A GLOBAL ROAD SCENE ANALYSIS SYSTEM FOR AUTONOMOUS VEHICLES

F. Jurie, P. Martinet, J. Gallice
L.A.S.M.E.A. - CNRS U.R.A 1793, Université Blaise Pascal, F-63177, Aubière Cedex
email: jurie@zephyr.univ-bpclermont.fr

Abstract. This paper reports on recent developments of the PSA-LASMEA autonomous vehicle. Since 1993, a new vision module have been added, extending the scope of the perception system. It has been implemented on a specific architecture based on DSP processors (96002). Both the algorithm and the architecture are described in this paper. Experimental results are also given.

Key Words. Vision based control, Real time image processing, Vehicle modelling, Adaptive control.

1. HIGH SPEED VEHICLE GUIDANCE

Research on autonomous vehicle navigation from imagery acquired from a vehicle mounted camera are being conducted in several laboratories. Results obtained by Dickmanns et al. (1992) are actually the most relevant ones. Works of Kanade et al. (1987) and Waxman et al. (1987) in the United States, works of Kuan et al. (1992), works of Thomas et al. (1993) are also notable.

The french car manufacturer Peugeot SA has been working for four years on the automatic guidance of motorways vehicles, in full cooperation with the public research laboratory LASMEA. Guidance information is derived from the analysis of video from an on-board camera.

1.1. *The PSA-LASMEA autonomous vehicle*

The foundations of our work were presented in our previous paper (Jurie et al., 1994). Our experimental vehicle is a standard car equipped with an additional device connected to its steering system. This device is an input allowing a velocity control of the steering angle of the wheels. Our research are intended to devise a complete closed-loop control approach based on artificial vision and to validate it both in simulation and in real experiments. The basic idea, which was first given by E. Dickmanns, is based on the use of a 3D geometric road model both for localization and detection purposes.

Concerning the vision aspects, it has been shown (Chapuis et al., 1992) how, assuming the road is piecewise planar, the lateral position and deviation of the car with respect to the road axis can be directly related to the pixel coordinates of the lateral white lines projected onto the image frame. The measurements are a set of pixels detected on the 3 lines in the image, and the slope of these lines at each point detected. Due to the measurement noise, these estimations are not directly usable in the closed-loop control scheme and have to be filtered. An extended Kalman filtering is used to produce an optimal estimation of the state vector.

A road vehicle is a complex dynamic system. Fortunately, it is not necessary to deal with the whole complete model for computing robust and efficient control schemes. In practice, a planar "bicycle" approximation is used, taking into account dynamic effects and a simplified model of tires behaviour. The technique also explicitly takes into account some delays due to the hydraulic steering system of the car. To control such a model, a technique of pole assignment embedded in a discrete-time controller dealing with the delays and the sampling of the control input is used.

These algorithms have been implemented on an experimental car and have been tested on private roads. A great number of experimental results are given in our previous paper. They shows that our system is able to drive automatically the experimental car under motorways conditions.

2. GLOBAL ROAD MARKING ANALYSIS

2.1. *The need for using a global road marking analysis system*

The previous vision module was relying on the following principles : projecting the model in the image makes the use of a very simple white band detector possible. It operates only on a predicted areas of interest. That is why it simple, very fast and well suited for real time applications.

But if many detections are simultaneously erroneous (ie. because of surrounding vehicles) the model can not be correctly reconstructed ; then areas of interest are not projected onto the road marking, and the tracking diverge. That is because informations that are not included into areas of interest are not taken into account. This phenomena, which occures rarely, can give rise to irretrievable errors.

Then a new road mark detection algorithm has been carried out, in order to increase the reliability of the autonomous vehicle. This new module is based on the global organisation of the road marking rather than on local detections. Is has also been designed to be extendable to detection and tracking of more complex objects (like road signs, other vehicles,...). From a grey level video image, it provides a dynamic description of the road markings.

2.2. *Overview*

Due to the lack of place, this presentation is limited to the explanation of general concepts (see (Jurie et al., 1994) for a complete description). The image analysis system consists of two phases : segmentation and interpretation. During the phase of segmentation, grey level images are turned into polygonal regions. Then the interpretation process labels each one of these regions. Labels gives a meaning to regions (e.g. road, white lines, etc.). The interpretation process is based on a dynamic hypotheses network, which is described below.

2.3. *Segmenting the images*

This processing involves several steps : searching edge points, linking the edge points, fitting edges into straight lines, horizontally extending their extremities to the neighbouring lines, producing trapeziums (see Fig. 1).

2.4. *Interpreting regions*

2.4.1. *Information spaces.* The internal representation of the scene (the outside world) is a set of information. An information is made of a representation and one or more interpretation, respectively taking their values in a representation space and in an interpretation space.

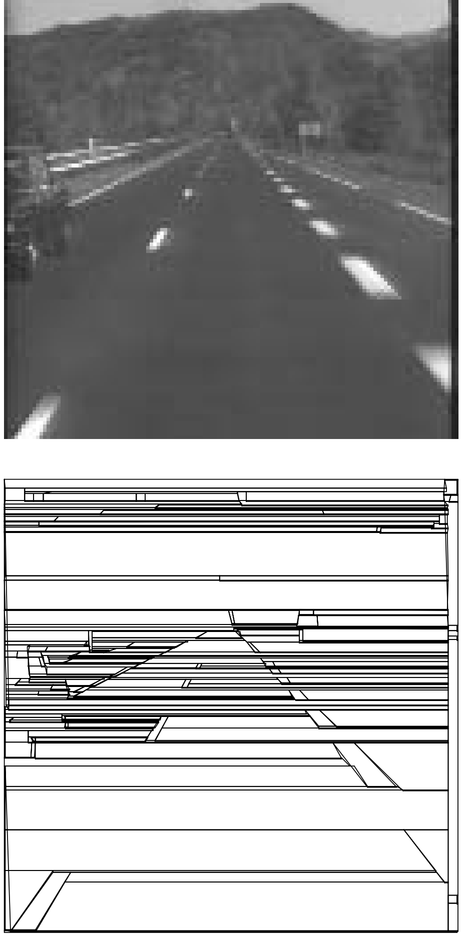


Fig. 1. Image Segmentation

The representation is the material support of information. It is, as example, a string of letters or the result of a measurement. A representation may have many interpretations, like an object name or a list of properties.

In our system, atoms of information are called *hypotheses.* They are constituting a *network of hypotheses.*

An hypothesis is made of :

- a *name* , which is the label of the *hypothesis class* that the hypothesis is belonging to
- a *state vector* : it encodes the coordinates of the hypothesis into a representation space
- a *certainty* : it is a value in $[0, 1]$ representative of the veracity of the hypothesis.

Hypotheses are interacting through the network by four kinds of relationship :

- "*is made of*" is included in order to define hypotheses organization

- ”*in concurrence with*” is provided for representing a relationship between two incompatible hypotheses
- ”*evoke that*” links an hypothesis with an other one suggested by the presence of the first one
- ”*imply that*” is used when an hypothesis implies an other one

2.4.2. *Knowledge representation.* The knowledge is the set of relationship expressing the meaning of the interpretations, according to the representation spaces. Information are always defined in respect with a knowledge.

We represent knowledge under the formalism of *semantic networks*. Relationship between the different categories of knowledge are fuzzy functions.

2.5. *Control*

The scene description is the sum of all the information included into the network. Updating of this information network is realised by propagating interaction effects all over the network.

Interactions are propagated in the top-down direction as well as in the bottom-up one.

There is no strict distinction between high level and low-level processes. There are various levels of abstractions, from image pixels to complex and abstract relations, which are interacting throughout the same network.

The control strategy requires four major components. These four cycles are coordinated by a sequencer whose goal is to control the amount of time spent on each cycle and to respect given ratios. These four cycles (that will be described below) are *the certainty update cycle, the link updating cycle, the hypotheses creation cycle* and *the hypotheses tracking cycle.*

Certainty update cycle. Hypotheses certainties are values in $[0, 1]$ which are to be dynamically adjusted. In order to do that we define a function giving the confidence of a hypothesis according to its place in the network. The confidence is only computed from local properties of the hypothesis and from certainties of linked hypotheses. This function must keep the network consistent (for example when competitive hypotheses are added). It also includes a temporal filtering to limits the risk of oscillations. A certainty evaluated by this function represents the usefulness of hypothesis with respect to the situation : reliability of the information, certainties of other hypotheses supporting it, reliability of the ancestors and hypothesis participation in other groupings.

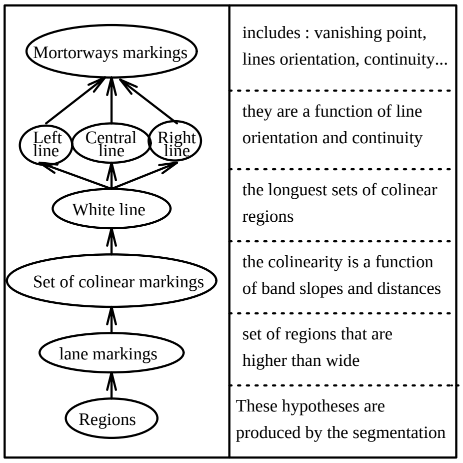
2.5.1. *Link updating cycle.* A hypothesis has a limited number of links. The usefulness of a link is depending on the situation. The less usefull links are removed to give greater place to themost usefull ones. It means that the interactions between information is also dynamically updated.

2.5.2. *Hypothesis creation cycle.* The size of the network is limited. When a hypothesis creation is requested by the creation cycle, the system must determine if this hypothesis is beneficial to the network. This task is performed by evaluating the certainty of the hypothesis to be created. If its certainty is higher than the lowest one of the network, this last hypothesis is deleted and the new one is substituted. The creation cycle is based on a purely data driven mechanism.

2.5.3. *Hypothesis tracking cycle.* The tracking cycle is devoted to take into account evolution of hypotheses constituting a grouping. The change is propagated along the network. It consists in a search for new hypotheses in areas of the information space near the changes, leading to special creations in these areas. In our case, tracking is not an update of hypothesis state vector but a creation of new hypotheses in area of changes.

2.6. *Application to road scene analysis*

We programmed our system to recognize and follow various kind of road markings : white bands, motorway entrances, motorway exits.



includes : vanishing point, lines orientation, continuity...

they are a function of line orientation and continuity

the longuest sets of colinear regions

the colinearity is a function of band slopes and distances

set of regions that are higher than wide

These hypotheses are produced by the segmentation

Fig. 2. Road markings symbolic model.

Instead of being described by a 3D analytic model, the scene is described by the symbolic appearance of objects (visual prototypes) and by the relationship between objects. The informations are among the following ones : motorways markings, central, left, and right lines, white lines, colinear markings, markings. The Fig. 2 shows the ”made-of” relationships of the ”motorway markings” model. The model also includes ”competition” relationships linking ”motorway” hypotheses, meaning that there is only one of these hypotheses that can be real at a given time.

Hypothesis certainty is a function of its likeness to the model and of the certainty of their components. For example, the certainty of a ”motorway markings” will be increased accordingly to the precision of their vanishing point and according to the certainty of the ”line” hypotheses is is made of.

3. REAL TIME IMPLANTATION

The next step is to implement this algorithm on one of our specific architectures WINDIS, characterized by its modularity and its real time abilities.

3.1. Architecture overview

This architecture, developed in collaboration with l'INRIA de Sophia Antipolis-France Martinet et al. (1991), Rives et al. (1993), implements the concept of active windows. An active window is attached to a particular region of interest in the image and is required to extract a desired feature in this region of interest. At each active window, there is a temporal filter able to perform the tracking of the feature along the sequence. Several active windows can be defined (spleeting phase) at the same time in the image, and different processing can be done in each window: low and intermediate level processing. High level processing is ensured from the merging information (merging phase). This aspect is clearly taken into account at the hardware level by using a mixed architecture :

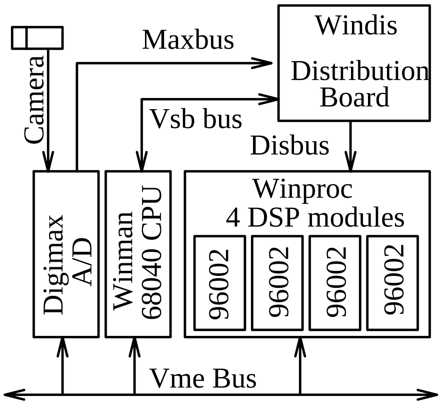


Fig. 3. Windis Architecture

- **Pipeline Architecture** is associated to the low level making efficient processing like convolution possible which is repetitive through the window. In practice, this is done by means of VLSI convoluors which admit up to a 5X5 mask size. This step can be performed in real time with duration depending on the mask size. The input of the algorithm is the active window and the values of the mask parameters corresponding to the desired processing. The output will be constituted by a list of pixels which potentially belong to the geometric features present in the window.
- **MIMD Architecture** has in charge the intermediate level. At this level, we find merging-like algorithms which use the list of pixels of interest provided by the low level stage and merge them to compute a more structured representation of the geometric features present in the window (for example, slope and distance to the origin for a line or coordinates of center and radius for a circle). In our hardware, this part is done by using multi DSP boards.

Windis architecture comprises three basic modules (corresponding to three different VME boards). Depending on the complexity of the application, the system will be built around one or more of these basic modules in a such a way that we will be able to attempt video rate performances. The three basic modules are (see Fig. 3) :

- **WINDIS Window Distributor Subsystem:** This module is used for window extraction, the execution of low level processing and the distribution of active windows toward the Window Processing Subsystem. Window distribution consists in dispatching lists of selected pixels and grey levels through the window bus at the rate of 20 Mhz.
- **WINPROC Window Processing Subsystem:** We have associated one to sixteen DSP 96002 modules with one distributor module. DSP modules are put together on mother boards and execute medium level processing on windows. Window processing modules provide a geometric description of the required primitive in each window. Four DSP modules can be used with each VME mother board. Each modules have 256k32 bits of general memory data and can execute program developped en C at the rate of 33 Mhz.
- **WINMAN Window Manager Subsystem :** The window manager controls distributor and DSP modules, and executes high level processing of application tasks. Moreover, it is used for the tracking of the active windows throughout the sequence. A 68040 based cpu board implements this module.

For each level, we have introduced parallelism allowing us to reach video rate for most of the application tasks. All the modules satisfy VME requirements and are compatible with MAXBUS video bus from Datacube. The management of such a system is a tricky job and requires the use of a real time operating system. For facility reasons, we have chosen to develop all the system level under VxWorks real time operating system.

4. RESULTS AND DISCUSSION

The recognition algorithm described below (segmentation and recognition) has been first implemented on a workstation for purpose of testing. It has been tested on several hundred of real video images. The results shown in Fig. 4 are extracted from one of these tested sequences. This sequence is rather complicated ecause there are two vehicles hidding the signalization, while, in the same time, the experimental car is overtaking the darker vehicle. White lines that have been recognized are displayed on the right the original image. For all the images that have been tested results were reliable and contents enough information for reconstructing the 3D model of the road. Processing time is about 1 second on a HP workstation (serie 700).

Results seem to be reliable. But one have to be very cautious about this subjet. This algorithm will be considered as reliable only after processing many thoushand images. Only the real time implementation will allow to process a such quantity of images.

The real time implementation is in the process of being realized. For this implementation we use the following configuration : one Window distributor, one Winproc subsystem with 4 DSP modules and one Window Manager Subsystem.

The segmentation is implemented through the low level and the intermediate level of the architecture. The processing time is around 240 ms. One note that this phase can be programmed in parallel through the DSP modules, if we split the image into several horizontal parts. By this way, the processing time will be strongly reduced.

The high level algorithm will be implemented on another DSP module. We plan to obtain a processing time lower than 240 ms.

Due to the structure of the global algorithm, we can execute segmentation step and high level step in parallel (through a pipeline approach). We think that it's not an utopic idea to consider a global processing time around 240 ms.

This system can be easily extended to the detection and tracking of various element of the scene, by adding new relations in the semantic network. Hypotheses tracking cycle makes this system particularly well designed for dynamic applications, including autonomous vehicle guidance.

ACKNOWLEDGMENTS

The authors would like to thank R. Chapuis for all his help with the real time implementation, and the French car manufacturer PSA for his financial and technical help.

5. REFERENCES

N.W. Campbell and B.T. Thomas (1993), Navigation of an autonomous road vehicle using lane boundaries markings", in *Proc. of IFAC Int. workshop on Intelligent Autonomous Vehicles*, Southampton, April 1993, pp.499-504.

Chapuis,R., J. Gallice, F. Jurie, J. Alizon, "Real time road mark following", Signal Processing, n. 24, 1991, pp. 331,343.

E.D. Dickmanns, E.D. and B.D. Mysliewetz (1992), Recursive 3-D Road and Relative Ego-State Recognition, in *Transactions on Pattern Analysis and Machine Intelligence*, vol.14, n.2, pp.199-213.

Jurie, F., P. Rives, J. Gallice and J.L. Brame (1994),High Speed Vehicle Guidance Based on Vision, in *Control Engineering Practice*, vol.2 , n.2, 1994.

F. Jurie, J. Gallice, "Interpreting Outdoor Image Sequences by Constraint Propagation in a Hypotheses Network", to be published in *Annals of Mathematics and Artificial Intelligence, Special Issue on Formal Methods in 2-D Shape Analysis*, 1995.

Kanade, T., et al. (1987), 3D Vision for an Autonomous Vehicle, in *Proceeding of International Workshop on Industrial Applications of Machine Vision and Machine Intelligence.*

Kuan, D., G. Phipps and A.C. Hsuech (1992), Autonomous Robotic Vehicle Road Following, in *Transactions on Pattern Analysis and Machine Intelligence*, vol. 10 n.5, pp.648-658.

Martinet P., P. Rives, P. Fickinger, and J.J. Borrelly (1991). Parallel Architecture for Visual Servoing Applications. *Workshop on Computer Architecture for Machine Perception, CAMP'91*, Paris, France.

Rives, P., J.J. Borrelly, J. Gallice, and P. Martinet (1993). A Versatile Parallel Architecture for Visual Servoing Applications. *Workshop on Computer Architecture for Machine Perception, CAMP'93*, News Orleans, USA.

Waxman, A.M., J. Le Moigne and L.S. Davis (1987), A visual navigation system for autonomous land vehicle, in *IEEE Robotics and Automation*, RA-3(2), pp.124-141.

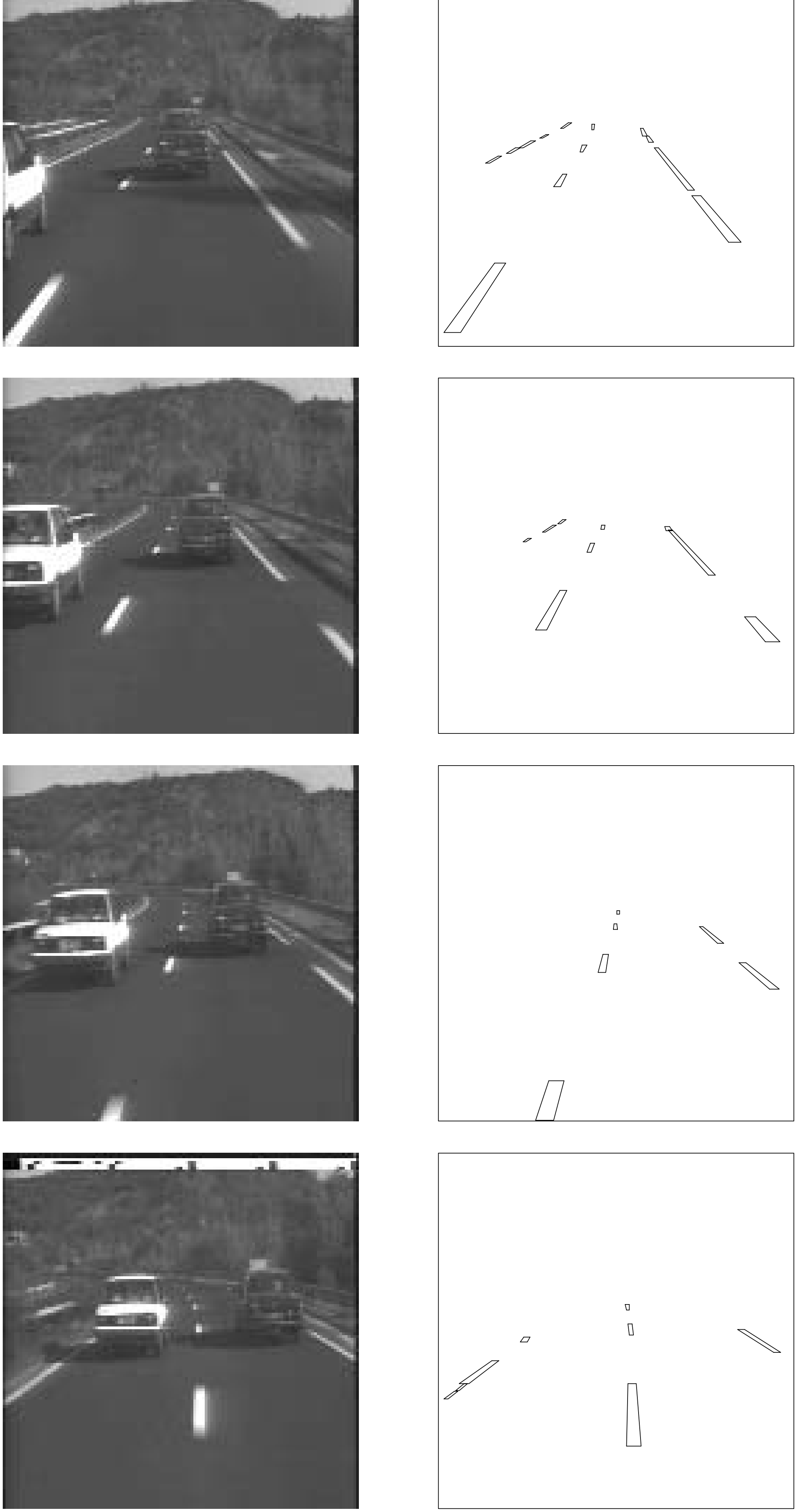


Fig. 4. Results