International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 4 Issue 1, December 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Q-Learnıng Based Real Tıme Path Plannıng for Mobıle Robots

Halil Cetin[1], Akif Durdu[2], M. Fatih Aslan[3], M. Mustafa Kelek[4]

[1, 2, 3, 4]Department of Electrical and Electronic Engineering,
[1]Konya Technical University, Konya, Turkey
[2]Ohio State University, Columbus, USA
[3]Karamanoglu Mehmetbey University, Karaman, Turkey
[4]Kocatepe University, Afyonkarahisar, Turkey

ABSTRACT

Decision making and movement control are used for mobile robots to perform the given tasks. This study presents a real-time application in which the robotic system estimates the shortest way from robot's current location to target point via Q-learning algorithm and makes decision to go the target point on the estimated path by using movement control. Q-Learning algorithm is known as a Reinforcement Learning (RL) algorithm. In this study, it is used as a core algorithm for estimation of the path that is optimum way for mobile robot in an environment. The environment is viewed by a camera. This study includes three phases. Firstly, the map and the locations of all objects including a mobile robot, obstacles and target point in the environment are determined by using image processing. Secondly, Q-Learning algorithm is applied for the problem of the estimation of the shortest way from the current location of the robot to target point. Finally, a mobile robot with three omni wheels was developed. Experiments were carried out using this robot. Two different experiments are performed in experimental environment. The results obtained are shared at the end of the paper.

KEYWORDS: *Q-learning, path planning, mobile robot*

***How to cite this paper**:* Halil Cetin | Akif Durdu | M. Fatih Aslan | M. Mustafa Kelek "Q-Learnıng Based Real Tıme Path Plannıng for Mobıle Robots" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 | Issue-1, December 2019, pp.631-636, URL: www.ijtsrd.com/papers/ijtsrd29625.pdf

IJTSRD29625

INRODUCTION

Thanks to new technologies and rapidly advancing science and technology, a new world is established in nowadays. The images transferred to the computer from the outside are evaluated in the computer by using the help of digital image processing. Object recognition and feature extraction for different images can be done by image processing techniques which include a series of image processing operations such as morphological operations and conversion of the digital input images. The operations begin with the capture of the images and continue with the use of different techniques for the purpose [1].

Image processing applications have been increased depending on the development of computer systems [2]. The increase of applicability of the image processing allows for the spread of mobile robot applications. Because only robots that perform fixed tasks have limited application area. This has led to the need for more useful mobile robots [3]. Mobile robots have the ability to move autonomously by gathering information from the environment [4]. Sensors are used to make full use of mobile robots. Sensors provide relevant information about the environment in which the robot is located. In most applications, a camera is used as the sensor. Thus, with image processing techniques, the robot can perform autonomous tasks, and decide against the events surrounding it. Thus, with image processing techniques, the robot can perform autonomous tasks and the robot can decide against the events around it.

Robots that can move autonomously and decide can start to be more desirable in terms of the services they provide. The decision mechanisms of such autonomous robots depend on estimation algorithms and are quite complex systems [5]. Today, most of the applications on autonomous mobile robots involve Machine Learning [6] methods. Because behaviour of robot in unknown environment cannot be modelled mathematically. Machine Learning methods are used to analyse the available data. Accordingly, appropriate decisions are made regarding the new data. There are three types of Machine Learning. Q-Learning [7], the RL [8] method, was used in this study. RL learns by using trial and error method in dynamic environment [9]. The situation that arises as a result of the trial is characterized by using rewards and punishment. Learning is provided according to reward and punishment situations. In this way, the robot can decide autonomously.

In autonomous tasks, motion planning is an important area for mobile robots. Planning the movement of a mobile robot requires both the planning of the path and the planning of its orientation [10]. Q-Learning is a method used in path planning applications. Using the Q-Learning algorithm, the most convenient way can be found by trial and error method.

In this study, the problem is the estimation of the shortest way from robot's current location to target point via Q-learning algorithm. The robot and obstacles are positioned randomly in the environment. Robot, obstacles and target in colored image frames received from the environment is determined by using an image processing algorithm in the computer. The algorithm is based on color of the objects. After the object detection, Q-learning algorithm, which is a reinforcement algorithm, is applied for the estimation the shortest path between the robot and target. Two experiments are performed in the real-time system. In the following sections, methods used in this study and design of general system are explained. The results are given in the fifth section. Finally, discussions for obtained results and future works are mentioned in the last section.

RELATED WORKS

Q-learning algorithm has been widely used in path planning for autonomous mobile robots. Li et al. [11] have used adaptive path planning based on layered Q-learning method. A novel prior formula search method (PQA) is used for the static obstacle avoiding behaviour. In another study [12], it is assumed that the current state of the robot and the distance information to the target is available. Accordingly, an alternative method for Q-Learning has been proposed. Klidbary et al. [13] have used four different optimization algorithms with Q-Learning. Thus, a comparative study has been carried out. In a different study [14], results of preliminary studies related to how neural networks can be used in path planning in square grids have been presented. According to these results, neural Q-Learning algorithm was used successfully to reach target in small maps. Unlike the classical Q-learning algorithm, Goswami et al. [15] and Das et al. [16] have updated the Q-table. Thus, a study has been carried out which reduces both the area and time complexity. In Vien et al.'s study [17], the path planning approach is used with the Ant-Q algorithm. Finally, in another study, Jaradat et al. [18] made a new definition for the state space and in this study the number of states is limited. This reduces the size of Q-table and thus increases the speed of the navigation algorithm.

METHODS

In this study, two algorithms are used. The first algorithm is color-based image processing algorithm for object detection and tracking. The second algorithm is Q-learning algorithm for estimation of the shortest way between the robot and the target point.

A. Color-Based Image Processing Algorithm

Each image is digitalized as a matrix form in the computer. The matrix has the same size with the image and these matrix values vary according to the color of the pixels that make up the image. All color in an image can be obtained from Red, Blue, Green, so this color space is expressed as RGB. However, sorting in OpenCV library is BGR. The HSV (Hue, Saturation and Value) color space is the most suitable color space for color-based image segmentation used in object tracking. Therefore, RGB color space is converted to HSV color space during image processing [19].

B. Q-Learning Algorithm

Reinforcement learning is a learning method based on trial and error. It is usually used for learning with unexpected results depend on performance of a mobile robot. This learning method can also be used to implement robots' online learning. Online learning and adaptation are commonly desired properties for robot's learning algorithms during collecting enough quantity of data from examples for the required experience. In a complex environment, online learning is required to interact with the dynamic environment. In this study, firstly, the map was created by determining the location and object, then the shortest path was found on the map, and finally the movement of the robot on the determined route was achieved. Q-learning is one of the most preferred RL algorithms in robotic applications because of its simplicity and availability to online learning [20].

In Q-Learning, all of possible states and all of the movements in the given state of an agent is known as deterministic. In other words, for a given A agent, there are n number of possible cases including $S_0, S_1, S_2 \dots S_n$ and m number of possible movements including $a_0, a_1, a_2 \dots a_m$ in all the cases. In a specific state-movement couple, the reward of the couple for an agent is named as instant reward. For example, $r(S_i, a_j)$ specifies the reward when A agent has a movement a_j in case of S_i. An agent does the procedure of selection process for the next state and location from the currents by following a certain rule. This rule attempts to maximize total reward on the next state and all next possible movements. For example, an agent which is on state of S_i is waiting for choosing the next best state. Q-value of state of S_i depends on a_j movement is given in (1) [21].

$$Q(S_i, a_j) = r(S_i, a_j) + \gamma \, max_{a'} Q\left(\delta(S_i, a_j), a'\right) \qquad (1)$$

Where γ is learning coefficient and it is usually selected as the value in the range of 0 and 1. When it equals to 0, it can be said that any learning does not exist or Q-values only equal to the reward-value of $r(S_i, a_j)$ from (1).

$\delta(S_i, a_j)$ specifies the next state due to a_j movement selection in the state S_i. Get S_k as the next state. If we get S_k as the next state, then $Q\left(\delta(S_i, a_j), a'\right) = Q(S_k, a')$. As a result, the selection of (a') which maximizes $Q(S_i, a_j)$ is an important problem. Q-Learning determines Q-values for all possible (a') values for a state S_k. During the determination of the best next state, it is hardly needed to read data from memory to get Q-values for all possible movements in a specified state. So, more time is spent to select the next state [21].

The stopping criterion of the Q-learning algorithm is that the values in the Q-state table remain below a certain amount of change. Decrease in the amount of change means that the table becomes stable and path planning can be done based on this table. It has been observed that a state table (matrix) can give correct results without becoming stable [22]. Therefore, the number of iterations is considered as the stopping criterion in this study. Furthermore, a disadvantage of the algorithm is that too much time is required for the stable table. All of the process which starts from the initial point and ends with reaching to the target point is taken as an iteration. Learning algorithm is stopped after a specified number of iterations.

C. Path Planning

Path planning is to determine the appropriate path between two points which are a specific point and the destination point [20]. Among these paths, it is preferable to choose the shortest or longest path according to the solution of the problem, to select the path that has the least maneuverability or least complexity, and to select the paths with the most turns and complex paths.

The Q-state table (matrix) obtained by the application of the Q-learning algorithm is like a conditional map used to make decisioning and path planning for robot. The R-reward penalty table will be used when running the algorithm and will be basically a reference table. In other words, that is map. In the path determination phase, the robot algorithm, which has finished all operations, now uses the Q-state table to determine the decision mechanism on the map. The algorithm that determines the values on the Q table ensures that the lowest value is assigned to the farthest point where the target can be reached no matter where the robot is. After a certain iteration, all the values are determined according to the point where the reward is in the status table and become constant.

The robot takes a list of all situations that start at what point, do not have a penalty value around, that is, they are free to move. Then, from this list, the state having the largest Q value is selected. If two or more cases have the same value, they choose any of them. Because the distance to the target/prize is the same as the situation with the same value. In this way, the Q-learning algorithm can be used for path determination on a free surface and can also be used in a multi-handed environment such as a labyrinth. It also selects the shortest path through the labyrinths that form for the path to the destination, depending on the environment.

GENERAL APPLICATION DESIGN

This work has two phases; In the first stage, the design and integration of the robot which is used as hardware is done and in the second stage, the most efficient algorithms suitable for real time application as software are investigated.

In the hardware phase, the robot used is designed as an omni wheel and is made as shown in Fig.1. It is shown in Fig. 2 that thanks to the omni wheel, the robot can move directly to the desired position-forward, backward, right, left, and diagonal without turning.

To increase the direction sensitivity, the number of directions can be increased as much as desired with software. Thanks to the omni wheel system used, the robot is able to move faster and respond more quickly to surrounding events. Furthermore, the Bluetooth module was used in the robot in order to communicate with the computer. In addition, in this application, three motors (175 Rpm 10 kg/cm Torque), two motor driver cards (L298) and a microcontroller (Arduino Mega 2560) which evaluates information provided by computer and controls the movements of our robot are used.

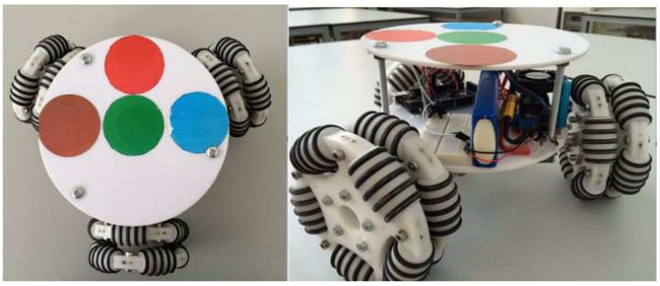


Figure 1: Top (a) and side (b) view of omni wheel robots which use in the application

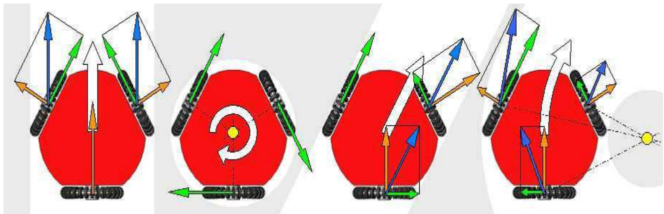


Figure 2: A robot with omni wheel motions analysis

During the software phase, the image taken from the camera was evaluated by image processing techniques in the computer environment and object identification was provided. Thus, the target, the robot and the obstacles are distinguished and so, their positions are determined. Depending on the detected position data, the robot can reach the next destination point using the shortest path. According to the flow diagram shown in Fig. 3, the image taken from the environment is processed in the computer, and then, the obtained data is transmitted via a Bluetooth module to robot.

A. Sensing and Communication System

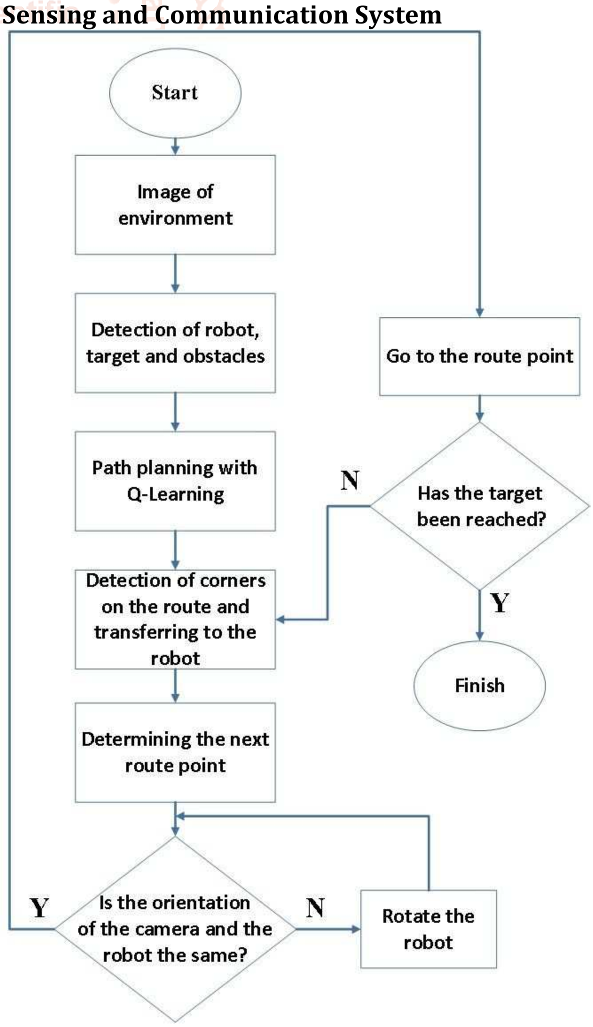


Figure 3: Flow diagram

Different color markers are designed to make it easier to locate and track the robot on the environment [20]. Color makers was made up by gluing four different colored circular pieces on a white mica, as shown in Fig. 1. The location and the orientation of the robot was detected by ceiling-mounted camera by recognition of colored signs, as shown in Fig. 4. After making the definitions, in accordance with the flow diagram shown in Figure 3, the necessary commands are sent with Bluetooth communication to move the robot to the most appropriate coordinates for the target. The DC motors connected to the omni wheels used to provide robot motion are controlled by Pulse Width Modulation (PWM) signals generated by the Arduino microcontroller.

It is difficult for the robot to perform a real-time task. To overcome this problem, The OpenCV Library with C++ language are used to avoid delays in simultaneous image transfer. Four different color markers have been used to determine robot coordinates and to compare the axes of the robot and the camera. Three of the perceived four different colors are used to determine the orientation of the robot and to compare the axis of the robot with the axes of the camera, and the final color is used to determine the position of the robot. The obtained algorithm depending on the perceived colors, constitute the input data which is necessary for comparison. The screenshot of the application as the result of the study is shown in Fig. 4. According to the light intensity of the environment, the tones of the colors that the camera perceives change. For this reason, it is necessary to define the colors again in each different environment.

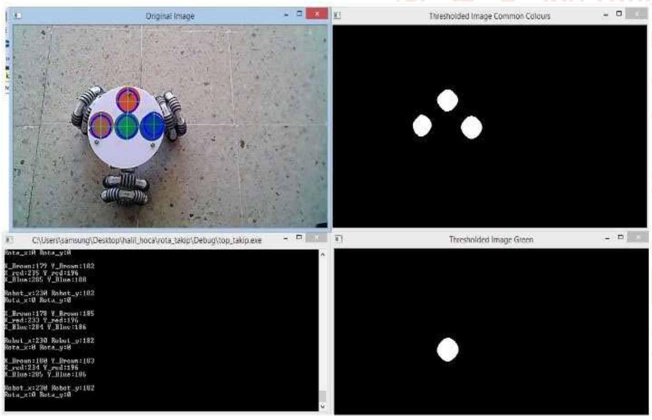
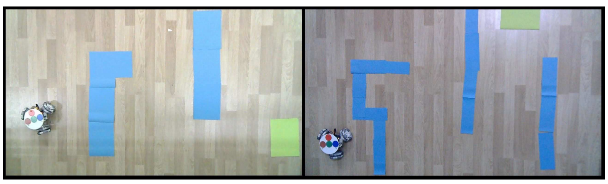


Figure 4: The image of the computer program that is used for the color recognition method

TRIAL AND SIMULATION

Among the studies made, the stability of the algorithm was checked by focusing on two environments with different light intensities. These environments are shown in Fig. 5. Though the environments are similar to each other in both applications, the obstacle thicknesses, the light intensity of the environment and the routes the robot can pass are different. In order to reduce the operation times, image reduction is applied on the images taken instantaneously.



(a) (b)

Figure 5: The environments where the first and second experiments are carried out (a) First experiment (b) Second experiment

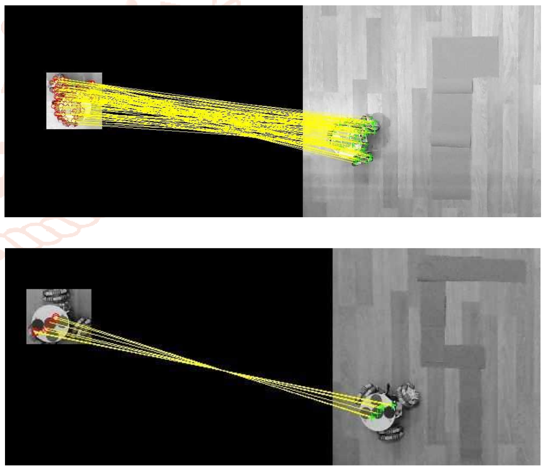
The operation flow of the algorithm used in this study can be explained in Algorithm 1 as follows;

Algorithm 1

1: Detecting of the robot on environment, calculating the vertical-horizontal lengths and midpoint.
2: Reduction of obtained instant images for use in the Q-Learning algorithm.
3: Finding obstacles and targets by classifying colors.
4: Enlarging of obstacles' and targets' virtual size in such a way that the dimensions of the robot can pass.
5: Path planning with Q-Learning algorithm.
6: Detection of corner points on the path and sending to robot for parallel programming.

A. Detection the Robot

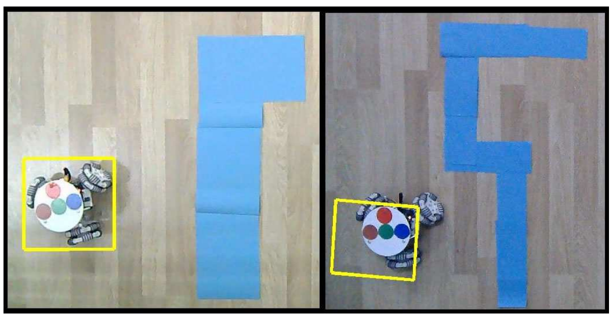
Using the colors found on the robot, the robot found in the frame received has been detected as real time. The dimensions and midpoint of the robot were found to determine the position of the robot and the ways it could enter. The output of the first application is shown in Fig. 6(a), and the output of the second application is shown in Fig. 6(b). Feature extraction was applied on the environment image through an image belonging to robot. As a result of the feature extraction, pixels matching with the robot on the media image are determined using the lines in Fig. 6. Fig. 7(a) shows the results of the robot detection process for the first application and Fig. 7(b) shows the results of the robot detection process for the second application. The boundaries of the robot are determined by surrounding the detected robot with a square (see Fig. 7).



(a)

(b)

Figure 6: Detection of the robot according to the results of feature extraction (a) for first trial (b) for second trial



(a) (b)

Figure 7: The robots detected in the first (a) and second (b) applications

B. Detection of Obstacles and Target

After detecting the robot on the media image, the location and dimensions of the target and obstacles in the environment are also determined. In this way, the necessary input data for the application has been reached. In this stage, the classification of colors is used to determine the obstacles and the target, also the images are reduced for Q-Learning algorithm and thus, the operation time is shortened. It has been seen that reducing images and shortening the processing time do not affect the accuracy of the results. On the contrary, the Q-Learning algorithm, which runs on a large map obtained as a result of the values obtained from the images with high resolution, causes errors. It has been determined that the Q agent moving at random is able to enter an endless loop in three-sided enclosed regions. The reason of this, the basic algorithm that computers use to generate random numbers. In Fig. 8, in the first application, and in Fig. 9, in the second application, outputs for determining the robot, obstacles and the target have been shown.

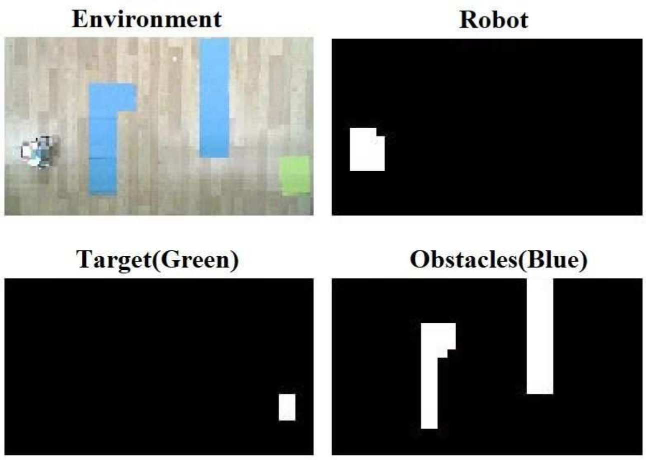


Figure 8: Detection of robots, obstacles and target for first trial

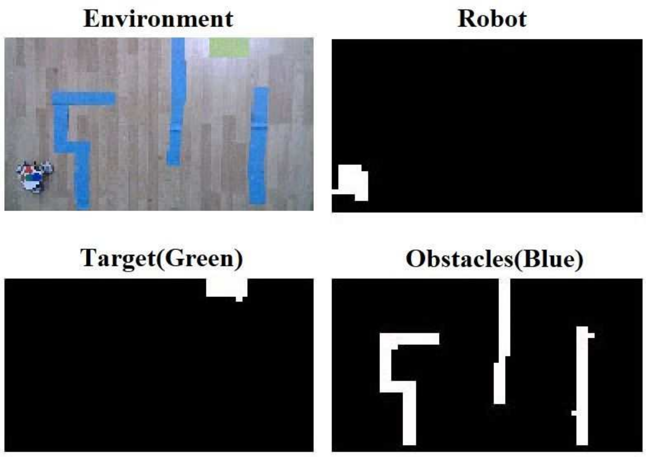
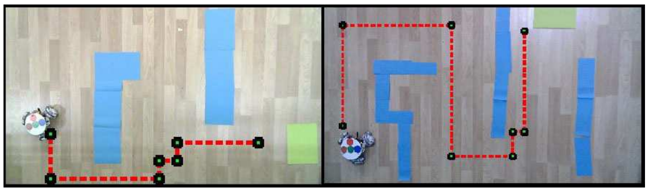


Figure 9: Detection of robots, obstacles and target for second trial

In Fig. 8 and Fig. 9, the first image gives the current image of the environment. After this image is taken, the process steps explained above are applied. Thus, the robot shown in the second picture, the target in the third picture, and the obstacles in the fourth picture are detected.

C. Path Planning with Q-learning Algorithm

The Q-Learning algorithm is runs on a map generated from the acquired images, so that the state matrix is obtained. Through this matrix, path planning is done (see Fig. 10). Route information obtained during path planning is sent to the application used for robot motion.



(a) (b)

Figure 10: Route obtained in the first (a) and second (b) application.

The values obtained in the state matrix are numerical values that progressively increase towards the target. By choosing the next highest value among these values, the robot will go to the destination using the shortest path. Thus, by starting from the point where the robot was identified, the route was created and the path planning was carried out. Only the corner coordinates of the obtained route are sent to the robot motion control application. In other words, for the robot, there is a movement on the route from the corner to the corner. Furthermore, depending on the speed of the robot, it is avoided to ignore the next points given on the route. In this application, the position of the robot is determined by using the colors on the robot. Thus, the direction movement required for the robot to move to the next point is controlled.

CONCLUSION

As a result of these applications, it has been seen that a mobile robot can move to a determined target by moving autonomously. All the data obtained show how important the assistive and preparative algorithms are. The Q-Learning layer prepared with these algorithms worked faster and more stable [22]. The state matrix obtained from the Q-learning layer is used in the path planning. The route information obtained by the path planning process was sent to the motion control layer of the robot and autonomous motion was started and the behaviours observed were recorded.

In the study, object detection methods have been tried and the most suitable methods for the purpose have been chosen. Since the descriptive colors on the robot are used in robot control application, feature extraction is used in robot detection. Also, color based recognition has been made in the determination of target and obstacle. The obtained data are reduced by the resizing method for use in the Q-Learning algorithm, so that accurate and stable results are obtained. In addition, image enhancement and filtering operations are applied on the images taken instantaneously.

After all these preparations, it was observed that path planning process lasted 5.79 seconds on average in 5 trials for the first application. In an application where no preparative auxiliary algorithms were used, it was observed that this user exceeded the average 40 second limit [22]. As a result of 5 trials made for the second application, the mean path planning time was determined as 23.99 sec. Although the environment created for this application is similar to the first application environment, it has a more complex structure than the first in terms of the size of the snapshot taken. In the first application, 1280x720 resolution images were used, while in the second application this value was increased to 1920x1080. It has been observed that this size difference causes time delay in the path planning phase.

Operating the path planning and robot control layers in different applications causes some disadvantages. Applications that are more suited to the principle of simultaneous working, basically working in a single layer, can be developed. In the future studies, it is planned to design a fully synchronized algorithm structure by combining the path planning and motion control layers. Thus, it is aimed to achieve a dynamic structure. Instead of planning a single path on the environment, renewal can be done depending on various criteria. Due to the dynamic structure of displaced obstacles, re-detection and re-planning of the pathway can make the scientific literature work more efficiently. In addition, different learning and image processing methods can be used in real world applications where there is no possibility of receiving images from above.

ACKNOWLEDGEMENTS
Authors are thankful to RAC-LAB for providing the trial version of their commercial software for this study.

REFERENCES

[1] SAMTAŞ, G. and GÜLESIN, M. Sayısal Görüntü İşleme ve Farklı Alanlardaki Uygulamaları. Electronic Journal of Vocational Colleges. 2012, Vol. 2 No.1. pp. 85-97.

[2] SHOJAEIPOUR, S., HARIS, S. M. and KHAIRIR, M. I. Vision-based mobile robot navigation using image processing and cell decomposition. International Visual Informatics Conference, Springer. 2009, pp.90-96.

[3] UZER, M. S., YILMAZ, N. and BAYRAK, M. A Real-Time Tracking Application of Different Coloured Objects with a Vision Based Mobile Robot. Journal Of The Faculty Of Engineering and Architecture Of Gazi University. 2010, Vol. 25, pp. 759-766.

[4] JIN, T. and LEE, J. Mobile robot navigation by image classification using a neural network. IFAC Proceedings Volumes. 2004, Vol. 37, pp. 203-208.

[5] SCHMITT, T., HANEK, R., BEETZ, M., BUCK, S. and RADIG, B. Cooperative probabilistic state estimation for vision-based autonomous mobile robots. IEEE Transactions on robotics and automation. 2002, Vol. 18, pp. 670-684.

[6] MICHALSKI, R. S., CARBONELL, J. G. and MITCHELL, T. M. Machine learning: An artificial intelligence approach. Springer Science & Business Media. 2013.

[7] LIN, J.-L., HWANG, K.-S., JIANG, W.-C. and CHEN, Y.-J. Gait balance and acceleration of a biped robot based on Q-learning. IEEE Access. 2016, Vol. 4, pp. 2439-2449.

[8] KOBER, J., BAGNELL, J. A. and PETERS, J. Reinforcement learning in robotics: A survey. The International Journal of Robotics Research. 2013, Vol. 32, pp. 1238-1274.

[9] BUSONIU, L., BABUSKA, R. and DE SCHUTTER, B. A comprehensive survey of multiagent reinforcement learning. IEEE Trans. Systems, Man, and Cybernetics. 2008, Part C, Vol. 38, pp.156-172.

[10] SHILLER, Z. Off-line and on-line trajectory planning. Motion and Operation Planning of Robotic Systems, Springer. 2015, pp. 29-62.

[11] LI, Y., LI, C. and ZHANG, Z. Q-learning based method of adaptive path planning for mobile robot. Information Acquisition, IEEE International Conference. 2006, pp.983-987.

[12] KONAR, A., CHAKRABORTY, I. G., SINGH, S. J., JAIN, L. C. and NAGAR, A. K. A deterministic improved Q-learning for path planning of a mobile robot. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2013, Vol. 43, pp. 1141-1153.

[13] KLIDBARY, S. H., SHOURAKI, S. B. and KOURABBASLOU, S. S. Path planning of modular robots on various terrains using Q-learning versus optimization algorithms. Intelligent Service Robotics. 2017, Vol. 10, pp. 121-136.

[14] PANOV, A. I., YAKOVLEV, K. S. and SUVOROV, R. Grid Path Planning with Deep Reinforcement Learning: Preliminary Results. Procedia Computer Science. 2018, Vol. 123, pp. 347-353.

[15] GOSWAMI, I., DAS, P. K., KONAR, A. & JANARTHANAN, R. Extended Q-learning algorithm for path-planning of a mobile robot. Asia-Pacific Conference on Simulated Evolution and Learning. 2010, Springer, Vol. 6457, pp.379-383.

[16] DAS, P. K., MANDHATA, S., BEHERA, H. and PATRO, S. An improved Q-learning algorithm for path-planning of a mobile robot. International Journal of Computer Applications. 2012, Vol. 51, No. 9, pp.40-46.

[17] VIEN, N. A., VIET, N. H., LEE, S. and CHUNG, T. Obstacle avoidance path planning for mobile robot based on ant-q reinforcement learning algorithm. International Symposium on Neural Networks, Springer. 2007, pp.704-713.

[18] JARADAT, M. A. K., AL-ROUSAN, M. and QUADAN, L. Reinforcement based mobile robot navigation in dynamic environment. Robotics and Computer-Integrated Manufacturing. 2011, Vol. 27, No. 1, pp. 135-149.

[19] BAŞLAR, I. Tüm Gökyüzü Kamerasından Alınan Görüntüler Kullanılarak Bulutluluk Ölçümü Yapılması. Türkiye Bilimsel Ve Teknolojik Araştırma Kurumu. 2012.

[20] LI, C., ZHANG, J. and LI, Y. Application of artificial neural network based on q-learning for mobile robot path planning. Information Acquisition, IEEE International Conference. 2006, pp.978-982.

[21] GOSWAMI, I., DAS, P. K., KONAR, A. and JANARTHANAN, R. Conditional Q-learning algorithm for path-planning of a mobile robot. IEEE International Conference on Industrial Electronics, Control & Robotics (IECR). 2010, pp. 23-27.

[22] CETIN, H. and DURDU, A. Path planning of mobile robots with Q-learning. IEEE Signal Processing and Communications Applications Conference (SIU). 2014, pp. 2162-2165.