Vision-Based Vehicle Guidance

This implementation of lane and obstacle detection for an autonomous, self-guided vehicle succeeds by tailoring vision and computational techniques to an affordable SIMD architecture.

Massimo Bertozzi and Alberto Broggi
Università di Parma

From automatic transmissions to cruise control to antilock braking systems, passenger vehicles have incorporated automated subsystems that make driving easier and safer. Moreover, a growing number of microprocessors on late-model automobiles now monitor and control engine operations to improve mileage and reduce emissions. Thus an eventual move to autonomous, self-guided vehicles might be viewed as a logical progression rather than a radical break with the tradition of entirely human-controlled vehicle operation.

The ability to perceive, or sense, the surrounding environment is essential to driving and thus to the development of autonomous self-guided vehicles. Though typically associated with living organisms, perception can also be performed by radio, acoustic, magnetic, and tactile sensors. These active sensors can measure quantities directly and generate small amounts of data; however, there are applications in which only machine vision, which acquires data noninvasively (by passive sensors such as cameras), can be successfully employed. For example, when multiple robots move about in the same indoor environment, the signals they emit may interfere with each other. This problem is greater outdoors in an unstructured environment, where many vehicles may be moving simultaneously. Hence, passive sensors are far better for automobile traffic applications.

But automatic vehicle guidance in outdoor environments presents other problems intrinsic to the use of vision. For example, such environments do not present structured information. Passenger vehicles should be able to traverse all types of roads, without the integration of expensive infrastructure modifications. Without structure, key parameters such as illumination and contrast cannot be assumed but must be measured directly by the vision sensor. Hence, processing must be robust enough to adapt to different road and weather conditions and to tolerate sudden changes from sunlight to shadow—or even to the inside of a tunnel. Camera movement and drifts in its calibration must be tolerated, too.

In addition, image processing for automatic vehicle guidance requires equipment that can compute a lot of data rapidly, because vehicle speed is proportional to, and bounded by, the processing rate. Computer architectures and processing techniques optimized for real-time performance offer a solution, but only low-cost systems can succeed in the marketplace. Thus, designers must use off-the-shelf components or ad hoc dedicated low-cost solutions.

Despite its demands and complexity, computer vision offers a powerful way to sense the environment. It has been widely used for such vehicle-related tasks as road following, platooning (where an automatic vehicle follows a manually driven one), overtaking and passing slower vehicles, and automatic parking. One of the most challenging areas for researchers is road following, which requires the two basic functionalities of *lane detection* and *obstacle detection*.

LANE DETECTION

In most autonomous vehicle prototypes, *road feature identification* determines the vehicle's relative position in the lane, and then *actuators* keep the vehicle in a safe position. Although some systems have been designed to work on completely unstructured roads and terrains, lane detection has generally been reduced to the localization of specific features, such as lane markings painted on the road surface. Localizing specific features presents two basic problems: Shadows (projected by trees and other objects) can alter the road's visual texture, and other vehicles can partly occlude road markings.

Fortunately, lane marking localization can be performed by analyzing a single still image. Moreover, certain assumptions can aid detection and/or speed processing:

Obstacle detection can be reduced to identifying the free space into which the vehicle can safely move.

- Instead of processing entire images, a computer vision system can analyze specific regions (the "focus of attention")[1] to identify and extract the features of interest.
- The system can assume a fixed or smoothly varying lane width and thereby limit its search to almost-parallel lane markings.
- If the system makes certain shape assumptions, road geometry reconstruction can be simplified; for example, some implementations[1,2] assume a curve whose curvature changes without discontinuities.
- A system can exploit its knowledge of camera calibration and the assumption of a precise 3D road model (for example, a flat road without bumps) to more easily localize features and simplify the mapping between image pixels and their corresponding world coordinates.

There are two approaches to lane detection: *model-driven*,[3-5] in which deformable templates[6] are iteratively modified, and *feature-driven*, in which features are extracted and localized. We use the latter approach because it limits the computation-intensive processing of images to extracting features of interest.

OBSTACLE DETECTION

In some systems obstacle determination is limited to the localization of vehicles by means of a search for specific patterns, possibly supported by other features such as shape, symmetry, or the use of a bounding box. Here too, processing can be based on the analysis of a single still image, but the approach fails for obstacles other than vehicles because they don't match the vehicle model.

If, more broadly, we search for an object that can obstruct the vehicle's path, obstacle detection is reduced to identifying the *free space* (the area into which the vehicle can safely move). There are different techniques for addressing this more general approach. The most common ones—analysis of the optical flow field and the processing of stereo images—involve processing two or more images. With optical-flow-field analysis, multiple images are acquired at different times; stereo images, of course, are acquired simultaneously from different points of view.

Both techniques have higher computational complexity because they require processing *sets* of images. Moreover, the techniques must be robust enough to tolerate the noise caused by vehicle movement and drifts in the acquisition system's calibration. Optical-flow-based[7] techniques detect obstacles indirectly by analyzing the velocity field. Stereo image techniques identify the correspondences between pixels in the different images. Stereo vision has advantages in that it can detect obstacles directly and, unlike optical-flow-field analysis, is not constrained by speed.

A very promising approach is to remove the inherent perspective effect[5,8] from acquired stereo images. Similar image-warping techniques[2] have been used for both obstacle detection and stereo image processing, but removing the perspective effect can also help with lane detection. Thus an integrated approach lets us use a common initial transform for both problems.

ENABLING SIMD COMPUTATION

Massively parallel SIMD (single instruction, multiple data) architectures have long been seen as an answer to the demanding task of real-time low-level image processing, but they have been too expensive to place onboard mass-market vehicles. Thanks to technological evolution, both general-purpose processors (such as Intel's MMX-enhanced processors) and low-cost special-purpose systems can now exploit spatial parallelism to boost performance.

SIMD machines perform the same operation on each image pixel without taking into account their different meanings. Unfortunately, depending on the scene and the angle of view, each pixel carries a different amount of information and must be processed in a different manner, depending on its position within the image. This kind of processing cannot be implemented efficiently on SIMD systems.

To cope with this problem, we use a geometrical transform called *inverse perspective mapping* (IPM).[9] Using a priori knowledge of both the scene and the acquisition device, the IPM technique lets us remove the perspective effect and produce a new image in which the information content is homogeneously distributed among all pixels. In the remapped image, the amount of information carried by each pixel no longer depends on the pixel's position, making the SIMD approach practical.

Stereo image analysis

When an image is acquired from a mobile vehicle traversing a flat road, the IPM technique results in a remapped image that represents an overhead view of the road surface, or the *road texture*. Figures 1a-1d represent left and right views before and after remapping.

The acquisition process can be devised as a transform from the 3D world space to the 2D image space. The inverse transform, which represents a 3D reconstruction of the world from a 2D image, is usually indeterminate because information is lost during acquisition. Stereo vision helps recover scene depth. The literature presents many different ways to analyze the two views, but their computational complexity is too great for our application.

However, using IPM on stereo images can greatly simplify the localization of homologous points:

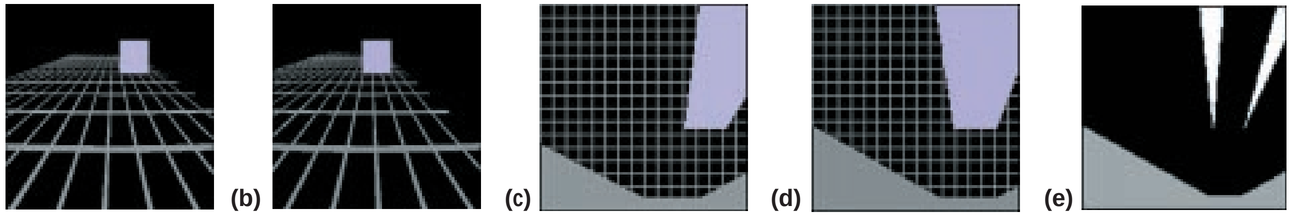


Figure 1. Acquisition of an ideal square, homogeneous obstacle: (a) left image, (b) right image, (c) left remapped image, (d) right remapped image, and (e) difference image. The gray area at the bottom represents the region of the road not seen by (c) the left camera, (d) the right camera, and (e) both cameras.

Instead of searching for correspondent features to determine the scene depth, we use a priori knowledge of the environment model to replace the search algorithm with a check for model consistency.

Shaping the horopter

Vision system setup identifies a 3D surface (called a *horopter*) whose points are projected by the acquisition process onto image points that have the same coordinates in both stereo images. Hence, when a stereo system captures an object that matches the horopter shape and position, the two acquired stereo images are identical. Thus, the search for homologous points becomes trivial, and the difference between pixels with equal coordinates is always zero. This property can be exploited to determine whether a 3D object has the same shape and position as the horopter.[9] The underlying idea is to shape the horopter according to the road model, so that detecting obstacles can be reduced to detecting differences between the two images.

In the case of two identical cameras with coplanar axes, the horopter has a spherical shape. The horopter size and displacement can be changed by altering vision system parameters, such as camera vergence, but unfortunately its *shape* cannot be modified by altering only the acquisition setup; hence, the horopter cannot be superimposed onto a generic surface.

Electronic vergence

The horopter is moved onto the road plane by *electronic vergence*—the use of the IPM technique on both stereo images. When the difference between both remapped images presents sufficiently large clusters of nonzero pixels, the flat road hypothesis is not met. This happens when an obstacle is in front of the vehicle: In this case the difference image, generated by disparities between the object shape and the horopter surface, presents large clusters of nonzero pixels with a triangular shape, as shown in Figure 1e. Since the IPM algorithm maps lines perpendicular to the road surface into lines that pass through the camera projection onto the road surface, in the difference image the obstacle's two vertical edges are transformed into two triangles whose sides' prolongations intersect the projections of the two cameras on the road plane. The position and shape of these triangles are used to locate obstacles, establishing their distance and exact position in the 3D world space.

THE GOLD SYSTEM

The Generic Obstacle and Lane Detection (GOLD) system is a stereo-vision-based hardware and software architecture developed at the University of Parma. A special-purpose computer architecture, Paprica,[10] was developed to provide powerful computation at modest cost. It consists of a SIMD massively parallel system with 256 processing elements, enhanced by a dedicated stereo acquisition device and a hardware system that performs the IPM transform in real time.

To make generic obstacle detection fast and resilient to camera calibration and vehicle movement, the process is reduced to determining the free space in front of the vehicle, without any 3D world reconstruction.[2] Free-space determination, based on stereo IPM, lets the system detect any object that rises significantly from the flat road surface. The left remapped image is also used for lane detection, since it is typically more centered with respect to the road. Localization of obstacles helps in detecting possible occlusions of lane markings. In occluded regions the position of lane markings can be extrapolated from previous images in the sequence and from other constraints on road geometry.

How GOLD detects obstacles

In the case of a square, homogeneous obstacle, the difference image presents two disjoint triangular clusters. The lowest corner of the triangles designates the point of contact between the obstacle and the road, as illustrated in Figure 1e. Even when generic obstacles are not square or do not have a homogeneous color, two nearly triangular clusters are discernible in the difference image, though they may not be completely disjoint. In Figure 2 there are two obstacles in

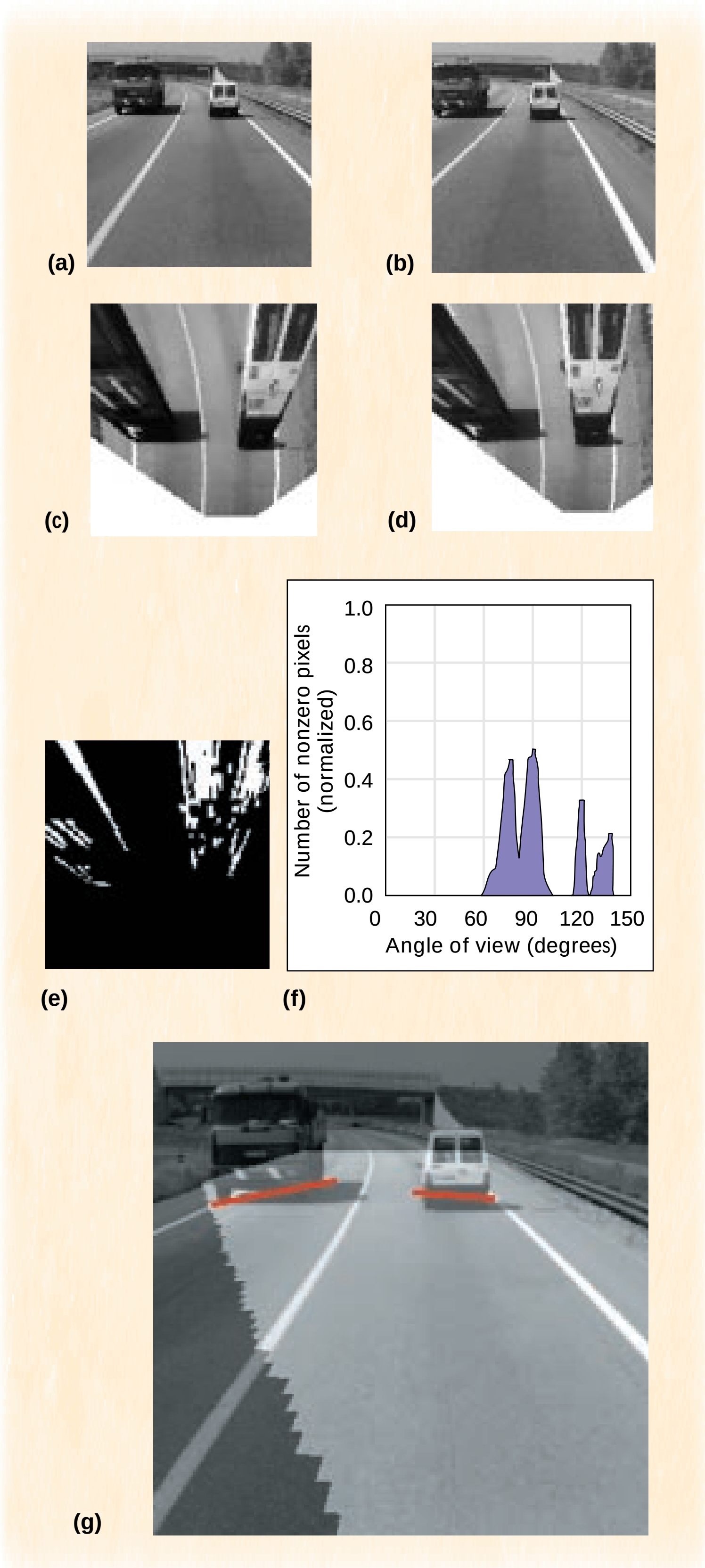


Figure 2. Obstacle detection: (a) left and (b) right stereo images; (c) and (d) the remapped images; (e) the difference image; (f) corresponding polar histogram; (g) the result of obstacle detection, with red markers superimposed on the original left image. The light area in (g) represents the road region visible to both cameras.

the roadway. Figures 2a and 2b show the original left and right images, Figures 2c and 2d show the remapped images, Figure 2e shows the difference image, Figure 2f shows the polar histogram, and Figure 2g shows the result of obstacle detection.

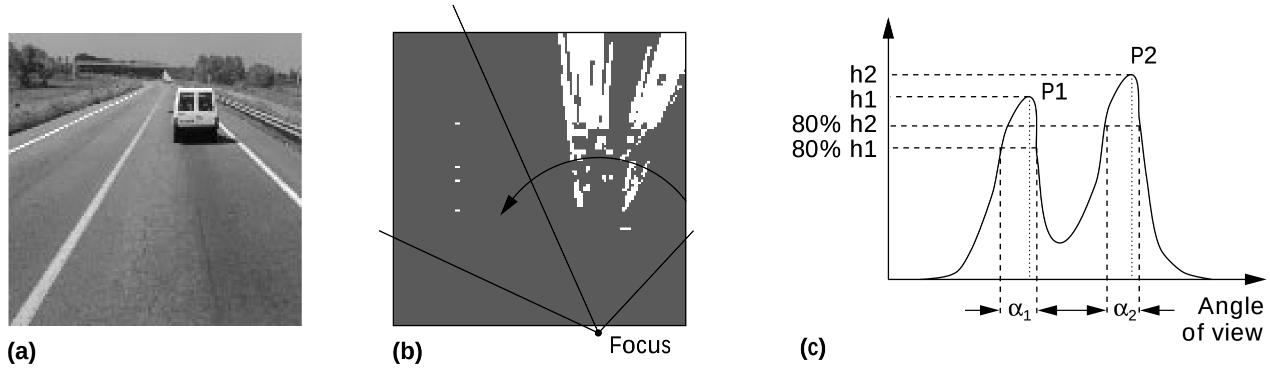
Obstacle detection is then reduced to the search for triangles in the difference image. For simplification, we create a *polar histogram* and use it to search for triangles. To create the histogram, the system scans the difference image along all the lines that pass through the middle point (the focus) between the two cameras' projections onto the road plane and computes the number of nonzero pixels for each direction; the resulting histogram is then low-pass filtered to reduce the influence of noise. Figure 2f is a polar histogram for its respective acquired image, and Figure 3c is the polar histogram for the image in Figure 3a, which has a single obstacle.

Peaks in the polar histogram signify an obstacle's vertical edge. Obstacle detection is further reduced to detecting *pairs* of peaks in the polar histogram. Unfortunately, the presence of more than one obstacle produces several triangles, while a partially visible obstacle produces only one triangle. After analyzing a large number of situations, we determined a criterion for grouping peaks that belong to the same obstacle: It is based on the analysis of different parameters, such as the peaks' amplitude and width and the area subtended by each peak. The position of a group of peaks determines the angle the obstacle is seen from.

The next step is to analyze the difference image along the directions indicated by each peak of the polar histogram. We define a sector to be used in computing a *radial histogram*. The size of this sector is determined using an adaptive threshold whose value is 80 percent of the peak's value. In this way, as Figures 3c and 3d show, even if the vertex of the peak is not centered within the polar histogram peak, the sector to be analyzed should generally include the object's entire vertical edge. To create the radial histogram, we compute the number of nonzero pixels for each direction as a function of the distance from the focus. A threshold is applied and the point of the radial histogram with the smallest coordinate (the lower corner of the triangle, corresponding to the point of contact between the obstacle and the road) is determined. The coordinate of this point lets us estimate the obstacle's distance for feedback to the vehicle driver or to an automatic control system. Figure 3e shows the radial histogram relative to peak α_2 of Figure 3c.

How GOLD detects lanes

Figure 4 shows the steps involved in lane detection. The first phase of lane detection is based on the search for dark-bright-dark horizontal patterns with a given size, because lane markings are represented by bright,



nearly vertical lines of constant width surrounded by a dark background. Every pixel is compared to its left and right horizontal neighbors at a given distance, and a new lane-marking gray-level image is computed. This image, encoding the presence of lane markings, is thresholded (binarized) and scanned to build a list of straight segments; then segments sharing an extremum are concatenated to build polylines. To eliminate possible errors caused by occlusion or noise, GOLD joins two or more polylines that satisfy some criteria, such as closeness or similar orientation. A set of assumptions helps in selecting the groups of polylines that are the most likely to represent road markings. GOLD evaluates every selected group of polylines according to parameters such as lane width, number of lanes, and road curvature; the group of polylines with the highest score is chosen.

Finally, since the external environment model (the flat road) lets us determine the spatial relationship between image pixels and the 3D world, the choice of a polyline group enables us to derive both the road geometry and the vehicle position within the lane.

ROAD TEST

The GOLD system[9] was installed in an experimen-

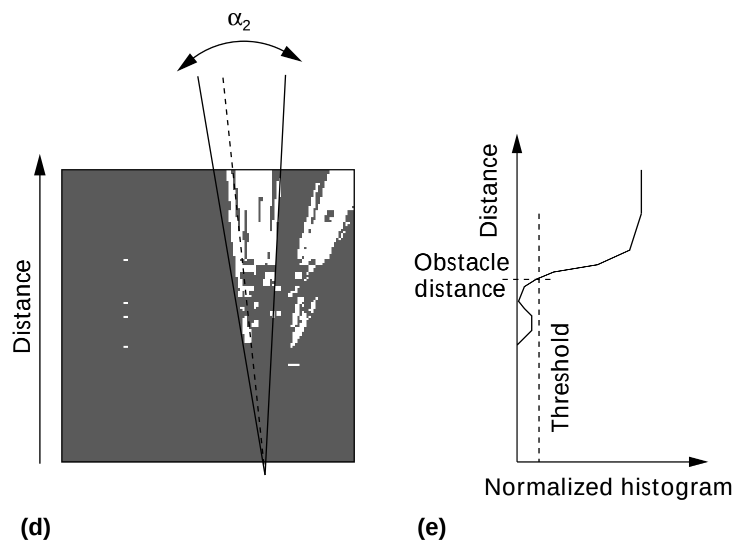


Figure 3. The angle of view from which the obstacle is seen and the determination of its distance: (a) original image; (b) scanning the difference image; (c) polar histogram with two peaks representing a single object; (d) the sector of the difference image that is analyzed in order to compute the radial histogram relative to peak α_2; (e) radial histogram and obstacle distance.

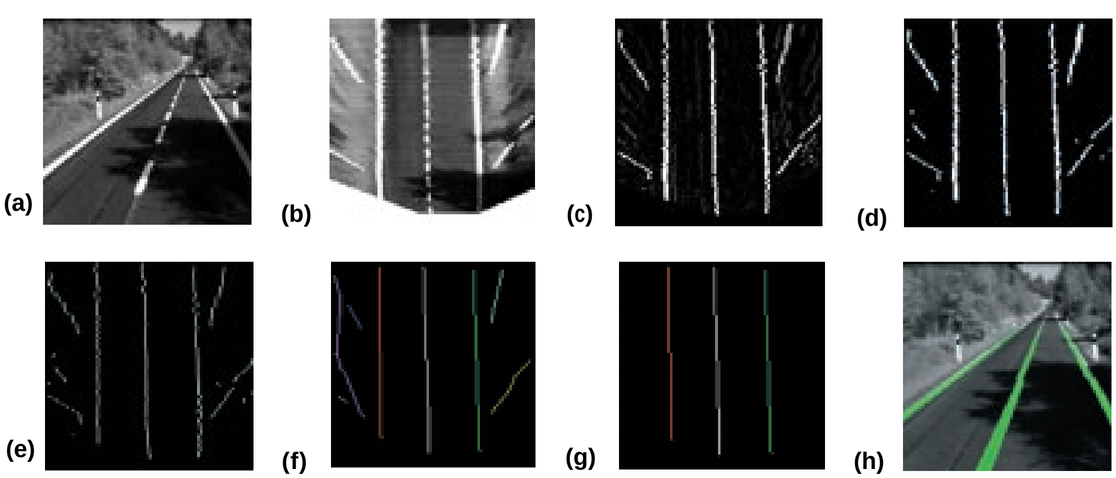


Figure 4. Steps involved in lane detection: (a) acquired image; (b) result of inverse perspective mapping (IPM); (c) enhanced version of the remapped image; (d) binarization of the enhanced image; (e) list of segments (but shown as images here and in (f) and (g)); (f) concatenation of the segments; (g) selected polylines that are likely to represent road markings; and (h) result superimposed on the acquired image. (The inverse of the IPM was used to obtain (h).)

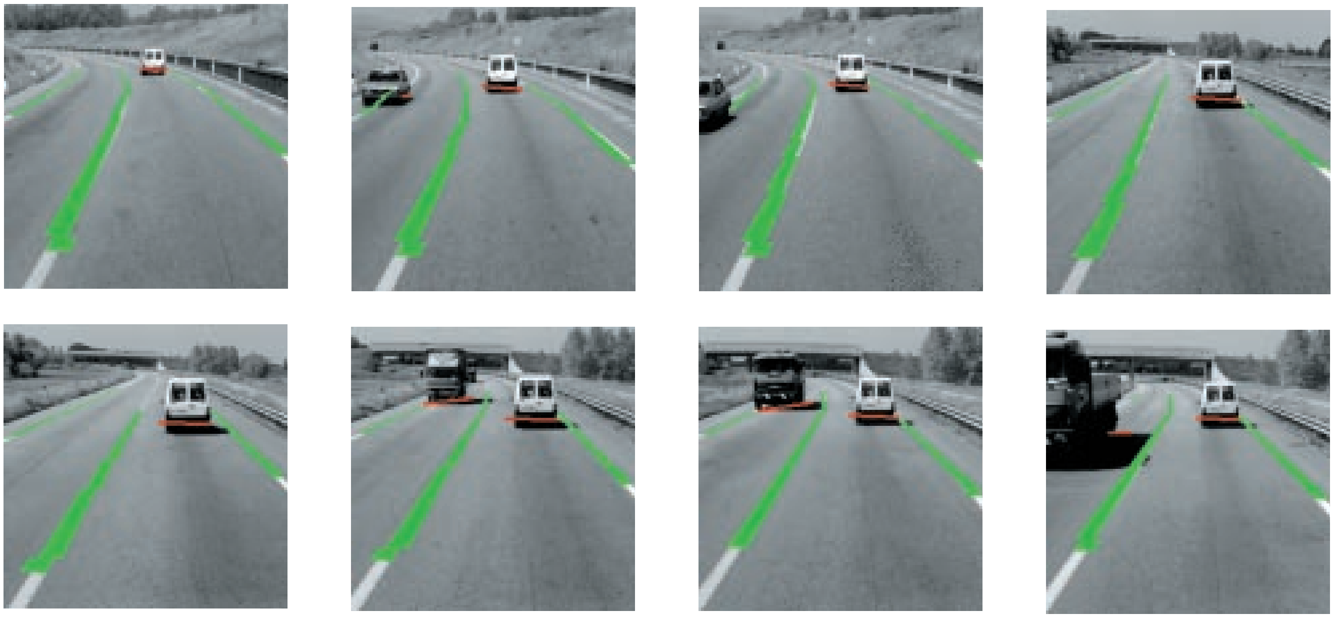


Figure 5. The results of integrated obstacle and lane detection: Lane markers are enhanced in green, and obstacles are localized with red markers.

tal van called the Mob-Lab and tested for more than 3,000 kilometers over extra-urban roads and freeways at up to 80 kilometers per hour. A control panel in the manually driven van showed the lane position and the presence of obstacles, and it warned the driver. The system proved to be reliable under varying illumination and traffic conditions and resilient to vehicle movement (pitch and roll). Because the goal of obstacle detection is not to reconstruct the 3D world but simply to determine the free space in front of the vehicle, vision system calibration is not critical and no dynamic recalibration[2] is required. Tests emulating vehicle movement (up to ±2 degrees in camera orientation and ±10 cm in camera height) demonstrated that obstacles are always detected, even if the difference images are particularly noisy.[9]

Figure 5 shows the results of combined obstacle and lane detection for images acquired under varying road conditions. More results showing the processing of images acquired in many different situations, including image sequences in MPEG format, can be found at http://www.ce.unipr.it/computer_vision.

The Paprica massively parallel system speeds up the detection of both generic obstacles and lane position. In our tests it took less than 140 milliseconds to detect obstacles and localize lane markings for an area extending about 50 meters in front of the vehicle. The availability of two computing engines (Paprica and its front-end processor, a Sparc-based VME board) let us pipeline the low- and high-level phases to achieve a processing rate of about 10 frames per second.

Encouraged by these results, we are integrating the GOLD system into a new prototype vehicle called Argo, named for both the mythological ship used by Jason and the many-eyed god Argus, who had the gift of all-around vision. This Lancia Thema passenger car with autonomous steering capability will soon be enhanced with automatic speed control. For this implementation we are evaluating a new computing engine, a Pentium MMX-based PC with stereo frame-grabber.

Many national and international projects launched over the past decade are studying different approaches to autonomous vehicles. *Computer*'s Web site includes a table detailing several of these projects (http://computer.org/pubs/computer/extras/r7bert.htm). Although prototype vehicles show extremely promising results in both reliability and reduced production and operating costs, much work is still needed. Besides the many technical problems, nontechnical issues such as legal responsibility in the event of faults and/or accidents must still be addressed. So it's a little early to be looking for autonomous vehicles in your automobile dealers' showrooms. ❖

Acknowledgments

This work has been supported by the Progetto Finalizzato Trasporti of the Italian National Research Council under the framework of the Eureka Prometheus Project. We thank Francesco Gregoretti and members of the Department of Electronics at the Politecnico di Torino for their enthusiastic joint development of the Paprica system. We wish to acknowledge significant contributions by the students involved in this project, particularly Alessandra Fascioli. Finally, we owe a great debt to Gianni Conte and Giovanni Adorni for their continuous support of this research.

References

1. E.D. Dickmans and B.D. Mysliwetz, "Recursive 3-D Road and Relative Ego-State Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, May 1992, pp. 199-213.
2. D. Koller et al., "An Integrated Stereo-Based Approach

to Automatic Vehicle Guidance," *Proc. Fifth Int'l Conf. Computer Vision*, IEEE CS Press, Los Alamitos, Calif., 1995, pp. 12-20.
3. V. Graefe and K.D. Kuhnert, "Vision-based Autonomous Road Vehicles," in *Vision-based Vehicle Guidance*, I. Masaki, ed., Springer-Verlag, New York, 1991, pp. 1-29.
4. J. Crisman and C. Thorpe, "SCARF: A Color Vision System that Tracks Roads and Intersections," *IEEE Trans. Robotics and Automation*, Feb. 1993, pp. 49-58.
5. D. Pomerleau and T. Jochem, "Rapidly Adapting Machine Vision for Automated Vehicle Steering," *IEEE Expert*, Apr. 1996, pp. 19-27.
6. K. Kluge and S. Lakshmanan, "A Deformable Template Approach to Lane Detection," *Proc. IEEE Intelligent Vehicles 95*, IEEE Press, Piscataway, N.J., 1995, pp. 54-59.
7. M. Otte and H.-H. Nagel, "Optical Flow Estimation: Advances and Comparisons," *Proc. Third European Conf. Computer Vision*, Springer-Verlag, New York, 1994, pp. 51-60.
8. J. Kosecka, R. Bajcsy, and M. Mintz, "Control of Visually Guided Behaviors," in *Real-time Computer Vision*, C. Brown and D. Terzopoulos, eds., Cambridge Univ. Press, New York, 1994.
9. M. Bertozzi and A. Broggi, "GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection," *IEEE Trans. Image Processing*, 1997, to appear.
10. A. Broggi et al., "Design and Implementation of the PAPRICA Parallel Architecture," *J. VLSI Signal Processing*, 1997, to appear.

Massimo Bertozzi is a PhD student in information technology in the Dipartimento di Ingegneria dell'Informazione, Università di Parma, Italy. His research interests focus on the optimization of machine code and the application of image processing to vehicle guidance. He received a Dr. Ing. in electronic engineering from the Università di Parma, where he chairs the local IEEE student branch.

Alberto Broggi is an assistant professor in the Dipartimento di Ingegneria dell'Informazione at the Università di Parma, Italy. He is the main investigator for the university's Argo project, aimed at designing, developing, and testing an autonomous prototype vehicle. His research interests include real-time computer vision algorithms for unmanned vehicle navigation and the development of low-cost computer systems for autonomous robots. He is the guest editor for an upcoming issue of IEEE Expert *on vision-based driving assistance in vehicles of the future. Broggi received a Dr. Ing. in electronic engineering and a PhD in information technology from the Università di Parma.*

Contact the authors at the Università di Parma, Dipartimento di Ingegneria dell'Informazione, I-43100 Parma, Italy; {broggi, bertozzi}@ce.unipr.it.