Vision-based Blind Spot Detection using Optical Flow

M. A. Sotelo[1], J. Barriga[1], D. Fernández[1], I. Parra[1],
J. E. Naranjo[2], M. Marrón[1], M. Gavilán[1]

[1] Department of Electronics, University of Alcalá, Alcalá de Henares, Madrid, Spain
{sotelo, llorca, marta, parra}@depeca.uah.es
[2] Industrial Automation Institute, CSIC, Arganda del Rey, Madrid, Spain
jnaranjo@iai.csic.es

Abstract. This paper describes a vision-based system for blind spot detection in intelligent vehicle applications. A camera is mounted in the lateral mirror of a car with the intention of visually detecting cars that can not be perceived by the vehicle driver since they are located in the so-called blind spot. The detection of cars in the blind spot is carried out using computer vision techniques, based on optical flow and data clustering, as described in the following lines.

Keywords: Computer Vision, Optical Flow, Blind Spot, Intelligent Vehicles.

1 Introduction

A vision-based blind spot detection system has been developed for Intelligent Vehicle Applications. The main sensor for providing detection of cars is vision. Images are analyzed using optical flow techniques [1] in order to detect pixels that move in the same direction as the ego-vehicle. Pixels producing movement as described are grouped following the clustering techniques described in [2]. The resulting clusters are considered as potential vehicles overtaking the ego-vehicle. A double-stage detection mechanism has been devised for providing robust vehicle detection. In a first stage, a pre-detector system computes the mass center of the resulting clusters and determines whether the detected cluster is a potential vehicle according to the size of detected pixels.

In a second stage, another detector looks for the appearance of vehicles frontal parts. Any object looking like the frontal part of a vehicle is considered as a potential vehicle, whenever the mass center pre-detector triggers the pre-detection signal. Thus, a sufficiently big object in the image plane, producing optical flow in the same direction as the ego-vehicle, and exhibiting a part similar to the frontal part of a car is validated as a car entering the blind spot. The position of the vehicle in the image plane is computed and tracked using a Kalman Filter. Tracking continues until the vehicle disappears from the scene, and an alarm signal is triggered indicating the driver that a vehicle has entered the blind spot zone.

2 System Description

The description of the algorithm is provided in figure 1 in the form of flow diagram. As can be observed, there are several computation steps based on optical flow computing at image level, pixel-wise clustering, analysis of clusters and final vehicle detection. As previously stated, the system relies on the computation of optical flow using vision as main sensor providing information about the road scene. In order to reduce computational time, optical flow is computed only on relevant points in the image. These points are characterized for exhibiting certain features that permit to discriminate them from the rest of point in their environment. Normally, these salient features have prominent values of energy, entropy, or similar statistics. In this work, a salient feature point has been considered as that exhibiting a relevant differential value. Accordingly, a Canny edge extractor is applied to the original incoming image. Pixels providing a positive value after the Canny filter are considered for calculation of optical flow. The reason for this relies on the fact that relevant points are needed for optical flow computation since matching of the points have to be done between two consecutive frames.

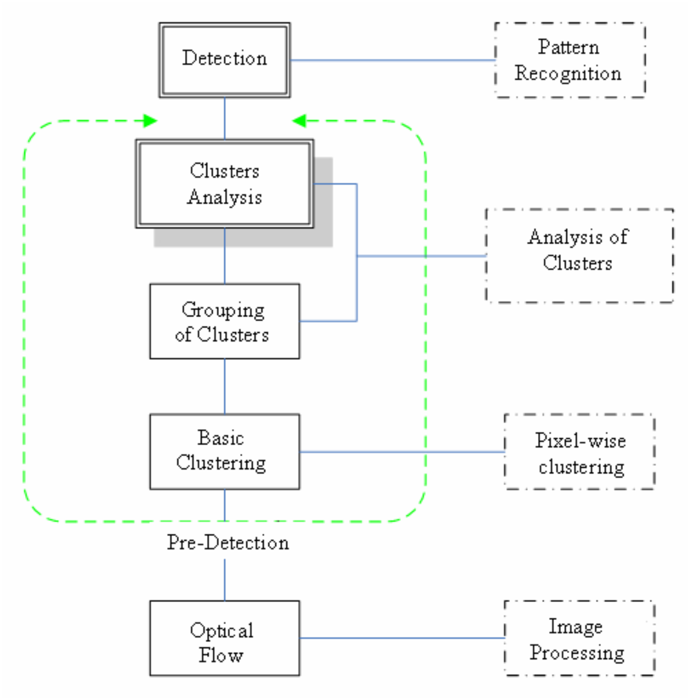

Fig. 1. Flow Diagram of the Blind-spot detection algorithm.

After that, Canny edge pixels are matched and grouped together in order to detect clusters of pixels that can be considered as candidate vehicles in the image. Classical clustering techniques are used to determine groups of pixels, as well as their likelihood to form a single object. Figure 2 depicts a typical example of matched points after computing optical flow and performing pixels clustering.

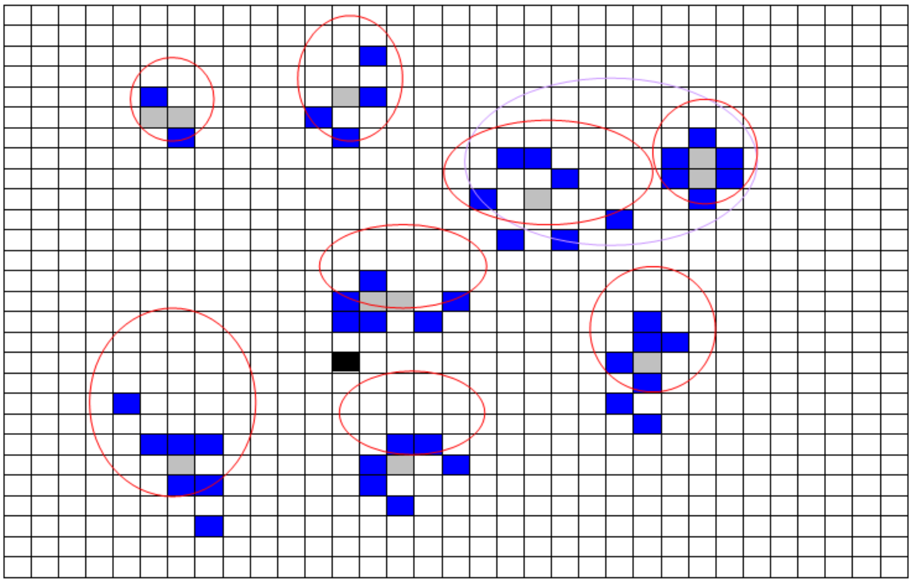

Fig. 2. Clustering of pixels providing relevant optical flow.

Pixels in blue represent edge points that have produced relevant optical flow in two consecutive frames. Red ellipses stand for possible groups (clusters) of objects. Violet ellipses represent ambiguous groups of objects that could be possibly split in two. Gray pixels represent the mass center of detected clusters. Even after pixels clustering, some clusters can still be clearly regarded as belonging to the same real object. A second grouping stage is then carried out among different clusters in order to determine which of them can be further merged together into a single blob. For this purpose, simple distance criteria are considered. As depicted in figure 3, two objects that are very close to each other are finally grouped together in the same cluster. The reason for computing a two-stage clustering process relies on the fact that by selecting a small distance parameter in the first stage interesting information about clusters in the scene can be achieved. Otherwise, i.e. using a large distance parameter in single clustering process, very gross clusters would have been achieved, losing all information about the granular content of the points providing optical flow in the image.

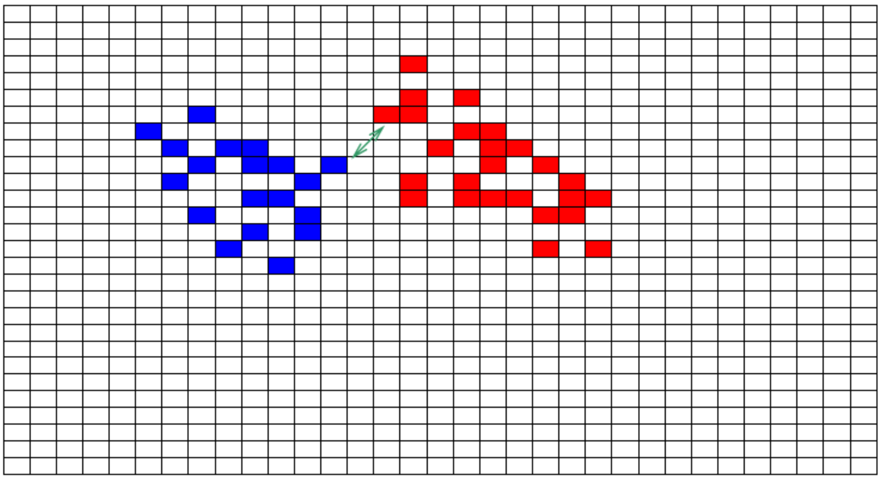

Fig. 3. Grouping of close clusters in a second clustering stage.

The selected clusters constitute the starting point for locating candidate vehicles in the image. For that purpose, the detected positions of clusters are used as a seed point for finding collection of horizontal edges that could potentially represent the lower part of a car. The candidate is located on detected horizontal edges that meet certain conditions of entropy and vertical symmetry. Some of the most critical aspects in blind spot detection are listed below:

1. Shadows on the asphalt due to lampposts, other artifacts or a large vehicle overtaking the ego-vehicle on the right lane.
2. Self-shadow reflected on the asphalt (especially problematic in sharp turns like in round-about points), or self-shadow reflected on road protection fences.
3. Robust performance in tunnels.
4. Avoiding false alarms due to vehicles on the third lane.

The flow diagram of the double-stage detection algorithm is depicted in figure 4. As can be observed, there is a pre-detector that discriminates whether the detected object is behaving like a vehicle or not. If so, the frontal part of the vehicle is located in the Region Of Interest and a pre-warning is issued. In addition, the vehicle mass centre is computed. In case the frontal part of the vehicle is properly detected and its mass centre can also be computed a final warning message is issued. Vehicle tracking starts at that point. Tracking is stopped when the vehicle gets out of the image. Some times, the shadow of the vehicle remains in the image for a while after the vehicle disappears from the scene, provoking the warning alarm to hold on for 1 or 2 seconds. This is not a problem, since the overtaking car is running in parallel with the ego-vehicle during that time although it is our of the image scene. Thus, maintaining the alarm in such cases turns out to be a desirable side-effect.

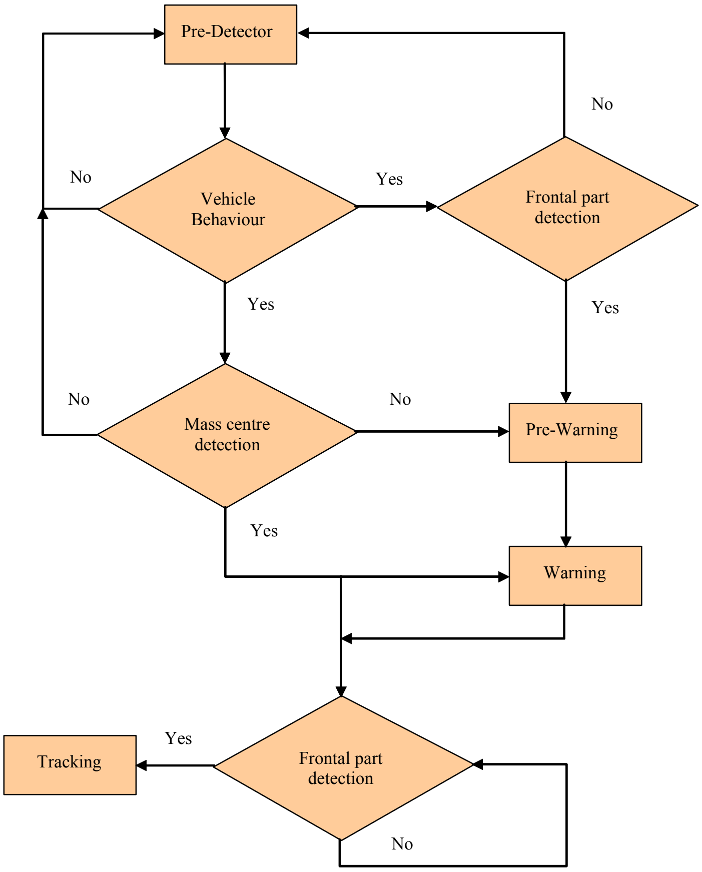

Fig. 4. Pre-detection Flow Diagram.

After locating vehicle candidates, these are classified using a SVM classifier previously trained with samples obtained from real road images.

3 Implementation and Results

A digital camera was mounted in the lateral mirror of a real car equipped with a Pentium IV 2.8 GHz PC running Linux Knoppix 3.7 and OpenCV libraries 0.9.6. The car was manually driven for several hours in real highways and roads. After the experiments, the system achieved a detection rate of 99% (1 missing vehicle), producing 5 false positive detections. Figure 5 shows an example of blind spot detection in a sequence of images. The indicator depicted in the upper-right part of the figure toggles from green to blue when a vehicle enters the blind spot area (indicated by a green polygon). A blue bounding box depicts the position of the detected vehicle.

Fig. 5. Example of blind spot detection in a sequence of images. The indicator in the upper-right part of the figure toggles from green to blue when a car is detected in the blind spot.

Our current research focuses on the development of SVM-based vehicle recognition for increasing the detection rate and decreasing the false alarm rate, as demonstrated in [3], where SVM was used for vehicle detection in an ACC application.

Acknowledgments. This work has been funded by Research Project CICYT DPI2005-07980-C03-02 (*Ministerio de Educación y Ciencia, Spain*).

References

1. Giachetti, A., Campani, M., Torre, V.: The use of optical flow for road navigation. IEEE Transactions on Robotics and Automation, vol. 14, 1, (1998)
2. Kailunailen, J.: Clustering algorithms: basics and visualization. Helsinki University of Technology. Laboratory of Computer and Information Science. T61.195, Special Assignment 1, (2002)
3. Sotelo, M. A., Nuevo, J., Ocaña, M., Bergasa, L. M.: A Monocular Solution to Vision-Based ACC in Road Vehicles. Lecture Notes in Computer Science, Vol. 3643. Springer-Verlag, Berlin Heidelberg New York (2005) 507–512