MVA2000 IAPR Workshop on Machine Vision Applications, Nov. 28–30, 2000, The University of Tokyo, Japan

3—12

Real time vision based road lane detection and tracking

Romuald Aufrère Roland Chapuis Frédéric Chausse
LASMEA, UMR 6602 du CNRS
Université B. Pascal
63177 Aubière Cedex FRANCE
{aufrere,chapuis,chausse}@lasmea.univ-bpclermont.fr

Abstract

In this article, we describe a new method designed to detect and track the road lane of a vehicle equipped with a monocular monochromatic camera. The goal of this work is to assist the conductor in the framework of a trajectory supervision and control. The work described concerns the detection of the roadsides in the image based on the exploitation of a statistical model of the lanes, the localisation of the vehicle on its lane and the management of the number of lanes. The method presented here has been implemented on our experimental vehicle VELAC.
Results obtained show the robustness and the precision of the proposed approach.

1 Introduction

Since several years now, the scientific community works on projects related to the driving assistance problem. In this context, two types of methods can be identified whether they consider a road sides model or not.
On the one hand, the methods without model are often heavy to implement (such as for example the works based on a morphological filter [6] or the inverse perspective mapping [2]). Nevertheless, these methods do not manage the occlusions or the coherence between the features which involves bad detections. Moreover, the vehicle localization on its lane is carried out by an additional module which in any case exploits a geometrical model of the road-vehicle system.
In the other hand, many methods ([3, 7]) using a road model have been proposed but their major disadvantages are a weak precision if the model is simple (straight [5]) and a strong sensitivity to noise if the model is more complex (clothoidal curves [4], splines [8], ...). Moreover, these approaches run either for marked roads or on not marked roads but are very seldom applied to these two road types.

Taking into account these various remarks, our method is composed of four modules:

- A recognition phase allows to accurately determine the position of the road sides in the image. This phase is based on the exploitation of a statistical model of the lane.
  The method is applied both on marked and not-marked roads images. This step is described more precisely in [1].
- A localization step permits to deduce the lateral position and the orientation of the vehicle according to its lane, to obtain the camera inclination angle and some road parameters (lateral curvature and width).
  These parameters are calculated starting from the data resulting from the detection step.
- The next module updates both the number of lanes of the road (on highways) and the lane number on which the vehicle is running in order to complete the driving assistance system.
- A tracking module is then applied in order to limit the search space of the road edges in the next image starting from the localisation parameters computed at the previous image.

We describe first more precisely the various steps of our method. After our experimental vehicle and the dedicated architecture are presented. Then, some typical experimental results are given.

2 Description of our approach

For this algorithm, we use a statistical model composed of a vector $\underline{X}$ and its covariance matrix $\mathbf{C_X}$. $\underline{X}$ and $\mathbf{C_X}$ are decomposed in the following way:

$$\underline{X} = \begin{pmatrix} \underline{X}_d \\ \underline{X}_l \end{pmatrix} \text{ and } \mathbf{C}_X = \begin{pmatrix} \mathbf{C}_{X_d} & \vdots \\ \cdots & \mathbf{C}_{X_l} \end{pmatrix} \quad (1)$$

where $\underline{X}_d = (u_{1l}, ..., u_{nl}, u_{1r}, ..., u_{nr})^t$ represents horizontal image coordinates of both right and left roadsides for different image rows v_i ($i \in [1, n]$) ($n = 10$ in our case) and $\mathbf{C}_{X_d}$ the covariance matrix of $\underline{X}_d$. Only $\underline{X}_d$ and $\mathbf{C}_{X_d}$ will be used for the recognition process. A road edge corresponds to a list of $(n-1)$ segments.
$\underline{X}_l = (x_0, \psi, \alpha, C_l, L)^t$ represents localization parameters. x_0 is the lateral position of the vehicle on the roadway, ψ: vehicle steer angle, α: camera inclinaison angle, C_l: lateral curvature of the road and L: road width.
$\mathbf{C}_{X_l}$ is the covariance matrix of $\underline{X}_l$.
The initial value of the complete model $(\underline{X}, \mathbf{C_X})$ is obtained by an off-line training phase. In this phase, the vector $\underline{X}$ and its covariance $\mathbf{C_X}$ are calculated over a great number N of achievements of a road model (used in [3]) with random parameters: $x_0, \psi, \alpha, C_l, L$.
Figure 1 presents the complete organization chart of the algorithm.

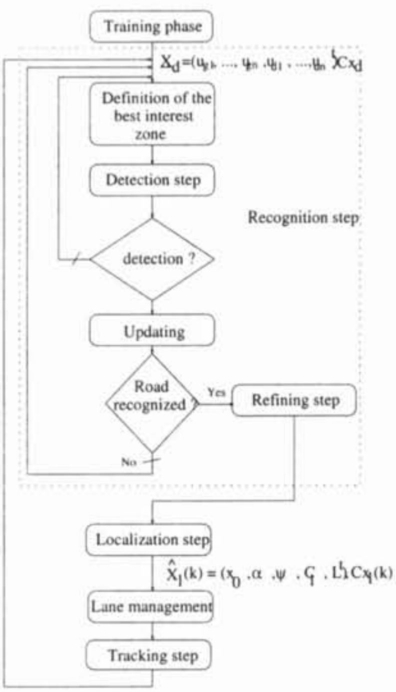


Figure 1: Organization chart of the algorithm.

2.1 Recognition step principle

This step is based on a recursive recognition of the roadsides driven by the probabilistic model $(\underline{X}_d, \mathbf{C}_{X_d})$ previously defined.
After the training phase (analysis depth $p = 0$), the smallest variance in the diagonal of $\mathbf{C}_{X_d}$ permits to characterize the best interest zone in which a part of road edges will be detected.

For each row, inside the search zone, the points of maximum gradient are located. These points are fitted to a straight line segment by a median least squares method taking into account a constraint on the segment's slope (deduced from $\mathbf{C}_{X_d}$) in order to provide a segment according to the need of the current model.
It can happen that no segment could be detected in the interest zone. In this case, the detection is attempted in a new one defined by the next smallest variance of the covariance matrix and so on.
If no segment is detected in all of the non-tested zones with the iteration depth p, the algorithm leaves this branch of the search tree, returns to the depth: $p-1$ and re-iterates the process on this new branch. The detection step gives two points (intersections between the detected segment and the borders of the search zone) which are used by a Kalman filter to update $\underline{X}_d$ and $\mathbf{C}_{X_d}$.
This search process (choice of the interest zone, detection, update) is recursively re-iterated and stops when a criterion Q is reached, the road is then assumed to be found.
This criterion is defined as follows, three probabilities are considered:

$$Q = [P_l = \frac{N_l}{\Delta_v} > 0.1 \text{ AND } P_r = \frac{N_r}{\Delta_v} > 0.1 \text{ AND } P_t = \frac{N_l + N_r}{2\Delta_v} > 0.3]$$

where N_l and N_r are respectively the number of detected points on the left and right sides and Δ_v (about $300 pixels$) represents the interval of the analysis zone $(v_n - v_1)$.
The thresholds have been fixed according to experimental considerations.
Nevertheless criterion Q can be reached without a complete analysis of the image. In order to obtain an optimal precision, the process restarts in the non-tested zones of the same branch. The final result of this step corresponds to the vector $\underline{X}_d$ resulting from the greatest probability P_t.
If Q is not reached, the algorithm considers the road is not found in the image (lack of information for example) but the best solution is provided.
Thereafter, the best estimation of $\underline{X}_d$ (result of recognition step) for image k is denoted by $\widehat{\underline{X}_d}(k)$.

2.2 Localization step

The goal is to compute the vehicle location on its lane by using vector $\widehat{\underline{X}_d}(k)$ and previous state vector $\underline{X}(k-1)$ in order to update the vector $\underline{X}_l(k)$ in the following way:

$$\begin{cases} \underline{X}(k) &= \underline{X}(k-1) + \mathbf{K}[\widehat{\underline{X}}_d(k) - \mathbf{H}\underline{X}(k-1)] \\ \mathbf{C}_X(k) &= \mathbf{C}_X(k-1) - \mathbf{KHC}_X(k-1) \end{cases}$$

- $\mathbf{K} = \mathbf{C}_X(k-1)\mathbf{H}^t\left[\mathbf{H}\mathbf{C}_X(k-1)\mathbf{H}^t + \mathbf{C}_{X_d}(k)\right]^{-1}$
- $\mathbf{H}$ is such as $\widehat{\underline{X}}_d(k) = \mathbf{H}\underline{X}(k) + \underline{w}$,
- $\widehat{\underline{X}}_d$: result vector of road detection process,
- $\mathbf{C}_{X_d} = E[\underline{w}\underline{w}^t]$: covariance matrix of $\widehat{\underline{X}}_d$.

Moreover, in the case of multiple lanes, the value of x_0 is used to determine on which lane our vehicle is located.

2.3 Management of the number of lanes

In order to complete the driving assitance system on highways, the algorithm manages the number of lanes. To carry out this process, $2n$ parameters are added to $\underline{X}_d$ vector in order to detect not only the lane sides but also the 2 lateral lines besides the circulation lane. This new model ($4n$ image parameters) is obtained by the training phase.
From the result of the recognition step (1 lane = $2n$ parameters), an updating of the complete model ($4n$ parameters) permits to limit the search space of lateral lines on the same image. Thereafter, the recognition process is restarted in the $(2n-2)$ zones corresponding to the lateral lines ($(n-1)$ for the left lateral white line and $(n-1)$ for the right lateral white line).
A lateral line is considered found if the associated probability ($\frac{N_{ll}}{\Delta_v}$ and/or $\frac{N_{rl}}{\Delta_v}$) is greater than 0.1. N_{ll} and N_{rl} represent the number of detected points on the left and right lateral white lines.
A line detection permits to update a table of probality on the presence or not of the lateral lanes and another on the identification of the lane where our vehicle is running.

2.4 Tracking step

The recognition step for the next image is more efficient if it begins with a search space coming from present image.
To carry out this process, vector $\underline{X}_l(k) = (x_0, \psi, \alpha, C_l, L)^t$ (deduced from $\underline{X}(k)$) becomes $\underline{X}_l(k+1)$ according to the displacement Δ_Y of the car between two successive images.

$$\left\{ \begin{array}{lcl} \underline{X}_l(k+1) & = & \mathbf{M}\underline{X}_l(k) + \underline{w}_t \\ \mathbf{C}_{X_l}(k+1) & = & \mathbf{M}\mathbf{C}_{X_l}(k)\mathbf{M}^t + \mathbf{Q} \end{array} \right.$$

- $\mathbf{M}$ evolution matrix such as:

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

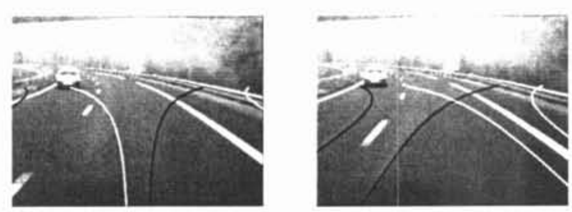
- $\mathbf{Q} = E[\underline{w}_t\underline{w}_t^t]$ errors evolution matrix (experimental values).

Thereafter, to define the optimized initial image area in the next image, the model ($\underline{X}$, $\mathbf{C}_X$) resulting from the training stage is updated using vector $\underline{X}_l$ and matrix $\mathbf{C}_{X_l}$ previously computed.

$$\left\{ \begin{array}{lcl} \underline{X}(k+1) & = & \underline{X}(k) + \mathbf{K}_t[\underline{X}_l(k+1) - \underline{X}_l(k)] \\ \mathbf{C}_X(k+1) & = & \mathbf{C}_X(k) - \mathbf{K}_t\mathbf{H}_t\mathbf{C}_X(k) \end{array} \right.$$

- $\mathbf{K}_t = \mathbf{C}_X(k)\mathbf{H}_t^t\left[\mathbf{H}_t\mathbf{C}_X(k)\mathbf{H}_t^t + \mathbf{C}_{X_l}(k+1)\right]^{-1}$
- $\mathbf{H}_t$ is such as $\underline{X}_l = \mathbf{H}_t\underline{X}$.

Figure 2(a) shows the initial search zone of the lines in the image (black curve for the left side and white curve for the right side) resulting of the training phase and figure 2(b) presents the initial search zone after a tracking step.



(a) (b)

Figure 2: Initial search zone (a) without tracking and (b) with tracking.

3 Implementation

For a real time running, the algorithm has been implemented on an architecture dedicated to vision applications. The system is on-boarded in our experimental vehicle (VELAC) integrating vision sensors (fig 3). This architecture is based on an MIMD module using 4 DEC alpha type processors. Actuators are controlled via a CAN bus. Our algorithm has been ported on a single DEC Alpha processor.

Figure 3: Our experimental vehicle (VELAC) and its architecture.

4 Results

Figure 4 presents results related to the recognition step both on marked and not-marked roads.

The camera used have a focal $f = 12mm$ and was placed at a height $z_0 = 1.5m$. The lanes are perfectly detected despite the shadows or the sunlights

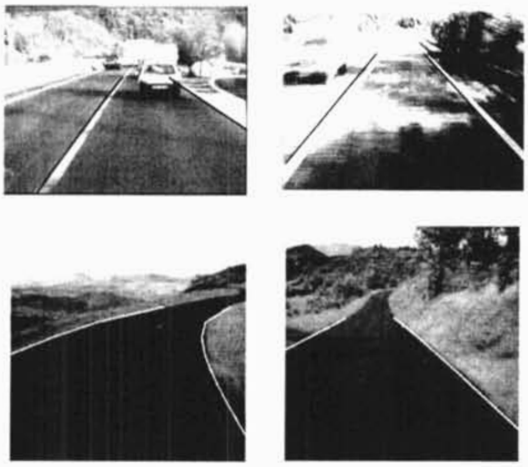


Figure 4: Results on marked and not marked roads.

Figure 5 shows images where several lanes are present.

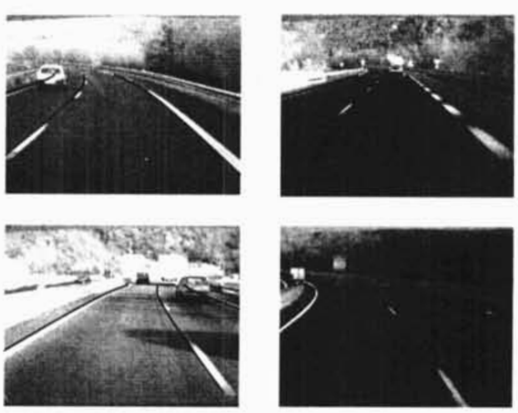


Figure 5: Results of detection on several lanes.

To validate the approach chosen, the method was the subject of a complete study on thousands images.
The process running on only one processor DEC alpha, the computational time generated by the method is approximately of 60 to $180ms$ per image. These times are obtained when the algorithm considers the road as found in the image which correspond to number of iterations between 18 and 150. When the algorithm considers the road not found, the best solution is obtained before 200 iterations in a large majority of the cases which correspond to a computational time about $250ms$. The results obtained by this method are very encouraging both on marked and not-marked roads.

5 Conclusion and perspectives

A new road tracking method has been designed and can be used for the driving assistance. Its main advantages are the precision (high order of the model), the local and powerful detection principle (but global research), the weak sensitivity to occlusions or marking imperfections and the possibility to apply the method on marked and not marked road images as well. Furthermore, this aproach is able to manage several lanes.
The prospects related to this work are varied. Initially, we wish to learn on-line the initial model in order to adapt the recognition process whatever the kind of road. Then, we will plan a collaboration with an obstacles detection algorithm in order to pick out the vehicle representing the greatest danger.
Finally, the whole process will be implemented in our experimental vehicle in order to validate a complete driver assistance system.

References

[1] R. Aufrère, R. Chapuis, and F. Chausse. Détection précise de bords de route par vision monocculaire embarquée. In *12ème congrés Reconnaissance des Formes et Intelligence Artificielle*, Paris, France, 1-3 Février 2000.

[2] M. Bertozzi and A. Broggi. Gold: a parallel realtime stereo vision system for generic obstacle and lane detection. In *IEEE Transactions on Image Processing*, pages 62–81, January 1998.

[3] R. Chapuis, A. Potelle, J.L. Brame., and F. Chausse. Real time vehicle trajectory supervision on highway. *International Journal of Robotics Research*, 14(6):531–542, December 1995.

[4] E.D. Dickmanns and V. Graefe. Dynamic monocular machine vision and applications of dynamic monocular vision. *International Journal of Machine Vision and Application*, 1:223–240 and 241–261, 1988.

[5] S.K. Kenue. Lanelok: Detection of lane boundaries and vehicle tracking using image-processing techniques-parts i and ii. In *SPIE Mobile Robots IV*, 1989.

[6] B. Serge and B. Michel. Road segmentation and obstacle detection by a fat watershed transform. In *Intelligent vehicles symposium*, pages 296–301, Paris, France, october 1994.

[7] L.T. Shaaser and B.T. Thomas. Finding road lane boundaries for vision guided navigation. *Proceedings of the Rountable discussion on Vision Vehicle Guidance*, pages 10.1–10.9, 1990.

[8] Y. Wang, D. Shen, and E.K. Teoh. Lane detection using catmull-rom spline. In *Intelligent vehicles symposium*, volume 1, pages 51–57, Stuttgart, Germany, 28-30 October 1998.