Remotely Controllable Driverless Vehicle Equipped With Computer Vision

Abstract. In this paper a self-driving car is designed. This car is equipped with camera. Here we used histogram analysis to easily identify road. Using histogram method requires little memory and low processing power. Thus a Raspberry Pi computer is utilized. Some pre-processing is applied such as binary thresholding, perspective warping, Region of Interest. The visual system can work with any colored path. This car can drive itself on the road even if the road is curved. Drive optimization is done by averaging method for smoothing out road following movement. In this novel method, use of histogram makes the process applicable in small computers. Though small computers like Raspberry Pi have very small amount of computing capability, our system is capable to run the full operation. The image processing method we proposed here is far better for efficient use of memory and processing power. We used histogram for its superiority in any color intensity variation.

It is also designed for remote control so surveillance works can be carried out. The system is designed as it is expandable and modular. The modular approach make it fit for small scale applications and quick integration. The car is prototype and can be used for further experimentation and improvement. Our presented remote control is suitable for portable setups

Keywords: Histogram, HSV Thresholding, Raspberry Pi, Remote Control, colored path follow.

1 INTRODUCTION

A driverless vehicle is also known as self-driving car. It implements a hardware and software integration to enable a vehicle to sense its environment and moving safely without any human input. It generally combines a variety of sensors to perceive its surroundings. Advanced Control systems interpret sensor values and determines optimum navigation path. There is a great amount of road accidents occurring in Bangladesh due to two way roads. Most of them are caused by face-to-face collision. As reckless drivers try to overtake, they jump into the wrong lane and this creates catastrophic accidents. According to SAE International, there is six levels of automation.

Implementing self-driving improves road safety by incorporating automation in cars. Images from front camera are processed by on board computer to detect and drive the car on lane. This will increase safety and eliminate wrong lane collisions. Now a days cars are fully manual, drivers need to take full control all the time and every moment. But with the help of this system, the car will be able to detect road and even drive through the lane. The computer takes visual snapshots and with processing detects road. It can drive the car along the lane.

As the system uses modular design, future improvements are possible by integrating new sensors and updating software.

Also the car has remote control feature. It can be remotely controlled. When remote control mode is activated, the driver operates the car using his remote from a safe and remote position. For autonomous vehicle, remote control is an emergency feature. Whenever there is present such an environment, which the system is not designed for; then an external human operator should operate the vehicle. Also remote control is very useful in surveillance operations.

Patiyuth et al [1] proposed a system that is mostly accurate in an environment where there is no object or obstruction, the road is clearly seen and the light is not too bright. They used Hough transformation for lane line detection. Pritha Gupta et al [2] proposed an autonomous car that also features obstacle avoidance and remote control. They uses Hough line transformation. They also have an extra feature, region of interest. The region of interest is a very good way to reduce calculating area in the view. Only lower third of the view is used for finding lane. Narayan P. P. et al [3] uses ultrasonic sensor to avoid obstacles. Though they used Hough line transformation for finding lane lines and gave satisfactory results. They also compressed there video before streaming. Network bandwidth made a significant impact on video transmission. R. Muthalagu et al [4] proposed a better way to detect lane or road markings. They emphasized on preprocessing of the image. Both Sobel operator and color thresholding is used. Sobel operator is related to canny edge detection, which is further used with Hough transformation. At the end they applied sliding window search technique. A third order spline is fitted over first identified points of maxima. The spline marks the first lane. There is a shortcoming in this method, it heavily depends on preprocessing of image. This preprocessing is computationally expensive. Also if there is a steep curve then it might be missed from defined ROI. Rami Watheq Yaseen et al [5] designed a miniature vehicle. Here lane detection is done with Hough transformation. For lane tracking they used Kalman filter. Also they included path planning, lane following, parking strategies even road sign detection. Sobel operator, Hough transformation is the base of their lane detection method.

2 METHODOLOGY

There are two sections in this work. One is the autonomous driving and the other one is remote control. Autonomous driving is a multi-step procedure. The computer takes picture through the camera and process the image frame to find road. After separation of the road from the image background, calculates the curve of the road. If the road goes straight, then command the driver motors to go forward; if the road curves to left, then turns the wheel so the car goes left; if the road curves to right, then the left wheel turns more to make the car turn right. The process then loops again from taking image.

The remote control section is programmed separately. As there is a computer inside the car, a keyboard is set up as the remote. The keyboard is continuously monitored for key-press. Whenever there is a keypress on the selected keys, program run the motors as programmed.

2.1 System Design

The system mainly consists of a central processing computer, input and output. This is shown in following illustration [Fig. 1]. All the logics are stored in the memory of Raspberry Pi computer. All the coding is done with python3 programming language. As there is plentiful resource available for python3 and OpenCV, it is easy to prepare a car with these two programs in the least amount of time. This setup is useful in quick deployment of a running system. We just need to import popular libraries. With this approach, we are able to experiment with road detection methods. We then can change different parameters to check viability.

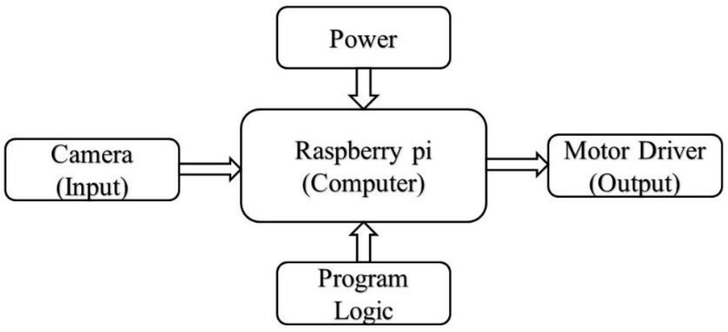

Fig. 1. Main components

Raspberry pi can incorporate all the functionalities of a micro-computer and a microcontroller. For capturing and processing images various libraries are needed. Also a lot of memory is required.

For driving motors we used the GPIO pins of Raspberry pi. This gives us flexibility in direct control of actuators from single interface. The python programming language is flexible enough for even a novice can start with. Python programming language is not suitable for realtime interfacing. But it can help in preliminary stages.

The system deploys a car of two wheeled drive system. Both wheel is independently controllable. It simplified drive system control. “Gpiozero” library for Raspberry pi allows simple control commands for two wheeled cars.

A l293d motor controller IC is used for wheel control. This is manadatory because motors can use high amount of current (from couple of hundreds miliampere to some 1 or 2 Amperes of current). But logic level devices generally uses only half Amperes of current. So motor drive is electrically isolated from logic level electricity.

2.2 Vision Processing

The image processing part is crucial for automatic driving. In this experiment the road is differently colored from the environment. The system detects the road by differentiating the color. The color is specified beforehand. The road can be any color but it should be distinctive.

The image processing pipeline is summarized in following illustration [Fig. 2]

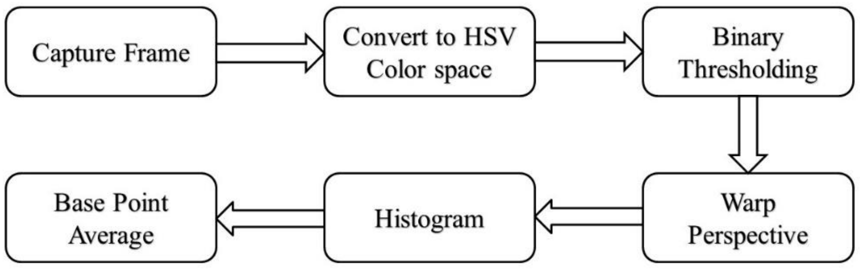

Fig. 2. Image processing pipeline

The frame we capture is in RGB format. To easily separate the desired color, we convert it to HSV color space. Binary thresholding is done by masking with binary value. The pixels matched with our value is retained and any other pixels are zeroed. Threshold operation effect is shown in following illustration [Fig. 3].

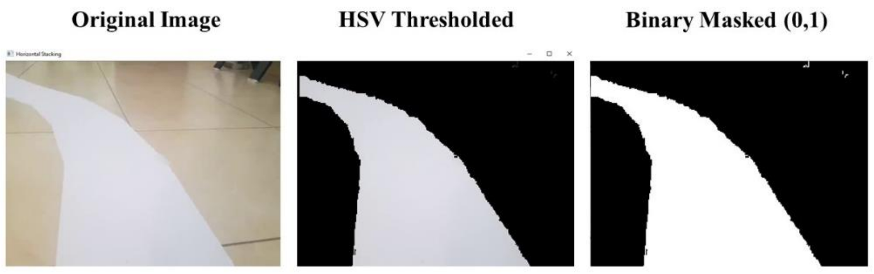

Fig. 3. Binary thresholding (left is original, right side is result)

For better detection of road curvature angle and correct our camera perspective, we warp detected road. It is also called Birds-eye view. For example is shown in following illustration [Fig. 4].

Warping is required for camera calibration. The camera is mounted in the front of the car. The camera captures the road from a perspective view. To rectify the perspective variation, simple 4 point warp is done. We need a top view to determine the curvature correctly. Perspective transformation function of OpenCV library is used to change the perspective to our desired top view form.

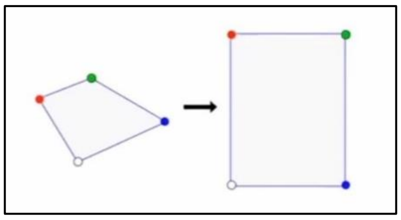

Fig. 4. Birds eye view (Perspective transformation)

After warping the image looks like following illustration [Fig. 5, 6].

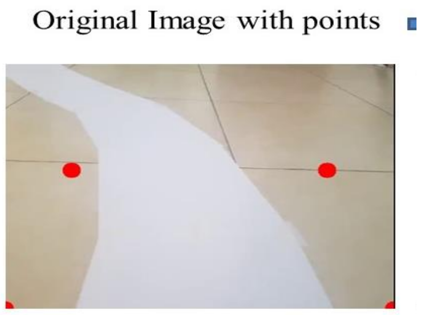

Fig. 5. Before warp

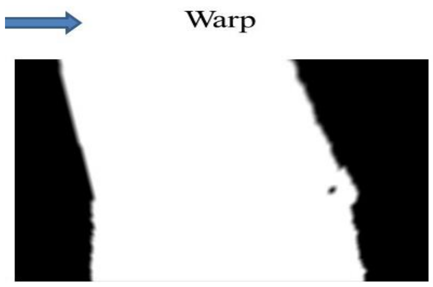

Fig. 6. After warp

Then histogram method gives us values depending on the weight of the pixels of the road. If the road is curved right then the sum of the pixel values is high on the right side. Movement decision criteria is shown in following illustration [Fig. 7].

The histogram method is useful in terms of pixel value change. As the road curves, it's color change can be captured and analyzed easily. This method resembles human's capability of following a different colored path. As long as the path is colored differently from the environment, it is possible to separate. It needs not to be colored artificially. As car or humans pass by the same path, or even there is any regular movement through the same path; the color of the path changes. This can be tracked simply by color thresholding.

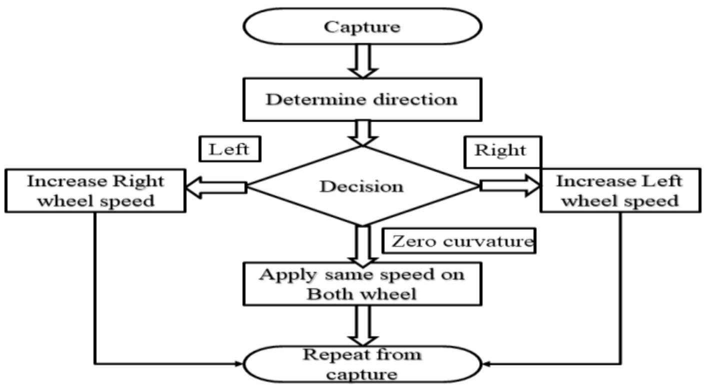
Capture
Determine direction
Left
Right
Decision
Increase Right wheel speed
Increase Left wheel speed
Zero curvature
Apply same speed on Both wheel
Repeat from capture

Fig. 7. Decision flow

On the histogram it looks like a plot of pixels. We summed up the pixels in every column and averaged the columns to get the middle column. The middle column is then set for further calculation of curvature. There is a threshold set for taking the columns. If column value is less than the threshold then it will be counted as garbage value. This column will not be counted as detected road segment.

After getting value for curvature, we average last ten values to reduce arbitrary movement values. The values are stored in a list and the last value is removed. We enter latest value in the list. Then the list values are added and divided by the list counts. Actual results are calculated by subtracting average value to current curve value.

The decision making process depends on the curve value. It is a continuous loop. Decision making process is simplified for easy debugging. Curvature value is not applied for simultaneously going forward and turning. When it turns only turn is applied. Then separately forward applied when curvature value is between certain limit.

More advanced algorithm is available where turn angle of the motor is accompanied with curve value. Though it requires tuning. For compensating image processing lagging, we first lowered motor speed to the lowest, so that the motors just get enough power to make least amount of movement. Then speed is gradually increased to check if the system can accommodate that speed and also correctly follow the path. If with the increased value car cannot follow the path, then the speed is lowered.

2.3 Remote Control

The remote control system is implemented separately for simplicity [Fig. 8]. This requires only two software modules. In a loop any keypress is continuously scanned and recorded. UP, RIGHT and LEFT button is set for matching. If any these three button is pressed, small movement is made via motor control. Keypress is continuously monitored. If no button is pressed down then the car is stopped. This is shown in following flowchart [Fig. 9].

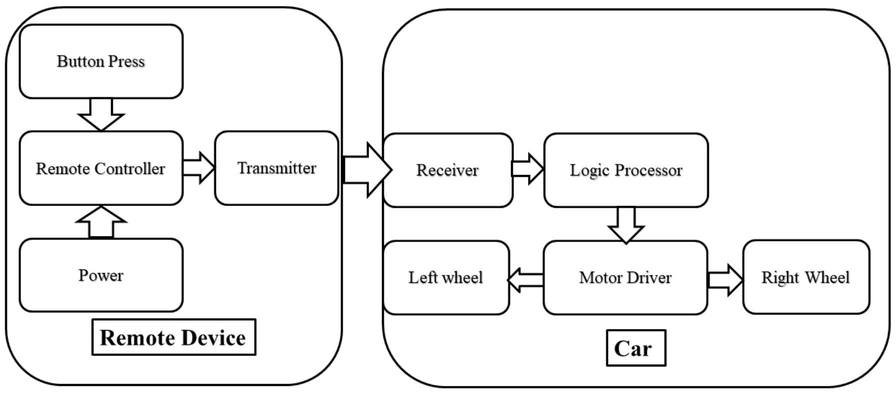

Fig. 8. Remote control system

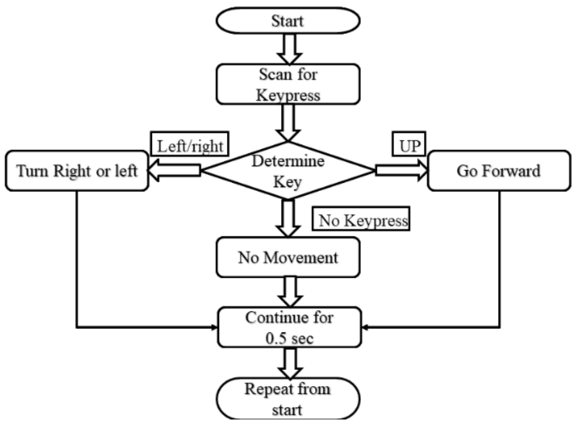

Fig. 9. Remote control algorithm

3 RESULT

The car is working properly on the custom made path in controlled environment. The road is differently colored than the environment. Actual car functional [Fig. 10]

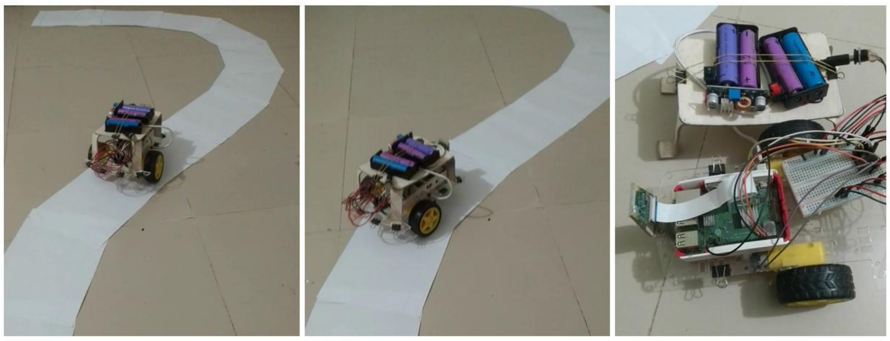

Fig. 10. Actual car photo

3.1 Automatic driving

The car follows the road. If the road take a curve it also follows the curve. We can achieve a high speed run when the road is mostly straight. White color is tested correctly. In a wide curve, the car gets off the road but it follows the curve. The car is not able to follow a very steep curve less than 45 degrees.

There are some limitations in this setup. The car's speed is reduced voluntarily to synchronize image processing time to motor actuation time. As raspberry pi does not have a powerful processor, it is slow to process images. Thus the overall process lags.

3.2 Remote Control

A wireless keyboard connected with Raspberry pi is used as remote controller. As the preselected key is pressed, the computer process the scan and commands the motor to make a small move. Forward, backward, left turn and right turn is working correctly. As the processing is fast, movement seems continuous.

4 CONCLUSION

Using PID controller will reduce jerking movement. This will make the movement smoother. More sensors and road sign recognition can be added. But this requires simultaneously combining all the data and decisions. In this case multi-threading is required.

The remote control scans for button press separately, thus cannot make both forward and side movement simultaneously. Also the controller does not provide information about variable speed. This requires controller with variable input support and more complex programming. There is a limitation of range. It is determined by the wireless technology used in keyboard controller. If internet based remote control is used, then there would be no limitation of range but then another limitation will be added. That is internet coverage. This can be mitigated by satellite communication.

Also the speed of the car can be increased by using powerful computer. Although the car is now operating with usable functionality. These are proposed for future works.

References

1. P. Pramkeaw and C. Thongchaisuratkrul, “Automated Driving System of Lane Detection Using Image Processing Techniques”, International Journal of the Computer, the Internet and Management Vol.27 No.1 (January-April, 2019) pp. 58-63
2. Gurjashan Singh Pannu, Mohammad Dawud Ansari and Pritha Gupta. Article: Design and Implementation of Autonomous Car using Raspberry Pi. International Journal of Computer Applications 113(9):22-29, March 2015
3. N. P. Pawar and M. M. Patil “Driver Assistance System based on Raspberry Pi”, International Journal of Computer Applications (0975 – 8887),Volume 95– No.16, June 2014
4. Raja Muthalagu, Anudeepsekhar Bolimera, V. Kalaichelvi, Lane detection technique based on perspective transformation and histogram analysis for self-driving cars, Computers & Electrical Engineering, Volume 85, 2020, 106653, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2020.106653. (https://www.sciencedirect.com/science/article/pii/S0045790620305085)
5. B. Blaga, M. Deac, R. W. Y. Al-doori, M. Negru and R. D?nescu, "Miniature Autonomous Vehicle Development on Raspberry Pi," 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 2018, pp. 229-236, doi: 10.1109/ICCP.2018.8516589.