Line Detection Using Arranging Coordinate Point Method

Rumaisa Ramadhani
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
rumaisa.ramadhani@students.itb.ac.id

Arief Syaichu Rohman
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
ariefsyaichu@lskk.itb.ac.id

Yulyan Wahyu Hadi
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
yulyan@lskk.ee.itb.ac.id

***Abstract*—Line detection system is a system line that can recognize the line mark painted on the road. This is one of the digital driver assistance tools that improve driving safety. The video streaming captured using a camera which is installed in front of the vehicle as the input system will detected the line by using the Arranging Coordinate Point Algorithm. The system will provide the correction value of the vehicle distance from the center of the road and guide the driver to stay on the track. Based on the experiment result, the system could detect a straight and curved line. The line is well detected by the system in good condition such as less of noise of the road, good weather, and clear lane line. The system has a computational process 0.0625 fps with average error calculation of position from the center of the road is 0.0992 m and standard deviation is 0.62448 m.**

Keywords— Line Detection, Digital Assistance, Autonomous Driving Car, Arranging Coordinate Point, Video Streaming.

I. Introduction

Human error is the most causes of traffic accidents, e.g. distracted driver, excess speed and out off the track. To reduce the number of traffic accidents, the car should be equipped with a safer and safety features. Nowadays there are many driver assistance system (DAS) developed, such as camera-assisted system which takes real time video from surroundings and displays relevant information to the driver that can be used to help preventing driver's mistakes and reduce traffic accidents effectively [1].

The camera is mounted in front of the vehicle to take real time video. The camera is tilted towards the ground at 45 degree, so that the camera capture video of the road only [2].

(1a)

(1b)

Fig. 1 Camera Position (1a) and Image captured from Camera (1b)

Figure (1a) show the camera position in the vehicle. The camera is tilted towards the ground. Figure (1b) is the image captured from camera. The image represents 0.045 cm in the real condition.

The video streaming was recorded as the input system. The video was divided into frames and each frame would have a image filtering process that will filter the frame using HSV color filter, contour filter and ROI filter so it can be implemented to detect the line by using Arranging Coordinate Point Algorithm. Then the algorithm will provide the correction value of the vehicle distance from the center of the road and the system will guide the driver to stay on the track.

This following block diagram show you about the general process of the system.

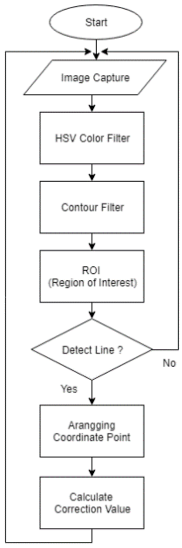


Fig. 2 General Process of System

II. Image Filtering

Image filtering is the fundamental process in image processing. Image filtering is the process of removing noise and removing properties from the image that we don't need. Image filtering is not only used to increase the image quality, but also selectively extract certain aspects of image that are considered as important information in the context of given application [3]. To identify the line, we used HSV Filter to segment the color of image and Contour Image Filter to select the region of line. Region of Interest (ROI) method is used to remove unnecessary things around the road.

A. HSV Filter

Color image segmentation is more utilizable than gray scale image segmentation because of its capability to enhance the image analysis process for improving the segmentation result. But when we consider color image segmentation, then choosing a proper color space becomes the most important issue. HSV color space is one of frequently chosen color space [4].

We can represent HSV color space with help of a hexacone in three dimensions where the central vertical axis represents the intensity [5]. The 'Hue' is an angel in the range [0, 2π] relative to the red axis with red at angel 0, green at 2π/3, blue at 4π/3 and red again at 2π [6] [7]. The 'Saturation' represents how pure the hue is with respect to a white reference. This can be thought of as the depth or purity of color and is measured as radial distance from the central axis with values between 0 at the center to 1 at the outer surface[7]. The 'Value' is represented by the amount of light illuminating a color which is percentage from 0 to 100 [8].

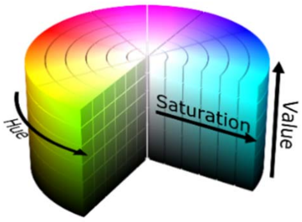


Fig. 3 HSV Color Space

In [7], the authors analyzed the properties of the HSV color space with emphasis on visual perception of the variation in Hue, Saturation and Intensity value of an image pixel. The segmentation using this method gives better identification of objects in an image compared to those generated using RGB color space. Following figures show the RGB original image converted in to HSV Color space.

(4a) (4b)

Fig. 4 Original Image (4a) and Image HSV Result (4b)

B. Contour Image

Contour image filter was designed to segment objects with predefined size and simple shape characteristics using the ratio of area and perimeter, and the second central moment of the contour. These priors allow targeted selection of objects by their size and shape with certain flexibility [9]. In this paper we used contour filter to remove object that have size out of the range of line size.

In [10], for a 2D continuous function $f(x, y)$, the moment (sometimes call "raw moment") of order $(p + q)$ is define as:

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p \, y^q \, f(x, y) \; dx \, dy \qquad (1)$$

For $(p, q) = 0,1,2, \ldots$

Adapting this to scalar (grayscale) image with pixel intensities $I(x, y)$ raw image moments M_{ij} are calculated by:

$$M_{ij} = \sum_x \sum_y x^i \, y^j \, I(x, y) \qquad (2)$$

Central moments are defined as:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p \, (y - \bar{y})^q \, f(x, y) \; dx \, dy \qquad (3)$$

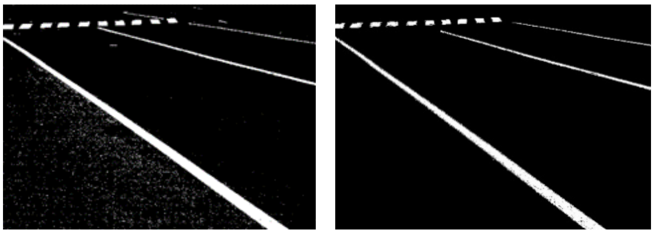
where

$$\bar{x} = \frac{M_{10}}{M_{00}} \qquad \bar{y} = \frac{M_{10}}{M_{00}} \qquad (4)$$

are the component of centroid. If $f(x, y)$ is digital image, then the previous equation become:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p \, (y - \bar{y})^q \, I(x, y) \qquad (5)$$

This following figure show the result of Contour Filter area:



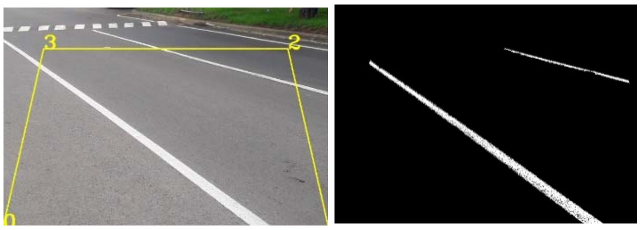
(5a) (5b)

Fig. 5 Image HSV Result (5a) and Image Contour Area Result (5b)

In the figure (5a) there are white spot on the side of line. The white spot is caused by imperfect HSV filtering process, and it can distract the line detection process. To make the line detection process clearly and easily, the white spots is removed by using contour area filter. Contour filter will remove object that have size out of the range of line. Figure (5b) show the image result of contour area filter, the figure show that the white spot have been removed and the line can be detected clearly.

C. Region of Interest (ROI)

ROI (Region of Interest) was designed to remove the interference of unnecessary information, noise from the image and improving road recognition accuracy [1]. The size and range of ROI depends on the area of object that we want to process. In this case, the size of ROI was set to the average lane width of 2.5 meters with average length of 1-2 meters using highway perspective.



(6a) (6b)

Fig. 6 ROI Area (6a) and The Result of ROI (6b)

In figure (6a) the yellow line is ROI area, which means the area out of the yellow line will be removed. Figure (6b) is the result of masking area between contour filter image and ROI area and that figure will be processed to detect the line.

III. Line Detection Algorithm

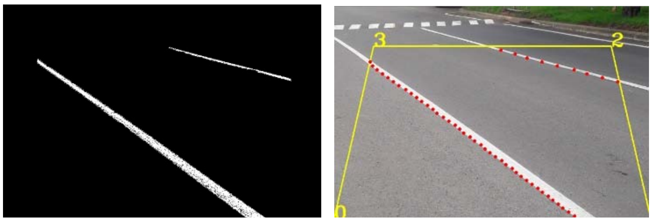
In this section we will discuss the process to detect line. There are 3 main steps to detect the line: scanning line, arranging coordinate point and prediction line.

A. Scanning Line

Before identifying the line, we should determine the number of sample point. Sample point will be used as the location for checking the line. So, we don't need to check all pixels of image, it will reduce the computational complexity, and increase the speed of operation. The number of sample point affects the level of precision of detecting the line.

$$Sampling\ Point\ =\ \frac{ROI\ Result\ Image}{Number\ of\ desired\ Sampling\ Point} \qquad (6)$$

After determining the sample points, the scanning process will be carried out vertically and horizontally from pixel (0,0) to the last pixel based on sampling point. If the sample point is detected as a point line, the coordinate value at that point will be stored. This following picture is the result of line scanning process.



(7a) (7b)

Fig. 7 Binary Image (7a) and Line Detection Result (7b)

This is the stored value of the pixel coordinates detected as a line from the image above.

Table. 1 Stored Value of Coordinate Point which detected as line

Point	Coordinate Value	
	1	2
1	[0,0]	[0,0]
2	[337, 88]	[0,0]
3	[366, 96]	[0,0]
4	[390, 104]	[0,0]
5	[73, 112]	[420, 112]
6	[78, 120]	[451, 120]
7	[87, 128]	[485, 128]
8	[98, 136]	[511, 136]
9	[109, 144]	[542, 144]
10	[120, 152]	[573, 152]

B. Arranging Coordinate Point

Arranging Coordinate Point is an algorithm which design to arrange the stored value of pixel coordinate which detected as line. The main point of this algorithm is comparing the value of coordinate. We know that coordinate have two value (x, y), this algorithm will compare the value of "x" coordinate from the point which is detected as line with the reference point.

The first coordinate point with the highest number of columns is used to be reference point to arrange the others coordinate points. After the reference point was known, the process comparing value of pixel coordinate is started. The value of "x" coordinate from the reference point is compare with the value of "x" coordinate that will be identified. If the difference value between that two points out of the range, the coordinate value will shift, the comparison process will be carried out again and value of previous point become zero.

Based on data from table 1, This is the value of the pixel coordinates detected as a line which have been arranged

Table. 2 Value of Pixel Coordinate Detected as line

Point	Coordinate Value	
	1	2
1	[0,0]	[308,80]
2	[0,0]	[337, 88]
3	[0,0]	[366, 96]
4	[0,0]	[390, 104]
5	[73, 112]	[420, 112]
6	[78, 120]	[451, 120]
7	[87, 128]	[485, 128]
8	[98, 136]	[511, 136]
9	[109, 144]	[542, 144]
10	[120, 152]	[573, 152]

Table 2 show that the value of coordinate point has been arranging. The value of "x" coordinate in the first column is smallest than the second column. The [0,0] coordinate means that there is no line detected on that point.

C. Prediction Line

Some of lane line on the road are merging. To make the detection process running well, we should predict the next coordinate point. Formulation of line equation through two points is used to predict the next coordinate of line.

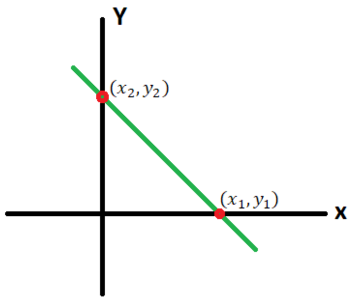


Fig. 8 Line Equation Through Two Points in Cartesian Coordinates

$$\frac{y-y_1}{y_2-y_1} = \frac{x-x_1}{x_2-x_1} \quad (7)$$

D. Correction Position

To make the vehicle stay in the center of the road, we should update the position of vehicle against line. When the vehicle goes too left or right side, the system will calculate the correction value position to make the vehicle stay on the center. This is the illustration of the correction position calculation.

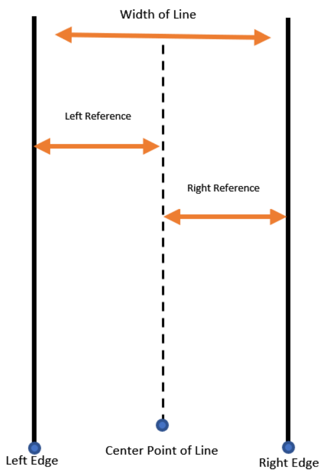


Fig. 9 Error Point Illustration

There are 3 condition of correction position calculation. First when the system detected 2 line (it usually happen when the vehicle runs on straight road or sometimes in curve road), the other condition is when the system only detected one line of the road, right line or left line (it usually happen when the vehicle run on turn right or turn left road or sometimes in curve road). Let's see this formulation:

Center of frame = width frame / 2 (8)

Center of line = width line / 2 (9)

1. Detected 2 Line
   Correction Value = Center Line – Center Frame

2. Detected Right Line
   Reference Right Edge = Center Line – Right Edge

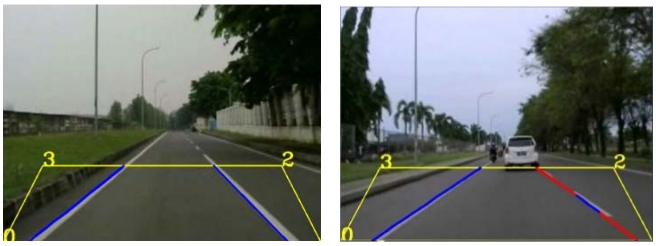
3. Detected Left Line
   Reference Left Edge = Left Edge – Center Line

IV. Experiment Results and Discussion

The lane line detection system will be tested in urban area. Some parameters that will be tested are accuracy of detection and position correction. This experimental was held in MM2100 Industrial Area. MM2100 was chosen as the experimental place because the condition is not too crowded and have a consistent road size in some areas. For this experimental we use the road which has 2,5 m of width. Camera Logitech C270 which have 1280 x 720 screen resolution for video streaming is used as input system. The camera is installed in the center of the vehicle and mounted tilted towards the ground. This is the scenario of the test:

To test the performance of line detection, the vehicle will travel around the industrial area with speed around 60km/hour. The condition of line that will be tested are straight line and curve line.

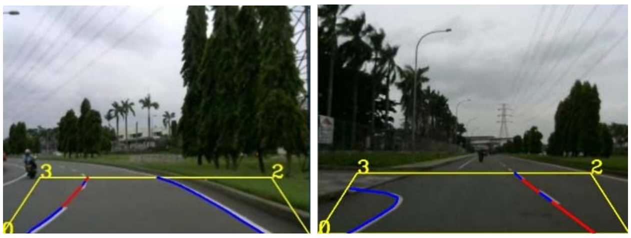
This following picture show the result of accuracy line detection:



(10a) (10b)

Fig. 10 Straight Line Detection

In figure (10 a) and (10b) we can see that the system can detect the straight line very good. The blue line represents the result coordinate of scanning line and the red line represents the result of coordinate point prediction using line equation. In figure (10 b), the coordinate position prediction is good.



(11 a) (11b)

Fig. 11 Curve Line Detection

Figure (11 a) and (11b) show that the system can detect the curve line quite good. Figure (12b) show that the system also can make a good prediction of coordinate line point.

To test the correction coordinate value of line detecting system, we used two scenarios. The first scenario is used to validate the result value of calculation with the actual condition. First, we set the vehicle in the center of the road, so for the first trial the system will detect 2 line. This is the fundamental step, because with this step the system will know the reference point (reference point is used to calculate the correction position when the detected line is only one). After that, we move the vehicle in some distance. So we will know the result of correction position value and then the data from system with the manual measurement.

This is the sample result of correction position when the system detected 2 line:

Table. 3 Sample Result of Correction Position System when Detected 2 Line

System (m)	Actual (m)	Error
0.096	0.1	0.004

0.183	0.2	0.017
0.271	0.3	0.029
0.374	0.4	0.026
0.501	0.5	0.001
0.085	- 0.1	0.015
0.156	- 0.2	0.044
0.228	- 0.3	0.072
0.287	- 0.4	0.113
0.355	- 0.5	0.145

From this experimental data, we know that the position correction value has a value average error 0.0466 and standard deviation 0.0689.

This is the sample result of correction position value when the system only detects the right line:

Table. 4 Sample Result of Correction Position System when Only Detected the Right Line

System (m)	Actual (m)	Error
0.087	0.1	0.091
0.169	0.2	0.031
0.269	0.3	0.031
0.364	0.4	0.036
0.458	0.5	0.042
0.086	- 0.1	0.014
0.171	- 0.2	0.029
0.278	- 0.3	0.022
0.328	- 0.4	0.072
0.387	- 0.5	0.011

From this experimental data, we know that the position correction value has a value average error 0.0379 and standard deviation 0.0472.

This is the Sample result of correction position from left line:

Table. 5 Sample Result of Correction Position System when Only Detected the Left Line

System (m)	Actual (m)	Error
0.092	0.1	0.008
0.187	0.2	0.013
0.274	0.3	0.026
0.383	0.4	0.017
0.476	0.5	0.024
0.082	- 0.1	0.018
0.145	- 0.2	0.055
0.211	- 0.3	0.089
0.326	- 0.4	0.074
0.403	- 0.5	0.097

From this experimental data, we know that the position correction value has a value average error 0.0421 and standard deviation 0.0556.

The second scenario in the vehicle is travel around the Industrial Area with average speed 44,3 Km/hour. The second scenario in the vehicle is travel around the Industrial Area with average speed 44,3 Km/hour.

This is the result from the experiment:

Table. 6 The correction value when the system travel around

System (m)
0.087
0.095
0.131
0.230
0.525
0.683
0.718
0.852
0.633
0.547
0.480
0.283
0.175
0.089
0.091

V. CONCLUSION

Line Detection System uses a camera as one of the digital driver assistance tools that improve driving safety. The system will provide the correction value of the vehicle distance from the center of the road. Some experiment was performed for testing the accuracy of the line detection and vehicle position to the center of the road. The experiment was done using a manual car in good weather in an area with a clear lane line and less intensity of the crowd. The results showed that this system could detect a straight and curved line. If we compare with Hough Transform method, Line Detection System using Arranging Coordinate Point Algorithm could represent a line with good computation process time 0.0625 fps with average error calculation of position from the center is 0.0992 m.

REFERENCES

[1] G. Kaur dan A. Chhabra, "Curved Lane Detection using Improved Hough Transform and CLAHE in a Multi-Channel ROI," *Int. J. Comput. Appl.*, vol. 122, no. 13, hal. 32–35, 2015.

[2] S. Habib dan M. A. Hannan, "Lane departure detection and transmisson using hough transform method," *Prz. Elektrotechniczny*, vol. 89, no. 5, hal. 141–146, 2013.

[3] N. Patel, A. Shah, M. Mistry, dan K. Dangarwala, "A Study of Digital Image Filtering Techniques in Spatial Image Processing," *Int. Conf. Converg. Technol.*, hal. 1–7, 2014.

[4] D. J. Bora, A. K. Gupta, dan F. A. Khan, "Comparing the Performance of L * A * B * and HSV Color Spaces with Respect to Color Image Segmentation," no. June, 2015.

[5] A. Journal, A. Sciences, dan S. Publications, "Clustering and Watershed Algorithms," vol. 8, no. 12, hal. 1349–1352, 2011.

[6] Linda G. Shapiro and George C. Stockman, "*Computer Vision*, " Prentice Hall; 1 edition, 2001.

[7] S.Sural, G. Qian, and S.Parmanik "Segmentation and Histogram Generation Using The HSV Color Space for Image Retrieval," hal. 589–592, 2002.

[8] http://www.dig.cs.gc.cuny.edu/manuals/Gimp2/Grokking-the-GIMP-v1.0/node51.html

[9] J. Moinar, A. I. Szucs, C. Molnar, dan P. Horvath, "Active contours for selective object segmentation," *2016 IEEE Winter Conf. Appl. Comput. Vision, WACV 2016*, 2016.

[10] A. Kour, "Face Recognition using Template Matching," *Int. J. Comput. Appl.*, vol. 115, no. 8, hal. 10–13, 2015.