Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

Image and Vision Computing 22 (2004) 269–280

www.elsevier.com/locate/imavis

Lane detection and tracking using B-Snake

Yue Wang[a], Eam Khwang Teoh[a,*], Dinggang Shen[b]

[a]*School of Electrical and Electronic Engineering, Nanyang Technological University, Black S2, Nanyang Avenue, Singapore, Singapore, 639798*
[b]*Department of Radiology, University of Pennsylvania, Philadelphia, PA 19104, USA*

Received 17 July 2002; received in revised form 26 September 2003; accepted 1 October 2003

Abstract

In this paper, we proposed a B-Snake based lane detection and tracking algorithm without any cameras' parameters. Compared with other lane models, the B-Snake based lane model is able to describe a wider range of lane structures since B-Spline can form any arbitrary shape by a set of control points. The problems of detecting both sides of lane markings (or boundaries) have been merged here as the problem of detecting the mid-line of the lane, by using the knowledge of the perspective parallel lines. Furthermore, a robust algorithm, called CHEVP, is presented for providing a good initial position for the B-Snake. Also, a minimum error method by Minimum Mean Square Error (MMSE) is proposed to determine the control points of the B-Snake model by the overall image forces on two sides of lane. Experimental results show that the proposed method is robust against noise, shadows, and illumination variations in the captured road images. It is also applicable to the marked and the unmarked roads, as well as the dash and the solid paint line roads.

Keywords: Lane detection; B-Spline; Snake; Lane model; Machine vision; Intelligent vehicle

1. Introduction

Autonomous Guided Vehicles (AGV) have found many applications in the industries. Their applications had been explored in areas, such as patient transportation in hospitals, automated warehouses and other hazardous related areas. In most applications, these AGVs have to navigate in the unstructured environments. Path findings and navigational control under these situations are usually accomplished from the images captured by camera mounted on the vehicles. These images are also interpreted 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 the unpainted roads.
- It should handle the 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.

Up to present, various vision-based lane detection algorithms have been developed. They usually utilized different lane patterns (solid or dash white painted line, etc.) or different road models (2D or 3D, straight or curve), and different techniques (Hough, template matching, neural networks, etc.). Basically, there are two classes of approaches used in lane detection: the feature-based technique and the model-based technique. The feature-based technique localizes the lanes in the road images by combining the low-level features, such as painted lines [5–10] or lane edges [1,2], etc. lane segments that are detected by traditional image segmentation. Accordingly,

* Corresponding author. Tel.: +65-6790-5393; fax: +65-6791-2687.
E-mail addresses: eekteoh@ntu.edu.sg (E.K. Teoh), s2633175g@ntu.edu.sg (Y. Wang), dgshen@rad.upenn.edu (D. Shen).

0262-8856/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2003.10.003

this technique requires the studied road having well-painted lines or strong lane edges, otherwise it will fail. Moreover, as it has the disadvantage of not imposing any global constraints on the lane edge shapes, this technique may suffer from occlusion or noise.

On the other hand, the model-based technique just uses a few parameters to represent the lanes. Assuming the shapes of lane can be presented by either straight line [11,12,13,16] or parabolic curve [3,4,14,15], the processing of detecting lanes is approached as the processing of calculating those model parameters. This way, the model-based technique is much more robust against noise and missing data, compared with the feature-based technique. To estimate the parameters of lane model, the likelihood function [3,4,11,12,16], Hough transform [13], and the chi-square fitting [14,15], etc. are applied into the lane detection. However, as the most lane models are only focused on certain shapes of road, thus they lack the flexibility to modeling the arbitrary shape of road.

Motivated by the above problems, we here present a new B-Snake based lane detection and tracking algorithm for the outdoor application of AGV. The main characters of our method are the following:

1. A novel B-Snake based lane model which describes the perspective effect of parallel lines is constructed with *dual external forces* for generic lane boundary or marking, it is able to describe a wider range of lane structures than other lane models such as straight and parabolic models. In addition, it is robust against shadows, noises, etc. due to the use of the parallel knowledge of roads on the ground plane. The lane detection problem is formulated by determining the set of lane model control points.
2. A robust algorithm called Canny/Hough Estimation of Vanishing Points (CHEVP) is presented for providing a good initial position for the B-Snake lane model. This algorithm is robust to noises, shadows, and illumination variations in the captured road images, and is also applicable to both the marked and the unmarked, dash paint line and solid paint line roads.
3. Using Gradient Vector Flow (GVF) to construct the B-Snake external force field for lane detection, a minimum error method called Minimum Mean Square Error (MMSE) that finds the correspondence between B-Snake and the real edge image is presented to determine the parameters of road model iteratively. Road tracking is carried on after successful lane detection, by a simple external force field and MMSE method, tracking is efficient and speed is fast.

Besides B-Spline, other kind splines also can be used in our lane model. Our early version of lane model used Catmull-Rom spline [24,25,26]. The different between the B-Spline and the other kind splines is the locations of the control points.

The remained structure of this paper is arranged as follows. Section 2 introduces a novel B-Spline lane model with *dual* external forces. In Section 3, the CHEVP is described for B-Snake lane model initialization. Section 4 presents a minimum error method, MMSE, to determine the parameters for lane detection and lane tracking. This section also shows some representative results of applying the proposed algorithm to various types of roads under different environments. This paper concludes in Section 5.

2. Road model

2.1. The modeling of lane boundaries

Lane model plays an important role in lane detection. The lane modeling has to make some assumptions about the road's structure in the real world in order to fully recover 3D information from the 2D static image. In this paper, we focus on constructing the 2D lane model, by assuming that the two sides of the road boundaries are parallel on the ground plane as shown in Fig. 1(a).

In addition, let us assume that the right side of road is the shifted version of the left side of road at a distance, $D = (x_r - x_l)$, along the x axis in the ground plane. Here, x_r and x_l are the x coordinates of the two correspondence points, $P_l(x_l, y)$ and $P_r(x_r, y)$, in the ground plane. After projection from the ground plane to the image plane, the horizontal distance $d = (c_r - c_l)$ between the corresponding points $p_l(c_l, r)$ and $p_r(c_r, r)$, which are the projected points of $P_l(x_l, y)$ and $P_r(x_r, y)$, is:

$$d = \frac{\lambda^2 D(r - hz)}{H(\lambda^2 + hz^2)} \tag{1}$$

where λ is the *focal length* of the lens, H is the height of the camera location, hz is the position of vanish line in the image pane, and r is the vertical coordinate used in the image plane (see Fig. 1(b) for reference).

The horizontal distance d can be represented as

$$d = k(r - hz) \tag{2}$$

where

$$k = \frac{\lambda^2 D}{H(\lambda^2 + hz^2)}$$

Let us define the mid-line of the road in the image plane as

$$L_{\text{mid}} = (c_m, r_m) \tag{3}$$

Thus the left side of the modeled road is

$$L_{\text{left}} = (c_l, r_l), \tag{4}$$

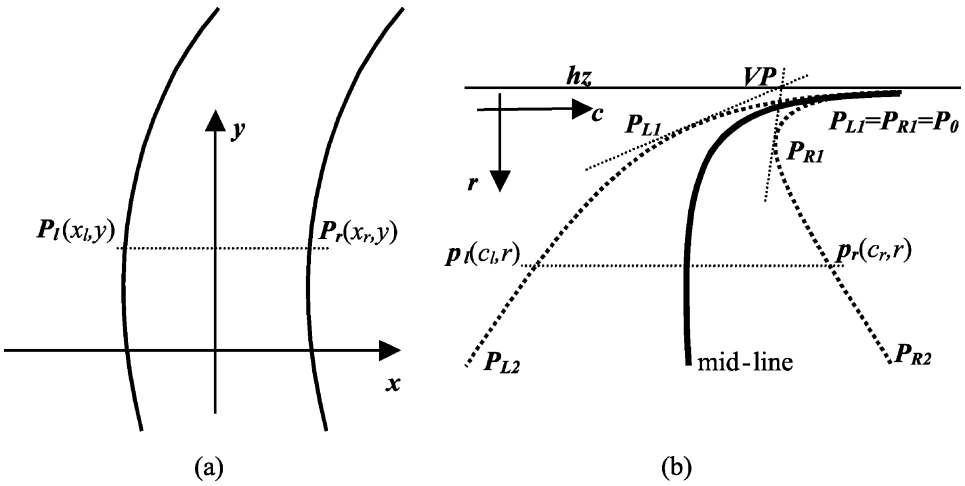


Fig. 1. Parallel lines on ground plane and image plane.

where

$$c_l = c_m - \frac{1}{2}d = c_m - \frac{1}{2}k(r_l - hz) \text{ and } r_l = r_m. \tag{5}$$

Similarly, the right side of the modeled road is

$$L_{\text{rights}} = (c_r, r_r) \tag{6}$$

where

$$c_r = c_m + \frac{1}{2}d = c_m + \frac{1}{2}k(r_r - hz) \text{ and } r_r = r_m. \tag{7}$$

From the above modeling, it is easy to observe that the problem of detecting two sides of road can be formulated as the problem of detecting the mid-line of road. In following Sections, we would show that k can be estimated directly from image data without any camera's parameters.

2.2. B-Spline snake

Snakes [17], or *active contours*, are curves defined within an image domain which can move under the influence of internal forces from the curve itself and external forces from the image data. Once internal and external forces have been defined, the snake can detect the desired object boundaries (or other object features) within an image. Snakes have been used widely in many applications, such as edge detection [17], shape modeling [18,19], segmentation [20,21], and motion tracking [20,22].

A more economical realization of snake can be reached by using far fewer state variables by cubic B-Splines. The B-Splines are piecewise polynomial functions that provide local approximations to contours using a small number of parameters (control points). It can represent curves by four or more state variables (control points). As required, the represented curves may be open or closed. The flexibility of the curve increases as more control points are added. Each additional control point either allows one more inflection in the curve or, when multiple knots are used [23], reduces continuity at one point.

2.2.1. Uniform cubic B-splines

An open cubic B-Spline, with $n+1$ control points $\{Q_0, Q_1, \ldots, Q_n\}$, consists of $(n-2)$ connected curve segments, $g_i(s) = (r_i(s), c_i(s))$, $i = 1, 2, \ldots, (n-2)$. It is C^2 continuous and has both its continuous slopes and curvatures. Each curve segment is a linear combination of four control points by the parameter s, where s is normalized between 0 and 1 ($0 \le s \le 1$). It can be expressed as:

$$g(s) = \sum_i M_i(s) Q_i \tag{8}$$

where $M_i(s)$ are the spline basis functions.

According the B-Spline property, the B-Spline would pass through the control point by triple the corresponding control points.

2.3. Using B-Snake to describe lane markings (or boundaries)

We use a set of control points to describe the mid-line of the road by B-Spline, and a additional parameter k (as described in Section 2.1) to determine the left and the right sides of road model. In order to make B-Splines pass through the first and the last control points, we set the first three control points equal and the last three control points equal.

The mid-line of road model can be expressed by a B-Spline as

$$L_{\text{mid}} = (c_m, r_m) = M_R(s) \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix}, \tag{9}$$

$i = -1, 0, 1, 2, \ldots, n.$

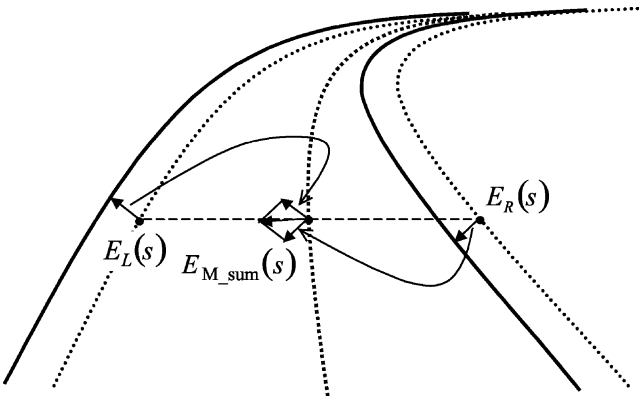


Fig. 2. Example on the external forces of the mid-line of lane model. Solid lines are the real road edges, while dash lines are the lane model.

The mid-line of lane model can be deformed by the external forces $E_{M\_sum}(s)$, which is the sum of the *dual external forces* calculated from both the left and the right sides of lane model, $E_L(s)$ and $E_R(s)$.

$$E_{M\_sum}(s) = E_L(s) + E_R(s) \tag{10}$$

In Fig. 2, $E_{M\_sum}(s)$ would push the lane model to the left.

Also, the difference of horizontal components of $E_L(s)$ and $E_R(s)$, denoted as $E^c_{M\_dif}(s)$, would lead to adjustment of the parameter k.

$$E^c_{M\_dif}(s) = E^c_L(s) - E^c_R(s). \tag{11}$$

Fig. 3 shows how $E^c_{M\_dif}(s)$ would lead to the adjustment of the parameter k, increasing (Fig. 3(b)) or decreasing (Fig. 3(a)). In Fig. 3(a), the left side of the estimated lane model is located at the left of the real road's left boundary, while the right side of the estimated lane model is located at the right of the real road's right boundary. As shown in Fig. 3(a), $E^c_{M\_dif}(s)$ points to the right, we can define it as leading to decreasing k. On the contrary, in Fig. 3(b), the left side of the estimated lane model is located at the right of the real road's left boundary; and the right side of the estimated lane model is located at the left of the real road's right boundary. This way, $E^c_{M\_dif}(s)$ points to the left, which will lead to increasing k.

Compare to other lane models, there are few advantages for B-Snake lane model with *dual external forces*:

1. B-Snake can describe much wider range of lane shapes while retains compact representation, since B-Spline has local controllability and can form *arbitrary* shape. For example, it can describe more complex road shape, such as 'S' or sharp corner turn, just by increasing the number of control points. Other lane model cannot describe those complex shapes, since they use only a single polynomial.
2. With *dual external forces*, B-Snake model would be robust against shadows, noises, occasional missing and false markings, etc. since the sampling locations for calculating the *dual external forces* are combined with the knowledge of parallel lines on the ground plane, the external forces for deformation of B-Snake is not depended on one, but both sides of lane model at a time.
3. The processing time will be reduced since two deformation problems for both sides of lane have been formulated to one deformation problem.
4. This B-Snake lane model is particular suitable for lane tracking application, since the parameters of lane model for the current frame is usually similar to those in the previous frame, i.e. the movements of the control points are smaller. On the contrary, for other lane models such as the second order polynomial lane model, when the road shape has a small change, it may cause a large change in the parameters of model.

For most lanes, we found that using 3 control points is efficient to describe their shapes. Therefore, we select 3 control points in this paper for constructing the lane model. Fig. 4(a) shows a lane model formed by a set of 3 control points, Q_0, Q_1 and Q_2 shapes (Q_0 and Q_2 are triple, so

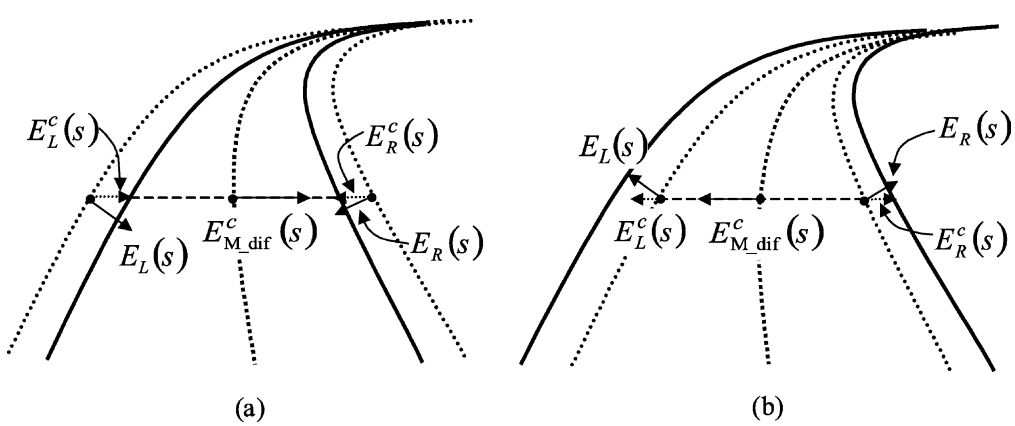
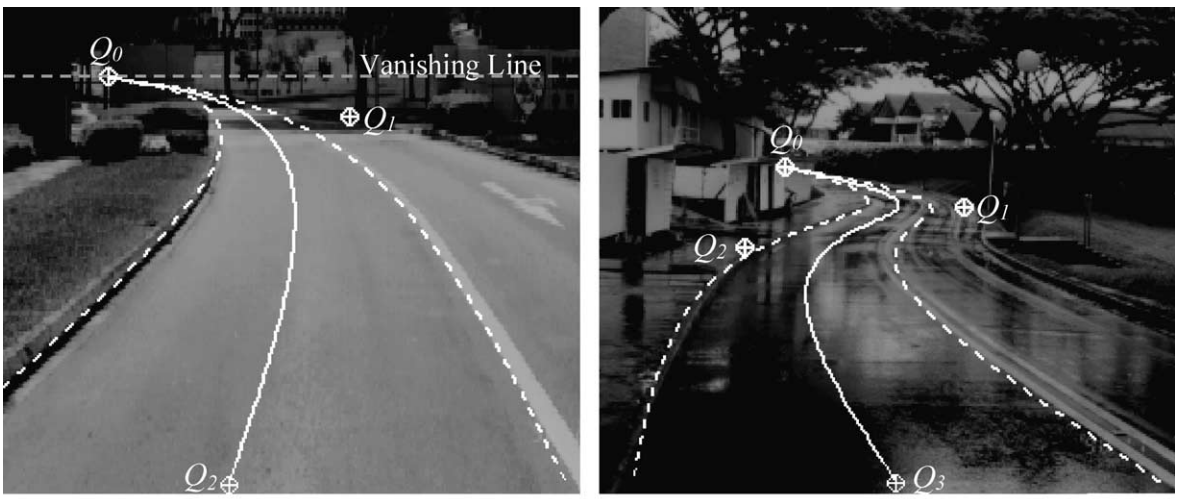


Fig. 3. Example on the adjustment of the parameter k. Solid lines are the real road edges, while dash lines are the lane model. (a) The case that would lead to decreasing k. (b) The case that would lead to increasing k.



(a) Using 3 control points (b) Using 4 control points

Fig. 4. B-Snake based lane model.

we actually compute a curve of two segments from the sequence of seven control points $Q_0, Q_0, Q_0, Q_1, Q_2, Q_2, Q_2$). Since we only concern the road in the camera's field of view, we limit the location of the first control point Q_0 in the vanishing line if the vanishing line is in the captured image. In the case that the vanishing line is above the top of image, Q_0 will be limited in the top row of the captured image. The end control point Q_2 is limited to the bottom row of image.

For the case that three control points are not sufficient to describe the shape of road, our structure-adaptive B-Snake [28] model can be implemented for auto-increasing the number of control point to adapt the shape of road. The more control points are used, the more complex shape can be formed. Fig. 4(b) gives an example of using four control points to describe a 'S' shape road.

3. Initialization of B-Snake lane model: CHEVP algorithm

Some lane detection algorithms required the operator to provide the initial estimate of the road location, while others required the *specific* road structure scene (such as straight road) as the first road image. These requirements on the road initializations are clumsy for the automatic road detection task. Therefore, automatic initialization technique, able to extract the location of any type of the lane shapes, is important and necessary.

3.1. Description of the CHEVP algorithm

The CHEVP (Canny/Hough Estimation of Vanishing Points) algorithm has been developed to meet these requirements. The road is assumed to have two parallel boundaries on the ground, and in the short horizontal band of image, the road is approximately straight. As a result of the perspective projection, the road boundaries in the image plane should intersect at a shared vanishing point on the horizon. Below we briefly introduce this algorithm, for full details please visit: www.ntu.edu.sg/home5/ps2633175g/chevp.htm. There are following five processing stages in CHEVP algorithm:

1. Edge pixel extraction by Canny edge detection. Canny edge detection is employed to obtain edge map.
2. Straight lines detection by hough transform.

The detected edge points are used to vote for possible lines in the space of line parameters. The image is here partitioned into a small number of horizontal sections, i.e. five as shown in Fig. 5(b), in order to accommodate the change in road vanishing point *due to* the bend of the road. The height of image section is gradually reduced as moving to the upper part of image. Notice that, each image section has its own space of line parameters, and edge points in each image section vote separately for possible straight lines in that section. By suitably thresholding the normalized accumulator spaces, line segments can be finally detected for each image section (Fig. 5(c)).

3. Horizon and vanishing points detection.

The detected straight lines of each image section are paired, and the intersections of any pair of lines vote for vanishing points on another Hough space. The votes are weighted by the sum of the paired lines' normalized accumulator values produced in the Step 2. This process is repeated for each image section separately, but vote in the same Hough space. The votes on each column of the Hough space are summed for detecting possible vanishing line. The row with the maximum support is chosen as the horizon (or vanishing line) in the image plane. Fig. 5(d) shows the detected vanishing line.

For each image section, its vanish point can be determined as the point around the horizon and with the strongest support. Fig. 6(a) shows both the vanishing point

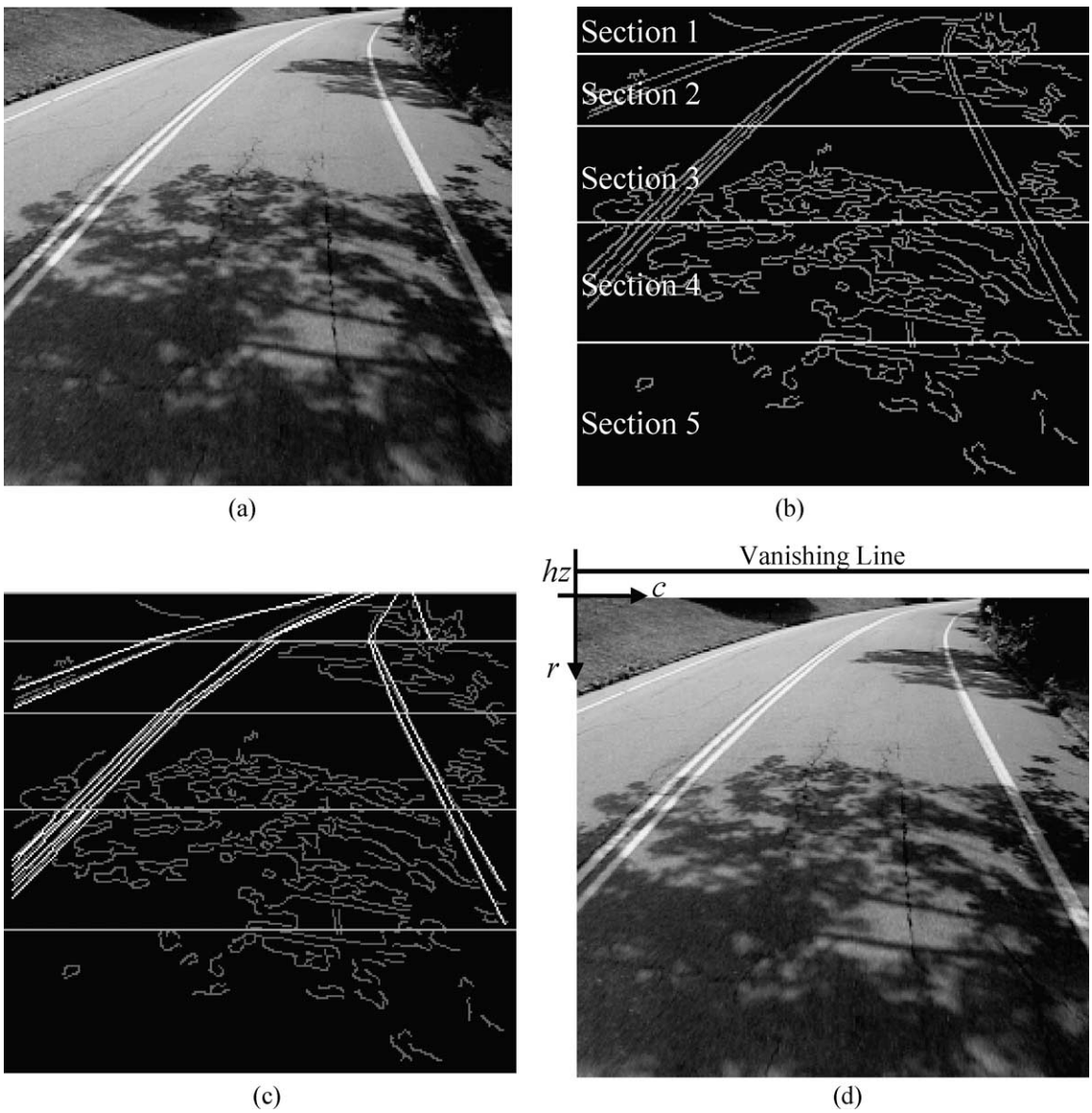


Fig. 5. Detection of straight lines and vanishing line.

of image Section 2 and a pair of lines voting for it. The detected vanishing points for all image sections are shown in Fig. 6(b). Notice that, no vanishing point exists for the image Section 5, since no lines can be detected in this image section.

4. Estimate the mid-line of road and the parameter k by the detected road lines.

The lines voting for vanishing point are assumed to be road lines in each image section. From the bottom image section upward, select the two detected road lines from the left and the right sides, which are closest to the mid column of that section. If these two road lines do not exist in the current image section, then the procedure will be repeated in the next higher image section *until* the required road lines are obtained. Fig. 7(a) shows the two lines L_1 and L_2 chosen in image Section 4, since no line exist in image Section 5. Then, connect the vanishing point ($vp4$) of this image Section 4 and the middle point (P_{m4}) of the two points (P_{l4} and P_{r4}) which are the intersection points of the two road lines L_1 and L_2 at the bottom row of that section.

The line passing through points $vp4$ and P_{m4} intersects at the bottom of Section 3 at P_{m3}. Then the parameter k can be estimated by:

$$k = \frac{c_{\text{right}} - c_{\text{left}}}{r_{\text{mid}} - hz}. \tag{12}$$

where hz is the vertical coordinate of vanishing line. In the case of Fig. 7(a),

$$c_{\text{left}} = c_{l4} \qquad c_{\text{right}} = c_{r4} \qquad r_{\text{mids}} = r_{l4} = r_{r4} \tag{13}$$

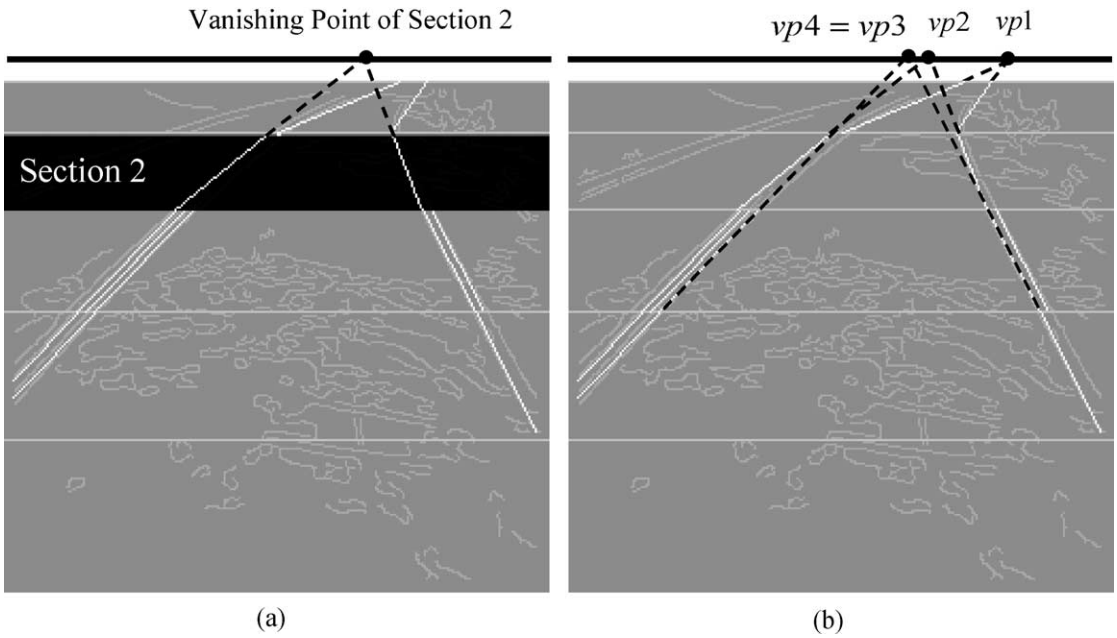


Fig. 6. Vanishing points detection. (a) The vanishing point of the image Section 2 and the lines which vote for it. (b) The detected vanishing points for all image sections.

Down from image Section 4, since in image Section 5 no vanishing point has been detected, we assume this section's vanishing point follows the vanishing point $vp4$ of Section 4. Extend the line (passing through $vp4$ and P_{m4}) and joint at the bottom of image Section 5 at P_{m5}. Similarly, in image Section 3 we can detect vanishing point $vp3$ ($vp3$ is the same point as $vp4$ in Fig. 7(b)). The line ($vp3 - P_{m3}$) intersects at the bottom of image Section 2 at P_{m2}. (In the case the vanishing point $vp3$ cannot be detected, we assume $vp3$ follows to $vp4$.) Image Section 2 has detected a vanishing point $vp2$, so just connect $vp2$ and P_{m2} and intersects at the bottom of Section 1 at P_{m1}. Image Section 1 also has a detected vanishing point $vp1$, then the line ($vp1 - P_{m1}$) intersects at the top of Section 1 at P_{m0}. Fig. 7(b) shows the whole mid-line. After constructing the mid-line of road, the both sides of road boundaries can be constructed based on the mid-line of road and the estimated k.

5. Initial the control points of the lane model to approach the mid-line detected by last step.

We first choose P_{m0} and P_{m5}, respectively, as the start control point Q_0 and the end control point Q_2 for lane model (see Fig. 8(a)). We know if knots of B-Spline are known, then the according control points can be gotten. The selection of the knot P_1 depends on the values of

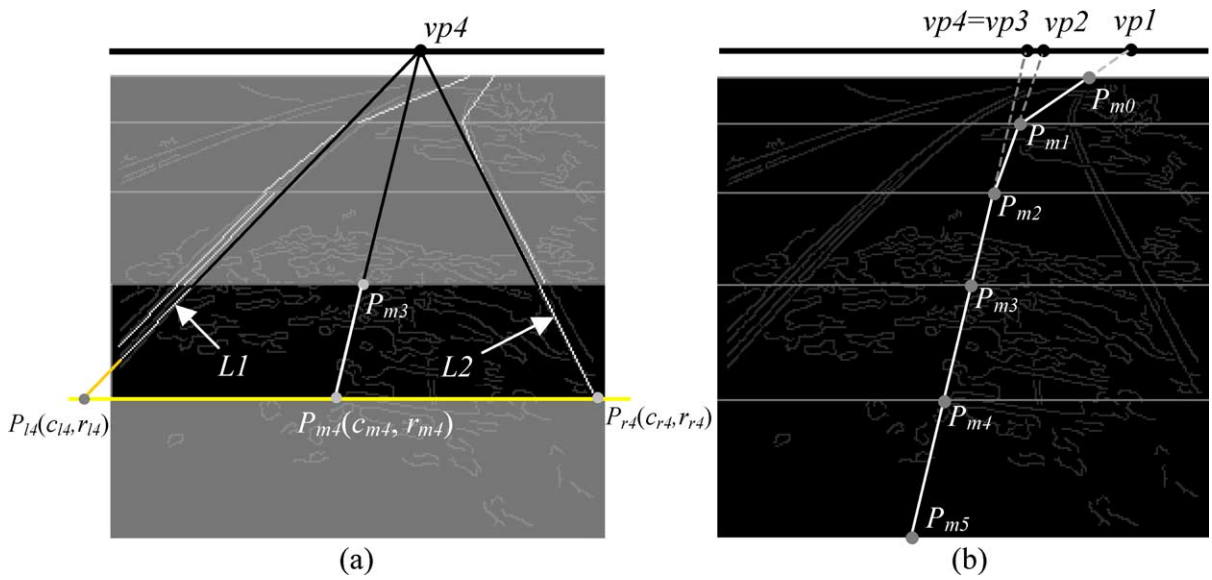


Fig. 7. Estimate mid-line of road. (a) Two lines in image Section 4 are chosen from both sides of road boundaries to estimate k. (b) Estimated mid-line of road.

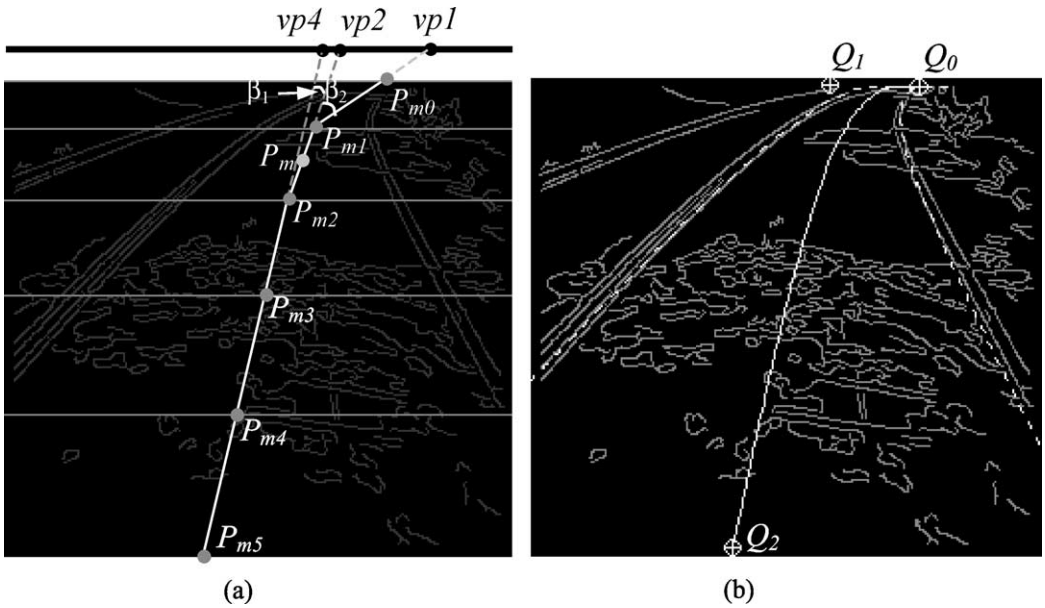


Fig. 8. Initialize the lane model to approach the mid-line detected. (a) Choose control points for lane model. (b) Final result of initialization for lane model.

angles β_1 and β_2 defined in Fig. 8. If angles β_1 and β_2 are not equal to zero, we choose P_{m} as the knot for Q_1. That is $P_1 = P_{\mathrm{m}}$, where P_{m} is the middle point of P_{m1} and P_{m2}. If $\beta 1 = 0$ and $\beta 2 \neq 0$, we choose P_{m1} as the knot P_1 (for Q_1). If $\{\beta 1 \neq 0 \text{ and } \beta 2 = 0\}$ or $\{\beta 1 = 0$ and $\beta 2 = 0\}$, we choose P_{m2} as the knot P_1 (for Q_1). Therefore, the control point Q_1 can be calculeted by

$$Q_1 = \frac{3}{2}P_1 - \frac{1}{4}(Q_0 + Q_2). \tag{14}$$

The estimated B-Spline is shown in Fig. 8(b). Notice, k is not changed here, just taken the same values from the last step.

3.2. Experiment results on testing CHEVP algorithm

The CHEVP algorithm has been applied to real road images grabbed by a camera at different locations and at different times. These images include straight and curve roads with painted or unpainted, solid or dash lines, and

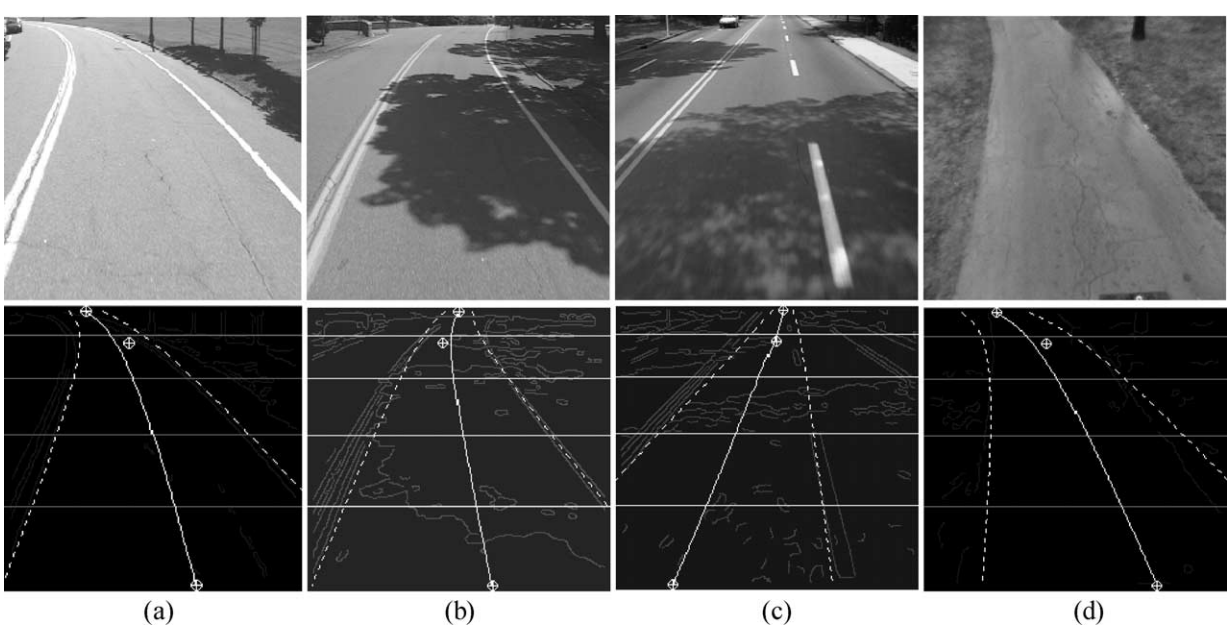


Fig. 9. Some results of CHEVP.

shadow. Some results are shown in Fig. 9. Fig. 9(a) shows the result of applying CHEVP to the images with the curved road. CHEVP locates the double paint lints on the left side of the lane, and the white stripe on the right side of the road, especially it remains high accuacy in the near place to camera. Notice that the location of the detected mid-line is not totally accurate in the top of image *due to* the curve lines being hard to be detected by straight line parameters in Hough space. However, it is well within the tolerance necessary to use them to make initial predictions for the lane model detection. Fig. 9(b) shows another examples of applying the CHEVP algorithm to curve-road image with strong shadow edges. It can be observed that, even the white paint line on the right passes through noisy shadow edges, the Hough transform and the shared vanishing point constraint allow CHEVP algorithm to successfully locate the feature position. On the contrary, edge tracking algorithms would become confused by the shadow edges, which would offer multiple possible continuations. Fig. 9(c) shows the results of the CHEVP algorithm on the multi-lane image taken on a divided highway with strong shadows. As CHEVP algorithm is designed to choose the lane which is closest to the centre column of the image(see Step 4 of CHEVP), hence, even under strong shadows, CHEVP algorithm successfully locates the solid white stripe on the left side of the lane, as well as the broken white stripe on the right side of the lane where the vehicle is located. An example applying CHEVP algorithm to the unpainted lane is shown in Fig. 9(d). The road image was taken on a 1-lane road for bicycle. It can be seen that the road is wet after raining. The CHEVP algorithm can only detect the vanishing points for Sections 1 and 2. However, the estimated mid-line of road is still acceptable. As we can observe in Fig. 9, combined with global constraint (vanishing line) and local constraint (straight lines and vanishing points of each section), the CHEVP algorithm shows promise to provide a robust method for extracting and identifying the lines composing the road, with an ability to reject 'weak' local optimality in an image. Moreover, the parameters of camera are not required. Being initialized by CHEVP, the B-Snake would deform to lane boundaries more precisely by using MMSE approach, which is presented in next Section.

CHEVP algorithm assumes that the horizon (or vanishing line) appears horizontal in the image plane, although it need not be in the camera's field of view. If the terrain varies in slope, CHEVP algorithm may be unable to correctly locate all the road lines due to their not having a vanishing point on the horizon row corresponding to the tangent plane at the vehicle location. However, CHEVP algorithm has ability to identify the best horizon row and vanishing point for each section of the image, but it would lead to a 3D road model, so we leave it to the future work.

4. B-Snake parameters updated from image data

Based on the initial location of the control points that are determined *either* by CHEVP algorithm *or* lane detection result of previous frame, the B-Snake would further approach to road edge accurately in the current frame. This Section deals with this problem.

4.1. Minimum mean square error approach

B-Snake should be updated to minimize (1) the sum of the external forces from the both sides of the road model *for* achieving accurate position of B-Snake, and (2) the difference of the external forces from the both sides of the road model *for* achieving suitable parameter k. In addition, external forces should be transmitted to each control point when updating B-Snake.

When the B-Snake approaches the road boundaries, its external force should satisfy the equation.

$$E_{\text{ext}} = 0 \tag{15}$$

where

$$E_{\text{ext}} = E_{\text{M\_sum}}(s) = E_{\text{L}}(s) + E_{\text{R}}(s). \tag{16}$$

If external force of the B-Snake is zero, then there is no change in both the position and the shape of the mid-line of road. So we can define the following equation for solving the requirement of external force being zero.

$$\begin{aligned} E_{\text{ext}} &= \gamma(L_{\text{mid}}(t) - L_{\text{mid}}(t-1)) \\ &= \gamma M_{\text{R}}(s)(Q(t) - Q(t-1)) = \gamma M_{\text{R}}(s)\Delta Q(t) \end{aligned} \tag{17}$$

where γ is a step-size and $\Delta Q(t)$ is defined as the adjustment of the control points Q in each iteration step.

$$Q(t) = Q(t-1) + \Delta Q(t) \tag{18}$$

External force can be sampled along the B-Spline of B-Snake at a certain distance. Then Eq. (17) can be solved digitally. Here, the MMSE solution for the digital version of the Eq. (17) is given as a matrix form.

$$\Delta Q(t) = \gamma^{-1}[M^{\text{T}}M]^{-1}M^{\text{T}}E_{\text{ext}} \tag{19}$$

where

$$M = \begin{bmatrix} M_{-1} & 0 & \cdots & \cdots & 0 \\ 0 & M_0 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdots & 0 & M_{n-1} & 0 \\ 0 & \cdots & \cdots & 0 & M_n \end{bmatrix},$$

$$M_i = \begin{bmatrix} s_{i,1}^3 & s_{i,1}^2 & s_{i,1} & 1 \\ s_{i,2}^3 & s_{i,2}^2 & s_{i,2} & 1 \\ \cdot & \cdot & \cdot & \cdot \\ s_{i,m}^3 & s_{i,m}^2 & s_{i,m} & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix}. \quad (20)$$

and m is the sampling points number in ith segment of the B-Spline. E_{ext} is the force vector digitized on the B-Spline. Here, $n = 2$ is for the case of using three control points.

The difference of the external forces from the left and the right side of lane model would lead to changing the parameter k (as given in Section 2.3). Estimation of the parameter k can be similarly given as follows.

$$E_k = E^c_{\text{M\_dif}} = (E^c_{\text{L}}(s) - E^c_{\text{R}}(s)) \quad (21)$$

We set the right-hand side of the equation equal to the product of a step size and the negative time derivative of the left-hand side. The resulting equation is:

$$E_k = \tau(k(t) - k(t-1)) = \tau\Delta k(t) \quad (22)$$

$$k(t) = k(t-1) + \Delta k(t) \quad (23)$$

where τ is a step-size for k. Thus,

$$\Delta k(t) = E_k/\tau \quad (24)$$

Here we choose GVF [27] as the external force for B-Snake to perform the lane detection, since GVF has a larger capture range. But GVF is time consuming. So, after successful lane detection, we use the traditional external force (directly calculated from image gradient) to speed up the lane tracking.

4.2. Application in lane detection

4.2.1. Update B-snake parameters for lane detection

In order to achieve the solutions in Eqs. (18) and (23), an iterative procedure is adopted. The steps contained in this iterative minimization process are as follows:

1. Initialization Step. Initialize the control point parameters by CHEVP algorithm introduced in Section 3.
2. Calculate the GVF of the edge road image as the external force of B-Snake.
3. Calculate MMSE in Eqs. (19) and (24) for obtaining $\Delta Q(t)$ and $\Delta k(t)$, respectively.
4. Obtain $Q(t)$ and $k(t)$.
5. If $\|\Delta Q(t)\| >$ threshold^1 and $\|\Delta k(t)\| >$ threshold^2, then set $Q(t)$ to $Q(t-1)$ and $k(t)$ to $k(t-1)$, and go to step 3; Otherwise, go to step 6.
6. Stop. The last estimations of $Q(t)$ and $k(t)$ are regarded as the solutions of MMSE.

4.2.2. Lane detection results

This lane detection algorithm has been simulated and tested on real road images. These lane images include curve and straight road, with or without shadows and lane marks. Some of these results are shown in Fig. 10. The initializations for proposed B-Snake lane model are all obtained from CHEVP algorithm. Fig. 10(a) is the approach result of a curve painted road without any shadow, it can be seen that the B-Snake lane model can achieve a very good result on approaching lane markings on the both sides of the road. Three examples of applying this lane detection algorithm to curve and straight roads with strong shadows are shown in Fig. 10(b)–(d). Their initialized locations of B-Snake are Figs. 8(b), 9(b) and (c), respectively. All results are correctly matching to the lane markings despite the shadows lay over the pained lines. Fig. 10(e) shows an example of MMSE approach to an unpainted road image, whose CHEVP result has been appeared in Fig. 9(d). Although its initialization is not quite correspondent, the lane model approaches the lane boundaries accurately even the both road boundary edges are not smooth near the bottom of the image. A slight curve road, which is detected by the proposed lane detection algorithm, is shown in Fig. 10(f). Please notice in the top of image the result is not matching very well due to weak lane edge pixels. However, the matching is still maintaining high accuracy in the part where is near to the camera.

For a 240×256-pixel road image, the whole processing time of CHEVP and lane detection on a Pentium 3 system with 128RAM is below 4 s, which is depending on the number of edge pixels. However, since it is only for initializing B-Snake and runs only one time at the start performing of lane detection, it would not effect the real-time performance of the lane detection.

4.3. Application in lane tracking

Lane tracking is much easier than lane detection. Considering there are only small changes between two consecutive frames, we can *regard* the estimated parameters of the lane model in the previous frame *as* the initial parameters for the current frame.

The algorithm for lane tracking is quite similar to lane detection except two differences:

- Instead of using CHEVP, the parameters of lane model in the current frame are initialized by the parameters estimated in previous frame.

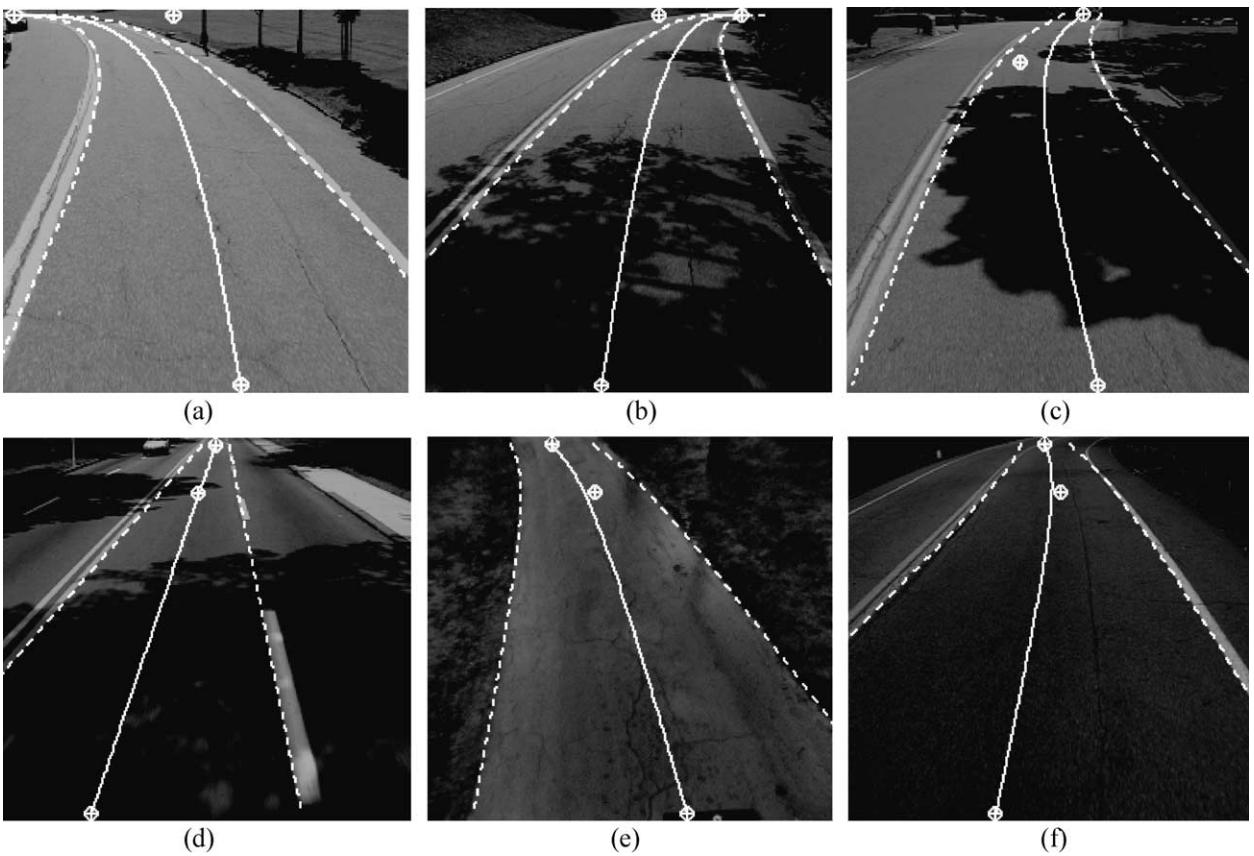


Fig. 10. Lane detection results.

- The GVF is replaced by a simple external force, which is directly calculated from the image gradient.

Several road sequences that include more than 700 road images have been tested for lane tracking. As the algorithm is simple and efficient, in real practice, we can achieve a speed of at least 2 frames/s. The percentage of correct lane tracking is over 95%, depended on the real road conditions. Some results of the lane tracking in one road sequence are shown in Fig. 11, which seems quite good. For the full

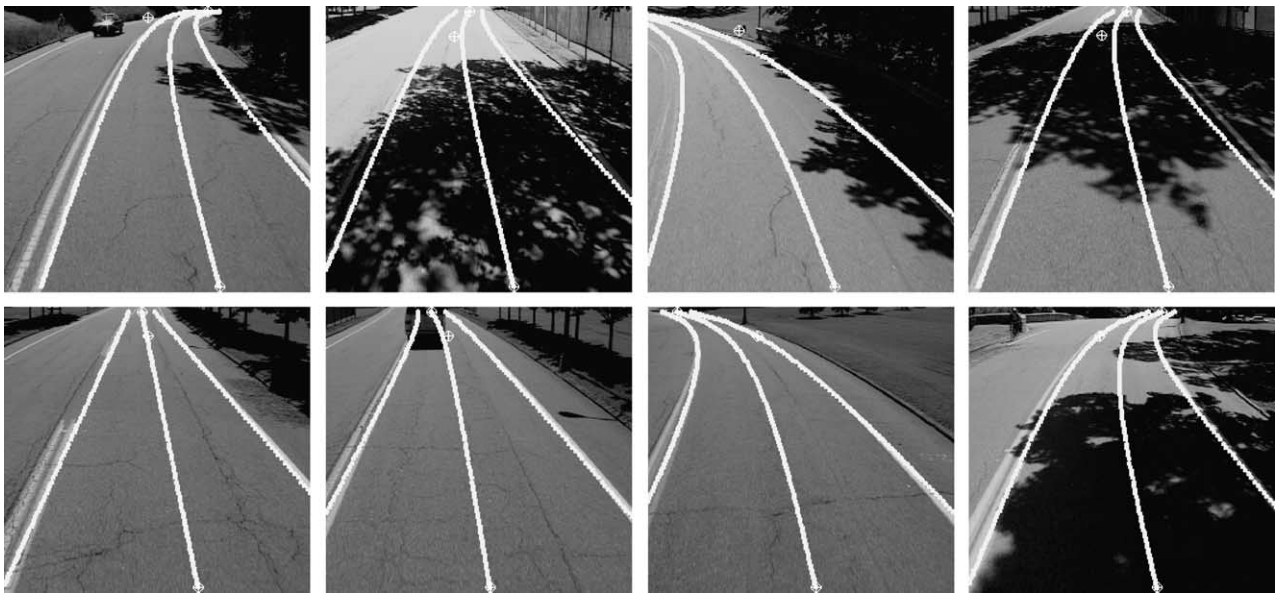


Fig. 11. Some results of lane tracking.

sequence, please visit: www.ntu.edu.sg/home5/ps2633175g/lane_tracking.htm.

5. Conclusion

In this paper, a novel B-Snake based lane model, that describes the perspective effect of parallel lines, has been established for generic lane boundaries (or markings). It is able to describe a wider range of lane structures than other lane models, such as straight and parabolic models. The problems of detecting both sides of lane markings (or boundaries) are merged here as the problem of detecting the mid-line of the lane. A robust algorithm, called CHEVP, is presented for providing a good initial position for the B-Snake lane model. This algorithm is robust against noise, shadows, and illumination variations in the captured road images. It is also applicable to the marked and the unmarked, as well as the dash and solid paint line roads. To approach the lane edges based on the initialized location, a minimum error method, MMSE, that measures the matching degree between the model and the real edge map is presented to determine the control points of road model for lane detection and tracking. In this method, *dual* external forces are sampled along the B-Spline and comprehensively transmitted to each control point for its update by minimizing *both* the sum of the external forces *and* the difference of horizontal components of external forces on the two sides of the road model. The obtained results are quite good and accurate under various conditions. Several extensions of model are possible. In this paper, we mainly focused on 2D lane model. However, it is easy to extend our lane model to 3D lane model by just simply adding in one more component for control points to describe the hill-dale geometry of road. For initializing the 3D lane model, the CHEVP algorithm has to be improved to meet the 3D lane model requirement. In order to improve the lane detection, more features of the road, such as such as color, texture, saturation and reflectance data from the laser scanner, should be used.

References

[1] B. Serge, B. Michel, Road segmentation and obstacle detection by a fast watershed transform, Proceedings of the Intelligent Vehicles '94 Symposium, 296–301, October 1994.
[2] Y. Xuan, B. Serge, B. Michel, Road tracking lane segmentation and obstacle recognition by mathematical morphology, Proceedings of the Intelligent Vehicles '92 (1992) 166–170.
[3] K. Kluge, S. Lakshmanan, A deformable template approach to lane detection, in: I. Masaky (Ed.), Proceedings IEEE Intelligent Vehicle '95, Detroit, September 25–26, 1995, pp. 54–59.
[4] S. Lakshmanan, K. Kluge, Lane detection for automotive sensor, ICASSP (1995) 2955–2958.
[5] A. Broggi, Robust real-time lane and road detection in critical shadow conditions, Proceedings IEEE International Symposium on Computer Vision, Coral Gables, Florida, November 19–21 (1995).
[6] A. Broggi, S. Berte, Vision-based road detection in automotive systems: a real-time expectation-driven approach, Journal of Artificial Intelligence Research 3 (1995) 325–348.
[7] A. Broggi, A massively parallel approach to real-time vision-based road markings detection, in: I. Masaky (Ed.), Proceeding IEEE Intelligent Vehicles '95, 1995, pp. 84–89.
[8] M. Bertozzi, A. Broggi, GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection, IEEE Transactions of Image Processing (1998) 62–81.
[9] H. Andrew, S. Lai, H. Nelson, C. Yung, Lane detection by orientation and length discrimination, IEEE Transactions On Systems, Man and Cybernetics, Part B 30 (4) (2000) 539–548.
[10] S.G. Jeong, C.S. Kim, K.S. Yoon, J.N. Lee, J.I. Bae, M.H. Lee, Real-time lane detection for autonomous navigation, IEEE Proceedings Intelligent Transportation Systems 2001 (2001) 508–513.
[11] D. Grimmer, S. Lakshmanan, A deformable template approach to detecting straight edges in radar images, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1996) 438–443.
[12] K. Kaliyaperumal, S. Lakshmanan, K. Kluge, An algorithm for detecting roads and obstacles in radar images, IEEE Transactions on Vehicular Technology 50 (1) (2001) 170–182.
[13] D. Jung Kang, J. Won Choi, I.S. Kweon, Finding and tracking road lanes using line-snakes, Proceedings of Conference on Intelligent Vehicle, 1996, pp. 189–194, Japan.
[14] A. Kaske, D. Wolf, R. Husson, Lane boundary detection using statistical criteria, International Conference on Quality by Artificial Vision, QCAV9 (1997) 28–30. Le Creusot, France.
[15] A. Kaske, R. Husson, D. Wolf, Chi-square fitting of deformable templates for lane boundary detection, IAR Annual Meeting '95, November 1995 (1995) Grenoble France.
[16] S.P. Liou, R.C. Jain, Road following using vanishing points, Computer Vision, Graphics, and Processing 39 (1987) 116–130.
[17] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, Internatuional Journal of Computer Vision 1 (4) (1987) 321–331.
[18] D. Terzopoulos, K. Fleischer, Deformable models, The Visual Computer 4 (1988) 306–331.
[19] T. McInerney, D. Terzopoulos, A dynamic finite element surface model for segmentation and tracking in multidimensional medical with application to cardiate 4D image analysis, Computerized Medical Imaging and Graphics 19 (1) (1995) 69–83.
[20] F. Leymarie, M.D. Levine, Tracking deformable objects in the plane using as active contour model, IEEE Transactions on Pattern Analysis Machine Intell 15 (6) (1993) 617–634.
[21] R. Durikovic, K. Kaneda, H. Yamashita, Dynamic contour: a texture approach and contour operations, The Visual Compter 11 (1995) 277–289.
[22] D. Terzopoulos, R. Szeliski, Tracking with Kalman Snakes, in: A. Blake, A. Yuille (Eds.), Active Vision, Artificial Intelligence, MIT press, Cambridge, MA, 1992, pp. 3–20.
[23] R.H. Bartels, J.C. beatty, B.A. Barsky, An introduction to splines for use in computer graphics and geometric modeling, Morgan Kaufmann, Los Altos, CA, 1987.
[24] Y. Wang, D. Shen, E.K. Teoh, Lane detection using Catmull-Rom Spline, IEEE International Conference on Intelligent Vehicles (1998) 51–57.
[25] Y. Wang, D. Shen, E.K. Teoh, A novel lane model for lane boundary detection, IAPR Workshop on Machine Vision Applications (1998) 27–30.
[26] Y. Wang, D. Shen, E.K. Teoh, Lane detection using spline model, Pattern Recognition Letters 21 (8) (2000) 677–689.
[27] C. Xu, J.L. Prince, Snakes, shapes, and GVF, IEEE Transactions on Image Processing (1998) 359–369.
[28] Y. Wang, E.K. Teoh, D. Shen, Structure-adaptive B-snake for segmenting complex objects, International Conference on Image Processing (ICIP 2001) (2001) 769–772. Thessaloniki, Greece, Oct 7–10.