

# A Correlation-Based Approach to Recognition and Localization of the Preceding Vehicle in Highway Environments\*

A. Broggi, P. Cerri, and S. Ghidoni

Dipartimento di Ingegneria dell'Informazione,  
Università di Parma, Parma, I-43100, Italy

**Abstract.** In this paper a new approach to the problem of recognizing the preceding vehicle on highways is presented. The system is based on monocular vision. Since on highways the position of the preceding vehicle in the image varies slowly, its previous and current positions are compared using correlation. Such image processing produces a very clear output, which, at a higher level, allows a simple and fast recognition.

## 1 Introduction

Highways represent a particular environment for drivers: the environment is more structured than a common road, some obstacles, like pedestrians and cyclists, are not present, nor exist crossroads or traffic lights, and all vehicles move in the same direction. On the other hand, cars run very fast; this fact requires a different driving style: it is necessary to look even at very distant vehicles and understand whether they are slowing down. The most important vehicle a driver should consider is the preceding one, to which a security distance should be kept. For these reasons, an automatic system capable of understanding where the preceding vehicle is would be very useful. Such systems are already available on some cars, with the name of ACC (Adaptive Cruise Control); most of them are based on radar or infrared technology [1]. In this paper, a system based on vision performing the same task is presented. The information source is a simple low-cost grey-scale 8-bit camera, working at 10 frames per second, connected to a personal computer. The camera is installed on the top of the windscreen in a central position, and acquires  $640 \times 480$  pixels images.

Many methods were developed in order to solve the discussed problem, some of them using a single camera, like the proposed one. A lot of approaches are model-based [2], [3]: even if those methods reach good results they need models that match different vehicle types. Other solutions are based on vehicle features search; symmetry is the most used feature for vehicle detection [4], [5]. The

---

\* The work described in this paper has been developed in the framework of the Integrated Project APALACI - PReVENT, a European research activity funded by the European Commission to contribute to road safety by developing and demonstrating preventive safety applications and technologies.

proposed algorithm uses a different feature: preceding vehicles are supposed to have a “U” shape<sup>1</sup>, so the presence of two vertical edges in correspondence to a horizontal edge is searched; other works are developed using this assumption [6]. Correlation is a well-known method for vehicle tracking [7]: in the proposed work it is used both for detection and tracking; the tracking method is similar to the method proposed by Betke [8], while just basic research has been conducted on detection using correlation.

The purpose of the system being presented is to recognize if there is a preceding vehicle going approximately at the same speed of the experimental vehicle. This means the target vehicle can be found in similar positions in two subsequent frames.

The algorithm can be divided into two parts: the first one is image processing, the second deals with target recognition and tracking. The paper reflects this organization: in Sect. 2 a description of low-level tasks is given, while Sect. 3 describes how it is possible to find a vehicle and to track it. Finally, in Sect. 4 some results are given.

## 2 Computation of Correlation Images

The first step of the algorithm concerns low-level tasks. Some processing is required to obtain images which can make target recognition and tracking easier. As said, these targets remain approximately in the same position within two subsequent frames; for this reason, the concept of correlation can be exploited.

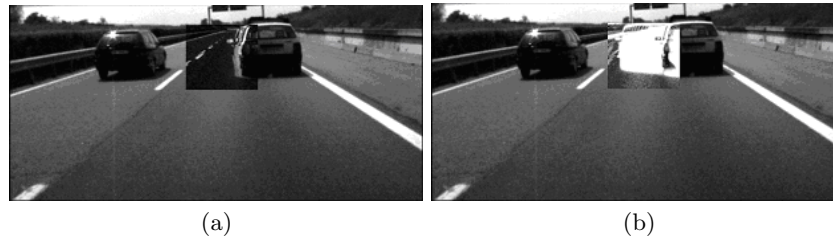
### 2.1 Correlation of Images

Correlation is used to measure how a given quantity changes. Applying this concept to a video, correlation can produce an image that shows how objects change their position in the scene. Such an image can be produced by evaluating, for each pixel, the difference between the current image and the previous ones. The number  $N$  of considered frames of the past should therefore be defined. The grey-level of the generic pixel found in row  $i$  and column  $j$ ,  $P_{i,j}^k$ , of the correlation image at frame  $k$  can be computed as follows:

$$P_{i,j}^k = \sum_{m=1}^N |G_{i,j}^k - G_{i,j}^{k-m}|, \quad (1)$$

where  $G_{i,j}^k$  is the grey-level of the pixel in position  $(i, j)$  at frame  $k$ . The camera acquires 256 grey-level images, therefore  $G_{i,j}^k$  is a number varying in the range 0–255, and the condition  $P_{i,j}^k \leq N \cdot 255$  holds. This means values obtained from (1) exceed 255, and it is thus necessary to limit them by truncation  $P_{i,j}^k$  to a maximum value of 255. Such an operation causes a loss of information beyond a certain variation limit, emphasizing small variations.

<sup>1</sup> Namely, two vertical edges and one bottom edge.



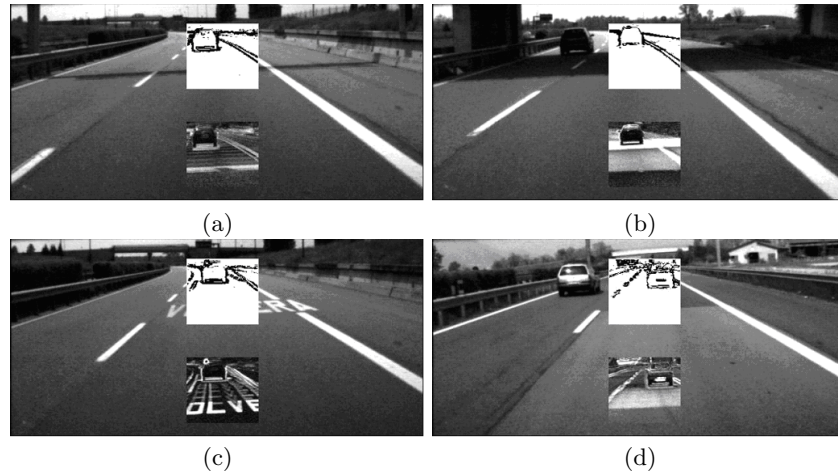
**Fig. 1.** Examples of correlation images: correlation with 2 (a) and 19 (b) frames.

Computation of the correlation image is a time-consuming task, mainly if it involves a lot of past frames. Short computation times can be obtained reducing the use of correlation. Because of the fact that the developed algorithm is intended to work on highways, which are almost straight, it is possible to assume that preceding vehicles will be positioned in a small portion of the image. Therefore it will be sufficient to focus on this small sub-image, called window of interest, to which the evaluation of correlation can be restricted. Two examples of correlation sub-images obtained applying (1) are shown in Fig. 1. The sub-image size is  $110 \times 100$  pixels, and it is overlapped to the analyzed frame. Indeed, the number of frames considered in the computation affects the white trace of the moving car. White color means that analyzed frames are different, while black means they are equal or, at least, similar. Figure 1 shows correlation images obtained using few (a) or many (b) past frames. After some tests, it was found that with  $N = 7$  the best results were obtained.

## 2.2 Correlation of Edges

The concept of correlation can be further exploited by applying it to edges. A useful technique to detect vehicles is to find edges in the image using the well-known Sobel algorithm. This technique, however, also produces some noise, because lots of edges are present, due to the background and to signs painted on the road. The developed algorithm should focus on vehicles which remain almost in the same position between different frames: thus it is possible to find targets by analyzing edges that are persistent. Results of Sobel processing are therefore binarized and saved in memory, and at each new frame edges are compared to those found in the past. Finally, the algorithm generates a binarized image containing edges present at least in five of the seven frames in memory. Such image is called the edges correlation sub-image. This processing is applied only to the window of interest.

The results of the described processing are a good starting point for further steps because a lot of noise is filtered out. Examples can be found in Fig. 2, where the result is superimposed to the original frame acquired by the camera. To compare the effects of the two image processing techniques, also the correlation sub-image is reported in the frames, just under the edges correlation sub-image



**Fig. 2.** Examples of low-level processing. The binarized edges correlation sub-image is superimposed on the frame in the position of the window of interest. The correlation sub-image is also shown below. In (a) and (b) thin and thick shadows do not leave any trace in the edges sub-image. In (c) a horizontal road sign is completely filtered, and in (d) the same happens to the junction of two patches of asphalt with different colors.

for a better comparison. As Fig. 2 shows, horizontal road signs and shadows are not present in the edges correlation image.

### 3 Vehicle Recognition and Tracking

This section deals with the problem of recognizing the preceding vehicle by analyzing the results of image processing previously described. At this level, the system switches between two different states: vehicle recognition and tracking. The algorithm starts in the recognition phase, and tries to find a vehicle in the window of interest. When this happens, it starts tracking it, and continues until it is lost; in this case, the system goes back to the recognition phase.

#### 3.1 Recognition Phase

The aim of this phase is to test if there is a vehicle in the window of interest. The algorithm starts by analyzing the edges correlation sub-image, looking for a rectangular shape of appropriate size. From Fig. 2 it is clear that a vehicle's outline will result in long black lines in the edges correlation image; so the algorithm looks for long black edges. Because the analyzed image is binarized, as said in Sect 2.2, it is possible to enumerate black pixels for each row and column. This simple computation, however, does not consider if pixels are contiguous or not, so the same value would result for two lines with the same number of black pixel disposed in very different ways. Rather, a vehicle's edge will be an uninterrupted segment, so it is necessary to distinguish among black pixels which are contiguous from those which are not.

**Edge Detection.** In order to detect a vehicle's edge, a value, depending on the number of black pixels and on their placement according to a simple rule, is assigned to each row and column of the edges correlation sub-image. A weight is assigned to each black pixel, that is equal to the number of preceding black pixels plus one. Each white pixel has zero weight. Finally, all weights are added together to obtain the final number assigned to the row or column.

The described method is able to find the most visible edges. The algorithm then selects the two columns associated with the highest values to detect lateral edges of the vehicle. As an additional constraint, the two columns should be at a minimum distance from each other of five pixels, otherwise they may belong to the same edge (which may be composed by few black columns). Subsequently, the lower bound of the vehicle is searched for; this is done in the same way, but examining rows instead of columns. Because the left and right edges are already known, a rough estimation of the vehicle's size in the image can be done, allowing a restriction of the lower edge search area. Often the upper edge of the vehicle is not present in the region where the correlation is computed; in other cases, it is positioned in a region where the background produces a lot of noise, as can be observed in Fig. 2 (a), (c) and (d). Thus, the algorithm does not look for the vehicle roof, which however is not important for the estimation of the vehicle's position, and can therefore be neglected: the roof position can be used to classify the kind of vehicle. A fictitious upper edge is nevertheless chosen to represent the upper limit of the window of interest, because some checks should be anyway carried out on the area inside the edges. Even if the upper limit does not come from a real detection, when a vehicle is in the scene, it is displayed in a rectangle. In this way a box that may contain a vehicle is anyway obtained.

**Vehicle Check.** Edges computed in the way described so far are found in each frame. It is therefore necessary to understand whether or not a vehicle is present, so some checks are performed. If at least one fails, no vehicle is found in the current frame, otherwise the recognition is successful.

*Size Check.* The first test is simple and concerns the mutual positions of the lateral and lower edges of the vehicle. Moreover, if an edge is shorter than a threshold (ten pixels), the check fails, because the recognition of a vehicle which is too small in the image would be too unreliable.

*Correlation Check.* The second test considers the correlation sub-image, and verifies that in the rectangle of interest the correlation is high: this ensures that the box represents a vehicle preceding the experimental one, going at about the same speed. This test is carried out by adding together all values of the correlation image in the rectangle of interest, normalizing it over the total number of pixels and comparing the obtained value to a threshold.

*Edge Check.* The last check is performed on the edge correlation sub-image. As said, the selected edges are the row and columns with the highest numbers of black pixels, considered together with their positions. But if the sub-image

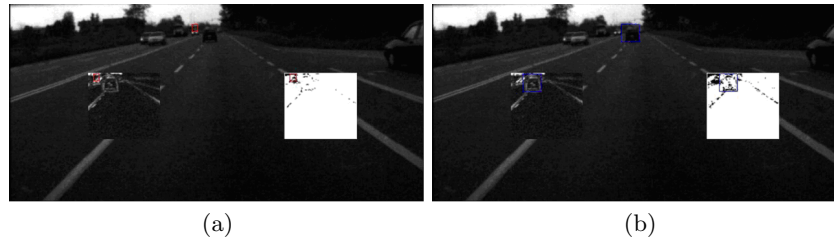
has very few black pixels, edges will be composed by a large number of white pixels, and the recognition will be too unreliable. To avoid this situation, the third check verifies that the edges are composed by pixels that are black for a fraction exceeding a threshold. The fraction that gave best results is 40%: it could seem low, but it is enough to filter out the great majority of false positives, still keeping a good recognition. This check of course is not performed on the upper edge. Because some shapes with very evident lateral edges can satisfy the previous condition even if the lower edge is barely visible, an additional test is done on it, requiring that at least 30% of its pixels are black.

### 3.2 Tracking Phase

This phase is reached by the algorithm when a vehicle is found in the recognition phase. At this point, the information on presence and position of a vehicle is known. So, the first action that is performed is to check whether the vehicle recognized in the past is still in the scene. For this reason, the portion of grey-level image containing the vehicle in the past frame is compared with the content of the same rectangle in the last acquired image. Comparison, again, means correlation between the two portions of image inside the rectangles. Since a vehicle should be tracked also if it is slowly moving in the image, correlation with translated copies of the rectangle of interest found in the past are also computed. The box is translated in all directions, saving the corresponding correlation values. When all possible translations are considered, the best matching position suggests the motion direction of the vehicle, and the rectangle containing the target is accordingly moved. It is even possible that the rectangle needs to be resized, if the vehicle slowly gets closer or further, so, in the edges correlation sub-image, rows and columns near current edges are analyzed, and may become new edges if they have higher number of black pixels or a better placement. After translation and resizing, a new box is found. It should anyway be checked, so all tests described in the case of the recognition phase are performed.

To produce robust results, the presence of a vehicle is declared only after it has been recognized for a number of frames; in this way some spurious false positives are filtered out, with the small drawback that vehicles are found a little later than they could be. Once a vehicle has been found, it is tracked until it fails checks for five successive frames. This behavior is necessary to continue tracking the vehicle in some situations, when particular lighting conditions or shadows may produce some critical frames, causing at least one check to fail. For instance, when a vehicle goes under an overpass, the shadow causes the loss of the lower edge of the car. It was found that good results were found declaring the presence of a vehicle after it was detected for two frames.

**Automatic Thresholds.** Three thresholds used by the algorithm are particularly crucial and must therefore be carefully chosen; one is used to binarize the Sobel edge image, another to define how many times an edge should be present in the past to be in the edges correlation sub-image, and the last one is used in the correlation check. After some testing, values giving good results were found, but



**Fig. 3.** In (a) a frame is processed with normally used thresholds. In (b) the same frame is processed using modified thresholds, which makes recognition possible.

there are still different values that could perform better in particular situations. So, an automatic adjustment of such thresholds has been implemented.

At each new frame, results of low-level processing are analyzed to understand if they bear poor information; in this case, the mentioned thresholds are modified trying to get better low-level images. If, for example, the edges correlation sub-image is composed by too few black pixels, like in Fig. 3 (a), the binarization threshold is lowered, as well as the number of frames in the past each edge must appear in to be visible in the edges correlation sub-image. Fig. 3 (b) represents the same frame as in (a) processed using these modifications. The threshold of the correlation check is modified when the target vehicle is particularly small. Such check, actually, often fails because each movement of the camera, even of a few pixels, becomes significant if related to the small number of pixels inside the rectangle of interest. Threshold corrections can improve edge detection, but also increase noise in the low-level images.

## 4 Results

The algorithm described so far was tested on some sequences acquired in highway environments, in different lighting conditions, for a total of more than 5 000 frames. When a preceding vehicle (car or truck) is present in the scene, the recognition rate is 70.53%. Otherwise, when there is no vehicle in the acquired scene, the false detection rate is only 5.71%. While the latter is completely satisfactory, some remarks can be made about the former, which may be improved. Some problems were encountered due to bad lighting conditions, such as at sunset, when images appear dark, like it can be seen in Fig. 3. In some cases the automatic thresholds are able to fix the problem, but in other cases this does not happen, and, in general, the recognition becomes harder.

### 4.1 Further Improvements

An open issue is the fact that the edges correlation sub-image still does not filter lane markers. Along bends, these become curve and may cause some columns to have a high number of nearby black pixels, leading to an erroneous evaluation of the rectangle of interest. Moreover, by lowering the number of frames used in the

corresponding computation the algorithm may be able to work also in contexts different from highways. An automatic recognition of the kind of environment may therefore be very useful to obtain the best behavior in different roads.

## 4.2 Processing Time

The computational time is a strong aspect of the algorithm. On a Pentium 4 processor, working at 2.80 GHz, with 1 GB RAM, the average and maximum processing times for each frame are 14 ms and 20 ms respectively, which allow the algorithm to work at a frequency of up to 50 Hz.

When the algorithm is in the recognition phase, the main part of time is used to compute correlation images (88%), while the remaining part is spent in processing edges (10%) and doing high-level tasks (2% only). In the tracking phase these proportions slightly change, but the computation of the correlation sub-image still takes the longest computational time (79%), while edges processing takes 10% of the time, while high-level now is 11% of total time. In the last case, high-level tasks are more time consuming than in the former because they need correlation too, when the rectangle of interest of the previous frame must be superimposed and best matched with the corresponding part of the last frame.

## References

1. Abou-Jaoude, R.: ACC Radar Sensor Technology, Test Requirements, and Test Solutions. *IEEE Trans. on Intelligent Transportation System* 4 (2003) 115–122
2. Denasi, S., Quaglia, G.: Obstacle Detection Using a Deformable Model of Vehicles. In: *Procs. IEEE Intelligent Vehicles Symposium 2001, Tokyo, Japan* (2001) 145–150
3. Fleischer, K., Nagel, H.H., Rath, T.M.: 3D-Model-based-Vision for Innercity Driving Scenes. In: *Procs. IEEE Intelligent Vehicles Symposium 2002, Paris, France* (2002)
4. Kuehnle, A.: Symmetry-based vehicle location for AHS. In: *Procs. SPIE - Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS. Volume 2902., Orlando, USA* (1998) 19–27
5. Broggi, A., Cerri, P., Antonello, P.C.: Multi-Resolution Vehicle Detection using Artificial Vision. In: *Procs. IEEE Intelligent Vehicles Symposium 2004, Parma, Italy* (2004) 310–314
6. Zeng, Z., Ma, S.: An Efficient Vision System for Multiple Car Tracking. In: *Procs. IEEE 16th Intl. Conf. on Pattern Recognition. Volume 2.* (2002) 609–612
7. Cao, G., Jiang, J., Chen, J.: An Improved Object Tracking Algorithm based on Image Correlation. In: *Procs. IEEE Intl. Symp. on Industrial Electronics, Rio de Janeiro, Brasil* (2003) 598–601
8. Betke, M., Haritaoglu, E., Davis, L.: Multiple Vehicle Detection and Tracking in Hard Real-time. In: *Procs. IEEE IV Symposium'96, Tokyo, Japan* (1996) 351–356