Proceedings of the 13th
KACC, October 1998

Lane Detection Using Road Geometry Estimation

Choon-Young Lee, Min-Seok Park, and Ju-Jang Lee
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373-1 Kusong-dong, Yusong-gu, Taejon, 305-701, Korea
Tel: +82-42-869-5432
Fax: +82-42-869-3410
Email: cylee@odyssey.kaist.ac.kr
jjlee@ee.kaist.ac.kr

Abstract

This paper describes how *a priori* road geometry and its estimation may be used to detect road boundaries and lane markings in road scene images. We assume flat road and road boundaries and lane markings are all *Bertrand curves* which have common principal normal vectors. An active contour is used for the detection of road boundary, and we reconstruct its geometric property and make use of it to detect lane markings. Our approach to detect road boundary is based on minimizing energy function including edge related term and geometric constraint term. Lane position is estimated by pixel intensity statistics along the parallel curve shifted properly from boundary of the road. We will show the validity of our algorithm by processing real road images.

1. Introduction

Autonomous navigation has been a challenging task in the past decades which includes many fields such as vehicle control, sensor fusion and planning, computer vision, and so on. One of the major functionality in navigation is to keep track of desired path. Many researchers used camera image for detecting road boundary and lane markings to achieve road following or lateral control of vehicles [1].

Road geometry information can be fused in the sensor data processing algorithm to produce more reliable result in the adaptive cruise control or collision warning system which uses range sensors and vision system [2].

The main problems that must be faced in the detection of road boundaries or lane markings are the presence of shadows, producing artifacts onto the road surface and the presence of other vehicles on the path, occluding the visible region of the road [3]. The technique implemented previously range from the characterization of lane markings by color to an edge-based detection or model-based detection [4].

The edge-based technique is simple to implement and shows good result if edge-threshold value is well-chosen, however, its performance is susceptible to noise pixel elements such as road crack parts, edge induced by shadow. Moreover, edge-threshold value may be easily affected by the change of illumination.

Model-based technique has been used to perform the analysis of intersection, however, it has several problems such as maintenance of an appropriate geometrical model, the difficulty in detecting and matching of features, and the computational heavy load [5].

In this paper, we used physical constraint in road geometry to aid the detection algorithm and speed up the computation. Flat road and continuous road boundaries are assumed in our lane detection algorithm. We used an active contour to detect continuous road boundaries by using energy function considering edge and geometrical constraints. Lane marking on the road can be detected reliably using the road boundary information and parallel curve assumption.

This paper is organized as follows: Section 2 presents the basics of the underlying approach used to find road boundary and lane marking detection with the fact assumed. Section 3 represents how active contour finds the continuous road boundaries. Section 4 describes the lane detection and some results from applying the algorithm on highway road scene. Section 5 ends the paper with a discussion about the problems of the proposed algorithm and conclusion with future direction.

2. Preliminaries

We want to know road geometry information to control vehicle to navigate on the road with safety. General road structure is constrained in its geometric properties. Road has boundaries, lane markings, and sign painting, etc. on its surface. Road where there is no intersection has minimum radius of curvature for the sake of driving safety. So, we have the following assumptions:

Assumption 1: Road is flat plane such that curves on a road are plane curves

Assumption 2: Road boundaries are always seen in the image and have continuous edges

Assumption 3: Road boundaries and lane marking are Bertrand curves[11], that is, two curves on a road, at any of their points, have a common principal normal.

Remarks: Assumption 1 guarantees that road region on the image can be reconstructed by inverse perspective projection even with single camera. Assumption 3 makes us find lane marking by using continuous boundaries according to Section 4.

Property of Perspective transform

We will study a camera model. We can choose the coordinate system (C,X_c,Y_c,Z_c) for the three-dimensional space and (c,u,v) for the image plane. The coordinate system (C,X_c,Y_c,Z_c) is called the *standard coordinate system* of the camera. From Figure 1 it should be clear that the relationship between image coordinates and 3-D space coordinates can be written as

$$u = x^* / t^*$$
$$v = y^* / t^*$$

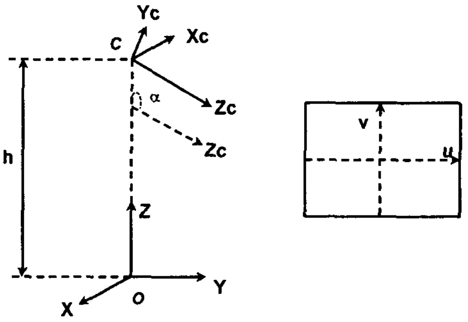
These equations allow us to interpret x^*, y^*, t^* as the projective coordinates of a point in the retina. If $t^* = 0$, the 3-D point is in the focal plane of the camera. Thus the coordinates u and v are not defined, and the corresponding point is at infinity. Now, we will change 3-D coordinate system for convenience. We go from the old coordinate system centered at the optical center C to the new coordinate system centered at O by a rotation and a translation. Then, the reconstruction of the image point and its inverse can be written as [6]

$$\begin{aligned} X &= \frac{A(u\sin\alpha) + B(-v\sin\alpha + f\cos\alpha)}{f(v\sin\alpha - f\cos\alpha)} \\ Y &= \frac{Af}{f(v\sin\alpha - f\cos\alpha)} \\ A &= vh\cos\alpha + vf + fh\sin\alpha \\ B &= uh\cos\alpha + uf \end{aligned} \quad (1)$$

$$\begin{aligned} u &= f\frac{-X}{-Y\sin\alpha + h\cos\alpha + f} \\ v &= f\frac{-Y\cos\alpha - h\sin\alpha}{-Y\sin\alpha + h\cos\alpha + f} \end{aligned} \quad (2)$$

Camera Calibration [9]

The ground plane coordinate frame (x_V, y_V) is defined with respect to the vehicle where y_V is straight ahead of the vehicle and x_V is to the right of the vehicle. The image is defined by column and row parameters as (u_I, v_I). This calibration assumes that the camera is not rolled, therefore every row location in the image maps to a unique y_V location on the ground. Specifically, only two rows of the



(a) (b)

Figure 1 Coordinate Systems (a) Camera and Vehicle Coordinate (b) Image Plane Coordinate

image are calibrated, v_1 and v_2. These rows correspond to y_1 and y_2 on the ground plane. Also for these rows in the image, we measure the number of horizontal pixels p_1 and p_2 in the image corresponding to a specific length on the ground plane. The column locations, u_1 and u_2 corresponding to $x_V = 0$ are measured. Therefore, using this model, pixels lying in the two calibrated rows, having position (u,v_i) can be back-projected to the ground plane by $y = y_i$ and $x = (u-u_i)/(p_i)$. Similarly, positions lying on the calibrated y locations y_1 and y_2 can be projected into the image using $v = v_i$ and $u = xp_i + u_i$.

To back-project a line from the image to the ground plane, the line is intersected with the calibrated row locations in the image to have a two-point description of the line. Then the positions are back-projected as described above. Similarly to project a line from the ground plane to the image, the line is intersected with the calibrated y locations in the ground to have a two point description of the line, and positions are projected as described above.

To back-project an arbitrary point, for example the kernel location of an intersection, from the image to the ground plane, we first define two different lines in the image that pass through the point and intersect with the calibrated row locations. We back-project the two lines and then compute their intersection on the ground plane. Similarly, to project an arbitrary point from the ground plane to the image, we first define two different lines on the ground that pass through the point and intersect with the calibrated y locations. We then project the two lines and then compute their intersection in the image.

This calibration uses a simple linear camera model with some camera position assumptions (i.e., the camera is not rolled with respect to the ground plane). This calibration does not account for lens distortion. However it is good enough to map the ground plane to the image as long as the assumption that road is planar and parallel to vehicle is valid.

Active Contours [8]

The methods of object tracking such as difference image method and spatio-temporal descent method have been researched by many researchers. These methods have some demerits those are long calculation time and the possibilities of successful identification of objects.. Snakes, that is active contour models, are deformable contours that have been used in many image analysis applications, including the image-based tracking of rigid and non-rigid objects. The snake equations of motion provide flexible tracking mechanisms that are driven by simulated forces derived from time-varying images [9].

Snakes are defined based on energy functions, and can be deformed to a certain contour form which would converge to the minimum energy states by the forces produced from energy differences. Snakes was introduced by Kass in 1987, and he modeled objects by snake functions. He optimized the energy states by using calculus of variation. In 1988 Amini solved some problems such as the phenomenon of concentration of snakes points to a certain point and instability through active contour model by using dynamic programming method. Laurent suggested 'Balloons model' which gives a kind of expansion force to prevent contraction phenomenon [7].

Snake is a kind of optimal energy spline that can be expressed equation (3) which has arc length 's' as a parameter.

$$\vec{v} = (x(s), y(s)) \quad (3)$$

Using this snake, active contour, the edges of objects can be extracted because snake has a tendency of natural moving to the boundaries of objects due to energy difference. So the definition of energy function is more important subject in active contour problems because snake moves along this energy difference. At the 1st stage Kass configured energy functions comprised of internal energy, image energy and external energy like equation (4).

$$\begin{aligned} E_{snake} &= \int_0^1 E_{snake}((\vec{v})(s))\,ds \\ &= \int_0^1 (E_{\text{int}}(\vec{v}(s)) + E_{\text{image}}(\vec{v}(s)) \\ &\quad + E_{ext}(\vec{v}(s))\ )\,ds \end{aligned} \quad (4)$$

Internal energy: This energy term make the energy contour that have smooth curved form and have no discontinuous points in a snake. Kass defined his internal energy as the function that can be divided into two sub energy terms like equation (5).

$$\begin{aligned} E_{\text{int}} &= \frac{1}{2}\,(E_{cont} + E_{curv}) \\ &= \frac{1}{2}\,(\alpha(s)|\vec{v}_s(s)|^2 + \beta(s)|\vec{v}_{ss}(s)|^2) \end{aligned} \quad (5)$$

where subscript letter 's' means the derivative of 's', and α, β are the weighting coefficients. The first term of above equation prevents the discontinuity of snake because it was gained by differentiating its snake. If a snake has discontinuities on its contours or radical variation then this energy term should have relatively big values. So by the characteristics of snake for finding the minimum state of energy snake doesn't have any tendency to have discontinuity.

The second term is called the curvature energy term because it prevents radical curvature on its contour. It can be produced by second derivatives.

Image energy: The image energy term is comprised of three terms such as line energy, edge energy, and terminal energy like equation (6).

$$E_{image} = \gamma_{\text{line}} E_{line} + \gamma_{\text{edge}} E_{edge} + \gamma_{\text{ter}} E_{ter} \quad (6)$$

The line energy term may be gained easily using the intensities, I(x,y), from each pixel point of image, and likes bright or dark areas. The edge energy term, E_{edge}, is a function for the consideration of edge of the objects in the image. E_{edge} have great values at the points at which the difference of intensities is bigger than surrounded ones. Above equation (6) may be changed to equation (7) by considering discrete system.

$$\begin{aligned} E_{snake} = \sum_{i=1}^{N} (\ & \alpha_i(s)|\vec{v}_i - \vec{v}_{i-1}|^2 \\ & + \beta_i(s)|\vec{v}_{i-1} - 2\vec{v}_i + \vec{v}_{i+1}|^2 \\ & + \gamma_i(s)(\ -|\nabla I(x,y)|^2\ )) \end{aligned} \quad (7)$$

Kass used calculus of variation for finding the minimum energy states, and configured some Euler's equations. But his solution should find the inverse of N by N matrix for a snake of N points. There are much computational time for finding his solution. By the way snake forces have somewhat small magnitude at the area in which the derivatives of gradient term is zero, so snake would move to the edges very slowly.

3.Road Boundary Detection

The boundaries of road usually have strong edge characteristics, so if we make energy functions have much image information then snakes would naturally converge to the boundaries. In active contour concept energy functions should have minimum value when certain conditions are satisfied. If the distance between surrounding two nodes are maintained, the curvatures at each node have not big values, and all the nodes stay on the edge spline then the value of snake energy function would converge to the minimum value. By this characteristics of snake contour we could find the road boundaries easily.

But the energy function of Kass may have much computation time and instability, and a method using a dynamic programming method gives somewhat complex calculation strategies. To overcome this mathematical complexity and instability of snake Jin-Woo Yi proposed the point algorithm using the steepest descent method and Brent's method [7]. He gained the directions of snake forces from the application of steepest descent method, and the minimum energy point by using Brent's method at the directions of snake forces. So he had a more fast convergence speed for a calculation of the minimum energy point, and could exclude mathematical instability due to noise and the differentiation of the energy function. Also he used recursive Gaussian filtering method to filter the noise in the image and to gain the gradients of image intensity without direct differentiating the energy function

Energy function of point algorithm: The energy function is comprised of three major terms such as continuous energy, curvature energy, and image energy like equation (8).

$$E_{snake} = \sum_{i=1}^{N} (\alpha_i E_{conti} + \beta_i E_{curvature} + \gamma_i E_{image}) \quad (8)$$

E_{conti} is a energy function for the continuous formation of snake contour, and maintains the distance between the nodes of snake within the average distance of all the nodes. It can be represented by equation (9)

$$E_{conti} = \left(\frac{\bar{d} - |\vec{v}_i - \vec{v}_{i-1}|}{\bar{d}}\right)^2 \quad (9)$$

where v_i, and $\bar{d}$ mean the i th node, the average distance between two nodes respectively.

$E_{curvature}$ is designed to maintain smooth curved form of contour using two vectors made of three surrounding nodes like equation (10)

$$E_{curvature} = \beta_i \left| \frac{\vec{u}_i}{|\vec{u}_i|} - \frac{\vec{u}_{i+1}}{|\vec{u}_{i+1}|} \right|^2 \quad (10)$$
$$where\ \vec{u}_i = \vec{v}_i - \vec{v}_{i-1}, \vec{u}_{i+1} = \vec{v}_{i+1} - \vec{v}_i$$

E_{image} may be the lowest value when the gradient of the pixel positioned (x, y) has maximum value. With this characteristics he can find edges of the objects easily. E_{image} can be expressed as equation (11). Threshold value should be selected a value above the maximum gradient value squared in the image for the exclusion of minus values of the energy function.

$$E_{image} = \frac{Threshold - |\nabla I(x,y)|^2}{Threshold} \quad (11)$$

Snake forces of point algorithm: Snake forces can be generated using the energy function. to maintain the average distance between two surrounding nodes, to make snake's form smoothly curved and to find the edges of the objects in the image. The magnitudes of the snake forces don't be used to find the minimum energy points but the directions of the snake's forces would be used to give the directions of the steepest descent gradient. If the directions of the snake forces are determined, then the minimum of the energy points may be calculated by using the Brent's method rather fast.

We applied the above scheme to a road scene as following steps. At first, we select initial snake points around road boundaries. At each step we calculated the directions of snake forces and found the points that had the minimum energy values on the direction by the Brent's method. We checked whether those points have the minimum energy value, and if the condition was satisfied then we finished iterations.

4.Lane Detection

Our lane detection algorithm is based on the fact that the line segment in 3-D world coordinate corresponds to the line segment in 2-D image segment [10].

Let p_1 and p_2 be two points in the 3-D world represented in homogeneous coordinate system. The line passing through p_1 and p_2 consists of all points having the form $kp_1 + (1-k) p_2$ for some constant k.

We let the perspective transform of p_1 and p_2 be $\begin{pmatrix} u_1/w_1 \\ v_1/w_1 \end{pmatrix}$ and $\begin{pmatrix} u_2/w_2 \\ v_2/w_2 \end{pmatrix}$, respectively, for the perspective transformation matrix T,

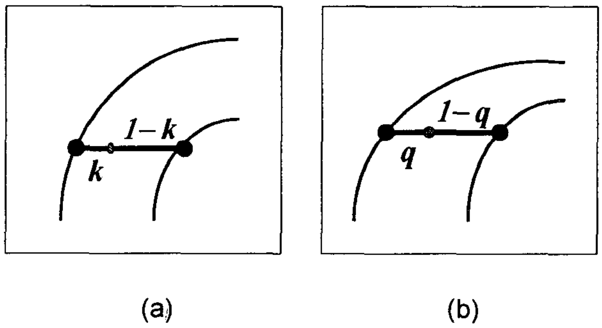


Figure 2 Relationship between line segments in (a) world coordinate and (b) image coordinate

$$\text{where } \begin{pmatrix} u_1 \\ v_1 \\ w_1 \end{pmatrix} = Tp_1 \text{ and } \begin{pmatrix} u_2 \\ v_2 \\ w_2 \end{pmatrix} = Tp_2 \tag{12}$$

The exact nature of the matrix T is not important in this discussion. We need only use the fact that T is a linear operator.

$$T[kp_1 + (1-k)p_2] = kTp_1 + (1-k)p_2$$
$$= k\begin{pmatrix} u_1 \\ v_1 \\ w_1 \end{pmatrix} + (1-k)\begin{pmatrix} u_2 \\ v_2 \\ w_2 \end{pmatrix} \tag{13}$$

The line segment on the image plane passing through $\begin{pmatrix} u_1/w_1 \\ v_1/w_1 \end{pmatrix}$ and $\begin{pmatrix} u_2/w_2 \\ v_2/w_2 \end{pmatrix}$ consists of all points of the form

$$q\begin{pmatrix} u_1/w_1 \\ v_1/w_1 \end{pmatrix} + (1\text{-}q)\begin{pmatrix} u_2/w_2 \\ v_2/w_2 \end{pmatrix} \tag{14}$$

for the some constant q.

To show that the perspective projection of every line in the 3-D world is also a line in the image plane and every line in the image plane corresponds to a line in the 3-D model, we need to show that for every parameter k, there exists a parameter q, and *vice versa*.

$$\begin{pmatrix} \dfrac{ku_1 + (1-k)u_2}{kw_1 + (1-k)w_2} \\ \dfrac{kv_1 + (1-k)v_2}{kw_1 + (1-k)w_2} \end{pmatrix} = q\begin{pmatrix} u_1/w_1 \\ v_1/w_1 \end{pmatrix} + (1\text{-q})\begin{pmatrix} u_2/w_2 \\ v_2/w_2 \end{pmatrix} \tag{15}$$

Therefore, the following relationship is obtained:

$$q = \frac{kw_1}{kw_1 + (1-k)w_2}$$
$$k = \frac{qw_2}{kw_2 + (1-k)w_1} \tag{16}$$

This equation states that if we know interior division point of line segment in 3-D world, we are able to know which point corresponds in image plane (Figure 2).

We will detect lane markings position by comparing pixel intensity statistics generated along the parallel curve shifted properly from the boundary of the road.

The algorithm is as follows:

Step 1: Road Boundary Reconstruction
Tow road boundaries are reconstructed in vehicle coordinate by inverse perspective projection
Step 2: Shifted Parallel Curve Generation
Select one interior division point on the line segment connecting two boundaries in normal direction
Step 3: Make Statistics
Calculate the histogram profile, mean intensity and variance along the curve
Step 4: Compare and Find Lane Marking
Since lane markings have high intensity generally, comparison between the intensity profiles in each curve makes us choose which is lane marking.

The above algorithm is implemented in real road image having shadows and vehicle occluding lane marking or boundaries. Since summation of intensities along curve has effect of noise suppression, we can detect lane markings reliably even though lane markings are discrete or damaged by some material or occluded partially by some other objects. Results are shown in Figure 3 - 6.

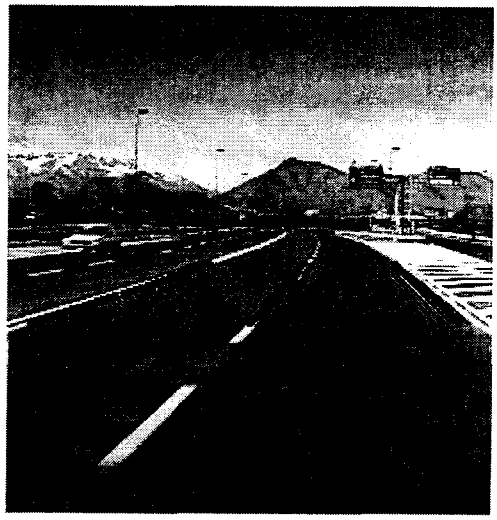


Figure 3 Lane Detection for normal road

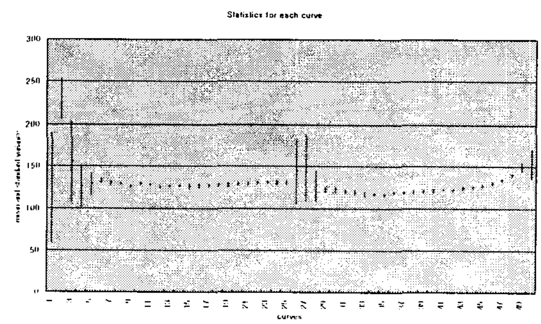


Figure 4 Statistics in each curve

In Figure 3, the most simple result is shown for the case that no vehicle exists on the road. The black line indicates curves parallel to the boundaries. The statistics of each curve is shown in Figure 4. Middle dot means average intensity value and the vertical bar indicates the variation by standard deviation. Curves along markings have high intensity and have different standard deviations from neighboring curves. We can have road region statistics by calculating off-line or may update it at every frame using bottom part of image. So, we can easily detect lane markings by the statistics of each curve parallel to boundary.

In Figure 5, there is a car running on the left lane and occludes road region. The occluding vehicle changes curve statistics and make it difficult to detect lane marking, however, we can use the fact that lane marking region is narrower than other road region and vehicle part. In Figure 6, Statistics of each curve in Figure 5 is shown. The vehicle affects standard variations of road and makes its distribution more wide. Moreover, shadows by the vehicle change statistics. In this case, we can use the knowledge about road structure and remove false lane position and select correct one.

Figure 5 Lane Detection in case that vehicle occludes road

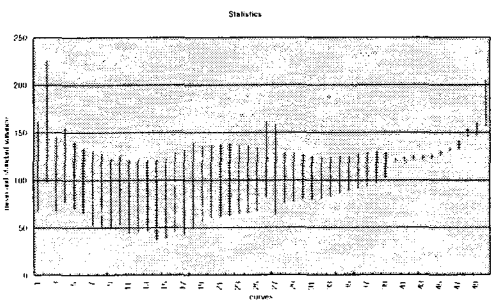


Figure 6 Statistics in case that vehicle occludes road

5. Conclusion

A method has been proposed in this paper that detects road boundaries and lane markings. In the proposed algorithm, we first find the road boundary using active contour by minimizing energy function, and then we find lane marking using previously determined boundary.

Our algorithm finds road boundary systematically not with heuristic search. The convergence of active contour to the road boundary can be affected by other line component. This can be overcome by setting geometry related term strong so that contour energy is dominated by road boundary geometry. Lane detection algorithm finds lane position by comparing curve statistics parallel to boundary.

6. References

[1] J. Kosecka, R. Blasi, C. Taylor, and J. Malic, "Vision-based Lateral Control of Vehicles," *Proc. of Intelligent Transportation Systems Conference*, 1997.

[2] C. Lee, and J. Lee, "Data Processing Algorithm of Laser Scanning Sensor for Adaptive Cruise Control," *Submitted to Korea SAE*, 1998.

[3] M. Bertozzi, and A. Broggi, "GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection," *IEEE Trans. on Image Processing*, Vol. 7, No. 1, pp. 62-81, 1998.

[4] J. Crisman and C. Thorpe, "SCARF: A color vision system that tracks roads and intersection," *IEEE Trans. on Robotics and Automation*, Vol. 9, pp. 49-58, 1993.

[5] D. Raviv and M. Herman, "A New Approach to Vision and Control for Road Following," *Proc. of Computer Vision and Pattern Recognition*, pp.217-225, 1991.

[6] N. Yoo, "Extracting Road Lane Boundaries using the Perspective Transform and Its Application to the Obstacle Detection on Curved Road," MS Thesis, KAIST, 1996.

[7] J. Yi, "A Study on the Development of the Tracking algorithm using Active contour model", Master Thesis, Dept. of Precision Engineering and Mechatronics, KAIST, December 1992.

[8] W. Kim, C. Lee, and J. Lee, "Visual Tracking of Objects for a Mobile Robot using Point Snake Algorithm," *Proc. of Korea Automatic Control Conference*, International Session, 1998.

[9] A. Blake and A. Yuille, *Active Vision*, MIT Press, 1992.

[10] R. Haralick and L. Shapiro, *Computer and Robot Vision Vol. II*, Addison-Wesley Publishing Company, 1993.

[11] E. Kreyszig, *Differential Geometry*, Dover Publication, 1991.