Stereo Vision-based Feature Extraction for Vehicle Detection

A. Bensrhair, M. Bertozzi, A. Broggi, A. Fascioli, S. Mousset, and G. Toulminet

Abstract— **This paper presents a stereo vision system for vehicle detection. It has been conceived as the integration of two different subsystems. Initially a stereo vision based system is used to recover the most relevant 3D features in the scene; due to the algorithm's generality, all the *vertical* features are extracted as potentially belonging to a vehicle in front of the vision system. This list of significant patterns is fed to a second subsystem based on monocular vision; it processes the list computing a match with a general model of a vehicle based on symmetry and shape, thus allowing the identification of the sole characteristics belonging to a vehicle.**

The system presented in this work derives from the integration of the research work developed by the University of Parma (Italy) and I.N.S.A. of Rouen (France). The two subsystems have been integrated into the GOLD software and are currently under testing using the ARGO experimental vehicle.

Keywords— **stereo vision, feature extraction, vehicle detection.**

I. Introduction

A widely used approach for vision-based vehicle detection is the search for specific patterns [1], for example: shape [2, 3, 4], symmetry [5, 6, 7, 8], texture [9], the use of an approximant contour, or the use of a specific model [10, 11].

In these cases, the processing can be entirely based on monocular vision. Anyway, a major problem still remains open: while the vehicle detection can be effective, the distance of detected vehicles cannot be accurately computed without the aid of other sensors, unless a flat road is assumed. Moreover, in the case of single image processing, noisy patterns on the scene (e.g. horizontal signs, concrete textures, or other artifacts on the scene) can potentially confuse the vision system introducing supplementary edges or textures and leading to incorrect results.

This paper introduces a stereo vision feature detection algorithm specifically tailored for Vehicle Detection. Compared to a traditional stereo-vision algorithm the discussed approach is not aimed at a complete 3D world reconstruction but to the mere extraction of features potentially belonging to a vehicle, namely only 3D vertical edges. The list of features is intended to be used by a monocular vision system that performs Vehicle Detection by means of a match with a vehicle model. Anyway, in this case the system can draw advantages from having additional information on edges' distances from the camera and from working on actually vertical characteristics, i.e. without misinterpretations caused by artifacts or road infrastructures. Therefore, besides a more reliable detection, also an accurate estimation of vehicle distance can be obtained.

The system presented in this work derives from the integration of the research work developed by the University of Parma (Italy) and I.N.S.A. of Rouen (France). Both systems have been integrated into the GOLD software and tested on ARGO, an experimental vehicle equipped for testing vision algorithms and autonomous driving [12].

This paper is organized as follows. Section 2 introduces the system used for developing the stereo vision algorithm, while section 3 describes the system used for testing the algorithm. Section 4 details the algorithm for the extraction of the features which will be used for vehicle detection as described in section 5. Section 6 ends the paper presenting some final remarks.

A. Bensrhair, S. Mousset, and G. Toulminet are with the Université de Rouen et INSA de Rouen, FRANCE. E-mail: {abdelaziz.bensrhair, stephane.mousset, gwenaelle.toulminet}@insa-rouen.fr.

M. Bertozzi, A. Broggi, and A. Fascioli are with the Dip. di Ingegneria dell'Informazione, Università di Parma, ITALY. E-mail: {bertozzi,broggi,fascioli}@ce.unipr.it.

II. Development set-up

The I.N.S.A. of Rouen has designed a passive stereovision sensor made up of a rigid body, two similar lenses and two Philip VMC3405 camera modules whose centers are separated by 12.7 cm (figure 1). An Imaging Technology PC-RGB frame grabber, installed into a Pentium III 800 MHz with Windows OS, controls these two cameras, and acquires simultaneously two images (720×568 or 720×284 pixels). Furthermore, the two camera-lens units are set up so that their optical axes are parallel and, in order to respect an epipolar constraint, the straight line joining the two optical centres is parallel to each images horizontal line. In order to respect these geometry constraints, lenses with sub-pixel geometry distortion have been applied. The Newton ring method has been used to verify the geometric parallelism of the two optical axis. Finally, collinearity of images' lines is set up by micrometer screw with sub-pixel precision : the calibration procedure is based on the use of two reference targets whose position is known. Based on this configuration, depth information is given in meters by:

$$Z = \frac{f \times e}{p \times \delta} \tag{1}$$

where e is the distance between the two optical centres, p is the width of the CCD pixel, f is the focal length of the two lenses, δ is given in pixels and is the horizontal disparity of two stereo-corresponding points. Let P_L and P_R be two stereo-corresponding points of a 3D point P of an object. Let (X_L, Y_L), (X_R, Y_R) and (X, Y, Z) be their coordinates. (X_L, Y_L) and (X_R, Y_R) are given in pixels, (X, Y, Z) is given in meters. Then, due to the epipolar configuration $Y_L = Y_R$ and $\delta = (X_R - X_L)$.

III. Testing set-up

In order to extensively test the system developed in laboratory by I.N.S.A., it was integrated into the ARGO prototype vehicle of the University of Parma [13].

ARGO is an experimental autonomous vehicle equipped with a stereovision system and automatic steering capabilities. It is able to determine its position with respect to the lane, to compute road geometry, to detect generic obstacles and pedestrians on the path, and localize a leading vehicle.

Fig. 1 The sensor of I.N.S.A. installed into ARGO.

The images acquired by a stereo rig placed behind the windshield are analyzed in real-time by the computing system located into the boot. The results of the processing are used for a number of driving assistance functions and to drive an actuator mounted onto the steering wheel.

The system is able to maintain the full control of the vehicle's trajectory, and two functionalities can be selected: **Road Following**, the automatic movement of the vehicle inside the lane, and **Platooning**, the automatic following of the preceding vehicle, which requires the localization and tracking of a target vehicle.

Initially only the I.N.S.A. sensor was installed on the ARGO vehicle to test the hardware system in real outdoor scenarios and in different conditions of light, weather and traffic. Thanks to these experiments the feature extraction algorithm has been enhanced and strengthened.

Subsequently, the algorithm was integrated into the GOLD system. GOLD is the software that provides ARGO with intelligent capabilities. It includes Generic Obstacles and Lane Detection, the two functionalities originally developed, and it integrates two other functionalities: Vehicle Detection and Pedestrian Detection. The feature extraction algorithm previously implemented in Rouen has been ported to the GOLD system and is currently included as an additional functionality. The final objective is the integration of this algorithm into the Vehicle Detection functionality as a preprocessing procedure, as explained in section 5.

The integration into the GOLD system allowed to test the algorithm also on the images acquired by the ARGO vision system. The stereoscopic system used on ARGO to sense the surrounding environment consists of two synchronized low cost cameras able to acquire pairs of grey level images simultaneously. The resolution of the images they provide is 768×576 or 768×288 pixels. To permit the detection of far objects, the two cameras were placed inside the vehicle at the top corners of the windshield, so that the longitudinal distance between them is maximum (95 cm). This distance is therefore significantly larger than the I.N.S.A. system's baseline. Moreover, while the two cameras of the I.N.S.A. sensor are set up so that their optical axes are parallel, the cameras installed on ARGO can be independently moved with respect to their pan, tilt and roll angles, only their distance is fixed. Hence a minor precision can be obtained in having parallel optical axes.

Since the process is based on stereo vision, camera calibration plays a fundamental role in the success of the approach. The ARGO calibration process is based on a grid with a known size painted onto the ground. Two stereo images are captured and used for the calibration. The image coordinates of the intersections of the grid lines are manually provided by the user. These intersections represent a small set of points whose world coordinates are known to the system: this mapping is used to compute the calibration parameters. The small number of homologous points, the human imprecision which may affect their coordinates, and the low resolution of the portion of the images representing far away intersections limit the accuracy of this calibration procedure, in particular with respect to the precision achieved by the I.N.S.A. calibration process.

IV. 3D Features extraction

Within the framework of road obstacles detection, road environment can be modeled by a class *Road* and a class *Obstacle*. In order to feed the vehicle detection functionality, 3D edge shapes are first constructed from 3D sparse maps, then they are identified as road edges or obstacles edges. Finally, the 3D shapes of the class *Obstacle* are extracted.

A. Construction of 3D edge shapes

A.1 Construction of 3D sparse maps

The algorithm of 3D sparse maps construction is a line by line processing designed for the configuration of the vision sensor of the INSA of Rouen. In the first step of the algorithm, the edge points of the right and left images are segmented by self-adaptive and mono-dimensional operator, the *declivity*. In a second step, the edge points of the right image are matched with the edge points of the left image, using a dynamic programming method. The matching algorithm provides depth information (equation 1), based on the positions of the left and right edge points. The result of the matching algorithm is a 3D sparse map. The evaluation of this result obtained from the processing of a pair of images of an inside scene is : 92.6% of the right edge points are associated with a left edge point. And among these associations 98% are correct [14].

A.2 Improvement of 3D sparse maps

Using criteria related to road environment, the improvement algorithm matches right edge points that have not been matched. It also detects and corrects wrong edge points associations. Because road environment is structured, the edges of road scenes are smooth 3D shapes. In the first step of the improvement algorithm, 3D shapes are built based on the result of the segmentation on the right image and the result of the matching algorithm. In a second step, we suppose that most of edge points associations are correct. And if the coordinates of a 3D point belonging to a 3D shape don't validate a smoothing criteria, then this 3D point is the result of a wrong edge points association. In the last step, a left edge point validating a smoothing criteria is searched for each right edge point that has not been matched or that has been wrongly matched. In the following sections, the steps of the improvement algorithm are described.

Construction of 3D shapes: an actual 3D edge shape can be constructed using its projections in the right and left images.

The construction of 3D shapes starts with the construction of their projections in the right image. The result of the matching algorithm provides the estimations of their projections in the left image.

Characteristics of 2D right shapes construction: by means of a line by line processing, 2D right shapes are made based on right edge points so that:

- a right edge point belongs to one and only one 2D right shape
- a 2D right shape starting at line l_s and ending at line l_e ($l_e \geq l_s$), has one and only one point on each line between l_s and l_e

Algorithm of 2D right shapes construction : At the first line of the right image, each edge point that has a stereo-correspondent generates a 2D shape. For each edge point of other lines of the image, the following steps are performed.

step 1 : Let d_r be a right edge point whose coordinates in the right image are (l, p). The set R is constructed with edge points whose coordinates in the right image are $(l-i, p+j)$, with $j \in \{-2,-1,0,1,2\}$ and $i \in \{1,2\}$.

step 2 : A priority level is computed for each set $\{d_r, d_{rR}\}$ with $d_{rR} \in R$. The priority level evaluates the extension by d_r of the shape to which d_{rR} belongs to. For this computation

- we use the coordinates of d_r and d_{rR} in the right image. And if the stereo-correspondents of d_r and d_{rR} both exist, then we use their coordinates in the left image
- we take into account the characteristics of the 2D shapes construction

step 3 : The highest priority level of shape extension is considered. Let S be the shape that must be extended.

step 4 : If a highest priority level of extension of a shape has been computed, then d_r and eventually a point at line $(l-1)$ extend S. Otherwise, if d_r has a stereo-correspondent, it generates a new 2D shape.

Detection of wrong edge points associations: let S_r and S_l be respectively the projection in the right image and the estimation of the projection in the left image of an actual 3D shape. At the beginning of the detection algorithm, we supposed that all edge points associations of all 3D shapes are certain. Then, the detection algorithm is applied on each 2D shape S_l, and is divided in two steps.

Step 1 : The first step aims to detect uncertain edge points associations. It is applied on each point d_l of S_l that is not the first point of S_l. Let (l, p_{d_l}) be the coordinates in the left image of d_l. If there is a point d_{lp} of S_l whose coordinates in the left image are $(l-1, p_{d_l}+j)$ with $j \in \mathcal{N} - \{-\in, -\infty, \prime, \infty, \in\}$, then $\{d_r, d_l\}$ and $\{d_{rp}, d_{lp}\}$ are uncertain edge points associations.

Step 2 : The second step aims to detect among the uncertain edge points associations the ones that are wrong edge points associations. It is applied on each point u_l of S_l whose association with its right stereo-correspondent u_r is uncertain. Let $\{c_{rp}, c_{lp}\}$ and $\{c_{rf}, c_{lf}\}$ be two certain 3D points. c_{lp} and c_{lf} belong to S_l and their coordinates in the left image are $(l-i_p, p_p)$ and $(l+i_f, p_f)$ with $i_p \in \{1,2\}, i_f \in \{1,2\}$ and $(i_p+i_f) \leq 3$. If c_{lp} and c_{lf} exist a constraint is defined between u_l, c_{lp} and c_{lf}. If a smoothing criteria is validated then $\{u_r, u_l\}$ is a certain 3D point. Otherwise, $\{u_r, u_l\}$ is a wrong 3D point. If c_{lp} or c_{lf} doesn't exist then $\{u_r, u_l\}$ is a wrong 3D point.

Correction of wrong edge points associations: the correction algorithm is applied on each point w_r of S_r that has not been matched or wrongly matched. Let (l, p) be its coordinates in the right image. Let $\{c_{rp}, c_{lp}\}$ and $\{c_{rf}, c_{lf}\}$ be two certain 3D points. c_{rp} and c_{rf} belong to S_r and their respective coordinates in the right image are $(l-1, p_p)$ and $(l+1, p_f)$. If $\{c_{rp}, c_{lp}\}$ and $\{c_{rf}, c_{lf}\}$ both exist, then we look for a left edge point w_l that has not been matched or wrongly matched so that a constraint defined between w_l,c_{lp} and c_{lf} validates a smoothing criteria. If w_l exists, then w_l is the correct left stereo-correspondent of w_r.

B. Identification of 3D shapes

In order to identify the 3D shapes previously computed as road edges or obstacles edges, two methods cooperate. The first one selects 3D shapes by thresholding the disparity value of their 3D points. The second one selects 3D straight segments by thresholding their inclination angle. From these two selection results, 3D shapes that are obstacles edges are identified. The 3D shapes that don't belong to the class *Obstacle*, belong to the class *Road*.

B.1 3D shapes selection by thresholding disparity values

Using the principles of the *Inverse Perspective Mapping* [15], each pixel of an image can be associated to a disparity value, provided that the position, the orientation, the angular aperture and the resolution of the camera are known, and supposing that the road is flat. The orientation of our right camera allow us to associate each line l of the right image to a disparity value $disp(l)$. The function $disp$ represents the disparity of the road and is used as a threshold function to select 3D points from the improved 3D map : if the disparity of a correct 3D point $\{d_r, d_l\}$ of line l is higher than $disp(l)$, then $\{d_r, d_l\}$ is selected because it is supposed to belong to an obstacle.

Finally, sets of selected 3D points are constructed so that the three following propositions are validated.

Proposition 1 : the 3D points of a set belong to the same 3D shape S_{3D}.

Proposition 2 : the 3D points of a set follow one another in S_{3D}.

Proposition 3 : a set has a minimal number of points.

The constructed sets are portions of the 3D shapes constructed in section IV-A.2. They are the 3D shapes selected.

B.2 3D segments selection by thresholding inclination angle

As road environment is structured, the 3D shapes constructed in section IV-A.2 can be approximated by means of one or several 3D straight segments. By an iterative partition method, 3D shapes are decomposed into 3D segments. In order to select 3D segments that belong to road obstacles, we suppose that the road is flat, then we calculate and threshold the inclination angles of 3D segments. For this calculation, we use the equations of the projections of each 3D segment in the right and left images. The equation of the right projection is calculated in $(R_r X_r Y_r)$ (figure 2), and the equation of the left projection is calculated in $(R_l X_l Y_l)$. They are :

$$x_r = m_r \times y + b_r \qquad x_l = m_l \times y + b_l \tag{2}$$

with m_r, m_l, b_r, b_l calculated by a least square method. Note that the improvement of 3D sparse maps is of paramount importance for the calculation of m_l and b_l, and so for a reliable estimation

of the inclination angle. Using geometry propriety, the tangent of the inclination angle of a 3D segment is given by the equation 3.

$$\tan\beta = \frac{p_y\left((b_l - b_r) + \frac{h}{2}(m_l - m_r)\right)}{\sqrt{p_x^2\left((m_r b_l - m_l b_r) - \frac{w}{2}(m_r - m_l)\right)^2 + f^2(m_r - m_l)^2}} \quad (3)$$

p_x and p_y are the width and height of the CCD pixel, f is the focal length of the two lenses, and $w \times h$ is the resolution in pixels of the cameras. The 3D segments whose inclination angle is higher than a threshold angle are selected.

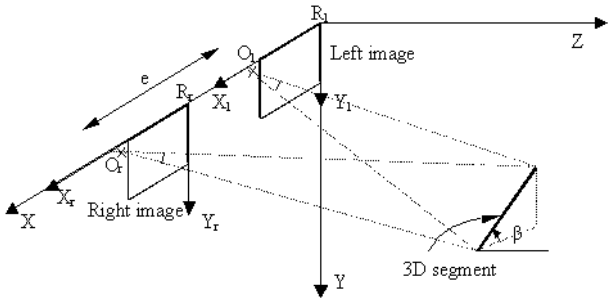


Fig. 2 The inclination angle β of a 3D segment.

B.3 Selection of obstacles edges

Let F_1 be the set of 3D shapes that have been selected by thresholding the disparity value of their 3D points. Let F_2 be the set of 3D segments that have been selected by thresholding their inclination angle. And, let F be the set of 3D shapes identified as edges of obstacles. In a first step of the construction of F, F contains the 3D shapes of $F_1 \cap F_2$ that has a minimum number of point. In a second step, an array T of dimension h is filled with the lowest disparity values of the 3D shapes of F. h is the height in pixels of the image, and the lowest disparity value of a 3D shape $disp_{min}$ is defined as the lowest disparity value of its certain 3D points. Here again, the improvement of 3D sparse map is of paramount importance. For each 3D shape and for each line l with $l \geq 0$ and $l \leq l_{max}$, if $disp_{min} < T(l)$ then $T(l) = disp_{min}$. l_{max} is the line that verifies $disp(l_{max}) \leq disp_{min}$ and $disp(l_{max}+1) > disp_{min}$ with the function $disp$ calculated in the section IV-B.1. In the last step, the array T is used to extend F with the 3D shapes of $F_1 \cup F_2$: if a 3D shape S_{3D} of $F_1 \cup F_2$, has a minimum number of certain 3D points whose disparity is higher than the threshold values defined in T, then S_{3D} extends F.

C. Experimental results

Figure 3 shows some experimental results of the extraction of 3D edge shapes identified as edges of obstacles. Column (a) shows the segmentation results of the right image superimposed on the original image. Column (b) displays the improved sparse depth map. The higher the grey level the lower the depth. Column (c) displays the results of the construction of 3D shapes. Column (d) presents the results of the extraction of obstacles' 3D edge shapes, superimposed on the acquired images.

The average processing time of a pair of stereo images on a PC Pentium III 450 MHz with Linux is 960 ms. It is important to be noticed that the code has not been optimized yet.

V. Exploiting 3D features in the Vehicle Detection functionality

This sections briefly presents the vehicle detection scheme originally implemented on the ARGO vehicle [8], and describes how the stereo-based feature extraction process described in this paper can help this functionality.

The platooning functionality (the automatic following of the preceding vehicle) developed in the last few years by the group of Parma University and tested on the ARGO experimental vehicle was originally based on the localization of ahead obstacles, but it was demonstrated not to be robust enough to allow smooth vehicle following. A specific vision-based functionality was therefore explicitly designed to support automatic platooning, based on the localization of the ahead vehicle only.

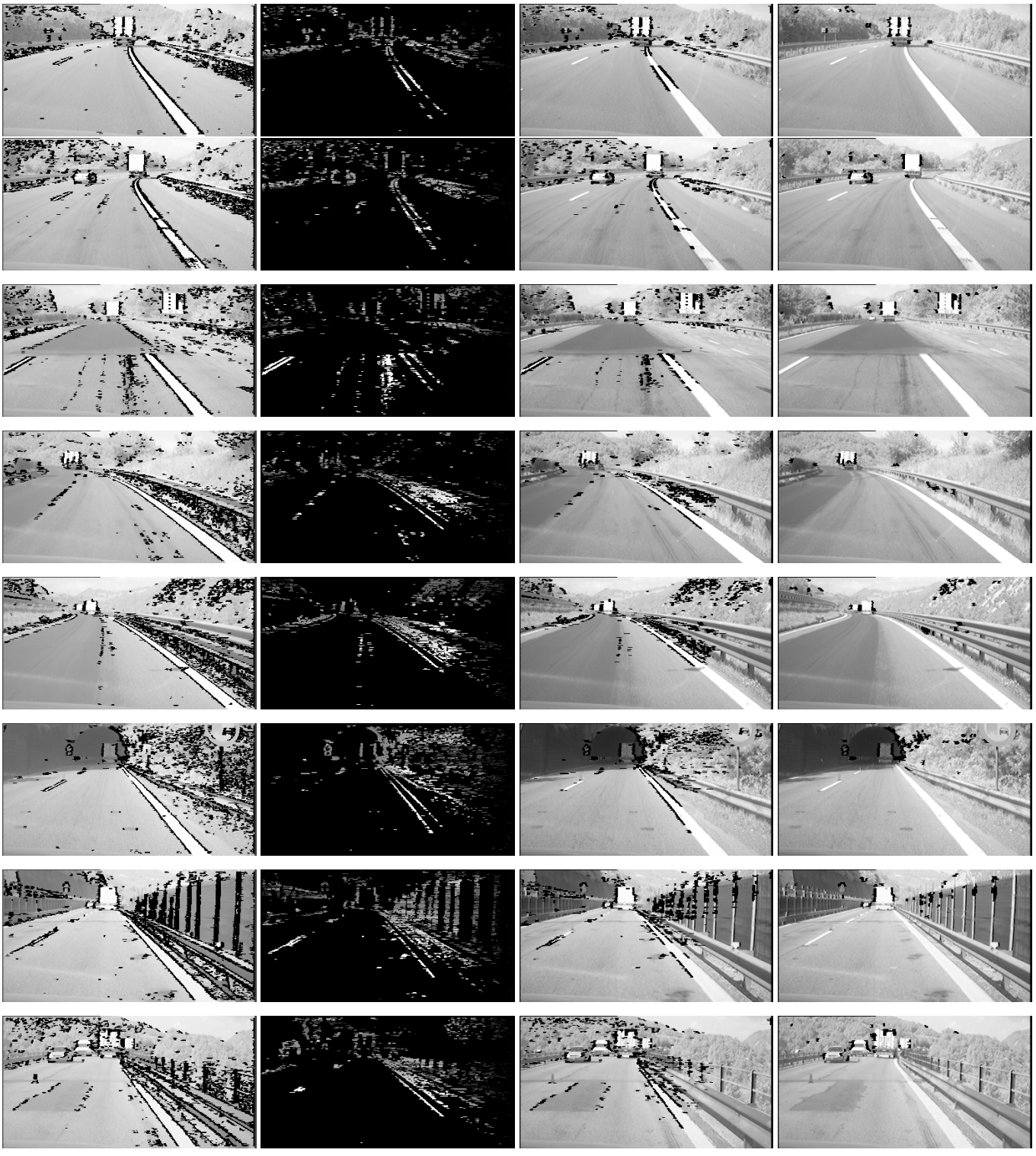
The relative position, speed, direction of the vehicle to be followed is computed by analyzing a monocular sequence of images coming from a single camera; then a refinement of this data is computed by a high-level matching of some selected features in a stereo view of the same scene. In other words, the main localization process is performed with low-level techniques (based on the processing of collections of pixels), while the refinement is done using a higher level stereo matching technique. The main advantage of this approach is that, though less computationally expensive (the main processing is done on a single image), it is still able to recover 3D information by using stereography at a later stage. Unfortunately, the use of monocular images (with a lower information content than stereo frames) in the low-level process may lead to incorrect labelling of important features.

Vehicle detection is in fact based on the following steps:

1. grey-level symmetry detection
2. horizontal and vertical edges extraction
3. edges symmetry detection
4. localization of vehicle's bottom and top borders
5. check on size and distance using perspective constraints associated with a correct camera calibration
6. search for the same bounding box in the other stereo view in order to compute a correct distance.

The above steps work correctly in many situations, but there are cases in which an unfriendly environment may lead the algorithm to false detections. For example, figure 4 shows some false detections.

In the above situations and in some other cases in which objects at different distances may be erroneously grouped together, the stereo-based feature extraction process previously described may help. Since it is able to discriminate 3D vertical edges and compute edges' distance from the camera, this knowledge can be exploited in the computation of edges symmetries. In other words, some features may be filtered out so that symmetries can be computed only on edges that actually represent 3D vertical objects, and edges that do not lie at similar distances will not



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

Fig. 3 Experimental results : (a) edge points of the right image, (b) improved 3D sparse map, (c) 3D edge shapes, (d) 3D edge shapes of obstacles

be matched together. Moreover a first check on vehicle's size with respect to its distance can be performed at this early stage: candidates of vehicles' left and right vertical edges can be filtered out if they represent an object too narrow or too large with respect to a vehicle lying at the distance estimated by the stereo-based procedure.

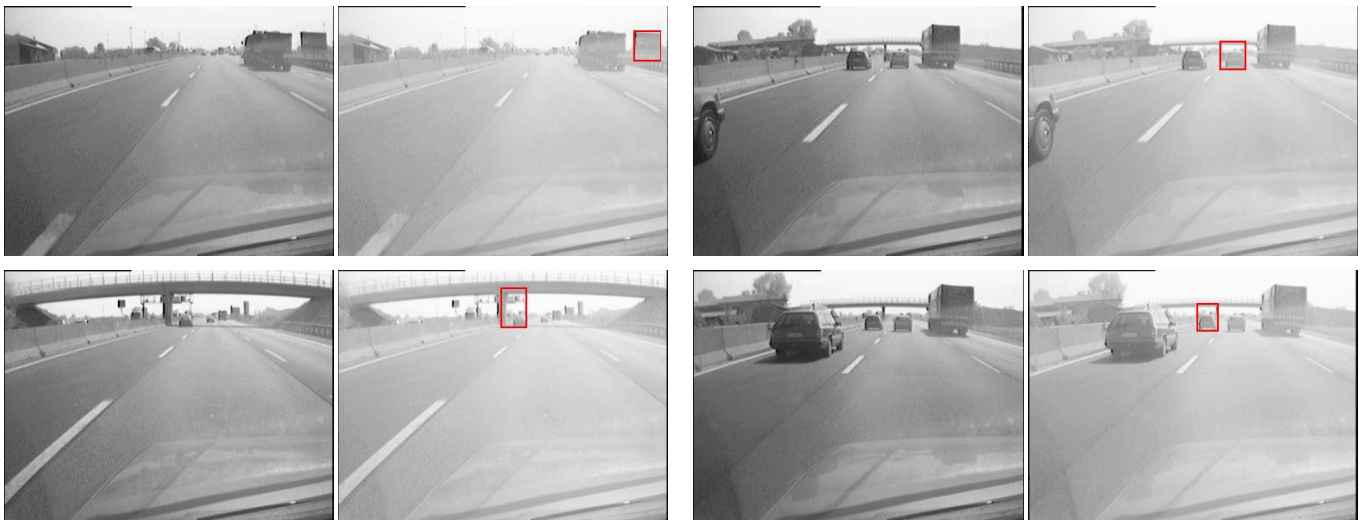


Fig. 4 Situations in which a mere monocular processing may yield to false detections: original images (on the left) and brighter versions (on the right) of them with superimposed the wrong result.

VI. Conclusions

This work presents the results of a joint research collaboration between the University of Parma and I.N.S.A. of Rouen. The cooperation is aimed at integrating sensors and algorithms for vehicle detection and testing them on a real vehicle prototype.

The stereoscopic sensor and algorithm initially developed by the French party were first tested on the ARGO vehicle. Then a tighter integration of the two systems allowed to test the feature extraction algorithm using images coming from the sensors of ARGO. Thanks to the promising and encouraging results, the final target will be the integration with the vehicle detection capability already available on ARGO.

Acknowledgments

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

References

[1] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based Intelligent Vehicles: state of the art and perspectives," *Journal of Robotics and Autonomous Systems*, vol. 32, pp. 1–16, June 2000.

[2] S. M. Smith, "ASSET-2: Real-time motion segmentation and object tracking," *Real Time Imaging Journal*, vol. 4, pp. 21–40, Feb. 1998.

[3] G. S. K. Fung, N. H. C. Yung, and G. K. H. Pang, "Vehicle Shape Approximation from Motion for Visual Traffic Surveillance," in *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems 2001*, (Oakland, USA), pp. 610–615, Aug. 2001.

[4] 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.

[5] S. Kyo, T. Koga, K. Sakurai, and S. Okazaki, "A robust Vehicle Detecting and Tracking System for Wet Weather Conditions using the IMAP-VISION Image Processing Board," in *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems'99*, (Tokyo, Japan), pp. 423–428, Oct. 1999.

[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, USA), pp. 19–27, Nov. 1998.

[7] 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.

[8] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, "Stereo Vision-based Vehicle Detection," in *Procs. IEEE Intelligent Vehicles Symposium 2000*, (Detroit, USA), pp. 39–44, Oct. 2000.

[9] 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.

[10] S. Denasi and G. Quaglia, "Obstacle Detection Using a Deformable Model of Vehicles," in *Procs. IEEE Intelligent Vehicles Symposium 2001*, (Tokyo, Japan), pp. i145–150, May 2001.

[11] M. Lützeler and E. D. Dickmanns, "Road Recognition with MarVEye," in *Procs. IEEE Intelligent Vehicles Symposium'98*, (Stuttgart, Germany), pp. 341–346, Oct. 1998.

[12] A. Broggi, M. Bertozzi, A. Fascioli, and G. Conte, *Automatic Vehicle Guidance: the Experience of the ARGO Vehicle*. World Scientific, Apr. 1999. ISBN 9810237200.

[13] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miché, S. Mousset, and G. Toulminet, "A Cooperative Approach to Vision-based Vehicle Detection," in *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems 2001*, (Oakland, USA), pp. 209–214, Aug. 2001.

[14] A. Bensrhair, P. Miché, and R. Debrie, "Fast and automatic stereo vision matching algorithm based on dynamic programming method," *Pattern Recognition Letters*, vol. 17, pp. 457–466, 1996.

[15] M. Bertozzi, A. Broggi, and A. Fascioli, "Stereo Inverse Perspective Mapping: Theory and Applications," *Image and Vision Computing Journal*, vol. 8, no. 16, pp. 585–590, 1998.