Proceedings of the
IEEE Intelligent Vehicles Symposium 2000
Dearbon (MI), USA • October 3-5, 2000

Stereo Vision-based Vehicle Detection*

M. Bertozzi[1] A. Broggi[2] A. Fascioli[1] S. Nichele[2]

[1]Dipartimento Ing. dell'Informazione
parco area delle Scienze, 181A
43100, Parma
e-mail: {bertozzi,fascioli}@ce.unipr.it

[2]Dipartimento Informatica e Sistemistica
via Ferrata, 1
27100, Pavia
e-mail: alberto.broggi@unipv.it

Abstract

This paper presents the methods for sensing vehicles (localization and tracking) implemented on the ARGO vehicle. The perception of the environment is performed through the sole processing of images acquired from a stereo vision system installed on board of the vehicle.

1 Introduction

This work presents the Vehicle Detection functionality developed for the ARGO autonomous vehicle.

Automatic Vehicle Driving is a generic term referring to the techniques aimed at the entire or partial automation of some driving tasks. The functionalities that automatically driven vehicles should be able to perform include the possibility to follow the road and keep within the correct lane, maintaining a safe distance between vehicles, regulating the vehicle's speed according to environment and road conditions, moving across lanes in order to overtake vehicles and avoid obstacles, helping to find the correct and shortest route to a destination, and the movement and parking within urban environments.

Techniques used in the detection of obstacles may vary according to the definition of obstacles [1]. When the definition of obstacle is reduced to the specific, e.g. if obstacle means a vehicle, the detection can be based on a search for specific patterns, possibly supported by other features, such as texture [2], shape [3], symmetry [4, 5, 6], or the use of an approximant contour [7]. In this case the processing can be limited to the analysis of a single still image. While this approach has been widely demonstrated to be effective for a mere vehicle detection, it is difficult to accurately determine the vehicle distance. Moreover, in the case of single image processing, specific patterns on the scene (e.g. shadows, lane markings, or other artifacts on the road surface) can potentially deceive the vision system.

In the approach discussed in this work, vehicles are detected and tracked relying on a monocular image sequence; only a rough guess of the vehicle distance is computed using monocular vision. Then a validation of the result, as well as a distance refinement are computed using a simple stereo vision algorithm.

This paper is organized as follows: section 2 briefly depicts the ARGO project; section 3 presents the vehicle detection algorithm; results and timings performance are discussed in section 4; section 5 ends the paper with some final remarks.

2 The ARGO Project

ARGO is an experimental autonomous vehicle equipped with vision systems and featuring automatic steering capability [7]. The main target of the ARGO Project is the development of an active safety system which can also act as an automatic pilot for a standard road vehicle.

Initially, it was conceived as a safety enhancement unit: in particular it is able to supervise the driver behavior and issue both optic and acoustic warnings or even take control of the vehicle when dangerous situations are detected. Nevertheless, further developments have extended the system functionalities to automatic driving.

ARGO is able to determine its position with respect to the lane, to compute road geometry, to detect generic obstacles on the path, and to localize a leading vehicle.

Only passive sensors are used on ARGO to sense the surrounding environment: A stereoscopic vision system is installed at the top corners behind the windshield consisting of two low cost synchronized cameras able to acquire two stereo grey level images at once.

The architectural solution currently installed on the ARGO vehicle is based on a off-the-shelf 450 MHz Pentium II PC located into the boot. Images are analyzed in real-time by the computing system at a rate of 50 frames/s. The results of the processing are used to drive an actuator mounted

*This work was partially supported by the Italian National Research Council (CNR) in the framework of the MADESS2 Project.

onto the steering wheel and other driving assistance devices such as a pair of loudspeaker for acoustic messages.

A button-based control panel enables the driver to modify a few driving parameters, select the system functionality, issue commands, and interact with the system.

3 Vehicle Sensing

The Obstacle Detection functionality of ARGO, originally used also for detecting vehicles, has been proven to be not sufficiently accurate in computing obstacle distance for vehicles far away ahead of the vision system [8]. While the precision of vehicle's distance is not critical for *Road Following*, it is a key value for other functionalities like *Platooning*. Thus, a new algorithm, specifically tailored for Vehicle Detection, has been developed: the vehicle is localized and tracked using a single monocular image sequence while the correct distance is computed thanks to the availability of stereo images.

The Vehicle Detection algorithm is based on the following considerations: a vehicle is generally symmetric, characterized by a rectangular bounding box which satisfies specific aspect ratio constraints, and placed in a specific region of the image. These features are used to identify vehicles in the image in the following way: first an area of interest is identified on the basis of road position and perspective constraints. This area is searched for possible vertical symmetries; not only gray level symmetries are considered, but vertical and horizontal edges symmetries as well, in order to increase the detection robustness. Once the symmetry position and width have been detected, a new search begins, aimed at the detection of the two bottom corners of a rectangular bounding box. Finally, the top horizontal limit of the vehicle is searched for, and the preceding vehicle localized.

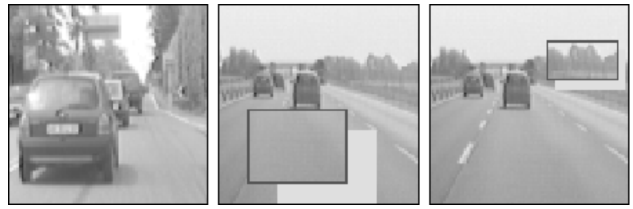
The tracking phase is performed through the computation of the correlation between the portion of the image contained into the bounding box of the previous frame (partially stretched and reduced to take into account small size variations due to the increment or reduction of the relative distance) and the new frame.

3.1 Vehicle Detection

3.1.1 Symmetry Detection

The analysis of gray level images is not sufficient for determining symmetrical features. Unfortunately, as figure 1 shows, strong reflections cause irregularities in vehicle symmetry, while uniform areas and background patterns may present highly correlated symmetries. In order to cope with these problems, also symmetries in other domains are computed.

To get rid of reflections and uniform areas, vertical and horizontal edges are extracted and thresholded, and symmetries are computed into these domains as well. Figure 2 shows that, although a strong reflection is present on the left side of the vehicle, edges are anyway visible and can be used to extract symmetries; moreover, in uniform areas no edges are extracted and therefore no symmetries are detected. Figure 3 shows two examples in which gray level symmetries alone can be successful for vehicle detection, while figure 4 shows the result of edge symmetry.

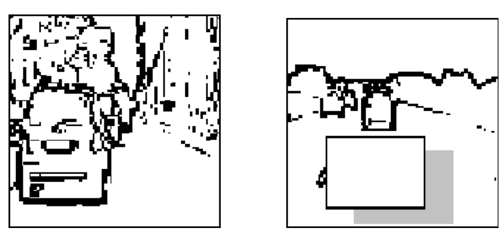


(a) (b) (c)

Figure 1: Typical road scenes: (*a*) a strong sun reflection reduces the vehicle gray level symmetry; (*b*) a uniform area can be regarded as a highly symmetrical region; (*c*) background symmetrical patterns.

For each image, the search area is shown in dark gray and the resulting vertical axis is superimposed. For each image its symmetry map is also depicted both in its original size and –on the right– zoomed for better viewing. Bright points encode the presence of high symmetries. The 2D symmetry maps are computed for different values of the axis' horizontal position within the grey area in the original image (horizontal axis) and the horizontal width of the symmetry area (vertical axis). The lower triangular shape is due to the limitation in scanning large horizontal windows for peripheral vertical axes.

Similarly, the analysis of symmetries of horizontal and vertical edges produces other symmetry maps, which –with specific coefficients detected experimentally– can be combined with the previous ones to form a single symmetry map. Figure 5 shows all symmetry maps and the final one,



(a) (b)

Figure 2: Edges enforce the detection of real symmetries: (*a*) strong reflections have lower effects while (*b*) uniform areas are discarded since they do not present edges.

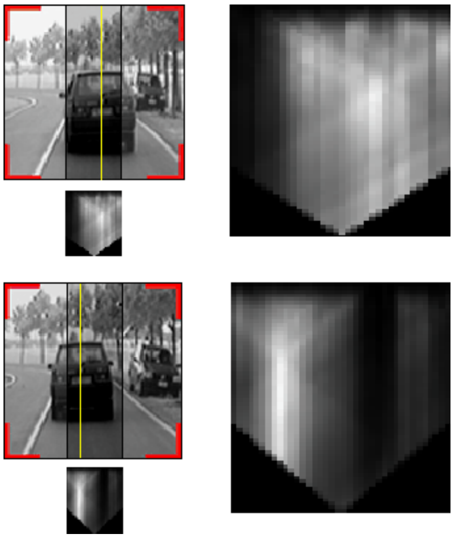


Figure 3: Grey level symmetries: the two rightmost images show the enlarged symmetry maps encoding high symmetries with bright points.

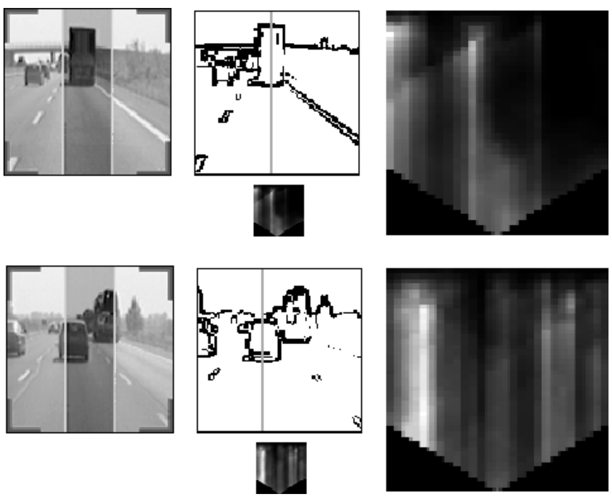


Figure 4: Edge symmetries: the symmetries are computed on the binary images obtained after thresholding the gradient image.

that allows to detect the vehicle.

3.1.2 Bounding Box Detection

The bounding box of the vehicle is detected through a search of its corners. Initially, the symmetrical region in the edge image is checked for the presence of two corners representing the bottom of the bounding box. Perspective and size constraints are used as search criteria. Figure 6 shows possible and impossible bottom parts of the bounding box, while figure 7 presents the results of the lower corners detection.

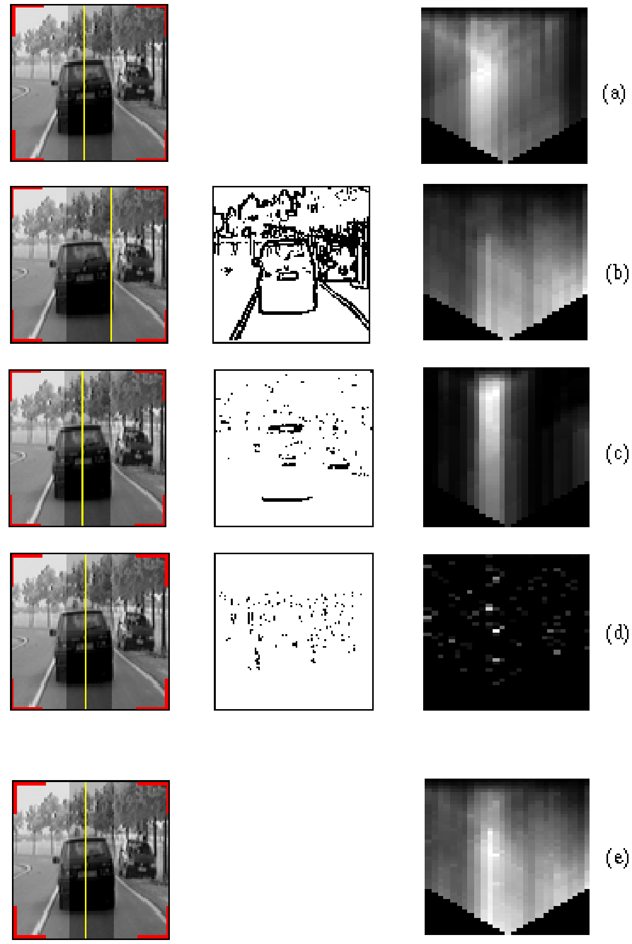
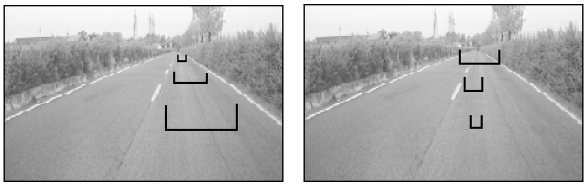


Figure 5: Computing the resulting symmetry: (*a*) grey-level symmetry; (*b*) edge symmetry; (*c*) horizontal edges symmetry; (*d*) vertical edges symmetry; (*e*) total symmetry. For each row the resulting symmetry axis is superimposed onto the leftmost original image.

This process is followed by the detection of the top part of the bounding box, which is looked for in a specific region whose location is again determined by perspective and size constraints. Figure 8 shows the search area.

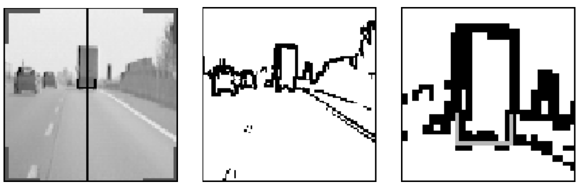
3.1.3 Backtracking

Sometimes it may happen that in correspondence to the symmetry maximum no correct bounding boxes exist; a backtracking approach is used: the symmetry map is again scanned for the next local maximum and a new search for a bounding box is performed. Figure 9 shows a situation in which the first symmetry maximum, generated by a building, does not lead to a correct bounding box; on the other hand, the second maximum leads to the correct detection of the vehicle.



(a) (b)

Figure 6: Detection of the lower part of the bounding box: (*a*) correct position and size, taking into consideration perspective constraints and knowledge on the acquisition system setup, as well as typical vehicles' size; (*b*) incorrect bounding boxes.



(a) (b) (c)

Figure 7: Detection of the lower part of the bounding box: (*a*) original image with superimposed results; (*b*) edges; (*c*) localization of the two lower corners.

Figure 8: The search area for the upper part of the bounding box is shown in dark gray. It takes into account knowledge about the typical vehicles' aspect ratio.

3.1.4 Distance Refinement

The distance to the leading vehicle is computed thanks to the knowledge of the camera calibration. Unfortunately, it may assume wrong values since it may happen that the lower part of the vehicle is not correctly detected. Sometimes, in fact, the luminance gradient of the region between the rear bumper and the chassis is so high to be misinterpreted as the lower part of the vehicle. In order to refine this measurement, which is of basic importance for the platooning functionality, an adjustment step is mandatory: it is performed taking advantage of stereo techniques. Starting from the distance value estimated from the left image, a portion of the right image is searched for a pattern similar to the one enclosed into the bounding box.

This step relies onto the following assumptions:

- the rear side of the vehicle is approximated as a vertical plane;

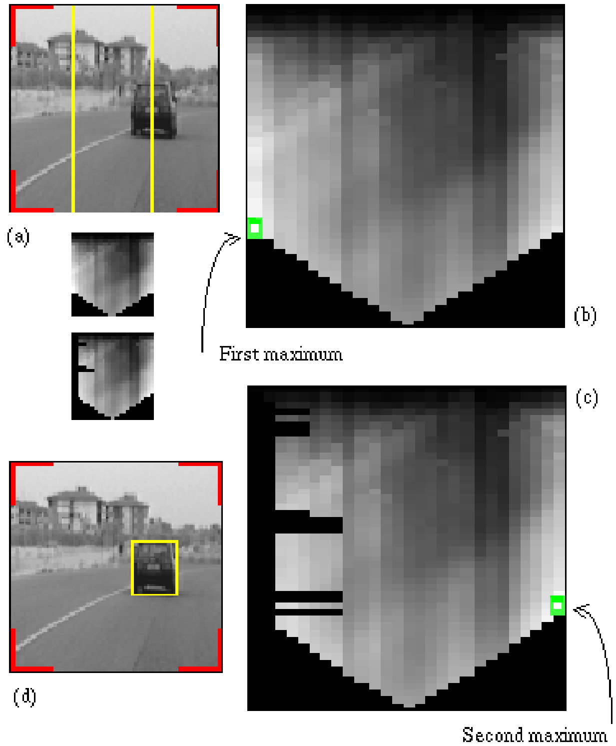


Figure 9: A case in which the background symmetry is higher than the vehicle symmetry: (*a*) original image; (*b*) first symmetry map; (*c*) second symmetry map after the backtracking process has removed the peak near the maximum; (*d*) final bounding box.

- luminance differences in the vehicle pattern, caused by light reflections, are negligible in the two views.

Since the correct calibration of the two cameras is known, once the same pattern enclosed into the bounding box is detected on the right image, a simple triangulation allows to detect the vehicle distance: the offset of the bounding boxes containing the vehicle, measured in both images, is used to compute the vehicle distance.

Besides being simpler than the traditional stereo-based techniques, this approach has the following advantages:

- it only requires one triangulation since the computation of the vehicle distance is the only final goal;

- errors are reduced to a minimum since the triangulation refers to a large and complex pattern whose identification is fairly easy;

- since not only the search pattern is known, but an estimate of the vehicle distance as well, the search is performed only in a reduced region of the image and therefore this step is not as computation intensive as traditional stereo techniques.

Figure 10 shows the steps used for distance refinement: figure 10.b shows the incorrect result of the detection step, figure 10.c shows that using a null offset the vehicle in the two images does not overlap, while figure 10.d shows that a specific offset brings the two rears to a perfect correspondence.

Figure 10: Distance refinement: (*a*) left and right stereo images; (*b*) incorrect result of the detection step (the lower part of the bounding box indicates a wrong distance); (*c*) superimposition of stereo images with a null offset; (*d*) superimposition of correctly shifted stereo images.

3.2 Vehicle Tracking

Temporal correlation amongst consecutive frames is used to track detected vehicles. In order to take in account fluctuations in size of the detected vehicle due to the increment or reduction of relative distance, a number of templates are computed expanding and reducing the portion of image that contains the vehicle (the bounding box). Each template is matched against the new frame and a correlation (C_T) is computed using the formula:

$$C_T(\triangle x, \triangle y) = \sum_{x=0}^{X_A} \sum_{y=0}^{Y_A} ((L(x,y) - R(x+\triangle x, y+\triangle y))^2$$

where:

- x and y are the pixel relative coordinates within the template,
- X_A and Y_A represent the template size,
- functions R and L return pixel intensity, and
- $\triangle x$ and $\triangle y$ are varied by shifting the template on the new frame in a area where the vehicle is expected to be found.

The minimum value of C_T identifies an area into the new frame where the vehicle is looked for.

	Vehicle Tracking	Vehicle Detection
Pentium 200MMX	24.8 ms	47.6 ms
Pentium II 450MMX	8.8 ms	19.9 ms
Speedup	2.8	2.4

Table 1: Timings of Vehicle Detection on two different architectures.

4 Results

Figure 11 shows some qualitative results of Vehicle Detection in different situations: the preceding vehicle is correctly detected at different distances, even on complex scenes.[1] Table 1 shows the timing performance on two different processing engines; due to the different computational burden of Vehicle Detection when looking for a vehicle or tracking an already found one, two distinct timings for Vehicle Detection and Tracking are shown.

5 Conclusions

In this work a vision-only technique for Vehicle Detection has been presented. The originality of this work lies in the exploitation of specific vehicle characteristics to obtain a robust search and tracking. The simple stereo distance

[1] A few sequences in MPEG format are available at http://millemiglia.ce.unipr.it/vdmpegs.html.

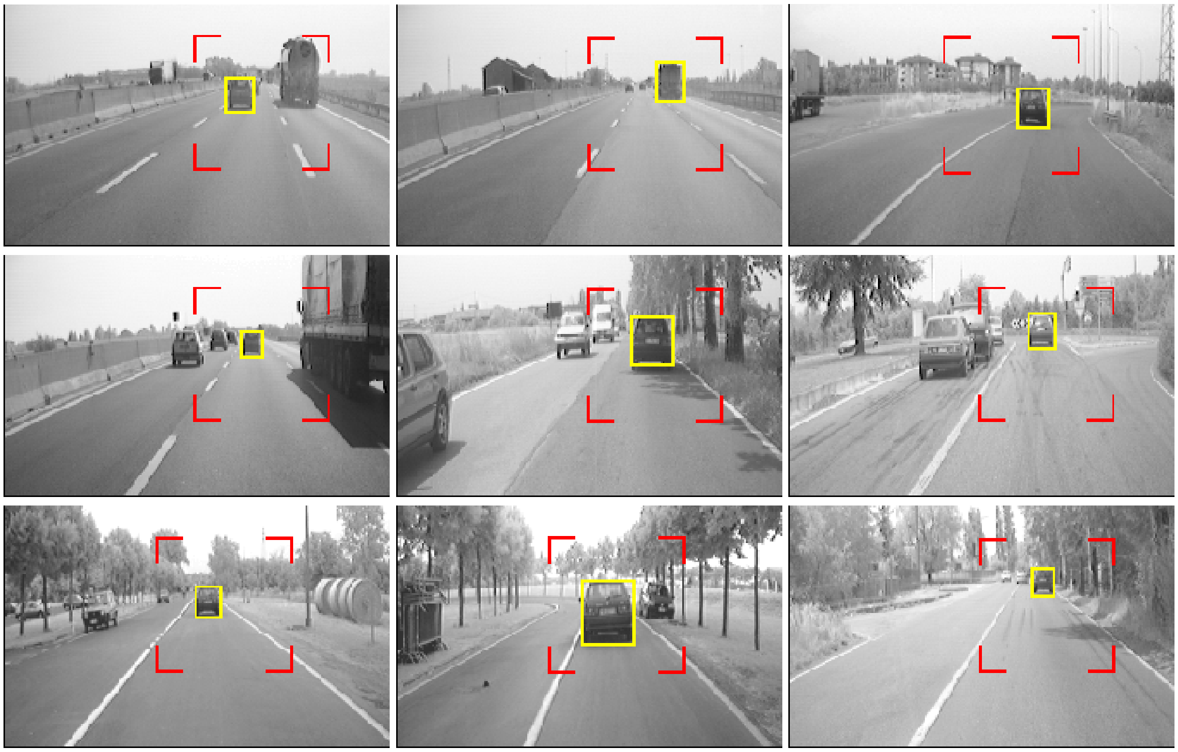


Figure 11: Vehicle Detection: the images show the search area and the detected vehicle with bright markings superimposed onto the original image.

refinement delivers precise results at a very low price in terms of computational power and time. Initially, vehicles are detected looking for specific patterns in monocular images. Since this first step could be affected by the presence of shadows or artifacts on the road, a second phase based on a simplified stereo vision algorithm is performed to validate results and to accurately compute vehicle distance.

The system was tested with images acquired by the stereo vision system installed on-board of the autonomous ARGO vehicle [7]. The tests demonstrated the system to be reliable and robust with respect to noise caused by shadows, different road textures, or varying illumination conditions.

With very few modifications, the algorithm presented in this work has been successfully adapted for detecting and tracking pedestrians. In a very similar manner, in fact, a pedestrian is treated as an almost-symmetrical object with a specific aspect ratio.

References

[1] M. Bertozzi and A. Broggi, "GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection," *IEEE Trans. on Image Processing*, vol. 7, pp. 62–81, Jan. 1998.

[2] T. Kalinke, C. Tzomakas, and W. von Seelen, "A Texture-based Object Detection and an Adaptive Model-based Classification," in *Procs. IEEE Intelligent Vehicles Symposium'98*, (Stuttgart, Germany), pp. 341–346, Oct. 1998.

[3] F. Thomanek, E. D. Dickmanns, and D. Dickmanns, "Multiple Object Recognition and Scene Interpretation for Autonomous Road Vehicle Guidance," in *Procs. IEEE Intelligent Vehicles Symposium'94*, (Paris), pp. 231–236, Oct. 1994.

[4] G. Marola, "Using Symmetry for Detecting and Locating Objects in a Picture," *Computer Vision Graphics and Image Processing*, vol. 46, pp. 179–195, May 1989.

[5] 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.

[6] A. Kuehnle, "Symmetry-based vehicle location for AHS," in *Procs. SPIE - Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS*, vol. 2902, (Orlando, FL), pp. 19–27, Nov. 1998.

[7] A. Broggi, M. Bertozzi, A. Fascioli, and G. Conte, *Automatic Vehicle Guidance: the Experience of the ARGO Vehicle*. World Scientific, Apr. 1999. ISBN 981-02-3720-0.

[8] M. Bertozzi, A. Broggi, and A. Fascioli, "Performance Analysis of a Low-Cost Solution to Vision-Based Obstacle Detection," in *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems'99*, (Tokyo, Japan), pp. 350–355, Oct. 1999.