Journal of Soft Computing and Artificial Intelligence

JSCAI, 1(1): pp:18-27 (2020), www.dergipark.org.tr/en/pub/jscai
Received: May 23 2020 Accepted: June 15 2020

JSCAI
ISSN: 2717-8226

Global Vision Based Path Planning for AVGs Using A* Algorithm

[1]Mahmut Dirik , [2]A. Fatih Kocamaz

[1]Department of Computer Engineering, Sırnak University, TURKEY
[2]Department of Computer Engineering, Inonu University, TURKEY

Corresponding author: Mahmut Dirik (mhmd.dirik@gmail.com)

ABSTRACT One of the most studied problems in robotics is robot path planning. Many strategies have been invented. Image processing and machine vision technology also have been utilized in this regard. Studies are still underway to improve path planning methods. This paper proposes an implementing visual servoing-based technique using the A* algorithm to achieve efficient searching capabilities of path planning in complicated maps with a combination of LabVIEW and MATLAB software. The proposed algorithm is divided into three parts. Firstly, the environment model or robot motion environment is conducted. In this stage, the visual information extracted from a single ceiled camera. Secondly, the position and orientation of the objects (robot, obstacles etc.) under the visibility of the camera are generated from visual information. Thirdly, the A* algorithm is used as a path planning method. This algorithm is not guaranteed the generated path to be safe and desirable with obstacle-free. To solve this problem image processing techniques are utilized. This gives an effective improvement and high performance to A* in a complex environment and gives a safe path as a comparison to the traditional version of A*. The experimental results, considering the optimal path lengths and execution time, show that the proposed design is more effective and faster to generate the shortest path.

KEYWORDS: A* Algorithm, AVGs, Image Processing, Path Planning, Visual Servoing

1. INTRODUCTION

Path planning is a heavily studied because of its application in the field of industrial and daily manipulator of mobile robot research. The fundamental purpose is to find an optimal, safe, and collision-free path between the starting to the target point [1]. During the last two decades, a great deal of research focuses on the path planning problem [2-8]. To perform a task with the mobile robot finding a feasible solution in critical applications in real-life, one needs to solve path planning and path tracking problems efficiently [9, 10]. The environment type (static or dynamic) and path-planning algorithms are two important factors in solving the path planning problem. The path planning algorithms can be classified into two categories: global (off-line) or local (on-line) algorithms [11-13]. Global path planning methods required the environment model (robot map) to be static and completely known. There are many algorithms designed for global path planning such as A* [14,15] which is an extension of the Dijkstra algorithm[16,17], Genetic algorithm (GA) [18-21], Probabilistic Road Map (PRM) [24], Rapidly Exploring Random Tree (RRT) [25], Bidirectional-RRT (BRRT) [26, 27], Artificial Potential Fields (APF) [28, 29], Fuzzy Type 1 and Fuzzy Type 2 path planning algorithm [30-32]. Traditionally, different sensing techniques enable the robot to detect obstacles such as infrared detectors, laser scanner, ultrasonic sensors [33-35]. These sensors may cause systematic and unsystematic errors when the robot is moving. Systematic errors are often caused by the encoder, sensor, and physical design of the robot parts. However, non-systematic errors are often caused by external causes such as slipping, hitting, and falling.

On the other hand, vision sensors provide low-cost motion control and effective in decreasing errors, as mentioned. They are also useful robotic sensors that allow for non-contact measurement of the environment. The information provided by vision sensors in a feedback loop known as visual servoing. It is classified into a position based, image-based, and hybrid based visual servoing (VS) system. The proposed method using VS is designed to extract the position and orientation information of interest objects. The main advantage of the visual servoing [36-38] is that it requires fewer sensor data, suitable to control multiple robots, internal and external sensors on robots generally are not needed, in terms of scalability; it provides more operating area by increasing imagining devices and so on. For these regards, various approaches have been used. These are background subtraction based, feature-based, gradient-based, statistical model-based, template-based, and optical flow object detection methods [39, 40].

The aim of this paper is how to combine visual servoing and A* based on an effective mobile robot path planning. Many experiments have been carried out to test the validity of the proposed technique. The results have shown that the system is effective and fast.

This paper is organized as follows. Problem formulation, image analysis, and environment model are described in section 2. Path planning and shortest path algorithm are discussed in section 3. The experimental results are presented in section 4. Finally, the paper conclusion is given in section 5.

2. MATERIAL AND METHODS

2.1 Environment Model

In the overall design, the proposed method uses only one non-contact sensor (overhead camera). Information to generate the desired robot path is extracted from image frames that acquire from this sensor. The robot poses estimated at each location from image frames using feature extraction and template-matching based methods. The work environment is configured as shown in Figure 1.

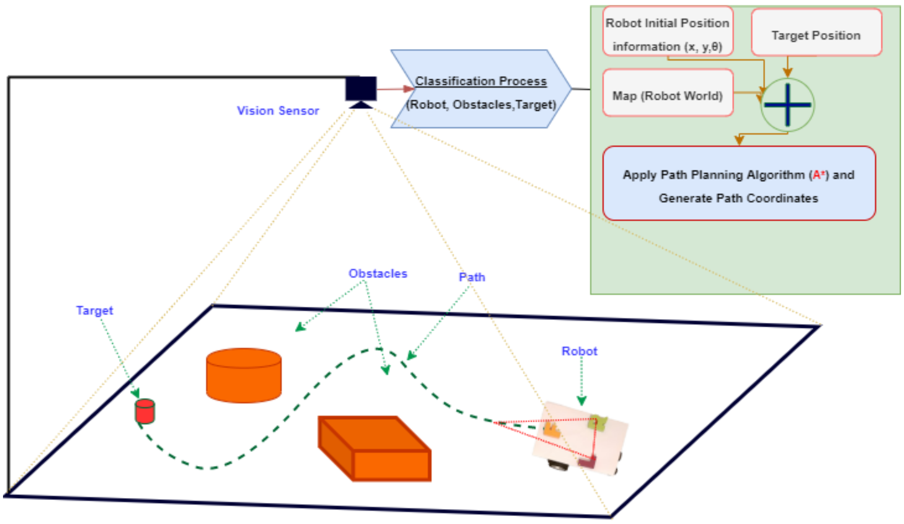


Figure 1. Overall system configuration block diagram

The base infrastructure hardware component has consisted of a mobile robot motion environment, a mobile robot, an overhead camera, and a host computer system. The implemented software component includes both LabVIEW and Matlab image processing tools and control modules. First, the classification process of the position and orientation of the robot, target, and obstacles are handled. Secondly, the initial parameters of the path planning algorithms are determined. In this stage, A* path planning algorithms have been considered. Three inputs parameters are required for the A* algorithm. These are the robot map, robot starting, and target point parameters. Several mathematical equations are required to obtain these input parameters.

The robot marks parameters and target parameters such as color and shape are initialized. These marks properties are template images as shown in Figure 2.

Figure 2. LabVIEW VI implementation of template matching system

Robot localization and position information were obtained using these templates (R, L, and F). Each template is uniquely identified by an onboard geometric pattern. To obtain an accurate representation of the boundaries of the obstacles, the proper selection of a color threshold value is essential for image analysis. The process of obtaining the robot map by determining the robot's initial position information, targets, and obstacles are shown on the image in figure 3. Detailed information about these processes is given in [41].

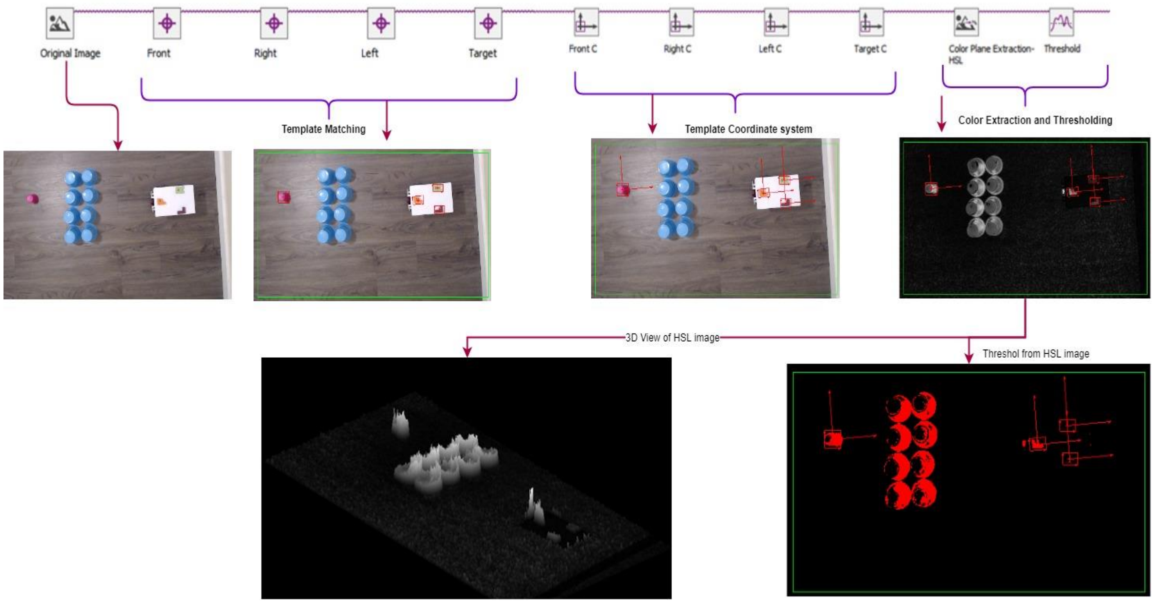


Figure 3. Image conversion process

After determining the initial position of the mobile robot and the target point, other calculations are performed. The required parameters are characterized in Figure 4.

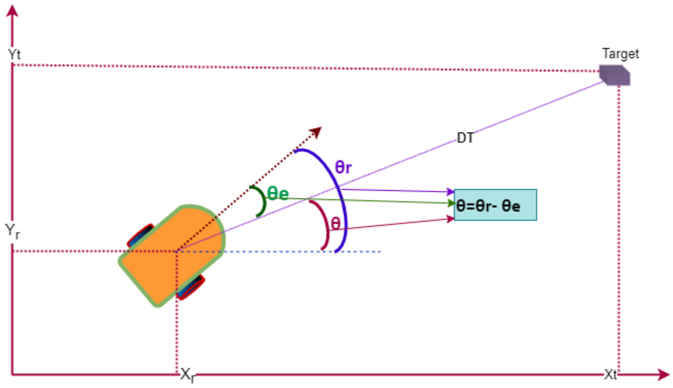


Figure 4. Path planning algorithm virtual inputs (angle to the goal (θ=θr- θe), distance from the target (DT))

The initial position information and target coordinates of the mobile robot are graphically represented in this figure. The equations used to obtain the input parameters are calculated as follows.

$$
\begin{aligned}
e_x &= X_t - X_r = DT * \cos(\theta \mathrm{r}) \\
e_y &= Y_t - Y_r = DT * \sin(\theta \mathrm{r})
\end{aligned}
\tag{1}
$$

The positioning error computations are calculated as in (1). Where, DT corresponds to the current distance between the mobile robot and target, which is expressed in Equation (2).

$$DT = \sqrt{(e_x)^2 + (e_y)^2} \tag{2}$$

The robot current angle (θr) according to the target is computed as in Equation (3).

$$\theta \mathrm{r} = tan^{-1} \frac{e_y}{e_x} \tag{3}$$

The error of the angle is given in Equation (4).

$$\theta \mathrm{e} = \theta \mathrm{r} - \theta \tag{4}$$

These are the planning algorithm's initial parameters.

3. A* ALGORITHM

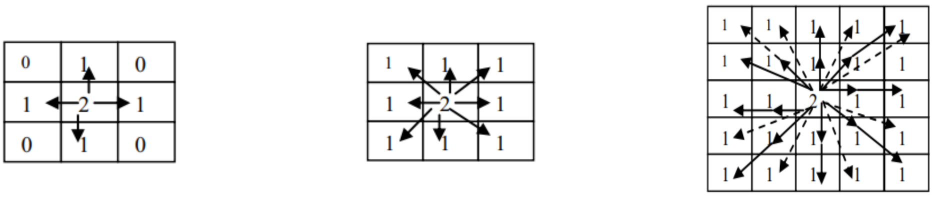
A star algorithm proposed by Haret et al.[17]. The A* algorithm is a practical search algorithm for path-finding and graph traversals in the real-world problem, which is a class of intelligent search algorithms in the Uniform Cost Research (UCS) philosophy developed based on Dijkstra [42] algorithm that it can find the shortest path. The key of the A* algorithm is to establish the evaluation function given in (5).

$$f(n) = g(n) + h(n) \tag{5}$$

where f(n) represents the expected cost from source to goal via node n, g(n) represents the exact cost of the path from the starting point to any vertex n, and h(n) represents the heuristic estimated cost from vertex n to the goal. The specific domain information in the problem is the heuristic function, which is an estimated distance of the node n to the goal. The Euclidean distance (ED) between the node n and the goal is usually taken as the value of $h(n)$that is an estimated cost of reaching the goal. When the value of $g(n)$ is constant, the value of $f(n)$is mainly affected by the value of $h(n)$ which is the cost value from the successor node to the destination node corresponds to the Manhattan distance (heuristic). The algorithm is optimal as a graph search using both an open and a closed set of nodes while ensuring acceptability and consistency. When using the A * algorithm, it is necessary to model the problem as a standard graphical search algorithm.

In our experiments, the converted binary images (pixel graph) are used as a searching node. All regions of the acquired image pixels are searched one by one to find the shortest path, and the unobstructed path from the source to the destination is determined. All black pixels are defined as obstacles; all white pixels are defined as a free node. The total cost constitutes the evaluation function's cost calculated between the free nodes.

The higher resolution of the map, the better results, but undesirable because it increases computational time in real-time applications [43]. Each pixel of the reduced resolution map is taken as a corner, and the connection paths between the pixels are taken as the edge. Robot motion is regarded in three possible matrix connections as shown in figure 5. These are rectilinear, rectilinear, diagonal, and many moves. All possible movements are indicated by 1, and impossible movements by 0. For increasing the rotational flexibility of the robot, the Cardinality of numbers in the matrix can be increased, but the addition of these can result in more calculation costs. In our experiments, the "Rectilinear and Diagonal" matrix was used.



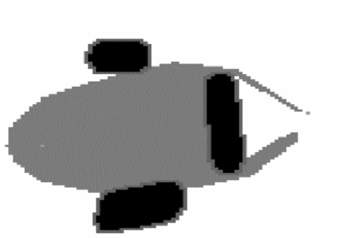
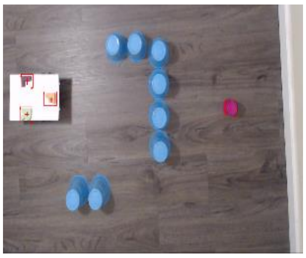
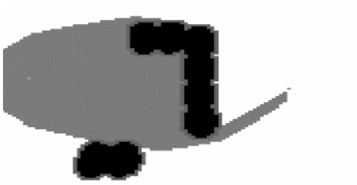
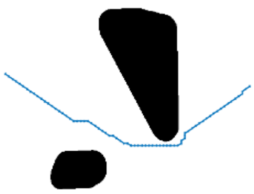
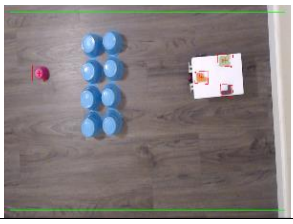
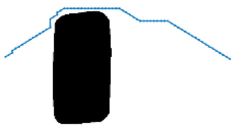
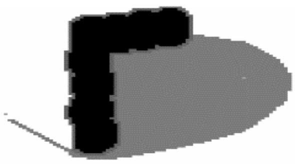
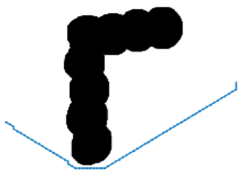
a) b) c)

Figure 5. Connection matrices (a) Rectilinear, b) Rectilinear and Diagonal, c) Many Moves)

4. SIMULATION RESULTS

In order to verify the effectiveness of the proposed algorithm, LabVIEW and Matlab software are implemented practically in real-time. We tested the algorithm for the various test cases. In all test cases it is observed that the algorithm is able to generate a soft and suitable solution. It is more convincing to propose the algorithm to be used in real-world mobile robot tracking experiments. In the experiments, first of all the ceiling camera configuration set up and visual information for a planner are obtained. The main parameters include mobile robot initial position information, target position, and obstacles position. The experimental study was carried out in two stages. In the first stage, the environment map, which is normally applied to image processing and sharp transitions are eliminated. In the second stage, convex hull method was applied to the maps of the environment in order to eliminate situations such as the local minimum and to minimize a collision. Experiments were carried out in structured environments as shown in figure 6.

As a binary image which is a 2D matrix of elements that can only hold two values where the white pixels (values 1) correspond to the free space and the black pixels (values 0) correspond to an obstacle area, it suffices to represent a fill grid with only two color levels, since the robot can only move within the free space. The obstacle-free path and other operations obtained by using this algorithm are shown on eight experimental results (see Figure 6).

Exp.	**Initial color Image**	**Expansion and path searching**	**Path**
1			
2A			
2B			
3			
4A			

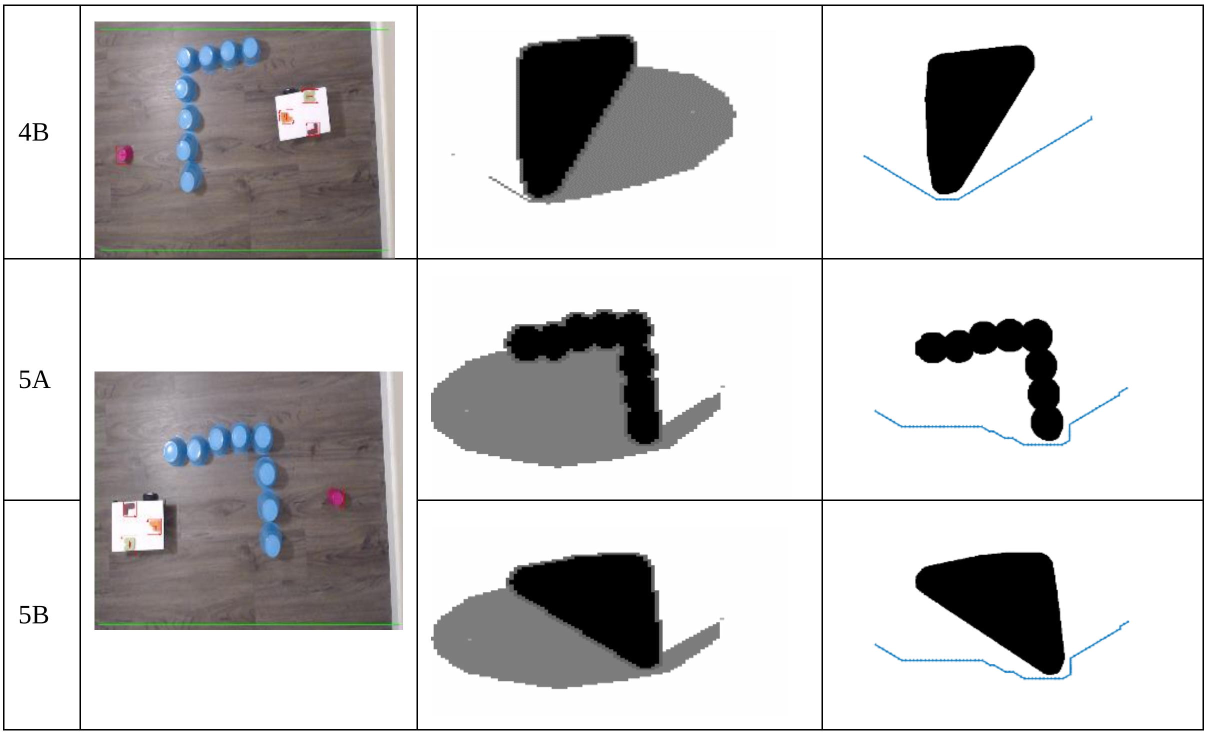


Figure 6. Experimental Sample Results using A* algorithm (A in (2, 4, 5) are the normal environments; B in (2, 4, 5) are convex hull applied maps.

To check the result of the experiments, the system is considered algorithm processing time and path length. The optimum path between the start and goal obtained by using A* path searching algorithm are marked as shown in figure 6. To obtain a feasible path that the robot can follow, it is aimed to reduce the possibility of collision during the movement of the robot by using dilatation and convex hull operation.

For this reason, the obstacle boundaries have been extended to the radius (half dimension) of the actual robot and the robot's safe operation without collision has been realized. In this case, the environment map has been changed so that the free space near the obstacles can be regarded as a disability area at distances below the radius. In the test cases, it is observed that the algorithm was able to find a feasible path solution to use by any robotic controller to move the robot physically. The execution time and path lengths obtained from the experiments are summarized in Table 1.

Table 1. Path Lengths (PL-px) and Execution Time (ET-sc) obtained in different configuration spaces

	Experiments							
	1	**2A**	**2B**	**3**	**4A**	**4B**	**5A**	**5B**
PL-(px)	757,00	800,56	793,29	762,02	756,91	736,88	785,20	770,20
ET-(sc)	5,00	3,91	4,51	8,05	3,98	3,42	5,19	3,28

As seen from the table, the convex hull method applied maps were completed in a shorter time and resulted in less path cost. All experiments have been successfully completed. Both execution and path length were evaluated together. It should be pointed out that these planning periods are the times taken for path planning on the acquired image of the real environment. The path cost is more critical parameter. Because of robot to be operated in a real environment will spend time and energy according to the obtained path costs. In this case, it is stated that the path cost parameter is more critical in terms of enabling the robot to operate efficiently.

5. CONCLUSIONS

This paper performs real-time path planning algorithm of an indoor wheeled mobile robot using a single ceiling camera is implemented. The contribution here is the development of the A * algorithm in complex environments for mobile robot path planning. The predictive function of an improved A * algorithm is used as an intuitive function to improve search efficiency and smoothness of the path using image processing operation. To optimize and secure the path created with A *, image processing widening application (dilatation, convex body) implemented. The generated path is located as near as possible to the obstacle (s), by a distance determined by the robot radius. As a result, short travel distances for the robot were made possible in a short time by consuming less power along the way. This project has shown that a robot can be directed to move in an indoor environment without hitting any obstacles, even if it does not have any internal sensors. The system also takes into account the robot's information, such as the turning radius, depth, and width. The overall system is experimentally verified under the same conditions by using LabVIEW and Matlab software together. The results are convenient and reliable for the mobile robot to follow the path created.

REFERENCES

[1] X. Dai, S. Long, Z. Zhang, and D. Gong, "Mobile robot path planning based on ant colony algorithm with a∗ heuristic method," *Front. Neurorobot.*, 2019, doi: 10.3389/fnbot.2019.00015.

[2] A. Cherubini, F. Chaumette, and G. Oriolo, "A position-based visual servoing scheme for following paths with nonholonomic mobile robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2008, pp. 1648–1654, doi: 10.1109/IROS.2008.4650679.

[3] E. A. Elsheikh, M. A. El-Bardini, and M. A. Fkirin, "Practical Design of a Path Following for a Non-holonomic Mobile Robot Based on a Decentralized Fuzzy Logic Controller and Multiple Cameras," *Arab. J. Sci. Eng.*, vol. 41, no. 8, pp. 3215–3229, Aug. 2016, doi: 10.1007/s13369-016-2147-x.

[4] T. T. Mac, C. Copot, T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Rob. Auton. Syst.*, vol. 86, pp. 13–28, 2016, doi: 10.1016/j.robot.2016.08.001.

[5] J.-Y. Jhang, C.-J. Lin, C.-T. Lin, and K.-Y. Young, "Navigation Control of Mobile Robots Using an Interval Type-2 Fuzzy Controller Based on Dynamic-group Particle Swarm Optimization," *Int. J. Control. Autom. Syst.*, vol. 16, no. 5, pp. 2446–2457, Oct. 2018, doi: 10.1007/s12555-017-0156-5.

[6] R. Kala, A. Shukla, R. Tiwari, S. Rungta, and R. R. Janghel, "Mobile robot navigation control in moving obstacle environment using genetic algorithm, artificial neural networks and A* algorithm," in *2009 WRI World Congress on Computer Science and Information Engineering, CSIE 2009*, 2009, doi: 10.1109/CSIE.2009.854.

[7] R. Lagisetty, N. K. Philip, R. Padhi, and M. S. Bhat, "Object detection and obstacle avoidance for mobile robot using stereo camera," in *Proceedings of the IEEE International Conference on Control Applications*, 2013, doi: 10.1109/CCA.2013.6662816.

[8] K. Zheng, D. F. Glas, T. Kanda, H. Ishiguro, and N. Hagita, "Supervisory control of multiple social robots for navigation," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2013, doi: 10.1109/HRI.2013.6483497.

[9] J. Han and Y. Seo, "Mobile robot path planning with surrounding point set and path improvement," *Appl. Soft Comput.*, vol. 57, pp. 35–47, Aug. 2017, doi: 10.1016/j.asoc.2017.03.035.

[10] B. K. Patle, D. R. K. Parhi, A. Jagadeesh, and S. K. Kashyap, "Application of probability to enhance the

performance of fuzzy based mobile robot navigation," *Appl. Soft Comput.*, vol. 75, pp. 265–283, Feb. 2019, doi: 10.1016/j.asoc.2018.11.026.

[11] R. Kala, A. Shukla, and R. Tiwari, "Robotic path planning in static environment using hierarchical multi-neuron heuristic search and probability based fitness," *Neurocomputing*, vol. 74, no. 14–15, pp. 2314–2335, Jul. 2011, doi: 10.1016/j.neucom.2011.03.006.

[12] M. Dirik, O. Castillo, and A. Kocamaz, "Visual-Servoing Based Global Path Planning Using Interval Type-2 Fuzzy Logic Control," *Axioms 2019, Vol. 8, Page 58*, vol. 8, no. 2, p. 58, May 2019, doi: 10.3390/AXIOMS8020058.

[13] G. Antonelli, S. Chiaverini, and G. Fusco, "A Fuzzy-Logic-Based Approach for Mobile Robot Path Tracking," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 2, pp. 211–221, Apr. 2007, doi: 10.1109/TFUZZ.2006.879998.

[14] F. Duchoň *et al.*, "Path Planning with Modified a Star Algorithm for a Mobile Robot," *Procedia Eng.*, vol. 96, pp. 59–69, 2014, doi: 10.1016/j.proeng.2014.12.098.

[15] G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, *Wheeled Mobile Robotics, From Fundamentals Towards Autonomous Systems*. Butterworth-Heinemann, © 2017 Elsevier Inc., 2017.

[16] S. A. Fadzli, S. I. Abdulkadir, M. Makhtar, and A. A. Jamal, "Robotic Indoor Path Planning using Dijkstra ' s Algorithm with Multi-Layer Dictionaries," pp. 1–4, 2015.

[17] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968, doi: 10.1109/TSSC.1968.300136.

[18] S. Salmanpour, H. Monfared, and H. Omranpour, "Solving robot path planning problem by using a new elitist multi-objective IWD algorithm based on coefficient of variation," *Soft Comput.*, vol. 21, no. 11, pp. 3063–3079, 2017, doi: 10.1007/s00500-015-1991-z.

[19] P. Sudhakara, V. Ganapathy, and K. Sundaran, "Genetic algorithm based optimization technique for route planning of wheeled mobile robot," in *Proceedings of the 4th IEEE International Conference on Advances in Electrical and Electronics, Information, Communication and Bio-Informatics, AEEICB 2018*, 2018, doi: 10.1109/AEEICB.2018.8480937.

[20] A. Elshamli, H. A. Abdullah, and S. Areibi, "Genetic algorithm for dynamic path planning," in *Canadian Conference on Electrical and Computer Engineering*, 2004.

[21] AL-Taharwa, "A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment," *J. Comput. Sci.*, vol. 4, no. 4, pp. 341–344, Apr. 2008, doi: 10.3844/jcssp.2008.341.344.

[22] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning," *Procedia Comput. Sci.*, vol. 127, pp. 180–189, 2018, doi: 10.1016/j.procs.2018.01.113.

[23] Jianping Tu and S. X. Yang, "Genetic algorithm based path planning for a mobile robot," in *International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2003, vol. 1, pp. 1221–1226, doi: 10.1109/ROBOT.2003.1241759.

[24] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, 1996, doi: 10.1109/70.508439.

[25] J. Bruce and M. Veloso, "Real-time randomized path planning for robot navigation," in *IEEE/RSJ International Conference on Intelligent Robots and System*, 2002, vol. 3, pp. 2383–2388, doi: 10.1109/IRDS.2002.1041624.

[26] E. Dönmez, A. F. Kocamaz, and M. Dirik, "Bi-RRT path extraction and curve fitting smooth with visual based configuration space mapping," in *IDAP 2017 - International Artificial Intelligence and Data Processing Symposium*, 2017, doi: 10.1109/IDAP.2017.8090214.

[27] R. Sadeghian, S. Shahin, and M. T. Masouleh, "An experimental study on vision based controlling of a spherical rolling robot," in *Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*, Dec. 2017, pp. 23–27, doi: 10.1109/ICSPIS.2017.8311583.

[28] T. Weerakoon, K. Ishii, and A. A. F. Nassiraei, "An Artificial Potential Field Based Mobile Robot Navigation Method To Prevent From Deadlock," *J. Artif. Intell. Soft Comput. Res.*, vol. 5, no. 3, pp. 189–203, Jul. 2015, doi: 10.1515/jaiscr-2015-0028.

[29] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992, doi: 10.1109/70.163777.

[30] J.-Y. Jhang, C.-J. Lin, C.-T. Lin, and K.-Y. Young, "Navigation Control of Mobile Robots Using an Interval Type-2 Fuzzy Controller Based on Dynamic-group Particle Swarm Optimization," *Int. J. Control. Autom. Syst.*, vol. 16, no. 5, pp. 2446–2457, Oct. 2018, doi: 10.1007/s12555-017-0156-5.

[31] T. W. Liao, "A procedure for the generation of interval type-2 membership functions from data," *Appl. Soft Comput.*, vol. 52, pp. 925–936, Mar. 2017, doi: 10.1016/j.asoc.2016.09.034.

[32] A. Pandey, R. K. Sonkar, K. K. Pandey, and D. R. Parhi, "Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller," *2014 IEEE 8th Int. Conf. Intell. Syst. Control*, pp. 39–

41, 2014, doi: 10.1109/ISCO.2014.7103914.
[33] A. Pandey, “Multiple Mobile Robots Navigation and Obstacle Avoidance Using Minimum Rule Based ANFIS Network Controller in the Cluttered Environment,” *Int. J. Adv. Robot. Autom.*, vol. 1, no. 1, pp. 1–11, 2016, doi: 10.15226/2473-3032/1/1/00102.
[34] K. Srinivasan and J. Gu, “Multiple Sensor Fusion in Mobile Robot Localization,” in *Canadian Conference on Electrical and Computer Engineering*, 2007, pp. 1207–1210, doi: 10.1109/CCECE.2007.308.
[35] A. Shitsukane, W. Cheruiyot, C. Otieno, and M. Mvurya, “Fuzzy Logic Sensor Fusion for Obstacle Avoidance Mobile Robot,” *IST-Africa Week Conf.*, no. May, pp. 1–8, 2018.
[36] S. R. Bista, P. R. Giordano, and F. Chaumette, “Combining line segments and points for appearance-based indoor navigation by image based visual servoing,” in *IEEE International Conference on Intelligent Robots and Systems*, 2017, doi: 10.1109/IROS.2017.8206131.
[37] F. Bonin-Font, A. Ortiz, and G. Oliver, “Visual Navigation for Mobile Robots: A Survey,” *J. Intell. Robot. Syst.*, vol. 53, no. 3, pp. 263–296, Nov. 2008, doi: 10.1007/s10846-008-9235-4.
[38] E. Dönmez, A. F. Kocamaz, and M. Dirik, “A Vision-Based Real-Time Mobile Robot Controller Design Based on Gaussian Function for Indoor Environment,” *Arab. J. Sci. Eng.*, vol. 43, no. 12, pp. 7127–7142, Dec. 2018, doi: 10.1007/s13369-017-2917-0.
[39] Y. Yoon, G. N. DeSouza, and A. C. Kak, “Real-time tracking and pose estimation for industrial objects using geometric features,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2003, doi: 10.1109/robot.2003.1242127.
[40] A. Elgammal, D. Harwood, and L. Davis, “Non-parametric model for background subtraction,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2000, doi: 10.1007/3-540-45053-x_48.
[41] M. Dirik, “Development of vision-based mobile robot control and path planning algorithms in obstacled environments,” Inonu University, 2020.
[42] L. M. S. Bento, D. R. Boccardo, R. C. S. Machado, F. K. Miyazawa, V. G. Pereira de Sá, and J. L. Szwarcfiter, “Dijkstra graphs,” *Discret. Appl. Math.*, vol. 261, pp. 52–62, May 2019, doi: 10.1016/j.dam.2017.07.033.
[43] R. Kala, A. Shukla, and R. Tiwari, “Fusion of probabilistic A* algorithm and fuzzy inference system for robotic path planning,” *Artif. Intell. Rev.*, vol. 33, no. 4, pp. 307–327, Apr. 2010, doi: 10.1007/s10462-010-9157-y.