A NOVEL LANE FEATURE EXTRACTION ALGORITHM BASED ON DIGITAL INTERPOLATION

Yifei Wang, Naim Dahnoun, and Alin Achim

Department of Electronic and Electrical Engineering
University of Bristol
Bristol BS8 1UB

ABSTRACT

This paper presents a novel lane edge feature extraction algorithm based on digital interpolation. Zooming towards the vanishing point of the lanes creates a visual effect of driving. The positions of the lanes should not change significantly on the image plane while the vehicle is moving along the lanes. Considering the position information, more accurate lane features can be extracted. A gradient based vanishing point detection algorithm is incorporated to select the zooming area. The proposed algorithm shows outstanding performance on extracting features belonging solely to the lanes from severe noise environment. The algorithm is capable of removing non-relevant features produced by shadows, other vehicles, trees, buildings etc. The extracted feature map was tested with a classic lane detection algorithm, used in LOIS system. The detection results show that the improved feature map is an important factor to the performance of the whole system.

1. INTRODUCTION

Throughout the last two decades, a significant amount of research has been carried out in the area of lane detection. A complete typical model-based lane detection system consists of four parts: lane modelling, feature extraction, detection and tracking. Lane modelling is concerned with the mathematical descriptions that best represent the lanes. Feature extraction aims to find particular lane features such as colour, texture, edge etc. The detection stage then fits the lane model to the feature map and selects the best set of parameters. Lane tracking could then be applied to follow the change of lanes and reduce the system complexity by reducing the search region in the parameter space.

Many lane detection systems have been suggested. However, a robust system which is able to cope with very complex situations is yet to come. [1] presented the Likelihood Of Image Shape (LOIS) lane detection system. The left and right lanes are modelled as two parallel parabolas on the ground plane. The perspective projected model parameters are then estimated by applying the Maximum A Posteriori (MAP) estimator [2] based on the image gradient. It is robust in noise environment. The LANA system [3] is similar to the LOIS system at the detection stage but uses frequency features of the lanes instead of the edges. [4] introduced a system using the B-spline lane model as well as the Canny/Hough Estimation of Vanishing Points (CHEVP) algorithm to locate the vanishing points of the horizontally segmented lanes. The control points are then positioned by the snake algorithm. [5] uses texture anisotropy field as features to segment the lane from the background. The SPRINGROBOT System [6] uses colour and gradient as lane features and the adaptive randomised Hough transform to locate the lane curves on the feature map. [7] presented a lane model based on the lane curve function (LCF). Each lane boundary is represented by two curves, one for the far-field and the other for the near-field. The algorithm uses lane edges as features. For most of the existing systems, the global shape information is only included in the detection stage but not in feature extraction.

This paper focuses on the lane feature extraction stage. The most commonly used feature is the image gradient or the edges. It requires small computational power and results in a sharp transition in the image intensity. Well-painted lane markings produce strong edges at the lane boundaries which benefit the detection of the lanes. However, as the environment changes, the lane edges may not be as strong and could be heavily affected by the shadows, rain etc. The choice of the edge threshold has always been a difficult task and some existing systems chose a very small value or use the gradient directly without thresholding [1, 8]. This means that many unwanted features are included such as edges corresponding to trees, cars, buildings, shadows and so on. The detection of lanes is thus more difficult and time consuming since a large number of outliers are involved. Other lane features, such as textures, have proved to be useful as well [5]. The computation of texture is much more complex than the gradient and it still considers only the local information. As a result, distractive features could also be included.

The proposed feature extraction algorithm considers the characteristics of the lanes and the global shape information. The idea is to gradually zoom towards the vanishing point of the lanes on a single frame in order to simulate the view seen by the driver. The edges of the zoomed images are compared with the original image edges and the previously zoomed edge maps. Most of the irrelevant features can be removed from the edge map after the process. The system block diagram is shown in Figure 1.

Section 2 describes the theory behind the algorithm and concentrates on the effect of digital interpolating a lane image. Section 3 and 4 presents the proposed algorithm in details. Section 5 briefly introduces the detection algorithm used for testing the proposed feature extraction algorithm and Section 6 shows the experimental results.

A.Achim was supported in part by the European Science Foundation through the COST Action TU0702

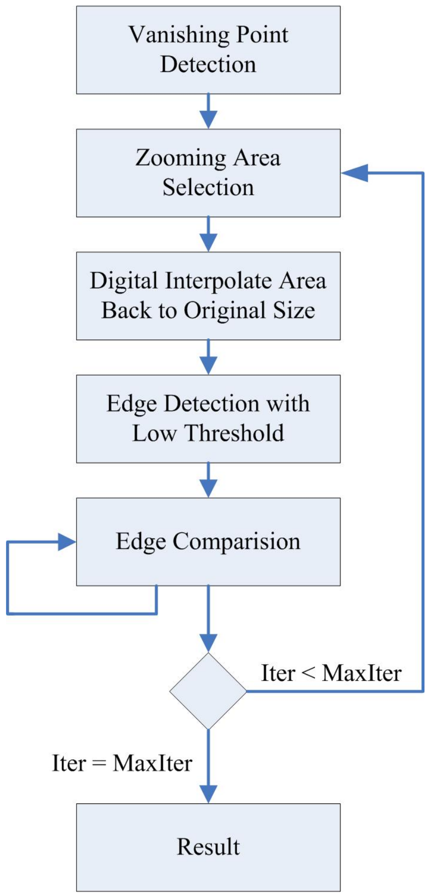

Figure 1: Block diagram of the proposed system.

2. DIGITAL INTERPOLATION ON LANE IMAGES

The purpose of the proposed algorithm is to find the features solely possessed by the lanes from the image. Suppose a vehicle is driven on a clear straight road with continuous lane marking and maintaining a constant lateral offset from the lanes. From the drivers point of view, it is easy to notice that the positions of lane markings does not change over short periods of time. Of course the lane markings are actually moving backwards as the vehicle moves forward. However, since the colour and the width of the markings are similar, the driver is tricked and think the lane markings are not moving. This algorithm takes advantage of the above phenomenon and tries to find the slightly changing features from the scene. However, instead of actually moving along the road, our algorithm is based on a single still image.

In order to simulate the view of driving, digital interpolation is applied. By carefully selecting a region of the image and interpolating this region back to the original image size, simulated view is obtained. All objects on the image will move backwards and their sizes and the positions will change. However, the lane markings or boundaries maintain similar appearances after changing their sizes and positions.

The first task is to select an appropriate area on the image. It is straight forward to see that the vanishing point of the left and right lanes is where the vehicle is heading towards.

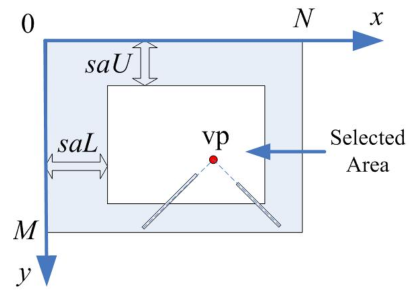

Figure 2: Selected zooming area of an image.

Also, after the interpolation, the vanishing point should stay at the same position. As illustrated in Figure 2, defining the position of vanishing point vp as (vpx, vpy), the total number of image rows as M, and the number of columns as N, the width and height of the selected area could be calculated as:

$$saN = z \times N \tag{1}$$

$$saM = z \times M \tag{2}$$

where z is the zooming ratio and z^2 is the ratio between the area of the selected region and the original image area. The selection of the zooming area must follow the rule that:

$$\frac{vpx}{N - vpx} = \frac{vpx - saL}{saL + saN - vpx} \tag{3}$$

$$\frac{vpy}{M - vpy} = \frac{vpy - saU}{saU + saM - vpy} \tag{4}$$

where saL and saU are the position of the left and upper border of the selected area.

Subsequently, the selected area is interpolated back to the original size. This operation moves all points except the vanishing point to new positions, which are calculated as:

$$x_I(t+1) = vpx + (x_I(t) - vpx) \times \frac{1}{1-z} \tag{5}$$

$$y_I(t+1) = vpy + (y_I(t) - vpy) \times \frac{1}{1-z} \tag{6}$$

where $x_I(t)$, $y_I(t)$ and $x_I(t+1)$, $y_I(t+1)$ represent the x and y coordinate of point I before and after the interpolation respectively. A point on a straight lane boundary before interpolation needs to stay on the same line after interpolation. To prove this, we assume a straight line:

$$y_I = ax_I + b \tag{7}$$

which passes through the vanishing point and a point I. Substitute Equation 7 into Equation 6:

$$y_I(t+1) = a \cdot vpx + b + (a \cdot x_I(t) + b - a \cdot vpx - b)\frac{1}{1-z} \tag{8}$$

which could be rearranged to give:

$$y_I(t+1) = a \cdot \left[vpx + (x_I(t) - vpx) \cdot \frac{1}{1-z} \right] + b \tag{9}$$

Substitute Equation 5 into Equation 9, we get:

$$y_I(t+1) = ax_I(t+1) + b \tag{10}$$

Equation 10 proves that the points on the lane will stay on the same line after interpolation.

So far we have assumed straight lanes and continuous lane markings. However, a multiple vanishing points detection algorithm, along with a iterative zooming process readily solves the problem for the cases of curved lanes and discontinuous lanes. This will be discussed in details in section 3 and 4.

3. VANISHING POINT DETECTION

Vanishing point detection is the first step of the algorithm. Its location is very important for the rest of the task. Although a few pixels variation of the vanishing point position does not significantly influence the system performance, the detected vanishing point has to be corresponding to the lanes. Most of the vanishing point detection algorithms are based on the Hough transform [4, 9, 10]. However, these methods require choosing hard thresholds for both edge detection and Hough space accumulator. It is very difficult to find a suitable set of thresholds for various tasks and environment. In this paper, we assume the road is flat. The vanishing line or the horizon could be calculated using the camera parameters. With this, the detection of the vanishing point is reduced to a one dimensional search.

First, the gradient map is generated by means of a Sobel edge mask. A very small threshold is applied to reduce the computation. The threshold in our case is between 10 and 40 from non-normalised gradient which is small enough to locate lane features under various conditions. Assuming an edge point is belonging to a lane, it is likely that the orientation of this edge is perpendicular to the gradient direction of the lanes. In this case, a line passing through this edge with the direction normal to its gradient orientation is generated to estimate the lane. The intersection of this line and the vanishing line can contribute to the estimation of the vanishing point.

A one dimensional accumulator with a length equals $2N$ ($-0.5N \sim 0.5N$) is created to account for the possibility of the vanishing point being outside the image. Each edge produces a line and each time the line intersects the vanishing line, the corresponding element in the accumulator increments by $(1+gm)$ where gm is the normalised gradient magnitude. The accumulator is then smoothed by a Gaussian low pass filter to compensate the inaccuracy of edge orientation. The element with the most votes corresponds to the vanishing point position. The problem is that if the lane is a curve, the vanishing point position of the lane changes gradually with distance. To solve this problem, the image is partitioned into a few horizontal sections as shown in Figure 3.

The vanishing points for different image sections are detected only using the edge points in the current section. In the far-field, the number of lane edges is comparably lower than that of the near-field. In this case, a tracking process is also included. The search region of the upper section is based on the vanishing point position in the lower sections and the previous vanishing point movement. An example of the multiple vanishing point detection result is given in Figure 3. The vanishing point corresponding to each image band is labelled respectively.

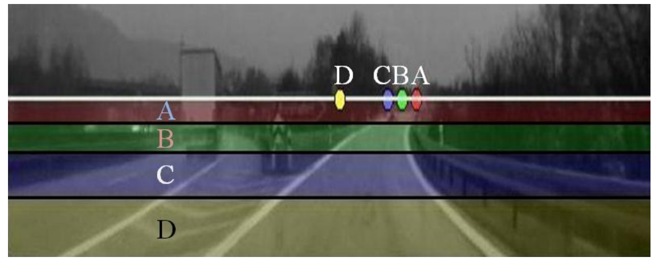

Figure 3: Vanishing point detection result. The image is horizontally partitioned and each partition is labelled. The white horizontal line is the horizon or the vanishing line. The vanishing point of each region is labelled respectively.

4. LANE FEATURE EXTRACTION

The task is now to find a way to extract the lane features. Since the points on the lane will stay on the same line after interpolation, the simplest idea is to apply the 'logical and' operator to the original image edge map and to the interpolated image edge pixel by pixel. This means that if the interpolated edges overlap with the original image edges, these edges are likely to be belonging to the lanes. Furthermore, the orientation of the overlapping edges should be similar. The allowed direction difference is set to be between $0 \sim \pi/2 rads$ in order to tolerate curves, edge detection errors and the orientation change caused by severe shadows. However, unwanted edges have the potential to overlay and have similar orientation as well. In this case, an iterative zooming process is suggested. Based on experiments, 10 iterations of a gradually zooming process are normally sufficient to remove most of the noise even under very severe conditions.

During the interpolation stage, bilinear interpolation is chosen for its low complexity and satisfactory performance. The one dimensional bilinear interpolation between two points (x_0, y_0) and (x_1, y_1) is given by:

$$y = y_0 + \frac{(x-x_0)(y_1-y_0)}{x_1-x_0} \tag{11}$$

In the 2D cases, interpolation is first applied in x-direction then in y-direction.

Edge detection is performed after each interpolation process. The edge map is then compared with the edge map generated by previous iterations. Only the positions occupied by similarly orientated edges throughout the whole process are preserved. Specifically, if the orientation of $I(x,y)_{original}, I(x,y)_1, \ldots, I(x,y)_{iter}$ are similar, then

$$\begin{gathered} I(x,y)_{final} = I(x,y)_{original} \& I(x,y)_1 \& \\ I(x,y)_2 \& \ldots \& I(x,y)_{iter} \end{gathered} \tag{12}$$

Another possibility is to accumulate the overlapping edges and set a threshold to ensure the final edge positions are occupied most of the time.

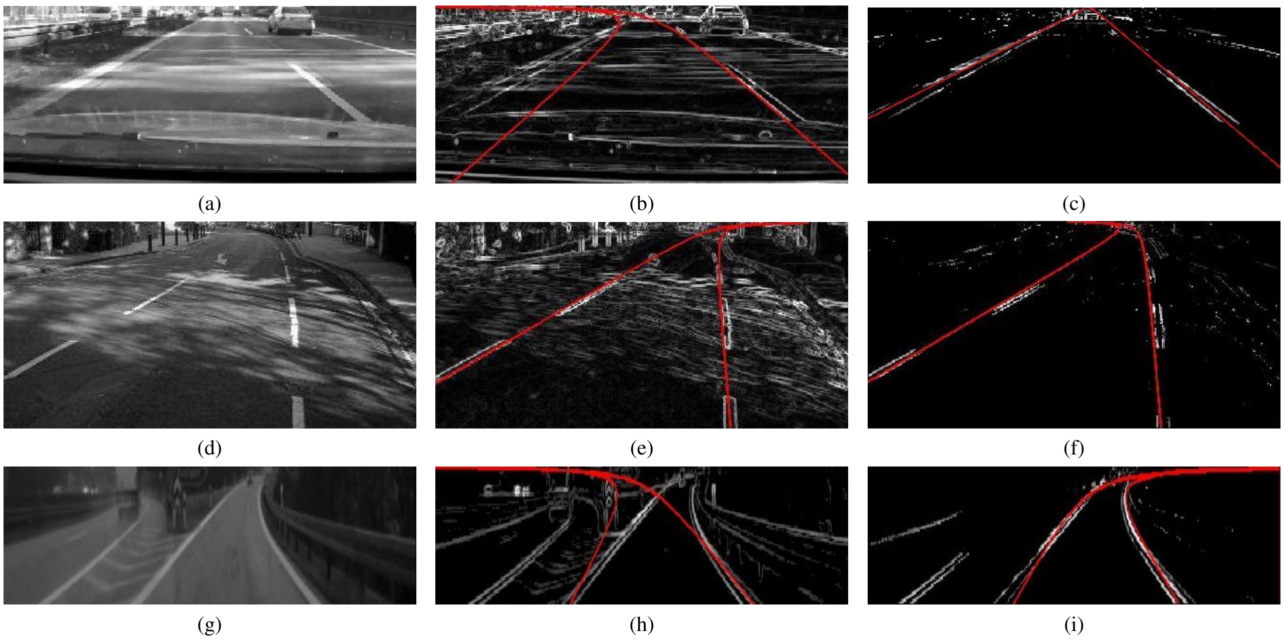

Figure 4: (a), (d) and (g): input images excluding the part above the vanishing line. (b), (e) and (h): detection results of (a), (d) and (g) respectively based on image gradient. (c), (f) and (i): detection results of (a), (d) and (g) respectively based on feature maps generated by the proposed algorithm.

The the zooming ratio, z, is also an important parameter. It is unreasonable to select a very small zooming area. This introduces large image distortion and also causes the system to remove a large number of lane features if the lanes are segmented lines. It has been found by experiment that the minimum degree of zooming should be round 85%. Consequently, a large zooming ratio is applied at each iteration (decremented from 100% to 85%) and only a very small portion of each segmented lane marking will be erased during the process. This allows the algorithm to deal with segmented or dashed lines.

For curved lanes, the vanishing point varies with distance. In this case, the zooming process is separated into different sections. Each section of the image zooms into the corresponding vanishing point. Also, the results of the edge detection on each section are compared separately.

Finally, most of the unwanted features are removed from the edge map and the remaining edges are marked to be ‘1’s. In order to give each pixel a different weighting, the ‘1’s are replaced by the corresponding gradient magnitudes. Furthermore, a weighted sum of the proposed feature map with the original gradient map produces a new feature map with magnified lane features. Some example feature extraction results are shown in Figure 4.

5. DETECTION

The detection of the lanes is implemented using the deformable template matching algorithm proposed in [1]. The lanes are modelled as two parallel parabolas as in the case of on the ground plane, and transformed to the image plane as:

$$x_L = \frac{s_1}{y - vpy} + s_2(y - vpy) + vpx \tag{13}$$

$$x_R = \frac{s_1}{y - vpy} + s_3(y - vpy) + vpx \tag{14}$$

where x_L and x_R are the x-coordinate of the left and right lane model. s_1, s_2 and s_3 are the three parameters need to be determined.

In contrast to LOISs method, by detecting the vanishing points, vpx becomes a known parameter. Specifically, it equals the vanishing point position of the lowest image band. The Metropolis algorithm [11] is applied to iteratively optimise the parameters and maximise the likelihood function.

6. EXPERIMENTAL RESULTS

In this section, we show the assessment of the proposed method. The algorithm (only feature extraction) is successfully implemented in real time on the TMS320DM6437 DSP platform from Texas Instruments. The system is able to achieve above 23 frames per second with a 352×240 video input. The frame rate could be further increased by optimising the code [12, 13]. It is worth noting that only the image gradient map is chosen here for comparison since the proposed algorithm only uses the gradient information during the entire process and could be easily extended to incorperate other features. Therefore, comparison between other types of feature maps is not relevant.

The test images included in this section are chosen from the most difficult scenes and from several video sequences. In Figure 4 (a) and (d), both scenes are affected heavily by shadows. A diverging lane scene is included in Figure 4 (g). The corresponding gradient maps are shown in Figure 4 (b), (e) and (h). All of these gradient maps contain a large number of unwanted feature points. Figure 4 (c), (f) and (i) show the feature map obtained using the proposed algorithm. Most of the unwanted features are removed. Comparing with the gradient maps, the proposed feature maps are much cleaner while the lane features are well preserved.

The detection of the lanes is based on the metropolis algorithm, which does not guaranty to find the global maximum. The parameters update is based on a random selection process. In this case, the detection result varies even based the same feature map. The parameter settings during the detection stage are optimised for both feature maps.

The input images shown in Figure 4 are tested 200 times and the resultant parameters: s_1, s_2 and s_3 from Equation 13 and 14, are compared with the manually selected true parameters. The average absolute error for each of the parameters is calculated. As the required accuracies and dynamic ranges of s_1, s_2 and s_3 are different, the error ratio between the detection results based on different feature maps would be illustrative. Defining the parameter estimation error based on proposed feature map as $EP(s)$ and the parameter estimation error based on gradient map as $EG(s)$. The relationship between $EP(s)$ and $EG(s)$ could be represented as:

$$ER(s) = \frac{EP(s)}{EG(s)} \tag{15}$$

Table 1 shows the ER value corresponding to different parameters calculated from Figure 4 (a), (d) and (g) as well as the detection time ratio T_P/T_G.

	Figure 4(a)	Figure 4(d)	Figure 4(g)
$ER(s_1)$	0.11	0.36	1.06
$ER(s_2)$	0.27	0.99	0.41
$ER(s_3)$	0.38	0.27	0.19
T_P/T_G	0.19	0.18	0.56

Table 1: ER values corresponding to s_1, s_2 and s_3 and the Time ratio T_P/T_G calculated from Figure 4 (a), (d) and (g).

As Table 1 shows, the proposed feature map exhibits significant advantage over the traditional gradient map in extremely noisy environment. The detection processing time based on the proposed feature map is also massively reduced because much less feature points are included.

It is also important to note that sometimes the combination of the two feature maps gives better results since the proposed algorithm removes edges corresponding to very short segments of the lane markings. The weighted sum of the two feature maps (normally a larger weighting for the proposed feature map gives better performance) includes all the features and magnifies the ones that most likely to be belonging to the lanes.

7. CONCLUSION

In this paper, a novel lane feature extraction algorithm has been presented. This algorithm not only uses local information but also includes the global shape information of the lanes. This is achieved by simulating the vision of driving based on digital interpolation. Difficulties were encountered while extracting features from curved and segmented lane markings. However, the problems are solved by a multiple vanishing points detection algorithm and an iterative zooming process. The results of this algorithm show huge advantages over the traditional gradient maps but at the expense of an increased computational complexity (although it does significantly reduce the computational cost needed at the detection stage). Experiments showed that the feature map is very important for the detection stage. Removing unwanted features in noise environment helps the detection algorithm to locate the lane features quickly and the error rate has been reduced.

Our current work is focused on the inclusion of more features, such as texture and colour, in order to increase the performance and reduce the number of iterations needed for noise removal.

REFERENCES

[1] K. Kluge and S. Lakshmanan, "A deformable-template approach to lane detection," in *Proceedings of the Intelligent Vehicles '95 Symposium*, 1995, pp. 54–59.

[2] Steven M. Kay, *Fundamentals of Statistical Processing, Volume I: Estimation Theory*, Prentice Hall Signal Processing Series, 1993.

[3] C. Kreucher and S. Lakshmanan, "LANA: a lane extraction algorithm that uses frequency domain features," *Robotics and Automation, IEEE Transactions on*, vol. 15, pp. 343 – 350, 1995.

[4] Yue Wang, Eam Khwang Teoh, and Dinggang Shen, "Lane detection and tracking using B-snake," *Image and Vision Computing*, vol. 22, pp. 269–280, 2004.

[5] Jinyou Zhang and H.-H. Nagel, "Texture-based segmentation of road images," in *Proceedings of the Intelligent Vehicles '94 Symposium*, 1994, pp. 260–265.

[6] Qing Li, Nanning Zheng, and Hong Cheng, "SPRINGROBOT: A prototype autonomous vehicle and its algorithms for lane detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, pp. 300–308, 2004.

[7] Jong Woung Park, Joon Woong Lee, and Kyung Young Jhang, "A lane-curve detection based on an lcf," *Pattern Recognition Letters*, vol. 24, pp. 2301–2313, 2003.

[8] Dong-Joong Kang and Mun-Ho Jung, "Road lane segmentation using dynamic programming for active safety vehicles," *Pattern Recognition Letters*, vol. 24, pp. 3177–3185, July 2003.

[9] J.A. Shufelt, "Performance evaluation and analysis of vanishing point detection techniques," *Pattern Analysis and Machine Intelligence, IEEE*, vol. 21, pp. 282–288, Mar. 1999.

[10] Yue Wang, Dinggang Shen, and Eam Khwang Teoh, "Lane detection using spline model," *Pattern Recognition Letters*, vol. 21, pp. 677–689, 2000.

[11] I. Beichl and F. Sullivan, "The Metropolis algorithm," *Computing in Science and Engg*, vol. 2, pp. 65–69, 2000.

[12] Naim Dahnoun, *Digital Signal Processing Implementation: using the TMS320C6000 processors*, Prentice Hall PTR, 2000.

[13] Naim Dahnoun, *C6000 DSP teaching ROM*, Texas Instruments, 2nd edition.