

Guard rail detection using radar and vision data fusion for vehicle detection algorithm improvement and speed-up.

Alberto Broggi, Pietro Cerri, Fabio Oleari and Marco Paterlini
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Parma
43100 Parma, Italy
{broggi, cerri, oleari, paterli}@ce.unipr.it

Abstract— This paper describes a method for detecting guard rails fusing radar and vision data in order to improve and speed-up vehicle detection algorithms. The method is based on the search for uninterrupted oblique lines that cross an interest area. The interest area is dynamically indicated by a radar sensor. A method to manage overlapping areas is also described. The method's efficiency, both in terms of time saving and correct detection rate, is numerically shown.

I. INTRODUCTION

Vehicle detection algorithms represent a basic functionality of Intelligent Vehicles and are useful in many new automotive applications designed to improve driving safety and comfort such as automatic cruise control and vehicle following. The main goal of this work is the improvement of the overall algorithm and its real-time execution. Moreover the guard rail detection algorithm adds new informations about the environment.

The system proposed in this paper is based on the fusion of data coming from radar sensors and cameras. The advantages and the problems of fusing radar and camera data for vehicle detection are well known [4], methods differ mainly for the fusion level: low level fusion [5] and high level fusion [3] have both proved to reach good results. This work is developed using high level fusion and focuses on slow objects to validate radar targets, as Sole shows too [2].

The vehicle detection algorithm used in this work is based on symmetry [1] and uses radar data in order to localize areas of interest. The data fusion operates at high level: the vision system is used to validate radar data and to make them more accurate. Every point identified as an obstacle by the radar is remapped into the image reference system using a perspective mapping transformation: an area is built around image coordinates representing the object. The size of the area has width and height proportional to object distance. In these areas the vision-based vehicle detection algorithm is used in order to evaluate vehicles' size and position with a good precision. Unfortunately a lot of data provided by the radar refer to something that is not a vehicle; searching for vehicles in these areas causes system slowdown and may cause false positives. In order to solve this problem a robust and fast solution is mandatory: discarding areas only according to object speed and position is not robust enough and may exclude very slow vehicles near the road border. The

algorithm described in this paper is applied only to objects with a very low real speed (measured by radar), lower than 5 m/s. If a radar data is labeled as a guard-rail no other elaboration is made in order to save time, otherwise the area is searched for the presence of a possible vehicle. The vehicle detection algorithm is explained in [1]; this paper is focused on guard-rail detection.

In the next sections it is shown that the proposed method is robust and fast: it fails very rarely and provides a satisfactory speed-up; to ensure a higher speed-up an overlapping box method has been added as well (section 3). Finally some results and conclusions are exposed in section 4.

II. GUARD-RAIL DETECTION

The guard-rail detection method is divided into two parts: the first regards preprocessing of the input image; the second, i.e. the main section, is based on line searching. First of all an image in which only the edges of probable guard-rails are present and well connected is built; then the algorithm analyzes the edges and, according to their length, figures out if the object is a guard-rail. The algorithm is applied only to objects with a very low real speed (measured by radar), lower than 5 m/s. The algorithm is different according to the side of the image where the object has been detected (left or right).

A. Preprocessing

First of all 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 corresponding approximatively to a guard rail are considered (Fig. 1).

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

Guard-rail edges are generally quasi-horizontal, therefore the erosion is applied using the structuring element in figure 2a that maintains horizontal components and erases isolated points and vertical segments. On the other hand the expansion reinforces both vertical and horizontal components, with a higher intensity for the last ones, in order to preserve the horizontal shape of guard-rails (2b).

This sequence of operations allows to achieve robust and continuous edges and to reduce noise. The noise due to points close to guard rail remained after Sobel's filter application, is

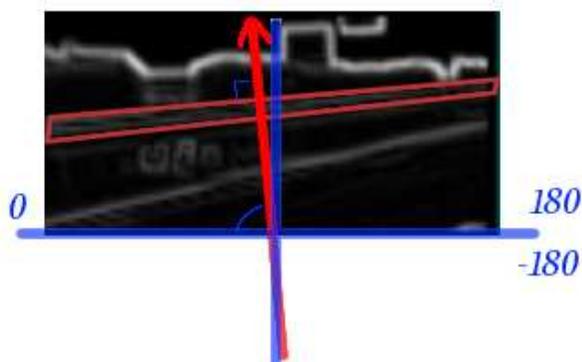


Fig. 1. Edges orientation.

0	0	0	0	0	255	0	0
255	255	255	255	255	255	255	255
0	0	0	0	0	255	0	0

(a) (b)

Fig. 2. Structuring elements for (a) erosion and (b) expansion.

processed by expansion shown in figure 2b reinforcing guard rail edge. This doesn't produce false positives because the noise itself doesn't produce a robust and continuous edge. Points far away, not erased by erosion in figure 2a, don't lead up to problems because they are not considered by line searching method, explained in the next paragraph.

B. Line Searching

As mentioned, the method needs to work in two cases: for right and left guard rails, sharing a symmetrical algorithm; in this section some examples of left guard rail detection are presented.

First of all the algorithm searches for the left-most 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 till the end of the image. A white pixel is searched in a rectangular neighborhood area according to the priority shown in 3.

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

Fig. 3. Priority of the first white pixel search: 1 means the highest priority, 14 the lowest one.

The search continues moving the neighborhood structure by three pixels down at a time until all the area A shown in figure 4 has been covered, or a white pixel is found.

Then, if a possible guard-rail is detected, the main research step starts. If the area A is occluded by an obstacle (car, road signal, ...) guard rail is not detected. Guard rails usually give a lot of radar points that are close and in sequence, creating a set of overlapping boxes (8). The guard rail can be recognized in one of the next boxes. A future back-tracking algorithm will be able to label also the first box as guard rail. Anyway this case is very rare.

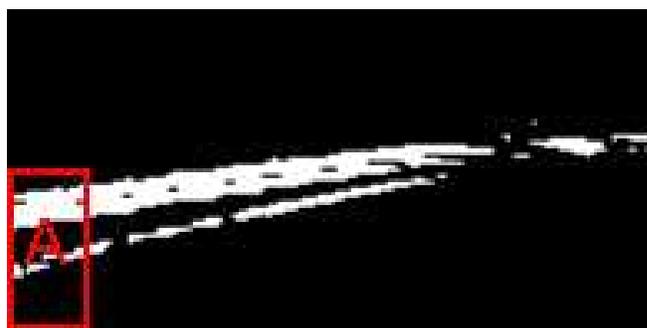


Fig. 4. First white pixel search area.

The search for contiguous edge pixels gives priority to pixels that most probably belong to a line with an orientation matching the model of a guard rail (according to the priority shown in figure 5).

4	2
.	1
5	3

Fig. 5. Pixel priority for contiguous edge search: 1 means the highest priority, 5 the lowest one.

Some guard rail edges may be non-contiguous due either to noise or to small obstructions. In order to avoid to miss the line, the algorithm needs to compensate for short interruptions in the oblique line; therefore the algorithm continues horizontally (as shown in figure 6) also if there are no edge pixels in the neighborhood of the last edge pixel. A maximum of 5 holes is allowed.

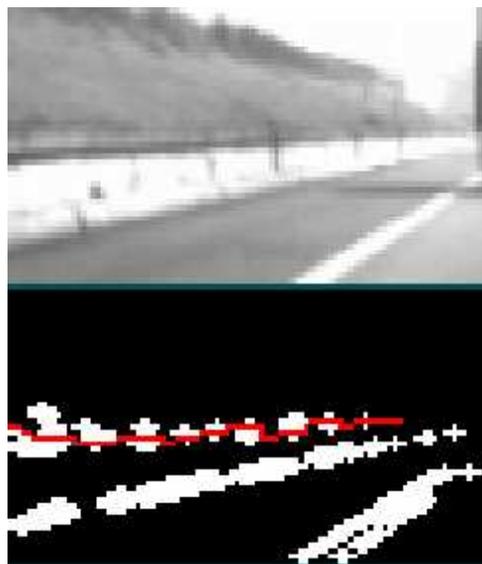


Fig. 6. Scratched guard-rail correctly detected skipping holes: red line represents found guard-rail.

The algorithm ends if more than 5 holes are found or if the right border of the searching area is reached: if the detected line is longer than 50% of the searching area, the object

will be labeled as a guard rail. As shown in figure 7 the threshold of 50% seems to be good: radar points that match guard rails quite always reach over 50% of image width, even if it contained a part of a vehicle moving on the road.

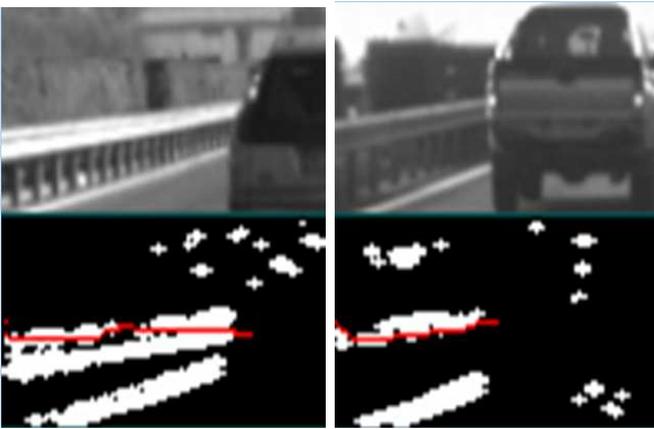


Fig. 7. In red two correctly detected guard rails.

III. OVERLAPPING BOXES MANAGEMENT

A single guard rail may generate more than one radar point, so many overlapping areas of interest may be present: searching for guard rail in all the areas is time wasting. The idea is to discard overlapping areas without applying the complete algorithm; in figure 8 the light-blue boxes represent objects identified as guard rail and the pink ones represent boxes overlapped by a box already labeled as containing a guard rail.

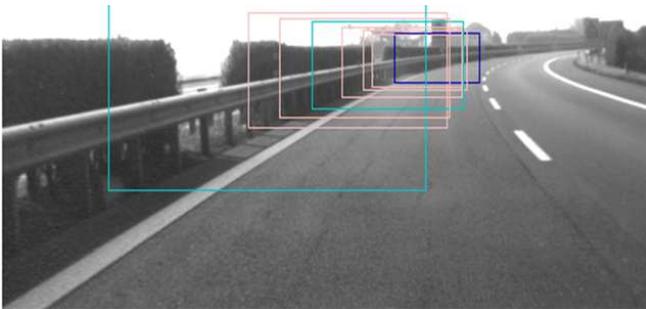


Fig. 8. Overlapping interest areas.

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 nearest to the furthest; then the guard rail algorithm is applied to the first one. 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 will be labeled as guard rail as well. This process is iterated until no other overlapping areas are detected. The guard rail detection is then re-applied to the first area that is not labeled as a guard rail, and iterated again.

IV. RESULTS AND CONCLUSION

The tests have been carried out on a vehicle equipped with radar and camera running on exurban Italian roads and highways. Urban roads have been missed out due to their guard rails lack and because they aren't in the case study of vehicle detection algorithm. Interesting results in critical conditions of high traffic and irregular, fragmented guard rail are reached in exurban roads. The processing system is based on a linux machine. Images acquired are 640x480 pixel 8 bit greyscale and are sampled at 10 fps.

The method was tested on 10 image sequences, totally about 25000 images and proved to reach satisfactory results.

The number of false negatives (guard-rail not detected) is very low: it's not possible to give their exact number because in some circumstances it's not possible to objectively understand if the obstacle returned by radar refers to a guard rail rather than another object in the box (cars, road signals, vegetables, ...). The false positives (something not a guard-rail labeled as a guard-rail) is even lower and by the way limited to objects close to road edge and similar to guard rails. Only one time, during test, an object in background has been labeled as a guard rail.

Some false negatives occur with very tall guard rails (see figure 10); in these cases the guard rail edges stay in the upper half of the image and do not enter in the A area shown in figure 4, therefore the start searching step fails.

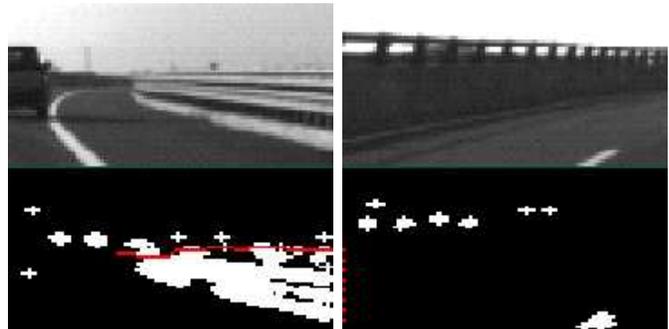


Fig. 10. Guard rails starting in the lower half of image (detected) (a) and in the upper half (not detected) (b)

Even in a beltway or in a narrow curve false negatives may occur because the guard rail appearance in the image may be even quasi-horizontal (figure 11).

In tables I and II the average algorithm execution times for different image sequences are reported.

The more guard rails are detected, the more time is saved; in sequence 12 a lot of guard rails are present and the time saved is about 36%. The average time saving is about 20% for highways environment, where a bigger number of guard rail is present, and about 2% for exurban roads; no wasting-time cases are seen. The overlapping management system contributes for about 30% to these results. A reduction of 4 or 5 ms is significant in real time elaboration, especially when other algorithms run on the same computer.

False positives produced by the original vehicle detection algorithm are reduced by the introduction of the proposed

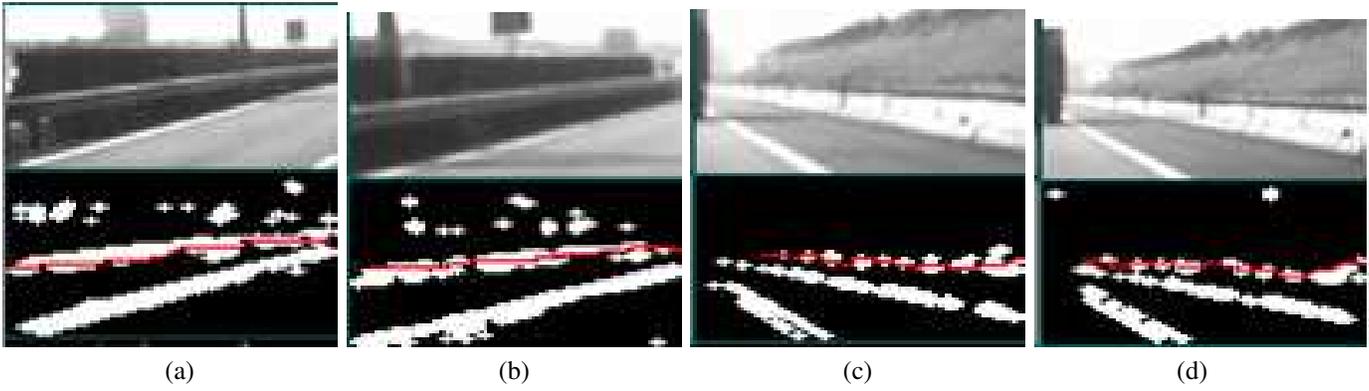


Fig. 9. Examples of correctly detected guard rails: from good edges (a) to non contiguous dashes(d).



Fig. 11. Beltway: guard rail is quite perpendicular to the camera.

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 I

EXECUTION TIME ON HIGHWAY. VD STANDS FOR VEHICLE DETECTION ALGORITHM ONLY, GRD STANDS FOR GUARD RAIL DETECTION ALGORITHM APPLIED TO VEHICLE DETECTION, OBM STANDS FOR OVERLAPPING BOXES MANAGEMENT APPLIED TO GUARD RAIL AND VEHICLE DETECTION. REDUCTION IS THE PERCENTUAL DECREMENT OF OBM REGARDING VD. THE AVERAGE REDUCTION IS THE MEAN OF REDUCTIONS.

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%

TABLE II

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

approach: for example in a beltway, where the guard rail is generally in front of the camera, the vehicle detection algorithm returns a high number of false positives due to guard rails presence and its practically perfect vertical symmetry. The proposed method reduces them of about 25%. The average false positives reduction can be rated about 10%.

No cases of vehicles labeled as guard rail occur. All boxes related to moving vehicles are not considered by guard rail detection algorithm, but, as shown in figures 12 and 13, also guard rails between two cars can be correctly detected.



Fig. 12. Good results reached in a traffic scene.

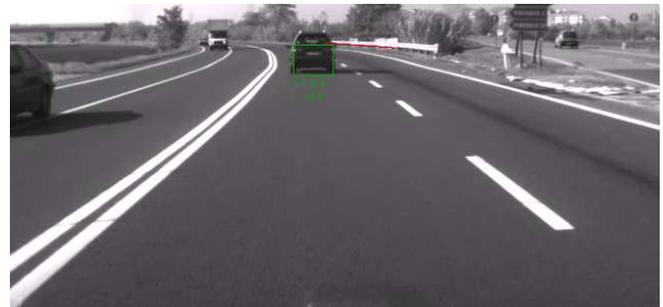


Fig. 13. Co-existence of the guard rail detection and vehicle detection algorithms: the vehicle is anyway detected also when the guard rails detection is running.

Moreover vehicles parked on the edge of the street are discarded by guard rail algorithm so they can be correctly detected as vehicles; the best results are reached when the

vehicles are quite close and frontal (see fig.14).

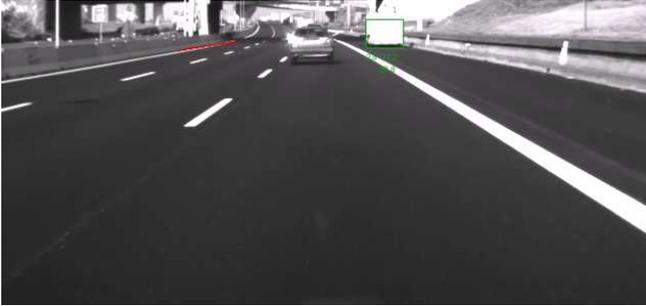


Fig. 14. Correctly detected parked vehicle. No guard rails are detected.

Even if the proposed method has already reached significant results some useful improvement can be made. False positives are never consecutive and may be easily filtered using a simple time correlation method. False negatives may be decreased too: a tracking method may be very useful to trace the presence of guard rails in subsequent frames and, according to egomotion, to predict their next position and validate them. False negatives may be decreased using inertial or odometric data too. If yaw rate and speed are known it is possible to detect whether the vehicle is driving along a curve or not; the algorithm threshold can be modified in order to detect also almost horizontal guard rail. New tests will be done with adverse meteorological conditions and more complex situations in order to verify the robustness of the algorithm.

V. ACKNOWLEDGMENTS

The authors gratefully acknowledge the grant provided by CRF as a support to this research.

REFERENCES

- [1] Broggi, P. Cerri and P.C. Antonello, Multi resolution vehicle detection using artificial vision, *IEEE Intelligent Vehicles Symposium Proceedings*, pp. 310-314, June 2004, Parma, Italy.
- [2] A. Sole, O. Mano, G.P. Stein, H. Kumon, J. Tamatsu and A. Shashua, Solid or not solid: vision for radar target validation, *IEEE Intelligent Vehicles Symposium Proceedings*, pp. 819-824, June 2004, Parma, Italy
- [3] U. Hofmann, A. Rieder and E. D. Dickmanns, Radar and vision data fusion for hybrid adaptive cruise control on highways, *International Conference on Computer Vision System*, pp.125-138, 2001, Vancouver, Canada
- [4] J. Laneurit, C. Blanc, R. Chapuis and L. Trassoudaine, Multisensorial data fusion for global vehicle and obstacles absolute positioning, *IEEE Intelligent Vehicles Symposium*, pp. 138-143, June 2003, Columbus, OH.
- [5] B. Steux, C. Lurgeau, L. Salesse and D. Wautier, Fade: a vehicle detection and tracking system featuring monocular color vision and radar data fusion, *IEEE Intelligent Vehicles Symposium Proceedings*, 2002