The Research of Lane Marker Detection Algorithm Based on Inverse Perspective Mapping

Jinghong XU[1, a], Changsheng CHENG[1, b] and Yan WAN[2, c]

[1]Computer Science and Technology Department, Jianqiao University, 1500 Kangqiao Road Pudong, Shanghai, China 201315

[2]Computer Science and Technology Department, Donghua University, 2999 North Renmin Road, Shanghai, China 201620

[a]xingfuhong1314@163.com, [b]cscheng@dhu.edu.cn, [c]winniewanyan@gmail.com

Keywords: IPM, level-line, lane marker abstracted region, class point, rectangular approximation

Abstract. This paper presents a robust and fast algorithm for lane marker detection. It is based on generating an inverse perspective mapping image of the road, computing line abstracted regions and the class points from IPM image, which is followed by a post processing step using rectangular approximation to detect lane marker. A reference database containing over 100 images of natural road scenes was built with corresponding manually labeled ground truth images (available at http://www.lcpc.fr/english/products/). This database enables to evaluate lane marker detection algorithm in a systematic way. The experiment proves that the algorithm proposed can detect the lane marker on the road in various conditions.

Introduction

In recent years, Android systems provided higher performance with low cost and small size. The video based Android application, driving assistance system, can be implemented on a vehicle in real time. The application analyzes views in front of the car to help driver driving safely.

Lane marker detection[1,5,7,8,10,17] has received considerable attention in driving assistance system. Techniques used varied from using low level morphological operations[1] to probabilistic grouping and B-snakes[2,6,9], using monocular[1] to stereo vision[3,4]. However, most of these techniques were focused on detection of lane markers on highway roads, which is an easier task compared to lane marker detection in urban streets. Lane marker detection in urban streets is especially a hard problem. Challenges include: parked and moving vehicles, bad quality lines, shadows cast from trees, buildings and other vehicles, sharper curves, irregular/strange lane marker shapes, emerging and merging lane markers, sun glare, writings and other markings on the road (e.g. pedestrian crosswalks), different pavement materials, and different slopes.

This paper presents a simple, fast, robust, and effective approach to tackle this problem. It is based on taking a top view of the image, called the Inverse Perspective Mapping[1]. From IPM image, the lane markers abstracted region and class points are calculated. Referring to that, density of class point and rectangular approximation methods are used in algorithm to ignore noise in rectangle region. And then, the rectangle regions are verified by Helmholtz principle[15,16]. Finally, lane markers are abstracted from the possible rectangle region.

Theoretical Foundation

Frames Definition. To get the inverse perspective image from the captured image, it is assumed that the road is flat, and the camera intrinsic (focal length) and extrinsic (pitch angle, yaw angle, and height above ground) parameters is used to perform the transformation. For this aim, three frames are defined:

- Word Frame $F_w : \{x_w, y_w, z_w\}$

- Camera Frame $F_c : \{x_c, y_c, z_c\}$
- Image Frame $F_i : \{x_i, y_i\}$

And the relation of those frames listed above is illustrated in Fig. 1. The actual road conditions are chosen to build the world coordinate system to represent the three-dimensional space, and the camera imaging coordinate system is established to represent the two-dimensional image space.

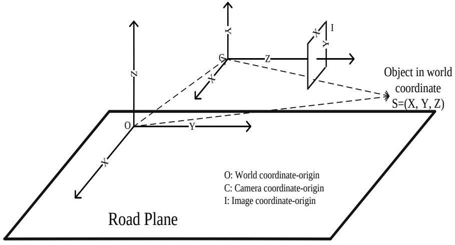

Y
I
X
Y
Z
C
Z
X
Object in world
coordinate
S=(X, Y, Z)
O
Y
X
O: World coordinate-origin
C: Camera coordinate-origin
I: Image coordinate-origin
Road Plane

Fig. 1 Relation of the frames

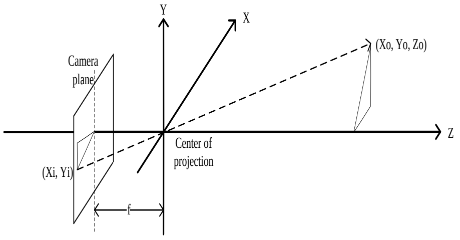


Fig. 2 Foreshortening factor

Perspective Projected Geometry. In computer vision system, how to project captured image from 3D to a 2D perspective view is a fundamental operation. Due to the perspective effect, the captured result will contain some distortions caused by the foreshortening factor and vanishing point issues[11,12,13].

1) Foreshortening factor

The coordinates of captured image can be described mathematically, using x_i and y_i to components ($F_i : \{x_i, y_i\}$), with respect to its actual coordinates in the world space ($F_w : \{x_w, y_w, z_w\}$). The relation is given by the equation (1).

$$(x_i, y_i) = (f\frac{x_w}{z_w}, f\frac{y_w}{z_w}) \tag{1}$$

Referring to Fig. 2 of the camera model, when object moves further away along the z axis, the value of z will be increased. From (1), it shows that the increase in the value of z causes the object (width x_i, height y_i) to reduce in size and this phenomenon is known as a foreshortening factor.

2) Vanishing point

A vanishing point occurs when a set of projected parallel lines appears to converge and intersect at a point. Fig. 3 illustrates the vanishing point phenomenon.

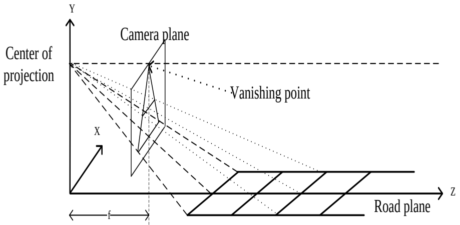
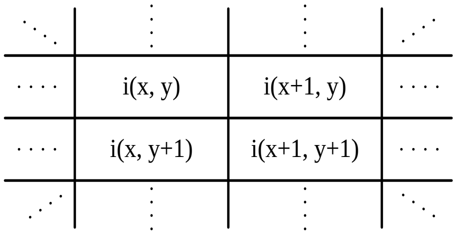


Fig. 3 Vanishing point



⋱	⋮	⋮	⋰
····	i(x, y)	i(x+1, y)	····
····	i(x, y+1)	i(x+1, y+1)	····
⋰	⋮	⋮	⋱

Fig. 4 Gradient computation

Gradient's Computation. The image gradient is computed at each pixel by using a 2×2 mask. That mask is showed by Fig. 4.

Note that $i(x, y)$ is the image gray level value at pixel (x, y), the image gradient is computed by the equation (2).

$$\begin{cases} g_x(x, y) = \dfrac{i(x+1, y) + i(x+1, y+1) - i(x, y) - i(x, y+1)}{2} \\ g_y(x, y) = \dfrac{i(x, y+1) + i(x+1, y+1) - i(x, y) - i(x+1, y)}{2} \end{cases} \tag{2}$$

The level-line angle is computed as the equation (3).

$$\arctan\left(\frac{g_x(x,y)}{-g_y(x,y)}\right) \tag{3}$$

And the gradient's magnitude is calculated as the equation (4).

$$G(x,y)=\sqrt{g_x^2(x,y)+g_y^2(x,y)} \tag{4}$$

The smallest mask size is used in gradient computation, thus the algorithm complexity is reduced as much as possible.

Tolerance Pseudo-Ordering. Sorting algorithm usually require $O(n\log n)$ operations to sort n values. However, tolerance pseudo-ordering makes it possible in linear time. 10 boxes are created corresponding to level-line angle's (is described in lane marker abstracted algorithm) value intervals between the min to the max in connected region. Pixels are classified into the boxes by their level-line angle. And that sorted pixels are been used to reduce the possible rectangle regions.

Lane Marker Detection Algorithm

Inverse Perspective Mapping Algorithm. The first step in this algorithm is to generate a top view of the road image[1] using inverse perspective mapping approach. That can get rid of the perspective effect of the captured image, and so lane marker that appear to converge at the horizon line are now nearly vertical and parallel. Here uses the main assumption that the road plane is flat. Algorithm can focus on lane marker detection without computation of road plane angle, which helps in reducing the running time considerably. The inverse perspective image is showed in Fig. 5, the left part is the original one, and the right one is the image without perspective effect.

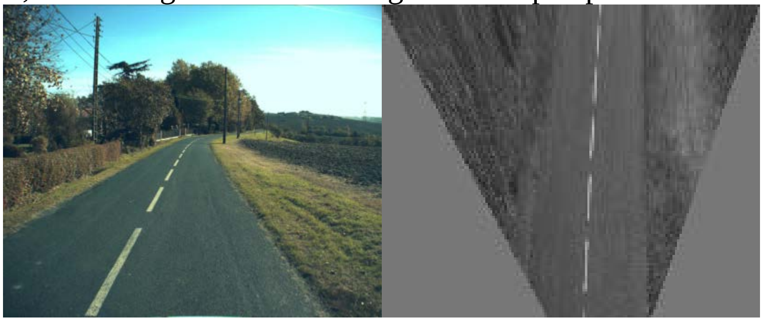
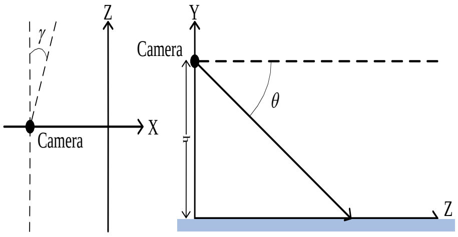


Fig. 5 Inverse perspective mapping

On the basis of section Ⅱ and the angles of the camera are illustrated in Fig. 6,



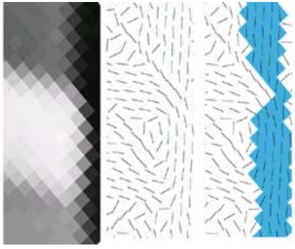


Fig. 6 Camera extrinsic parameters Fig. 7 Lane marker abstracted region

The mapping matrix[14] is constructed between world frame and camera frame by the equation (5).

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & -R\tilde{C} \\ O^T & 1 \end{bmatrix} \bullet \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{5}$$

Note that it is the rotation matrix composed by the angles of θ and γ. Symbol $\tilde{C}$ is the translation vector between O and C (in Figure 1). The projection of the points from the camera coordinates into image coordinates is given by the equation (6).

The symbol $\sim$ means equality except proportion. This latter matrix is denominated K .The transform between image points and world coordinates is immediately obtained by the equation (7).

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \sim \begin{bmatrix} \frac{-M}{\tan\alpha_u} & 0 & x_o & 0 \\ 0 & \frac{-N}{\tan\alpha_v} & y_o & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \sim KR\left[I_3 \,|\, -\tilde{C}\right] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{7}$$

Where the I_3 is the 3×3 identity matrix, and $M_T = KR\left[I_3 \,|\, -\tilde{C}\right]$ is usually called camera projection matrix.

Therefore, points of the real world belonging to the road, with world coordinates P_w are projected into the image at coordinates P_i through the projection matrix M_T as in (8).

$$P_i \sim M_T P_w \tag{8}$$

Lane Marker Abstracted Algorithm. After generation of IPM image, lane marker abstracted algorithm computes the level-line angle at each pixel to obtain a level-line field. All unit vectors are tangent to the level-line going through their base point. And then, the field is segmented into several connected regions that sharing the similar level-line angle up to a certain angle tolerance τ . In some cases, the tolerance angle τ is used to produces a wrong connected region. 11.25, 22.5, and 45 degree are used respectively over 100 images, and the result shows that the 11.25 case is too restrictive and the region is too small; with 45 degree regions often expand too far from the edge; 22.5 is a good compromise. These connected regions are called lane marker abstracted regions, shows in Figure 7. The left part of Figure 7 is the grey level image. Next to it is the line-level angle image. And the blue region is the connected region in the right part of Fig. 7.

Each lane marker abstracted region (a set of pixels) is a candidate for lane marker abstracted algorithm. The corresponding geometrical object must be associated with it. In this algorithm, the rectangle is chosen as the corresponding object, and the size of it (that is described in section of Rectangular Approximation) should cover the full region, as shown in Fig. 8.

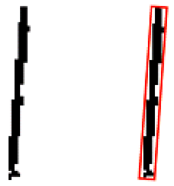


Fig. 8 Rectangle approximation

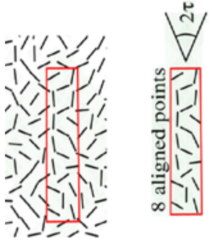


Fig. 9 Class points

Each rectangle is a subject in validation procedure. The pixels in it whose level-line angle up to a tolerance are called class points, see Fig. 9. In the rectangle region, total number of pixels is n , and the number of class points is c . Those two kinds of points are used to validate whether the rectangle region contains the lane marker. The validation procedure is based on the "a contrario approach" and the Helmholtz principle proposed by Desolneux, Moisan, and Morel[15,16]. The Helmholtz principle states that no perception should be produced on an image of noise. Accordingly, the "a contrario approach" proposes to define "a contrario model" where the desired structure is not present. Then, an event is validated if the expected number of events as good as the observed one is small on the "a contrario model". In other words, structured events are defined as being rare in the "a contrario model" [15,16].

In the case of lane marker abstracted, considering the event that a lane marker abstracted region in the "a contrario model" has as many or more class points as in the other observed lane marker abstracted region. Given an image i and a rectangle r, algorithm notes $c(r,i)$ the number of class points and $n(r)$ is the total number of pixels in r. Then, the expected number of events which are as good as the observed one is the equation (9).

$$N_{test} \bullet P_M\left[c(r,i) \geq c(r,I)\right] \tag{9}$$

Where the number of tests N_{test} is the total number of possible rectangle regions being considered, P_M is the probability on the "a contrario model" M (that is defined below), and i is a random image following M. M stochastic model fixes the distribution of the number of class points $c(r,i)$, which only depends on the distribution of the level-line field associated with i.

The "a contrario model" M used for lane marker abstracted algorithm is therefore defined as a stochastic model of the level-line field satisfying the following properties,

- $\{LLA(j)\}, j \in I$ is composed of independent random variables.
- $LLA(j)$ is uniformly distributed over $[0, 2\pi]$.

Where $LLA(j)$ is the level-line angle at the pixel j. Under hypothesis M, the probability that a pixel on the "a contrario model" is an class point is $P = \tau/\pi$ and $c(r,i)$ follows a binomial distribution. Thus, the probability is given by the equation (10).

$$P_M\left[c(r,i) \geq c(r,I)\right] = B(n(r), c(r,i), p) \tag{10}$$

Where the $B(n,c,p)$ is the tail of binomial distribution,

$$B(n,c,p) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j} \tag{11}$$

On the basis of the section inverse perspective mapping algorithm, the set of lane marker lines are nearly vertical and parallel. So, the connected region direction threshold is calculated with the first 8 boxes in tolerance pseudo ordering. The threshold can reduce the number of possible rectangle regions from N_{test} to N_γ.

Finally, the Number of False Alarms (NFA) is defined associated with a rectangle r on the image i as the equation (12).

$$NFA(r,i) = N_\gamma \bullet B(n(r), c(r,i), p) \tag{12}$$

A threshold ε is selected, and when a rectangle has $NFA(r,i) \leq \varepsilon$, ε is called meaningful rectangle and produces a lane marker detection. The theorem states that the average number of ε meaningful rectangles under the "a contrario model" M is less than ε. Thus, the number of detections on noise is controlled by ε and it can be made as small as desired.

Rectangular Approximation. A lane marker abstracted region corresponds to a geometrical object, a rectangle. Before evaluating a lane marker abstracted region, the rectangle associated with it must be found. Then, the direction of the rectangle is set by tolerance pseudo ordering. Finally, the width and length of the rectangles are set to the smallest values that make the rectangle to cover the full line-support region. In some cases, the angle tolerance τ method produces a wrong interpretation. This problem can arise when two straight edges are present in the image forming an angle between them smaller than the tolerance τ. The detection of this angle problem is based on the density of class points in the rectangle. And the density is defined by the equation (13).

$$d = \frac{c(r,i)}{length(r) \times width(r)} \tag{13}$$

Algorithm sets d to the value 0.7 (the threshold is calculated by the reference database) which provides good balance between solving the angle problem, without over-cutting the line segments. When this problem is not present, the rectangle is well adapted to the lane marker abstracted region and the density of class points is high. On the other hand, when the angle problem is present the density of class points is low. As the density of class points is not enough in this rectangle region, the following two steps are using for approximation.

Reduce angle tolerance

Reduce angle tolerance method tries to calculate a new threshold that adapts well to the region, and then get the rectangle region with the same seed but using the newly estimated tolerance value.

All the pixels in that rectangle area are evaluated, and the new tolerance threshold is set to twice the standard deviation of the level-line angles of these pixels. With new value, generating rectangle region on the same class points.

If all the pixels in the region were used in the estimation of the tolerance, the new threshold should be accepted by all the pixels not only by the pixels near the seeds. Actually, only the pixels whose distance to the seed point are less than the width of the rectangle are used. In this way, the lane marker region can be decreased during estimation of adapts.

Reduce region radius

The previous method, reduce angle tolerance, is tried only once, and if the resulting line-support region fails to satisfy the density criterion, the second method is repetitively attempted. The design of this second method is to gradually remove the pixels that are farther from the seed until the criterion is satisfied or the region is too small and rejected.

The Euclidean distance from the seed point to the farther pixel in the region is called the radius of the region. Each iteration of this method removes the farthest pixels of the region to reduce the region's radius to 90% (the threshold is calculated by the reference database) of its value. This process is repeated until the density criterion is satisfied or until there are not enough pixels in the region to be a meaningful rectangle. This is just a way of gradually reducing the region until the criterion is satisfied.

Experiment and Result

The images in Fig. 10 show the results of algorithm in detecting lane markers on streets with various conditions. The results of detecting lane markers in each image are independently without utilizing any temporal information. However, when detecting only the lane marker boundaries of the current lane marker, the algorithm achieve comparable results to other algorithms[2].

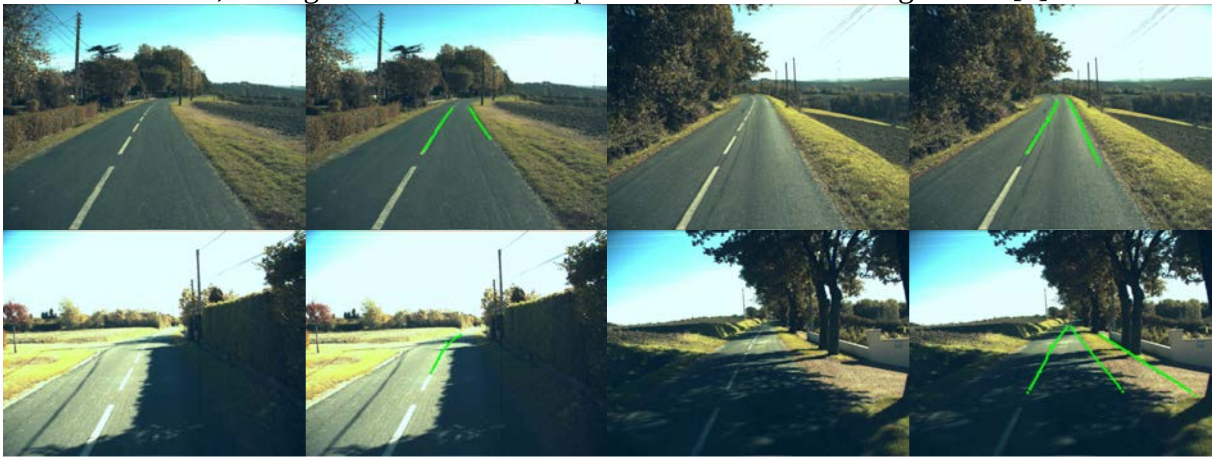


Fig. 10 Experiment result in various conditions

Conclusions

An efficient, real time, and robust algorithm is proposed for detecting lane markers in this paper. It is based on obtaining a top view of the road image, generating the lane marker abstracted region and class points. And lane markers are abstracted based on the verification by Helmholtz principle. The result shows that the algorithm can detect lane marker in various conditions.

But this algorithm assumes that the road is flat and the position and rotation of the camera with respect to the road is known, i.e. the camera is initially calibrated. However, these assumptions may be inaccurate taking into account that cameras are mounted on moving vehicles and the road is far from being perfectly flat. To solve the limitations associated with that assumption is the next.

References

[1] M. Bertozzi, A. Broggi, Real-time lane and obstacle detection on the gold system, In Intelligent Vehicles Symposium, Proceedings of the IEEE. 9(1996)213-218.

[2] Zu Kim, Realtime lane tracking of curved local road, In Intelligent Transportation Systems, Proceedings of the IEEE. 9(2006)1149-1155.

[3] U. Franke, D. Gavrila, S. Gorzig, F. Lindner, F. Puetzold, C. Wohler, Autonomous driving goes downtown, Intelligent Systems and Their Applications, IEEE. 13(1998) 40–48.

[4] U. Franke, I. Kutzbach, Fast stereo based object detection for stop & go traffic, In Intelligent Vehicles Symposium, Proceedings of the IEEE. 9(1996) 339–344.

[5] Jun, Wang, Tao, Mei, Bin, Kong, Hu, Wei, An approach of lane detection based on inverse perspective mapping. 2014 IEEE 17th International Conference on Intelligent Transportation Systems. 17(2014) 35-38.

[6] C. Kreucher, S. Lakshmanan. Lana, A lane extraction algorithm that uses frequency domain features, Robotics and Automation, IEEE Transactions. 15(1999)343–350.

[7] J.C. McCall, M.M. Trivedi, Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation, Intelligent Transportation Systems, IEEE Transactions. 7(2006)20–37.

[8] S. Sivermann, M. Trivedi, Improved video-based lane tracker performance using vehicle localization, IEEE Intelligent Vehicles Symposium. (2010)1221-1226.

[9] Hong Wang, Qiang Chen, Real-time lane detection in various conditions and night cases, Intelligent Transportation Systems, Proceedings of the IEEE. 9(2006)1226–1231.

[10] P. Moghadam, J. F. Dong, Road direction detection based on vanishing-point tracking, inIEEE/RSJ International Conference on Intelligent Robots and Systems. 10(2012)1553–1560.

[11] Roberts, L.G, Machine Perception of Three-Dimensional Solids, Optical and Electro Optical Information Processing. (1965)159-197.

[12] Haralick, R.M., Shapiro, L.G, Computer and Robot Vision, vol. 2. Addison-Wesley, Reading, Massachusetts, 1993.

[13] Jain, R., Kasturi, R., Schunk, B.G., Machine Vision, McGraw-Hill, New York, 1995.

[14] R. Hartley, A. Zisserman, Multiple View Geometry in computer vision, Cambridge University Press, 2001.

[15] Agnes Desolneux, Lionel Moisan, Jean-Michel Morel, Meaningful Alignments, International Journal of Computer Vision. 40(2000) 7-23.

[16] Agnes Desolneux, Lionel Moisan, Jean-Michel Morel, From Gestalt Theory to Image Analysis, a Probabilistic Approach, Springer 2008.

[17] Yongtao Liu, Lang Wei, Jie Qiao, Design of Lane Departure Warning System Based on ADSP-BF561, 2nd International Conference on Electronic & Mechanical Engineering and Information Technology. (2012)1774-1777.