



ELSEVIER

Pattern Recognition Letters 21 (2000) 677–689

Pattern Recognition  
Letters

www.elsevier.nl/locate/patrec

# Lane detection using spline model

Yue Wang, Dinggang Shen, Eam Khwang Teoh \*

*School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Block S1, Singapore 639798, Singapore*

Received 13 November 1998; received in revised form 8 February 2000

## Abstract

In this paper, a Catmull–Rom spline-based lane model which describes the perspective effect of parallel lines has been proposed for generic lane boundary. Since Catmull–Rom spline can form arbitrary shapes by different sets of control points, it can describe a wider range of lane structures compared with other lane models, i.e. straight and parabolic models. Moreover, the lane detection problem has been formulated here as the problem of determining the set of control points of lane model. The proposed algorithm first detects the vanishing point (line) by using a Hough-like technique and then solves the lane detection problem by suggesting a maximum likelihood approach. Also, we have employed a multi-resolution strategy for rapidly achieving an accurate solution. This coarse-to-fine matching offers us an acceptable solution at an affordable computational cost, and thus speeds up the process of lane detection. As a result, the proposed method is robust to noise, shadows, and illumination variations in the captured road images, and is also applicable to both the marked and the unmarked roads. © 2000 Elsevier Science B.V. All rights reserved.

*Keywords:* Lane detection; Catmull–Rom spline; Lane model; Machine vision; Maximum likelihood; Intelligent vehicle

## 1. Introduction

Autonomous guided vehicles (AGVs) have found many applications in many industries. Their utilization had been explored in areas such as hospitals for transportation of patients, automated warehouses and other hazardous related areas. In most applications, these AGVs have to navigate in unstructured environments. Path findings and navigational control under this situation are accomplished from the images captured by camera mounted on the vehicles. These images are inter-

preted to extract meaningful information such as positions, road markings, road boundaries, and direction of vehicle's heading. Among many extraction methods, the lane marking (or road boundary) detection from the road images had received great interest. As the captured images are usually corrupted by noises, lots of boundary-detection algorithms have been developed to achieve robustness against these noises.

The main properties that the lane marking (or boundary) detection techniques should possess are:

- The quality of lane detection should not be affected by shadows, which can be cast by trees, buildings, etc.
- It should be capable of processing the painted and unpainted roads.

\* Corresponding author. Tel.: +65-790-5393; fax: +65-792-0415.

*E-mail addresses:* wangyue@mbox4.singnet.com.sg (Y. Wang), dgshen@cbmv.jhu.edu (D. Shen), eekteoh@ntu.edu.sg (E.K. Teoh).

- It should handle curved roads rather than assuming that the roads are straight.
- It should use the parallel constraint as a guidance to improve the detection of both sides of lane markings (or boundaries) in the face of noises in the images.
- It should produce an explicit measurement of the reliability of the results obtained.

In Section 2, the reviews on the existing lane-detection techniques are presented, and the analyses of the shortcomings of each lane-detection technique are also given. Section 3 introduces a new lane model based on the Catmull–Rom spline. The relationship between the control points of both sides of lane model is also shown. In Section 4, a Hough-like method for the vanishing point (line) detection is presented. In addition, an algorithm which connectedly uses the new lane model and the maximum likelihood method is developed for lane detection. Section 5 shows some representative results of applying the proposed lane-detection algorithm to various types of roads and environments. This paper concludes in Section 6.

## 2. Related works

Up to present, various vision-based lane-detection algorithms have been developed. They usually utilized different road models (2D or 3D, straight or curve) and different techniques (Hough, template matching, neural networks, etc.).

An approach embodied in the ARCADE system (Kluge, 1994) uses robust estimation to determine the road curvature and orientation from the positions and orientations of the edge points, without prior grouping of the edge points into the lane edges. Once the road curvature and orientation have been determined, ARCADE uses them to *reduce* the problem of locating the lane edge offsets with respect to the vehicle *to* the problem of segmenting a one-dimensional signal. This signal is constructed by averaging together all images' pixels with the same offset relative to the road center, in order to smooth out variations due to shadows, texture, etc. The use of robust estimation allows ARCADE to work in cases where up to 50% of the input edge points are noise, but the

algorithm fails when the fraction of noise edges rises above that point.

The approach based on morphological filtering has been suggested (Serge and Michel, 1994; Xuan et al., 1992). This technique used the morphological “watershed” transformation to locate the lane edges in the intensity gradient magnitude image. Although this technique has the advantage of not requiring any thresholding for the gradient magnitudes, it has the disadvantage of not imposing any global constraints on the lane edge shapes.

A curve road model was proposed by Kluge and Lakshmanan (1995) and Lakshmanan and Kluge (1995). It was supposed that the lane boundaries can be presented by a parabolic curve on a flat ground. Although it can approximate normal road structures, it still cannot describe some cases, i.e. a *T* turn. A deformable template method was proposed by optimizing a likelihood function based on this model. However, this algorithm cannot guarantee a global optimum and the accuracy, without requiring huge computational resources.

An edge-based road detection algorithm was presented by Broggi (1995a,b), Broggi and Berte (1995) and Bertozzi and Broggi (1998), it could work nicely in well-painted roads even under shadowy condition, but it will fail for the unpainted roads.

The approach by Grimmer and Lakshmanan (1996), working in the domain of locating pavement edges in millimeter wave radar imagery, used a deformable template approach to find the best fit of a straight road model with unknown width and orientation to the radar data. The likelihood function used to judge how well a given template shape matched the radar data by combining geometric constraints with a model of the physics of the radar image formation. The Metropolis algorithm was used to identify the optimal set of template deformation parameters. This technique has a disadvantage of detecting straight road only.

An approach by combining the Hough transform and the Line–Snake model was presented by Jung Kang et al. (1996), it divided an image into a few sub-regions along the vertical direction. The

Hough transform was then performed for each sub-region to obtain an initial position estimation of the lane boundaries. Afterwards, the Line-Snake improved the initial approximation to an accurate configuration of the lane boundaries. This approach suffers from two problems. One is, in the case of broken lane markings, it may not extend all the ways to the bottom of the image. Another is, the contrast of one (or both) of the lane edges may not be high enough to detect near the bottom of the image.

In (Kaske et al., 1997, 1995), an approach of detecting lane boundary, especially for the country roads, by artificial vision was described. It used statistical criteria, i.e. energy, homogeneity, contrast and etc., to distinguish between the roads and the non-roads. It combined the random searching with the chi-square fitting to obtain the best set of parameters of a deformable template. However, they used the same road model as (Kluge and Lakshmanan, 1995; Lakshmanan and Kluge, 1995).

A road recognition method by fuzzy reasoning was presented in (Li et al., 1998), it integrated some special knowledge into fuzzy rule base to recognize road edges based on an approach for general edge extraction. This method has two drawbacks: First, a road map has to be saved in computer memory as knowledge; Second, in order to build a template of a road edge, its initial points on the road edge have to be determined manually.

Here, we present a lane model based on Catmull–Rom spline and a matching measurement between the model and the real road edges based on maximum likelihood.

### 3. Road model

#### 3.1. Catmull–Rom spline

The Catmull–Rom spline, also known as Overhauser spline, is a local interpolating spline developed for computer graphics purpose. It was initially used to design curves and surfaces in graphics, but recently it has been applied in several other applications.

Usually, we have a series of point positions and want to design a curve to smoothly interpolate (or pass through) them. In this situation, Catmull–Rom spline can be used to interpolate the points  $P_1$  to  $P_{m-1}$  from the sequence of points  $P_0$  to  $P_m$ . In addition, the tangent vector at point  $P_i$  is parallel to the line connecting points  $P_{i-1}$  and  $P_{i+1}$ , as shown in Fig. 1. Catmull–Rom spline was derived from Ferguson’s parametric cubic curve.

##### 3.1.1. Ferguson’s parametric cubic curves

Given the two control points  $P_0$  and  $P_1$ , and the slope of the tangents  $P'_0$  and  $P'_1$  at each point, we can define a parametric cubic curve that passes through  $P_0$  and  $P_1$ , with the respective slopes  $P'_0$  and  $P'_1$  by equating the coefficients of the polynomial function

$$P(t) = a_0 + a_1t + a_2t^2 + a_3t^3$$

with the values above. Namely

$$\begin{aligned} P(0) &= a_0, \\ P(1) &= a_0 + a_1 + a_2 + a_3, \\ P'(0) &= a_1, \\ P'(1) &= a_1 + 2a_2 + 3a_3. \end{aligned}$$

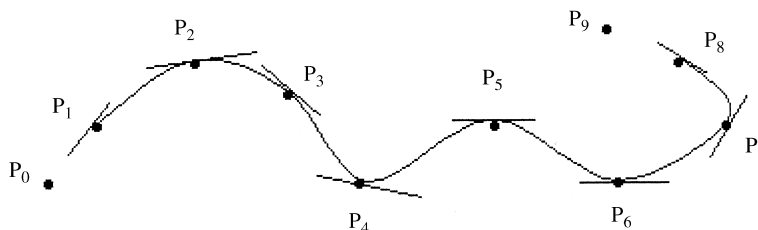


Fig. 1. A Catmull–Rom spline. The points are interpolated by the spline, which passes through each point in a direction parallel to the line between the adjacent points. The straight line segments indicate these directions.

Solving these equations simultaneously for  $a_0, a_1, a_2, a_3$ , and substituting these into the original polynomial equation and simplifying to isolate the terms with  $P_0, P_1, P'_0, P'_1$ , we have

$$P(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} P(0) \\ P(1) \\ P'(0) \\ P'(1) \end{bmatrix}.$$

It can be used to obtain a curve through a more general set of control points  $\{P_0, P_1, P_2, \dots, P_n\}$  by considering pairs of control points and using the Ferguson method for two points as developed above. It is necessary, however, to have the slopes of the tangents at each control point.

3.1.2. Development of the Catmull–Rom spline

Given  $n + 1$  control points  $\{P_0, P_1, P_2, \dots, P_n\}$ , we wish to find a curve that interpolates these control points (i.e. passes through them all), and that is local in nature (i.e. if one of the control points is moved, it only affects the curve locally). We define the curve on each segment  $P_i P_{i+1}$  by using the two control points and specifying the tangent to the curve at each control point to be  $(P_{i+1} - P_{i-1})/2$  and  $(P_{i+2} - P_i)/2$ , respectively.

Substituting these tangents into Ferguson’s method, we can obtain the matrix equation

$$P(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} P_i \\ P_{i+1} \\ \frac{P_{i+1} - P_{i-1}}{2} \\ \frac{P_{i+2} - P_i}{2} \end{bmatrix}.$$

Multiplying the two inner matrices, we can further obtain

$$P(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} M \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix},$$

where

$$M = \frac{1}{2} \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix}.$$

This matrix representation actually defines a cubic curve, which represents the portion of the total curve between the two successive control points. It can be applied to all segments of the curve except for the first and the last segments where  $P'_0$  and  $P'_n$  must be defined by other method.

3.2. Why Catmull–Rom splines?

According to the structure of the Catmull–Rom splines described above, we can obtain the following attractive properties, which are useful for shape representation and analysis. They are:

- (i) The splines passing through the control points. In our case, the boundary edges of lane can be regarded as the spline model’s control points.
- (ii) Feasibility of forming arbitrary shapes compared with the second or the third polynomials.
- (iii) Smoothness and continuity, which allows any curve to be constituted by a concatenation of curve segments yet be treated as a single unit.
- (iv) Decoupling of the  $x$  and  $y$  coordinates, which makes each coordinate have its own parametric representation.
- (v) Local controllability, which implies that local changes in shape are confined to the Catmull–Rom spline parameters local to that change.

3.3. Using Catmull–Rom spline to describe lane markings or boundaries

In the general situation (straight, turn left and turn right lane), we found that using two sets of three control points (lane left  $(P_{L0}, P_{L1}, P_{L2})$  and

lane right ( $P_{R0}, P_{R1}, P_{R2}$ ) is efficient to form two Catmull–Rom splines to approach the left- and the right-hand sides of lane boundaries (or markings). The two-spline-joint-point is called vanishing point, which is located at the horizon of the image. Although the Catmull–Rom spline interpolates all but the first and the last control points, it can be cheated by setting the first two control points equal and the last two control points equal. The Catmull–Rom spline implemented to real lane image is shown in Fig. 2.

Using spline to describe the lane model has a particular advantage, such as it only needs to increase the number of the control points when employed to model the complicated lane shapes, i.e. *S* shape and the road intersection *Y* shape. While, for the other lane models, i.e. straight and parabolic models, they can only describe the simple lane structures and thus are not flexible.

### 3.4. Control point search area

Fig. 3 shows the road shapes in both the image plane and the ground plane. After estimating the left-hand side of lane model by  $(P_{L0}, P_{L1}, P_{L2})$ , it is possible to reduce the searching area of the corresponding control point  $P_{R1}$  in the right-hand side of lane model by using parallel-line property in the ground plane.

The derivation of the slope of line  $P_{L1}P_{R1}$  in the image plane is more complicated, it has the form

$$k_{l_1, r_1} = \frac{(r_{l_1} - hz)(r_{l_1}k_{l_1} - r_{l_1} + hz)}{c_{l_1}^2k_{l_1} + k_{l_1}(\lambda^2 + hz^2) - c_{l_1}(r_{l_1} - hz)}, \quad (1)$$

where  $(r_{l_1}, c_{l_1})$  is the coordinate of point  $P_{L1}$  in the image plane,  $k_{l_1}$  the slope of the tangent of point  $P_{L1}$ ,  $\lambda$  the focal length and  $hz$  is the horizon in the image plane. Eq. (1) describes the possible location of point  $P_{R1}$  in the image plane.

Gray areas in Fig. 4 shows the possible location of point  $P_{R1}$  in terms of both the position of point  $P_{L1}$  and its tangent slope.

## 4. Maximum likelihood approach to lane detection

### 4.1. Vanishing point (line) detection

As mentioned above, two-spline-joint-point is located on the horizon of the image. Vanishing points are defined as the points at which the projections of parallel lines intersect. Vanishing points are located on the vanishing line.

If we assume the vanishing line is horizontal in image plane (it is reasonable in our case), the vertical coordinate of vanishing line can be approximated by the vertical coordinates of the vanishing points. Many approaches have been developed for detecting vanishing points. But, the methods based on Gaussian sphere (Magee and Aggarwal, 1984), hierarchical Hough transform (Quan and Mohr, 1989), statistical estimation (Toi

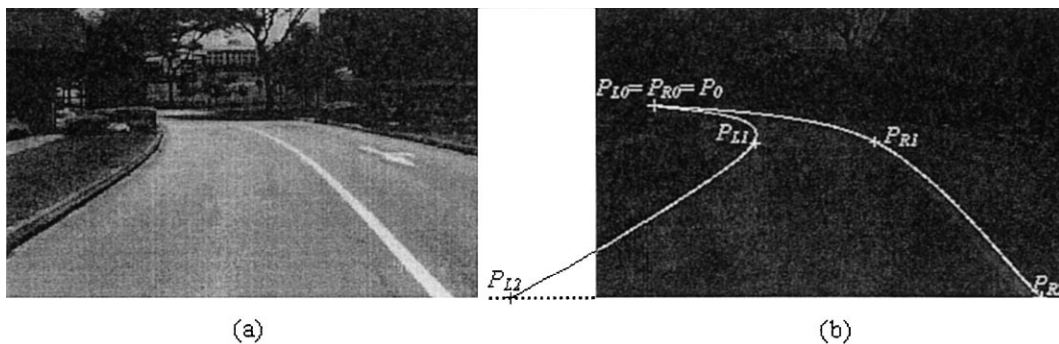


Fig. 2. Estimation of lane markings by Catmull–Rom splines: (a) Original lane image. (b) The lane markings are represented by Catmull–Rom splines, where  $(P_{L0}, P_{L1}, P_{L2})$  and  $(P_{R0}, P_{R1}, P_{R2})$  are two sets of the control points, respectively, for the left- and the right-hand sides of lane markings. Notice  $P_{L0}$  and  $P_{R0}$  are the same control points, which are regarded as the vanishing point.

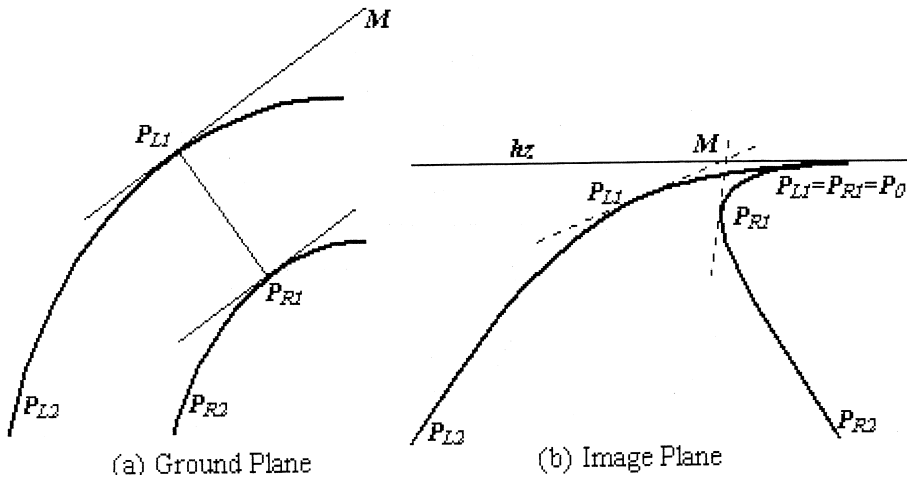


Fig. 3. The lane models, respectively, in the ground plane and the image plane.

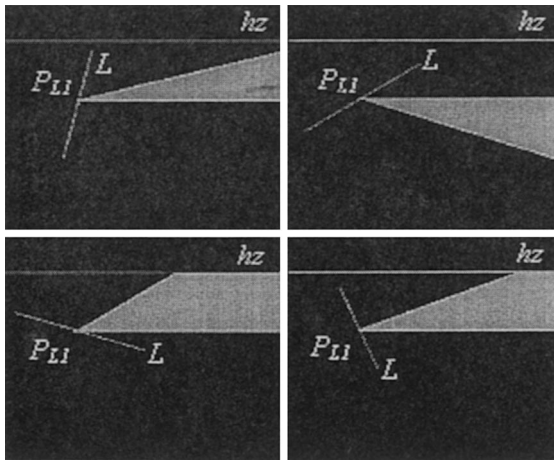


Fig. 4. Possible locations of point  $P_{R1}$ . Here  $hz$  is the horizon of the image plane, and  $L$  is the tangent at point  $P_{L1}$ .



Fig. 5. Line support region of Fig. 2(a).

et al., 1993), and line clustering (Mclean and Kotturi, 1995) are not always better than the method based on the *direct* Hough transform implementation. In this paper, we developed a method of detecting the vanishing point based on Hough transform by line-length voting.

The detection of straight line features is a prerequisite for all methods on vanishing point detection. Lines are formed by first grouping the connected regions of pixels which share similar gradient orientations, and then computing the line

equations which fit the connected regions just grouped. See Fig. 5 for example. This method is similar to the one described in (Burns et al., 1986), but with the exception that the line equations are obtained by the gradient magnitude of each grouped pixel.

The gradient magnitude and orientation at each pixel  $(i, j)$  are computed by using the following two  $2 \times 2$  gradient masks: Mask

$$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$$

is used to estimate  $G_V(i, j)$ , while mask

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$$

is used to estimate  $G_H(i, j)$ . Here  $G_V(i, j)$  and  $G_H(i, j)$  are respectively the vertical and the horizontal components of the gradient obtained from the masks applied at pixel  $(i, j)$ .

The way, the gradient orientation at pixel  $(i, j)$  can be computed by

$$\phi = \tan^{-1} G_V(i, j) / G_H(i, j).$$

In order to estimate the line support region (LSR), the gradient-orientation image was segmented by using overlapping partitions and the percentage of the total number of pixels voting for it (Burns et al., 1986). The regions selected are those that have a majority, i.e., the support is greater than 50%. It is shown in Fig. 5.

The covariance matrix for each LSR is calculated by

$$C = E \left[ (x - \bar{x})(x - \bar{x})^T \right],$$

where  $x$  is the feature vector

$$x = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

and  $\bar{x}$  is the mean

$$\bar{x} = \frac{1}{N} \begin{bmatrix} \sum x_i \\ \sum y_i \end{bmatrix}.$$

The covariance matrix of the LSR describes the spatial distribution of the data in the image plane. The ratio of eigenvalues ( $\lambda_1/\lambda_2$ ) can be used to determine the overall elongation of the LSR and is thus a good indicator of its overall suitability for forming the basis of a line equation. A line equation can be represented by a *point* and a *slope*. The *slope* of the line can be specified by the eigenvector associated with the larger magnitude eigenvalue. And, a *point* on the line can be chosen as the centroid of mass of the LSR. Thus, a line equation can be formed by the *point* and the *slope* determined above. Fig. 6 shows the straight lines detected from Fig. 5.

Each detected line was voted by its length in the Hough space. The Hough space size can be defined by an approximate range of the vanishing line possibly appearing in image plane. Of course, small Hough space size would reduce searching

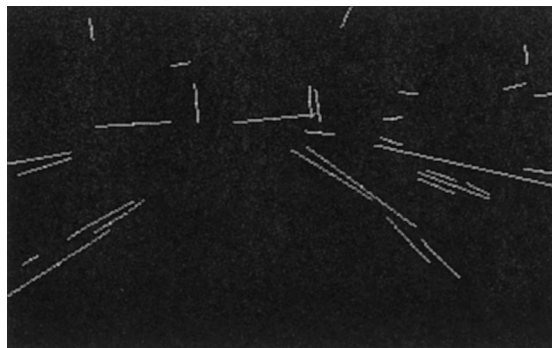


Fig. 6. The straight lines detected from Fig. 5.

time and saving memory. The vanishing line detected from Fig. 2(a) is shown in Fig. 7(c).

Applying this vanishing-line detection method to other images, we can get the results shown in Fig. 8.

#### 4.2. Edge detection

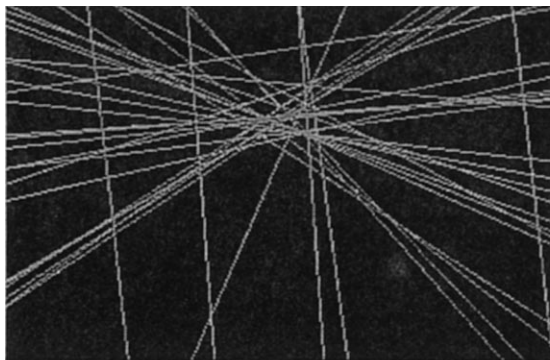
Canny edge detector is employed here to locate the position of pixels where significant edges exist. After applying the Canny edge detector to a lane image, two images that respectively denote the edge pixels and the gradient orientations can be obtained. Fig. 9 shows these two images, which are calculated from Fig. 2(a). In this paper, we have chosen  $\sigma = 1$  and a  $9 \times 1$  mask for Gaussian convolution in both  $X$  and  $Y$  directions, and the high and the low gradient thresholds for edge detector are respectively set to 90% and 40% of the maximum.

#### 4.3. Likelihood

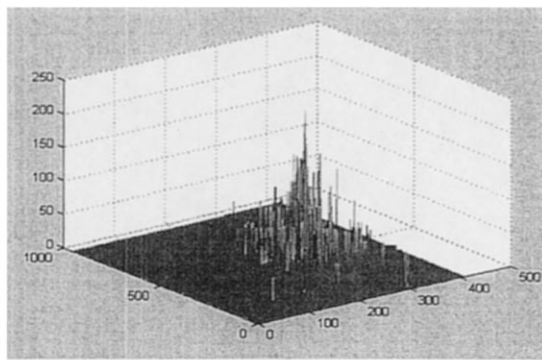
The likelihood specifies the probability of seeing the real lanes in the input road image, when given a lane model at a specific position, orientation and scale. It is measurement of the similarity between the lane model and the actual lane present in the image. The likelihood we propose here only uses the edge information in the input image.

##### 4.3.1. Computing potential edge field image

The lane model can be positioned to the actual lane in the input image by using a directional edge



(a) Straight lines in Hough space



(b) Hough space accumulator voting



(c) The detected vanishing line

Fig. 7. The demonstration of vanishing point detection.

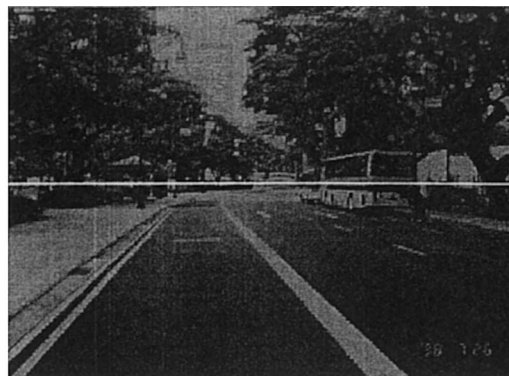
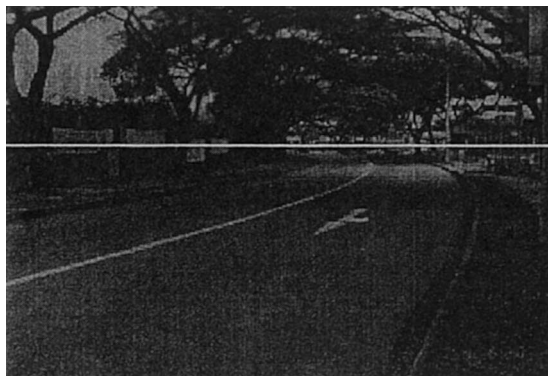
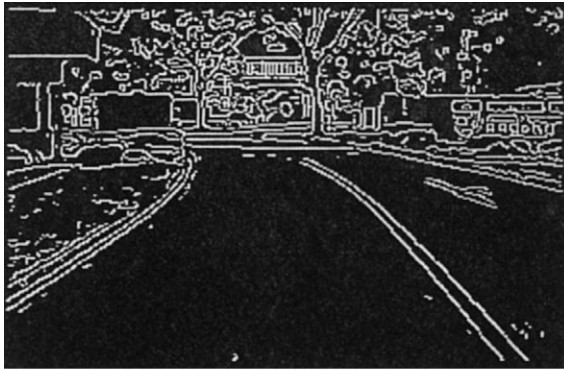
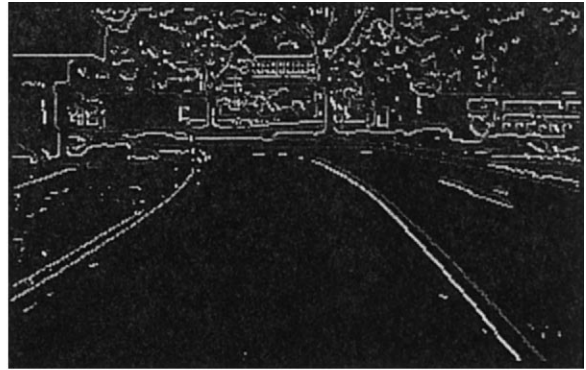


Fig. 8. Vanishing-point detection on other two images.



(a) Edge image



(b) Gradient orientation

Fig. 9. Canny edge detector results on Fig. 2(a).

potential field. The positions and directions of the edges in the input image determine this field. For a pixel  $(c, r)$  in the input image, its edge potential can be defined as

$$P(c, r) = \exp\left(-\frac{\delta c^2 + \delta r^2}{2\sigma^2}\right), \quad (2)$$

where  $(\delta c, \delta r)$  is the displacement to the nearest edge point in the image, and  $\sigma$  controls the smoothness of the potential field. Also, the potential orientation can be generated at the same time. It is simply equal to the nearest edge point's orientation, if it is in the edge point's potential field. In this paper, we use  $\sigma = 3$  to calculate the potential edge and orientation fields. Fig. 10 shows the potential edge and orientation images.

#### 4.3.2. Calculating likelihood

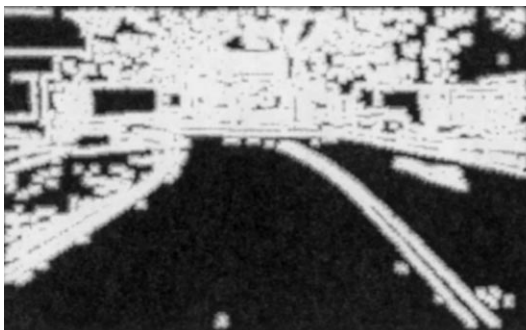
We modify the edge potential by adding to it a directional component. This new edge potential induces an energy function that relates a lane model to the real edges in the lane image.

The likelihood function is thus defined by

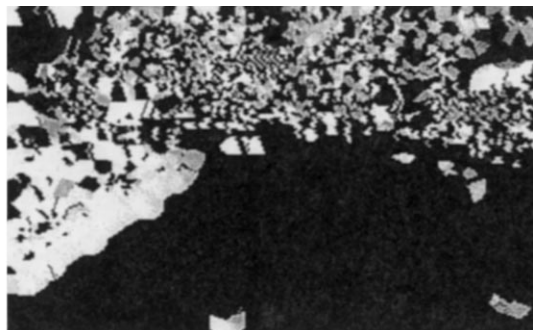
$$L = \frac{1}{n_T} \sum (P(c, r) \cdot |\cos(\beta(c, r))|), \quad (3)$$

where  $n_T$  is the number of pixels on the lane model, and  $\beta(c, r)$  is the angle between the tangent of the nearest edge and the tangential direction of the lane model at  $(c, r)$ .

This definition requires that the lane model agrees with the image edges not only in position, but also in the tangential direction. This requirement



(a) Potential edge field



(b) Potential orientation field

Fig. 10. Potential edge and orientation fields, where  $\sigma = 3$ .

is particularly useful in the presence of noisy edges. The higher this likelihood, the better the lane model matches the edges in the input image.

#### 4.4. Searching control points in edge image for lane model

In our method, we need two sets of three control points  $(P_{L0}, P_{L1}, P_{L2})$  and  $(P_{R0}, P_{R1}, P_{R1})$  to build the lane model by Catmull–Rom splines. We assume the ground is flat and the horizon is at row =  $hz$ , which can be determined by our vanishing-line detection method described above. In order to reduce search area, we define  $P_{L0}$  and  $P_{R0}$  as the same points at  $hz$ , namely the vanishing point. As shown in Fig. 11, the vanishing line is a line passing through the vanishing point.

Beginning from row  $hz$ , we search downwards to obtain an edge point,  $P_{L1}$ . The left-hand side of lane model can be constructed by using the position and orientation of point  $P_{L1}$ . We can also get the end point  $P_{L2}$ , which is the intersection of the lane model with the bottom line of image. The tangent at  $P_{L1}$  intersects with the row  $hz$  at  $M_{L1}$ . Then we calculate the likelihood  $L_L$  for left lane model. If  $L_L$  is bigger than a threshold, then update the record; otherwise, restart the searching for  $P_{L1}$ .

Next, we can search for the edge point  $P_{R1}$ . The search area is determined by Eq. (3). The tangent at the point  $P_{R1}$  intersects with the row  $hz$  at  $M_{R1}$ . The horizontal distance between  $M_{L1}$  and  $M_{R1}$  should be less than a threshold  $M_{d1}$ . Then the

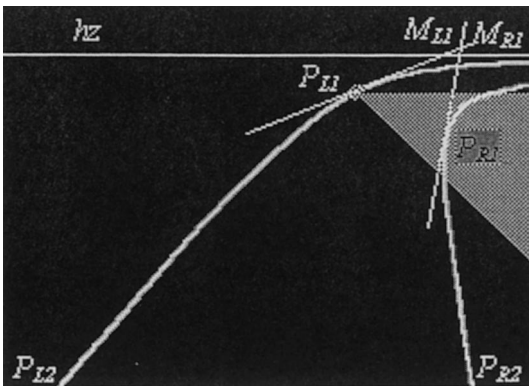


Fig. 11. Search area for the point  $P_{R1}$  when the point  $P_{L1}$  known.

likelihood  $L_R$  for the right lane model can be calculated. If  $L_R$  is bigger than a threshold, then update the record; otherwise restart the searching for  $P_{R1}$ .

Thus,  $P_{L1}$  and  $P_{R1}$ , which have a maximum likelihood, are able to construct both the sides of lane model.

#### 4.5. Speeding up the control point searching by multi-resolution strategy

We have employed a multi-resolution strategy to achieve an accurate solution rapidly. At the coarse stage, a  $M \times N$  input image is reduced to a  $(1/2)M \times (1/2)N$  image by Laplacian pyramid (Burt and Adelson, 1983). In this low resolution of original image, we can roughly and efficiently locate the global optima without regarding the accuracy. While, in the final stage, the fine-level matching is initialized by the best candidate screened from the coarse stage. Also, a smaller search area and a finer step size are used to obtain the better matching in the original image. This coarse-to-fine matching offers us an acceptable solution at an affordable computational cost, and thus speeds up the process of lane detection. Fig. 12 shows the multi-resolution approach.

## 5. Results

The proposed algorithm has successfully applied to real-time application. The results have shown an achievement of a rated execution speed of 4–5 frames/s. The system consists of a Pentium II 300 MHz computer with 32 MB of RAM and a frame grabber card. The programming is done in Visual C++ programming software. Fig. 13 shows some of our experimental results on lane boundary detection, where the detected lane boundaries are superimposed onto the original images with the halved intensities. These images contain both the paved and the unpaved roads (or lanes) which are either marked or unmarked. The proposed method is robust to noise, shadows, illumination variations, and different road conditions.

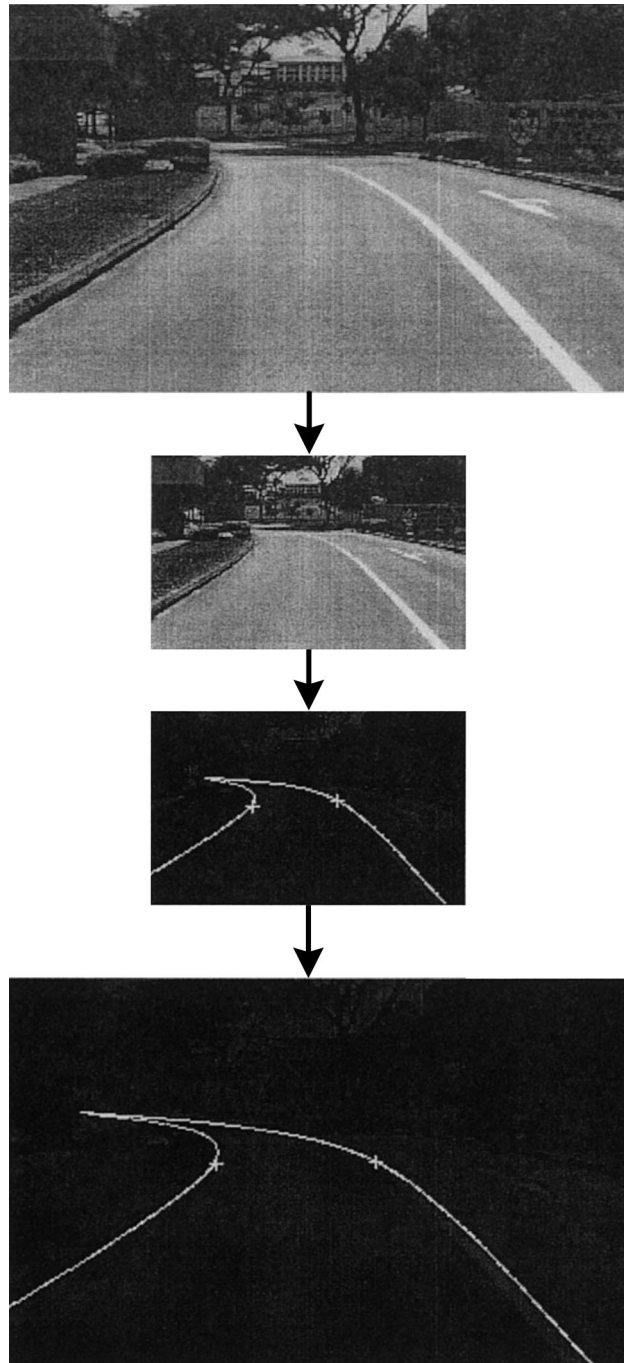


Fig. 12. Multi-resolution algorithm for determining the control points.

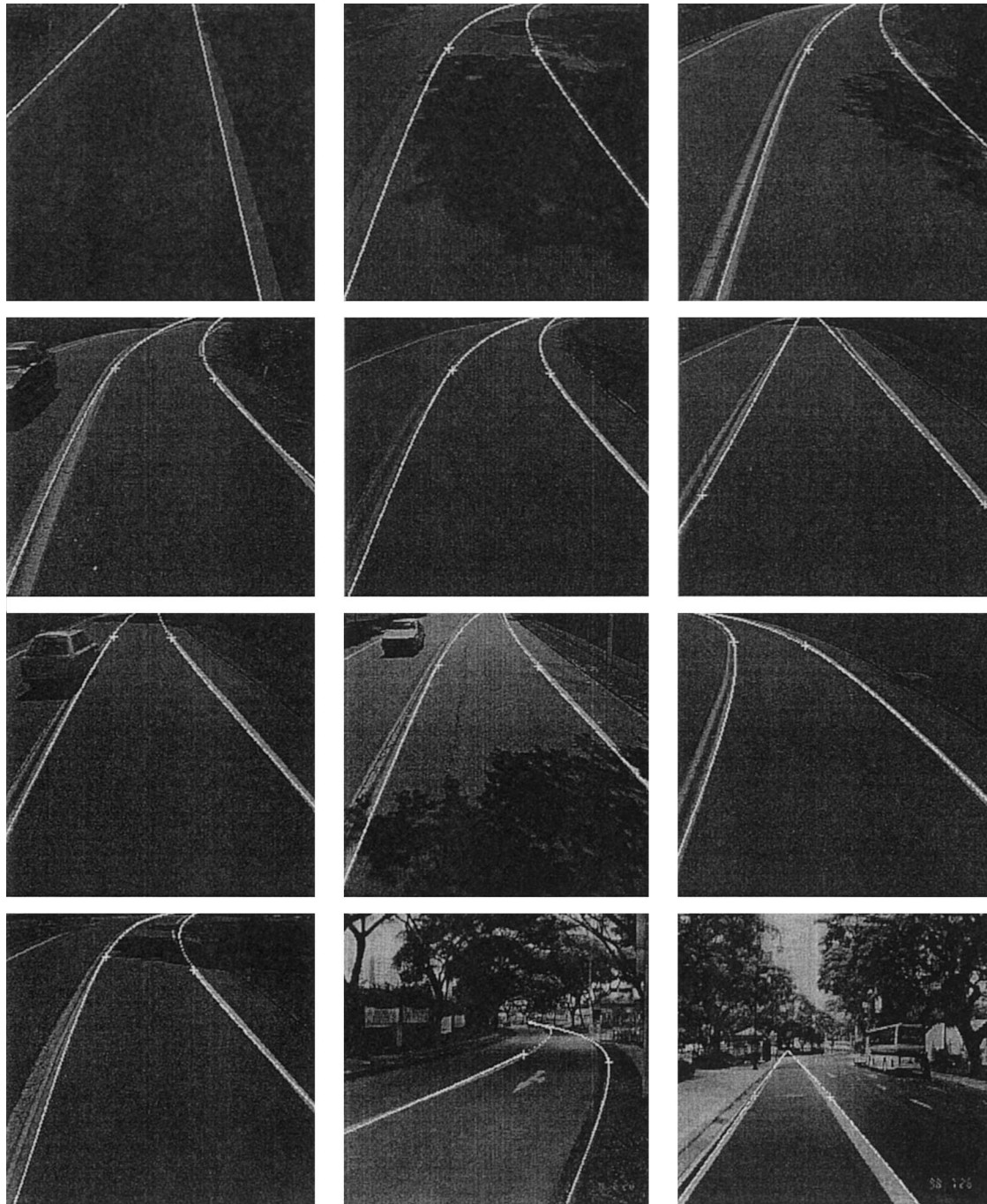


Fig. 13. Some results of lane detection.

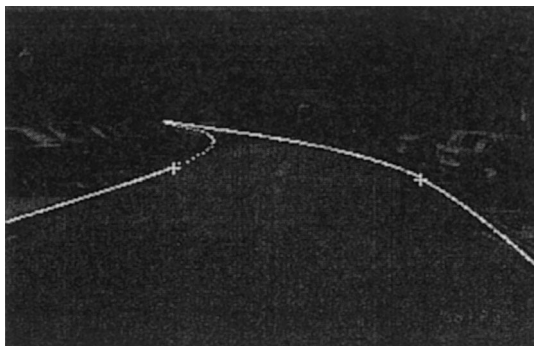


Fig. 14. A fail case in lane detection.

A fail case is shown in Fig. 14. Since there are no well-painted lanes and clear boundaries in this road image, thus not enough edge information can be used to accurately build a lane model.

## 6. Conclusion

We have addressed the problem of lane detection in this paper. A new Catmull–Rom spline based lane model, which describes the perspective effect of parallel lines, has been constructed for generic lane boundaries (or markings). The proposed algorithm is able to describe a wider range of lane structures compared with the other lane models, i.e. straight and parabolic models. In our algorithm, the lane detection problem has been solved by determining the sets of control points of lane model. Also, a maximum likelihood method has been used to measure the matching degree between the lane model and the real road edges. The experimental results obtained by our algorithm are good and accurate, even under the shadowy and noisy conditions.

## References

- Bertozzi, M., Broggi, A., 1998. GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Trans. Image Process.*, 62–81.
- Broggi, A., 1995a. Robust real-time lane and road detection in critical shadow conditions. In: *Proc. IEEE Internat. Sym. Comput. Vision*, Coral Gables, Florida, 19–21 November.
- Broggi, A., 1995b. A massively parallel approach to real-time vision-based road markings detection. In: Masaky, I. (Ed.), *Proc. IEEE Intelligent Vehicles'95*, pp. 84–89.
- Broggi, A., Berte, S., 1995. Vision-based road detection in automotive systems: a real-time expectation-driven approach. *J. Artificial Intelligence Res.* 3, 325–348.
- Burns, J.B., Hansen, A.R., Riseman, E.M., 1986. Extracting straight lines. *IEEE Trans. PAMI* 8, 425–455.
- Burt, P.J., Adelson, E.H., 1983. The Laplacian pyramid as a compact image code. *IEEE Trans. Commun. COM-31* (4), 532–540.
- Grimmer, D., Lakshmanan, S., 1996. A deformable template approach to detecting straight edges in radar images. *IEEE Trans. Pattern Anal. Machine Intell.* 18, 438–443.
- Jung Kang, D., Won Choi, J., So Kweon, I., 1996. Finding and tracking road lanes using line-snakes. In: *Proc. Conf. Intelligent Vehicle*, Japan, pp. 189–194.
- Kaske, A., Husson, R., Wolf, D., 1995. Chi-square fitting of deformable templates for lane boundary detection. In: *Proc. IAR Annual Meeting '95*, November, Grenoble France.
- Kaske, A., Wolf, D., Husson, R., 1997. Lane boundary detection using statistical criteria. In: *Proceedings of the International Conference on Quality by Artificial Vision, QCAV'97*, Le Creusot, France, pp. 28–30.
- Kluge, K., 1994. Extracting road curvature and orientation from image edge points without perceptual grouping into features. In: *Proc. of Intelligent Vehicles Symposium*, pp. 109–114.
- Kluge, K., Lakshmanan, S., 1995. A deformable template approach to lane detection. In: Masaky, I. (Ed.), *Proc. IEEE Intelligent Vehicle '95*, Detroit, 25–26 September, pp. 54–59.
- Lakshmanan, S., Kluge, K., 1995. Lane detection for automotive sensor. In: *ICASSP*, pp. 2955–2958.
- Li, W., Jiang, X., Wang, Y., 1998. Road recognition for navigation of an autonomous vehicle by fuzzy reasoning. In: *Fuzzy Sets and Systems*, Vol. 93/3. Elsevier, North-Holland, Amsterdam, pp. 275–280.
- Magee, M.J., Aggarwal, J.K. 1984. Determining vanishing points from perspective images. In: *Computer Vision Graphics Image Process.* 26, 256–267.
- McLean, G.F., Kotturi, D., 1995. Vanishing point detection by line clustering. *IEEE Trans. PAMI* 17 (11), 1090–1095.
- Quan, L., Mohr, R., 1989. Determining perspective structures using hierarchical Hough transform. In: *Pattern Recognition Letters* 9, 279–286.
- Serge, B., Michle, B., 1994. Road segmentation and obstacle detection by a fast watershed transform. In: *Proc. Intelligent Vehicles '94 Symposium*, pp. 296–301.
- Toi, A., Kittler, J., Petrou, M., Windeatt, T., 1993. Vanishing point detection. *Image and Vision Comput.* 11 (4), 240–245.
- Xuan, Y., Serge, B., Michel, B., 1992. Road tracking, lane segmentation and obstacle recognition by mathematical morphology. In: *Proc. Intelligent Vehicles '92 Symposium*, pp. 166–170.