Volume 6, No. 11, October 2015
Journal of Global Research in Computer Science
RESEARCH PAPER
http://rroij.com/global-research-in-computer-science.php
ISSN-2229-371X

STRAIGHT-LINE MOBILE ROBOT USING VIDEO PROCESSING TECHNIQUE

Mohd Fauzi Alias*, Mohamad Amir Adni Mansor, Mohamad Rosyidi Ahmad, Norzalina Othman
Electrical, Electronics and Automation Section, Universiti Kuala Lumpur, Malaysian Spanish Institute, Kulim
fauzialias@unikl.edu.my

Abstract: This paper presents a method for relocation a mobile robot using video processing technique. This method is able to detect the mapped line and followed by the mobile robot. This method consists of line detection and path recovery techniques to relocate the mobile robot automatically. Based on the application, images captured from webcam are used for detecting the line trajectory. Error pathway followed by mobile robot generates path's error data using line detection technique. Based on path's error data, speed and time period supplied to the mobile robot's wheels have been measured and experimented. The project implemented high definition webcam 1080hd and Arduino board to capture the video and relocate the mobile robot respectively. This video was framed and formed into 2D sequential images.

Keywords- Video processing, Arduino, Line detection, Path recovery, Motor Control.

INTRODUCTION

Navigation is an important factor of mobile robot movement efficiency. Robot's movement is controlled by relocation and localization terms which determine the positioning of a mobile robot respect to a global reference frame [1]. High accuracy in localization and relocation determine good initial position of a mobile robot and error correction in positioning respectively. There has been a lot of works on localization of mobile robot using sonar, IR sensor, but less work has been done on relocation [1]. Based on this factor, this project implements localization but focused on relocation of a mobile robot using image processing technique.

Straight-line detection mobile robot is one of navigation robot's types. The current researches of line tracking mobile robot using image processing based are quite popular instead of sensors. Based on the image processing technique, the vision-based systems of mobile robots are designed to extract the desired information from the images. Using the vision-based mobile robot, improvement in line tracking robot can be done accurately by applying related engineering equations. Furthermore, implementation of virtual line based can also be done for robot's control purpose especially on autonomous surface.

Vision-based line tracking system offers a lot of advantages in term of variety of informations compared to the ranging devices [2]. Therefore, the vision-based tracking mobile robot becomes an interesting topic to the research community due to the ability to extract information about the environment, which is not available in any combination of other types of sensors [3]. It offers better accuracy and reliability in vary lighting condition [4]. Besides, the price of the vision system especially high definition webcams are cheaper than any other sensors because the increasing of application to mobile robots [5].

Application of the CCD camera as image acquisition technique has been proven to result good performance in line detection [6]. Wuhan University applied the CCD image sensor in the four-wheeled line track robot and proved resulting better performance in line detection. This proved that the vision-based mobile robot results better detection rather than sensor based. The vision based system offers a lot of information in a single image thus makes the system more flexible. The Vision-Based Automated Guided Vehicle (V-AGV) from Universiti Teknikal Melaka (UTeM) was used to identify the guideline, signboard and obstacle simultaneously [7]. The V-AGV has successfully developed with functioning straight line detector.

VISION BASED MOBILE ROBOT

This paper uses vision based mobile robot system in detecting straight mapped line as well as robot's relocation. It consists of camera 1080hd, a mobile robot, MATLAB and Arduino board. The project has been divided into 2 units namely line detection and path recovery units as shown as Figure 1. Line detection unit implements image acquisition, binarization and line detection techniques. These techniques have been applied purposely for detecting error pathway by obtaining the accuracy of line detection from the robot's eyes. Image acquisition technique controls the quantity of framed images collected from camera. Image quantity reflects to the line detection's sensitivity and accuracy. Large quantity of image frames indicates high sensitivity of line detection.

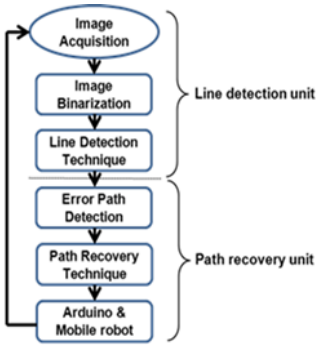

Figure 1: Method flowchart.

Line detection technique contributes the path's error data in form of degree unit. In order to return to the right position, these data will be implemented in the path recovery unit to determine speed and time period needed to be supplied to the robot's wheels. Based on the robot's eyes application, the system maps the virtual line and compares with the actual detected line (from the robot's eyes). The concept of comparing the virtual and actual detected lines is shown in Figure 2 and will further discuss in the next section. The experimental setup is shown as Figure 3.

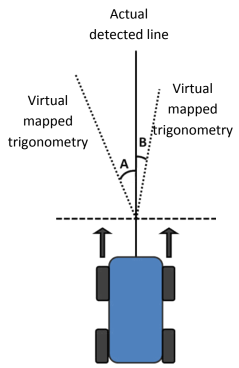

Figure 2: Mapped trigonometry on the robot's pathway.

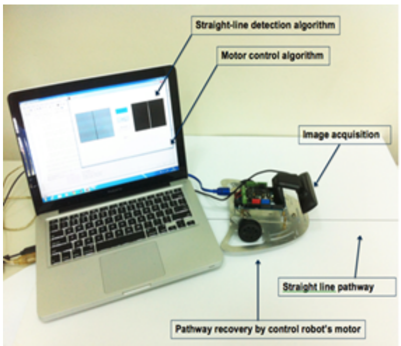

Figure 3: Experimental setup.

Line detection unit

Line detection unit implements a lightweight webcam for the mobile robot to capture real time images. Image binarization technique has been applied to reduce the noise and unwanted information from true color.

Binarization technique converts the gray scale and true color to binary image. Binarization produces white and black pixel indicates the detected line and background respectively. After the binarization process, line detection technique has been applied to extract the path's error data. The data appear in degree form and indicate the error of pathway done by the mobile robot. The path's error data has been measured based on trigonometry concept as expressed as equation (1) and Figure 4.

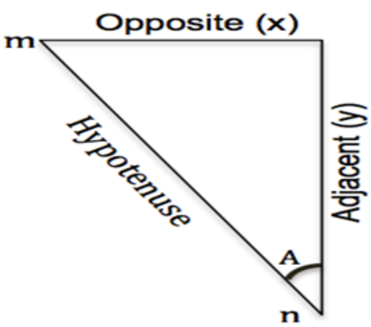

Figure 4: Trigonometry.

$$\tan A = \frac{\sin A}{\cos A} = \frac{opposite}{adjacent} = \frac{x}{y} \qquad \text{Equation (1)}$$

Trigonometry studies of triangle and the relationship between lengths, angle and opposite side. Based on Figure 4, the trigonometry structure has been mapped overlapped with the actual detected line. Two parallel points of white pixels have been selected automatically as center points, n. The error degree, A is measured based on the lengths of Hypotenuse; m and n. Hypotenuse is measured based on the Adjacent (y) and Opposite (x). Furthermore, the trigonometry is virtually mapped and overlapped on the top of the real detected line ahead of the robot's pathway. The process has been done using line detection algorithm. Based on the virtual mapped structure, the error pathway has been evaluated by comparing the Adjacent (y) and the detected lines. The A value determine the dispersion degree between the Adjacent and detected line. As an example, the 0 and 40 degree of A indicates the zero error and 40 degree error of pathway.

This process is repeated several times to produce an accurate result of A. The virtual mapped of trigonometry structure as shown as Figure 2.

Based on Figure 2, vision system is mounted ahead of the mobile robot for early line detection to ensure the right position of robot's movement due to the system's delay. Value of A and B represent the error pathway of left and right directions respectively. These values will be used for the robot's path recovery purpose in path recovery unit.

Path recovery unit

Path recovery unit consists of error pathway detection, path recovery technique and; Arduino and mobile robot controller. Error pathway detection is determined by comparing the degree result from the line detection unit to the straight line, 0 degree. Degree of 0° value indicates the straight line. A° and B° degrees represented by positive and negative values indicates left and right directions respectively. These values have been shared to the mobile robot via Matlab and Arduino controller to recover the robot's position as the mapped line.

The position of mobile robot is determined by both rear wheels. The right and left wheels movements produce left and right cornering respectively. The magnitude of cornering is determined by the time period supplied to activate the wheels. Time period supplied reflects to the magnitude of cornering as tabulated in the Tables 1-4.

Table 1. Time period of -10 degree for left wheel.

Number	Left motor	Right motor	Dispersion degree	Time period (s)
1	80	0	-10	0.76
2	80	0	-10	0.75
3	80	0	-10	0.75
4	80	0	-10	0.77
5	80	0	-10	0.76

Table 2. Time period of 10 degree for right wheel.

Number	Left motor	Right motor	Dispersion degree	Time period (s)
1	0	80	10	0.77
2	0	80	10	0.75
3	0	80	10	0.76
4	0	80	10	0.76
5	0	80	10	0.75

Table 3. Time period of -20 degree for left wheel.

Number	Left motor	Right motor	Dispersion degree	Time period (s)
1	80	0	-20	0.98
2	80	0	-20	1.00
3	80	0	-20	1.02
4	80	0	-20	0.98
5	80	0	-20	1.00

Table 4. Time period of 20 degree for right wheel.

Number	Left motor	Right motor	Dispersion degree	Time period (s)
1	0	80	20	1.00
2	0	80	20	0.98
3	0	80	20	1.01
4	0	80	20	1.03
5	0	80	20	1.00

RESULTS & DISCUSSION

Graphical user interface (GUI) has been developed to verify the output of error's degree and path recovery techniques. The GUI is shown as Figure 5.

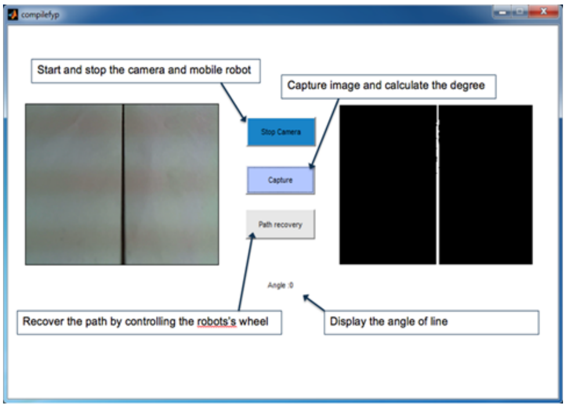

Figure 5: GUI of mobile robot.

Figure 5 consists of a single static text display, 3 pushbuttons and 2 image axes. The raw and binary images are captured and displayed at the left and right axes respectively. Capturing process executes the line detection unit measuring the degree of error's path done by mobile robot. The degree data is displayed through "Angle" static text. The "Path recovery" button executes the recovery of path error using Arduino board as motor's driver. Time period supplied to the Arduino determines the accuracy of path recovery. Experiments have been conducted to measure the performance of line detection and path recovery algorithms.

Line detection

The algorithm of line detection has been experimented at 0, -10, 10, -20 and 20 degrees. These error's degree values have been selected as references and labeled as actual degree. The line detection algorithm produces measurement degree as experimental results. Each actual degree has been experimented 5 times. Tables 5-9 show the measurement degree of line detection algorithm for actual degree of 0, -10, 10, -20 and 20 respectively.

Table 5. Line detection of 0 degree.

Number	Actual degree	Measurement degree
1	0	0
2	0	-1
3	0	0
4	0	0
5	0	0

Table 6. Line detection of -10 degree.

Number	Actual degree	Measurement degree
1	-10	-10
2	-10	-9
3	-10	-10
4	-10	-10
5	-10	-10

Table 7. Line detection of 10 degree.

Number	Actual degree	Measurement degree
1	10	10
2	10	9
3	10	10
4	10	10
5	10	10

Table 8. Line detection of 20 degree.

Number	Actual degree	Measurement degree
1	-20	-20
2	-20	-19
3	-20	-20
4	-20	-20
5	-20	-20

Table 9. Line detection of 20 degree.

Number	Actual degree	Measurement degree
1	20	20
2	20	20
3	20	19
4	20	20
5	20	20

Table 10 shows high accuracy of average measurement line for 0, -10, 10, -20 and 20 degrees with 98% to 100%. The line detection algorithm has been proven to produce similar result of the actual and measurement degrees. Therefore, the line detection algorithm has been implemented in the line detection and path recovery units.

Table 10. Accuracy of average measurement degree.

Actual degree	Number of data	Average Measurement degree	Accuracy (%)
0	5	0	100
-10	5	-9.8	98
10	5	9.8	98
-20	5	-19.8	99
20	5	19.8	99

Path recovery

The measurement degree data in line detection unit has been applied to decide the recovery path from robot's error movement. Speed value data of robot's wheel has been selected as 80. Based on the observation, 80 speed data is an appropriate speed based on the weight and application of mobile robot. The time period has been evaluated separately for each actual degree. The time period has been evaluated for both right and left wheels cornering. Table 7, 8 and 9, 10 show the right and left wheels controlling method for -10, 10 and -20, 20 degree respectively.

Right and left wheels are supplied Arduino based values 80 represents as motor's speed per second. Based on the Table 7, 8 and 9, 10, right and left wheels receive 80 values for -10, 10 and -20, 20 degrees respectively. These values produce different magnitude of mobile robot cornering. It has been proven that the 80 value of speed are compatible with cornering degree of 10 and 20 degrees respectively. Speed value of 80 is compatible for mobile robot based on the short time period needed for path recovery.

Time period for each -10, 10, -20 and 20 degrees have been evaluated for 5 times as shown in the Tables 7-100. At -10, 10 and -20, 20 degrees, the left and right wheels have been activated with 80 speed for 0.75 and 1.0 seconds respectively. After the time period, the mobile robot movement is recovered and returned to the actual mapped line.

CONCLUSION

Line detection and path recovery algorithm have been successful developed in this project. The line detection algorithm has been proven to produce high accuracy with 98% to 100% in detecting mobile robot's error pathway. Based on the result, the line detection algorithm has also been proven to produce small differences between actual and measurement degrees. Based on the experimental result, the line detection algorithm has been successfully implemented in the path recovery unit. The path recovery algorithm has been proven to recover the actual pathway based on 80 speed with almost similar time periods. These algorithms have been embedded into GUI for experimental and verification purposes. As conclusion, this project has been successfully implemented the line detection and path recovery algorithms for straight line mobile robot application.

ACKNOWLEDGEMENTS

I would like to acknowledge Universiti Kuala Lumpur, Malaysian Spanish Institute for the financial support of this project. I would also like to express my gratitude to Electrical, Electronics & Automation department for the laboratories usage.

REFERENCES

[1] Jong Hwan Lim and John J. Leonard, "Mobile Robot Relocation from Echolocation Constraints," IEEE Trans. Pattern Anal. Mach. Intell, vol. 22, no. 9, pp. 1035-1041, 2000.

[2] Jason Campbell, Rahul Sukthankar, Illah Nourbakhsh and Aroon Pahwa, "A Robust Visual Odometry and Precipice Detection System Using Consumer-grade Monocular Vision," Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, 2005.

[3] Mehmet Serdar Guzel and Robert Bicker, "A Behaviour-Based Architecture for Mapless Navigation Using Vision, International Journal of Advanced Robotic Systems," Int J Adv Robotic Sy, vol. 9, no. 18, pp. 13, 2012.

[4] Norhashim Mohd Arshad, Noorfadzli Abdul Razak, "Vision-based Detection Technique For Effective Line-tracking Autonomus Vehicle," Signal Processing and its Applications (CSPA), IEEE 8th International Colloquium, pp 441 – 445, 2012.

[5] Shahed Shojaeipour, Sallehuddin Mohamed Haris and Muhammad Ihsan Khairir, "Vision-Based Mobile Robot Navigation Using Image Processing and Cell Decomposition," Visual Informatics: Bridging Research and Practice Lecture Notes in Computer Science, Vol. 5857, pp 90-96, 2009.

[6] Lijun Zhang, Xiangtao Zhuan, Xukai Gao, "Design and Implementation of a Vision Based Four-Wheeled Line Track Robot," IEEE proceeding of: Information Engineering (ICIE), WASE International Conference, 2010.

[7] Sulaiman Sabikan, Marizan Sulaiman, Sy. Najib Sy. Salim and M. Fahmi Miskon, "Vision-Based Automated Guided Vehicle for Navigation and Obstacle Avoidance," The 2nd International Conference on Engineering and ICT, 2010.