Veri Bilim Derg, 4(3), 89-95, 2021

VERİ BİLİMİ DERGİSİ
www.dergipark.gov.tr/veri

Autonomous self-parking robot using A-star in VREP

Umut ÇELEBİ[1]*, Muhammad Umer KHAN[1], Mert Anıl DEMİRHAN[2], Mohammad Hassan GOL MOHAMMADZADEH[1]

[1]*Atılım University, Faculty of Engineering, Department of Mechatronics Engineering, Ankara*
[2] *Çankaya University, Faculty of Engineering, Department of Electrical and Electronics Engineering, Ankara*

Makale Bilgisi

Başvuru:
07/07/2021
Kabul:
15/10/2021

Abstract

Autonomous self-driving cars are usually equipped with key features, such as lane detection, speed controls, emergency braking, and self-parking. Despite all the technological advancements, these features cannot be considered fully reliable and often require human driver control. This study focuses upon the problem of autonomous self-parking assuming that the maps are already available. To prove the effectiveness of the approach, the pioneer 3dx mobile robot is used as an implementation platform in VREP. Using the A-star path planning algorithm, the path is defined from the current position of the mobile robot to the closest available slot. The concept of path smoothing has also been introduced in the existing algorithm to avoid sharp turns. Due to the proposed approach, the mobile robot tends to perform parallel, backward, and forward parking depending upon the space constraint of the parking slot. Through results in the simulated environment, it is witnessed that the mobile robot effectively approaches the available slot and performs all three types of parking without any collision.

Keywords: *self-parking, A-star, path planning, mobile robot, VREP*

VREP uygulamasında A-star kullanılarak otonom park eden robot

Özet

Otonom araçlarda genellikle şerit takibi, hız kontrolü, acil frenleme ve otonom park gibi bazı önemli özellikler bulunur. Teknolojideki ilerlemeye rağmen bu özellikler tamamıyla güvenilir değildir ve bazen sürücü kontrolüne ihtiyaç duyarlar. Bu çalışmada, park alanı haritaları önceden bilinen mekanlardaki otonom park üzerine odaklanıldı. Çalışma VREP uygulamasında, pioneer 3dx mobil robot kullanılarak test edildi. A-star yol planlama algoritması kullanılarak anlık pozisyondan, uygun en yakın park alanına yol oluşturuldu. Yol düzleştirme, keskin dönüşleri yumuşatmak için sisteme tanıtıldı. Yöntemi test etmek için oluşturulan çevrede mobil robot farklı boyutlarda olan park alanlarına paralel, düz ve geri olmak üzere farklı şekilde park etti. Simulasyondan çıkarılan sonuçlara göre robotun etkili bir biçimde uygun park alanlarına her hangi bir çarpışma yaşamadan park ettiğini gözlemledik.

Anahtar Kelimeler: *otonom park, A-star, yol planlama, mobil robot, VREP*

***** İletişim e-posta: uc.umutcelebi@gmail.com

** Part of this work was presented orally at the IV. International Conference on Data Science and Applications 2021.

1 Introduction

In today's modern world, autonomous vehicles play an important role in many areas. Since technological developments have been accelerated, their effects on people's daily lives have increased equally. The large aspect of using autonomous vehicles can be seen from factories that use automated guided vehicles (AGVs) which are used to lower the labor cost and increase efficiency, to the usage of self-driving cars on a daily life basis.

A car can be considered autonomous if it can understand the environment with provided information by its sensors and without any human involvement in the process of navigation and sensing of the environment [1]. Such cars can be named driverless [2], self-driving [3], or robotic [4] vehicles. More recently, many tech giants have also shown keen interest in autonomous vehicles, such as Google and Apple. The former started the self-driving car project named Chauffeur [5-6].

Even though the fully autonomous drive of a car is a rarer concept, self-parking systems are being integrated by manufacturers in most of their latest models. BMW has introduced a self-parking assistance to their 730i model that can park the car in the presence of the driver [7]. Parking guidance systems have been developed by Mercedes, Audi, and Toyota to assist the drivers during parking processes [8-10].

Detecting a free parking space is one of the most important steps of the self-parking process. By using a vision-based method, a car can search for a free slot in the whole parking area. In this study, using image processing techniques, certain positions and the numbers of the free slots are determined. Vision-based parking space detection has many advantages; easy to install, low cost, the detector can be adjusted easily, and obtained information from an image is abundant. However, the success of the process is highly dependent on the position of the vision sensor [11], which can also lead to the failure of the whole process. Moreover, failure is also possible due to the physical disruptions in the environment. Despite its disadvantages, searching for a parking slot simultaneously can be beneficial for the driver. Although, simultaneous searching in some cases can be beneficial, for some specific cases, parking by using path planning algorithm can be more useful. Such an approach can be used easily in places like malls where a certain map of the parking area is already available. Parking in a place, that has a predefined map, by using a path planning algorithm could be advantageous. Time-saving and less energy consumption can be considered as the most notable advantages.

Even if the self-driving cars' most useful feature can be considered as saving time and energy, recent researches are debating over a more important feature as to avoid serious collisions where human reflexes could not be able to provide safe maneuvers. According to WHO (World Health Organization), 90% of collisions are caused by human error [12]. Tesla's accident data of 2021 proves that the autopilot and active safety features have helped to reduce the number of road accidents [13]. Since this technology is making people's lives safer, similar technology can be used for a safer parking process.

In this paper, an autonomous parking method is provided to assist the human driver in the parking space. In particular, the contributions of this paper can be summarized as follows:

- A-star path planning algorithm with three different heuristic functions is introduced to perform self-parking in a place that has a predefined map. This algorithm is commonly renowned for finding optimal paths, but not necessarily smooth ones. To shorten the path, by turning sharp turns into smooth turns, path smoothing is also applied.
- Simulations of the parking system are performed via VREP (renamed as CoppeliaSim) for fourteen parking slots in which there are eight slots to be parked backward or forward and six slots to be parked only parallel. Evaluation of energy consumptions, by considering time and distance are also presented..

2 A-star algorithm and path smoothing

The A-star algorithm is a heuristic search algorithm and is also considered as one of the most effective ways to determine the shortest path in the static/dynamic environment. Its first use was registered for path optimization in game development. It has its wide applications from game development to robotics e.g., robot path planning, urban intelligent transportation graph theory, automatic control, among many others [14].

In this algorithm, heuristic search functions are utilized to generate a path. There are three more commonly used heuristic functions: Manhattan, diagonal, and Euclidean distances. The reason for introducing these three heuristic functions is their good performance on the grid. For the number of allowed movements, different heuristic functions are suggested to be used.

- On a square grid that allows 4 directions of movement, use Manhattan distance.
- On a square grid that allows 8 directions of movement, use Diagonal distance.
- On a square grid that allows any direction of movement, it is suggested to use Euclidean distance.

If A-star is finding paths on the grid but the allowed movements are not on the grid, other representations of the map can also be considered.

The basic distance along the x-axis and y-axis are measured using the current node and the goal node as d_x and d_y, respectively.

$$d_x = |node_x - goal_x| \tag{1}$$

$$d_y = |node_y - goal_y| \tag{2}$$

The Manhattan distance heuristic function is calculated as in Equation (1):

$$f_{node} = D \times (d_x + d_y) \tag{3}$$

The Diagonal distance heuristic function is defined in Equation (4) as:

$$f_{node} = D \times (d_x + d_y) + (D^2 - 2 \times D) \times \min(d_x, d_y) \tag{4}$$

The heuristic function can also be defined based upon Euclidean distance:

$$f_{node} = D \times \sqrt{(d_x^2 + d_y^2)} \tag{5}$$

where D is considered as a constant [15].

The A-star path planning algorithm can find the shortest path to the goal position but it might encounter some sharp turns where the vehicle might get stuck. To make both the sharp turns easier and the path shorter, path smoothing is introduced to the system

The path[i] stores the ith position on the path, whereas newpath[i] holds the ith position on the new path.

The smoothed path is calculated by:

$$update_1 = weight_{data} \times (path[i] - newpath[i]) \tag{6}$$

$$update_2 = weight_{smooth} \times (newpath[i-1]) + newpath[i+1] - 2 \times newpath[i] \tag{7}$$

where, $weight_{smooth} = 0.65$ and $weight_{data} = 0.1$.

Eventually, the resultants of $update_1$ and $update_2$ are the arrays consist of smoothed path.

3 Problem description of self-parking with path planning

3.1 Introducing walls and parking lanes as obstacles

To use an A-star path planning algorithm for generating a correct path, static obstacles can be introduced so that paths can be generated by avoiding them. Since the parking area's map is already known by the user, the walls of the parking area can easily be introduced to the system as obstacles (Figure 1). Parking slots' demarkation is also introduced as obstacles that enable the vehicle to avoid the lanes and park between them.

By providing sufficient information about the environment and the goal position, the mobile robot can safely move towards the goal using its prior knowledge and real-time information obtained by the sensors. This process can be referred to as robot navigation [16]. Image processing techniques are used to specify walls of the environment and parking lanes as static obstacles. By the use of color thresholding, obstacles are assigned the value of 1; whereas, non-obstacles are represented by 0.

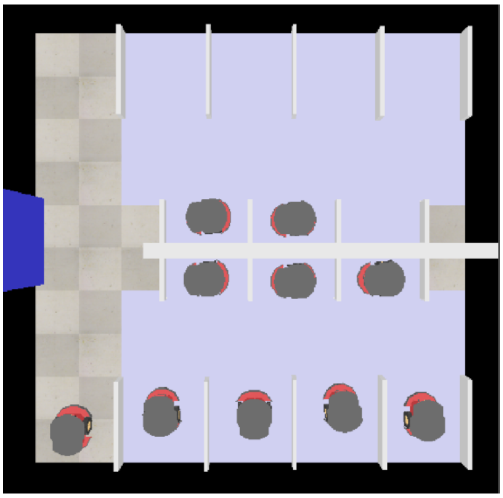

Figure 1. Simulation scene by the top-view camera

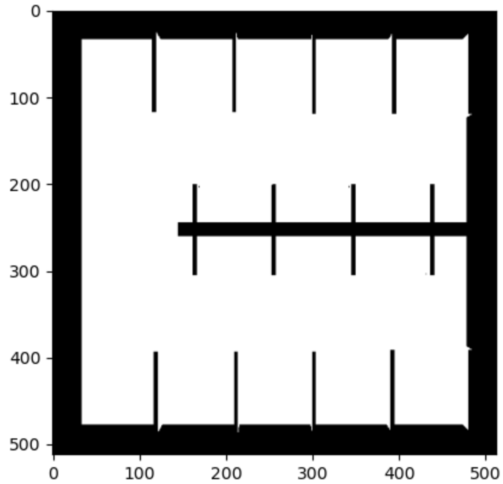

Figure 2. Color thresholding applied on the scene

Through the difference between the two values introduces, the obstacles are identified. In Figure 1, by using the VREP top view, a snapshot is taken and fed to the system after modifying it by the color thresholding, as shown in Figure 2.

In the simulation, it is assumed that the mobile robot can provide the current position and the orientation by using the information from GPS and gyroscope. To transform position information from pixels in the image to the simulation scene, basic scaling is applied as:

$$scene_{pos} = (image_{upsidedown} \times -scaling_{factor}) + [2.74, 2.74] \quad (8)$$

Where, $scaling_{factor} = 0.01075269$ is assumed.

Similarly,

$$image_{pos} = (image_{upsidedown} \times -scaling_{factor}) + [256, 256] \quad (9)$$

Where, $scaling_{factor}$ is considered as 93.

Both resultants are arrays and the first components of the arrays represent x coordinate position while the second components of the arrays represent y coordinate position.

3.2 Determining the empty for parking

Path planning includes generating a geometric path from the robot's current location to the goal if a map and goal position is provided [16]. To generate a path, goal position information is required. Since there are fourteen parking slots and each of the slot has different positions, some parking order has to be introduced to the system. To check the availability of the parking slot, ray type infrared sensors are installed at each slot. If a sensor returns a true value, it means that the lot is not available. Smaller numbered parking slots have the priority to be parked because they are closer to the entrance. If there is no available slot, the user is informed about the situation.

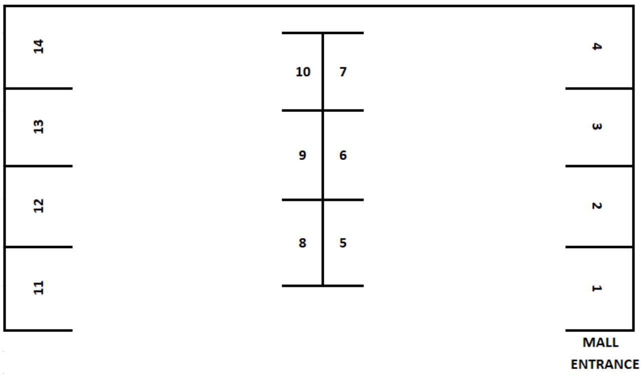

Figure 3. Parking slots and their numbers

In Figure 3, a general overview of the environment is provided to make the environment more understandable by the user.

Slots with the numbers of 1, 2, 3, 4, 11, 12, 13, and 14 are available for backward and forward parking depending on the preference; whereas, the slots 5, 6, 7, 8, 9, and 10 are only available for parallel parking.

3.3 Energy consumption differences for different ways of parking

Information that can be obtained from the simulation is the distances between goal and start positions, and the time required to plan the path and cover the distance. To observe the energy consumption, information from the simulation should be used.

One of the easiest and most common ways of determining the vehicle's energy consumption is using the shortest distance as an evaluation index which is a static index. It is decided by the static length of each section of the optimal path, so the vehicle's consumption on the optimal path uses the shortest distance as evaluation index in Equation (10):

$$F_D = \sum_{k=0}^{n} D_k \quad (10)$$

where F_D represents the total energy consumption that is dependent on distance from the start position to the goal position over n steps; whereas, D_k is the kth position's distance from the start point.

The minimum time is used as an evaluation index, being a dynamic index, it is not easy to realize.

During the whole process of path optimization, the average speed of the vehicle on each section in real-time is required, no doubt it will increase the computational complexity. The vehicle's consumption under this evaluation index on the optimal path is [17]:

$$F_t = \sum_{k=0}^{n} \frac{D_k}{V_k} \tag{11}$$

where F_t represents the energy consumption dependent on travelling time to park the robot; whereas V_k is the average velocity that robot experiences from start position to the kth position on the path.

In this case, since velocity is controlled by a PID controller and is a changing variable during travel, energy consumption cannot be observed by only considering traveling time. To compare energy consumptions for different parking slots, Equation (10) can be used. The farther the parking slot is from the entrance, the higher the energy consumption will be. To compare energy consumptions for the same slots but different ways of parking, Equation (11) can be used.

It is observed that time consumed in the parking is directly proportional to the energy consumption; moreover, this consumption can vary depending upon different ways of parking. To compare and detect the energy consumptions for different slots, either distance or time indexes are used.

4 Simulation results

Two different analyses are made based upon three different scenarios in the simulation environment. The first analysis highlights the energy consumption differences among different ways of parking to the same slots and parallel parking. The second analysis discusses the position error of the mobile robot as it maneuvers towards the goal position.

Information considered for evaluation is traveled distance, traveling time, path generation time, ways of parking, and position error between the current position and the goal position.

In the simulation results, distances are represented in centimeters, and time is represented in minutes. TD represents traveled distance measured using Manhattan distance, TT represents traveling time, PGT represents path generation time and WoP represents the way of parking. So, the resultants of Equations (10) and (11) are TD and TT, the information is used to determine the energy consumptions.

Table 1. All information of every way of parking to each slot

Slot #	TD(cm)	TT(min)	PGT(min)	WoP
1	254	1.14	2.51	Backward
2	524	1.12	2.05	Backward
3	917	1.27	1.82	Backward
4	1333	1.35	2.88	Backward
11	771	1.35	1.89	Backward
12	1241	1.53	1.63	Backward
13	1779	2.15	2.54	Backward
14	2143	2.30	3.46	Backward
1	479	0.38	1.48	Forward
2	839	0.49	1.69	Forward
3	1258	0.59	3.00	Forward
4	1727	1.09	3.59	Forward
11	1099	1.01	2.03	Forward
12	1589	1.15	2.13	Forward
13	2055	1.36	2.59	Forward
14	2321	1.58	2.84	Forward
5	528	1.50	0.62	Parallel
6	904	2.04	0.80	Parallel
7	1333	2.24	1.00	Parallel
8	1241	2.21	1.05	Parallel
9	1779	2.30	1.20	Parallel
10	2143	2.52	1.36	Parallel

Table 1 is recorded to show the necessary information for each slot based upon three different scenarios.

Scenario 1

In scenario 1, available slots open for parking are 1, 5, and 10. According to the algorithm, the closest slot determined is 1; hence, the distance travelled is minimum from the entrance to that parking slot.

Scenario 2

In scenario 2, available slots open for parking are 11, 12, and 13. Due to the algorithm, the mobile robot decided to park at slot 11.

Scenario 3

In scenario 3, the mobile robot is given the task to perform fourteen iterations to park at un-visited slot in every iterations according to their way of parking.

As it can be observed from Table 1, backward parking's path generation times and traveled distances are lower while traveling times are higher. The reason that traveled distances and path generation times are lower is that the vehicle would

have made some maneuvers to park backward. In the end, the traveled distance is the same but the traveled distance on a path is different. Since generated path for backward parking has to be in front of a parking lot, the path is generated for a shorter distance. So that path generation times and the traveled distances resulted lower.
Since, $F_{D_{forward}} \cong F_{D_{backward}}$ and $F_{t_{backward}} > F_{t_{forward}}$, energy consumption for backward parking is higher (Table 1). In this case, forward parking provides two main advantages: lower energy consumption and parking in a lesser time.

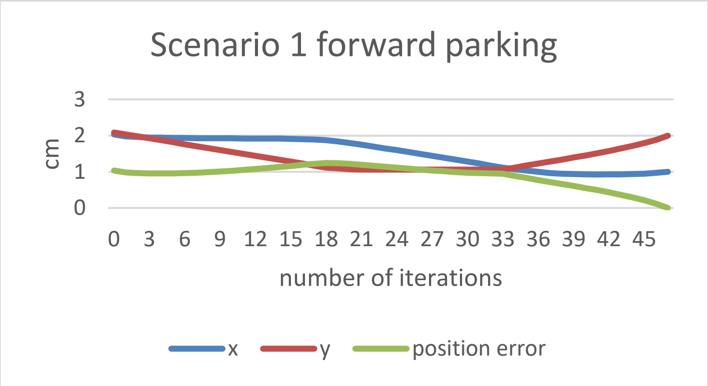

Figure 4. Position error for scenario 1

In Figure 4, the robot's position in x, y and error is depicted for the forward parking in scenario 1.

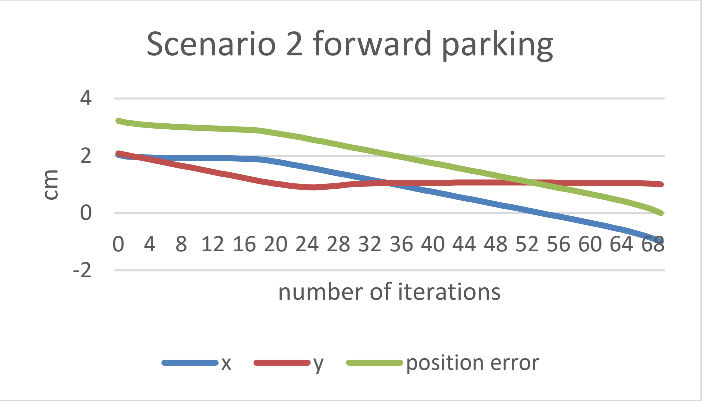

Figure 5. Position error for scenario 2

Similar to the Figure 4, Figure 5 is recorded to observe the position error in scenario 2 for forward parking.

Conclusion

Self-parking achieved by the A-star path planning algorithm has some advantages over free slot detection by a vision-based method where the map of the parking area is available. These advantages can be considered as parking in less time and low energy consumption during the parking process. Some sort of detection errors due to the vision sensor's position and disruption on the parking lanes have also been overcome since parking lanes have been introduced as obstacles to the system.

Though the A-star algorithm can find the shortest path, not necessarily the smoothed path. Therefore, path smoothing is introduced to smooth the sharp turns, which would have been challenging for the vehicle, and to shorten the path. Different ways of parking resulted in different amounts of energy consumptions. After simulating the system, it is observed that forward parking has some advantages over other ways of parking. As there are some maneuvering is required for the mobile robot to park in parallel and backward, it takes more time and higher energy consumption. For further works, the proposed system can be tested on a hardware platform and the system's behavior can be observed for both static and dynamic environments.

References

[1] Gehrig SK, Stein FJ. "Dead reckoning and cartography using stereo vision for an autonomous car". *IEEE/RSJ 1999 International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No. 99CH36289)*, Kyongju, Korea, 17-21 October 1999.

[2] Liden D. "What Is a Driverless Car?". *WiseGeek*, 11 October 2013.

[3] Kelly H. "Self-Driving Cars now Legal in California". https://edition.cnn.com/2012/09/25/tech/innovation/self-driving-car-california/index.html (17.07.2021).

[4] Thrun S. "Toward Robotic Cars". *Communications of the ACM*, 53(4), 99-106, 2010.

[5] Guizzo E. "How Google's Self-Driving Car Works". https://spectrum.ieee.org/how-google-self-driving-car-works (17.07.2021).

[6] Fisher A. "Inside Google's Quest To Popularize Self-Driving Cars". *Popular Science,* 18, 2013-09, 2013.

[7] Paul T. " BMW demos a self-parking BMW 730i". https://paultan.org/2006/07/12/bmw-demos-a-self-parking-bmw-730i/ (17.07.2021).

[8] Chrysler D. "New Mercedes CL Class: Parking Assistance Technology", Aug. 2006.

[9] Audi Industrial Group. "Audi Parking System", https://www.audi-technology-portal.de/en/electrics-electronics/driver-assistant-systems/park-assist (17.07.2021)

[10] Toyota Industrial Group. "Toyota to Launch Enhanced Parking Support Systems", https://global.toyota/en/detail/4201181 (17.07.2021)

[11] Bibi N, Majid MN, Dawood H, Guo P. “Automatic parking space detection system”. 2nd International Conference on Multimedia and Image Processing(ICMIP), Wuhan, China, 17-19 March 2017.

[12] WHO. “Road Traffic Injuries” https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries(17.07.2021)

[13] Tesla Inc. “Vehicle Safety Report” https://www.tesla.com/VehicleSafetyReport (17.07.2021)

[14] Min W, Liu G. “Footstep planning for biped robots based on A* algorithm”. *Computer Measurement & Control,* 21, 157-159, 2013.

[15] Amit P. “Amit’s thoughts on pathfinding”. http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html#a-stars-use-of-the-heuristic(17.07.2021)

[16] Maria I. R. “Obstacle avoidance,” Instituto de Sistemas e Robótica, Instituto Superio Técnico, 2005, pp. 1.

[17] Cheng L, Liu C, Yan B. “Improved hierarchical A-star algorithm for optimal parking path planning of the large parking lot”. *IEEE 2014 International Conference on Information and Automation(ICIA),* Hailar, China, 28-30 Jul 2014.