

Design of Lane Detecting and Following Autonomous Robot

Tushar Wankhade¹, Pranav Shriwas²
^{1,2}(International Institute of Information Technology, Pune)

Abstract : This paper presents a robot which is designed to autonomously navigate through a track by detecting lanes and centering itself between them by using a camera. We are proposing basic and easy algorithm for tracking and detecting lanes without considering any cameras' parameters. The car detects lanes through image input from a camera mounted at its front. All computations based on data from camera are handled by Laptop which is carried by car. The car controlling part is carried by ATmega8 microcontroller. Object tracking algorithms based on color, shape is used for tracking lanes. Based on the information retrieved, data/signals will be transferred to the controller, which will control the movement of motor. This project uses OpenCV library for digital image processing under LINUX environment.

Keywords - Lane detection, Hough transform, USART, PWM.

I. INTRODUCTION

Car accidents kill about thousands of people every year, most of these accidents are caused by driver faults [1]. Automating driving may help reduce this huge number of human injuries or deaths. One useful technology is lane detection. Different methods are used for lane detection and tracking like B-snake [3] and IPM (Inverse Perspective Mapping) [4]. Both these methods have their advantages and disadvantages. This paper presents design of a small prototype AGV (Autonomous Guided Vehicles) which will detect and follow the lanes by using simple machine vision algorithms. The AGV (Autonomous Guided Vehicles) are capable of performing required tasks in defined environments without continuous human guidance. Different robots are autonomous in different ways. The basic idea is to program the robot to respond a certain way. The autonomous robot which follows some defined path known as AGV (Autonomous Guided Vehicles). Generally an AGV requires a definite path like painted tracks, wired tracks and controller. The paper is organized as follows: section II gives a Basic idea of lane detection. Section III gives a detailed description of the approach followed by results in section IV Finally, a conclusion is given in section V

II. BASIC OF LANE DETECTION

There are different method for lane detection and tracking, but there are conman techniques used in all the methods to achieve the lane detection. By making use of these Digital Image Processing techniques such as smoothening, thresholding and line detection we designed algorithm for lane detection. In the hardware implementation ATmega8 microcontroller is used for controlling motor. The L298 and MAX 232 are used for driving motors and communicating with laptop respectively. Webcam is capturing video and laptop is used to carry out image processing part.

The basic block diagram is shown in figure 1

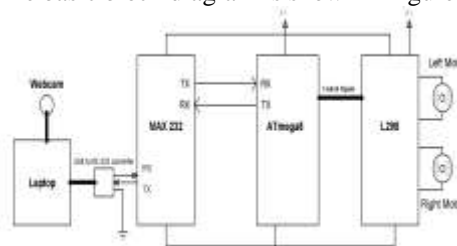


Fig. 1 Block Diagram.

III. APPROACH

A. Image smoothening.

The Average (mean) filter is used to smooth image data. This filter performs spatial filtering on each individual pixel in an image using the pixel values in a defined square window around each pixel, as shown in figure 2.

| | | |
|----|----|----|
| p1 | p2 | p3 |
| p4 | p5 | p6 |
| p7 | p8 | p9 |

Fig. 2 3*3 pixel grid.

The average filter computes the sum of all pixels in the filter window and then divides the sum by the number of pixels in the filter window.

$$\text{Filtered pixel (p1)} = (p1 + p2 + p3 + p4 \dots + p9) / 9 \tag{1}$$

An average filter is used to smooth out the images from camera. The filter used is of 3*3 size. The use of average filter minimises the illumination changes in images. In RGB image format each channel is considered separately for smoothing.

B. Colour detection

The Red colour is used for lanes. By taking sample images of lanes through camera the ranges of the RGB colour values for lanes are recorded. Now continues images from camera are taken and the values of the RGB colour ranges are matched with the each image, on the basis of that thresholding is done and stored on other image of same size.

C. Edge detection

Edge detection highlights image contrast. Edge detection gives boundaries of features in the images. Edges are detected where contrast i.e. difference in intensity occurs. Change in intensity can be found out by differencing adjacent pixel points. Differencing horizontally adjacent points will detect vertical changes in intensity and is called a horizontal edge-detector. A horizontal edge detector will not show horizontal changes in intensity since the difference is zero. [6]

A simple method for edge detection is used, instead of using any mask the contrast levels of threshold image are compared. As this fulfils the requirements and uses less computation. Also it is done only for vertical edges so any horizontal edge will not appear in image.

D. Centre finding

The edge detection has generated the edges of lanes, the centre of lanes can be found out by averaging the edge pixels in each row and dividing it by no of edge pixels in that row. This will provide centre of lanes with some unwanted pixels which is due to changes in illumination and edges of objects which are not lanes.

$$C = \sum_{i=0}^{\text{width}} \left(\frac{P_i}{P_n} \right) \tag{2}$$

Where:

Pi : Coordinates of pixel in a row

Pn: Total no. of pixel in a row

E. Hough transform

The Hough transform is a used to detect shapes in images. It is used to extract lines, circles and ellipses. The Hough Transform implementation defines a mapping from the image points into an accumulator space also called as parameter space.

Equation 3 is used to describe a line which is represented as shown in Figure 3.

$$r = x \cos \theta + y \sin \theta \tag{3}$$

Where –

r : perpendicular from line to origin.

θ : Angle of perpendicular from x-axis

x, y : coordinates of line

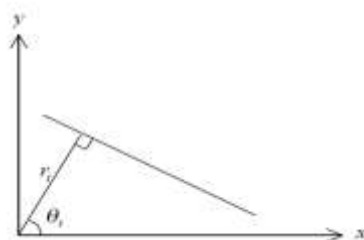


Fig. 3 Representation of line in (r, θ) space [10]

Value of r is calculated for different values of θ for a particular value of x and y by doing so for all the line pixels we will get plot of r against θ. For each value of coordinates we will get a curve in this plot these curves will meet each other at a point in plot which shows that line is present in an image.

Applying Hough transform for line detection will solve the above problem of unwanted pixels. Providing proper parameters for line detection will draw a centre line for the lanes.

Line parameters are:

1. Minimum no of pixels to count as a line.
2. Separation between two collinear lines.

Now coordinates of the centre line is stored in an array, these coordinates are used to calculate the duty cycle values for PWM and passed to controller serially for motor control.

F. Serial communication

In this project serial communication is required for transferring duty cycle for controlling speed of the motors. IC MAX 232 is interfaced with microcontroller for this purpose as shown in figure 4.

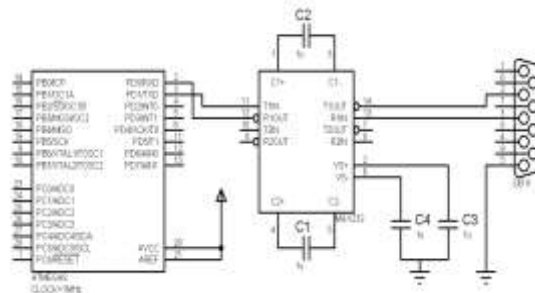


Fig. 4 Interfacing of MAX 232 with ATmega8

We have used 1Mz internal crystal oscillator so maximum baud rate is 4800 bits/sec. In ATmega8 to get desire baud rate one need to provide value to UBRR (USART Baud Rate Register) of controller which can be calculated as.

$$UBRR = F_{osc} / (16 * BAUD) - 1 \quad [11] \tag{4}$$

Where -

F_{osc} : Input Crystal frequency

BAUD: Required baud rate

In LINUX all devices, serial ports are accessed via device files. To access a serial port open the corresponding device file. Each serial port on a LINUX system has one or more device files (files in the /dev directory) associated with it. [7]

Following are the functions used for configuring serial ports [12].

- int open(const char *pathname, int flags)
- ssize_t write(int fd, const void *buf, size_t nbyte)
- int cfsetispeed(struct termios *termios_p, speed_t speed) and int cfsetospeed(struct termios *termios_p, speed_t speed)
- int tcgetattr(int fd, struct termios *termios_p)
- int tcsetattr(int fd, int optional_actions, struct termios *termios_p)

G. PWM Generation and motor drive.

Pulse width modulation is used for controlling speed of motors. In PWM instead of generating a constant DC voltage output a square wave is generated, which has high = 5V and Low = 0v. If the frequency of such wave is high about 500 Hz with the duty cycle of 50% the output will be half of 5v i.e. 2.5 volts. In PWM on and off may be varied from 0 to 100 percent. In this manner, a variable amount of power can be generated.

In microcontrollers counters are used to generate pwm signals. In the case of 8 bit counter the count value ranges from 0 to 255 which will generate duty cycle from 0 to 100% .The output voltage can be calculated by using Equation (5).

$$V_{out} = (\text{count} / \text{Max. count}) * V_{in}. \tag{5}$$

The internal timer is used for generating PWM signals.

It is configured as following.

1. 8 bit timer/counter
2. Fast PWM mode
3. Non-inverting mode
4. F_{osc}/256 prescaler

We have used L298 as motor driver IC to provide necessary power for motors. The circuit used for motor drive is as show in figure 5.

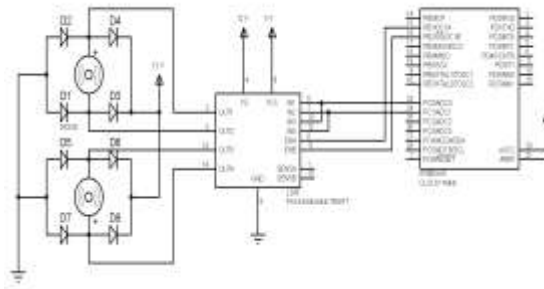


Fig. 5 Interfacing of L298 with ATmega8

IV. RESULTS

Figure 6 shows the result of our approach used for lane detection.

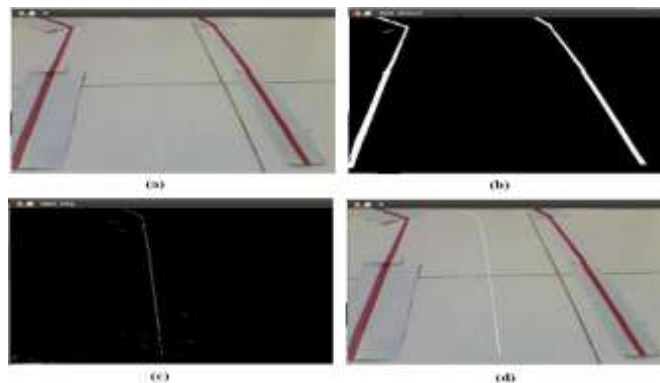


Fig. 6 Results of Lane detection. (a) Original image after smoothing, (b) Red colour detection (c)After edge detection centre of lanes, (d) Final centre line of lanes after Hough transform.

The figure 6 (a) shows the effect of smoothing the original image from camera, it helps to reduce the illumination effect in image, due to changing illumination the colour can't be detected properly. After choosing window of RGB colour values for defined lane colour i.e. Red, from one or two sample images from camera, the image is thresholded by using the RGB values show in figure 6 (b). The edges are found out from the thresholded image for calculating the centre of lanes as shown in figure 6 (c) but the edges of the lanes are not shown in images instead of the only the centre of lanes is shown because in the next step of applying Hough transform we will get three fluctuating lines. The centre line is detected and drawn on original image by applying Hough transform as shown in figure 6 (d).

V. CONCLUSION

This project uses basics of digital image processing methods for detection and tracking of the lanes. The illumination and camera elevation angle and position affects majorly in the algorithm used for detecting lanes. So maintaining proper lightning condition and adjusting camera a small prototype of autonomous vehicle is built and good results for detecting and following lanes are achieved.

REFERENCES

- [1] Transportation: Motor Vehicle Accidents and Fatalities [Online]. Available: http://www.census.gov/compendia/statab/cats/transportation/motor_vehicle_accidents_and_fatalities.html
- [2] P. Slavíková, M. Mudrová A. Michalcová and A. Procházka "Hough Transform Use in Image Processing of Microscopic Alloy Images", Institute of Chemical Technology Prague.
- [3] Yue Wanga, Eam Khwang Teoha, Dinggang Shenb "Lane detection and tracking using B-Snake", Department of Radiology, University of Pennsylvania, Philadelphia, PA 19104, USA.
- [4] Mohamed Aly "Real time Detection of Lane Markers in Urban Streets", Computational Vision, Lab Electrical Engineering California Institute of Technology, Pasadena, CA 91125
- [5] [Book] Gonzales, woods "Digital Image Processing"
- [6] [Book] Mark S. Nixon, Alberto S. Aguado "Feature Extraction and Image Processing"
- [7] Serial programming guide for POSIX operating systems [Online]. Available: <http://www.easysw.com/~mike/serial/serial.html>
- [8] RS232 Specifications and standard [Online]. Available: http://www.lammertbie.nl/comm/info/RS-232_specs.html
- [9] The RS-232 Standard [Online]. Available: <http://www.omega.com/techref/pdf/rs-232.pdf>
- [10] Ghassan Hamarneh, Karin Althoff, Rafeef Abu-Gharbieh "Automatic Line Detection", Image Analysis Group Department of Signals and Systems Chalmers University of Technology
- [11] Datasheets of ATmega8, L298, MAX232.
- [12] Linux manual page