Vision Based Trajectory Following Robot and Swarm

Ishraq Ul Haque, Abul Al Arabi, Shadat Hossain, Tanjina Proma, Nafi Uzzaman, and M Ashraful Amin
Computer Vision and Cybernetics Group, Department of Computer Science and Engineering,
Independent University, Bangladesh (IUB), Plot 16, Block B, Bashundhara. Dhaka 1229, Bangladesh
e-mail: {1410226, arabi, 1221226, 1320810, 1310552, aminmdashraful } @iub.edu.bd

***Abstract*—Robots based on vision guidance are very important to the future of vehicle navigation. They have the ability to navigate through a predetermined track and avoid obstacles. They operate remotely with a central computer tracking and guiding them. This allows for communication between robots. This can be used to implement traffic based navigation. Where routes can be recalculated depending on traffic flow. Use of server based system paves the way to collaborate among different robots.**

Keywords-trajectory follower; swarm robotics; image processing; cloud computing

I. INTRODUCTION

All around the world, streets are marked with lines and signs. These lines are used for navigation. They are also used as rules to maintain safety between all vehicles operating in the road. Different developers opt for different method for navigation. One such method [1] is placing special markers throughout the environment and have special sensors on-board the robot to detect them. This type of Autonomous Guided Vehicle (AGV) uses lasers to detect markers. This is done to find the current position of the robot and to plan its route. But for our project we will build a simple robot. Our goal is to build a robot that can follow a single line. This will be a simple example of an autonomous robot.

A technique [3] used by current manufacturers is to detect electrical charge from a wire. This is done by placing electrical wire under the floor and have electricity flow through it. Sensors on-board the robot detects the electricity and follows it. The electrical wire essentially forms a path for the robot. But this is an expensive technique and the cost of scaling is very high. To avoid this, we can rely on image processing [7, 8, 9]. All current roads already have lines, markers and signs [6]. So there is no need to have a special environment. This decreases the cost greatly. The procedure becomes more sophisticated [10, 11] with the usage of hardware like microcontrollers or microprocessors which are cost effective to manage the whole criterion of vision as easily as possible. As the images of the line being processed from the camera mounted on the tracking device, it gives the real time vision for the task to be dealt with [12, 13]. There are other methods used for making autonomous robotics using android systems [15]. Some research reports generating maps using sonar technology for autonomous navigation [16].

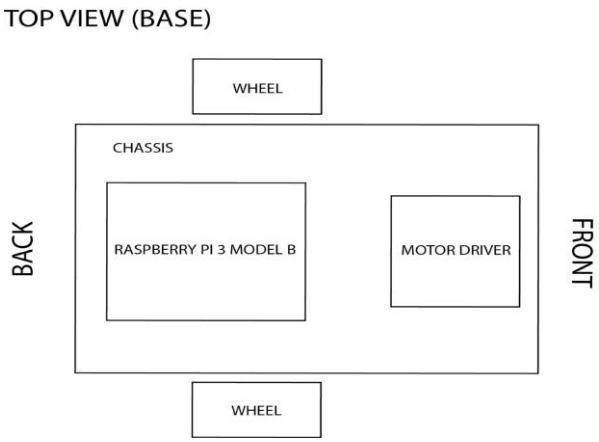

Figure 1. Top view of chassis layout (Block Diagram)

Another method that is employed is evolutionary robotics [4]. This is a field of research based on the behavior of robots. In this case, computation is used to design and model the robots. Machine learning is used to feed initial scenarios. Then learned robot takes inputs and evolves the model to make it more complex and intelligent. The robot will beplaced at a trajectory. With a contrast against the background, it is possible for the robot to differentiate the trajectory from the background with relative ease.

II. APPROACH AND SYSTEM DEVELOPMENT

A. Hardware

A chassis as the base hold the wheels, attached to the geared motors with a 550rpm. To balance the robot, we placed the support (omnidirectional) wheel at the back, underneath the Chassis. The Motor Driver board and the Raspberry Pi was fixed by screw-nut mechanism in the chassis. A second platform was created over the base chassis to house the Battery pack, Battery Bank and camera as shown in Fig. 1.

For logic and control we have used a Raspberry Pi 3. A raspberry pie is a small single board computer. It uses a quad core ARM A53 processor and has 1GB RAM. It has built in Wi-Fi module which are being used here to communicate without central computer. The Raspberry Pi3 was powered by 5v Battery and the Motors are powered by 10.5v, the camera is connected to Raspberry Pi3 using USB.

978-1-5090-3774-2/17/$31.00 ©2017 IEEE

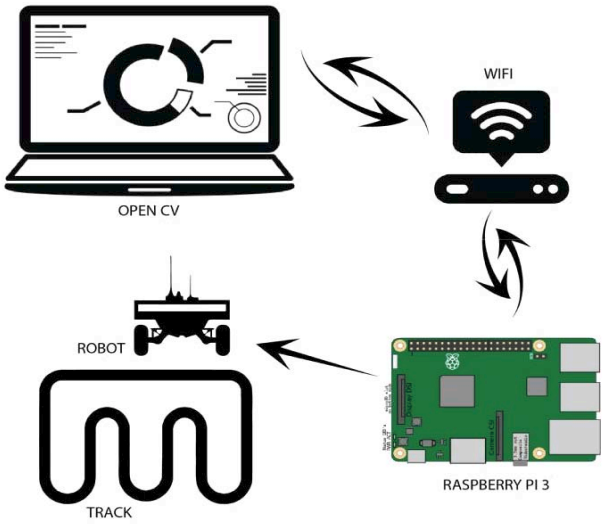

Figure 2. Overview of the proposed

In Fig. 2, the proposed system overview is given, and Fig. 3 we are providing the data flow diagram of the proposed system.

B. Image Processing and Logic Development

Image processing based robotic navigation system is also available in the literature [2]. There are methods proposed for more complex road network [14]. Our camera on-board takes 30 FPS at RGB color space. We process each frame individually and from that extract a result that we use to manipulate the motors. Our system uses time based framing method to extract individual instances for data collaboration.

The acquired frame is then fed to a system that at first converts the RGB color space into HSV. After morphological transformation, the image is converted into single channel from its multi-channel by background suppression method.

Using a threshold technique to create a binary image. Here any pixel value below a certain level is converted into white color. Anything above that value will be converted into black color. So it converts the black lines that the camera detects into white line and the white background into black. Flipping the colors makes the line stand out from the background. In the original image the line was a black line against a white background. Now the line is a white line against a black background.

C. Mathematical Model

Trajectory of the robot is then converted on the basis of that obtained and processed frame. The image captured has a size of 160x120 pixels. The mono color space frame indicates 1 for detected line and 0 for else. We create a 32x24 pixel sized mask Fig. 4. We place this mask over the original image and traverse it.

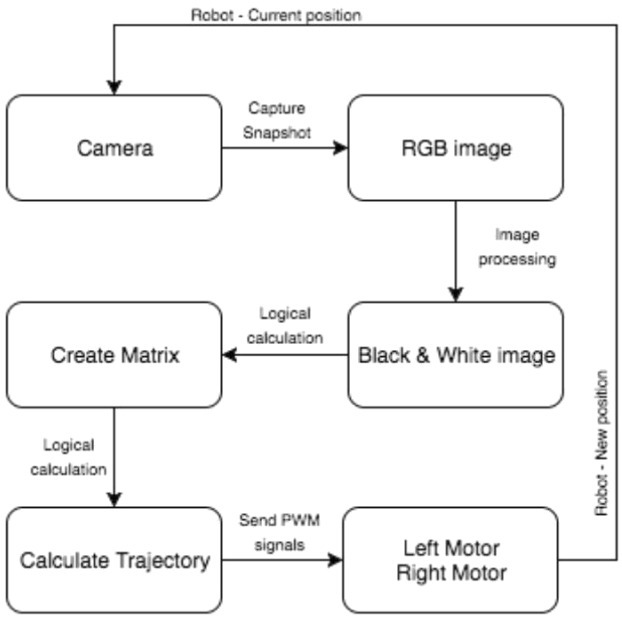

Figure 3. Feedback loop of the proposed system

The mask will be traversing the image $M * N$ times. Value of M and N can be found as following:

$$M = \frac{FrameWidth}{MaskWidth} \tag{1}$$

$$N = \frac{FrameHeight}{MaskHeight} \tag{2}$$

For our particular case the values obtained from of M be 5 and also the value of N be 5. So it requires $5 * 5$ i.e. 25 times for the mask to cover the whole image.

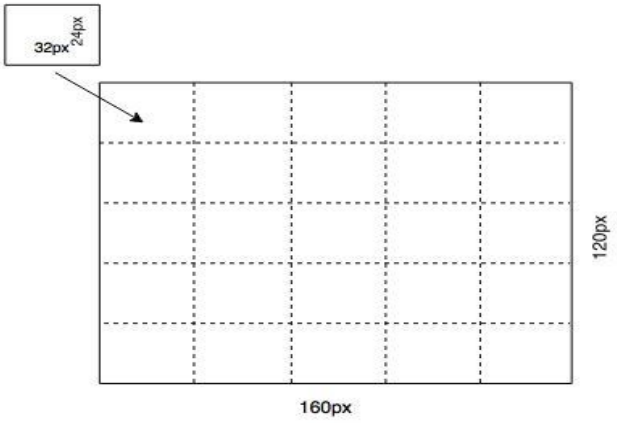

Figure 4. Image divided into 5x5 blocks

Figure 5. Image matrix representing straight line (red), curved to left (blue) and curved to right (green).

The system then develops a $M * N$ ($5*5$ for our case) matrix figure. 5. Each time after traversing the image with the mask, if the number of white pixels in the block is greater 20% of the total number of pixels in the block, then we place a 1 in the matrix. Otherwise, we place a 0 if the number of white pixels is not sufficient. In the end 5×5 matrix represents the image that have been captured.

Calculation of trajectory is found from equation 3. The value of x lies in between -1 to 1. It represents the position of the current position of the line with respect to the center line.

$$x=\left(5-\sum_{i=1}^{5}M_{i3}\right)\left[0.1*\left(\left(\sum_{i=1}^{5}\sum_{j=1}^{2}M_{ij}\right)-\left(\sum_{i=1}^{5}\sum_{j=4}^{5}M_{ij}\right)\right)\right] \quad (3)$$

With a suitable PID (Proportional-Integral-Derivative) system the motor speed is hence calculated. A PID controller continuously calculates error value (for our case that is x) as the difference between the desired line and the measured variable and correcting the speed of the motors based on proportional, integral, and derivative terms. The left and right motor speed (PWM) can be found from equation 4 and 5.

$$leftSpeed = baseSpeed + K_p * x + K_d * \frac{dx}{dt} + K_i \int_0^i x dt \quad (4)$$

$$rightSpeed = baseSpeed - K_p * x - K_d * \frac{dx}{dt} - K_i \int_0^i x dt \quad (5)$$

Here, K_p, K_i and K_d denotes the coefficients for the proportional, integral and derivative gain. K_p explains the proportional constant values of error. If the error is added then the speed of the motor will increase causing to take a turn. K_i explains the integrated values of error that being gathered over time and the controller responds by giving stronger actions. K_d predicts trends of error based on its rate of change. And thus these coefficients help to determine the speed of the motors.

D. Data Centralization & Swarm Robotics

Our system can perform all processing on a central computer. The computer and the robot are connected remotely through the Wi-Fi protocol. Images are captured on-board the robot and sent to the computer. The computer them processes the image and finds out the required trajectory required to follow the line in front. Once calculated, the computer sends PWM values to each individual motors on the robots.

The process is repeated as per fig. 2 in a smooth flow to create a continuous motion. A brief data handling flow chart is viewed in fig. 3

This wireless setup brings up the possibility to employ a swarm of robots. Multiple robots can be positioned on the [5] track and their calculations performed on the central computer. Since the central computer is tracking all the robots, they can also be programmed to avoid each other when they are close by. To increase efficiency, they can also be told to avoid certain paths to avoid having to wait for the other robot to clear the path. This can be a great application for avoiding traffic by detecting blockage and calculating another route.

III. Implementations and Field Test

Field testing has been done on a PVC arena, printed with synthetic colors. The background was white with intentional noises and the foreground was black. For wireless communication with the server the built-in raspberry pi Wi-Fi module was used. Static IP address was used to ensure proper connection with the server.

Frame has been taken at a lower dimension to avoid data burden on the Wi-Fi protocol. For processing the data MATLAB & OpenCV library has been used. L298N motor driver IC was hooked up with the GPIO pins of the raspberry pi for motor controlling.

IV. Result and Analysis

With the increase in frame size the accuracy of line position is improved. But simultaneously the data handling and image processing needs to be faster, leading towards using a high end device.

The lines are well detected through the algorithm Fig. 6, 7, 8 and the matrix is obtained as per Fig. 5. Although sometimes line contains noise Fig. 7, but it's automatically eliminated in the matrix due the sum threshold value of 20%.

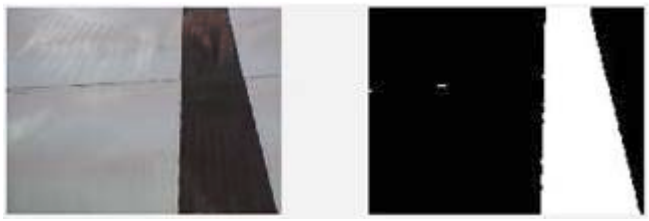

Figure 6. Forward Trajectory Detection

Figure 7. Left Trajectory Detection

Figure 8. Right Trajectory Detection

With the increase in robot speed, the minimal curvature, that the robot can follow, needs increment. Otherwise overshooting from trajectory occurs. A brief table I states the scenario. Using Gaussian, median or other filter can reduce the noise but also involves a lag in the system. Hence reducing the responsiveness of the robot.

TABLE I. SPEED AND CURVATUIRECONTRAST

Speed	FPS	Curvature
Decrease	Constant	Reduction in minimum curvature of the trajectory
Increase	Constant	Increase in minimum curvature of the trajectory
Constant	Constant	Constant curvature but better following
Increase	Increase	Constant curvature to follow, requiring a high end processing system

V. CONCLUSION AND FUTURE WORK

The image processing based trajectory follower can be implemented for Projection Mapping, Where the robot can process the images of a two or three dimensional objects and spatially map them on the program itself to mimic the real environment. The proposed method is to allow the robot to interact to fit the desired image of the surface of that environment. By traversing the whole surface and then calculating the images taken from it, it can finally generate the map.

The perception based system is considerably more propitious it can be further implemented in the sector of Swarm Robotics for search and rescue, Planetary or underwater exploration and surveillance. The local interaction between the robots and the environment in which they are present, results in the collective behavior of the robots, which is improved using Swarm Intelligence criterion. A fault tolerant approach will be used for maintaining the ideal application domain for the robots swarms. The robots access of data and programs will be conducted using cloud computing. So that the synced data and other information can be accessed from the web. Cloud computing depends on sharing computer resources rather than having local servers when it handles a particular application. It uses networks of large group of servers which typically runs low cost consumer PC technology connected specifically to share data processing chores across them.

ACKNOWLEDGMENT

This project is supported by a grant from the Independent University of Bangladesh (IUB).

REFERENCES

[1] Beccari, Giuseppe, Stefano Caselli, Francesco Zanichelli, and A. Calafiore. ”Vision-based line tracking and navigation in structured environments.” In Computational Intelligence in Robotics and Automation, 1997. CIRA’97., Proceedings., 1997 IEEE International Symposium on, pp. 406-411. IEEE, 1997.

[2] Ismail, A. H., H. R. Ramli, M. H. Ahmad, and M. H. Marhaban. ”Vision-based system for line following mobile robot” ISIEA, 2009.

[3] Feng, Liqiang, Johann Borenstein, and H. R. Everett. ” Where Am I?”: Sensors and Methods for Autonomous Mobile Robot Positioning. University of Michigan, 1994.

[4] Dupuis, Jean-Francois, and Marc Parizeau. ”Evolving a Vision-Based Line-Following Robot Controller.” In CRV, vol. 6, p. 75. 2006.

[5] Mondada, Francesco, Luca Maria Gambardella, Dario Floreano, Stefano Nolfi, J-L. Deneuborg, and Marco Dorigo. ”The cooperation of swarm-bots: Physical interactions in collective robotics.” IEEE Robotics & Automation Magazine 12, no. 2 (2005): 21-28.

[6] Kahn, Philip, L. Kitchen, and Edward M. Riseman. ”A fast line finder for vision-guided robot navigation.” IEEE Transactions on Pattern Analysis and Machine Intelligence 12, no. 11 (1990): 1098-1102.

[7] Das, Aveek K., Rafael Fierro, Vijay Kumar, B. Southall, J. Spletzer, and Camillo J. Taylor. ”Real-time vision-based control of a nonholonomic mobile robot.” In Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, vol. 2, pp. 1714-1719. IEEE, 2001.

[8] Snchez, Larry Andrs Prez, Cesar Augusto Aceros Moreno, and Holguer Andrs Becerra Daza. ”Design and construction of a line follower robot guided by pixels values of a camera connected to an FPGA.”, STSIVA, 2015.

[9] Ramer, Urs. ”Extraction of line structures from photographs of curved objects.” Computer Graphics and Image Processing 4, no. 2 (1975): 81-103.

[10] Amir, Samreen, Ali Akbar Siddiqui, Nimrah Ahmed, and Bhawani Shankar Chowdhry. ”Implementation of line tracking algorithm using Raspberry pi in marine environment.” In 2014 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 1337-1341. IEEE, 2014.

[11] Horak, K., and L. Zalud. ”Image Processing on Raspberry Pi for Mobile Robotics.” International Journal of Signal Processing Systems 4, no. 2: 1-5.

[12] Papanikolopoulos, Nikolaos P., Pradeep K. Khosla, and Takeo Kanade. ”Visual tracking of a moving target by a camera mounted on

[13] Rahman, Miftahur, Md Hasnaeen Rizvi Rahman, Abul L. Haque, and M. Towhidul Islam. ”Architecture of the vision system of a line following mobile robot operating in static environment.” In 2005 Pakistan Section Multitopic Conference, pp. 1-8. IEEE, 2005.

[14] Khondekar Mahabub Akram, Mirza M. Lutfe Elahi, **M. Ashraful Amin**, “Multiple Level Set Region Based Single Line Road Extraction”, *ICMLC, China,* 2013..

[15] Saha, A., Paul, S. K., Hasan, M., Amin, M. A., “Android Based Autonomous Arduino Bot”, CEET, 2015. Available from: http://www.seekdl.org/nm.php?id=5397.

[16] Arabi, A. A., Sarkar, P., Ahmed, F., Rafie, W. R., and Amin, M. A., “2D Mapping and Vertex Finding Method for Path Planning in Autonomous Obstacle Avoidance Robotic System”, ICCRE 2017