanced. A critical issue in this case is decreasing acquisition and transfer time. CMOS technology holds some potential in this direction (i.e., pixels can be addressed independently like in traditional memories).

In conventional vision systems, data processing takes place at a host computer. Building cameras with internal processing power (i.e., vision chip) is an important goal. The main idea is integrating photo-detectors with processors on a very large scale integration [111]. Vision chips have many advantages over conventional vision systems, for instance, high speed, small size, and lower power consumption, as well as a wide brightness range, etc. Several cameras available today allow to address and solve some basic problems directly at the sensor level (e.g., image stabilization can now be performed during image acquisition).

9.3 Sensor Fusion

Developing driver assistance system suitable for urban areas where traffic signs, crossings, traffic jams, and other participants (motorbikes, bicycles, pedestrians, or even live stocks) may exist poses extra challenges. Exclusively

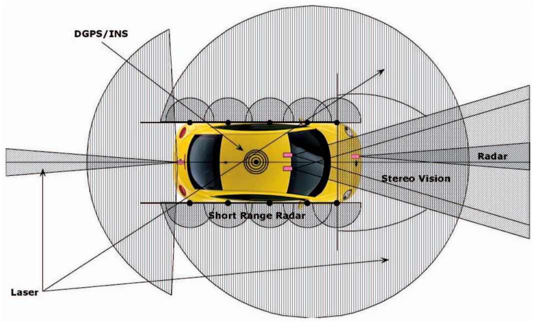

Fig. 12. An example of sensor fusion.

vision-based systems and algorithms are not yet powerful enough to deal with complex traffic situations. To extend the application of a driver assistance systems, substantial research efforts are required to develop systems employing information from multiple sensors, both active and passive, effectively (see Fig. 12).

Sensor characteristics reveal that each sensor can only perceive certain characteristics of the environment, therefore, a single sensor is not sufficient enough to comprehensively represent the driving environment [67], [37]. A multisensor approach has the potential to yield a higher level of reliability and security. Methods for sensor fusion and integration are concerned with improving the sensing capacity by using redundant and complementary information from multiple sensors. These sensors are able to obtain more accurate environment features that are impossible to perceive with a single sensor.

For example, acoustic sensors were fused with video sensors in [29] for both detection and tracking in order to take advantage of the complementary information available in the two sensors. In another study, a multisensor approach was adopted using sensor technologies with widely overlapping fields of view between different sensors [112]. Depending on the relevance of the area covered, they increased the degree of vehicle is surveyed by means of a single laser sensor, the sides are each covered by two independent laser scanners and several overlapping short range radar sensors, and the front of the car is covered by three powerful long-range sensors (i.e., stereo-vision, laser, and radar). The sensor signals are combined by sensor fusion into a joint obstacle map. By considering confidence and reliability measures for each sensor, the obstacle map computed by sensor fusion was shown to be more precise and reliable than any of the individual sensor outputs themselves.

Although sensor fusion has great potential to improve driver assistance system, developing the actual multisensor platform requires dealing with a series of problems including not only the conventional issues of sensor fusion and integration, but also some special issues in driver assistance system design. With the common geometry and time frames, sensor fusion needs to be implemented at the various levels:

- *Registration level*. To allow fusing the data from different sensors effectively, sensor data needs to be registered first.
- *Encapsulation level*. Registered data from different sensors can be fused to yield more accurate information for the detected vehicles based on the reliability/confidence levels of the attributes associated with different sensors. For instance, a more accurate position-velocity could be obtained by analyzing the registered radar and stereo-vision data. In other words, at this level, same type of information is encapsulated together to a more accurate and concise representation.
- *Perception-map level*. Complementary information can be fused to infer new knowledge about the driving environment. The position-velocity information of detected vehicles and road geometry information (from vision) can be fused to produce a primary perception map, where vehicles can be characterized as being either stationary/moving or inside/outside the lane.
- *Threat quantification level*. Vehicle type, shape, distance, and speed information can be fused to quantify the threat level of a vehicle in the perception map to the host vehicle.

Most vehicle detection approaches have been implemented as "autonomous systems" with all instrumentation and intelligence on-board the vehicle. Significant performance improvements can be expected, however, by implementing vehicle detection as "co-operative systems" where assistance is provided from external sources (i.e., the roadway, or from other vehicles, or both). Examples of roadway assistance include passive reference markers in the infrastructure and GPS-based localization. Vehicle-to-vehicle co-operation works by transmitting key vehicle parameters and intentions to close-by vehicles. Having this information available as well as knowing the type of surrounding environment through GPS might reduce the complexity of the problem and make vehicle detection more reliable and robust.

9.4 Software Issues

Vision-based vehicle detection systems should be modular, reconfigurable, and extensible to be able to deal with a wide variety of image processing tasks. The functionality of most vehicle detection systems today is achieved by a few algorithms that are hard-wired together. This is quite inefficient and cannot handle satisfactorily the complexities involved. Recently, there have been some efforts to develop a software architecture that can deal with different levels of abstraction including sensor fusion, integration of various algorithms, economical use of resources, scalability, and distributed computing. For example, a multiagent-system approach was proposed in [13] (i.e., ANTS or Agent NeTwork System) to address these issues. In another study [85], a hard real-time operating system called "Maruti" was used to guarantee that the timing constraints on the various vision processes were satisfied. The dynamic creation and termination of tracking processes optimized the amount of computational resources spent and allowed fast detection and tracking of multiple cars. Obviously, more efforts in this area would be essential.

9.5 Hardware Issues

On-board vehicle detection systems have high computational requirements as they need to process the acquired images at real-time or close to real-time to save time for driver reaction. For nontrivial velocities of the vehicle, processing latency should be small (i.e., typically no larger than 100 ms), while processing frequency should be high (i.e., typical in excess of 15 frames per second). Due to the constrains of low latency and the difficulty in sending and receiving video data reliably, most image processing must be done no site on the vehicle.

Computer vision algorithms are generally very computationally intensive and require powerful resources in order to comply with the real-time performance constraints. With increasing computing power of standard PCs, several systems have been demonstrated using general purpose hardware. For example, our group has developed a vehicle detection system that works at a frame rate of approximately 10 frame per second (NTSC: processing on average every third frame) using a standard PC machine (Pentium III 1133MHZ) [49]. Although we expect the development of more powerful, low-cost, general-purpose processors in the near future, specialized hardware solutions using off-the-self components (e.g., cluster of PCs) seems to be the way to go at present.

Vehicle detection for precrash sensing requires high enough sampling rate in order to provide a satisfactory solution. If the vehicle's speed is about 70 mph, then 10Hz corresponds to a 3 meter interval. The most time consuming step in our system is the computation of the vertical/horizontal edges. Most low-level image-processing algorithms employed for vehicle detection perform similar computations for all the pixels of an image and require only local information. Therefore, substantial speed-ups can be achieved by implementing them on appropriate hardware. Specialized hardware solutions are possible using low-cost general-purpose processors and Field Programmable Gate Arrays (FPGAs).

Recent advances in computation hardware allow us to have systems that can deliver high computational power, with fast networking facilities, at an affordable price. Several studies have taken advantage of hardware implementations to speed-up computations including edge-based motion detection [113], hardware-based optical flow estimation [114], object tracking [115], as well as feature detection and point tracking [116]. Sarnoff has also developed a powerful image processing platform called VFE-200 [117]. VFE-200 can perform several front-end vision functions in hardware simultaneously at video rates (e.g., pyramids, registration, and optical flow). It is worth mentioning that, most of the hardware implementations appeared in the literature have addressed smaller problems (e.g., motion detection, edge detection, etc.). Integrating all those hardware components together, as well as integrating hardware and software implementations seamlessly, requires more effort.

9.6 Benchmarks and Evaluation Methodology

The majority of vehicle detection systems reported in the literature have not been tested under realistic conditions (e.g., different traffic scenarios including simply structured highway, complex urban street, and varying weather conditions). Moreover, evaluations are based on different data sets and performance measures, making comparisons between systems very difficult. Future efforts should focus on assessing system performance along a real collision timeline, taking into account driver perception-response times, braking rates, and various collision scenarios.

The field is lacking representative data sets (i.e., benchmarks) and specific procedures to allow comprehensive system evaluations and fair comparisons between different system. To move things forward, exemplary strategies developed in related fields (e.g., face recognition [118], and surveillance [119]) should be adapted in order to develop and make available to the broader scientific community benchmarks and carefully designed evaluation procedures to enable performance evaluations in a consistent way. Relating level of performance in terms of complexity of driving scene is also of critical importance. Ideas developed in related fields (e.g., object recognition [120]) should be adapted to allow more effective designs and meaningful evaluations.

9.7 Failure Detection

An on-board vision sensor will face adverse operating conditions, and it may reach a point where it might not be able to provide good quality data to meet minimum system performance requirements. In these cases, the driver assistance system may not be able to fulfill its desired responsibilities correctly (e.g., issuing severe false alerts). A reliable driver assistance system should be able to evaluate its performance and disable its operation when it cannot provide reliable traffic information any more. We refer to this function as "failure detection." One possible option for failure detection is to use another sensor exclusively for this purpose, at the expense of additional cost. A better method might be extracting information for failure detection from the vision sensor. Some preliminary experiments have been reported in the scenario of distance detection using stereo vision [121], where the host vehicle and subject were both stationary. Further exploration of this issue is yet to be carried out.

10 Conclusions

We have presented a survey of vision-based on-road vehicle detection systems—one of the most important components of any driver assistance system. On-road vehicle detection using optical sensors is very challenging and many practical issues must be considered. Depending on the range of interest, different methods seem to be more appropriate. In HG, stereo-based methods have gained popularity but they suffer from a number of practical issues not found in typical applications. Edge-based methods, although much simpler, are quite effective but they are not appropriate for distant vehicles. In HV, appearance-based methods are more promising but recent advances in machine and statistical learning need to be leveraged. Fusing data from multiple cues and sensors should be explored more actively in order to improve robustness and reliability. A great deal of work should also be directed toward the enhancement of sensor capabilities and performance including the improvement of gain control and sensitivity in extreme illumination conditions. Hardware-based solutions using off-the-self components should also be explored to meet real-time constraints while keeping cost low.

Although we have witnessed the introduction of the first vision products on board vehicles in the automobile industry (e.g., the Lane Departure Warning System available in Mercedes and Freightliner's trucks [7]), we believe that the introduction of vision-based systems in the automobile industry is still several years away. In our perspective, the future holds promise for driver assistance systems that can be tailored to solve well-defined tasks that attempt to support, not replace the driver. Even though, several orders of improvement in sensor performance and algorithm robustness are needed before these systems can be deployed effectively.

In spite of the technical challenges that lie ahead, we believe that some degree of optimism is justifiable based on the progress that this domain has seen over the last few years. Judging from the research activities in this field worldwide, it is certain that it will continue to be among the hottest research areas in the future. Major motor companies, government agencies, and universities, are all expected to work together to make significant progress in this area over the next few years. Rapidly falling costs for the sensors and processors combined with increasing image resolution provides the basis for a continuous growth of this field.

Acknowledgments

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 the US National Science Foundation under CRCD grant No. 0088086.

References

[1] W. Jones, "Keeping Cars from Crashing," *IEEE Spectrum*, vol. 38, no 9, pp. 40-45, 2001.
[2] W. Jones, "Building Safer Cars," *IEEE Spectrum*, vol. 39, no. 1, pp. 82-85, 2002.
[3] E. Dickmanns, "The Development of Machine Vision For Road Vehicles in the Last Decade," *Proc. IEEE Intelligent Vehicle Symp.*, 2002.
[4] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-Based intelligent Vehicles: State of the Art and Perspectives," *Robotics and Autonomous Systems*, vol. 32, pp. 1-16, 2000.
[5] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artifical Vision in Road Vehicles," *Proc. IEEE*, vol. 90, no. 7, pp. 1258-1271, 2002.
[6] R. Bishop, "Intelligent Vehicle Applications Worldwide," *IEEE Intelligent Systems*, 2000.
[7] D. Gavrila and U. Franke, and C. Wohler, and S. Gorzig, "Real-Time Vision for Intelligent Vehicles," *IEEE Instrumentation and Measurement Magazine*, pp. 22-27, 2001.
[8] *IEEE Trans. Intelligent Transportation Systems*, special issue on Vision Applications and Technology for Intelligent Vehicles: Part I—Infrastructure, vol. 1, 2000.
[9] *IEEE Trans. Intelligent Transportation Systems*, special issue on Vision Applications and Technology for Intelligent Vehicles: Part II—Vehicles, vol. 1, 2000.
[10] *Image and Vision Computing*, special issue on applications of computer vision to intelligent vehicles, vol. 18, 2000.
[11] *IEEE Trans. Vehicular Technology*, special issue on in-vehicle computer vision systems, vol. 53, 2004.
[12] F. Heimes and H. Nagel, "Towards Active Machine-Vision-Based Driver Assistance for Urban Areas," *Int'l J. Computer Vision*, vol. 50, pp. 5-34, 2002.
[13] U. Franke et al., *From Door to Door—Principles and Applications of Computer Vision for Driver Assistant Systems.* Butterworth Heinemann, 2001.
[14] M. Schwarzinger, T. Zielke, D. Noll, M. Brauckmann, and W. von Seelen, "Vision-Based Car-Following: Detection, Tracking, and Identification," *Proc. IEEE Intelligent Vehicle Symp.*, pp. 24-29, 1992.
[15] B. Ulmer, "Vita—An Autonomous Road Vehicle (ARV) for Collision Avoidance in Traffic," *Proc. IEEE Intelligent Vehicles Symp.*, pp. 36-41, 1992.
[16] M. Bertozzi and A. Broggi, "Gold: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection," *IEEE Trans. Image Processing*, vol. 7, pp. 62-81, 1998.
[17] M. Bertozzi, A. Broggi, and A. Fascioli, "Obstacle and Lane Detection on Argo Autonomous Vehicle," *IEEE Intelligent Transportation Systems*, 1997.
[18] S. Tsugawa and Sadayuki, "Vision-Based Vehicle on Japan: Machine Vision Systems and Driving Control Systems," *IEEE Trans. Industrial Electronics*, vol. 41, no. 4, pp. 398-405, 1994.
[19] *Vehicle-Highway Automation Activities in the United States.* US Dept. of Transportation, 1997.
[20] C. Thorpe, J.D. Carlson, D. Duggins, J. Gowdy, R. MacLachlan, C. Mertz, A. Suppe, and C. Wan, "Safe Robot Driving in Cluttered Environments," *Proc. 11th Int'l Symp. Robotics Research*, 2003.
[21] D. Pomerleau and T. Jochem, "Rapidly Adapting Machine Vision for Automated Vehicle Steering," *IEEE Expert*, vol. 11, pp. 19-27, 1996.
[22] C. Thorpe and T. Kanade, "Vision and Navigation for the Carnegie Mellon Navlab," *Proc. DARPA Image Understanding Workshop*, 1985.
[23] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Vision and Navigation for the Carnegie-Mellon Navlab," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, pp. 362-373, 1988.
[24] M. Walton, "Robots Fail to Complete Grand Challenge," *CNN News*, Mar. 2004.
[25] M. Herbert, "Active and Passive Range Sensing for Robotics," *Proc. IEEE Int'l Conf. Robotics and Automation*, vol. 1, pp. 102-110, 2000.
[26] S. Park, T. Kim, S. Kang, and K. Heon, "A Novel Signal Processing Technique for Vehicle Detection Radar," *2003 IEEE MTT-S Int'l Microwave Symp. Digest*, pp. 607-610, 2003.
[27] C. Wang, C. Thorpe, and A. Suppe, "Ladar-Based Detection and Tracking of Moving Objects from a Ground Vehicle at High Speeds," *Proc. IEEE Intelligent Vehicles Symp.*, 2003.
[28] J. Hancock, E. Hoffman, R. Sullivan, D. Ingimarson, D. Langer, and M. Hebert, "High-Performance Laser Ranger Scanner," *Proc. SPIE Int'l Conf. Intelligent Transportation Systems*, 1997.
[29] R. Chellappa, G. Qian, and Q. Zheng, "Vehicle Detection and Tracking Using Acoustic and Video Sensors," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, pp. 793-796, 2004.
[30] 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.
[31] G. Marola, "Using Symmetry for Detecting and Locating Objects in a Picture," *Computer Vision, Graphics, and Image Processing*, vol. 46, pp. 179-195, 1989.
[32] A. Kuehnle, "Symmetry-Based Recognition for Vehicle Rears," *Pattern Recognition Letters*, vol. 12, pp. 249-258, 1991.
[33] T. Zielke, M. Brauckmann, and W. von Seelen, "Intensity and Edge-Based Symmetry Detection with an Application to Car-Following," *CVGIP: Image Understanding*, vol. 58, pp. 177-190, 1993.
[34] W. von Seelen, C. Curio, J. Gayko, U. Handmann, and T. Kalinke, "Scene Analysis and Organization of Behavior in Driver Assistance Systems," *Proc. IEEE Int'l Conf. Image Processing*, pp. 524-527, 2000.
[35] J. Crisman and C. Thorpe, "Color Vision for Road Following," *Proc. SPIE Conf. Mobile Robots*, pp. 246-249, 1988.
[36] S.D. Buluswar and B.A. Draper, "Color Machine Vision for Autonomous Vehicles," *Int'l J. Eng. Applications of Artificial Intelligence*, vol. 1, no. 2, pp. 245-256, 1998.
[37] D. Guo, T. Fraichard, M. Xie, and C. Laugier, "Color Modeling by Spherical Influence Field in Sensing Driving Environment," *Proc. IEEE Intelligent Vehicle Symp.*, pp. 249-254, 2000.
[38] H. Mori and N. Charkai, "Shadow and Rhythm as Sign Patterns of Obstacle Detection," *Proc. Int'l Symp. Industrial Electronics*, pp. 271-277, 1993.
[39] E. Dickmanns et al., "The Seeing Passenger Car 'Vamors-P'," *Proc. Int'l Symp. Intelligent Vehicles*, pp. 24-26, 1994.
[40] C. Tzomakas and W. Seelen, "Vehicle Detection in Traffic Scenes Using Shadows," Technical Report 98-06, Institut für Neuroinformatik, Ruht-Universitat, Bochum, Germany, 1998.
[41] M. Bertozzi, A. Broggi, and S. Castelluccio, "A Real-Time Oriented System for Vehicle Detection," *J. Systems Architecture*, pp. 317-325, 1997.
[42] N. Matthews, P. An, D. Charnley, and C. Harris, "Vehicle Detection and Recognition in Greyscale Imagery," *Control Eng. Practice*, vol. 4, pp. 473-479, 1996.
[43] 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.
[44] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. Seelen, "An Image Processing System for Driver Assistance," *Image and Vision Computing*, vol. 18, no. 5, 2000.

[45] P. Parodi and G. Piccioli, "A Feature-Based Recognition Scheme for Traffic Scenes," *Proc. IEEE Intelligent Vehicles Symp.*, pp. 229-234, 1995.

[46] M. Betke, E. Haritaglu, and L. Davis, "Real-Time Multiple Vehicle Detection and Tracking from a Moving Vehicle," *Machine Vision and Applications,* vol. 12, no. 2, 2000.

[47] N. Srinivasa, "A Vision-Based Vehicle Detection and Tracking Method for Forward Collision Warning," *Proc. IEEE Intelligent Vehicle Symp.*, pp. 626-631, 2002.

[48] T. Bucher, C. Curio, J. Edelbrunner, C. Igel, D. Kastrup, I. Leefken, G. Lorenz, A. Steinhage, and W. von Seelen, "Image Processing and Behavior Planning for Intelligent Vehicles," *IEEE Trans. Industrial Electronics,* vol. 50, no. 1, pp. 62-75, 2003.

[49] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A Real-Time Precrash Vehicle Detection System," *Proc. IEEE Int'l Workshop Application of Computer Vision,* Dec. 2002.

[50] T. Kalinke, C. Tzomakas, and W. von Seelen, "A Texture-Based Object Detection and an Adaptive Model-Based Classification," *Proc. IEEE Int'l Conf. Intelligent Vehicles,* pp. 143-148, 1998.

[51] R. Haralick, B. Shanmugam, and I. Dinstein, "Texture Features for Image Classification," *IEEE Trans. System, Man, and Cybernetics,* pp. 610-621, 1973.

[52] R. Cucchiara and M. Piccardi, "Vehicle Detection under Day and Night Illumination," *Proc. Int'l ICSC Symp. Intelligent Industrial Automation,* 1999.

[53] R. Mandelbaum, L. McDowell, L. Bogoni, B. Beich, and M. Hansen, "Real-Time Stereo Processing, Obstacle Detection, and Terrain Estimation from Vehicle-Mounted Stereo Cameras," *Proc. IEEE Workshop Applications of Computer Vision,* pp. 288-289, 1998.

[54] U. Franke and I. Kutzbach, "Fast Stereo Based Object Detection for Stop and Go Traffic," *Intelligent Vehicles,* pp. 339-344, 1996.

[55] G. Zhao and S. Yuta, "Obstacle Detection by Vision System for Autonomous Vehicle," *Intelligent Vehicles,* pp. 31-36, 1993.

[56] M. Bertozzi and A. Broggi, "Vision-Based Vehicle Guidance," *Computer,* vol. 30, no 7, pp. 49-55, July 1997.

[57] C. Knoeppel, A. Schanz, and B. Michaelis, "Robust Vehicle Detection at Large Distance Using Low Resolution Cameras," *Proc. IEEE Intelligent Vehicle Symp.,* pp. 267-272, 2000.

[58] M. Okutomi and T. Kanade, "A Multiple-Baseline Stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, pp. 353-363, 1993.

[59] T. Williamson and C. Thorpe, "A Trinocular Stereo System for Highway Obstacle Detection," *Proc. IEEE Int'l Conf. Robotics and Automation,* 1999.

[60] A. Giachetti, M. Campani, and V. Torre, "The Use of Optical Flow for Road Navigation," *IEEE Trans. Robotics and Automation,* vol. 14, no. 1, pp. 34-48, 1998.

[61] D. Fleet, J. Barron, and S. Beauchemin, "Performance of Optical Flow," *Int'l J. Computer Vision,* vol. 12, pp. 43-77, 1994.

[62] W. Kruger, W. Enkelmann, and S. Rossle, "Real-Time Estimation and Tracking of Optical Flow Vectors for Obstacle Detection," *Proc. IEEE Intelligent Vehicle Symp.,* pp. 304-309, 1995.

[63] J. Weng, N. Ahuja, and T. Huang, "Matching Two Perspective Views," *IEEE Trans. Pattern Analysis and Machine intelligence,* vol. 14, pp. 806-825, 1992.

[64] S. Smith and J. Brady, *ASSET-2: Real-Time Motion Segmentation and Shape Tracking,* vol. 17, 1995.

[65] D. Koller, N. Heinze, and H. Nagel, "Algorithmic Characterization of Vehicle Trajectories from Image Sequence by Motion Verbs," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* pp. 90-95, 1991.

[66] B. Heisele and W. Ritter, "Obstacle Detection Based on Color Blob Flow," *Proc. IEEE Intelligent Vehicle Symp.,* pp. 282-286, 1995.

[67] T. Ito, K. Yamada, and K. Nishioka, "Understanding Driving Situations Using a Network Model," *Intelligent Vehicles,* pp. 48-53, 1995.

[68] U. Regensburger and V. Graefe, "Visual Recognition of Obstacles on Roads," *Intelligent Robots and Systems,* pp. 73-86, 1995.

[69] V. Graefe and U. Regensburger, "A Novel Approach for the Detection of Vehicle on Freeways by Real-Time Vision," *Intelligent Vehicles,* 1996.

[70] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miche, S. Mousset, and G. Moulminet, "A Cooperative Approach to Vision-Based Vehicle Detection," *IEEE Intelligent Transportation Systems,* pp. 209-214, 2001.

[71] J. Wu and X. Zhang, "A PCA Classifier and Its Application in Vehicle Detection," *Proc. IEEE Int'l Joint Conf. Neural Networks,* 2001.

[72] D. Noli, M. Werner, and W. von Seelen, "Real-Time Vehicle Tracking and Classification," *Intelligent Vehicles,* pp. 101-106, 1995.

[73] H. Schneiderman and T. Kanade, "A Statistical Method for 3D Object Detection Applied to Faces and Cars," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* pp. 746-751, 2000.

[74] M. Weber, M. Welling, and P. Perona, "Unsupervised Learning of Models for Recognition," *Proc. European Conf. Computer Vision,* pp. 18-32, 2000.

[75] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *Int'l J. Computer Vision,* vol. 38, no 1, pp. 15-33, 2000.

[76] Z. Sun, G. Bebis, and R. Miller, "Quantized Wavelet Features and Support Vector Machines for On-Road Vehicle Detection," *Proc. IEEE Int'l Conf. Control, Automation, Robotics, and Vision,* Dec. 2002.

[77] Z. Sun, G. Bebis, and R. Miller, "On-Road Vehicle Detection Using Gabor Filters and Support Vector Machines," *Proc. IEEE Int'l Conf Digital Signal Processing,* July 2002.

[78] S. Agarwal and D. Roth, "Learning a Sparse Representation for Object Detection," *Proc. European Conf. Computer Vision,* 2002.

[79] B. Leung, "Component-Based Car Detection in Street Scene Images," technical report, Massachusetts Inst. of Technology, 2004.

[80] D. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. Int'l Conf. Computer Vision,* 1999.

[81] J. Ferryman, S. Maybank, and A. Worrall, "Visual Surveillance for Moving Vehicles," *Proc. IEEE Workshop Visual Surveillance,* pp. 73-80, 1998.

[82] A. Rajagopalan and R. Chellappa, "Vehicle Detection and Tracking in Video," *Proc. IEEE Int'l Conf. Image Processing,* pp. 351-354, 2000.

[83] X. Clady, F. Collange, F. Jurie, and P. Martinet, "Cars Detection and Tracking with a Vision Sensor," *Proc. IEEE Intelligent Vehicle Symp.,* pp. 593-598, 2003.

[84] X. Li, X. Yao, Y. Murphey, R. Karlsen, and G. Gerhart, "A Real Time Vehicle Detection and Tracking System in Outdoor Traffic Scenes," *Proc. Int'l Conf. Pattern Recognition,* 2004.

[85] M. Betke, E. Haritaglu, and L. Davis, "Multiple Vehicle Detection and Tracking in Hard Real Time," *Proc. IEEE Intelligent Vehicles Symp.,* pp. 351-356, 1996.

[86] G. Lefaix, T. Marchand, and P. Bouthemy, "Motion-Based Obstacle Detection and Tracking for Car Driving Assistance," *Proc. 16th Int'l Conf. Pattern Recognition,* pp. 74-77, 2002.

[87] C. Hoffman, T. Dang, and C. Stiller, "Vehicle Detection Fusing 2D Visual Features," *Proc. IEEE Intelligent Vehicles Symp.,* pp. 280-285, 2004.

[88] H. Gish and R. Mucci, "Weak Target Detection Using the Entropy Concept," *Proc. SPIE,* vol. 1305 1990.

[89] S. Avidan, "Support Vector Tracking," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2001.

[90] L. Fletcher, L. Peterson, and A. Zelinsky, "Driver Assistance Systems Based on Vision In and Out of Vehicles," *Proc. IEEE Intelligent Vehicles Symp. (IV2003),* 2003.

[91] M. Suwa, Y. Wu, M. Kobayashi, and S. Ogata, "A Stereo-Based Vehicle Detection Method under Windy Conditions," *Proc. IEEE Intelligent Vehicle Symp.,* pp. 246-249, 2000.

[92] M. Bertozzi, A. Broggi, and A. Fascioli, "Self-Calibration of a Stereo Vision System for Automotive Applications," *Proc. IEEE Int'l Conf. Robotics and Automation,* pp. 3698-3703, 2001.

[93] C. Morimoto, D. Dementhon, L. Davis, R. Chellappa, and R. Nelson, "Detection of Independently Moving Object in Passive Video," *Proc. IEEE Intelligent Vehicle Symp.,* pp. 270-275, 1995.

[94] Y. Liang, H. Tyan, S. Chang, H. Liao, and S. Chen, "Video Stabilization for a Camcorder Mounted on a Moving Vehicle," *IEEE Trans. Vehicular Technology,* vol. 53, 2004.

[95] Z. Sun, G. Bebis, and R. Miller, "On-Road Vehicle Detection Using Evolutionary Gabor Filter Optimization," *IEEE Trans. Intelligent Transportation Systems,* vol. 6, no. 2, pp. 125-137, 2005.

[96] Z. Sun, G. Bebis, and R. Miller, "Boosting Object Detection Using Feature Selection," *Proc. IEEE Int'l Conf. Advanced Video and Signal Based Surveillance,* 2003.

[97] Z. Sun, G. Bebis, and R. Miller, "Evolutionary Gabor Filter Optimization with Application to Vehicle Detection," *Proc. IEEE Int'l Conf. Data Mining,* 2003.

[98] Z. Sun, G. Bebis, and R. Miller, "Object Detection Using Feature Subset Selection," *Pattern Recognition,* vol. 37, pp. 2165-2176, 2004.

[99] R. Collins and Y. Liu, "On-Line Selection of Discriminant Tracking Features," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.
[100] J. Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis.* Cambridge Univ. Press, 2004.
[101] J. Collado, C. Hilario, A. de la Escalera, and J. Armingol, "Model-Based Vehicle Detection for Intelligent Vehicles," *Proc. IEEE Intelligent Vehicles Symp.,* 2004.
[102] K. She, G. Bebis, H. Gu, and R. Miller, "Vehicle Tracking Using On-Line Fusion of Color and Shape Features," *Proc. IEEE Int'l Conf. Intelligent Transportation Systems,* 2004.
[103] J. Wang, G. Bebis, and R. Miller, "Overtaking Vehicle Detection Using Dynamic and Quasi-Static Background Modeling," *Proc. IEEE Workshop Machine Vision for Intelligent Vehicles,* 2005.
[104] Z. Sun, G. Bebis, and R. Miller, "Improving the Performance of On-Road Vehicle Detection by Combining Gabor and Wavelet Features," *Proc. IEEE Fifth Int'l Conf. Intelligent Transportation Systems,* Sept. 2002.
[105] L. andreone, P. Antonello, M. Bertozzi, A. Broggi, A. Fascioli, and D. Ranzato, "Vehicle Detection and Localization in Infra-Red Images," *Proc. IEEE Fifth Int'l Conf. Intelligent Transportation Systems,* 2002.
[106] P. Barham, X. Zhang, L. Andreone, and M. Vache, "The Development of a Driver Vision Support System Using Far Infrared Technology: Progress to Date on the Darwin Project," *Proc. IEEE Intelligent Vehicles Symp.,* 2000.
[107] D. Pomerleau and T. Jochem, "Automatic Vehicle Detection in Infrared Imagery Using a Fuzzy Inference-Based Classification System," *IEEE Trans. Fuzzy Systems,* vol. 9, no. 1, pp. 53-61, 2001.
[108] M. Kagesawa, S. Ueno, K. Ikeuchi, H. Kashiwagi, "Recognizing Vehicles in Infrared Images Using IMAP Parallel Vision Board," *IEEE Trans. Intelligent Transportation Systems,* vol. 2, no. 1, pp. 10-17, 2001.
[109] K. Sobottka, E. Meier, F. Ade, and H. Bunke, "Towards Smarter Cars," *Proc. Int'l Workshop Sensor Based Intelligent Robots,* pp. 120-139, 1999.
[110] T. Spirig, M. Marle, and P. Seitz, "The MultiTap Lock-In CCD with Offset Subtraction," *IEEE Trans. Electron Devices,* pp. 1643-1647, 1997.
[111] K. Yamada, "A Compact Integrated Vision Motion Sensor for Its Applications," *IEEE Trans. Intelligent Transportation System,* vol. 4, no. 1, pp. 35-41, 2003.
[112] C. Stiller, J. Hipp, C. Rossig, and A. Ewald, "Multisensor Obstacle Detection and Tracking," *Image and Vision Computing,* vol. 18, pp. 389-396, 2000.
[113] Y. Fang, M. Mizuki, I. Masaki, and B. Horn, "TV Camera-Based Vehicle Motion Detection and Its Chip Implementation," *Proc. IEEE Intelligent Vehicle Symp.,* pp. 134-139, 2000.
[114] W. Enkelmann, V. Gengenbach, and W. Kruger, "Obstacle Detection by Real-Time Optical Flow Evaluation," *Proc. IEEE Intelligent Vehicle Symp.,* pp. 97-102, 1994.
[115] S. Ghiasi, H. Moon, and M. Sarrafzadeh, "Collaborative and Reconfigurable Object Tracking," *Proc. Int'l Conf. Eng. of Reconfigurable Systems and Algorithms,* pp. 13-20, 2003.
[116] A. Benedetti and P. Perona, "Real-Time 2-D Feature Detection on a Reconfigurable Computer," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1998.
[117] R. Mandelbaum, M. Hansen, P. Burt, and S. Baten, "Vison for Autonomous Mobilty: Image Processing on the Vfe-200," *Proc. IEEE Int'l Symp. Intelligent Control,* 1998.
[118] J. Phillips, H. Moon, S. Rizvi, and J. Rauss, "The Feret Evaluation Methodology for Face-Recognition Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, pp. 1090-1104, 2000.
[119] "Real-Time Event Detection Solutions (CREDS) for Enhanced Security and Safety in Public Transportation Challenge," *Proc. IEEE Int'l Conf. Advanced Video and Signal Based Surveillance,* 2005.
[120] M. Boshra and B. Bhanu, "Predicting Object Recognition Performance under Data Uncertainty, Occlusion and Clutter," *Proc. IEEE Int'l Conf. Image Processing,* 1998.
[121] M. Nishigaki, M. Saka, T. Aoki, H. Yuhara, and M. Kawai, "Fail Ouput Algorithm of Vision Sensing," *Proc. IEEE Intelligent Vehicle Symp.,* pp. 581-584, 2000.

Zehang Sun received the BEng degree in telecommunication and the MEng degree in digital signal processing from Northern Jiaotong University, China, in 1994 and 1997, respectively, the MEng degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 1999, and the PhD degree in computer science and engineering from the University of Nevada, Reno, in 2003. Dr. Sun joined eTreppid Technologies, LLC immediately upon his graduation. His expertise is in the area of real-time computer vision systems, statistical pattern recognition, artificial intelligence, digital signal processing, and embedded systems. He is a member of the IEEE.

George Bebis received the BS degree in mathematics and the MS degree in computer science from the University of Crete, Greece, in 1987 and 1991, respectively, and the PhD degree in electrical and computer engineering from the University of Central Florida, Orlando, in 1996. Currently, he is an associate professor in the Department of Computer Science and Engineering at the University of Nevada, Reno (UNR) and the director of the UNR Computer Vision Laboratory (CVL). His research interests include computer vision, image processing, pattern recognition, machine learning, and evolutionary computing. His research is currently funded by US NSF, NASA, ONR, and the Ford Motor Company. Dr. Bebis is an associate editor of the *Machine Vision and Applications Journal* and serves on the editorial board of the *Pattern Recognition Journal* and the *International Journal Artificial Intelligence Tools*. He has served on the program committees of various national and international conferences, and has organized and chaired several conference sessions. In 2002, he received the Lemelson Award for innovation and Entrepreneurship. He is a member of the IEEE and the IAPR Educational Committee.

Ronald Miller received the BS degree in physics in 1983 from the University of Massachusetts, and the PhD degree in physics from the Massachusetts Institute of Technology in 1988. His research has ranged from computational modeling of plasma and ionospheric instabilities to automotive safety applications. Dr. Miller heads a research program at Ford Motor Company in intelligent vehicle technologies focusing on advanced RF communication, radar, and optical sensing systems for accident avoidance and telematics.