Tracking Occluded Lane-Markings for Lateral Vehicle Guidance

Jean-Philippe TAREL Frédéric GUICHARD Didier AUBERT
INRETS
2, avenue du Général Malleret-Joinville
BP 34-94114 Arcueil-Cedex
FRANCE
Email: Jean-Philippe.Tarel@inrets.fr

Abstract:- We present a real time algorithm for computing the orientation and the lateral pose of a vehicle with respect to the road observed by an on-board video camera. The advantage of this approach is to provide robust measures when lane-markings are dash, partially missing, perturbed by shadows, highlights, other vehicles, and noise. Moreover, a calibrated camera may be used as well a uncalibrated camera.

Robustness to intensity perturbations is obtained, as much as possible, by taking into account all the edges in each image without doing any a priori thresholding based on the gray-level amplitudes. This leads to numerous edges to be processed. Nevertheless, we propose an algorithm extracting straight line segments directly in gray-level images in less than 0.05 second for a 256x256 image on a Pentium 200Mhz.

For robust estimates of the orientation and of the lateral pose under geometric perturbations such as missing data, the algorithm relies on few basic assumptions on the observed cues: aligned and straight cues. Moreover, we assume that the road profile is changing slowly from one frame to another.

The orientation of the vehicle with respect to the road is computed by estimating the position of the focus points of the markers along the line of horizon. From the extracted straight line segments, the algorithm builds an histogram which represents the current lateral road profile. The relative lateral pose of the vehicle is obtained by comparing the lateral profile of the current road to a reference lateral road profile. Using the consistency over time of the lateral road profile, the reference is updated. Initial reference is built during the initialization of the process when the vehicle is assumed well enough aligned with the road.

In the proposed approach, if lane-markings are missing, estimated orientation and pose can still be computed using the other cues seen in the scene such as cues from side-walks, road shoulders, herb-sides, and guard rails. This allows us lateral guidance without explicit recognition of the left and right lane-markings as it is usually proposed.

First experiments on real images show correct estimations in presence of dash lane-markings, missing markers, highlights, shadows, curves, and noise. The whole process can run typically at a rate from 5 to 15 images per second, and the obtained measures can be used for helping the driver in the lateral guidance of its vehicle, in case of uncontrolled lane departure for instance. This system may help toward a solution of major security problems of road traffic, such as obstacle detection near the vehicle.

Key-Words:- Pattern Analysis and Machine Intelligence, Computer Vision, Image Processing, Projective Geometry, Real Time Systems, Feature Extraction, Edge and Line detection, Identification, Tracking, Applications to Vehicle Guidance.

1 Introduction

An efficient lane-markings detector is of importance for vehicle guidance. Lane-markings detection can be used for helping the driver in the lateral guidance of its vehicle, for instance. In case of an uncontrolled lane departure, an alarm may be sent to the driver or an automatic braking/steering may be applied onto the vehicle.

For lateral vehicle guidance, lane-markings detection must at least provide estimates of the relative orientation and of the lateral position of the vehicle with respect to the road. The accuracy of the estimated values is also required. The process must be in real time, i.e, at a rate of at least 5 images per second, for a correct control of the vehicle at a speed of 30 m/s.

Automatic vehicle guidance has been a subject of investigations from many years [**?**, **?**], but until now, to our knowledge, there is no technique tackling the problem of partially missing lane-markings which is of importance for real data application where lane-markings may be dash, hidden by shadows, highlights, and vehicles. Therefore, we propose a technique which focuses on robustness to missing data.

We first describe our geometric assumptions about the lane-markings the system has to detect for lateral guidance of a vehicle, in section 2. This model allows us to detect lane-markings by processing each image in real time. Then, we explain the temporal assumption about the road lateral profile evolution for dealing with perturbations. The proposed image process consists in three steps. First, we describe in section 3 our fast straight line segment detector. Second, the focus point, and thus the orientation of the vehicle with respect to the road, is estimated (section 4). Finally, the road profile is computed and the pose of the car with respect to the lane-markings is obtained (section 5).

2 Road Lane-markings Model

We chose to base our approach on a minimal number of assumptions, and thus to use simple geometric cues. Indeed, we expect to have a better control of the algorithm by minimizing the number of controlling parameters. This is important for experimental validations.

The interesting markers for lateral guidance of a vehicle on a road are the continuous and dash horizontal lane-markings. The orientation and the lateral pose of the vehicle have to be estimated only with respect to the lane-markings near the vehicle. This leads to detect markers by using straight line segment cues in the images.

The boundary of the markers are straight lines on the road with a good approximation depending on:

- the considered lane length,
- the altitude variations of the road,
- and the maximum curvature of the road.

The chosen cues for describing markers on the road does not contain any information about the width of the markers, and the distance between the lane-markings. Indeed, depending on the road type and on the road section these features may vary. We did not assume any a priori values for these parameters, but they are assumed to change slowly. The previous assumption is valid most of the time, but not for certain specific road configurations such as lane crossing, and lane merging. In such cases, the proposed algorithm is not able to provide a correct lateral pose. But the orientation and the time update of the lateral pose is still obtained allowing us to perform adaptive lateral control in this degraded mode.

The previous assumption of consistency over time is needed to improve the robustness of the lane detection in front of the following difficulties:

- dash lane-markings and missing data,
- perturbations in the drawing of the lane-markings boundary,
- additive edges due to highlights and shadows.

This constraint allows us to dynamically build the lateral profile of the road, i.e, the representation of how cues are distributed along a cross section of the road. This last is used for estimating the relative lateral position of the vehicle with respect a reference lateral road profile.

3 Straight Line Segment Detector

Straight line segments are chosen as features. Extracting straight line segments from an image may be a time consuming task. We design a fast algorithm (typically 0.05 second on an image) without any pre-processing on the gray-level image.

3.1 Edgel

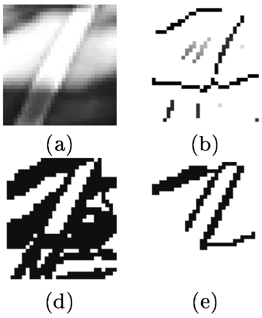

Figure 1: *(a) original image of a marker perturbed by highlights, results of Canny-Deriche edge detector with a 1 pixel size smoothing: (b) no threshold on the gradient magnitude and (c) 40 gray levels threshold. On the second line, results of the line segment detector for different values of the minimal length: (d) 4 pixels, (e) 15 pixels and (f) 20 pixels.*

Most of the edge-detectors algorithm involve (at least) a smoothing and a threshold steps [?]. Both steps perform a selection on the edgels, and therefore on the remaining information:

Smoothing: It removes from the image small details created by noise. Since, the chosen filtering is often linear, a small detail is a reduced set of pixels with low gray-level amplitude. Information on intensity and spatial size is thus merged. As a consequence, the selection is harder on low-contrast zones. For example, we show in Fig. 1 (b) or (c), a Canny-Deriche edge detector [?] applied on an image of an highlighted white marker. The magnitude of the gradient along the shadow is so strong that the smoothing removes one edge of the marker we want to detect. Ideally, the smoothing should be tuned to remove only what can be attributed to noise.

Thresholding: it discards low contrasted edgels, which again involves an selection based on gray-level amplitudes. Choosing the adequate threshold is a difficult issue since as noticed in [?], "*The resulting edge quality varies greatly with the choice of parameters*". No thresholding seems better that a not fully justified selection on gray-level amplitudes when lightening conditions are under control.

If we remove the gray-level smoothing and threshold steps as much as possible, edgels in images are numerous, and thus a criterion for selecting the edgels of interest is required (see Fig. 1(d)(e)(f)). We based this selection on geometrical features. We believe this is a better alternative than a selection based on gray-level amplitudes, as illustrated in Fig. 2.

3.2 Straight Line Segments of Edgels

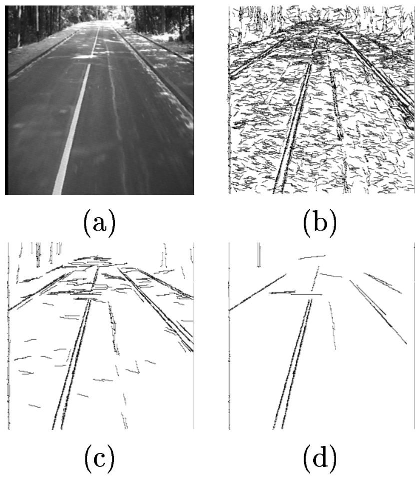

Figure 2: *(a) the original image and the result of the line segment detector for different values of the minimal length (b) 8 pixels, (c) 16 pixels and (d) 32 pixels.*

We propose to define edges as the set of all the level lines of the image. As defined in [?], we call *level line* the boundary of a level set L_μ, the set of pixels having an intensity larger or equal to μ. Of course, another kind of edges may be used (e.g. lines given by the zero crossing of the Laplacian). The important thing, at this point, is to avoid, as much as possible, a selection based on gray-level amplitudes.

Due to image grid, there is only 8 possible local directions for edgels. These directions are coded by a number between 0 and 7, the well known Freeman codes. The list of directions of these connected edgels is the so-called chain code.

Different algorithms has been proposed for recognizing when a chain code of a list of connected edgels is a straight line or not [?]. Any of these algorithms allows us to construct a tree of the whole possible straight line chain codes up to a specified length N. Due to the $\frac{\pi}{4}$ symmetry of the process, Freeman [?] shows that at most two basic directions are present in the chain code and these can differ only by unity, modulo 8. Thus this tree is a binary tree and the size of this tree is relatively small. Moreover, from [?], we know that an asymptotic estimate of the number of straight chain codes of length N is $\frac{N^3}{\pi^2}$.

After building this tree, a very fast algorithm for following connected straight segments of edges can be implemented. Given a starting edgel, the edge line is followed until we stand on a straight line. This detection is repeated until all the straight line segments are extracted in the image.

Figure 3: *Extracted straight line segments. Line segments with a slope closed to horizontal are discarded.*

In the framework of our application, line segments closed to horizontal are discarded for improving the speed of the detector. Segments far from the vehicle are also discarded as shown in Fig. 3. This enforces the assumption of straight lane-markings in case of a curve of the road.

4 Orientation of the Vehicle

The next step in the processing is to estimate the focus point of the markers and other cues.

(a) (b)

Figure 4: *(a) the up-side-down histogram displays the accumulated lengths of the line segments crossing the line of horizon. The used line segments are shown in Fig. 3. The line of horizon and the focus point are shown. (b) the histogram displays the accumulated lengths of the line segment crossing a vertical line going through the focus point. The median and extent of both histograms are displayed with lines.*

The observed road is assumed planar and the camera has no roll angle with respect to the road. Therefore, the transformation between the road plane (x^*, y^*) and the image plane (x, y) is:

$$x = l_x \frac{x^*}{y^*} \tag{1}$$

$$y = l_y \frac{1}{y^*} \tag{2}$$

where l_x and l_y are only functions of the camera calibration parameters. In (1) and (2), we set the origin of the image coordinate system to the center of the line of horizon (see Fig. 4(a)). Since lane-markings edges are parallel straight lines on the road plane, the lines converge in the image to a point on the line of horizon. It is the focus point.

From (1), we deduce that the position of the focus point along the line of horizon is linearly related to the tangeante of the angle between the camera axis and the lane-markings. Similarly from (2), the elevation of the line of horizon is linearly related to the slope of the observed road.

Experimentally on real images, it turns out that the position of the focus point on the line of horizon is more accurately estimated than the elevation of the horizon. Hopefully, contrary to the orientation of the vehicle, the slope of the road varies very slowly. Therefore, elevations of horizon are averaged along several frames for robust estimates.

Knowing the elevation of the line of horizon, the compute the histogram of the accumulated lengths of the line segments crossing the line of horizon. This is fast to compute. As shown in Fig. 4(a), this histogram consists mainly in a shape with a unique mode, and can be seen as the probability density of presence of the focus point along the line of horizon. Its median is a robust estimate of the focus position. Its extent estimates the accuracy of the obtained position.

Since the angle between the lane-markings and the vehicle is directly related to the position of the focus point, there is no need to recognize each marker separately. Therefore, using the focus point allows us to estimate the orientation of the vehicle even if no white line markings are present. Indeed, any other line cues such as sidewalks, herb-sides, guard rails are contributing to this estimate.

Then another histogram is build by crossing the extracted segments with a vertical line going toward the focus position. A typical example of this kind of histogram is shown is Fig. 4(b). As previously, the median and the size of the mode are used to update the estimate of the horizon elevation.

5 Lateral Pose of the Vehicle

For computing the lateral pose of the vehicle, previous algorithms are usually based on recognition of the left and right lane-markings. The difficulty is that the width of the markers and the distance between the lane-markings varies from one road to another. Even more difficult, one marker may be missing in several frames and thus confused to another closed markers. We propose an approach taking advantage of that the lateral pose of the vehicle may be estimated without recognition of the lane-markings.

From the ratio of (2) and (1), we deduce that the arc-tangeante of the angle of the image of a line segment is linearly related to lateral position of the line segment on the road. Thus, the histogram of the accumulated lengths of segments as a function of the arc-tangeante of their angles is linearly related to the road profile. The road profile is a kind of summary of the position of the observed markers and other cues along a road section. For example in Fig. 5, each mode corresponds to one lane-marking as displayed by the arrows.

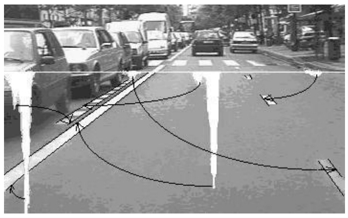

Figure 5: *The histogram displays the accumulated lengths of the line segments with respect to their angles. Used line segments converge to the focus point.*

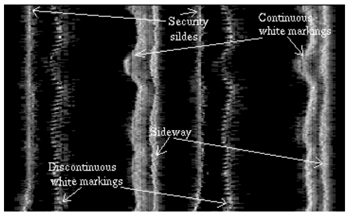

Figure 6: *Evolution along time of the road profile.*

In Fig. 6, each line is an lateral profile. Different kinds of markings may be detected along time. Due to the affine mapping the lateral position of the vehicle is the relative translation of this histogram. Given a reference histogram and from the current histogram, the current lateral pose error may be estimated using a cross-correlation technique for instance. A more sophisticated technique doing histogram mode matching by dynamic programming may be also used to allow small variations of the marker positions.

At the initialization of the lateral control process, the vehicle is assumed in the right position. At this time, the reference road profile is computed. Then the algorithm continually estimates the alignment error between the vehicle and the road by computing the error of alignment between the current and reference histograms. Moreover, the reference histogram is dynamically updated with an exponential averaging of the new road profile after alignment (usually the averaging is on 300 frames). Without camera calibration the error of lateral pose and orientation are known up to a scale factor. Therefore, the lateral control may be performed with as well as without camera calibration.

6 Experiments

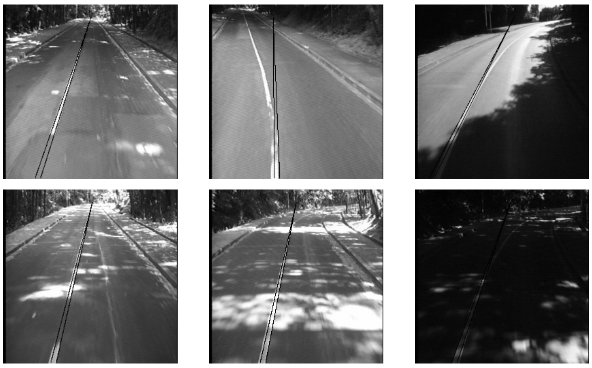

Figure 7: *Main lane-markings detected in presence of perturbations such as missing data, shadows, highlights, road curves.*

In Fig. ??, the main white lane-markers is correctly posed and oriented despite the various perturbations. Size of the images is 256x256. Typical computation time on a Pentium 200Mhz, 32Mo is 0.05 second for straight line segment detection (keeping only these that are at least 8 pixels long), and 0.02 second for the focus and lateral pose estimation.

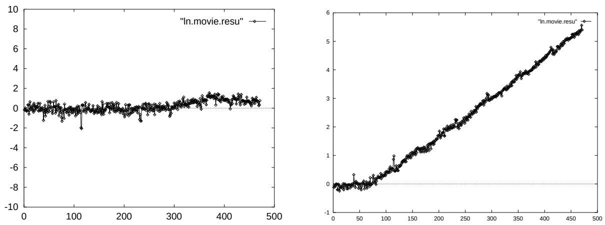

Figure 8: *Estimated angle (in pixels) and lateral pose of the vehicle (in meters/10).*

For testing how reliable is the estimated orientation and pose, we run experiments on sequences of synthetic images where true values are known. In Fig. ??, the estimated angle has to be compared with constant zero value. The true pose is zero during 50 frames and then increases linearly during the last 450 frames as it can be seen in Fig. ??. The standard deviation of the angle is 1.1 pixels (0.2 degree) and 1.5 cm.

7 Conclusion

We proposed an efficient technique for computing the orientation and the lateral pose of a vehicle with respect to the observed road. The orientation is computed by estimating the position of the focus points of the markers and other cues along the line of horizon. The relative lateral pose of the vehicle is obtained by comparing the current road profile to a reference lateral profile of the road. This reference profile is dynamically updated. The main advantage of this technique is to provide robust measures when lane-markings are partially missing, dash, perturbed by shadows, highlights, missing data, and noise. This real time algorithm is based on a fast straight line segment detector in gray-level images (from 5 to 15 frames per second). Extensions of this system for curved road is under investigation.

Efficient lane-markings detection systems can help toward a solution of major security problems of road traffic, such as obstacle detection near the vehicle.

References

[1] C. Berenstein and D. Lavine. On the number of digital straight line segments. *PAMI*, 10(6):880–887, November 1988.

[2] T. K. C. Thorpe, M. Herbert and S. Shafer. Vision an navigation for the canegie-mellon navlab. *PAMI*, 10(3), 1988.

[3] R. Deriche. Using canny's criteria to derive an optimal edge detector recursively implemented. *International Journal of Computer Vision*, 1(2), 1987.

[4] E. Dickmanns and A. Zapp. A curvature-based scheme for improving road vehicle guidance by computer vision. In *Proceedings of SPIE Conference on Mobile Robots S.161-16*, volume 727, 1986.

[5] F. Guichard. *Axiomatization of images and movies scale-space.* PhD thesis, University of Paris IX-Dauphine, 1994.

[6] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer. A robust visual method for assessing the relative performance of edge detection algorithms. *PAMI*, 19(12):1338–1359, December 1997.

[7] L. Wu. On the chain code of a line. *PAMI*, 4(3):347–353, May 1982.