IOP Conf. Series: Materials Science and Engineering **297** (2018) 012050 doi:10.1088/1757-899X/297/1/012050

Lane detection using Randomized Hough Transform

Peerawat Mongkonyong[1,*], Chaiwat Nuthong [1], Supakorn Siddhichai[2] and Masaki Yamakita[3]

[1] International College, King Mongkuts Institute of Technology Ladkrabang (KMITL), Bangkok 10520, Thailand
[2] National Electronics and Computer Technology Center (NECTEC)
[3]Tokyo Institute of Technology, Tokyo (Tokyo Tech)

* Corresponding Author: 56610013@kmitl.ac.th, 0822087697

Abstract. According to the report of the Royal Thai Police between 2006 and 2015, lane changing without consciousness is one of the most accident causes. To solve this problem, many methods are considered. Lane Departure Warning System (LDWS) is considered to be one of the potential solutions. LDWS is a mechanism designed to warn the driver when the vehicle begins to move out of its current lane. LDWS contains many parts including lane boundary detection, driver warning and lane marker tracking. This article focuses on the lane boundary detection part. The proposed lane boundary detection detects the lines of the image from the input video and selects the lane marker of the road surface from those lines. Standard Hough Transform (SHT) and Randomized Hough Transform (RHT) are considered in this article. They are used to extract lines of an image. SHT extracts the lines from all of the edge pixels. RHT extracts only the lines randomly picked by the point pairs from edge pixels. RHT algorithm reduces the time and memory usage when compared with SHT. The increase of the threshold value in RHT will increase the voted limit of the line that has a high possibility to be the lane marker, but it also consumes the time and memory. By comparison between SHT and RHT with the different threshold values, 500 frames of input video from the front car camera will be processed. The accuracy and the computational time of RHT are similar to those of SHT in the result of the comparison.

1. Introduction

Nowadays, demands on vehicles are increasing because of the rapid population growth. This leads to the major causes of higher road accidents [1]. Nearly 3,500 people die on the road every day. It is found that the lane changing is the cause of most accidents. The advantage of LDWS should be an optional tool to relieve such disaster. LDWS is one of the methods that are proposed to solve the problem. LDWS uses the image processing mechanism to detect the lines of the image, selects the current lane markers, warns the driver when the vehicle leaves the current lane and tracks the lane marker for the next frame. In this article, overviews of SHT and RHT are presented and the algorithms are briefly introduced and compared.

Kalviainen et al. [2] presented the probabilistic and non-probabilistic Hough transforms overview and comparisons. They compared a lot of algorithms such as SHT, RHT, Dynamic RHT and Window RHT. They tested line detection with synthetic and real-world images demonstrating the high speed and low memory usage. The result of the comparison was a good model for the road image.

Lim et al. [3] performed lane-vehicle detection and tracking. Vertical mean distribution was used to remove the sky region by averaging the gray values of each row on the image. A big jump of this method indicated the line that divided the image into sky region and road region. But the result would be errorneous if some parts of the car appeared in the image such as a console. Because the averaged value would have changed. To solve this problem, the vanishing point detection was used for this section.

Schreiber et al. [4] presented the lane detection that the vanishing points were discussed. Vanishing points were detected using Hough Transform (HT). This method was used the vanishing point to detect straight line segments. But the vanishing point also can remove the unwanted area to reduce the computational time. In edge detection, Canny edge detection was used in image preprocessing section. In term of computational time, some type of edge detection was better than Canny edge detection [5].

Kultanen et al. [6] introduced the RHT. The mechanisms of a random sampling of point pairs from the x-y plane, converted to the ρ-θ plane and voted the accumulator. In the lane marker detection, the accuracy of this method can increase by assuming one point of the pair to be the vanishing point. Because the vanishing point was the part of the straight line in the image.

Wang et al. [7] proposed a B-Snake based lane detection and tracking algorithm without any parameters of the camera. This article used Canny/Hough estimation of vanishing points (CHEVP) to provide a good initial position for the B-Snake. The snake got drawbacks. It often gets stuck in local minima states which can be counteracted by simulated annealing techniques. Minute features were often ignored during energy minimization over the entire contour. And their accuracy was governed by the convergence criteria used in the energy minimization technique.

To solve the problems, a lane detection is proposed in this article as outlined by the flow diagram in figure 1. Lane boundary detection is divided into 3 parts. The image that obtained from the camera has many ingredients. The image preprocessing is applied to make it appropriate for processing. Image preprocessing is performed on the road image to locate the left-right lane edges by separating the sky region and analyzing road region known as the region of the interest (ROI) in this article to extract the prominent road features such as lane markers. The information of the vanishing point is used to limit the area in line detection and refer the point in RHT. Lane marker detection part attempts to identify the lane marker from the detected lines.

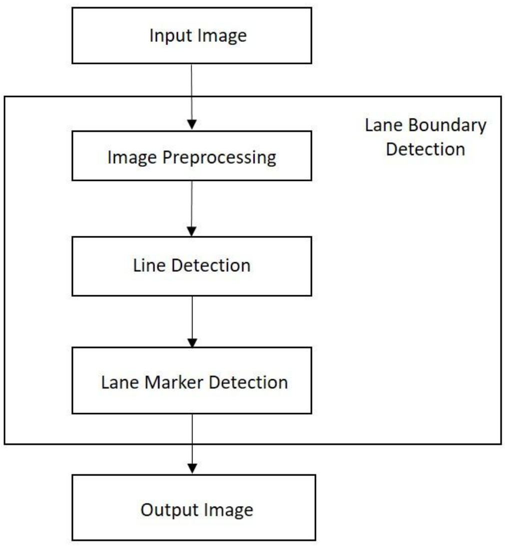

Figure 1. System overview.

This article is organized as follows. Section 2 discusses about the lane boundary detection and explains the detection technique. Some simulation results are shown in section 3 followed by the conclusion and future works.

IOP Conf. Series: Materials Science and Engineering **297** (2018) 012050 doi:10.1088/1757-899X/297/1/012050

2. Lane Boundary Detection

Lane boundary detection consists of 3 parts, i.e. image preprocessing prepares a proper image for further processing from the original image as shown in figure 2, line detection to extract the lines contained in the images and lane marker detection to detect the current lane markers.

Figure 2. Original image.

2.1. Image preprocessing

The proposed image preprocessing includes the grayscale image conversion, image filtering, image morphology as shown in figure 3, and sky region removing which applies some technique to remove some part of the image outside the ROI. This can reduce the computational time in the vanishing point detection method. In this article, the sky region removing removes the upper half of the image as shown in figure 4. Consuming some part of the road surface is unnecessary because this method focuses on the lane markers.

Figure 3. The result of image preprocessing.

Figure 4. The result of sky region removing.

2.2. Line detection

The line detection part consists of the edge detection that includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply as shown in figure 5. Many edge enhancement algorithms have been proposed. The Sobel edge detection is used for this article because it is better than Canny edge detection in term of being less time-consuming [5]. After edge detection method, the image contains only white and black pixels.

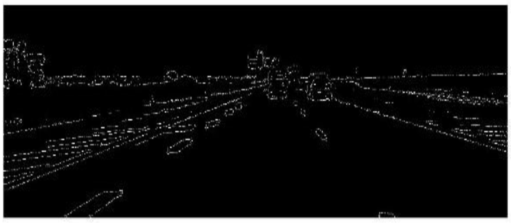

Figure 5. The result of edge detection.

A vanishing point is a point intersection of the projections of parallel lines in the image as shown in figure 6. HT can evaluate the vanishing point by detecting the lines and the intersection point of those lines in the image.

The objective of HT is to find lines in a given image. They are 2 types of HT to be considered in this article, i.e. SHT and RHT. They are different mainly in the resource usage, which is one of the concerned issues for the researchers.

2.2.1. Standard Hough Transform. The SHT computation consists of calculating the parameter and accumulating the data in the accumulator, finding the local maxima which represent line segments of the image and extracting the line segments from the maxima positions. The disadvantages of SHT are its computational complexity and huge storage memory consumption.

It assumes all of the white pixels of the image to be the points and converts them into ρ-θ plane. ρ is the line connecting the polar coordinate to the origin where the x-axis intersects the y-axis. θ is the angle between ρ and x-axis as shown in figure 7. Each point in the x-y plane will have the infinity lines that pass the point. Those lines are converted by equation (1).

$$\rho = x_0 \cos\theta + y_0 \sin\theta, \tag{1}$$

where x_0 and y_0 are the coordinates corresponding to the angles θ (x_0 , y_0). The line plot of those lines will be the sets of point in the ρ-θ plane. So, the point connection will be the sinusoidal wave for each point in the x-y plane as shown in figure 8. The intersection of lines in ρ-θ plane means they have the same value of ρ-θ. By voting θ in the accumulator will get the set of points in the x-y plane that has the high possibility to be on the same line.

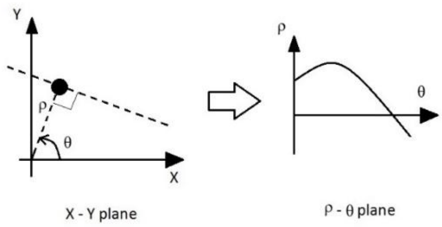

Figure 6. Principle of Hough transform. A point in x-y plane is converted into a curve in ρ-θ plane.

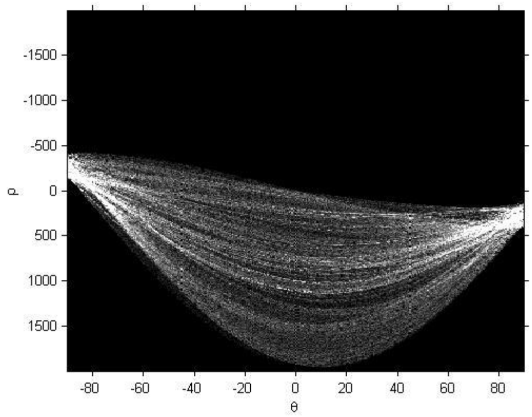

Figure 7. Data voting in ρ-θ plane.

Figure 8. The result of vanishing point detection.

2.2.2 Randomized Hough Transform. On the contrary, The RHT computation consists of selecting 2 random pixels from edge image and calculating the parameter from the line of the point connection as shown in figure 9, accumulating and voting the data in the accumulator, extracting the line segments from that data. The RHT algorithm lies in the fact that each point in ρ-θ plane can be expressed with 2 points or 1 line from the original binary edge image. The advantage of RHT are high parameter resolution, the infinite scope of the parameter space, small storage requirements and high speed.

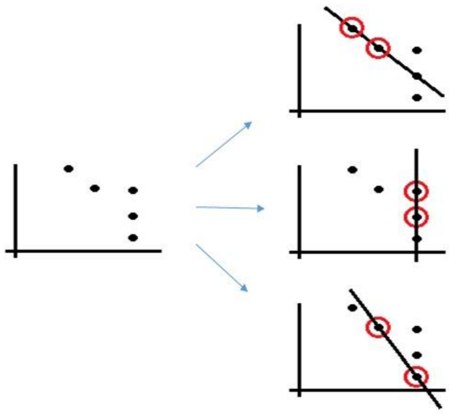

Figure 9. The random point pair method in RHT.

A line detection using the RHT will pick two points from the edge detection. The first point is the vanishing point and the second point is the random point. The random type is uniformly distributed pseudorandom integers. Then, create the line from that two points using linear equation. This method also uses equation (1) for the conversion. After that, the data are accumulated in the accumulator. The vote has 2 types. Firstly limiting the number of input θ and picking the highest one after voting. Secondly, limiting the vote and picking the first θ that is equal to the vote. This article uses the second type because the user can adjust the computation time. On the other hand, SHT uses the first type to vote the accumulator but the limit of the input is all of the data. So, the computation time of RHT should be lower than SHT because it uses only some points of the image.

The main difference between SHT and RHT is that while in SHT a single pixel in the original image is mapped to a curve in ρ-θ plane, in RHT a pair of pixels is mapped to a single cell in ρ-θ plane. SHT generates all parameter combinations, but RHT generates only a small subset of all parameter combinations.

After that, the area removing uses the information of the vanishing point to further remove the unwanted area. Changing the value of that pixel to 0 (black pixel), it can reduce some computational time and some error for this method as shown in figure 10. After the vanishing point and other 2 points at the corner are specified. The corresponding linear equation can be found, i.e.

$$y = mx + c, \tag{2}$$

where m is the slope and c is the y-axis intersection. The vanishing point also can be used to divide the image into the left-hand side and the right-hand side. This method detects the lane marker of two sides as shown in figure 11 and figure 12.

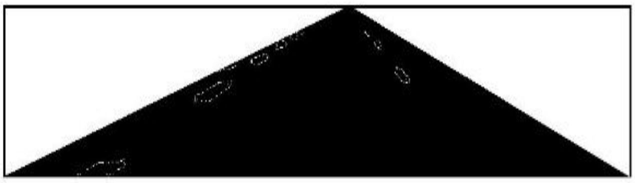

Figure 10. The result of area removing. The image shows only the ROI.

Figure 11. The right-hand side image that divided by the vanishing point.

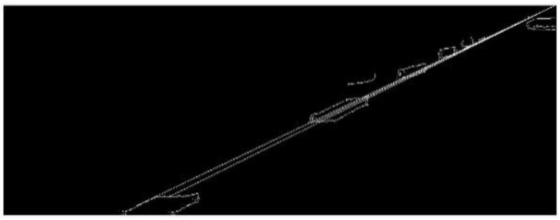

Figure 12. The left-hand side image that divided by the vanishing point.

In case of curve and junction, the difference between straight and curved road is some area under the vanishing line as shown in figure 13.

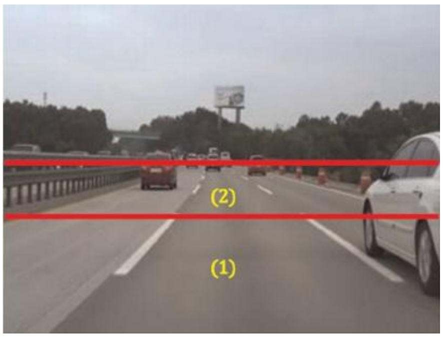

Figure 13. (2) is the area affected by curved road and (1) is the area not effected by curved road.

However, the high possibility lines would be the lane markers because of the number of the lane marker pixels.

2.3. *Lane marker detection*

After line extraction, the lane marker detection will select the lowest θ value of each side. They are assumed to be the lane marker as shown in figure 14.

Figure 14. The lowest θ will get the high possibility to be the lane marker.

Generally, the lane marker should be located at the middle of the road surface and bottom of the image. For each side, the system needs to find at least two different θ's from the vote and compares them. The lowest θ will have the high possibility to be the lane marker of that side.

3. Result

The system runs on MATLAB R2013a on ASUS K550J CPU Intel Core i7-4710HQ, up to 3.5 GHz RAM 8 GB. A video input was captured under clear condition on a sunny day about 4 p.m. in a highway environment using iPhone 6 camera.

The accuracy in table 1 is calculated from the corrected image of 500 frames input video. The corrected image is the image that has the edge of current lane region on the lane marker as shown in figure 15. The result of computational time is from MATLAB function that calculates the processing time of the project.

IOP Conf. Series: Materials Science and Engineering **297** (2018) 012050 doi:10.1088/1757-899X/297/1/012050

Table. 1 The comparison table.

Type	Accuracy (%)	Computational time (s)
1. SHT	93.0	150.146
2. RHT (threshold = 5)	93.6	142.488
3. RHT (threshold = 10)	93.8	142.639
4. RHT (threshold = 15)	95	146.587
5. RHT (threshold = 20)	95.2	150.047

Figure 16 shows that the pixels of the shadow are longer than the right lane marker. This means that the system has the higher possibility to randomize the point on the shadow and assume it to be a lane marker. To solve this problem, some additional methods for image preprocessing such as image filtering, another type of edge detection and the threshold of RHT can be applied. However, this would increase computation time of the system.

Figure 15. The result of the corrected image.

Figure 16. The mistake from the detection because of the heavy shadow.

The threshold increasing in RHT will increase the accuracy and computational time. At the higher threshold, the value of accuracy tends to be stable as shown in figure 17. On the other hand, the system tends to consume a lot of computational time in the higher threshold state as shown in figure 18.

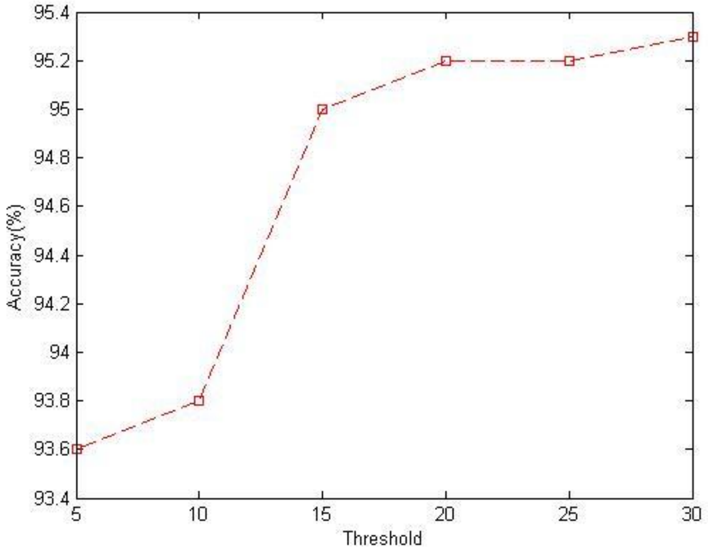

Figure 17. The accuracy of each threshold in RHT.

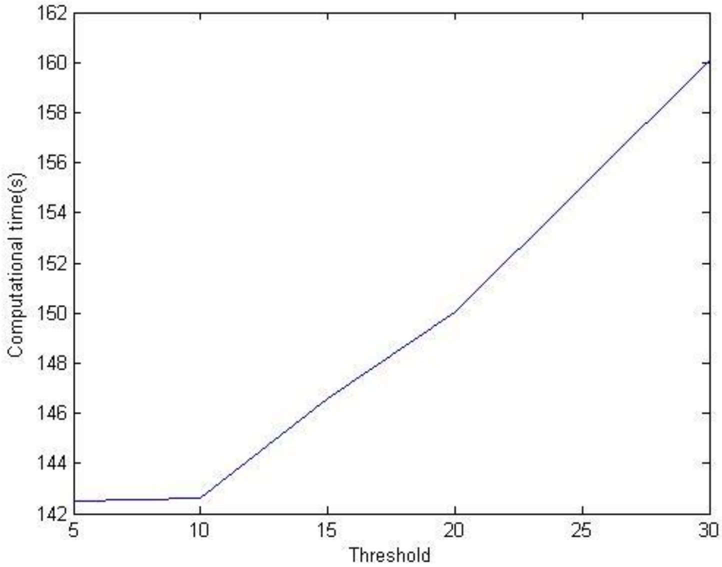

Figure 18. The computational time of each threshold in RHT.

4. Conclusion

Lane detection has been presented in this article image preprocessing, line detection and lane marker detection. It has been shown that the main advantage of the RHT method over the SHT is computational time. The difference in computational time between SHT and RHT with 5 threshold is about 5.1 percent, RHT with 10 threshold is about 4.99 percent, RHT with 15 threshold is about 2.73 percent and RHT with 20 threshold is about 0.06 percent, according to table 1.

The system still has two further parts to complete, i.e. the warning part that finds the condition of lane changing and the tracking part that reduces the computational time of the current frame using the information of the previous frame.

In the future work, the developer can develop the system into the application on a smartphone, which can be attached to the car front.

References

[1] Royal Thai Police, Thailand (2006). *The situation of traffic accident cause of the accident by a person and environment causes of the equipment used in driving, whole kingdom*, URL: service.nso.go.th/nso/web/stat-series/tables/00000_Whole_Kingdom/14.18.2\\A1.xls, accessed on 23/02/2017

[2] Heikki Kalviainen, Petri Hirvonen, Lei Xu and Erkki Oja 1995 *Image and Vision Computing* **13** pp 239 – 252

[3] King Hann Kim, Li-Minn Ang, Kah Phooi Seng and Siew Wen Chin 2009 *IMECS* **2** pp 1124 – 1129

[4] David Schreiber, Bram Alefs and Markus Clabain 2005 *IEEE Int. Conf. on Intelligent Transportation Systems* pp 1114 – 1119

[5] Raman Maini and Dr. Himanshu Aggarwal 2009 *IJIP* **3** pp 1 – 12

[6] Pekka Kultanen, Lei Xu and Erkki Oja 1990 *IEEE* **1** pp 631 – 635

[7] Yue Wang, Eam Khwang Teoh and Dinggang Shen 2004 *Image and Vision Computing* **22** pp 269 – 280