A real time car tracker

F. Marmoiton, F. Collange, Philippe Martinet, J P Derutin

A real time car tracker

F. Marmoiton, F. Collange, P. Martinet, J.P. Derutin
LASMEA
UMR 6602 CNRS - Université B. Pascal
Aubière, FRANCE 63177

Abstract

This paper details a vehicle detection and localization algorithm from a single perspective view. A video camera, embedded in a car, provides a grey scale image of a lead car. This lead vehicle is equipped with three visual marks to make easier low and high level processes. The algorithm is designed for real time implementation. It is conceived from a direct 3D localization method coupled with a tracking method based on a rigidity criteria of the trajectories of N points. The processes are implemented on a specialized hardware architecture to satisfy real time constraints.

Keywords : *computer vision, real time implementation, car tracking.*

1 Introduction

Vision techniques to create efficient cars driving devices have a large potential of development. Researchs around the autonomous guided vehicles theme show the possibilities of such methods for vehicle guidance on roads, obstacle detection or traffic scene interpretation.

The LAboratoire des Sciences et Materiaux pour l'Electronique, et d'Automatique (LASMEA) worked on vision cars systems like estimation of the current position of a vehicle with respect to a normalized road, obstacle detection and tracking with a range finder coupled with a camera or vehicles control [10, 7]. The laboratory works today on a driver assistance based on a monocular camera sensor.

Because of the speed of the cars, such systems involve to be efficient short processing times. Vision based devices give large amount of data so need high speed calculator and dedicated architecture to obtain a sufficient performance level. The LASMEA works on vision hardware architecture to test its algorithms in a real environment.

Platooning operations [2, 6] consist of automatic convoys of vehicles where one or more slaves follow a master. Convoys are usually led by a human being and slaves are driven by an automatic device which involves a lateral and longitudinal control.

In this context, this paper only concerns the perception aspects of a platooning device or of an Adaptative Intelligent Cruise Control. It describes a software and hardware vision system to detect obstacles in a road scene.

Our approach aims to be able to localize one or many vehicles equipped with visual marks [9] from a single perspective view. An embedded camera provides a grey level image of road scenes.

To make easier low and high level processes, three visual marks are used on a lead vehicle. These marks are three lights, one on top and two at the back of the lead car. This installation can be generalized to actual cars with the two back lights, and the other included into the third stop light.

In a camera image of the lead car, the 2D projections of the three visual marks are detected and the lead vehicle is then localized in the camera coordinate system. A tracking module permits to recognize the vehicle in an image sequence.

We chose and designed both the algorithm and the

Figure 1: The lead vehicle and its visual marks.

embedded computer to meet video real time criteria (40 ms per image). To ensure good performances, low level processes are limited to regions of interest (prediction verification method).

This paper is divided into two parts. In a first part, we present the different steps to obtain the position and the orientation of the lead car from the camera image and to have a good reliability of our device. In a second part, we describe the computer architecture and the real time implementation.

2 The algorithm

The first difficulty to develop an efficient perception vision algorithm is to be able to detect and recognize vehicles in a road scene. The collision avoidance devices involve a high level of robustness and reliability. The algorithm doesn't have to be perturbed by false alarms, noise, or events in the road scene

The second difficulty is to be able to track a vehicle along an image sequence to update the information about this moving object. This step must be robust enough to filter all external perturbations. A high level module must measure the quality of the tracking to detect problems and target losses.

2.1 Vehicle detection

Vehicle detection in a monocular image is a complex problems. The camera is embedded in a car, so solutions to detect moving object like the computation of the velocity field [1] in the image plane are not suitable. Others approaches like [8] aim to detect a vehicle in a grey scale image. The extraction of geometrical characteristics like coins, segments or correlation methods permit to recognize a vehicle in the image plane. In our applications, the lead vehicle is equipped with visual marks. This permits to avoid problems due to a lack of contrast between the visual features (here the marks projections) and the background. So, detecting a vehicle in the image plane doesn't involve to detect a geometrical characteristic of the car but to look for the three visual marks in the 2D image plane.

The extraction of the marks of the image plane is made of two step. An adaptative thresholding permits to extract the most luminous pixels of the image. The unpredictable variations of the road scene illumination commands the use of an adaptative thresholding based on the grey level histogram. The second step aims to bring together the luminous pixels into sets of pixels, and to compute the centers of gravity of these sets of pixels.

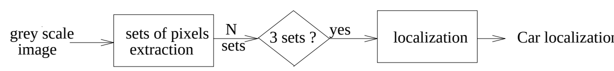

Figure 2: Low level extraction and localization

If three projections of visual marks are detected in the image, we assume that these marks belong to the lead car. The three extracted marks are used to localize this car. The localization method is developed by DeMenthon and Davis [3]. The car is localized from the three 2D points view under a weak perspective assumption [5], knowing the 3D position of the marks in a coordinate system fixed to the lead car, and the intrinsec parameters of the camera. The three points extracted in the image plane are used to compute the position of the lead car in a coordinate system fixed to the camera.

A vehicle is identified by its three 2D marks projections, by its 3D position and orientation and by its dynamic comportement. If in three successive images three visual marks are extracted and the localization is successfull, we assume that the three visual marks belong to a vehicle. To complete the parametrization of our mobile, the 3D movements are computed from the last three localization results. The mobile is completly identified and the tracking module can be started.

2.2 Vehicle tracking and supervision

As we know that there is a vehicle in the field of view of the camera (results of the step of vehicle detection), we want now to follow it through an image sequence, or to detect when the car is no more visible in the camera field of view.

Two modules must be developed, a module to track the car in the images sequence, and another to supervise the tracking to detect target disparitions.

2.2.1 Tracking module

As our data are perturbed by noise, the correspondance of occurences of feature point along an image sequence is difficult to determine. The method employed here is a prediction verification method (cf fig.3). We must filter the visual marks projection between noise and false detections. The first two steps of the algorithm permits to sort out the good features. Regions of interest permit to exclude non representative data. A second filter, based on a rigidity criteria of the trajectory of the three points, permits to sort out the data used for the localization.

The knowledge of the 3D motion of the car permits to place regions of interest in the image plane. The amount of data to be processed is limited to these regions of interest. A part of the noise or false alarms may be avoided.

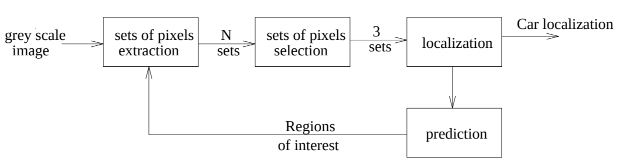

Figure 3: The tracking module

Visual marks are detected in the regions of interest. The method to extract the marks is the same as in the first step of vehicle detection.

A second step aims to choose the three visual marks in the image plane. The noise immunity of the algorithm depends on this step. Noise due to problems during the digitalization step, reflections or events in the road scene is filtered. When an obstacle appears between the camera and the tracked car, or if a light is no more in the field of view of the camera, data may be lost. This step ensures a good algorithm behaviour in these unexpected situations. So we need to choose between marks if more than three marks have been detected or to predict a mark position in the case of lack of data.

To select the good marks, we suppose that the three characteristic points have homogeneous trajectories. A rigidity criteria between the trajectory of the three points permits to determine if the point is a car mark projection or not.

We compute all possible trajectories with one point per window of interest. So if s_p points are selected in window W_i, there are T possible trajectories :

$$T = \prod_{i=1}^{N} (s_i) \tag{1}$$

with N the number of characteristic points of the model (number of region of interest). If p_i is the predicted point, p_i is the center of the region of interest W_i, and $s_{i,n}$ is the nth point of the window W_i.

In each region of interest W_i and for each detected mark, we compute the polar coordinate $(d_{i,n}, \alpha_{i,n})$ and the cartesian coordinates (x_i, y_i) of the selected points $s_{i,n}$ in an image coordinate system centered on p_i.

For each combination t, we calculate an average model of the three points view with :

$$\overline{d}_t = \frac{1}{N} \sum_{i=1}^{N} d_{i,n} \tag{2}$$

$$\overline{x}_p = \frac{1}{N} \sum_{i=1}^{N} x_i(p) \tag{3}$$

$$\overline{y}_p = \frac{1}{N} \sum_{i=1}^{N} y_i(p) \tag{4}$$

From 3 and 4, we compute :

$$\overline{\alpha}_t = \sum_{i=1}^{N} \arctan\left(\frac{\overline{y}_p}{\overline{x}_p}\right) \quad (5)$$

Then, we compute the distance between the average model placed with $\overline{d}_t$ and $\overline{\alpha}_t$, and the selected points to search the combination which minimizes this distance.

This distance must be smaller than a threshold to select the combination. Else, two points combinations are tested. Predictions are returned if no three and no two points combination can be returned.

From the three 2D estimated marks, the position and the orientation of the lead car is processed as described above. The results are then used to place and size regions of interest for the next iteration.

2.2.2 Supervisor module

During the tracking stage, we suppose above that the lead car must be found in the image. But partial or complete occultation of the car, problems of divergence of the tracking, or if the car is too far to be correctely detected may cause a bad running of the tracking module. The supervisor module aims to detect tracking problems to launch if necessary a new step of vehicle detection. The supervisor system analyses the results of the 2D low level extraction.

If in the last three images, three visual features have been selected and use to localize the vehicle, we suppose that the tracking module works well and that the extracted features belong to a car of known position and orientation.

Target losses are detected by a time criteria. A measure of the time of localization with 2 lights, localization with no light permit to decide to continue or to stop tracking. If this time becomes too important, we suppose that the target is lost and we consider that there is no car in the image. The step of vehicle detection is then run to detect if the car reappears in the image.

3 Implementation

To obtain real time processes, the computer architecture is adapted to software processes.

The hardware is separated into two worlds:

- A distributed memory MIMD world well adapted to localized treatments. The amount of data image processing can be distributed on the different processors. This world is conceived around a Video Node board based on communicated processors.
- a sequential world manages high level processes. It is based on a 68040 CPU board running in VxWorks software environment.

Data transferts between these worlds are managed via a VME bus.

Each Video Node board includes four communicated processors ring connected [4]. Communicated processors data links ease efficient data exchanges. Each Video Node can access its private RAM and to a special memory where the image data is stored.

Communicated processors and the 68040 CPU board are linked through a special VME interface.

Video nodes, which have a direct access to the video flow, processes low level computing. A video node, called root, synchronizes workers which manages each one a region of interest according to the informations (position, size) given by the high level process.

Results of workers are collected by the root processor and send back to the MC68040 board where high level tasks are sequentially implemented.

The simplicity of the localization method makes that the MC68040 is powerfull enough to satisfy real time constraints.

Experimental results

Software and hardware systems are tested on video taped images collected in a real outdoor environment. These images sequences offers difficulties like hills, bumps, S-curves or traffic lights.

The visual marks are well detected, extracted and the car is located with a good accuracy.

Here is an exemple of difficulty frequently encountered in our sequence:

Figure 4: A grey levels image

Figure 4 shows the grey scale image, entry of the algorithm. This image is obtained from a grey scale video camera with an infrared filter to ease low level extraction. Four lights have been detected: three concern the tracked car and the fourth belongs to a motorcycle situated at the left side of the road.

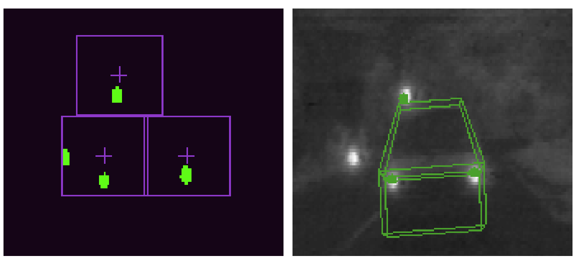

Figure 5: Extraction and localization results.

Figure 5 shows a zoom of the image of figure 4 after processing. Windows of interest are sized and placed around the last 2D projection. Luminous pixels are thresholded: four 2D projections are detected. W_1 contains $s_{1,1}$ and $s_{1,2}$, W_2 contains $s_{2,1}$ and W_3 contains $s_{3,1}$. Between these four detections, one must be eliminated to localize the vehicle. We detail here the sets of pixels selection.

If we select one point per region of interest, there is two possible trajectories: $(s_{1,1}, s_{2,1}, s_{3,1})$ or $(s_{1,2}, s_{2,1}, s_{3,1})$.

For each combination the average model gives:

Traj	$\overline{\alpha}$	$\overline{d}$
0	-96.71	5.74445
1	-162.76	12.72671

The average distance between the average model and the $p_{i,j}$ gives 1.94404 and 13.62630 pixels respectively for trajectory 0 and 1. The solution retained is then trajectory 0.

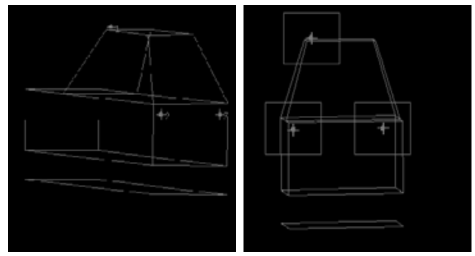

Figure 6: Three points extraction

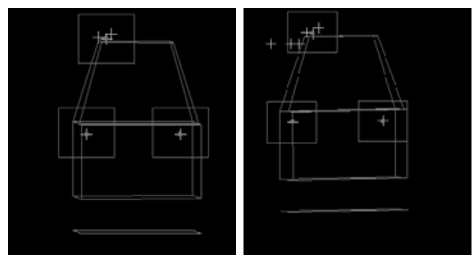

Figure 7: N points extraction

Figures 6 and 7 show algorithm points selection results (crosses represent extracted points). Figure 6 shows results where only three visual marks have been extracted. Figure 7 shows the solution proposed if N points have been extracted. The processing time in tracking mode never exceeds 40 ms. The time to process the sets of pixels extraction is data dependant. However, this time can be estimated to around 28 ms. The amount of data to be processed explains this large computation time. The set of pixel extraction lasts 2 ms, localization 3 ms and prediction 1 ms.

4 Conclusion and perspective

Software and hardware architectures are validated in real conditions. The whole set can be integrated into existing systems as a module for a number of applications like convoying operations or intelligent cruise control on highways. This involves the development of a high level decision module which ensures the connection between the perception modules and the control module.

One important limitation is that this algorithm tracks only one vehicule. This is not sufficient in normal traffic situations even if cars are marks equipped. So, to complete this system, future versions must allow multi-target tracking and localization.

The multi-target algorithm must process target disparitions and apparitions, and target occultations. Such a module needs the implantation of a high level device to manage all moving obstacles. Environmental informations can be brought by parallel processes like straight line detection to have a good supervision of the whole system.

References

[1] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of the optical flow techniques. *The International Journal of Computer Vision*, 12:43–77, February 1994.

[2] P. Daviet and M. Parent. Platooning for small public urban vehicles. In *fourth International Symposium on Experimental Robotics*, Stanford, California, USA, July 1995.

[3] D. F. Dementhon and L. S. Davis. New exact and approximate solutions of the three point perspective problem. *IEEE Transactions on Pattern analysis and Machine Intelligence*, pages 1100–1105, November 1992.

[4] J. P. Derutin, P. Legrand, and R. Canals. Edge and region segmentation processes on the parallel vision machine Transvision. In *Computer Architecture for Machine Perception, CAMP'93 Workshop*, pages 410–420, New Orleans, USA, December 1993.

[5] R. Horaud, S Christy, and F. Dornaika. Object pose : The link between weak perspective, para perspective, and full perspective. Technical Report 2356, Technical report INRIA, December 1994.

[6] H. Mäkelä, P. Kaarmila, and K. Koskinen. Convoy navigation. In *3thrd IFAC symposium on Intelligent Autonomous Vehicle*, volume 1, pages 31–36, Madrid, Spain, March 1998.

[7] P. Martinet, C. Thibaud, B. Thuilot, and J. Gallice. Robust controller synthesis in automatic guided vehicles applications. In *AVCS'98*, Amiens, France, to appear in July 1998.

[8] N.D. Matthews, P.E. An, D. Charnley, and C.J. Harris. Vehicle detection and recognition in greyscale imagery. In *2d IFAC symposium on Intelligent Autonomous Vehicle*, pages 31–36, Espoo, Finland, June 1995.

[9] E. Montagne, J. Alizon, P. Martinet, and J. Gallice. Real time 3D location of a car from three characteristic points observed in a video image sequence. In *7th IFAC/IFORS Symposium on Transportation Systems*, volume 2, pages 385–390, Tianjin, China, August 1994.

[10] L. Trassoudaine, J. Alizon, J. Gallice, and F. Collange. Visual tracking by a multisensorial approach. In *1st IFAC/ Int. Workshop on Intelligent Autonomous Vehicles*, pages 113–118, Southampton, April 1993.