Vehicle and Guard Rail Detection Using Radar and Vision Data Fusion

Giancarlo Alessandretti, Alberto Broggi, *Senior Member, IEEE*, and Pietro Cerri, *Member, IEEE*

***Abstract*—This paper describes a vehicle detection system fusing radar and vision data. Radar data are used to locate areas of interest on images. Vehicle search in these areas is mainly based on vertical symmetry. All the vehicles found in different image areas are mixed together, and a series of filters is applied in order to delete false detections. In order to speed up and improve system performance, guard rail detection and a method to manage overlapping areas are also included. Both methods are explained and justified in this paper. The current algorithm analyzes images on a frame-by-frame basis without any temporal correlation. Two different statistics, namely 1) frame based and 2) event based, are computed to evaluate vehicle detection efficiency, while guard rail detection efficiency is computed in terms of time savings and correct detection rates. Results and problems are discussed, and directions for future enhancements are provided.**

***Index Terms*—Fusion, radar, vehicle detection, vision.**

I. Introduction

DIFFERENT preventive safety functions are now introduced on road vehicles to assist the driver and to reduce the risk of accidents. Key points for improved operation are the effectiveness and the information content in the perception of the surrounding scenario, including road features and obstacles. Radar–vision fusion is an interesting approach that is often based on complementary devices, which can provide several advantages: in particular, improved reliability from multiple detections and the merging of position measures with good longitudinal and lateral accuracy. The purpose of this paper is to investigate how the results of different sensors can be fused together, benefiting from the best performance of each sensor.

The advantages and problems of fusing radar and camera data for vehicle detection are well known [1]. The methods differ mainly for the fusion level: Low-, intermediate-, and high-level fusions have all gotten good results. Low-level fusion combines several sources of raw data to produce new raw data that are expected to be more informative and synthetic than the inputs [2]. In intermediate-level fusion, various features such as edges, corners, lines, texture parameters, etc., are combined into a feature map that is then used by further processing stages. In high-level fusion, each source of input yields a decision, and all the decisions are combined [3].

This paper is developed using high-level fusion and focuses on the validation of radar targets, as shown by Sole [4]. In this context, radar targets can correspond to a vision target, in our case a vehicle, or not: Different vision algorithms can be used for this purpose.

Some methods use models to identify a vehicle. Many different models have been used, ranging from trivial models to complex ones, for example, deformable models that add details of the car approaching the camera [5], or three-dimensional (3-D) models that take into consideration vehicle misalignment with the camera [6]. All these methods need models that match different vehicle types.

The search for vehicle features provides a simplified way of localizing vehicles. For example, symmetry is a characteristic that is common to most vehicles. Some research groups have already used symmetry to localize vehicles [7], [8] and tested a variety of methods to find symmetry on images using edges, pixel intensity, and other features.

The vehicle detection algorithm used in this paper is based on symmetry [9] and uses radar data in order to localize areas of interest. Data fusion operates at high level: The vision system is used to validate radar data and to increase the accuracy of the information they provide.

Two different setups, with two different radars, have been tested. The first one is a long-range radar with a 77-GHz frequency that is not capable of data classification. This kind of radar provides a large number of nonvehicle objects (mainly guard rails). A guard rail detection system is then mandatory in order to reduce false positives and speed up the processing.

The second setup uses two scanned radars with a 24-GHz frequency mounted above the front bumper and connected to a dedicated Electronic Control Unit (ECU). In this case, only obstacles up to 40 m can be detected with an accuracy of 0.25 m. The radar can compute the relative speed and position of the object, while the ECU can compute the absolute speed using Controller Area Network (CAN) data. The radar azimuth angle is 30°, but the field of view of the complete system is nearly the same as the camera.

In both setups, a progressive grayscale camera is mounted inside the cabin close to the rearview mirror. The camera horizontal aperture is about 45°. The image resolution is 640×480,

Manuscript received February 16, 2006; revised July 3, 2006, October 2, 2006, October 9, 2006, and October 23, 2006. This paper was developed in the framework of the Integrated Project PReVENT, subproject APALACI, which is a European research activity funded by the European Commission to contribute to road safety by developing and demonstrating preventive safety applications and technologies. The Associate Editor for this paper was A. Hegyi.

G. Alessandretti is with the Innovative Technologies of Centro Ricerche, Fabbrica Italiana Automobili Torino, 10043 Torino, Italy (e-mail: giancarlo.alessandretti@crf.it).

A. Broggi and P. Cerri are with the VisLab, Dipartimento di Ingegneria dell'Informazione, Università di Parma, 43100 Parma, Italy (e-mail: broggi@ce.unipr.it; cerri@ce.unipr.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2006.888597

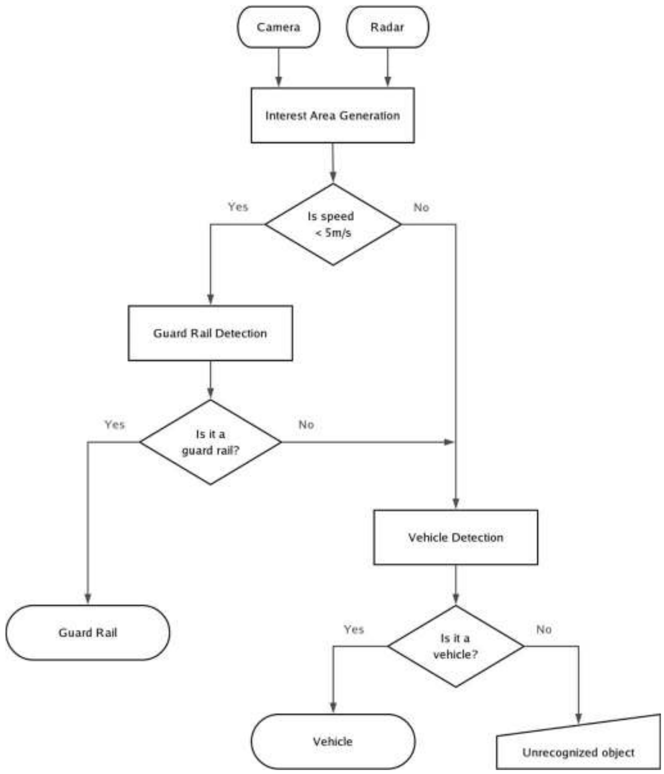
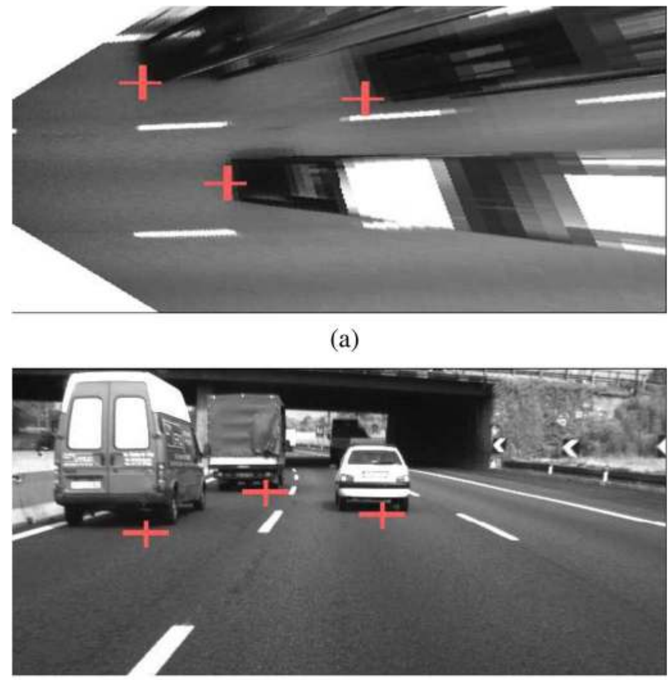


Fig. 1. Algorithm flow chart.



(a)

(b)

Fig. 2. Radar object position (a) in the world reference system and (b) in the image reference system. Note that (a) is rotated 90°.

but only 640 × 300 pixels are used by the vision system, because the remaining rows contain useless information (sky and hood).

Radar and camera calibration are performed separately, and then the two outputs are double-checked together using an obstacle in a known position.

This paper is organized as follows: In Section II, some considerations about radar performance and fusion levels are presented. In Section III, the proposed fusion method is explained. In Section IV, vision processing is presented. Finally, some results and conclusions are presented in Sections V and VI, respectively.

Fig. 1 shows the algorithm flow chart. Interest areas are generated on images using radar data. The possible presence of a guard rail is searched in the interest areas that refer to slow objects. The interest areas that refer to fast objects or that refer to slow objects where no guard rail was found are the candidates for the search of vehicles.

II. Radar Performance and Fusion Levels

Radar reliability and provided features must be taken into account in order to choose the appropriate fusion level.

Different radar performances suggest different fusion levels. High-level fusion needs good radar reliability, while low- or medium-level fusion can be performed even in the case of nonhighly reliable radar data. A fusion system driven by radar, such as the one proposed here, needs a very reliable radar because any radar miss cannot be recovered by vision. Anyway, although the radars used in this paper do not feature a 100% detection rate, they have been demonstrated to be able to detect all the most dangerous obstacles.

Radar can supply different object features. Relative position and speed are the most commonly supplied features, but other ones, such as classification and tracking identification number, can be returned as well. A large number of object features make high- and medium-level fusion easier.

Another aspect that must be taken into account is the sensor's field of view. Sensors with perfectly overlapping fields of view can be managed more easily since all obstacles detected by a sensor are also potentially detectable by the other one. Otherwise, it is required that different approaches be developed to manage both obstacles that lie in areas covered by just one of the two sensors and obstacles that lie in areas covered by both sensors.

III. Fusion

The first step of the algorithm converts radar objects into the image reference system using a perspective mapping transformation that projects the radar points onto the objects base. This transformation is performed using calibration data achieved by fine intrinsic and extrinsic camera parameter measurements as well as radar calibration.

Fig. 2(a) and (b) shows the radar data on an inverse perspective mapping of the image and the radar objects converted into the image reference system, respectively.

Since parameter measurement is performed only once, at system setup, and no stabilization is currently applied, errors may occur when extrinsic parameters change (mainly due to vehicle pitch) due to road roughness or vehicle speed up (see Fig. 3). Moreover, radars may intrinsically provide an incorrect

Fig. 3. Camera miscalibration caused by pitch variation. The radar point is not correctly placed on the vehicle base (i.e., the object base).

Fig. 4. Lateral radar error. The radar data falls outside of the object shape.

lateral position: Points may not be centered onto the obstacle shape or even fall outside it, as shown in Fig. 4.

In the definition of the image area used by vision to validate radar objects, wide margins are used on its left and right sides in order to compensate for possibly inaccurate radar data (lateral offset): Values between 2.5 and 4 m have been tested. The area height is defined to be half of its width, and the area base is positioned below the radar points for 30% of area height, in order that the vehicle should be included even in the case of strong pitch variations. As mentioned in Section VI, an image stabilization system (hardware or software) may reduce the need of working on overdimensioned search areas. Only radar data that refer to points inside the image are considered. Since the chosen radars' horizontal angular field of view are smaller or approximately the same as the camera one, almost all the radar points can be remapped into the image. Nonetheless, a radar point can be remapped to an image point very close to the left or right margin. In such situation, a part of the search area, built as previously explained, may lie outside the image. In order to solve this problem, two different approaches have been tested. The first one is based on moving the search area horizontally until the whole area fits inside the image. This solution may not be very efficient because this new search area may contain a part of the image very distant from the radar point. The use of this area for vehicle search can cause false detections. The second approach is based on cropping the search area. In this case, only the useful part of the image is processed.

In order to simplify and speed up the following steps of the algorithm and to delete details of too close vehicles, all the areas are resampled to a fixed size. To avoid image deformations, all the areas are reduced, preserving their original aspect ratio.

(a) (b)

Fig. 5. Symmetry computation. Arrows indicate where the symmetry value is stored for (a) a single column width and (b) the largest width. Symmetry is computed in the area enclosed in the bright rectangle and using the dark vertical line as an axis.

IV. Interest Area Evaluation

Vehicle detection is the main vision algorithm. It is preceded by a preprocessing step aimed at filtering out radar data given by echoes on the guard rail. The main goal of guard rail detection, which is extremely fast, is the algorithm speed up toward real-time execution. Moreover, the guard rail detection algorithm adds new information about the environment. Guard rail detection is applied first in order to classify radar data that represent a guard rail and remove them from the further analysis of the vehicle detection algorithm. Nonetheless, the vehicle detection algorithm is presented first as it is the main part of this paper.

A. *Symmetry Computation*

Symmetry computation is the basis of the algorithm and the most time consuming part as well. Only binarized edges are used in order to reduce the execution time. Gray-level symmetry is time consuming and does not provide more information than edge symmetry. First of all, the Sobel operator is used to compute edge module and orientation. Then, two images are built: one containing the almost-vertical edges and the other with the almost-horizontal edges. The idea is to detect all vertical and horizontal edges: even the weakest one. Therefore, a very low threshold is used on edge modules; vertical edges are labeled considering their orientation.

Symmetry is computed for every column of the vertical edge image on different-sized bounding boxes whose height matches the image height and with a variable width ranging from one pixel to a predetermined maximum number of pixels. The computed value is saved in a two-dimensional (2-D) data structure (hereinafter referred to as an image) whose coordinates are determined as follows: The column is the same as the symmetry axis, and the row depends on the considered bounding box width (Fig. 5).

Symmetry is computed as

$$simm = \frac{s^2}{n}$$

where s is the number of symmetrical vertical edge points with respect to the considered axis, and n is the number of all vertical edge pixels. This operation is used on the vertical edge image. Two vertical edges are considered symmetrical when featuring an opposite orientation. Since this operation has to be executed

Fig. 6. Interesting columns evaluation. A column with a high value in the symmetry image (white pixels) is considered as the center of a vehicle only if the value of symmetry is low for narrow symmetry areas (top rows) and high for wide ones (bottom rows).

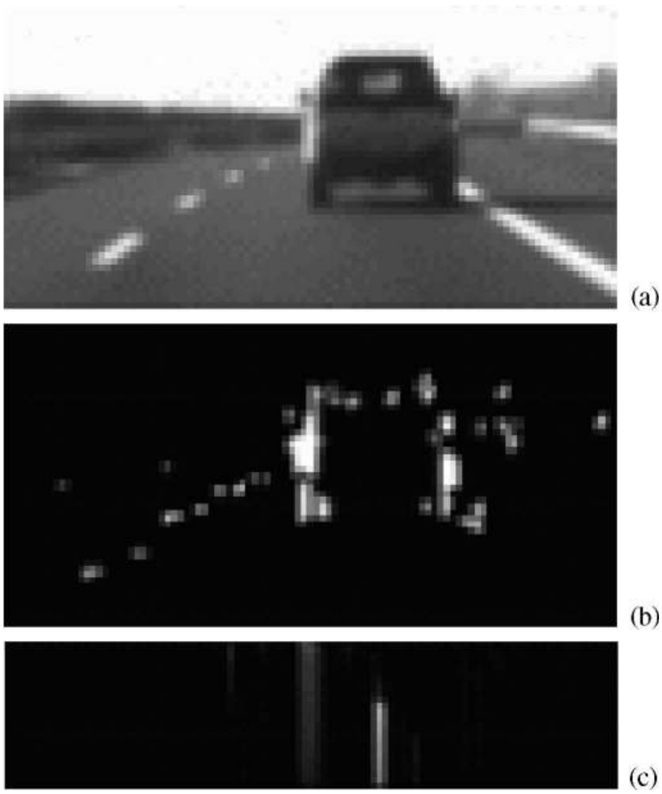


Fig. 7. (a) Original image. (b) Vertical edges. (c) Symmetry image. In the symmetry image, two columns are bright: The left one refers to the vehicle left border, and the right one refers to the vehicle center. The first column is entirely bright because symmetry is due to edges close to each other, namely it could represent a thin object like a pole. The second column is only partially bright. The upper rows are dark because they correspond to narrow symmetry areas, where no symmetrical edges are present. The lower rows that correspond to large symmetry areas are bright. The analysis of the symmetry image can in fact also provide the object size. The vehicle right border does not generate a peak in the symmetry image since its edges are too thin.

for a large number of times, it must be as simple and as quick as possible. This image is then used to search for interesting columns.

B. Columns Evaluation

An interesting column is defined as having a high value in the symmetry image. A columnwise histogram is then used to locate candidate columns. In correspondence to these columns, the vertical edges symmetry is checked to obtain the expected vehicle width. More specifically, if a high value of symmetry is also present for smaller widths, it means that the algorithm has detected a small object; in this case, the column is discarded.

Fig. 6 shows an example. The leftmost vertical peak is discarded because it presents a high symmetry value (bright pixel) for both large (bottom rows) and small widths (top rows).

Fig. 8. Box width computed from peak height.

On the other hand, the rightmost peak presents an appreciable symmetry value only for widths above a certain size (the same behavior is also shown in Fig. 7).

C. Bounding Boxes Generation

Up to now, the algorithm provides information about the vehicle's center position, but since vehicles need to be detected with high accuracy, a precise bounding box detection is mandatory. Each peak in the vertical edges symmetry image that survived the previous filterings is used. The width of the symmetry box is given by the distance between the peak itself and the top of the symmetry image. The box is then centered within the column (Fig. 8).

The shadow under the car is a strong invariant that is always present even in dark days. The algorithm looks for the vehicle shadow in order to find its base. Since other shadows are present on the road as well (e.g., bridges), the algorithm looks for a high concentration of edges above the horizontal edge. If no base can be detected in correspondence to the peak, the column is discarded.

During sunshine or sunset, shadows are elongated, and the vehicle shadow may fall distant from the vehicle. This might be a cause for errors. In order to solve this problem, the vehicle base is searched for in all the lower half of the interesting area. In such a way, the bumper edge can be detected as well. Although the misrecognition of a bumper as a vehicle base can cause errors in distance estimation, these errors are solved by using the distance measurement provided by the radar, which is highly reliable.

The search for the vehicle top is performed as well, but the box is validated even if the top is not found because sometimes, it can be very hard, or actually impossible, to spot it. Truck tops often fall outside the search area or even outside the image. If the top cannot be found, a box with a fixed ratio between width and height is used [see Fig. 18(b)].

D. Filters

The algorithm is designed to return only one vehicle for each radar point, but more than one possible vehicle can be detected

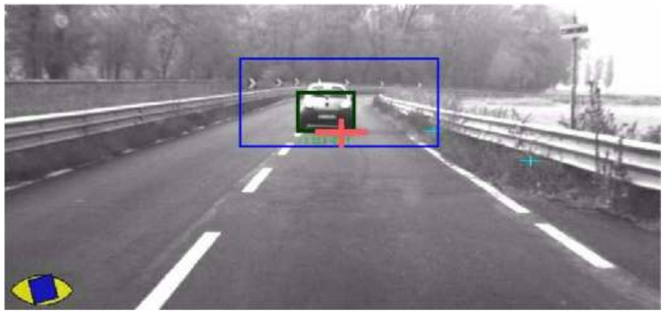


Fig. 9. Intermediate results. Vision is able to refine the lateral position of the object detected by the radar.

in a single area. A filter that determines which vehicle has to be validated is mandatory. This filter simply chooses the most central box. This filter, anyway, can be critical in heavy traffic situations, where many vehicles may be running very close to each other.

E. Results Merging

When all radar data have been examined, all the boxes framing the detected vehicles are resampled to their original size and mixed together. Overlapping vehicles are also managed.

Using an inverse perspective mapping transformation, the real width and position of vehicles can be computed. In the computation of these values, the radar provides distance while vision provides position and width so that the radar precision on distance measurement and the vision refinement ability are capitalized together.

Unfortunately, not all detected boxes are correct. Some false positives, caused by road signs or other objects in the scene, can be present as well. A filter is used to discard some false positives: it removes too large or too small boxes that are unlikely to represent a vehicle.

It is also possible that a vehicle is detected in more than one search area, so overlapping results may be present. Only one box per vehicle is expected as a final result, so a further step is required to merge similar boxes and eliminate redundant ones. When two boxes have different sizes, the largest box is preferred since the smallest one is often generated by vehicle parts. Furthermore, when two rectangles with similar size have their base approximately at the same height in the image, an average is computed, and this new box is considered.

Fig. 9 shows an intermediate result. The cross represents the radar point, the large rectangle represents the interest area, and the small rectangle represents the detected vehicle.

F. Guard Rail Detection

The guard rail detection method is divided into two parts. The first one deals with the preprocessing of the input image. The second one, i.e., the main section, is based on line searching. First of all, an image in which only the edges of potential guard rails are present and well connected is built. Then, the algorithm analyzes the edges and, according to their length, determines whether the object is a guard rail. The algorithm is applied only to objects with a real speed (measured by the radar) lower than 5 m/s.

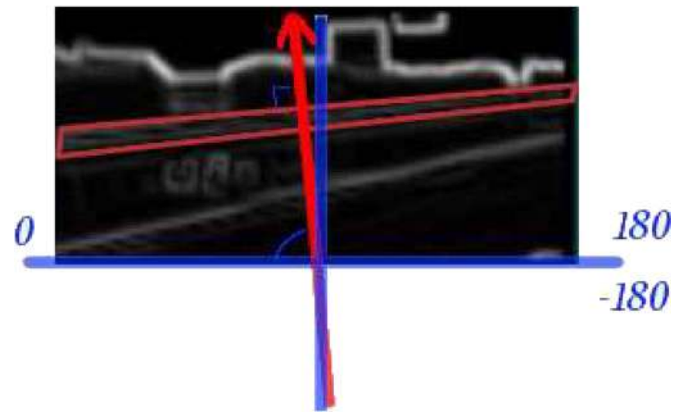


Fig. 10. Guard rail edge orientation in acquired images.

(a)

0	0	0
1	1	1
0	0	0

(b)

0	0	1	0	0
1	1	1	1	1
0	0	1	0	0

Fig. 11. Structuring elements for (a) erosion and (b) dilation.

1	3	6	7	8
.	4	9	10	11
2	5	12	13	14

Fig. 12. Priority of the first white pixel search. 1 means the highest priority and 14 the lowest. This priority list was defined experimentally in order to follow the line nearest to the border of the interest area and nearest to the middle of the interest area.

1) Preprocessing: A Sobel filter is applied to the input image. In order to reduce the number of edges where the search must be performed, only the ones with an orientation approximately matching a guard rail are considered (Fig. 10).

The Sobel image is binarized with a very low threshold on module, and then morphological operations are applied.

Guard rail edges are generally almost horizontal; therefore, a morphological erosion is applied using the structuring element shown in Fig. 11(a), which maintains horizontal components and erases isolated points and vertical segments. On the other hand, a morphological dilation reinforces both vertical and horizontal components, with a higher intensity for the last ones, in order to preserve the horizontal shape of guard rails [Fig. 11(b)].

This sequence of operations allows to achieve robust and continuous edges and to reduce noise.

2) Line Searching: The algorithm is designed to have a different behavior according to the side of the image where the object is detected (left or right). In this section, some examples of left guard rail detection are presented.

The algorithm searches for the leftmost pixel of a possible guard rail that should be found in the first few columns of the image. The search is performed starting from the middle-height pixel to the bottom of the image. A white pixel is searched for in a rectangular neighborhood area according to the priority shown in Fig. 12.

Fig. 13. Area A: first white pixel search area.

4	2
.	1
5	3

Fig. 14. Pixel priority for contiguous edge search. 1 means the highest priority and 5 the lowest. Since guard rails are represented by a slightly oblique line, the priority list is made in a way that the search continues almost horizontally.

The search continues by moving the neighborhood structure by three pixels down at a time until all the area A, shown in Fig. 13, has been covered or a white pixel is found.

Then, when a possible guard rail is detected, the main search step starts. If area A includes an obstacle (a car, a road signal, ...), which occludes the visibility of the guard rail, then the guard rail may not be detected. Anyway, guard rails usually give a large number of radar points that are close to each other and in sequence, thus creating a set of overlapping boxes (see Fig. 17). Each area of interest is then used to search for the presence of the guard rail, and although it could be occluded in the closest area, it may get detected in other farther areas of interest. A future backtracking step might allow to help the detection in the closest areas.

The search for contiguous edge pixels gives priority to the pixels that most probably belong to a line with an orientation matching the model of a guard rail (according to the priority shown in Fig. 14).

Some guard rail edges may be noncontiguous due to either noise or small obstructions. In order to avoid to miss the line, the algorithm needs to compensate for short interruptions of the oblique line; therefore, the algorithm also continues horizontally (as shown in Fig. 15) if there are no edge pixels in the neighborhood of the last edge pixel. A maximum of five holes is allowed in this implementation.

The algorithm ends if more than five holes are found or if the right border of the search area is reached. If the detected line is longer than 50% of the search area, the object will be labeled as a guard rail. As experimentally demonstrated and shown in Fig. 16, a threshold of 50% seems to be a good tradeoff. Radar points that match guard rails quite always reach over 50% of the image width, even if it contained a part of a vehicle moving on the road.

An interrupted guard rail could not be identified, even if it is detected by the radar, because it could be shorter than the fixed threshold. Anyway, this is not a great problem because guard rail detection is developed in order to speed up the algorithm and not as a standalone guard rail detector.

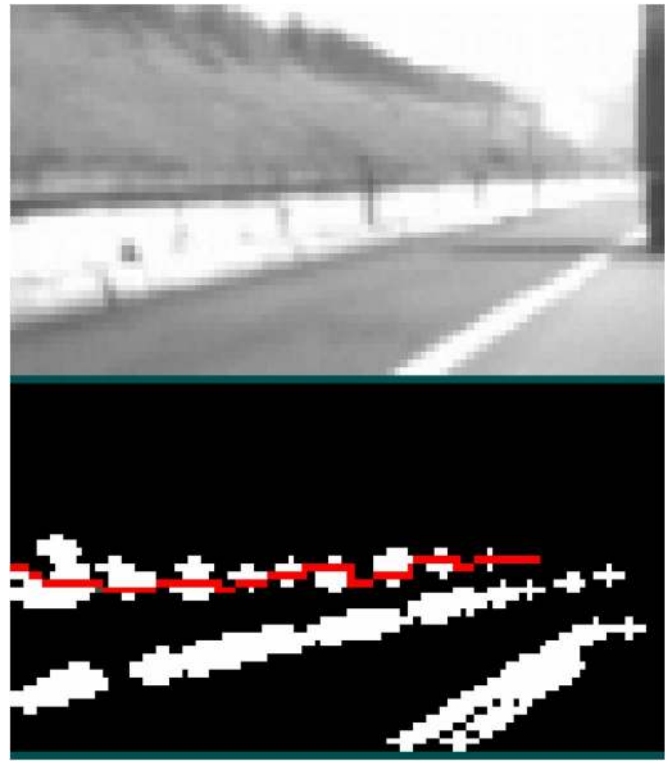


Fig. 15. Guard rail with noncontinuous edges is correctly detected. The line represents the identified guard rail.

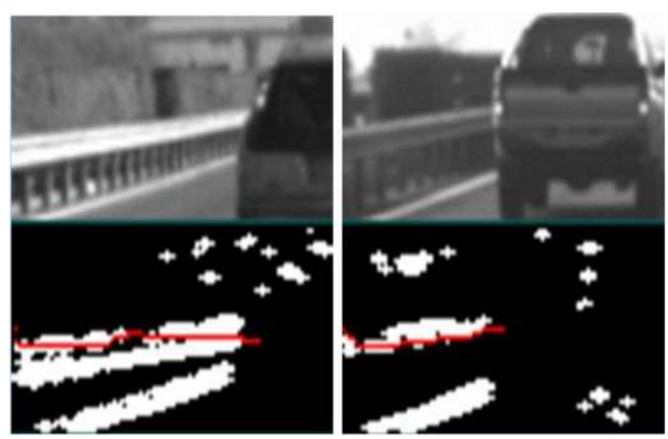


Fig. 16. Two correctly detected guard rails (lines).

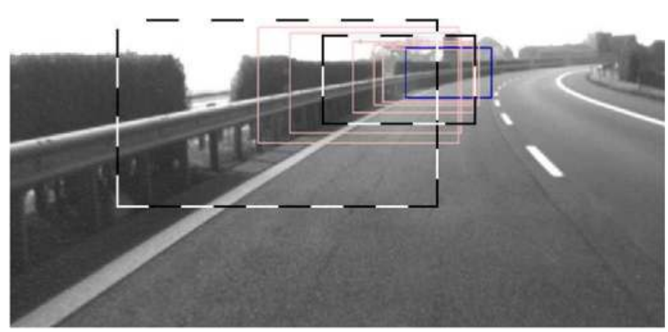


Fig. 17. Overlapping interest area management. If a guard rail is detected in an area (the white–black dashed boxes), all the areas overlapping with it are labeled as a guard rail (bright boxes).

Fig. 18. Examples of correct results. (a) The algorithm works reliably in simple cases. (b) It detects both vehicles moving away and approaching. It works even in hard cases, such as (c) rainy and (d) noisy scenarios (note the double radar detection). It can detect (e) multiple cars and (f) trucks.

G. Overlapping Box Management

A single guard rail may generate more than one radar point, so many overlapping areas of interest may be present. Searching for guard rails in all the areas is time consuming, so overlapping areas are discarded without applying the complete algorithm. In Fig. 17, the white–black dashed boxes represent objects identified as guard rail, and the bright ones represent boxes overlapped with a box already labeled as containing a guard rail.

In order to discard boxes overlapping with a box identified as a guard rail, the areas referring to slow objects are sorted according to their distance from the vehicle, from the nearest to the furthest, and then, the guard rail algorithm is applied to the closest. If a guard rail is detected in the closest area, and if the following area is overlapped with the first one more than a threshold, the object appearing in the second area is then labeled as guard rail as well without any further processing. This process is iterated until no other overlapping area is detected. Guard rail detection is then applied to the first nonoverlapping area, and this process is iterated again.

V. Results

Performance assessment heavily depends on many aspects, such as the hardware system and setup, the scenarios examined, and the final goal of the application. In the literature, many works dealing with this topic can be found, e.g., [2]–[4] provide some results too, but a quantitative comparison between results obtained applying different approaches is very difficult to perform: It will be an important topic for future research.

In this paper, we made every effort to identify critical situations and assess the performance with the final goal of improving the accuracy of the lateral position of detected vehicles.

This vehicle detection system has been tested in rural and highway environments with good results. To evaluate system performance, ground truth was manually collected by annotating the presence of each vehicle in more than 12 000 images. For each vehicle, a human operator annotated its position, size, and shape. The annotated sequences represent a mix of all the possible scenarios: high and low traffic, rural and highway, fast, slow, and stopped vehicles, sunshine, cloudy, and rainy. It is

Fig. 19. Examples of errors. (a) and (b) The most frequent errors are due to traffic, but some errors may also happen in simple cases. (c) A vision false negative due to low contrast on the left border. (d) A radar false negative (the oncoming vehicle). (e) An inaccurate detection. (f) A false positive due both to vision and the radar.

important to remember that no tracking is used at the present moment as it will be introduced at a later stage of the project.

The percentage of vehicles whose position and size are detected with an extremely high precision (with an error lower than 40 cm) independently in every frame is about 50%. This number also includes sensor-related issues, such as radar misses and vision misses due to bad environmental conditions (such as rain or darkness). Five different performance indexes were defined for these statistics, namely 1) refined detections (RD), 2) false negatives (FN), 3) radar false negatives (RFN), 4) false positives (FP), and 5) nonvehicle obstacle (NVO).

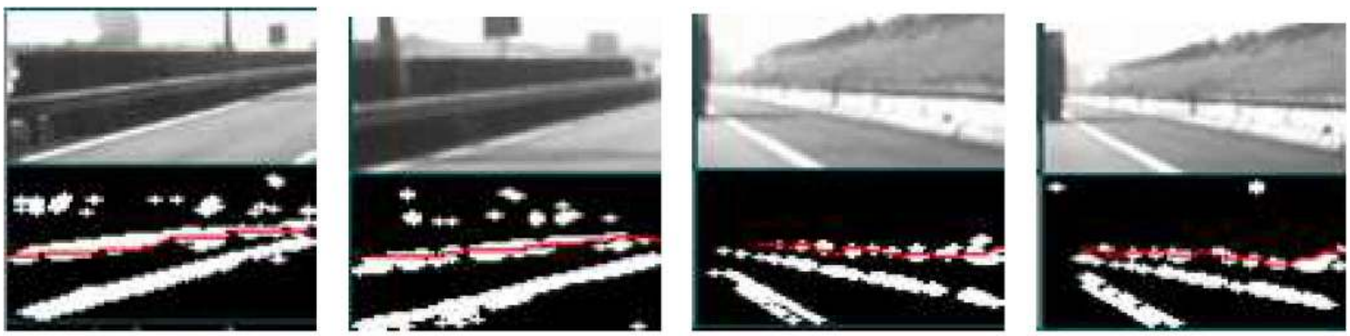
- RD vehicles detected by radar and validated and refined by vision;
- FN vehicles detected by radar but not validated or not correctly refined by vision;
- RFN vehicles not detected by radar or detected with a too low precision;
- FP nonvehicle obstacles detected by radar and validated by vision;
- NVO nonvehicle obstacles detected by radar and not validated by vision.

Although the definition of *refined detections* is straightforward, the other indexes need an explanation. Radar false negatives are defined as vehicles not contained, or not entirely contained, in any search area. This value obviously depends on the search area size. The chosen interest area width is 2.5 m. The RFN is 39% of framed vehicles. This value can be decreased by raising the area width, but this change will likely increase false positives as well. More than half of the radar false positives refer to vehicles partially contained in a search area. The same consideration can be made for false negatives. The false negative density is about 13%, but only 5% is due to actually missed vehicles. The remaining 8% is due to vehicles detected with a low precision. The number of false positives is low. In all the test sequences, only one persistent false positive is present.

Event-based statistics were computed as well, considering an event as a vehicle present in more than ten frames. Radar

TABLE I
RESULTS ON DIFFERENT SEQUENCES

SCENARIO	EVENTS	RADAR MISS	NOT REFINED	REFINED BY VISION
highway	5	0	1	4
highway with traffic	9	3	1	5
highway with shadow problems	10	0	3	7
trucks on highway	2	0	0	2
highway junction	14	5	4	5
rural with curves	1	0	0	1
rural with approaching vehicles	8	1	2	5
rural with strong shadows	10	0	2	8
complex environment	10	0	3	7
total	**69**	**9 (13%)**	**16 (23%)**	**44 (64%)**



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

Fig. 20. Examples of correctly detected guard rails. From (a) easy edges to (d) noncontiguous dashes.

completely misses 13% of the events (mainly due to traffic), while vision is not able to refine 23% of the events. According to these data, 64% of the events are correctly detected and refined, and 73% of the object supplied to vision are then correctly refined.

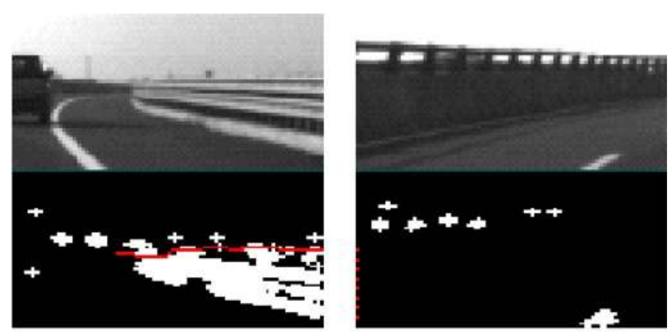
Figs. 18 and 19 show, respectively, good results and errors obtained in different scenarios. Fig. 19(a) and (b) shows traffic cases. In the first image, a single radar point is generated by the vehicles close to each other. Its position is not suitable to detect any vehicle. The second image shows a delivery van individuated by two radar points and some other nondetected vehicles.

Event-based statistics obtained on different sequences are proposed in Table I. Radar misses are present only in heavy traffic or complex scenarios. The main issues are traffic and environment complexity together with shadow or general illumination problems.

As already mentioned, nonprecise detections, due to traffic or low visibility conditions, may happen. In our performance analysis, these cases are classified as false negatives since the final goal of this system is to estimate the vehicle shape with very high accuracy.

Heavy traffic is a hard scenario for this method because the detection of occluded vehicles can be difficult (or even impossible) using symmetry. Anyway, not occluded vehicles can be detected even in such a situation.

The guard rail detection system was tested as well. Urban roads were not considered since no guard rail is present. Interesting results in critical conditions of high traffic and irregular or fragmented guard rails are reached in exurban roads (see Fig. 20).



(a) (b)

Fig. 21. Guard rails starting (a) in the lower half of the image (detected) and (b) in the upper half (not detected).

The method was tested on ten image sequences (totally about 25 000 images) and proved to reach satisfactory results.

The number of false negatives (guard rail not detected) is very low. It is not possible to give their exact number because in some circumstances it is not possible to clearly understand whether the obstacle returned by the radar refers to a guard rail or to another object in the box (cars, road signals, vegetation, . . .). The false positives (something not a guard rail erroneously labeled as a guard rail) are even lower and are limited to objects close to road edges and are very similar to guard rails. Only once, during the test, was an object in the background labeled as a guard rail.

Fig. 22. Beltway. Guard rail is almost horizontal and is not identified.

TABLE II
EXECUTION TIME ON HIGHWAYS. VD STANDS FOR VEHICLE DETECTION ALGORITHM ONLY, GRD STANDS FOR GUARD RAIL AND VEHICLE DETECTION ALGORITHMS, AND OBM STANDS FOR OVERLAPPING BOXES MANAGEMENT, GUARD RAIL, AND VEHICLE DETECTION. THE DECREMENT OF OBM WITH RESPECT TO VD IS SHOWN IN THE REDUCTION COLUMN

sequence	VD [ms]	GRD [ms]	OBM [ms]	reduction
11	22	19	17	22%
12	25	18	16	36%
13	30	28	26	13%
14	27	24	22	18%
15	29	27	26	10%
average	26,6	23,2	21,4	20%

TABLE III
EXECUTION TIME ON EXURBAN ROADS. LOW TIME REDUCTION PERCENTAGE IS DUE TO POOR PRESENCE OF GUARD RAILS

sequence	VD [ms]	GRD [ms]	OBM [ms]	reduction
8	25	25	25	0%
16	25	25	25	0%
17	26	26	25	4%
18	26	25	25	4%
average	25,5	25,3	25	2%

Some false negatives occurred in correspondence to very tall guard rails (see Fig. 21). In these cases, the guard rail edges lie in the upper half of the image and do not enter in area A, as shown in Fig. 13; therefore, the initial searching step fails.

Some critical cases were found in beltways or in narrow curves where false negatives may occur, when the guard rail appearance in the image is quasi-horizontal (Fig. 22).

The average algorithm execution times for different image sequences are reported in Tables II and III.

The more guard rails are detected, the more time is saved. In sequence 12, many guard rails are present and detected, and the time saved is about 36%. The average time saving is about 20% for highways environment, where a larger number of guard rails is present, and may drop to as low as about 2% or even 0% in exurban roads. No wasting-time cases are reported due to the very high speed of the algorithm. The overlapping management system contributes to these results for about 30%. A reduction of 4–5 ms is definitely significant in real-time applications, especially when other algorithms may run on the same processing engine.

False positives produced by the vehicle detection algorithm alone are reduced by the introduction of guard rail detection.

Fig. 23. Correct results reached in a traffic scene.

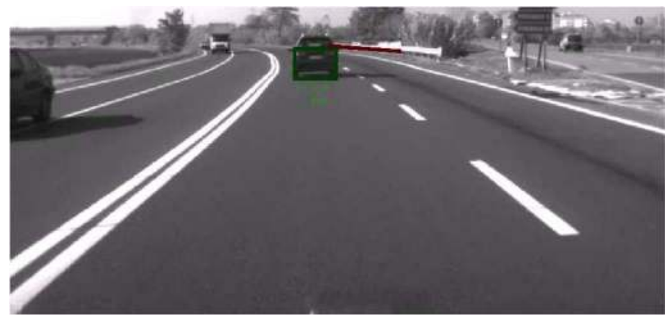


Fig. 24. Coexistence of the guard rail detection and vehicle detection algorithms. The vehicle is anyway detected also when the guard rail detection is running.

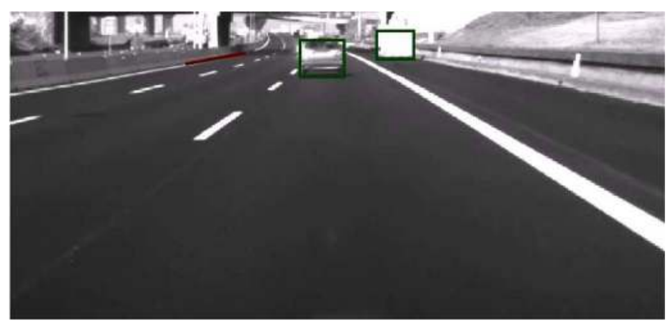


Fig. 25. Correctly detected stopped vehicle. No guard rails are detected.

For example, in a beltway, where the guard rail is generally in front of the camera, the vehicle detection algorithm may return a high number of false positives. The proposed method reduces them of about 25%. The average false-positive reduction can be rated at about 10%.

No cases of vehicles labeled as guard rail occurred. All boxes related to moving vehicles are not considered by the guard rail detection algorithm, but, as shown in Figs. 23 and 24, guard rails between two vehicles can also be correctly detected.

Moreover, vehicles parked on the edge of the road are not considered by the guard rail algorithm so they can be correctly detected as vehicles. The best results are reached when side vehicles are quite close and frontal (see Fig. 25).

VI. CONCLUSION

In this paper, a method to fuse radar data and vision is described. The method was tested with two different radar sensors.

This method gets good results in both rural and highway environments. Even if not all vehicles are detected in all images, the system is promising for safety applications because all the closest and most dangerous vehicles are correctly and precisely detected.

The strength of the proposed approach is that it is composed of simple operations only. The computational complexity is extremely limited, and its current implementation can reach video rate performance (about 40 Hz).

A hardware or software image stabilization might provide a more precise perspective mapping transformation. The radar points might be remapped to the correct image row, thus making detection easier and reducing the height of the search area.

A tracking step might be very helpful to increase the robustness of the system and the detection persistence and will be introduced in the future.

The use of other methods to generate areas of interest and the search for more than one vehicle in an interest area may solve some radar sensor problems, such as its inability to detect all vehicles and to distinguish vehicles close to each other.

The guard rail detection method gets good results both in time savings and in false-positive reduction. The Hough transformation can be used to detect guard rail as well, but while Hough is time consuming, the proposed method is faster and easier.

Guard rail detection could be improved using tracking as well. Moreover, if vehicle yaw rate and speed are known, it would be possible to detect whether the vehicle is driving along a curve, and the algorithm thresholds could then be modified in order to also detect almost horizontal guard rails.

REFERENCES

[1] J. Laneurit, C. Blanc, R. Chapuis, and L. Trassoudaine, "Multisensorial data fusion for global vehicle and obstacles absolute positioning," in *Proc. IEEE Intell. Vehicles Symp.*, Columbus, OH, Jun. 2003, pp. 138–143.

[2] B. Steux, C. Laurgeau, L. Salesse, and D. Wautier, "Fade: A vehicle detection and tracking system featuring monocular color vision and radar data fusion," in *Proc. IEEE Intell. Vehicles Symp.*, Versailles, France, Jun. 2002, pp. 632–639.

[3] U. Hofmann, A. Rieder, and E. D. Dickmanns, "Radar and vision data fusion for hybrid adaptive cruise control on highways," in *Proc. Int. Conf. Comput. Vis. Syst.*, Vancouver, BC, Canada, 2001, pp. 125–138.

[4] A. Sole, O. Mano, G. Stain, H. Kumon, Y. Tamatsu, and A. Shashua, "Solid or not solid: Vision for radar target validation," in *Proc. IEEE Intell. Vehicles Symp.*, Parma, Italy, Jun. 2004, pp. 819–824.

[5] S. Denasi and G. Quaglia, "Obstacle detection using a deformable model of vehicles," in *Proc. IEEE Intell. Vehicles Symp.*, Tokyo, Japan, May 2001, pp. 145–150.

[6] K. Fleischer, H. H. Nagel, and T. M. Rath, "3D-model-based-vision for inner city driving scenes," in *Proc. IEEE Intell. Vehicles Symp.*, Paris, France, Jun. 2002, pp. 477–482.

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

[8] C. Hoffman, T. Dang, and C. Stiller, "Vehicle detection fusing 2D visual features," in *Proc. IEEE Intell. Vehicles Symp.*, Parma, Italy, Jun. 2004, pp. 280–285.

[9] A. Broggi, P. Cerri, and P. C. Antonello, "Multi-resolution vehicle detection using artificial vision," in *Proc. IEEE Intell. Vehicles Symp.*, Parma, Italy, Jun. 2004, pp. 310–314.

Giancarlo Alessandretti received the degree in physics from Scuola Normale of Pisa, Pisa, Italy, in 1974.

After a period with Pisa University, in 1977, he joined the Innovative Technologies of Centro Ricerche, Fabbrica Italiana Automobili Torino, Torino, Italy, where he was involved in applied industrial research, with a focus on automotive technologies. From 1977 to 1988, he was a Researcher and head of unit in applied optics. From 1989 to 1994, he was a Project Coordinator in the fields of optics, ergonomy, and human–machine interface. He was then responsible for new emerging activities on microsystems and smart materials. Since 2000, he has been coordinating research activities regarding integrated safety on vehicles. He has coordinated several European projects and is now in the Core Group of the Prevent Integrated Project. He is author of more than 40 papers and conference publications. His research interests include preventive safety, obstacle detection techniques, and driver–vehicle interaction.

Dr. Alessandretti is a member of professional groups like the EUCAR Association of Car Manufacturers and platforms for the coordination and promotion of European research, like e-Safety and ERTRAC.

Alberto Broggi (SM'89–S'93–A'96–SM'06) received the Dr.Eng. degree in electronic engineering and the Ph.D. degree in information technology from the Università di Parma, Parma, Italy, in 1990 and 1994, respectively.

He was an Associate Researcher with the Dipartimento di Ingegneria dell'Informazione from 1994 to 1998, an Associate Professor of artificial intelligence with the Dipartimento di Informatica e Sistemistica from 1998 to 2001, and is currently a Full Professor of computer science with the Università di Parma. He has authored more than 150 refereed publications in international journals, book chapters, and conference proceedings, and delivered invited talks at many international conferences. His research interests include real-time computer vision approaches for the navigation of unmanned vehicles and the development of low-cost computer systems to be used in autonomous agents.

Dr. Broggi is the Editor-in-Chief of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and a member of the IEEE ITS Council Executive Committee. He has been the Editor on the subject of ITS for the *IEEE Intelligent Systems Magazine* since 1999.

Pietro Cerri (S'05–M'07) received the degree from Università di Pavia, Pavia, Italy, in 2003. He is currently working toward the Ph.D. degree in information technology at the Dipartimento di Ingegneria dell'Informazione, Università di Parma, Parma, Italy.

In 2003, he received a research grant from the Associazione Tecnica dell'Automobile, Italy. He is currently with the VisLab, Dipartimento di Ingegneria dell'Informazione, Università di Parma. His research is mainly focused on computer vision and sensor fusion approaches for the development of advanced driver-assistance systems.