

# SHORTEST PATH PLANNING AND CONTROL ALGORITHM TO FOLLOW THE PATH FOR MOBILE ROBOTS USING VISION

## ĐIỀU KHIỂN HỒ TIẾP ẢNH CHO MOBILE ROBOT TÌM VÀ BÁM THEO ĐƯỜNG ĐI NGẮN NHẤT

*Le Duc Hanh, Nguyen Duy Anh*

*Ho Chi Minh City University of Technology – VNU - HCM*

**Abstract:** This paper presents the shortest pathfinding algorithm for a mobile robot with a global path planning approach in a planar surface and control robot to follow the processed path (virtual). The path is planned based on Image Processing of the real working space map, which is caught by a camera. Firstly, the input image of the map is processed to determine the objects (starting point, goal point, obstacle vertices) as the basis for the road map construction. Then Dijkstra's optimal algorithm is applied to the constructed road map for the shortest path calculation. Eventually, the robot will follow that planned path. Experimental results are considered to be good when the robot can follow the path to reach the goal point with acceptable tracking errors compared with the theoretical simulation results.

**Keywords:** Mobile robot, shortest path, Visibility Graph, Dijkstra, path planning, image processing, obstacle avoidance.

**Classification number:** 2.2

**Tóm tắt:** Bài báo này trình bày phương pháp tìm đường đi ngắn nhất theo bài toán toàn cục cho mobile robot trong môi trường không gian hai chiều có vật cản, và điều khiển mobile robot bám theo đường đi (ảo) ấy. Đường đi được xây dựng dựa trên việc xử lý hình ảnh bản đồ bằng camera đặt trên cao. Với đầu vào là hình ảnh vùng không gian hoạt động, vị trí đầu, điểm đích và các đỉnh vật cản sẽ được xác định, làm cơ sở cho việc xây dựng bản đồ đường đi. Sau đó giải thuật tối ưu Dijkstra sẽ được áp dụng trên bản đồ đường đi vừa xây dựng được để tính toán đường đi ngắn nhất. Cuối cùng robot sẽ bám theo đường đi đã hoạch định. Kết quả thực nghiệm tốt khi robot có thể bám theo đường đi vạch ra đến đích với sai số không quá lớn so với kết quả mô phỏng lý thuyết.

**Từ khóa:** Robot di động, đường đi ngắn nhất, Visibility Graph, Dijkstra, hoạch định quỹ đạo, xử lý ảnh, tránh vật cản.

**Chỉ số phân loại:** 2.2

### 1. Introduction

Nowadays, mobile robots are widely used in many areas such as exploration, security, and defense, transportation, search and rescue, human support in life... Therefore, they have been attracting much scientists' interest. The most considerable problem for mobile robot research is "Navigation". There are many approaches to solve this problem. Nakib Hayat Chowdhury et al. [1] used an on-off control method that combines optical sensors or Juing-Huei Su et al. [2] used infrared sensors and Fuzzy control algorithms for mobile robots to follow the given line. However, methods using a fixed-line have many drawbacks: the line, after a period of use, is blurred and hard to detect, susceptible to the impact of the surrounding environment and complicated to change the route. Since then the solution for the "Navigation" problem, which is based on camera

application, has attracted much interest as it can surmount the limitations of the traditional line-detecting method and also provides the ability to optimize the path.

With the more and more advanced computers in terms of power as well as speed, the use of cameras as a sensor gives mobile robots a high potential to grow. Thanks to the camera, a mobile robot is able to detect the objects of the surrounding operating environment. To identify the objects, the image captured from the camera will be processed to output the coordinates of the objects on the map. Heramb Nandkishor et al. [3] used a combination of two basic image segmentation methods of image processing (edge separation and thresholding) to identify obstacles on the map. Himanshu Borse et al. [4] used a camera mounted directly on the robot to identify objects by comparing object images obtained from the camera and a library

of images of known objects that are stored in memory. This method has practical efficiency for the given objects that need to be identified, but because that it completely depends on the image library so the robot is not able to recognize objects that do not exist in the library. Jeongdae Kim et al. [5] used a fixed camera to help robots avoid collisions with moving objects using the block-based motion estimation (BBME) method. This method allows robots to quickly identify the moving objects, but the camera must be fixed and opposite the moving object, which is often not met in practice. Huu Danh Lam et al. [6] used a CMU camera as a sensor that enables a mobile robot to follow the line optimally at intersections, or G. H. Lee et al. [7] employed a CCD camera to control a two-wheel robot to track the given line.

In addition, the study of the ability to construct virtual paths to overcome the disadvantages of the traditional line-detecting method is also considered. This ability allows the robot to be able to change the direction of movement when it is needed as the operating environment can be altered (such as change of obstacles' positions). This also attracted the attention of researchers. S. Tang et al. [8] wrote a brief review of about 100 studies of path planning. Besides, this paper also used approximately 200 studies to determine the percentages of applications of these path planning methods and the tables and graphs that compare the frequency of application of these methods in the last 30 years.

From the discussions above, using camera and image processing is a very useful method in path planning for mobile robots in the home or warehouses. The idea of the topic is from a starting point, a goal point and obstacles, use camera and open source code to process the map image to find the shortest path in form of waypoints, then use a controller to help the robot follow the obtained path. Besides, during the operation of the robot, the camera also collects the actual position of the robot to calculate the errors compared with theory, which is used as a basis for evaluating the controller.

## 2. Visibility - Graph in “finding the shortest path” problem

Visibility - Graph is one of the road maps based on combinatorial optimization approach, best suited to the problem of finding the shortest path in terms of distance [9]. This road map is constructed from the vertices and edges of the objects on the map, with the following idea: Any path between 2 points can be shortened to a polylines path and the shortest path, which still ensures that robot will not collide with obstacles, is the polylines constructed from the vertices of the obstacles [10]. This idea is illustrated in figure 1.

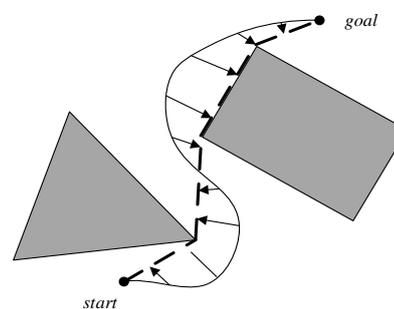


Figure 1. The idea of Visibility Graph.

From that idea, the construction of a Visibility Graph can be done as followings:

The robot's working space can be considered as a graph called  $G(V, E_o)$ , where  $V$  is the set of all vertices on the map (starting point, goal point and vertices of obstacles) and  $E_o$  is the set of all edges of obstacles.

All the edges, which can be derived from  $V$ , are examined. If the edge, which is being examined, does not cross any of the obstacles' edges, then add that edge to matrix  $E$  (where  $E$  stands for all the paths which can be followed by the robot without any collision with obstacles). Note that all boundary edges of the obstacles are also in  $E$ .

Hence, the obtained Visibility Graph is a graph  $G_{VG}(V, E, W)$  where  $W$  is the weight matrix containing length of edges in  $E$ . Note that edges crossing the obstacles all have the length of  $\infty$  in  $W$ .

Figure 2 illustrates a constructed Visibility Graph.

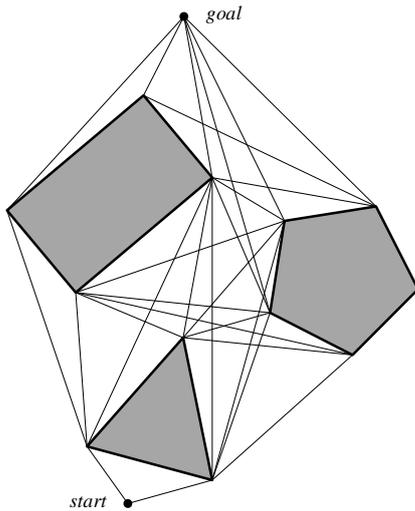


Figure 2. A Visibility Graph.

The theory of constructing a Visibility Graph above is based on the assumption of that robot is a point in the working space. Practically, the distance between the examined point on robot (usually the center) and boundary edges of obstacles needs to be calculated to ensure collision does not happen.

### 3. Dijkstra's optimal algorithm

The algorithm is based on the designation of the label for nodes in a weight graph (corresponding to vertices in Visibility Graph) whether temporary or fixed. The label of each node informs the upper limit of the shortest path from the starting node to that node. These labels will be changed after loops. At each loop, a label will be changed from temporary to fixed and a fixed node is one of the nodes in the derived shortest path [10]. The output of this algorithm is a path from starting node  $s$  and goal node  $g$  so that the sum of the length of the edges creating that path is minimal. The calculation of this algorithm is based on the following definitions:

$P$  is the set of nodes in the derived shortest path. Initiate  $P = [s]$ , which means at the beginning,  $P$  only contains starting node  $s$ , the others will be added to  $P$  later.

$L_v$  is the previous node of the being examined node  $v$  where  $v, L_v \in P$ . Initiate  $L = null$

$T_v$  is a boolean variable, which contains the state of node  $v$ , if  $T_v = true$  then  $v$  is in  $P$ , this ensures the repetition of  $v$  in the shortest

path does not happen. Initiate  $T_v = false \forall v \in V$ .

$D_v$  is the shortest distance from the starting node to the being examined node  $v$ . Initiate  $D_v = \infty$  and  $D_s = 0$ .

The process of the algorithm will operate through the following steps:

Step 1: Initiate the parameters as above.

Step 2: Pick up  $v \in V$  so that  $T_v = false$  and  $D_v$  min.

If  $D_v = \infty \forall v$  then concludes that there is no possible path from  $s$  to  $g$ .

If  $v \equiv g$ , which means the goal node  $g$  has been reached. The process is done.

Step 3: Assign  $T_v = true$ , add  $v$  to  $P$ .

Step 4: Examine all the nodes  $u$  so that  $u$  is adjacent to  $v$  and  $T_u = false$ :

If  $D_u > D_v + W_{uv}$  then  $D_u = D_v + W_{uv}$  (this is to update the shortest distance from  $s$  to  $u$  when  $v$  is added to  $P$ ), assign  $L_u = v$  and go to step 2.

The obtained result would be a series of points orderly making the shortest path

### 4. Control Algorithm for Robot to track the obtained path

After deriving the shortest path in the form of series of waypoints, the control problem is: control the robot to move from current position point  $G$  to a point  $I$  on the map. When this problem is solved, the path can be finished easily by repeating this problem on each pair of points of the path serially until the goal point is reached. In order to solve this problem, there are two steps to finish: robot modeling and design the control algorithm based on the modeling result.

#### Robot modeling:

The modeling step aims to find out the kinetics relationship between robot's position (coordinates, direction) and angular velocity of three wheels, which is used as basis for control algorithm. The parameters of robot and modeling process are shown in figure 3:

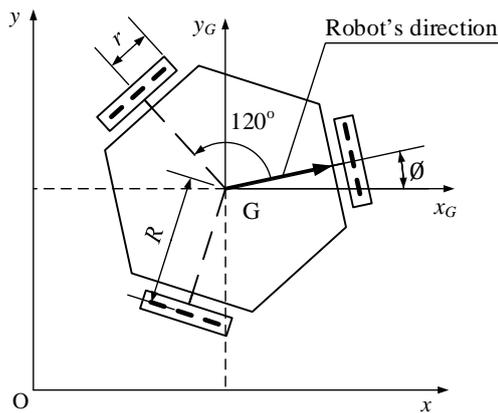


Figure 3. Parameters of the Modeling process.

Where:

$Oxy$  is the global coordinates,  $Ox_Gy_G$  is the coordinates at the robot's center.

$\phi$  is the angular deviation between the robot's direction and  $Ox$  axis.

$R$  is the robot's radius: the distance between the robot's center to each wheel.

$r$  is the radius of the Omni wheel.

Angular velocity of 3 Omni wheels:  $[\omega_1 \ \omega_2 \ \omega_3]^T$ .

Note: Assign that wheel number 1 is the wheel whose axis is coincident with the robot's direction, the next wheels are numbered 2 and 3 counter-clockwise, respectively.

After some calculations, the result of the modeling process is:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} \sin(\phi) & -\cos(\phi) & -R \\ \sin\left(\phi + \frac{2\pi}{3}\right) & -\cos\left(\phi + \frac{2\pi}{3}\right) & -R \\ \sin\left(\phi + \frac{4\pi}{3}\right) & -\cos\left(\phi + \frac{4\pi}{3}\right) & -R \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \quad (1)$$

Control algorithm:

With the modeling result above, the next step is to design a control algorithm to solve the control problem, which is performed in figure 4:

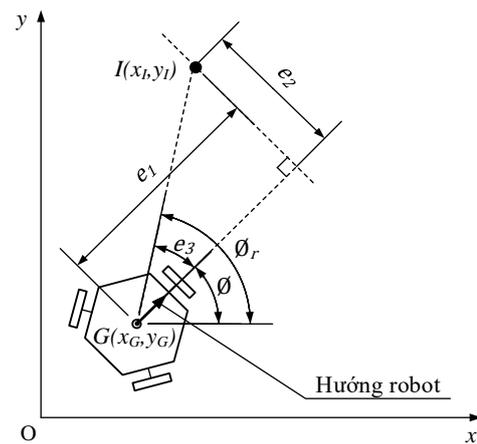


Figure 4. The model of control problem.

Where:

$I(x_I, y_I)$  is the goal point

$e_1, e_2, e_3$  are the errors between robot and goal point.

Calculation of the errors:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_I - x_G \\ y_I - y_G \\ \phi_r - \phi \end{bmatrix} \quad (2)$$

A derivative of the errors

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3)$$

With 2 control parameters are velocity  $v$  and angular velocity  $\omega$ , the objective of the needed controller is to eliminate the errors  $[e_1, e_2, e_3]^T = [0, 0, 0]^T$

The Lyapunov's stability method is used for the controller.

Theorem: Consider a system which is described by a state equation  $\dot{x} = f(x_1, \dots, x_n)$ . If exist a positive-definite  $V(x)$  with all state variables, so that its time derivative is a negative-definite function then that system is stable.

Choose a positive-definite Lyapunov function:

$$V = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + (1 - \cos e_3) \quad (4)$$

Derivative of  $V$ :

$$\dot{V} = -ve_1 - \omega \sin e_3 \quad (5)$$

Choose  $(v, \omega)$  so that  $\dot{V}$  is negative-definite:

$$\begin{cases} v = k_1 e_1 \\ \omega = k_2 \sin e_3 \end{cases}, \quad k_1, k_2 > 0 \quad (6)$$

Substitute  $v$  and  $\omega$  into (5), we have:

$$\dot{V} = -k_1 e_1^2 - k_2 \sin^2 e_3 \leq 0, \forall k_1, k_2 > 0 \quad (7)$$

Finally, calculate the angular velocity of Omni wheels concerning  $(v, \omega)$  for the control algorithm, the result is:

$$\begin{cases} \omega_1 = -\frac{R\omega}{r} \\ \omega_2 = \frac{\sqrt{3}}{2r}v - \frac{R\omega}{r} \\ \omega_3 = -\frac{\sqrt{3}}{2r}v - \frac{R\omega}{r} \end{cases} \quad (8)$$

With the obtained result above, the control problem is solved by the flowchart in figure 5:

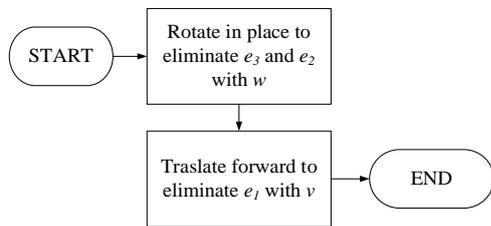


Fig. 5 Solution to control the problem.

**5. Experimental result**

Working space for the experiment is arranged as followings:

The camera is fixed above the working plane (top plane), with the working area is 2500 x 1400 (mm). four rectangular-shaped obstacles with unified color. Goal point, center point, and direction point are the centers of 3 distinct color cards, with area of each of them is 50 x 50 (mm). The structure of robot's control system is shown in figure 6 with a master microcontroller, whose objective is to receive the series of waypoints of the shortest path from computer, calculate the needed angular velocity of three wheels and send that result to the corresponding slave microcontroller; three slave microcontrollers, they have the same objective which is to receive the angular velocity and control the motor to reach that velocity perfectly.

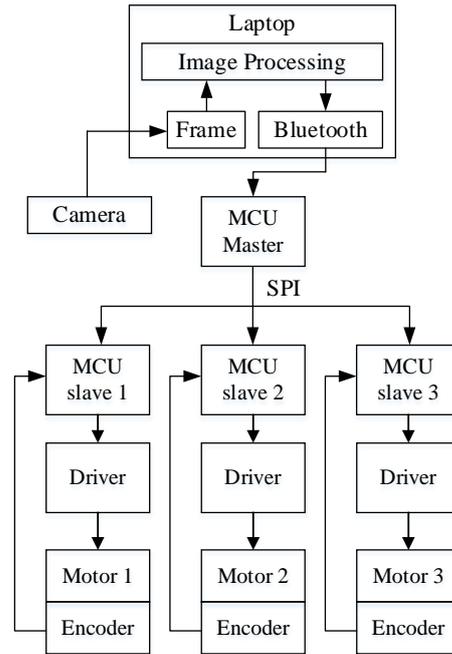


Fig. 6 Structure of the control system.

The result of the construction of Visibility Graph and the shortest path is performed in figure 7:

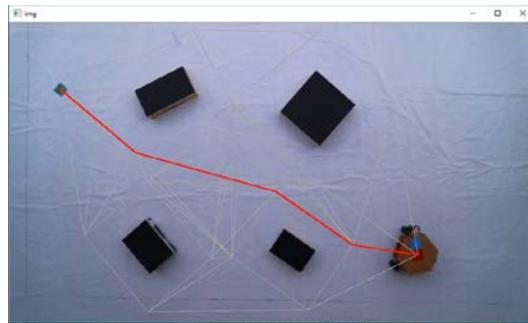


Figure 7. Result of Navigation problem.

The result of following the path in simulation, in practice and the comparison between those results are shown in figure 8, 9, 10

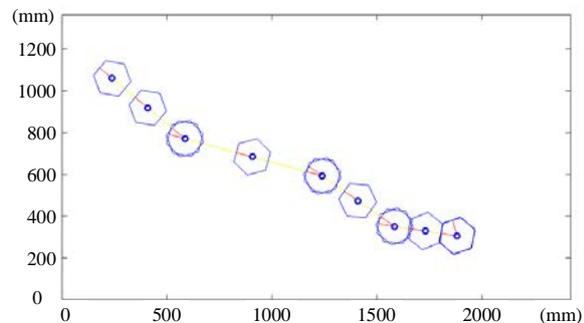


Figure 8. Result of path tracking in simulation.

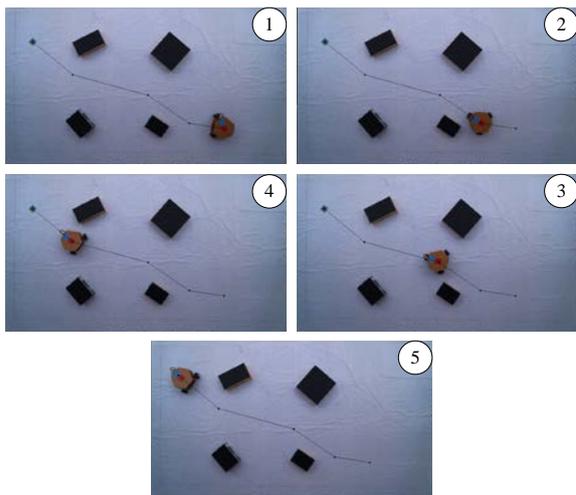


Figure 9. Result of path tracking in practice.

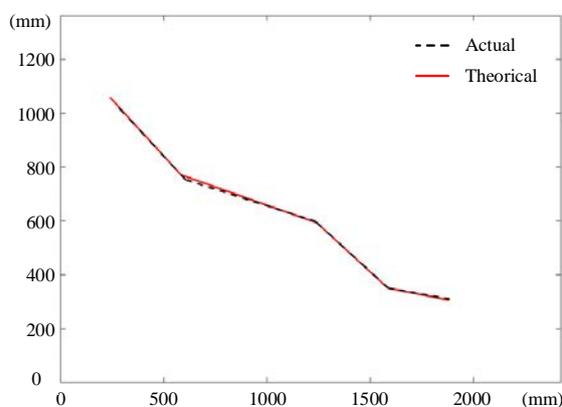


Figure 10. Comparison between simulation and practice.

On the other hand, the robot's position processed from the camera (while the robot is operating) is used to calculate the errors  $e_1$ ,  $e_2$ ,  $e_3$ . Then this result is compared with theoretical errors in simulation to evaluate the quality of the controller. The results of this comparison are shown in figure 11 and 12:

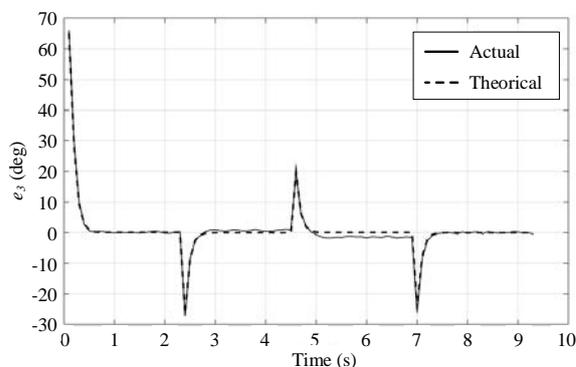


Figure 11. Angular error  $e_3$ .

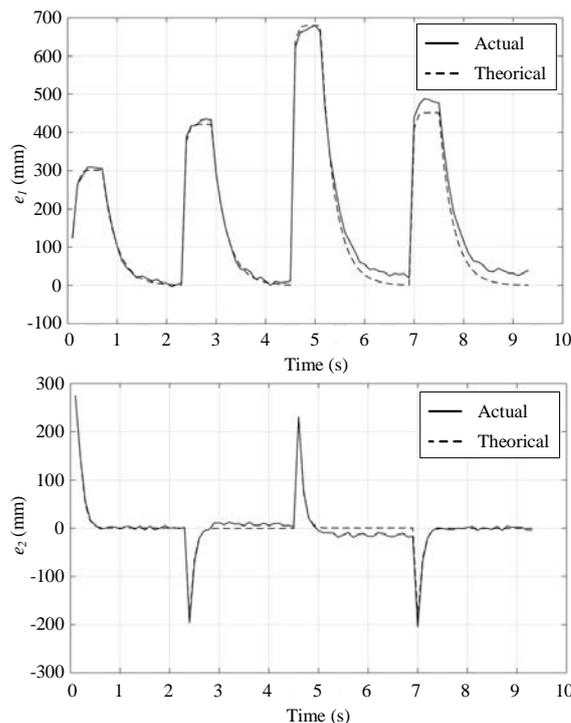


Figure 12. Errors  $e_1$ ,  $e_2$ .

According to the results above, it is seen that the combination of the camera and control algorithm works, which helps the robot follow the path well with a small deviation.

### 6. Conclusion

This paper presents a path planning approach, which helps robots operate in narrow space such as in warehouses or the home. The path is optimized in terms of distance by the application of image processing. In addition, the robot can follow the path thanks to a designed controller. Hence, the path is not only optimized but also becomes more flexible than the ones in traditional methods. Also, the further development for the subject can be making use of additional sensors attached to the robot's body combined with more advanced algorithms so that robots can be able to adapt to changes of the surrounding environment while working □

### Acknowledgment

This research is funded by Ho Chi Minh City University of Technology –VNU-HCM, under grant number To-CK-2018-01.

### References

[1] N. H. Chowdhury, D. Khushi and Md. M. Rashid, *Algorithm for Line Follower Robots to Follow*

- Critical Paths with Minimum Number of Sensors*, International Journal of Computer (IJC), 24 (2017), pp. 13-22.
- [1] J. H. Su, C. S. Lee, H. H. Huang, S. H. Chuang, and C. Y. Lin, *An intelligent line following robot project for introductory robot courses*, *World Transactions on Engineering and Technology Education*, 8 (2010), pp. 455-461.
- [2] H. N. Joshi et al, *An Image-Based Path Planning and Motion Planning for Autonomous Robot*, *International Journal of Computer Science and Information Technologies*, 5 (2014), pp. 4844-4847.
- [3] H. Borse, A. Dumbare, R. Gaikwad and N. Lende, *Mobile Robot for Object Detection Using Image Processing*, *Global Journal of Computer Science and Technology*, 12 (2012), Issue 11.
- [4] J. Kim, Y. Do, *Moving obstacle avoidance of a mobile robot using a single camera*, *International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012)*, pp. 411-416.
- [5] Huu Danh Lam, Tran Duc Hieu Le, Tan Tung Phan, and Tan Tien Nguyen, *Smooth tracking controller for AGV through junction using CMU camera*, *The 7th Vietnam Conference on Mechatronics (VCM-2014)*, pp. 597-601.
- [6] G. H. Lee and S. Jung, *Line Tracking Control of a Two-Wheeled Mobile Robot Using Visual Feedback*, *International Journal of Advanced Robotic Systems*, 10, 177 (2013).
- [7] S. H. Tang, W. Khaksar, N. B. Ismail and M. K. A. Ariffin, *A Review on Robot Motion Planning Approaches*, *Pertanika Journal of Science and Technology*, 20 (2012), pp. 15-29.
- [8] Thi Nhu Nguyet Tran, Duc Lung Vu, and Van Hoai Tran, *Applying Visibility-Graph to Shortest Path Problem for Robots*, *Can Tho University Journal of Science, Part A*, 32 (2014), pp. 42-56.
- [9] Quoc Thang Thai, *Tìm Hiểu và Xây Dựng Phần Mềm Hỗ Trợ Bài Toán Tìm Đường Đi Ngắn Nhất Tránh Vật Cản Cho Xe Tự Hành Trong Không Gian 2D*, Master Thesis in Information Technology, Lac Hong University, Dong Nai (2013).

**Ngày nhận bài: 17/12/2019**

**Ngày chuyển phản biện: 20/12/2019**

**Ngày hoàn thành sửa bài: 10/1/2020**

**Ngày chấp nhận đăng: 16/1/2020**