Machine Vision and Applications (2001) 13: 95–107

Machine Vision and Applications

A model-driven approach for real-time road recognition

Romuald Aufrère, Roland Chapuis, Frédéric Chausse

LASMEA, UMR 6602 du CNRS, Université Blaise Pascal, 63177 Aubière Cedex, France; e-mail: {aufrere,chapuis,chausse}@lasmea.univ-bpclermont.fr, Tel.: +33-4-73407256, Fax: +33-4-73407262

Received: 18 November 2000 / Accepted: 7 May 2001

Abstract. This article describes a method designed to detect and track road edges starting from images provided by an onboard monocular monochromic camera. Its implementation on specific hardware is also presented in the framework of the VELAC project. The method is based on four modules: (1) detection of the road edges in the image by a model-driven algorithm, which uses a statistical model of the lane sides which manages the occlusions or imperfections of the road marking – this model is initialized by an off-line training step; (2) localization of the vehicle in the lane in which it is travelling; (3) tracking to define a new search space of road edges for the next image; and (4) management of the lane numbers to determine the lane in which the vehicle is travelling. The algorithm is implemented in order to validate the method in a real-time context. Results obtained on marked and unmarked road images show the robustness and precision of the method.

Key words: Lane recognition – Driving assistance – Onboard systems – Real-time computer vision – Model-driven image analysis

1 Introduction

Many studies have looked into the development of systems able to assist a driver in his or her driving activity. This has included several major projects, included “RALPH” (Pomerleau 1995), “MOB-LAB” (Broggi 1995) and “VAMORS” (Dickmanns and Mysliwetz 1992). The work presented here focuses on the recognition of road edges, since this aspect represents a major component of such projects.

1.1 Approaches to road detection

Most of the approaches used to recognize marked or unmarked roads are based on the detecting the road edges in the image. Image segmentation is difficult in a motorway context, because of the discontinuous marking and the traffic present. The approaches taken in the recognition of road edges fall into two categories: the image approach and four-dimensional (4-D) approach. The first approach exploits a road model in the image grouping together with detected indices according to an image model of the road edges.

Some approaches are based on a rectilinear model of the road edges (Chen 1997; Tarel et al. 1999). Other methods, such as the minimalist approach of Herman et al. (1997), model the edges as simple parabolas. The classification is made according to a distance criterion using the model deduced from the previous image. Kreucher and Lakshmanan (1999) used a hyperbola which was updated using indices resulting from a discrete-cosine-transform frequency analysis. A Bayesian classification was associated with this frequency analysis. The hyperbolic model of the road edges used the same parameters for both sides of the road. This forced coherence was not explicitly used to drive the detection. Wang et al. (1998) used splines to model the road edges. Pomerleau (1995) developed the so-called RALPH system based on the minimization of curve parameters using a projection of a bird’s-eye view of the road. Broggi (1995) exploited a similar representation associated with a morphological approach in the so-called MOB-LAB demonstrator. Other recent work has used more complex models, such as Guichard and Tarel (1999) who calculated the indices according to the longest coherent curve in the image. In this image-based approach – except in the rectilinear case – a search of the coherence between the road edges in the image is not carried out. This represents a considerable drawback of this kind of approach, since utilizing the coherence makes it possible to detect partially occluded road edges by taking into account the knowledge available about the other side. However, it is not easy to characterize the coherence of the two curves, so this is often performed as a separate process.

The 4-D approach takes into account the kinematic knowledge of the vehicle and its evolution on the road. The coherence of the road edges in the image is thus ensured. The principle is to model the road in 3-D space, as well as the vehicle and its kinematic evolution. The measurements taken from the image are used to update the road model parameters and the vehicle state (localization parameters of the

Correspondence to: R. Aufrère

vehicle in its lane). This approach benefits from the realism of the models used, which can be complex (Dickmanns and Mysliwetz 1992). It predicts in a precise way the position of the primitives for the next image according to a prediction/updating principle. This prediction reduces instabilities due to the updating of a non-linear high-order model, but it does present several drawbacks.

First, an initialization step is necessary. This stage is restricting in the case of road sides tracking losses because the re-initialization time can be significant and the initial conditions may not be reached. Secondly, the 4-D approach is more noise sensitive than an image-based approach due to the approximations associated with the small angles involved, camera calibration errors and vehicle vibrations. Taking these drawbacks into account, we chose to develop an image-based approach driven by the following constraints:

1. It must work on marked and unmarked roads.
2. The road model used must provide an accurate representation of the road, irrespective of its type.
3. The algorithm must be robust to occlusions, marking quality and weather conditions, and must take into account the coherence of the road edges.
4. It must be efficient in terms of computational time (at least 5 images/s on a low-cost architecture).
5. It must be able to collaborate with other modules (e.g. obstacle detection and GPS localization).

The image-based approach implies the use of a pattern recognition technique. The traditional approaches applicable in our case can be decomposed into two categories. The first one is a search in the transformation space. An example is the Hough transform (Maitre 1985; Breuel 1991). This technique is well-adapted overall to problems where the dimension of the parameters vector is low ($\leq$ 6). However, this approach depends on standardization of the lanes, which makes it unsuitable in our case. We have instead chosen to work in a high-order space in order to accurately represent the roadway – this high-order model does not depend on any lane standardization.

The second category involves working in the correspondence space. In this case, each detection is associated with each primitive that forms the object to be recognized. The combinatory reduction is generally performed by pruning the branches of the search tree as rapidly as possible (Grimson 1990). This method must use various heuristics that are aimed at gathering the features that are consistent with the model. Our recognition approach is based on this principle. This approach does not require detecting all the features in the entire image as is the case in accumulation methods.

1.2 Our approach

Our approach detects and tracks the road edges in the image. It is composed of four modules (see Fig. 1): road recognition, vehicle localization, tracking and lane-number management.

The recognition step accurately determines the position of the road edges in the image. The search is performed in the correspondences space. Indeed, the dimension of the search space (20 in our case) prevents almost all other approaches. This step is based on a statistical model of the lane sides. Starting from the data resulting from the recognition step, the localization stage calculates the lateral position and orientation of the vehicle in its lane. The tracking module is then applied in order to limit the search space of the road edges in the next image. The last step identifies the number of the lane (on motorways) in which the vehicle is travelling. This article first describes in more detail the various steps of our method. Our experimental vehicle and its dedicated architecture are then presented, and some typical experimental results are given.

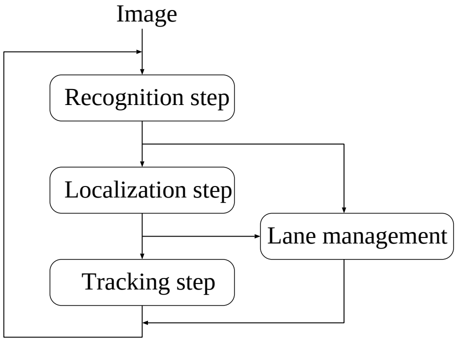

Fig. 1. Organization chart of the algorithm

2 Recognition step

2.1 Principle

This method is based on recursive recognition driven by a probabilistic model of the road edges in the image (Aufrère et al. 2000a). The organization chart of the recognition step is presented in Fig. 2.

2.2 Model of road edges

2.2.1 Model definition

We used several criteria when deciding upon the type of model to use for the road edges: (1) high order, so that the lane edges are modelled accurately; (2) a simple and linear updating step, to limit the processing time and to improve the stability; and (3) weak abstraction in order to clearly understand each recognition step. Taking these criteria into account, our statistical model is composed of $2n$ image parameters (n = 10 in our case). n can be chosen according to the application (e.g. motorways, main roads or country roads). This model is represented by a vector $\boldsymbol{X}_d$ and its covariance matrix $\mathbf{C}_{\boldsymbol{X}_d}$:

$$\boldsymbol{X}_d = \begin{pmatrix} u_{1\mathrm{L}} \\ \cdot \\ u_{n\mathrm{L}} \\ u_{1\mathrm{R}} \\ \cdot \\ u_{n\mathrm{R}} \end{pmatrix}, \mathbf{C}_{\boldsymbol{X}_d} = \begin{pmatrix} \sigma^2_{u_{1\mathrm{L}}} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \sigma^2_{u_{n\mathrm{L}}} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \sigma^2_{u_{1\mathrm{R}}} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \sigma^2_{u_{n\mathrm{R}}} \end{pmatrix}$$

where $(u_{1\mathrm{L}}, \ldots, u_{n\mathrm{L}})$ and $(u_{1\mathrm{R}}, \ldots, u_{n\mathrm{R}})$ represent the horizontal image coordinates of the left and right road edges, respectively, for different image rows v_i ($i \in [1, n]$).

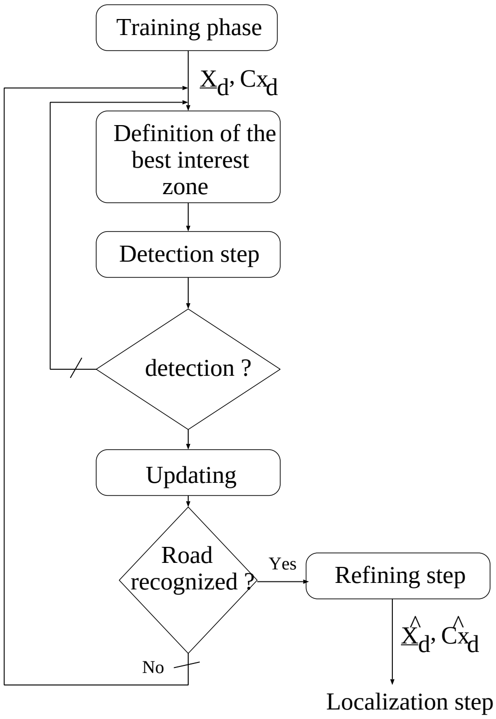

Fig. 2. Organization chart of the recognition step

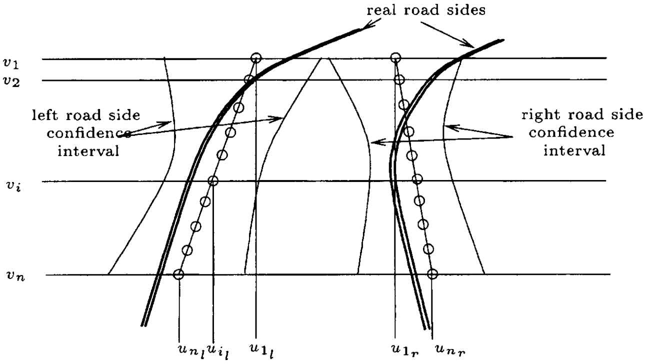

Fig. 3. Model and road-edge confidence intervals resulting from the training phase

$(u_{1\mathrm{L}}, \ldots, u_{n\mathrm{L}})$ and $(u_{1\mathrm{R}}, \ldots, u_{n\mathrm{R}})$ correspond to the image columns (see Fig. 3). Matrix $\mathbf{C}_{\boldsymbol{X}_d}$ defines the possible variations of $u_{i\mathrm{L}}$ and $u_{i\mathrm{L}}$ in the image. The initial value of the model $(\boldsymbol{X}_d, \mathbf{C}_{\boldsymbol{X}_d})$ is obtained during an off-line training phase. This phase ensures that only realistic road configurations are searched for in the image.

2.2.2 Model training

The goal of this procedure is to provide an initial value of the model $(\boldsymbol{X}_d, \mathbf{C}_{\boldsymbol{X}_d})$, in order to limit the search zone of road edges in the image. This is a preliminary calculation and is not entered into the recognition process. In this step, we use a model of the road vehicle unit developed in Chapuis (1991), but other geometrical models could also be used. This model represents a relationship between u coordinates of road edges in the image and v_i ordinates, vehicle localization parameters (x_0, lateral position of vehicle on the roadway; ψ, vehicle steering angle), a camera parameter (α, camera inclination angle) and the road geometry (C_l, lateral curvature of the road; L, road width):

$$(u_{i\mathrm{L}}, u_{i\mathrm{R}})^\mathrm{T} = g(v_i, x_0, C_\mathrm{l}, \psi, L, \alpha)$$

The model is defined by:

$$v_{i\mathrm{R}} = E_u \left(\frac{v_i - E_v \alpha}{E_v z_0} \left(x_0 + \frac{L}{2} \right) - \frac{E_v z_0}{2(v_i - E_v \alpha)} C_\mathrm{l} - \psi \right) \quad (1)$$

$$v_{i\mathrm{L}} = E_u \left(\frac{v_i - E_v \alpha}{E_v z_0} \left(x_0 - \frac{L}{2} \right) - \frac{E_v z_0}{2(v_i - E_v \alpha)} C_\mathrm{l} - \psi \right) \quad (2)$$

where $E_u = f/d_u, E_v = f/d_v$, f is the focal distance of the camera, d_u and d_v are the width and height of a pixel in the image, and z_0 is the height of the camera.

The average vector $\boldsymbol{X}_d$ and its covariance matrix $\mathbf{C}_{\boldsymbol{X}_d}$ are calculated from (1) and (2). $\boldsymbol{X}_d$ is obtained starting from average values of the essential parameters $(x_0, \psi, C_\mathrm{l}, \alpha, L)$ established from initial knowledge of the vehicle, the road and the camera, and of various ordinates v_i in the image. Variations in these parameters according to the required dispersion interval (according to the foreseeable disparity of these parameters on the roadway under consideration), will led to a set of realistic roads configurations in the image. From these various configurations, it is easy to extract the covariance matrix $\mathbf{C}_{\boldsymbol{X}_d}$ by using the Jacobian J_g of function g:

$$\mathbf{C}_{\boldsymbol{X}_d} = J_g \mathbf{C}_p {J_g}^\mathrm{T}$$

where

$$\mathbf{C}_p = \begin{pmatrix} \sigma_{x_0}^2 & & & & \\ & \sigma_\psi^2 & & 0 & \\ & & \sigma_{C_\mathrm{l}}^2 & & \\ & 0 & & \sigma_\alpha^2 & \\ & & & & \sigma_L^2 \end{pmatrix}$$

with $\sigma_{x_0}, \ldots, \sigma_L$ representing the authorized variations around the localization parameters. This step is simple and sufficient to initialize the recognition process. Nevertheless, the approach can be easily extended to a much more complex model (e.g. taking into account rolling or a vertical curve). Moreover, $\boldsymbol{X}_d$ and $\mathbf{C}_{\boldsymbol{X}_d}$ could be learned on-line in order to adapt the recognition in an optimal way to the current road. Figure 4 shows the results of the training step. For each edge of the lane, a confidence interval centred on $\boldsymbol{X}_d$ is represented (white lines) whose width (of one standard deviation) is deduced from the $\mathbf{C}_{\boldsymbol{X}_d}$ matrix. v_i ordinates are also presented (black lines), which represent the height of the search zones in the recognition procedure. These heights are fixed experimentally.

The next phase of our recognition process is investigating the road edges in the image using $\boldsymbol{X}_d$ and $\mathbf{C}_{\boldsymbol{X}_d}$. In this procedure, the model vector $\boldsymbol{X}_d$ and its covariance matrix $\mathbf{C}_{\boldsymbol{X}_d}$ for an analysis depth p will be denoted $\boldsymbol{X}_d(p)$ and $\mathbf{C}_{\boldsymbol{X}_d}(p)$.

2.3 Research procedure

Our goal is to find the optimal value of $\boldsymbol{X}_d$ representing the road edges in the image. After the training phase, the zone

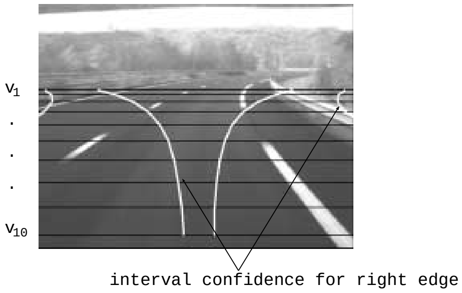

Fig. 4. Model resulting from the training step

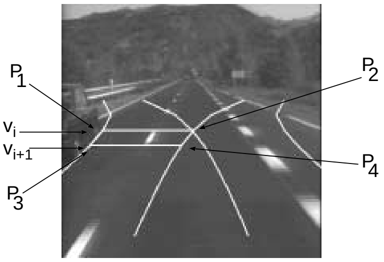

Fig. 5. The initial model and the first zone of interest

of greatest interest is located in the image. In this zone, a segment is detected for an analysis depth $p = 0$. The procedure is iterative: the model is updated, a new optimal zone is located and the algorithm works now at a depth $p+1$. The algorithm considers that the road edges have been found in the image if a certain criterion is reached. A refining stage improves the search result.

2.3.1 Interest zones

The zone of greatest interest is a horizontal part of the confidence interval deduced from $\mathbf{C}_{\boldsymbol{X}_d}$. This interval I is defined using the smallest variance σ_i of $\mathbf{C}_{\boldsymbol{X}_d}$, for one standard deviation, by the equation

$$I = [u_i - \sigma_i, u_i + \sigma_i]$$

The interest zone is then defined by a trapezoid of nodes P_1–P_4 (see Fig. 5): $P_1(u_i - \sigma_i, v_i), P_2(u_i + \sigma_i, v_i), P_3(u_{i+1} - \sigma_{i+1}, v_{i+1})$ and $P_4(u_{i+1} + \sigma_{i+1}, v_{i+1})$, where v_i is the ordinate of the image row corresponding to the horizontal ordinate u_i.

Figure 5 shows an image with the initial model (we in fact represent the confidence intervals deduced from $\mathbf{C}_{\boldsymbol{X}_d}(0)$ for both right and left sides of the lane) resulting from the training phase as well as the zone of greatest interest. Here this zone is located on the left edge. For an iteration depth p, the number of zones to be tested is $(2n - 2 - p)$; $(18 - p$ zones in our case).

2.3.2 Detection

This phase consists of detecting the road edges in the interest zone defined previously. For each row inside the interest zone, the points of maximum horizontal gradient are located. These points are fitted to a straight line segment by a median least-squares method. This detection will be made in a quasi-optimal way for the following reasons:

1. The search zone is limited in size (to improve the computational time and the signal-on-noise ratio).
2. A confidence interval on the segment slope is deduced from the $\mathbf{C}_{\boldsymbol{X}_d}$ matrix. This limits the search time and the possibility of false detections.

This detector in essence tries to find a segment consistent with the current model. This technique is much more accurate than a classical whole-image segmentation. If no segment is detected in the interest zone, the detection is attempted in a new area defined by the next smallest variance of the covariance matrix, and so on. If no segment is detected in all of the $18-p$ zones to be tested at the p iteration depth, the algorithm leaves this branch, returns to depth $p-1$ and re-iterates the process on this new branch.

At each analysis depth p the detection provides two points (intersections between the detected segment and the horizontal border of the search zone) which correspond to a measurement $\boldsymbol{y}(p) = (\widehat{u}_i(p), \widehat{u}_{i+1}(p))^{\mathrm{T}}$ updating the model $(\boldsymbol{X}_d(p-1), \mathbf{C}_{\boldsymbol{X}_d}(p-1))$ for the current depth p.

2.3.3 Updating

This phase consists of calculating a new vector $\boldsymbol{X}_d(p)$ and a new covariance matrix $\mathbf{C}_{\boldsymbol{X}_d}(p)$ from the observation $\boldsymbol{y}(p)$ and preceding state $\boldsymbol{X}_d(p-1)$ and $\mathbf{C}_{\boldsymbol{X}_d}(p-1)$ in the following way:

$$\begin{cases} \boldsymbol{X}_d(p) = \boldsymbol{X}_d(p-1) + \mathbf{K}_d[\boldsymbol{y} - \boldsymbol{x}(p-1)] \\ \mathbf{C}_{\boldsymbol{X}_d}(p) = \mathbf{C}_{\boldsymbol{X}_d}(p-1) - \mathbf{K}_d\mathbf{H}_d\mathbf{C}_{\boldsymbol{X}_d}(p-1) \end{cases}$$

where $\mathbf{K}_d = \mathbf{C}_{\boldsymbol{X}_d}(p-1)\mathbf{H}_d^{\mathrm{T}}\left[\mathbf{H}_d\mathbf{C}_{\boldsymbol{X}_d}(p-1)\mathbf{H}_d^{\mathrm{T}} + \mathbf{R}\right]^{-1}$, $\boldsymbol{x} = (u_i, u_{i+1})^{\mathrm{T}}$, $\mathbf{H}_d$ is such that $\boldsymbol{x} = \mathbf{H}_d\boldsymbol{X}_d$, and $\mathbf{R}$ is the covariance matrix of the detection error $\boldsymbol{w}_d$ (such that $\boldsymbol{y} = (\widehat{u}_i, \widehat{u}_{i+1})^{\mathrm{T}} = \boldsymbol{x} + \boldsymbol{w}_d$) a priori fixed at 5 pixels (here $\mathbf{R} = 5^2\mathbf{I}$, where $\mathbf{I}$ is the 2×2 identity matrix). This update is in fact achieved by a degenerated Kalman filter for which no state-vector evolution is needed. For the next iteration p, this update defines a new search space with a smaller size.

2.3.4 Algorithm evolution

The search process (choice of the interest zone, detection and update) is recursively re-iterated at each new iteration and stops when a criterion Q is reached. The road is then assumed to be found. This criterion is defined as follows:

$$Q = \left\{ P_{\mathrm{L}} = \frac{N_{\mathrm{L}}}{\Delta_v} > 10\% \text{ AND } P_{\mathrm{R}} = \frac{N_{\mathrm{R}}}{\Delta_v} > 10\% \right. \\ \left. \text{AND } P_{\mathrm{t}} = \frac{N_{\mathrm{L}} + N_{\mathrm{R}}}{2\Delta_v} > 25\% \right\} \tag{3}$$

where N_{L} and N_{R} are the number of detected points on the left and right sides, respectively, and Δ_v (about 300 pixels) represents the height of the analysis zone $(v_n - v_1)$. This criterion means that it is necessary to detect at least 10% of each road edge and 25% of the lane sides to consider the road as found. The thresholds have been fixed according to

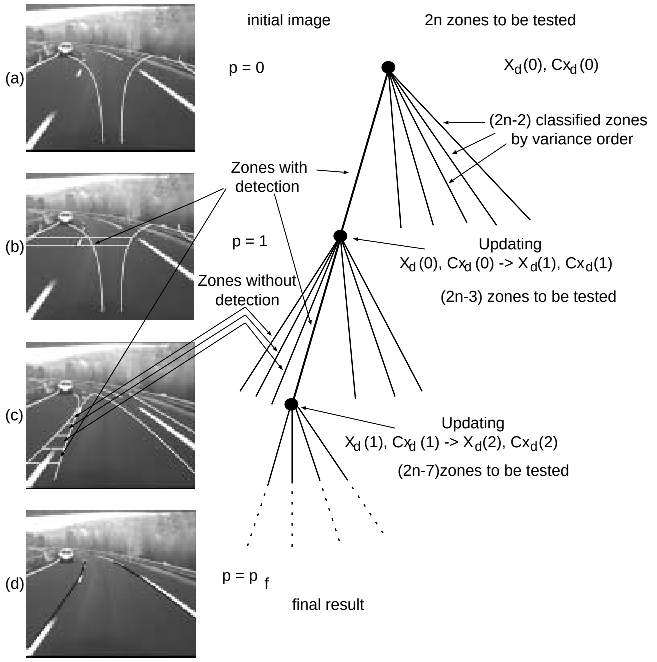

Fig. 6. Algorithm evolution on an image. The outcome of the following steps are presented: **a** training phase, **b** the first interest zone, **c** the next interest zones, **d** the final result

experimental considerations in order to provide, in all the cases, a correct result when it is reached.

Figure 6 presents an example of the algorithm evolution on an image. The search zone of the road edges in the image, resulting from the training phase, is shown in Fig. 6a. Figure 6b shows the first interest zone deduced from $\mathbf{C}_{\boldsymbol{X}_d}(0)$ and the detection made therein. Thanks to this detection, $\boldsymbol{X}_d(0)$ and $\mathbf{C}_{\boldsymbol{X}_d}(0)$ will be updated. Figure 6c demonstrates the significant reduction of the search interval at the $p = 1$ analysis depth, where the first, second and third zones defined by $\mathbf{C}_{\boldsymbol{X}_d}(1)$ do not provide any detection. In this case a detection could be carried out in the next best zone, and thus $\boldsymbol{X}_d(1)$ and $\mathbf{C}_{\boldsymbol{X}_d}(1)$ can be updated. The process is repeated until the criterion Q is reached, which may occur without a complete analysis of the confidence interval. In order to optimize the precision of the algorithm, the process restarts in the non-tested zones of the same branch. The final result (Fig. 6d) of this step corresponds to the vector $\boldsymbol{X}_d$ resulting from the greatest probability P_t (see Eq. 3). Actually, this process finds the first branch of the search tree that matches criterion Q. It is not supposed to be the best solution. However, we tested the process on the whole search tree looking exhaustively for the best solution, and in each case the solution found corresponded to the best one. If Q is not reached, the algorithm considers that the road is not found in the image but that the best solution has still been obtained. Thereafter, the best estimation of $\boldsymbol{X}_d$ (resulting from the recognition step) for image k is denoted by $\widehat{\boldsymbol{X}}_d(k)$.

One major advantage of our algorithm is its ability to use a backtracking search, as illustrated in the example shown in Fig. 7. Starting with the right-hand side of the lane (Fig. b1), the algorithm detects the wrong white line. It continues the model update/detection sequence until all the interest zones in the right-hand side have been processed. Then it begins with the left-hand side (Fig. b3), finds a detection in the first interest zone and tries the next ones in which it cannot find anything (Fig. a4). Then it goes back and restarts the search process with another initial zone (Fig. b4) until a solution is found (Fig. c6). It can be seen that in the case of backtracking, the algorithm does not use any detection result that comes from a wrong attempt.

3 Localization and tracking steps

The road-edges detection module described above is able to recognize the road in an image; our goal is now to compute the vehicle location in its lane using the result of the recognition process $\widehat{\boldsymbol{X}}_d(k)$ (Aufrère et al. 2000b).

3.1 Localization

The image approach we used in the recognition step is not based on 3-D constraints of the road/vehicle unit, but the localization stage does require such modelling. Our localization procedure uses an approach closely dependent on our detection method and the training model (see Sect. 2.2.2). We define a more general vector $\boldsymbol{X} = (\boldsymbol{X}_d, \boldsymbol{X}_l)^\mathrm{T}$, which contains vector $\boldsymbol{X}_d$ and also the state vector $\boldsymbol{X}_l = (x_0, \psi, C_\mathrm{l}, \alpha, L)^\mathrm{T}$ containing the parameters used for the algorithm initialization. The initial value of $\boldsymbol{X}(0)$ and its covariance matrix $\mathbf{C}_X(0)$ are easily obtained using an identical training phase (Sect. 2.2.2).

As in the training stage, a much more complex model can also be used here. The recognition process provides, for an image k, an observation vector $\widehat{\boldsymbol{X}_d}(k)$ and its covariance matrix $\widehat{\mathbf{C}}_{\boldsymbol{X}_d}(k)$. The update of vector $\boldsymbol{X}$ and its covariance matrix $\mathbf{C}_X$, for an image k, is carried out starting from this observation and the initial model $(X(0), C_X(0))$ in the following way:

$$\begin{cases} \boldsymbol{X}(k) = \boldsymbol{X}(0) + \mathbf{K}[\widehat{\boldsymbol{X}}_d(k) - \mathbf{H}\boldsymbol{X}(0)] \\ \mathbf{C}_X(k) = \mathbf{C}_X(0) - \mathbf{KHC}_X(0) \end{cases}$$

where $\mathbf{K} = \mathbf{C}_X(0)\mathbf{H}^\mathrm{T}\left[\mathbf{HC}_X(0)\mathbf{H}^\mathrm{T} + \mathbf{C}_{\boldsymbol{X}_d}(k)\right]^{-1}$, $\mathbf{H}$ is such that $\widehat{\boldsymbol{X}}_d(k) = \mathbf{H}\boldsymbol{X}(k) + \boldsymbol{w}$, $\widehat{\boldsymbol{X}}_d$ is the vector resulting from the road detection process and $\mathbf{C}_{\boldsymbol{X}_d} = E[\boldsymbol{w}\boldsymbol{w}^\mathrm{T}]$ is the covariance matrix of $\widehat{\boldsymbol{X}}_d$.

From $\boldsymbol{X}(k)$, we have now an estimate vector $\boldsymbol{X}_l(k)$ for image k. This vector $\boldsymbol{X}_l(k)$ characterizes the position x_0 and orientation ψ of the vehicle in its lane, the camera inclination angle α and several parameters of the roadway (width L and lateral curvature C_l). Moreover, starting from the tracking phase it is possible to estimate the vehicle position in the next image in order to refine the initialization model for the road recognition in the next image.

3.2 Tracking

In this module, our goal is to provide a smaller confidence interval than the one provided by the initial training stage in when the recognition module looks for road edges. The problem consists of calculating $\boldsymbol{X}(k + 1)$ and $\mathbf{C}_X(k + 1)$

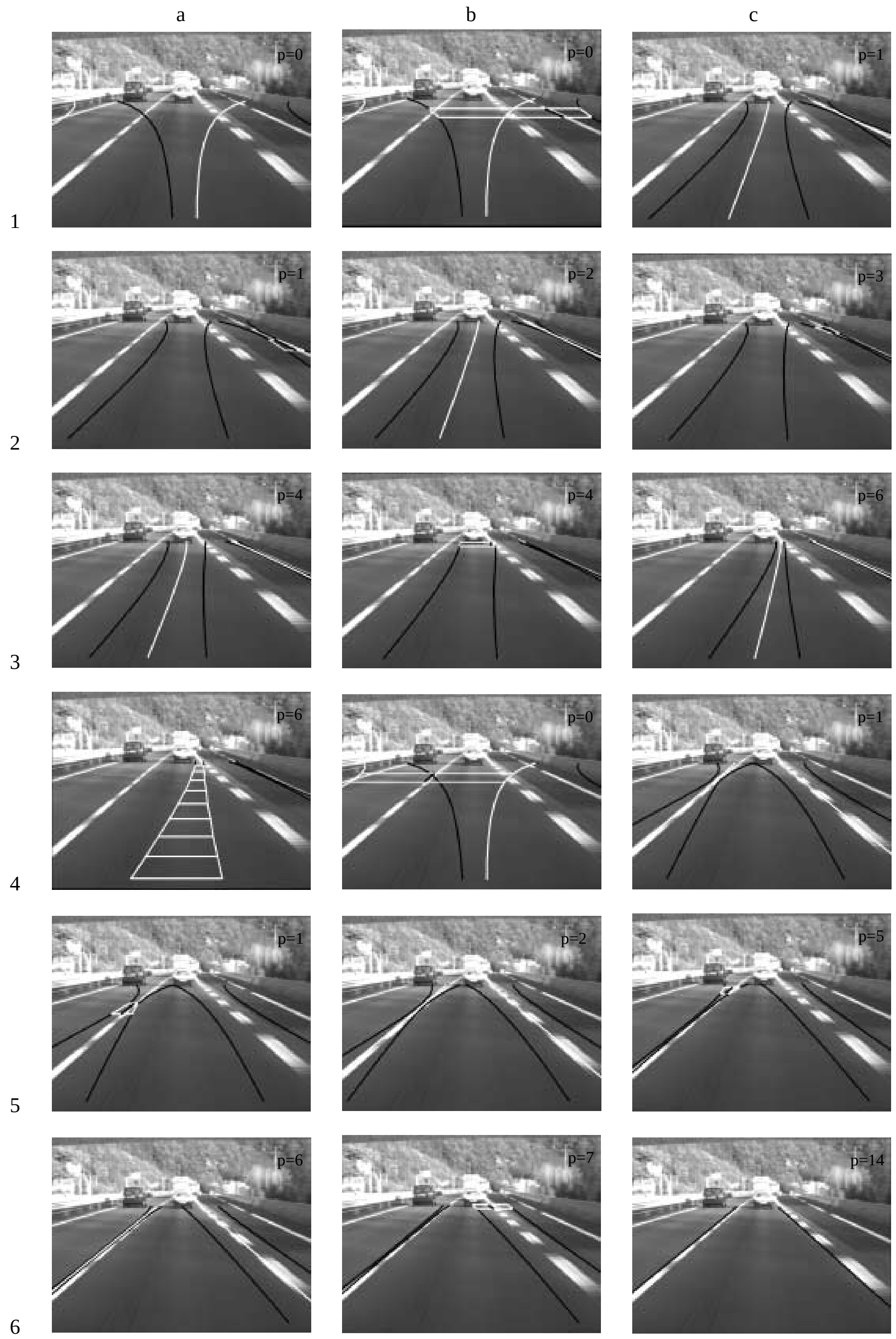

Fig. 7. Illustration of the backtracking capability of the algorithm. The value of p is given in each panel.

from knowledge of image k – $\boldsymbol{X}(k)$ and $\mathbf{C}_X(k)$ – and from the displacement of the car between two successive images. To achieve this we evolve vector $\boldsymbol{X}_l(k) = (x_0, \psi, \alpha, C_1, L)^{\mathrm{T}}$ deduced from $\boldsymbol{X}(k)$ for image $k + 1$ by taking into account the evolution errors:

$$\begin{cases} \boldsymbol{X}_l(k+1) = \mathbf{M}\boldsymbol{X}_l(k) + \boldsymbol{w}_t \\ \mathbf{C}_{X_l}(k+1) = \mathbf{M}\mathbf{C}_{X_l}(k)\mathbf{M}^{\mathrm{T}} + \mathbf{Q} \end{cases}$$

where $\mathbf{M}$ is the evolution matrix

$$\mathbf{M} = \begin{pmatrix} 1 & \Delta_Y & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

with Δ_Y being the displacement between two images, and $\mathbf{Q} = E[\boldsymbol{w}_t \boldsymbol{w}_t^{\mathrm{T}}]$ represents the errors in the evolution matrix (experimental values).

Thereafter, to define the optimized initial image area in the next image, the model ($\boldsymbol{X}$, $\mathbf{C}_X$) resulting from the train-

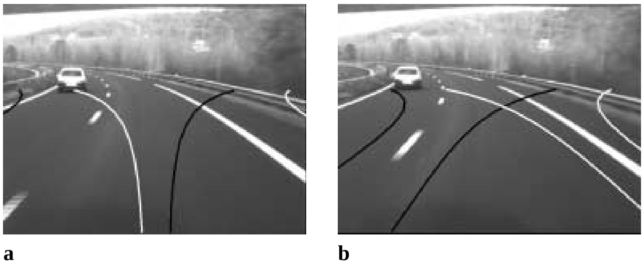

Fig. 8. Initial search zone (*black lines* for the left edge and *white lines* for the right edge): **a** without tracking, **b** with tracking

ing stage is updated using $\boldsymbol{X}_l$ and $\mathbf{C}_{X_l}$ as previously computed:

$$\begin{cases} \boldsymbol{X}(k+1) \; = \; \boldsymbol{X}(0) + \mathbf{K}_t[\boldsymbol{X}_l(k+1) - \boldsymbol{X}_l(0)] \\ \mathbf{C}_X(k+1) \; = \; \mathbf{C}_X(0) - \mathbf{K}_t\mathbf{H}_t\mathbf{C}_X(0) \end{cases}$$

where $\mathbf{K}_t = \mathbf{C}_X(0)\mathbf{H}_t^{\mathrm{T}}\left[\mathbf{H}_t\mathbf{C}_X(0)\mathbf{H}_t^{\mathrm{T}} + \mathbf{C}_{X_l}(k+1)\right]^{-1}$ and $\mathbf{H}_t$ is such that $\boldsymbol{X}_l = \mathbf{H}_t\boldsymbol{X}$.

Thanks to this update, the vector $\boldsymbol{X}_d(k{+}1)$ and its covariance matrix $\mathbf{C}_{\boldsymbol{X}_d}(k{+}1)$, included in $\boldsymbol{X}(k{+}1)$ and $\mathbf{C}_X(k{+}1)$, define a new confidence interval for the next image. Figure 8a shows the initial search zone of the lines in the image resulting from the training phase, and Fig. 8b presents the search zone after a tracking step – the area of the search space is greatly reduced and so the computation time of the recognition step is lower. This method integrates in a tidy way the 4-D approach mentioned in Sect. 1.1.

To manage changing of the lane, we update the model ($\boldsymbol{X}$, $\mathbf{C}_X$) by defining the new search zone with a modified value x_0 according to the configuration given:

$$\begin{cases} \text{if} \quad x_0(k) > L \;\Rightarrow\; x_0(k+1) = x_0(k) - L \\ \text{if} \quad x_0(k) < 0 \;\Rightarrow\; x_0(k+1) = x_0(k) + L \end{cases}$$

4 Lane management

In order to complete the driving assistance system on motorways (consisting of multiple lanes), the algorithm manages the number of lanes on the road and the lane number in which the vehicle is travelling. To achieve this, $2n$ parameters are added to vector $\boldsymbol{X}_d$ in order to detect not only the lane sides but also the two lateral lines beside the circulation lane. This new model (with $4n$ image parameters) is obtained by the training phase presented in Sect. 2.2.2. From the result of the recognition step, realized on one lane (1 lane = $2n$ parameters), an updating of the complete model ($4n$ parameters) limits the search space of lateral lines on the same image. Figure 9 shows the search space of lateral lines after a first recognition step on one lane.

Thereafter, the recognition process is restarted in the $2n-2$ zones corresponding to the lateral lines ($n-1$ for the left lateral white line plus $n-1$ for the right lateral white line). A lateral line is considered to be found if the associated probability ($\frac{N_{\mathrm{LL}}}{\Delta_v}$ or $\frac{N_{\mathrm{RL}}}{\Delta_v}$) is greater than 10%, where N_{LL} and N_{RL} represent the number of detected points on the left and right lateral white lines, respectively. The detection of a line updates one array of probability defining the presence of the lateral lanes and another defining the identification of the lane in which the vehicle is travelling. The latter is updated by taking into account the lateral position of the vehicle, as estimated in the previous step. Thereafter, if one or two lateral lines are considered to be found, the recognition step is started – in the next image – on two or three lanes ($3n$ or $4n$ image parameters). The table of probabilities on the presence of the lateral lines are re-initialized every 80 images in order to take into account any configuration change of the motorway (e.g. a change from 3 lanes to 2 lanes).

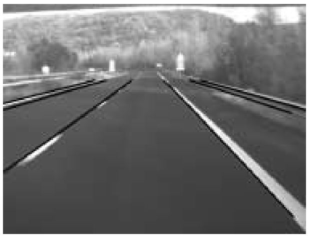

Fig. 9. Search zone of the lateral lines

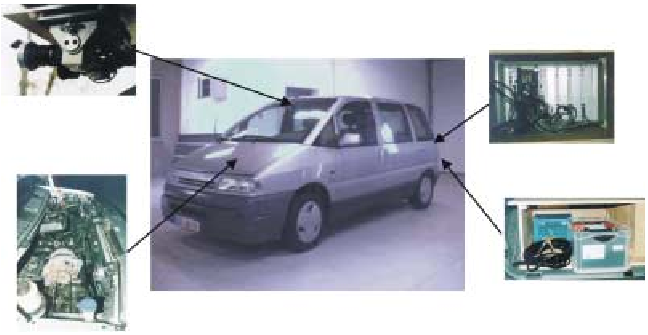

Fig. 10. The VELAC vehicle

5 Implementation

In order to validate the approach in a real-time context, the algorithm has been implemented on an architecture dedicated to vision applications, including an experimental vehicle (VELAC). A specific architecture was required, and is described in more detail in Marmoiton (2000). Our vehicle is completely autonomous from sensors to computer. Figure 10 shows the VELAC vehicle and its various components.

In VELAC the main power supply is provided by an alternator coupled to two 12-V batteries and a 24-to-220 V inverter. Our application uses a simple monochromic camera. This camera is installed beside the interior rear-view mirror. Other sensors (e.g. radar, GPS and gyrometer) and actuators will be embedded in our vehicle in order to achieve various applications (e.g. adaptative cruise control, and steering and accelerator controls). The computer consists of a virtual machine environment (VME)-type architecture which includes: a vision machine, a high-level module and a controller area network (CAN) bus interface. The vision machine is based on a multiple-instruction multiple-data module with distributed memory. It includes four DEC-Alpha-type processors connected to 32 MB of random-access memory. Each one of these processors is connected to a communications processor (a T9000 transputer) to ensure rapid data exchange. The high-level module receives information from

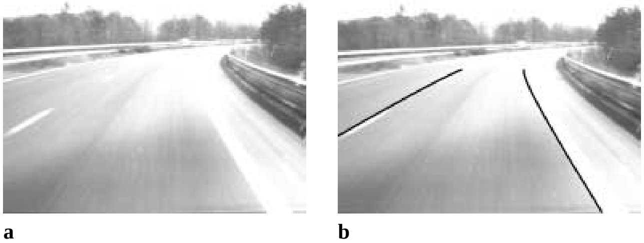

Fig. 11. Images that provided the weakest probability P_t

Table 1. Parameter values of the model used in the training stage (μ, mean; σ, standard deviation)

	$\alpha(°)$	$\psi(°)$	$C_\mathrm{l}(\mathrm{m}^{-1})$	L(m)	x_0(m)
μ	5	0	0	3.5	1.75
σ	0.5	6	0.005	0.5	1.75

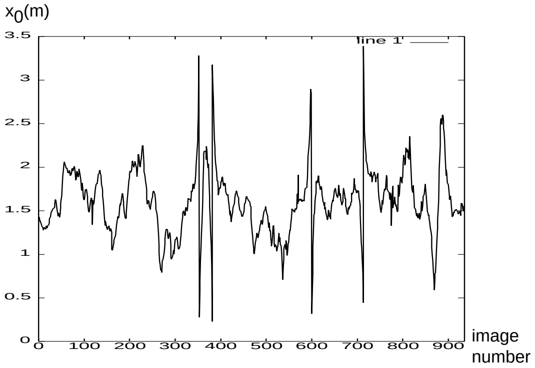

Fig. 12. Results of the x_0 evolution parameter

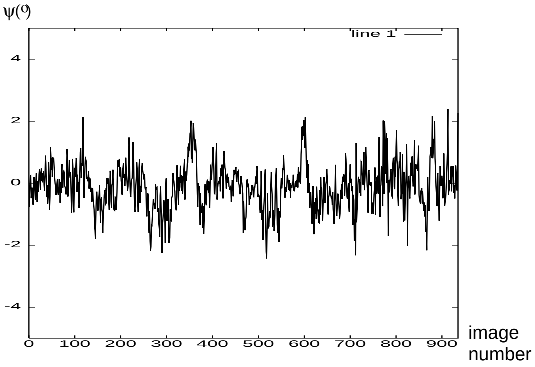

Fig. 13. Results of the ψ evolution parameter

the vision module and sensors. This module can thus calculate the commands for the actuators and inform the operator via a human–machine interface. The CAN bus interface operates the communication between the VME architecture, actuators and sensors. Our algorithm has been ported to run on a single DEC Alpha processor, and other processors can be used for other perception modules.

6 Results

The method has been tested on several kinds of road (marked and unmarked), using images comprising 640 × 480 or 512 × 512 pixels. The model vector for the recognition step is composed of 20 image parameters and corresponds to 18 analysis zones in the image (9 for each of the left and right edges). The camera used had a focal distance $f = 12\,\mathrm{mm}$ and was placed at a height $z_0 = 1.5\,\mathrm{m}$. The values used to calculate the initial model in the training stage (Sect. 2.2.2) are listed in Table 1.

6.1 Results on a workstation

This section presents the results obtained on a workstation and when using the implementation on the dedicated architecture. Figure 14 presents recognition results in various cases on a motorway. In each row, the first and third images represent the road original image and the second and fourth present the final result of the detection algorithm. Figures 15 and 16 present results obtained on urban and unmarked roads, respectively. The representation is similar to that in Fig. 14. The low-level detector detects the maximum and minimum horizontal gradients (knowing that in a given image, the sign of the gradient is constant for each edge). Here the results are quite satisfactory despite shadows (Figs. 14b2, 15a4 and 16c4), occlusions (Fig. 14d4 and 14e2) and lane changing (Fig. 14b4). Thanks to the coherence between the road edges, the algorithm copes with the presence of motorway exits (Fig. 14a4, 14c2 and 14c4). For the results in Figs. 14f2, 14f4, 15b2, 15e4, 15f2 and 15f4 the road is not considered as found because criterion Q is not reached (due to insufficient information). Nevertheless, the results show the best model provided by the algorithm.

In a sequence of 1000 images, the algorithm considers that the road is found (i.e. criterion Q is reached) in 95% of cases. In the other cases, it still provides a very acceptable result. Indeed, the image that provided the weakest criterion is shown in Fig. 11. The total probability P_t is 6%, but the result seems nevertheless very satisfactory.

The proportion of images (5%) where the algorithm considers the road as not found is important because criterion Q was fixed in order to provide, in all the cases, a correct result (see Eq. 3). Figures 12 and 13 present the evolution of parameters x_0 and ψ over a sequence of 1000 images, equivalent to travelling 6 km when taking 5 images/s. The variations in parameter x_0 are due to bends in the road and lane changes. The variations in ψ ($-2° < \psi < 2°$) also correspond to these situations, but the values of this parameter are much more disturbed because of the rapid variations in this angle.

Figure 17 presents results relating to our lane-number management module. These results represent the lane number of the road as well as the number of the lane in which the vehicle is travelling. In Fig. 17b4, the algorithm detects three lanes, and Figs. 17a2 and 17b2 represent cases with occlusions. The computation time of the detection algorithm (without the tracking step) varies from 70 ms to 200 ms on an HP J282 workstation (18–1000 iterations). In the tracking phase – with a small search zone – the computational time decreases, varying from 40 ms to 120 ms (18–500 iterations). The computation time of the algorithm is dependent

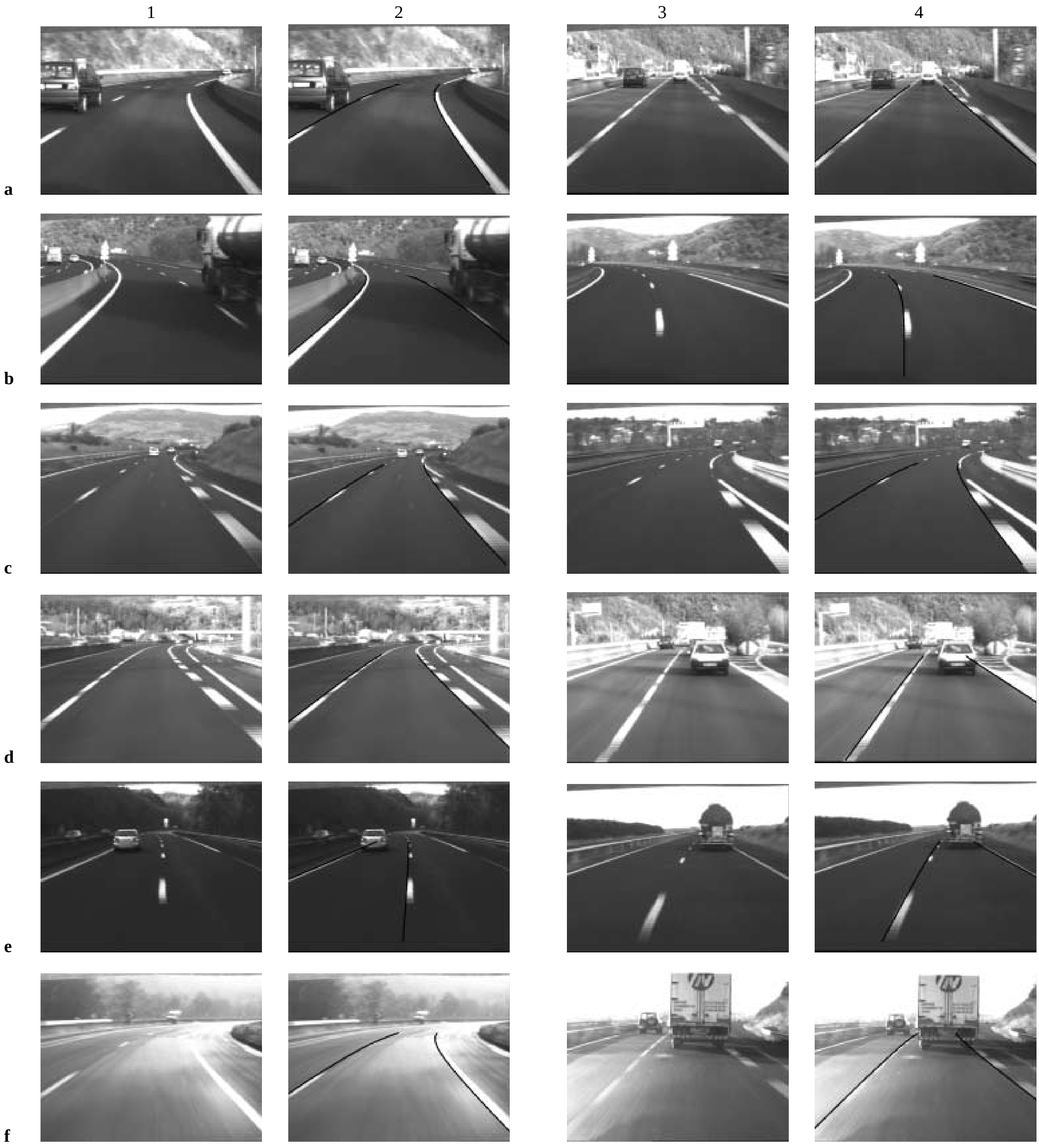

Fig. 14. Algorithm results on a motorway

on the number of iterations, and especially on the quality of the first segment detection.

6.2 Results on our vision machine

The code is sequential and has been implemented on a single T9000/DEC Alpha calculation module as described above. The advantage of our algorithm is the low quantity of image data needed for road detection: for each iteration only one small zone of interest in the image has to be transferred. The computation time of the method is not deterministic and the on-board application is required to process approximately 5 images/s. These two factors are antagonistic, and it is necessary to minimize the total calculation time.

A statistical study of the temporal behaviour of the algorithm has been performed. The tested sequence corresponds to travelling a distance of 12 km on a motorway with several sharp bends and road-edge occlusions, under lane-changing and poor lighting conditions. Among the 3000 images tested, the algorithm succeeds in finding the road (i.e. criterion Q is reached) in 2834 cases. The process is iterated 18–53 times

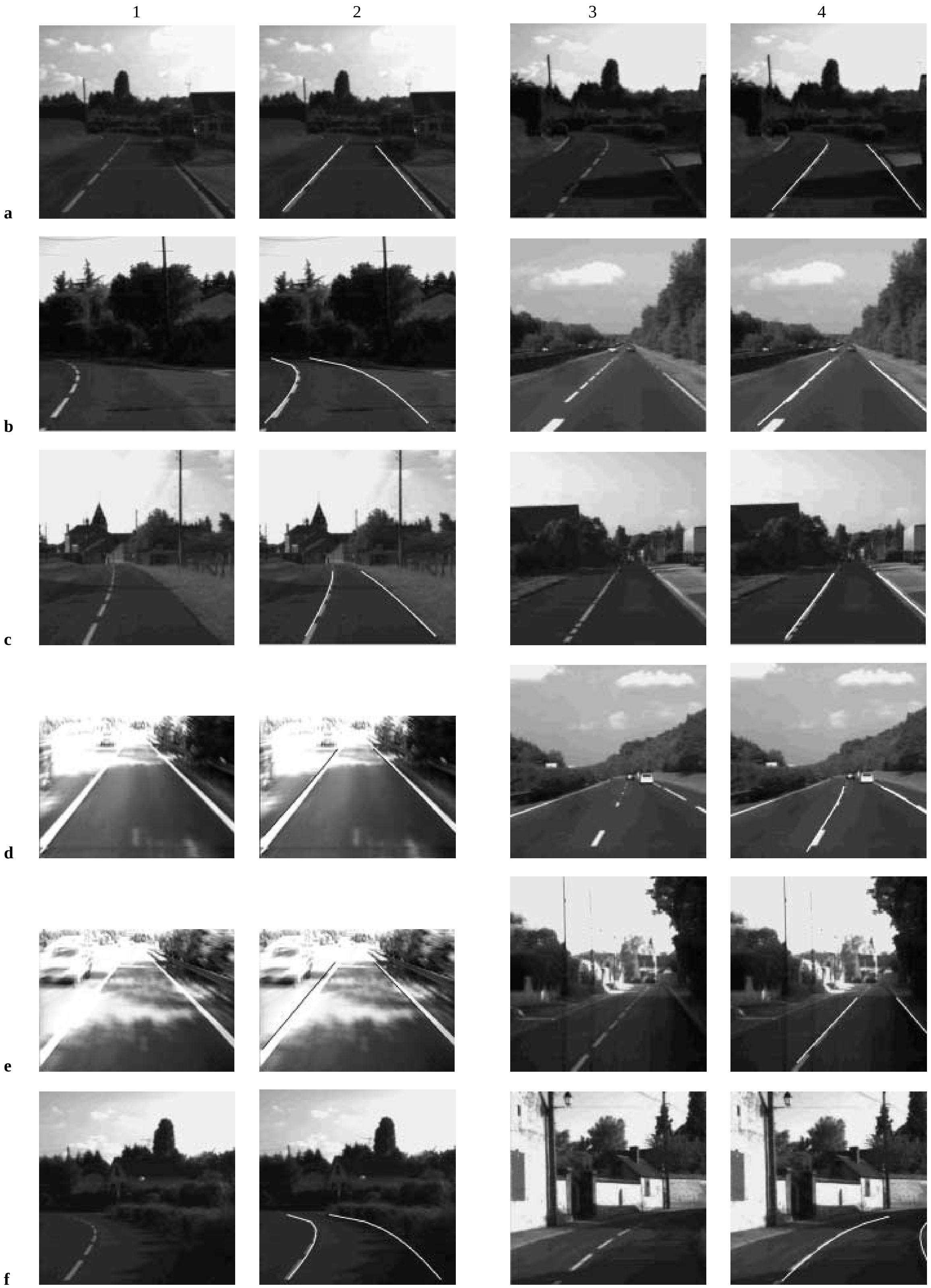

Fig. 15. Algorithm results on urban roads

per image, which corresponds to a computation time of 60–238 ms. The higher figure is rare, so that the processing rate of 5 frames/s is usually obtained.

The minimum and maximum times required for the algorithm to find the road are given in Table 2. The computation times for each step of the algorithm are also given. The time

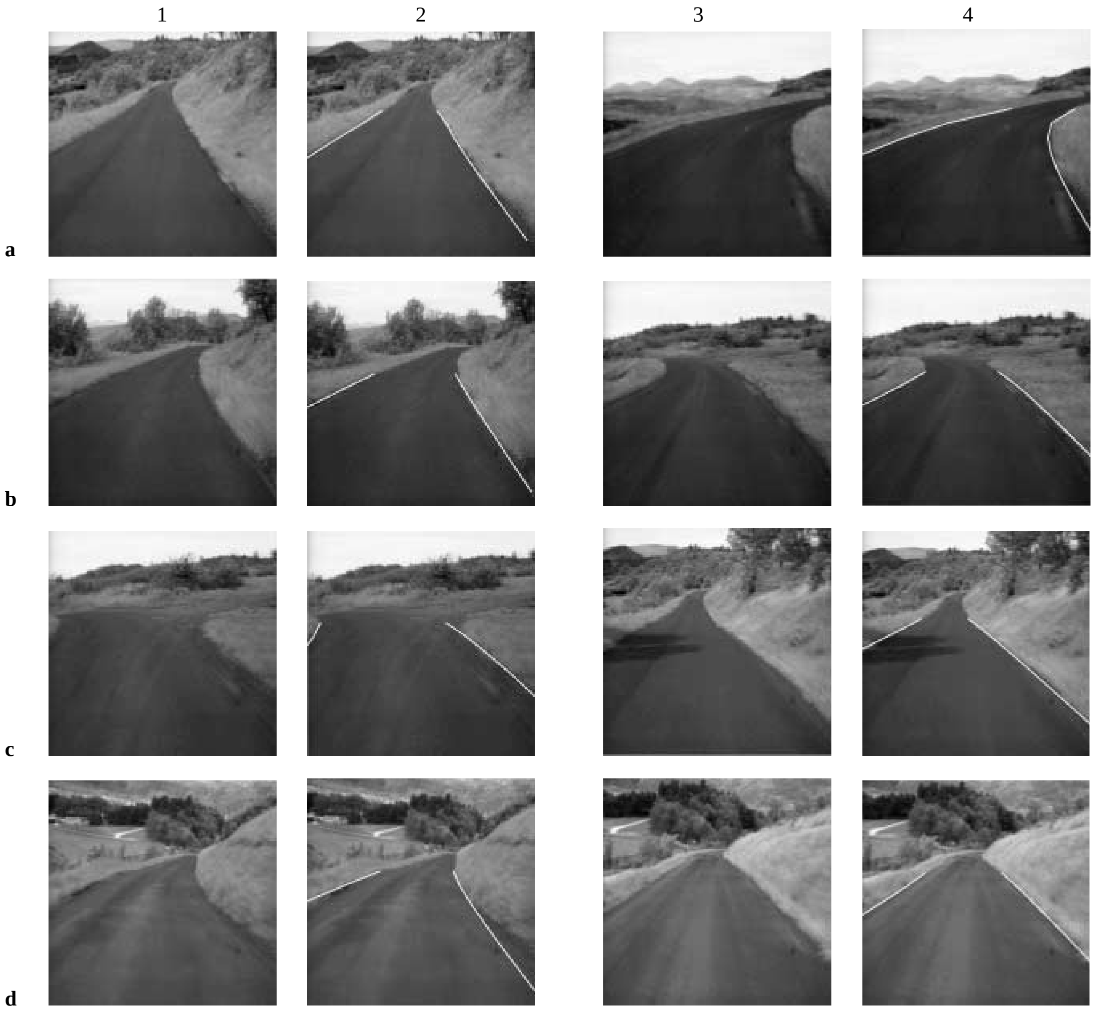

Fig. 16. Algorithm results on unmarked roads

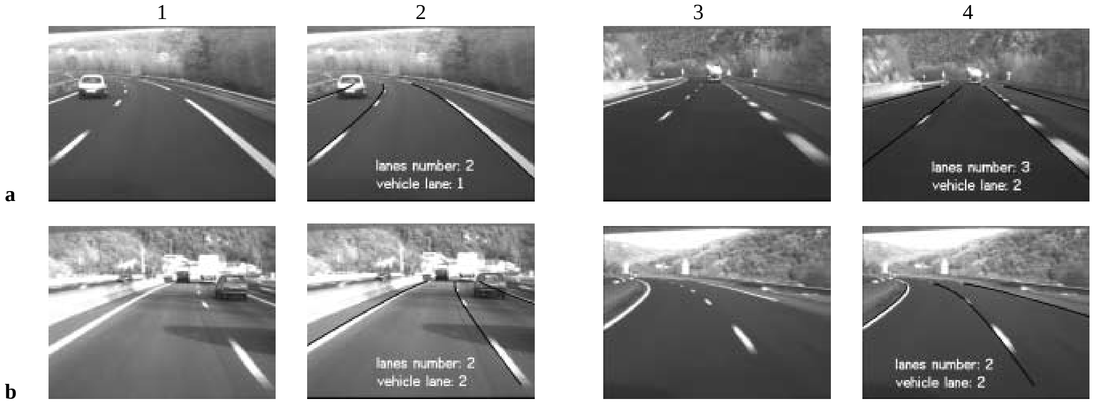

Fig. 17. Lane-number management results

required to transfer data from the T9000 to the DEC Alpha varies with the amount of transferred data, which depends on the size of the interest zone needed by the algorithm.

The histogram in Fig. 18 shows the distribution of the total computational time in the 2834 cases when the algorithm found the road. Figure 19 shows the distribution of the total number of iterations required to obtain the optimal solution in the 166 other cases when the algorithm could not found the road.

The algorithm provides its best solution in less than 150 iterations (480 ms), which means that continuing to detect the road after approximately 150 iterations is not useful. It is therefore legitimate to consider that the computation time does not exceed 500 ms. This seems rather too high, but

Table 2. Minimum and maximum calculation times for the different stages of the algorithm

Algorithm stage	Min	Max
Transfer from T9000 to DEC Alpha	28 ms	149 ms
Low level	4 ms	47 ms
Median least squares	16 ms	25 ms
Updating	1 ms	2 ms
Localization	1 ms	1 ms
Tracking	4 ms	4 ms
Total number of iterations	18	53
Total time	60 ms	238 ms

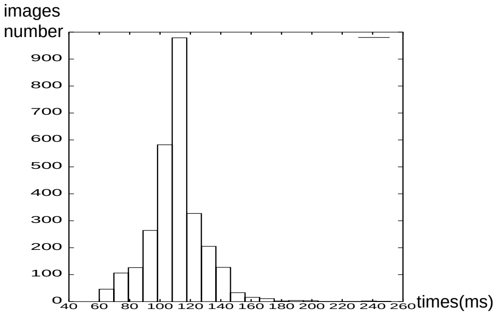

Fig. 18. Distribution of the total calculation times during the tested sequence where the algorithm succeeds in detecting the road

as illustrated in Fig. 19, it happens only a few times among the 166 cases. The best-detected road model is obtained after only 18 iterations for most (84%) of the images. According to these results, we have limited the number of iterations for a single image in the tracking phase to 150. This satisfies the time constraint of 5 images/s.

7 Conclusions and future work

A new road-tracking technique has been described in this article. The first module, which detects the road edges, uses a model that characterizes the road edges in the image. This model is updated in a recursive but optimal way after each detection. The main advantages of this process, which utilizes a powerful localized detection principle, are the precision (due to the high order of the model), the low sensitivity to occlusions or marking imperfections, and the possibility of applying it to marked and unmarked roads.

The second module computes both the vehicle location in its lane and the road parameters. The third module provides a confidence interval in which the first module will look for road edges in the next image. Finally, the last module manages the lane number. The implementation of the algorithm on our experimental vehicle demonstrates the validity of the approach suggested.

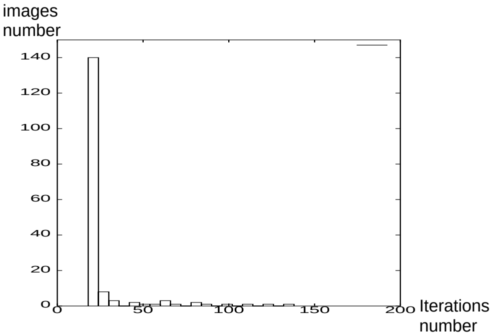

Fig. 19. Distribution of the total number of iterations required to obtain the best solution when the algorithm considers the road as not found

In the future we would like the algorithm to learn in real time the initial model in order to adapt the recognition process to any kind of road. Our method can be easily combined with other modules (e.g. radar and GPS). We also plan to incorporate an obstacle detector in order to determine the vehicle representing the greatest danger in the image. Thanks to the recognition method used, we can easily insert a new parameter into our initial vector to achieve this modification. This parameter will provide the obstacle-detection module with information on the image zone where it is necessary to seek an obstacle, but this would mean that the recognition module will not seek road edges when an obstacle is present. Finally, the whole process will be implemented in our experimental vehicle in order to validate a complete driver-assistance system.

References

Aufrère R, Chapuis R, Chausse F (2000a) Détection précise de bords de route par vision monoculaire embarquee (in French). In: Proceedings of the 12th French-speaking Conference on Pattern Recognition and Artificial Intelligence, Paris, France, 1–3 February, pp 229–238

Aufrère R, Chapuis R, Chausse F (2000b) A fast and robust vision based road following algorithm. In: Proceedings of the IEEE International Conference on Intelligent Vehicles, DearBorn, Mich., 4–6 October, pp 192–197

Breuel TM (1991) Model based recognition using pruned correspondence search. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, Lahaina, Maui, Hawaii, 3–6 June, pp 257–262

Broggi A (1995) An image reorganization procedure for automotive road following systems. In: Proceedings of the International Conference on Image Processing, Washington, D.C., 22–25 October, pp 532–535

Chapuis R (1991) Suivi de primitives image, application la conduite automatique sur route (in French). PhD thesis, Université Blaise Pascal, Clermont-Ferrand, France

Chen KH, Tsai WH (1997) Vision-based autonomous land vehicle guidance in outdoor road environments using combined line and road following techniques. J Robot Syst 14:711–728

Dickmanns ED, Mysliwetz BD (1992) Recursive 3-D road and relative egostate recognition. IEEE Trans Pattern Anal Mach Intell 14:199–213

Grimson WEL (1990) Object recognition by computer: the role of geometric constraints. MIT Press, Cambridge, Mass.

Guichard F, Tarel JP (1999) Curve finder combining perceptual grouping and a Kalman like fitting. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 21–24 September, pp 1003–1008

Herman M, Nashman M, Tsai-Hong H, Shneiderman H, Coombs D, Gin-Shu Y, Raviv D, Wavering A (1997) Minimalist vision for navigation. In: Aloimonos Y (ed) Visual navigation: from biological systems to unmanned ground vehicles. Erlbaum, Mahwah, N.J.

Kreucher C, Lakshmanan S (1999) A frequency domain approach to lane detection in roadway images. In: Proceedings of the International Conference on Image Processing, Kobe, Japan, 24–28 October, pp 31–35

Maitre H (1985) Un panorama de la transformation de Hough (in French). Traitement du signal 2:305–317

Marmoiton F (2000) Détection et suivi par vision monoculaire d'obstacles mobiles coopératifs partir d'un véhicule expérimental automobile (in French). PhD thesis, Université Blaise Pascal, Clermont-Ferrand, France

Pomerleau DA (1995) RALPH: Rapidly adapting lateral position handler. In: Proceedings of the IEEE Symposium on Intelligent Vehicles, Detroit, Mich., 25–26 September, pp 506–511

Tarel JP, Guichard F, Aubert D (1999) Tracking occluded lane-markings for lateral vehicle guidance. In: Proceedings of the Third IEEE Conference on Circuits, Systems, Communications and Computers, Athens, Greece, 4–8 July, pp 154-159

Wang Y, Shen D, Teoh EK (1998) Lane detection using catmull-rom spline. In: Proceedings of the Intelligent Vehicles Symposium, Stuttgart, Germany, 28–30 October, pp 51–57

Romuald Aufrère received his PhD from Blaise Pascal University, France, in 2001. His research into road-scene analysis for driving assistance is conducted at the LASMEA laboratory.

Roland Chapuis received his PhD from Blaise Pascal University, France, in 1991. He has been an assistant professor at the University Center for Sciences and Technics since 1992, and become a state doctor in 2000. His research into road-scene analysis for driving assistance is conducted at the LASMEA laboratory.

Frédéric Chausse received his PhD from Blaise Pascal University, France, in 1994. His research into road-scene analysis for driving assistance is conducted at the LASMEA laboratory.