Line Tracing Technique for Smooth Driving

John Patrick P. Banjao, Louis Van Hecke, Wansu Lim, Myung-Sik Kim

***Abstract**: In this paper, a line tracing algorithm for a robot was developed. The line tracing algorithm uses the robot's car attached camera to detect the black line. The steering angle of the robot car's wheels is then adjusted according to the black line detected. The performance of the proposed algorithm was evaluated by using the simple line tracking. We found out that using this technique the robot car is having a smooth drive following the black line. This study involves both hardware and software: raspberry pi as the microcontroller for the hardware and C++ for the software.*

***Keywords** : Line tracing algorithm, image processing.*

I. INTRODUCTION

Robot is one of the technologies rising in the industry. Robot is automated machine which program in software, this robot can be programming controlled using embedded software and constructed to make life easier and do the human job. Robot can be autonomous and capable of moving independently using programming and perform complex actions, this can be also replica of human being. One of the robots made up by the researchers and programmers is the four-wheeled car robot which can be basically used into line tracing robot. Line tracing robot is a machine that can detect a line a following the path which is predesigned and can be a black line on a white surface. This kind of program is usually used in the factory, where the researchers studying algorithm and mathematical model that computer systems use to progressively improve their performance on a specific task using of microcontroller.

Microcontroller is a small computer on a single integrated circuit to govern a specific operation in an embedded system. The usual microcontroller includes the memory, input and out(I/O) and processor. This can be varied to many applications and the memory ranges 4-bit, 8-bit, 16-bit, 32-bit and 64-bit microprocessor it depends on the complex of the program [1]. Microcontroller can use random access memory (RAM), flash memory, Erasable Programmable Read-Only Memory (EPROM) and Electrically Erasable Programmable Read-Only Memory (EEPROM). Flash memory can be electrically erased and reprogrammed. EPROM is a type of memory retain its data when its power supply is totally switched off and retrieve the data that is programmed when it is turned on. EEPROM is usually used in smart cards and remote keyless system, and other electronic devices to store relatively amount of data and can be re-write memory chip [2].

Many different kinds microcontroller in the world but there are two development board where usually used in the industry which called Arduino and Raspberry pi. Arduino is an open-source hardware and software, the company design this microcontroller to designed or building a digital device that can be sense of an object, moving the object using the program or control the objects using physical and digital world [3]. The other one is called Raspberry Pi. This was first developed in in United Kingdom by the Raspberry Pi Foundation to promote teaching computer science in countries to have an innovative world [4].

This paper focused on constructing an automated robot car following black line on the white surface. Alongside, the researchers also aim: to understand the environment of robot car interfaced using the Raspberry pi and to solve complexity of hardware design construction. Also, this project aimed to familiarize coding in the raspberry pi using the command prompt and to experiment on the compatibility of the working relation among the Raspberry pi microcontroller and other components that is belong in this study research.

The purpose of this paper is to improve and support electrical systems for the benefit of the community. One of the ways that is occurring today is the usage in deliveries of parcels. By this research paper, this can be contributed and helping the growing and innovating society by using some of the ideas being analyzed and tested by the researchers. This can be used in many areas like for example in domestic application, these can be used like for domestic purposes like cleaning the floor. Using also for using a guide for blind people going to mall, public places or other places to provide the path. Industrial application can be used as automated equipment carries in industries replacing traditional conveyer belt. For automobile application, using line follow running on the road following the path that is to designated place or can use also in military purposes with specific dimension and weight according to the military services. Many ways to generate a robot creating a human's life easier and effortless.

Revised Manuscript Received on November 30, 2019.
* Correspondence Author

John Patrick P. Banjao*, Department of Computer Engineering, Mapúa University, Manila, Philippines.

Louis Van Hecke, Department of Computer Engineering, Pôle Universitaire Léonard de Vinci, Paris, France

Wansu Lim*, Department of IT convergence Engineering, Kumoh National Institute and Technology(KIT), Gumi, South Korea. Email: wansu.lim@kumoh.ac.kr

Myung-Sik Kim*, School of Electronic Engineering, Kumoh National Institute and Technology(KIT), Gumi, South Korea

II. REVIEW OF RELATED LITERATURE

The three laws of robotics were made and defined by the Russian science fiction author Isaac Asimov or in shorter term “Asimov’s Law”. The Asimov’s Law is quoted from the “Handbook of Robotics, 56th Edition, 2058 A.D” [5,6] which is basically A robot should not mutilate or harm to human life physically.

The Asimov’s law of robotics has become a key part of a science fiction culture that has gradually become mainstream, by this nature of law the researchers might know what the robots they should make.

The first industrial robot was made by an American inventor George Charles Devol Jr. invented the first digital operated and programmable robot in late 50’s or 60’s together with his partner Joseph Engelberger. The robot was Unimates built that went online in a General Motors automobile factory in New jersey. The line detection concept on a line-follower four-wheeled car robot is to utilize the color that is captured by the camera to guide path. Many researches have been implemented and using some of the theories [7-12]. One of that theory is Fuzzy-trace theory (FFT) this theory proposed by Charles Breinard and Valerie F. Reyna. The concept of fuzzy set theory can be applied to a routing system and to predict and explain cognitive phenomena, particularly in the memory and reasoning domains.

RGB color line following robot can be used in line tracing and constructed by using light dependent resistor (LDR) and three-color Light Emitting Diode (LED) which is called Blue LED, Red LED and green LED. LDR is a resistance that changes the light intensity upon it [7]. When the color is too bright and high intensity the LDR will have a low value. So, LED can be a path guiding to a sensor to sense the reflected light and follow the desired color. [13] The researchers have studied the fuzzy logic simulation they process the LDR-based sensor output voltages; this simulation uses the input from sensor and generate the motor control if when to turn right or turn left. As you can see in the figure 1, the sample hardware for a four-wheeled car wherein the usual sample of ideal sensor located in the middle of the robot car so when the IR sensor sense color it will be in the middle and following the track middle.

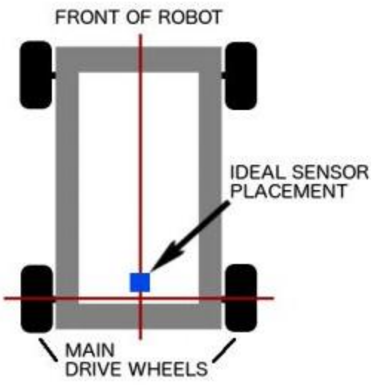


Fig. 1.Sample hardware of four-wheeled car with infrared sensor.

Ralph [14] introduces also in the research industry, this is an navlab autonomous vehicles, obtaining a hybrid version vision system that tracks a road and cars better than those two system in isolation. Ralph takes more global image processing approach and can provide this missing information, as well as a good estimate of curvature so that the combined curvature estimate is superior to both taken in isolation, also this improves also in tracking performance estimating the tracked performance of the car. If the tracking is better, a better where the road is and realize to implemented a fully autonomous vehicle.

One of the example tracing a line is “Bresenham's line algorithm” this algorithm determines the points of an n-dimensional raster that should be selected in order to form a close approximation to a straight line between two points. 11Second example could be “Line drawing Algorithm” this is graphical algorithm for approximating a line segment on discrete graphical media. On discrete media, such as pixel-based displays and prints, line drawing requires such an approximation. The elaborated concepts that has been mentioned is what we came up to this research paper following the line and driving smoothly using the gathered algorithm of RGB line following and concepts in navlab image processing.

III. METHODOLOGY

The line tracing robot was designed to follow a line that may be physical on the floor, the four-wheeled car robot was mounted with camera, and the camera will do the job communicating in microcontroller by sensing the line [9]. An algorithm was constructed using Altino, a robot car manufactured by SAEON in South Korea [15]. Altino is interfaced with a Raspberry Pi microcontroller. OpenCV codes are used to determine the designated color programmed which is black color. The representation of system architecture is shown in Fig. 2. The connection between the hardware and software parts is highlighted in the figure to provide information about the behavior of the system.

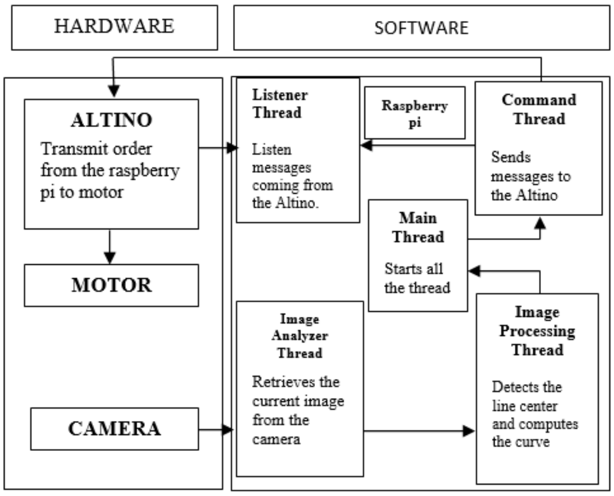

HARDWARE
SOFTWARE
ALTINO
Transmit order from the raspberry pi to motor
MOTOR
CAMERA
Listener Thread
Listen messages coming from the Altino.
Raspberry pi
Command Thread
Sends messages to the Altino
Main Thread
Starts all the thread
Image Analyzer Thread
Retrieves the current image from the camera
Image Processing Thread
Detects the line center and computes the curve

Fig. 2.System architecture.

Figure 2 shows the hardware and software structures of the system. For the hardware part there are 5 parts listener, command, image analyzer, image processing and main thread.

The process starts by capturing images with cameras. The captured image is then sent to the Raspberry pi. The Raspberry Pi analyses the image. The first one that received the data is the Image Analyzer Thread. The first thread involves recapturing the image with the camera and saving the data. After processing the image, it is time for the hardware for second component.

Image Processing Thread is detecting the line that is captured by the camera and analyze by the hardware, this is also to computes the curve of the line where to turn. Main thread, in this part this where programming starts and sending command going to another thread. Command thread, this thread is where the software communicating to the hardware and sending message what contained on the main thread. Listener thread function is to listen what does Altino react to the command thread and giving back feedback of the program. For hardware the robot car is receiving the commands from the raspberry pi wherein it communicate also to motor for turning right or left. The system required also for hardware requirements, wherein the Raspberry pi is the board controls of the whole system, 12 Volts power supply is used for power supply unit, the same with the software. Software needs a cable to connect from the Altino directly connected to a computer so that the user can control the Altino and do the programming side. In achieving the task, the controller is loaded with written Embedded in C++ languages with open computer vision codes. To achieve this the user must know about the environment of programming language like java, python, C or C++. The idea of this program is to follow the black line or line tracing and there are many different algorithms to guide us to smoother program. [10] Many algorithms, but this paper focus on making a difference on other algorithm and make its own and send commands to a robot, while the robot send the required information to the motor. Since the program is already programmed with the OpenCV it needs to implement about driving the car on which track should the car go.

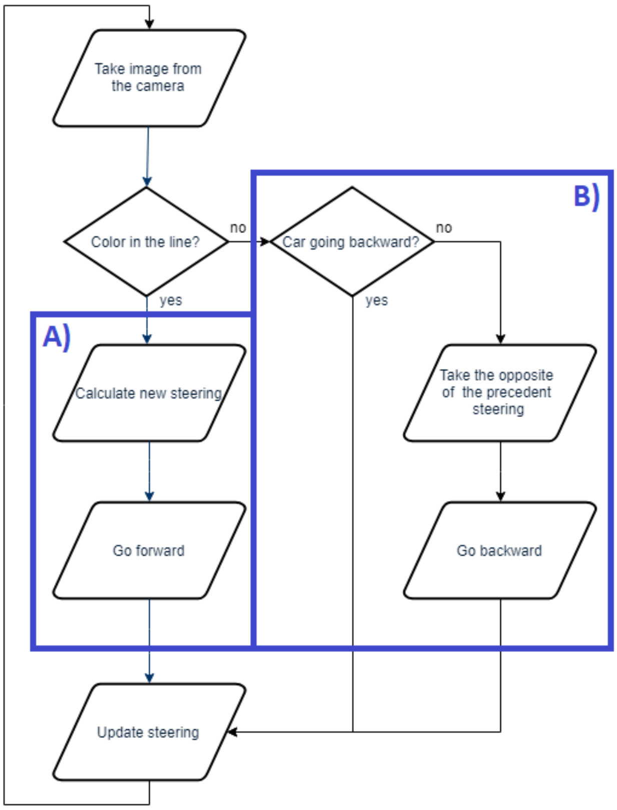


Fig. 3. Flow chart of the Line Tracing algorithm.

The objective of the flow chart of Fig.3 is to determine the steering of the car and the direction of the car. Depending of the situation, the car is going to go forward or backward, and the calculation of the steering will change with the direction. We have on the flow two parts, the part A when we have some pixels of the chosen color in the part of the image that the algorithm is treating and the part B when we do not have any pixels of the chosen color. After this, we update the steering of the car and we go back to the beginning of the code to treat the new image captured by the camera. We use a code to calculate the number of pixels of the chosen color in a define row divide in eight columns of same width. The number of pixels in each part is stock in an array which name is pixelCount and the total number of pixels of the chosen color is stock in the variable which name is totalPixelCount. We enter in the for loop to read through all the value of the array and every time we multiply the number of pixels by the define steering of the block. For example, the steering of the block on the left is -127 and for the block on the right it’s +127. After exiting the for loop, we divide the steering by totalPixelCount, the number total of pixels, to have our steering between -127 and 127 for the four-wheeled car.

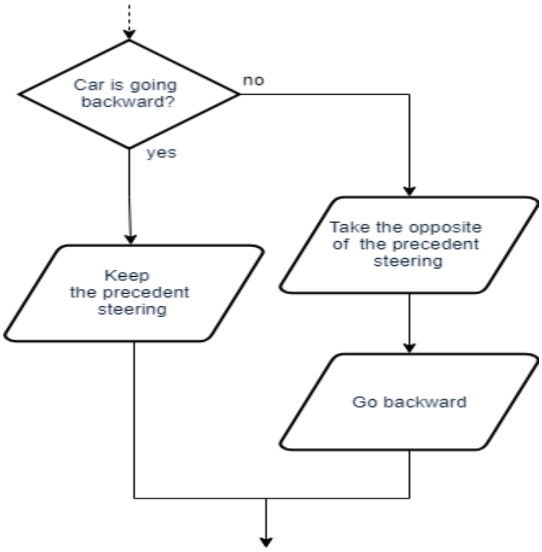

Car is going backward?
no
yes
Keep the precedent steering
Take the opposite of the precedent steering
Go backward

Fig. 4. Flow chart for the direction of robot car.

Before the calculation of the steering, we see if we have at least a pixel of the chosen color in the row and if it is not the case we enter in the two if cases of the Fig. 4. It represents here the case B of the Fig.3. We have again two different cases. The first one is when the car was not going backward and in this case the steering of the car is now the opposite steering of the car just before it lost the line and the car is now going backward. The other case is when the car is already going backward so in this case the car just continues to go backward with same steering. It will continue until the car sees some pixels of the chosen color in which case, we use the code of the Fig.3.

IV. PERFORMANCE EVALUATION

For the performance evaluation, we use an Altino to come up with the line tracing algorithm. We used a track which you can identify the in the Fig. 5 with composed of different turns where turns have different angle. In the track, Altino can follow the black line and driving smoothly using the technique that was proposed in this paper.

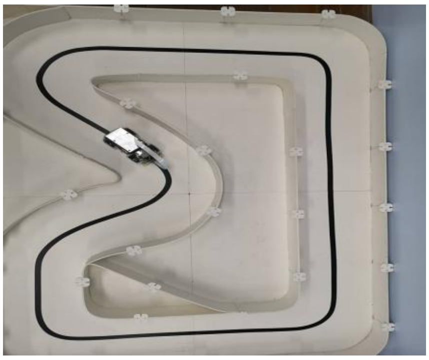
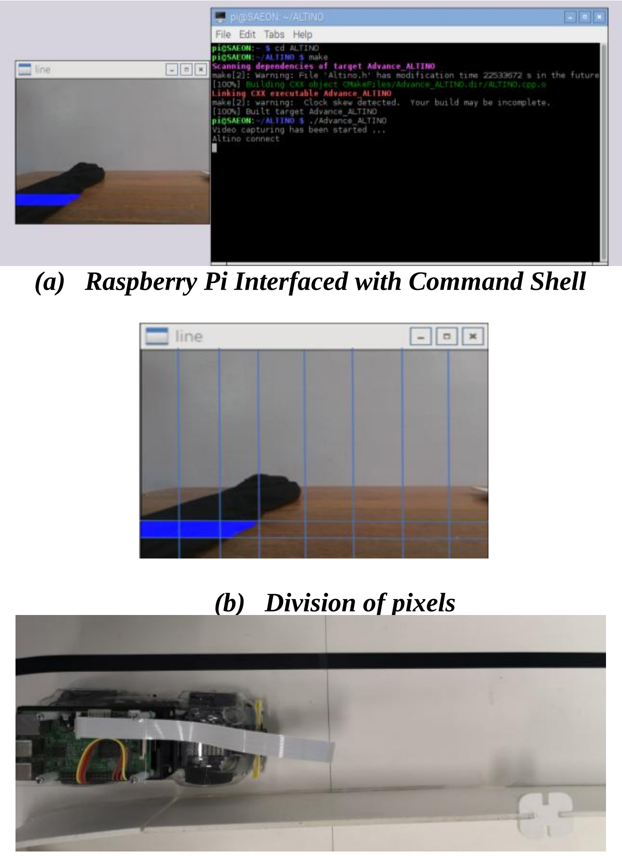


Fig. 5. Track line with Altino.

Using the technique RGB color space and canny edge detection converting images to grey scale, the program can identify the road if the road in the camera has identified the respective color of the line in the program. The researchers used a reference also using the RGB color code. This is wherein it constructs all the color in Red, Green and Blue.



```
pi@SAEON:~ $ cd ALTINO
pi@SAEON:~/ALTINO $ make
Scanning dependencies of target Advance_ALTINO
make[2]: Warning: File 'Altino.h' has modification time 22533672 s in the future
[100%] Building CXX object CMakeFiles/Advance_ALTINO.dir/ALTINO.cpp.o
Linking CXX executable Advance_ALTINO
make[2]: warning:  Clock skew detected.  Your build may be incomplete.
[100%] Built target Advance_ALTINO
pi@SAEON:~/ALTINO $ ./Advance_ALTINO
Video capturing has been started ...
Altino connect
```

(a) Raspberry Pi Interfaced with Command Shell

(b) Division of pixels

(c) Altino robot in the road track

Fig. 6. Raspberry Pi interface with command shell and Altino.

In Fig. 6(a), at the left most of the picture there is a black object that is on the camera and blue shading the black one. For Fig. 6(b), using the algorithm by dividing the vertical by 8 and have a strip in the the horizontal part. Raspberry pi camera when it is turned on, it will determine the number of pixels per column and row and the Altino program now determining if which is the part has the most higest pixels, by using this the logic program that can be figured out in the Algorithm section. For Fig. 6(c), this is the time when the car will detect that Altino and analyzing if the robot car should turn left or right because of detecting the black line. At this moment the Altino will turn left because the majority of the pixels of the track are on the left side.

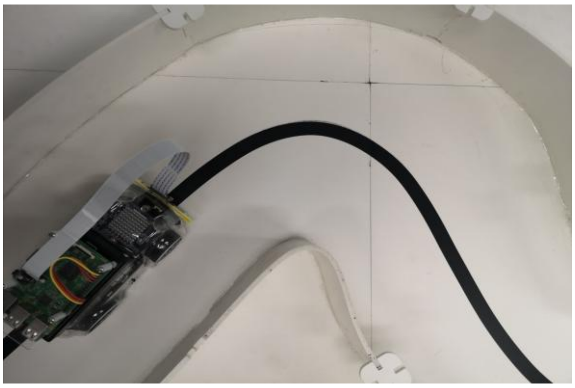
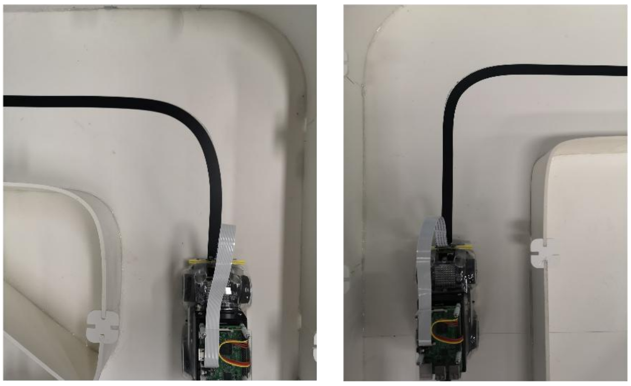


Fig. 7. Altino car in the track road.

In Fig. 7 shown, Altino is having a degree for turning right at this manner, the car will go drive to the right slightly until no black pixels detected. When no black pixels detected this is the time where the robot will go backward and slightly turn to the old steering value finding new black pixels to continue driving. Since we prioritize driving smoothly following the black pixels that is why it is turning at this manner.



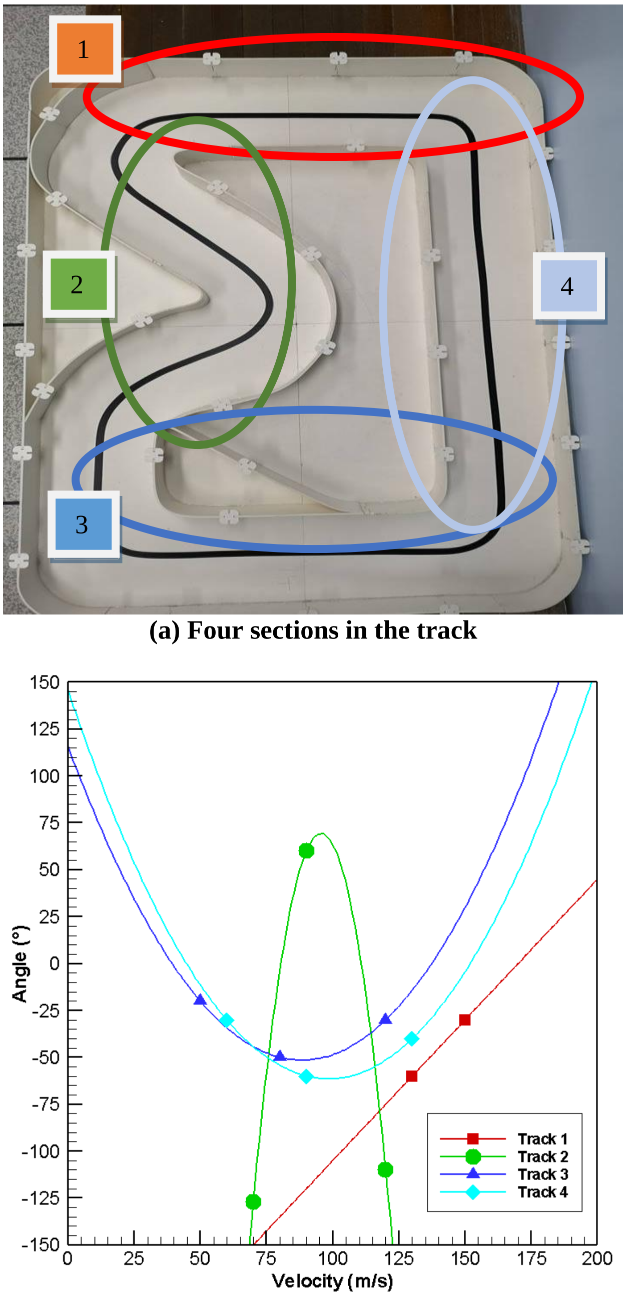
(a) Turning Left

(b) Turning Right

Fig. 8. Altino robot car turning right and left.

In Fig. 8(a), it is shown that the track having a solid turning left which is the black line wherein the Altino is smoothly rotating to respective side where the black pixels is going, in this manner the Altino is not turning back because having a perpendicular angle of the black line. In the manner for turning right,

Fig. 8(b), same with the behavior of part a wherein it will go drive turning right and continuing following the black pixels. In Fig. 9(a), each section has a different speed and turning point of the wheel wherein the angle of the caster wheel is changing each time. The purpose of this measurement is to determine the angle and how speed will the Altino speed when it is turning. For example, in the section 1, it is a straight line and when the Altino goes from right to left. The left corner is going to left wherein the manner of the wheel is to turning left smoothly. For the section 2, having an angle turning first from left and turning to right in such manner that the Altino will do many different turns and checking the accuracy test for examining the turning point of the wheel. For the section 3, same behavior for the first section but with different manner of speed and angle. Same as the section 4.



(a) Four sections in the track

(b) Angle and velocity measurements in each section

Fig. 9. Results of angle and velocity dependency.

Fig. 9(b) represents how speed of Altino will do what section, for the red part it is linear regression wherein it is straight and sometimes the angle is in -25 or -50, for the green is which is y=x2 the speed of the car and it is not straight and the point angle is for going right is 75 and left -125 & -100. While for the dark blue and sky blue is parabola but always going to left. The result show that the algorithm needs more calibration that should be done by the future researchers.

V. CONCLUSION

Through the gathering data from a application of algorithm, the researcher were able to implement line tracing algorithm using the PaLo Technique. During controlled testing, the researchers were able to tuned the program of the Altino robot car, in order to arrive with the best result in terms of accuracy and smooth driving.

Retrieval Number: A4630119119/2019©BEIESP
DOI: 10.35940/ijitee.A4630.119119
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

It is highly recommended for future researchers who wish to pursue to the study to utilize the device, which can provide high accuracy for turning the Altino in the accurate angle. The device Altino aimed to drive smoothly in the track with the black line that raspberry do read the black pixels. In by doing so, the device further aimed to help in future industrial application automated carrying equipment in the factories, military or in the airport doing such thing following the path and making the world being automotive.

ACKNOWLEDGMENT

This work was supported by Kumoh National Institute of Technology(2019-104-155).

REFERENCES

1. IEEE Standard for Microcontroller System Serial Control Bus," in IEEE Std 1118-1990 , vol., no., pp.1-252, 31 Jan. 1991
2. J. Paterson, "Adding analog, EPROM and EEPROM modules to CMOS logic technology: how modular?," International Technical Digest on Electron Devices Meeting, Washington, DC, USA, 1989, pp. 413-416.
3. A. A. Galadima, "Arduino as a learning tool," 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, 2014, pp. 1-4.
4. F. Salih and S. A. Mysoon Omer, "Raspberry pi as a Video Server," 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum, 2018, pp. 1-4.
5. Azizul, H., Sazia, A., Hossain, M. M., March 2015, Northern university of Bangladesh, Design & Implementation of Line tracking Robot
6. L. A. Ballard, S. Sabanovic, J. Kaur and S. Milojevic, "George Charles Devol, Jr. [History]," in IEEE Robotics & Automation Magazine, vol. 19, no. 3, pp. 114-119, Sept. 2012.
7. G. Sonal, P. Raninga and H. Patel, "Design and implementation of RGB color line following robot," 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, 2017, pp. 442-446.
8. J. Thirumurugan, M. Vinoth, G. Kartheeswaran and M. Vishwanathan, "Line following robot for library inventory management system," INTERACT-2010, Chennai, 2010, pp. 1-3.
9. Yuan Liao, "Some algorithms for transmission line parameter estimation," 2009 41st Southeastern Symposium on System Theory, Tullahoma, TN, 2009, pp. 127-132.
10. X. Wang, G. Geng, X. Li and J. Zhou, "A Cultural Relic Line Drawings Generation Algorithm Based on Explicit Ridge Line," 2015 International Conference on Virtual Reality and Visualization (ICVRV), Xiamen, 2015, pp. 173-176.
11. P. Sosnin, "Pseudocode programming of human-computer interaction in collaborative designing," International Conference on Electrical, Control and Computer Engineering 2011 (InECCE), Pahang, 2011, pp. 317-322.
12. M. B. Nugraha, P. R. Ardianto and D. Darlis, "Design and implementation of RFID line-follower robot system with color detection capability using fuzzy logic," 2015 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, 2015, pp. 75-78.
13. F. Dellaert, D. Pomerlau and C. Thorpe, "Model-based car tracking integrated with a road-follower," Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), Leuven, Belgium, 1998, pp. 1889-1894 vol.3.
14. M. Ozaki, Y. Adachi, N. Ishii and T. Koyazu, "CAI system to improve hand writing skills by means of fuzzy theory," Proceedings of 1995 IEEE International Conference on Fuzzy Systems., Yokohama, Japan, 1995, pp. 491-496 vol.2.
15. SAEON, http://www.saeon.co.kr

AUTHORS PROFILE

John Patrick P. Banjao is currently studying in Department of Computer Engineering, Mapúa University, Manila, Philippines.

Louis Van Hecke is currently studying in Department of Computer Engineering, Pôle Universitaire Léonard de Vinci, Paris, France.

Wansu Lim received the Ph.D. degree from the Gwangju Institute of Science and Technology (GIST), South Korea, in 2010. Since 2014, he has been an Assistant Professor with the Kumoh National Institute of Technology (KIT), South Korea.

Myung-Sik Kim received the Ph.D. degree from the Korea Advanced Institute of Science and Technology (KAIST) in 1992. Since 1992, he has been a Professor with the Kumoh National Institute of Technology (KIT), South Korea.