Design and implementation program identification of traffic form in self driving car robot

Lukman Yudokusumo\textsuperscript{1,2}, IG. Prasetya Dwi Wibawa\textsuperscript{1,3} and Cahyantari Ekaputri\textsuperscript{1,4}
\textsuperscript{1} Telkom University, School of Electrical Engineering, Bandung, Indonesia
E-mail: \textsuperscript{2} Lyudokusumo@gmail.com, \textsuperscript{3} prasdwibawa@telkomuniversity.ac.id
\textsuperscript{4} cahyantarie@telkomuniversity.ac.id

Abstract. The development of technology is growing more rapidly, one of them is a car robot without a driver who can help navigate the car to drive. The navigation is designed to determine the direction of the steering wheel to walk along the track properly. An image processing algorithm is designed to help navigate miniature cars. The design of this algorithm uses the Python programming language and has the OpenCV library. An image as an input while for the output is to have a miderror value to find out the position of the car in the middle of the track or not. The results in the miderror that are read in the python and the ruler on the straight track has an average of 5.485647966\%, and identification of the track shape can distinguish between the straight track, turn right, and turn left.

1. Introduction
Technological developments in Indonesia have increased very rapidly. One of them is in the field of automotive or vehicles, one of which is a vehicle of a private car. This development was followed by an increase in the number of accidents due to drivers of vehicles that might experience fatigue while driving a car.

To reduce the number of accidents caused by the drivers themselves, a system was developed that could detect road markers, to improve safety and comfort in driving, now a technology called ITS (Intelligent Transportation System) has been developed. According to the EU Directive 2010/40 / EU (July 7, 2010), ITS is defined as a system that combines information technology, electronics, and communication which is then applied to the field of transportation [1]. The application of the system is automatic navigation on four-wheeled vehicles. The vehicle can run following certain routes without the driver. The method used in the ITS system is to use the camera as image capture. The image of the camera will be processed through image processing. By processing this image, the vehicle can run on the route without leaving the road mark.

With this background and problems, a system for detecting road markers in car robots can be detected that can detect road markings and determine the error distance from the middle of the car to the road markings on a particular route. This system consists of a laptop equipped with a camera that will process images to detect road markings as a vehicle navigation guide. The distance from the results of image processing is sent to the controller as the driver of the car’s front wheel.
2. Literature Review

2.1 Robot self driving car using image processing

Robot without a driver is one type of automatic car control that can make it easier for humans to drive without the driver’s intervention. Robot without a driver car is useful when the driver experiences certain conditions that have to let go of the steering wheel such as nodding the phone, taking something on the dashboard, etc. [2].

For this reason, an image processing algorithm was designed and made on a prototype of a car that can move along the path automatically by using web camera as a vision sensor to detect road trajectories so the car can run on its trajectory.

2.2 Meaning of image

Image or image is visual information. An image is obtained by capturing the intensity of light reflected by an object. When light hits an object, some of that light will be reflected. These reflections will be received by optical sensing devices such as cameras, human eyes, scanners, and so on [3].

2.3 HSV and HLS

HSL stands for (Hue, Saturation, Lightness) and HSV stands for (Hue, Saturation, Value) are two of the most common cylindrical coordinate systems representing points in RGB color models, which rearrange RGB geometry in a perceptual attempt that is more relevant than cartesian coordinate representation. The difference between HSL and HSV is located in Lightness and Value, where Lightness will further brighten each hue (color) to the brightest color level (white), while the Value only gets to brighten up each hue (color) from the bottom position to the darkest color (black) [4].

2.4 Cropping and resizing image

One of the most fundamental in the process of photo manipulation is to remove unwanted subjects or irrelevant details in photos, change aspects of racism and to improve the composition as a whole. To do this can be done in several ways, such as cropping and resizing [5].

2.5 Contour

Contours can be interpreted as curves that combine all continuous points (along a boundary), which have the same color or intensity. Contours are useful for form analysis and object detection and recognition or can be interpreted simply as a curve with all points connected (along a boundary) and having the same color or intensity. Contours are very useful tools for field analysis and object detection and recognition. For better accuracy, you can use binary imagery [6].

2.6 Miderror

Miderror is a helpline to find out how long the camera's middle error line is towards the center of the contour of the track. The purpose of making errors in this task is to find out where the car robot is facing and to find the location of the robot car towards the middle of the track.

3. System Design

Designed a navigation guidance system for car robots without the driver using the camera as a visual sensor. The camera will be placed on the top of a car robot without a driver that will capture the track image that will be processed at the image processing stage, which will detect the black path. Data collected from image processing will be sent to the microcontroller, where the microcontroller itself uses Arduino to drive the servo as a steering robot car without the driver and controls the speed of the BDC motorbike on the rear wheels.
3.1 Block diagram
Block diagram for this system can be seen in Figure 1.

![Figure 1. Robot system block diagram](image1)

Block diagram in Figure 1 is the workflow of this system that has an input in the form of an image of a path and then processed its image using the Python programming language. Data that has been obtained from image processing is then sent to the microcontroller to control a BDC and servo motor.

![Figure 2. Image processing flowchart](image2)

The flow chart shown in the image can be seen in Figure 2 that to get the miderror value the first step is capturing the image itself, then the process of intersecting the image will pass the phase of image segmentation based on the color to be processed, and after passing the color segmentation stage based on color, will continue on stage determines the contour of the segment that has been segmented. And after getting a contour in the shape of a rectangle, it will get a midpoint, a midpoint that determines the distance of the car’s error against the middle of the track called miderror. The miderror data that has been obtained, will be sent to the microcontroller via serial communication.

3.2 Hardware Designing
In working on the system, the hardware is needed so that the system can run as desired, the hardware needed is shown in Figure 3.

![Figure 3. Hardware Design](image3)

Web camera used for image capture is Logitech C270 which has video recording up to 1280 x 720 pixels and has dimensions of Height x width x Thickness (cm) of 21 x 16 x 9. And image processing uses the Python Programming Language with OpenCV Library, image processing using a notebook PC.
And the captured data obtained from image processing in the form of miderror will be sent to Arduino Uno for servo control and BDC motors.

3.3 Image processing

The first step in digital image processing is to take or get the image to be processed, the image can be taken from internal files that are already on the computer or images were taken in real-time that are obtained using additional devices such as web cameras and scanners. This uses a web camera for image retrieval processes in real-time. The image taken in the form of a moving image in the form of video with the resolution of the web camera itself is 640 x 240 pixels with a maximum frame capture of 30 fps. This resolution allows the image matrix obtained is not too large and the next image process will run easier, 30 fps itself is the specification of the web camera used.

1) Cropping Image: After the capture of the resolution image (640,240), the next is cutting the image. Because this will only require two Region of Interest (ROI), resolution captured images (640,240), will be cut into two parts, the cut-off part of the image will be designed. These steps can be seen in Figure 4, Figure 5 and Figure 6.

2) Image Segmentation: Image segmentation is a method for separating images between objects that you want to process and background in the image. For this, we use color-based image segmentation methods to cut between black road tracks and white road markings. As the object that you want to detect is the path of the road, the segmentation process can be done in the HSV color space, so that the conversion from the RGB color space to HSV is carried out. This step can be seen in Figure 7 below.

3) Determine the Contour: The contour will connect the points on the edge of the area with the same intensity value. After getting a binary image, the contour will use the white edge of the area. The moment feature can be used to find the midpoint of the detected track or the midpoint of the contour itself. To find contours on OpenCV, there are parameters used, in this final task using a contour search mode, namely, CV_RETR_TREE searches all contours and rebuilds contour from a contour set, and in this final assignment also uses the contour estimation method,
CV_CHAIN_APPROX_SIMPLE summarizes the relationship between contour points both horizontally, vertically and diagonally. This step can be seen in Figure 8 below.

Figure 8. The contour of lane detection

4) Looking for the Contour Midpoint: At this stage, after getting and drawing the contour, it takes a midpoint to find out where the middle of the width of the track is. Because the contour is rectangular, then the midpoint will be found with a simple rectangle, where the area of the rectangle is divided in two, and the length of the rectangle is divided by two as well. This step can be seen in Figure 9 below.

Figure 9. MidPoint Of Contour

5) The negative miderror value is the center point of the camera going left compared to the contour midpoint value. Conversely, if the miderror value is positive, it indicates that the center point of the camera is more towards the right than the contour midpoint value.

Figure 10. Value Of Miderror

4. Experiments and analysis

The purpose of this test is to find out which mobile robot is based on the identification of the shape of the track. The condition said the car robot moves straight if the miderror ROI 1 and miderror ROI 2 values have a maximum difference of 1. Requirements said the car robot will move right if the miderror ROI 1 value is greater than 1 miderror ROI value. Turn left is the ROI 1 value is 1 value less than ROI 2.

For a straight track, the car robot gets the miderror ROI 1 value and miderror ROI 2 and does not have a difference of more than 1, the car robot will run along the track well on the straight track. The analysis results of track forwards can be seen in Figure 11.

Figure 11. Track Forwards
5. Conclusion
Car Robot can read straight trajectory, with 33.3% error. Car Robot can read right turning trajectory, with 6.67% error. Car Robot can read left turn trajectory, with 10% error.

References
[1] Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010 on the Framework for the Deployment of Intelligent Transport Systems in the Field of Road Transport and for Interfaces with Other Modes of Transport 2010 O.J. (L 207/1)
[2] Shanker R, Jonas A, Devitt S, Huberty K, Flannery S, Greene W, Swinbume B, Locraft G, Wood A, Weiss K, Moore J, Schenker A, Jain P, Ying Y, Kakiuchi S, Hoshino R and Humphrey A 2013 Autonomous Cars: Self-Driving the New Auto Industry Paradigm (New York: Morgan Stanley Research)
[3] Urmson C, Anhalt J, Bagnell D, Baker C, Bittner R and Clark M N 2008 Autonomous driving in urban environments: Boss and the urban challenge J. Field Robot 25 p 425-466
[4] Gonzalez C R and Woods E R 2008 Digital Image Processing 3rd Ed (USA: Prentice Hall, New Jersey)
[5] Solem and Jan Erik 2012 Programming Computer Vision with Python (Sebastopol: O’Reilly Media)
[6] Jaehyoung Y, Youngjoon H and Hernsoo H 2011 An Efficient Extraction of On-Road Object and Lane Information Using Representation Method 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems p 327-332