

Autonomous Real-Time Navigation Based on Dynamic Line and Object Detection



Pavan Rayar, Adarsh Prabhudesai, Samruddhi Pai, and Shaival Parikh

Abstract An Android application has been developed that captures images, detects a line and computes the lateral deviation as well as angular deviation from the path. It also takes into consideration the intensity variation of the surroundings which has been explained in the following paper. We have used the Sobel edge detection algorithm and Hough transform for the angular inference. Colour thresholding and matrix manipulation are done for calculating the lateral deviation. Finally, the counter-active actions and inferences are sent via BLE (low energy Bluetooth). An application was developed for directing the autonomous quadruped robot. The rare-view camera of the mobile was used for taking the input. This Android application can have multiple implementations apart from robotics with slight manipulations, such as the automobile industry, military, industrial automation and blind assistance, and these kinds of researches will boost concept of smart manufacturing.

Keywords Image processing · Filters · Matrix manipulation · Line and object detection · Automation · Robots · Android application

1 Introduction

Computer vision is one of the budding technologies which has an implementation in fields like the automation industry, health sector, financial services, etc. The perception of the real world with an Android application is a very useful tool in today's age. Edge detection is one of the significant fields of computer vision. Out of the multiple edge detection algorithm, Sobel edge detector is used in this application. Hough

P. Rayar (✉)

Department of Production Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

e-mail: pavan.rayar@djsce.ac.in

A. Prabhudesai · S. Pai · S. Parikh

Department of Electronics Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

© Springer Nature Singapore Pte Ltd. 2020

H. Vasudevan et al. (eds.), *Proceedings of International Conference on Intelligent*

Manufacturing and Automation, Lecture Notes in Mechanical Engineering,

https://doi.org/10.1007/978-981-15-4485-9_36

transform is a feature extraction method utilized to define geometrical shapes. Different aspects of visualization are tested in different colour spaces. Colour thresholding is one of the key methods of image segmentation. Only the required information is encoded into pixels to avoid background noise.

2 Literature Survey

Traditionally, line following for autonomous robots is done with infrared sensors. According to a study done by Prakash et al. [1] presented in ‘Computer Vision Assisted Line Following Robot’, they have used camera for object detection and IR sensor array for line detection and automating the supply chain industries. Object detection is done with the help of colour segmentation [2] for detecting defective shipments. Line following is used for navigation of an automated guided vehicle through various zones [3]. The major drawback of the infrared sensors is the functional inaccuracy when exposed to intensity variations of light. Owing to that in this study, we amalgamate the object detection with line following using an Android device which not only is more reliable and accurate but can handle the intensity variations with better precision [4, 5].

3 Methodology Used

Image acquisition is done with the help of a rear-view camera of the Android device. The video capture rate can be selected according to the dynamicity of the application. For our quadruped robot perception, 10 fps was sufficient. The capture image quality can be decided according to the requirement of the task as well as the computational power of the device. In our application, a 640×480 pixel image is captured. The image is parallel processed in two ways for finding out the angular deviation and the lateral deviation Fig. 1.

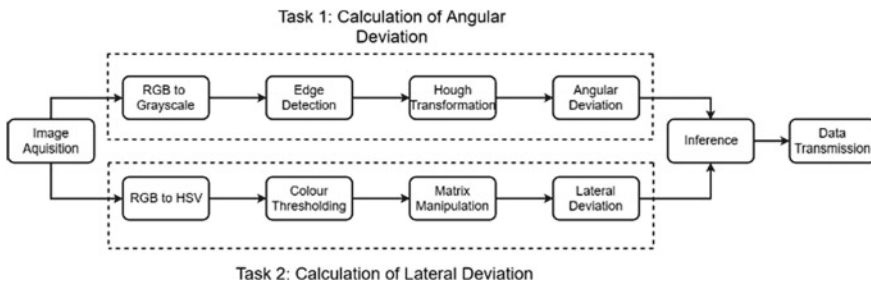


Fig. 1 Flow chart of the complete algorithm

For angular deviation, first, the image is converted into a greyscale image followed by the edge detection. Hough transformation refines the shape further giving out the angular deviation from the axis of the device.

In the second parallel track, RGB is converted into the HSV colour space as it is susceptible to less noise. In this specific application, colour thresholding is used to detect white colour, but any colour can be detected by changing the colour thresholds. Finally, the binary image is interpreted into the lateral deviation via matrix manipulation. Thus, taking these to values into consideration, the position of the robot can be accurately interpreted. The IR sensor of the mobile is utilized to measure the intensity of the surroundings so as to make sure the image is captured in suitable lighting conditions.

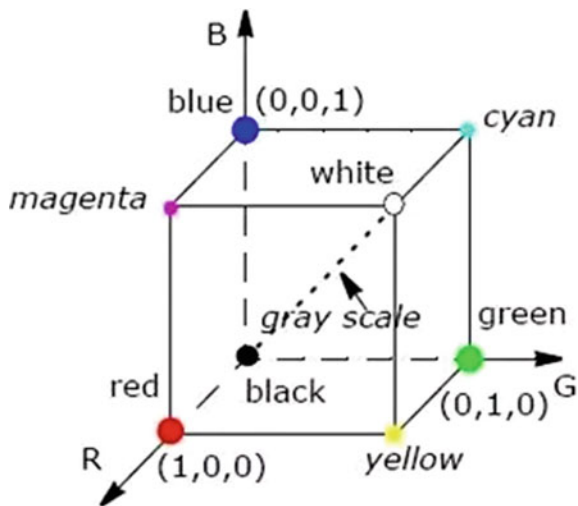
Task 1: Angular Deviation

1. RGB To Greyscale Conversion RGB to greyscale conversion is done to encompass the data of three matrices into a single matrix. RGB to greyscale conversion is obtained by computing the weighted sum of the red, green and blue matrices of the RGB image. The weights are adjusted to compensate for the variable wavelengths of the three colours. The equation used to calculate the equivalent greyscale pixels is as follows:

$$X_{ij} = ((0.3 * R_{ij}) + (0.59 * G_{ij}) + (0.11 * B_{ij})) \tag{1}$$

The RGB to greyscale conversion can also be depicted on a 3D plane using a unit cube with one vertex on the origin and three edges coincident with the coordinate axes. The origin corresponds to black, and point (1, 1, 1) corresponds to white. The diagonal joining these two points defines the greyscale threshold (Fig. 2).

Fig. 2 RGB to greyscale correlation in 3D space



2. Edge Detection In our application, we had to detect lines and objects like a step, ropes, slopes, etc. Sobel edge detection algorithm was used in the first stage of processing, in order to detect these objects.

A discrete approximation of the gradient can be used to detect the significant changes in the greyscale image. Sobel edge detector uses two convolution masks (generally 3×3), one is used for computing gradient along the x -axis and another one for computing gradient along the y -axis [4]. Wherever a sudden change in intensity is detected, a local-maxima is obtained there as an indication of an edge. These gradient values of each pixel are compared with the threshold value, and the one exceeds this value is detected as an edge. After multiple iterations, the threshold value was chosen to be 10 as it gave relatively better results.

Its convolution kernel used is designed in such a way that it smoothens the input image considerably and makes the operator less susceptible to noise. One of the advantages of the Sobel detector is that it does not consider only the single-pixel but takes differential of two rows or columns which enhances the edges making them seem brighter and thicker.

In the Sobel edge detection technique, the centre of the convolution mask coincides with the pixel of interest, and then, it calculates the G_x and G_y (gradient in each direction) for that particular pixel. For digital images, the derivatives are approximated by considering the difference of adjacent pixels. The gradient approximation used here is given by,

$$G_x \sim f[i, j + 1] - f[i, j] \quad (2)$$

$$G_y \sim f[i, j] - f[i + 1, j] \quad (3)$$

where $i =$ negative y -direction, $j =$ positive x -direction.

In this way, the gradient of each pixel is calculated by varying the values of i and j in a loop. The pixels of the first and last rows and columns cannot be calculated if a 3×3 convolution mask is used as while attempting to align the centre of the mask with the concerned pixel, and the mask goes outside the image boundary [2].

3. Hough Transform Edge detection does not give accurately the most prominent line and in it, a lot of pixels (points) are missing, so it becomes difficult to interpret the given shape of the figure. Hence, we further process the image for feature extraction using Hough transform [1].

We convert the given slope-intercept form ($y = mx + c$) into parametric equation ($x = r \cos \phi + r \sin \phi$).

All the given lines through a point are considered, and this yields a sine-like curve in the Hough space. Similarly, all the edge points are considered, and Hough peaks are taken into account. The global maxima of the set are considered to be the most prominent line as the intensity variation would be maximum, and the edge would be the one differentiating the background with the white line on it. The angle and rho

value are given as the output of the Hough transform block. The angle of the line is with respect to the axis of the Android device; hence, we get the perceptual angular deviation with respect to the axis.

Task 2: Lateral Deviation

1. RGB To HSV Conversion While RGB model depicts colour as a combination of primary colours, HSV model is used to define an output similar to how the human eye perceives colour. In applications where colour description would be more useful, HSV is preferred over RGB model.

‘Hue’ stands for the colour, ‘saturation’ defines the ratio to which the required colour is mixed with white, and ‘value’ defines the ratio to which the required colour is mixed with black. (grey level).

RGB to HSV conversion formula:

The values of R, G, B are divided by 255 to change their range from 0–255 to 0–1.

$$\begin{aligned}
 R_{\text{norm}} &= R/255 \\
 G_{\text{norm}} &= G/255 \\
 B_{\text{norm}} &= B/255 \\
 C_{\text{max}} &= \max (R_{\text{norm}}, G_{\text{norm}}, B_{\text{norm}}) \\
 C_{\text{min}} &= \min (R_{\text{norm}}, G_{\text{norm}}, B_{\text{norm}}) \\
 \Delta &= C_{\text{max}} - C_{\text{min}}.
 \end{aligned}$$

Hue calculation:

$$\text{Hue } H = \begin{cases} 0^\circ, & \Delta = 0 \\ 60^\circ \times \left(\frac{(G_{\text{norm}} - B_{\text{norm}}) \bmod 6}{\Delta} \right), & C_{\text{max}} = R_{\text{norm}} \\ 60^\circ \times \left(\frac{B_{\text{norm}} - R_{\text{norm}}}{\Delta} + 2 \right), & C_{\text{max}} = G_{\text{norm}} \\ 60^\circ \times \left(\frac{R_{\text{norm}} - G_{\text{norm}}}{\Delta} + 4 \right), & C_{\text{max}} = B_{\text{norm}} \end{cases}$$

Saturation calculation:

$$\text{Saturation } S = \begin{cases} 0, & C_{\text{max}} = 0 \\ \frac{\Delta}{C_{\text{max}}}, & C_{\text{max}} \neq 0 \end{cases}$$

Value calculation:

$$V = C_{\text{max}}$$

HSV or Hue Saturation Value are used to extract the luminance information from the given image and provide better inference of the surrounding light intensity when compared to RGB model.

2. Colour Thresholding Colour thresholding is done to extract the object of desired colour by masking all the other information of the image. In our application, we had to detect line and various objects. The line was white in colour, and all the other objects were of yellow colour. Thresholding operation was performed on the HSV image where different limits of Hue, Saturation and Value were applied to isolate white and yellow colour from the image. The pixels with all three values within the specified limits were assigned logic 1 and others as logic 0.

- If $\text{Hue}_{\min} < H < \text{Hue}_{\max}$, then $H = 1$, else $H = 0$
- If $\text{Sat}_{\min} < S < \text{Sat}_{\max}$, then $S = 1$, else $S = 0$
- If $\text{Val}_{\min} < V < \text{Val}_{\max}$, then $V = 1$, else $V = 0$

3. Matrix Manipulation For calculating the lateral deviation, we split the processed image matrix into five parts, partitioning column-wise. The matrix would be of 0s and 1s, and the subpart containing a greater number of pixels having value 1 would denote the white area.

$$A_k = \sum_{j=1+(k*\frac{n}{6})}^{(k+1)*n/6} 1 \sum_{i=1}^m W_{ij}$$

$$\begin{bmatrix} W(1, 1) & \cdots & W(1, m) \\ \vdots & \ddots & \vdots \\ W(m, 1) & \cdots & W(m, n) \end{bmatrix}$$

4. Lateral Deviation Calculation The sum of each submatrix is stored into an array named ‘Line_detect’ and then compared to identify in which part the line lies. The index of the maximum value of the array denotes the position of the line.

$$\text{Line_detect} = [A1 \ A2 \ A3 \ A4 \ A5]$$

$$[\text{max_value}, \text{postion}] = \max (\text{Line_detect})$$

The max() function outputs the maximum value and the index of the maximum value. The index is the maximum value and is the position of the white line. This position is transmitted via BLE (low energy Bluetooth) to other devices.

4 Applications

This application was primarily built to help a quadruped robot in perceiving its surrounding and according to take actions. The information about the angular as well as lateral deviation from the white line, that the robot had to follow, was transmitted to the on-board controller of the robot. According to the received information, the controller gave information to the actuators that in turn reacted to reduce the deviation. There were different functions designed for climbing the step of crossing the rope.

Whenever a step or rope was detected, the respective functions were called and the actuators responded accordingly. The application as a lag free user interface and multiple functional on-screen buttons. A slider is used to set the threshold value, and the 'Line_Detection' button is used for screen transition from binary output to RGB (as shown in Fig. 3a, b).

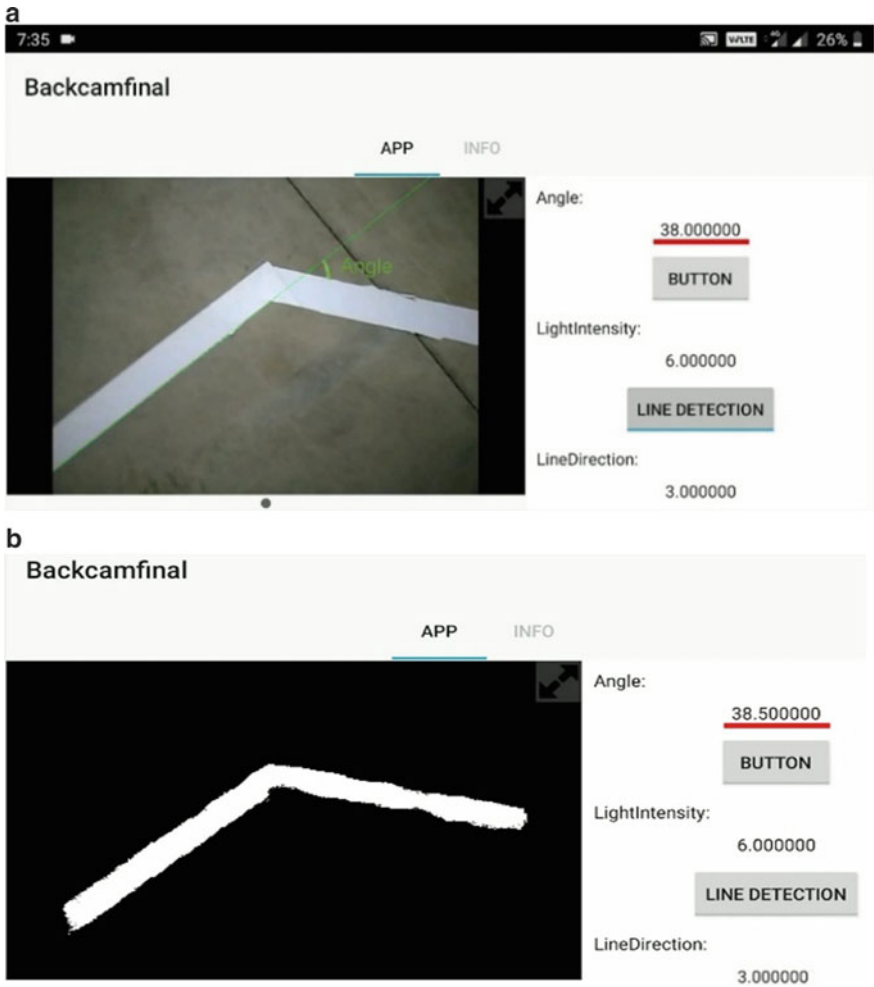


Fig. 3 a Input image displayed on the Android GUI. b Processed image

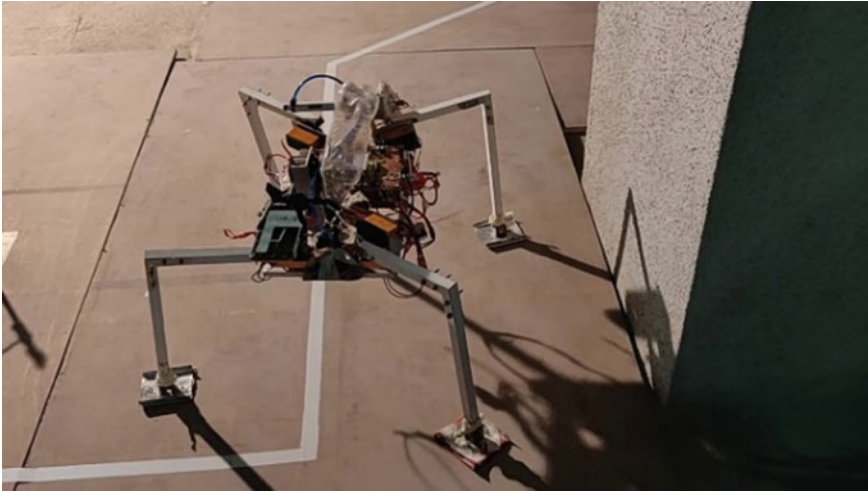


Fig. 4 Algorithm deployed on a quadruped robot

5 Conclusion and Future Scope

The developed Android application was used to guide the autonomous quadruped robot. It sent accurate data and counter-active responses to the robot. The application was tested in various light conditions and showed high accuracy when compared to typical line following approach using an infrared sensor array. Another added advantage of this application was the user-friendly UI which facilitated the changing of the threshold values in real-time during calibration (Fig. 4).

The generalized algorithm developed here can be modified and used for various other applications. This algorithm can be deployed on various platforms and can provide assistance to the user by detecting obstacles and providing counter-active actions by sending audio signals. The object detection algorithm can also be used in autonomous cars to detect lanes and any obstacles in their paths. The dynamic colour thresholding and line detecting part of the algorithm can be very useful in industrial automation applications like parcel segregation based on shape, orientation, colour, etc. We can also incorporate other sensors of the mobile to add more features in the application and increase its functionality and robustness.

References

1. Surya Prakash M, Ajay Vignesh K, Shyamsunthar J (2012) Computer vision assisted line following robot. In: International conference on modeling, optimization and computing (ICMOC-2012)

2. Malik R, Khurshid J, Ahmad SN (2007) Road sign detection and recognition using colour segmentation, shape analysis and template matching. In: 2007 international conference on machine learning and cybernetics, vol 6. IEEE, pp 3556–3560
3. Pakdaman M, Sanaatiyan MM (2009) Design and implementation of line follower robot. In: 2009 second international conference on computer and electrical engineering, vol 2. IEEE, pp 585–590
4. Gupta S, Mazumdar SG (2013) Sobel edge detection algorithm 2(2). ISSN: 2278-733X
5. Gonzalez RC, Woods RE Digital image processing, 2nd edn