International Conference on Recent Trends in Data Science and its Applications
DOI: rp-9788770040723.082

An Overview of different Autonomous Vehicle Algorithms for Sensing and Route Planning

B. GovindSudarshan
Department of Computational Intelligence,
School of Computing SRM Institute of Science and Technology
Kattankulathur, Chennai, Tamil Nadu, India

Maragatham G
Department of Computational Intelligence,
School of Computing SRM Institute of Science and Technology
Kattankulathur, Chennai, Tamil Nadu, India

***Abstract*—The topic of self-driving cars has recently gained popularity. This paper presents an overview of several algorithms often employed in autonomous vehicle sensing and path planning. We have compiled a list of different algorithms used for these two purposes and provided a description of the same. A brief description of different sensors typically used in autonomous vehicles has also been provided. Sensor algorithms have been divided into 'Locational and Positioning' and 'Environmental modeling'. Location and Positioning Algorithms require the use of GPS, IMU, and Encoders to estimate accurate distance and use some variant of the Kalman Filter. Environmental modeling algorithms mainly rely on Cameras and use deep-learning object detection models like YOLO. There are also methods that require the use of Cameras with LiDAR and RADAR. For path planning, a description of the two most popular methods A* and Dijkstra has been provided along with their comparison.**

***Index Terms*—Autonomous Vehicles, Positioning, Environment, Heuristics**

I. INTRODUCTION

The term autonomous vehicle refers to any vehicle that does not require human interaction to perform its task. Lately, with the increase in technology, especially advancements in the field of deep-learning, capabilities of autonomous vehicles have increased tremendously. A typical autonomous vehicle consists of standard locomotive machinery, sensors in order to gain insight into the environment, a central dedicated CPU that gives signals to the locomotive machinery based on user input (such as target destination, etc.) and a map database. Autonomous vehicles equipped with such technology can most possibly improve energy efficiency, reduce the risk of collisions and reduce pollution [1]-[2] since most autonomous vehicle are electric in nature. These are various algorithms available that coordinate the working of an autonomous vehicle(used by the CPU) that we have categorized. The paper focuses on different algorithms involved in different parts of the system. We start with the sensors. Sensors themselves can be categorized into various parts[3]. They include sensors like LiDAR[4], RADAR[5]-[6], Camera[7]-[8], Ultrasonic Sensors[9], GPS (Global Positioning System)[10] and IMU (Inertial Measurement Unit)[11]. Focusing on the routing algorithm or the main algorithm, there are many, however, they can be broadly classified into heuristic[12]-[14] and nonheuristicalgorithms[15]. These algorithms are also dynamic, they have an inbuilt obstacle avoidance system. The main input for these algorithms is the sensor values and the map database.

Heuristic algorithms may also take into account information like traffic density which would probably require an internet connection. Heuristic functions are mostly dependent on the heuristic function provided[16]. In path-finding algorithms, modern heuristic algorithms perform better than non-heuristic algorithms[17]-[18]. For our paper, we shall only consider one heuristic algorithm[19]-[20] (A*) and one non-heuristic algorithm[21](Dijkstra).

II. SENSOR OVERVIEW

A. LiDAR

LiDAR, which stands for light detection and ranging, is most commonly used to determine object ranges. It works by directing a beam of laser at an object and calculates the distance from the beam source by calculating the time it takes for the reflected laser beam to return. Using algorithms like SLAM(Simultaneous Location and Mapping), it is even possible to generate visual 3-D representations of the environment and as such, this sensor is extensively used in autonomous vehicles for environment detection to determine object ranges.

B. RADAR

RADAR(radio detection and ranging) is an algorithm very similar to LiDAR except for the fact that it requires the use of radio waves rather than laser beams. It is mostly used for the detection of objects and has been used in defense for detecting enemy vehicles. For use in autonomous vehicles, RADAR is generally used with radio waves having a frequency of 24,74, 77, and 79 GHz. Generally, two types of RADAR exist that have been used in autonomous vehicles namely Impulse RADAR and (FMCW) Frequency Modulated Continuous Wave Radar. Impulse RADAR emits only one pulse, the frequency of which remains constant. FMCW RADAR, on the other hand, emits pulses continually at different frequencies. FMCW is widely used owing to its high-depth perception[22].

C. Camera

A Camera is a form of sensor that records images of objects. It is comprised of a lens and a shutter used for projecting the lens image on a surface(typically a photosensitive film or an electronic sensor) or directly translated into electronic signals. In autonomous vehicles, a Camera is typically used with an object detection algorithm which detects surroundings and objects of interest. Using Photogrammetry methods,estimation of the ranges of objects is also possible. Most popular object detection methods for use with camera in Autonomous vehicles are YOLo(You Only Look Once) and SSD(Single shot detection) as they are faster than most object detection algorithms. Camera is the most widely used sensor for autonomous vehicles and in most scenarios has replaced LiDAR, RADAR and ultrasonic sensors.

Ultrasonic Sensors

DOI: rp-9788770040723.082

This sensor is used in fusion with other sensors in environ- mental modelling for error correction and to get an accurate reading of the ranges, It works similar to a LiDAR and a RADAR by using Ultrasonic Waves to determine the distance of objects. It works at a very high frequency and as such is limited to close range detection only. Ultrasonic sensor can be found in almost every modern car these days. They assist in parking.

D. GPS

It is a satellite based radio-navigation system. Most autonomous cars contain a GPS receiver which receives data from satellites continuously. For accurate measurement, the GPS typically receives data from 4 or more satellites. Satellites used for transmission of GPS signals use an atomic clock which provides the broadcast time to be encoded in the signal. The reciever uses the broadcast time, satellite location and the time that the data was received to calculate current location.

E. IMU

It is a fusion of a gyroscope and an accelerometer. Gyro- scope is used for the measurement of rotation and accelerometer is used for measuring odometry. In scenarios where GPS signals are not available, IMUs are used for location and positioning of autonomous vehicles. IMU's are typically used in co-ordination with encoders.

F. Encoders

An encoder is a sensor attached to a wheel or a rotation de- vice to measure rotation. It works similar to an accelerometerby measuring odometry of a vehicle given wheel radius. BothIMUs and Encoders are used to measure relative position of an autonomous vehicle. There are 4 main types of encoders namely optical, mechanical, magnetic and electromagnetic. The specific type of encoder typically used in autonomous vehicles is an incremental rotary encoder which can be of any type. Incremental rotary encoders are used widely because of their accuracy and ability to provide real time data with efficiency.

III. SENSOR ALGORITHMS

A. Location and Positioning

Autonomous cars have a map database kept inside them or obtain it through a server connection; nonetheless, for route planning, the current position with regard to the map must be estimated. A GPS system is used to determine the current approximate position however this is prone to errorand GPS outages. [23] provides a way for combining GPS, IMU sensors, and encoders. It is an odometry model that is based on Ackerman steering geometry and is computed using encoder data from the rear two wheels and steering. Because there is sensor fusion, it employs the Unscented Kalman Filter (UKF).

B. Environment modelling

There are various algorithms used for this purpose. For autonomous vehicles, the representation of environment should be dynamic i.e. it should take into consideration changes in the environment as well. [24] proposes a very simple solution based on Camera. It uses and object detection deep learning algorithm YOLO v2 trained on a set of vehicle images which is then used for identification of vehicles. They pass the information from YOLO v2 to a ROS node which then estimates the approximate distance of the vehicle. [25] proposes a similar method which uses Tiny YOLO and estimating neighbouring vehicle speed by tracking their taillights. [26] presents an approach to this problem for indoor environments by using a 2D LiDAR and a low cost camera.

IV. MAIN/ROUTE PLANNING ALGORITHMS

A. A* algorithm

A* is a well-known heuristic-based technique that is commonly employed in graph search issues. In 1968, Stanford Research Institute's Peter Hart, Nils Nilsson, and Bertram Raphael suggested it. By transforming the map to a graph and using heuristics as information supplied by sensor values, the route planning issue might be described in A*. The A* algorithm takes a greedy approach. It is a development of Dijkstra's Algorithm. The A* algorithm's evaluation function is provided by

$$f(n) = g(n) + h(n) \tag{1}$$

where$g(n)$ is the cost of getting to the current place and $h(n)$ is the expected cost of getting to the destination. The euclidean distance function or any other distance function may readily determine $h(n)$.

$$h(n) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

where (x_1, x_2) and (y_1, y_2) represent the co-ordinates of the current location and destination respectively. We can find their values using their latitudes and longitudes of the two points. We can also include heuristics like traffic and sensor information for more efficient path planning. The time complexity of this algorithm dependent on the heuristic function. [20] proposes an improved A* algorithm for path planning which focuses on obstacles and path proposals using circle equations. The algorithm proposed is more effective and faster by about 200%.

B. Dijkstra's algorithm

EdsgerWybeDijkstra, a computer scientist, devised this technique in 1956. It's a well-known non-heuristic path planning algorithm. This algorithm determines the shortest pathbetween two points on a graph. Given the current position and destination, this method returns the shortest distance or the shortest cost path. [27] describes how to apply this method on a network of interconnected roadways. Although Dijkstra always finds the shortest path, it does not ensure time efficiency. Because we must consider traffic, the shortest road is not always the quickest option. This approach has a temporal or time complexity of $O(V^2)$ when utilising an adjacency list representation of a graph, where V is the number of vertices in the graph.

Algorithm 1 A* Algorithm

```
1:  function A*(start, goal)
2:openSet    ← {start}
```

DOI: rp-9788770040723.082

closedSet ← {}
cameFrom ← {}
gScore[*start*] 0 ← {}
f Score[*start*] ← *heuristic*(*start, goal*)
while *openSet*is not empty **do**
current ← the node in *openSet*with the lowest *f Score*
if *current* is *goal* **then**
return reconstructPath(*cameFrom, current*)
end if
remove *current* from *openSet*
add *current* to *closedSet*
for all neighbors *neighbor* of *current* **do**
tentativeGScore ← *gScore*[*current*] + *dist*(*current, neighbor*)
if *neighbor* is in *closedSet* and *tentativeGScore* ≥*gScore*[*neighbor*] **then**
continue
end if
if *neighbor* is not in *openSet* or *tentativeGScoregScore*[*neighbor*] **then**
cameFrom[*neighbor*] ← *current*
gScore[*neighbor*] ← *tentativeGScore*
f Score[*neighbor*] ← *gScore*[*neighbor*] + *heuristic*(*neighbor, goal*)
if *neighbor* is not in *openSet***then**
add *neighbor* to *openSet*
end if
end if
end for
end while
return failure
end function

C. Comparison between A* and Dijkstra

However, because A* is primarily dependent on its heuristic function, its optimality can only be established by the effectiveness of its heuristic function. Dijkstra has consistently outperformed in circumstances with little to no traffic. However, while dealing with traffic and having to include sensor data, A* performs better if these data are incorporated into its heuristic function. This is owing to the fact that A* might easily incorporate dynamicity in it's approach.[28] also suggests that A* is more efficient than Dijkstra in terms of destination finding if provided appropriate heuristic. [29] compares computational efficiency of both the algorithms in a static scenario whereas [30] compares both the algorithm in different applications on the field. The results of which are summarized in the table below.

Algorithm 2 Dijkstra's Algorithm

function DIJKSTRA(*start, goal*)
openSet ← {*start*}
closedSet ← {}
cameFrom ← {}
gScore[*start*] ← 0
while *openSet*is not empty **do**
current ← the node in *openSet*with the lowest *gScore*
if *current* is *goal* **then**
return reconstructPath(*cameFrom, current*)
end if
remove *current* from *openSet*
add *current* to *closedSet*
for all neighbors *neighbor* of *current* **do**
tentativeGScore ← *gScore*[*current*] + *dist*(*current, neighbor*)
if *neighbor* is in *closedSet* and *tentativeGScore* ≥*gScore*[*neighbor*] **then**
continue
end if
if *neighbor* is not in *openSet* or *tentativeGScoregScore*[*neighbor*] **then**
cameFrom[*neighbor*] ← *current*
gScore[*neighbor*] ← *tentativeGScore*
if *neighbor* is not in *openSet***then**
add *neighbor* to *openSet*
end if
end if
end for
end while
return failure
end function

TABLE I COMPARISON OF DIJKSTRA AND A*

	Dijkstra	A*
Advantages	Guaranteedtofindshortestpath Canuseheuristictoguidesearch Noneedforadmissibleheuristic	FasterthanDijkstra Canusedifferentheuristics Consistentlyfindsoptimalpath
Disadvantages	CanbeslowerthanA* Doesnotconsiderdistancetogoal Nowaytoguidesearchtowardsgoal	Requiresadmissibleheuristic Notguaranteedtobefasterthan Dijkstra

A* relies heavily on it's heuristic function and therefore it may or may not be faster than Dijkstra. We can see that Dijkstra outperforms A* in long distance path planning algorithms however in terms of time and memory A* is moreefficient. In dynamic environments, A* always outperforms Dijkstra.

TABLE II IMPLEMENTATION RESULTS FOR A* AND DIJKSTRA ALGORITHMS

		Metric Type	Result
A*	Static short distance Dynamic	Time Time	Faster than Dijkstra Faster than Dijkstra
Dijkstra	Staticlongdistanc Static	Time	Faster than Dijkstra

V. CONCLUSION

An overview of different popular algorithms have been provided and a comparison of two path planning algorithms has also been provided. Most of the sensor methods require the use of camera only owing to developments in distance estimation methods and the low cost of using only one Camera sensor however, a few methods still use a fusion of Camera and Lidar. In path planning algorithms, Dijkstra provides the shortest path, however A* may also consider information from sensors and traffic information. A* is shown to more efficient than Dijkstra.

DOI: rp-9788770040723.082

REFERENCES

[1] M. V. Rajasekhar and A. K. Jaswal, "Autonomous vehicles: The future of automobiles," 2015 IEEE International Transportation Electrification Conference (ITEC), pp. 1-6, 2015,doi: 10.1109/ITEC-India.2015.7386874.

[2] S. Malik, H. A. Khattak, Z. Ameer, U. Shoaib, H. T. Rauf and H. Song, "Proactive Scheduling and Resource Management for Connected Autonomous Vehicles: A Data Science Perspective," in IEEE Sensors Journal, vol. 21, no. 22, pp. 25151-25160, 15 Nov.15, 2021, doi: 10.1109/JSEN.2021.3074785.

[3] S. Campbell et al., "Sensor Technology in Autonomous Vehicles : A review," 2018 29th Irish Signals and Systems Conference (ISSC), pp. 1-4, 2018,doi: 10.1109/ISSC.2018.8585340.

[4] C. Ilas, "Perception in autonomous ground vehicles", International Conference on Electronics Computers and Artificial Intelligence (ECAI), 2013.

[5] Jri Lee, Yi-An Li, Meng-Hsiung Hung and Shih-Jou Huang, "A Fully- Integrated 77-GHz FMCW Radar Transceiver in 65-nm CMOS Tech- nology", IEEE Journal of Solid-State Circuits, 2010.

[6] H. Rohling and C. Moller, "Radar waveform for automotive radar systems and applications", 2008 IEEE Radar Conference, pp. 1-4, 2008.

[7] A. Elmquist and D. Negrut, "Modeling Cameras for Autonomous Vehicle and Robot Simulation: An Overview," in IEEE Sensors Journal, vol. 21, no. 22, pp. 25547-25560, 15 Nov.15, 2021, doi: 10.1109/JSEN.2021.3118952.

[8] L. Xiaoming, Qin Tian, Chen Wanchun and Y. Xingliang, "Real-time distance measurement using a modified camera", 2010 IEEE Sensors Applications Symposium (SAS), pp. 54-58, 2010.

[9] Wan-Joo Park, Byung-Sung Kim, Dong-EunSeo, Dong-Suk Kim and Kwae-Hi Lee, "Parking space detection using ultrasonic sensor in parking assistance system", 2008 IEEE Intelligent Vehicles.

[10] W. Rahiman and Z. Zainal, "An Overview of Development GPS Navigation for Autonomous Car", 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2013.

[11] Rajesh, M., &Sitharthan, R. (2022). Image fusion and enhancement based on energy of the pixel using Deep Convolutional Neural Network. Multimedia Tools and Applications, 81(1), 873-885.

[12] Hoang, VD., Jo, KH. Path planning for autonomous vehicle based on heuristic searching using online images. Vietnam J Comput Sci., vol. 2, pp. 109–120, 2015.

[13] Abdallah A. Hassan, Hesham A. Rakha, A Fully-Distributed Heuristic Algorithm for Control of Autonomous Vehicle Movements at Isolated Intersections, International Journal of Transportation Science and Technology, vol. 3, issue 4, pp. 297-309, 2014,ISSN 2046-0430.

[14] Rahul Kala, and Kevin Warwick, "Heuristic based evolution for the coordination of autonomous vehicles in the absence of speed lanes," Applied Soft Computing, vol. 19, pp. 387-402, 2014,ISSN 1568-4946.

[15] G. Qing, Z. Zheng and X. Yue, "Path-planning of automated guided vehicle based on improved Dijkstra algorithm," 2017 29th Chinese Control and Decision Conference (CCDC), pp. 7138-7143, 2017,doi: 10.1109/CCDC.2017.7978471.

[16] Y. F. Yiu and R. Mahapatra, "Hierarchical Evolutionary Heuristic A Search," 2020 IEEE International Conference on Humanized Computing and Communication with Artificial Intelligence (HCCAI), pp. 33- 40, 2020,doi: 10.1109/HCCAI49649.2020.00011.

[17] T. Goto, T. Kosaka and H. Noborio, "On the heuristics of A* or A algorithm in ITS and robot path-planning," Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), vol.2,pp. 1159-1166, 2003,doi: 10.1109/IROS.2003.124880.

[18] S. Caselli, M. Reggiani and R. Rocchi, "Heuristic methods for randomized path planning in potential fields," Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No.01EX515), pp. 426-431, 2001,doi: 10.1109/CIRA.2001.1013238.

[19] Zheng-hong Hu and Jin Li, "Application and implementation of a* algo- rithm in picture matching path-finding," 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), pp. V5-224-V5-228, 2010,doi: 10.1109/ICCASM.2010.5619180.

[20] C. Ju, Q. Luo and X. Yan, "Path Planning Using an Improved A-star Algorithm," 2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan), 2020, pp. 23-26, doi: 10.1109/PHM-Jinan48558.2020.00012.

[21] D. -D. Zhu and J. -Q. Sun, "A New Algorithm Based on Dijkstra for Ve- hicle Path Planning Considering Intersection Attribute," in IEEE Access, vol. 9, pp. 19761-19775, 2021, doi: 10.1109/ACCESS.2021.3053169.

[22] Cadence System Analysis et al., The use of radar technology in Autonomous Vehicles, Cadence. 2022, Available at: https://resources.system-analysis.cadence.com/blog/msa2022-the-use-of-radar-technology-in- autonomous-vehicles (Accessed: November 19, 2022).

[23] Jeon, Daesung et al. "UKF data fusion of odometry and magnetic sensor for a precise indoor localization system of an autonomous vehicle." 2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI),pp. 47-52, 2016.

[24] M. Gluhakovic´, M. Herceg, M. Popovic and J. Kovacˇevic´, "Ve- hicle Detection in the Autonomous Vehicle Environment for Po- tential Collision Warning," 2020 Zooming Innovation in Con sumer Technologies Conference (ZINC), pp. 178-183, 2020,doi: 10.1109/ZINC50678.2020.9161791.

[25] Pazhani. A, A. J., Gunasekaran, P., Shanmuganathan, V., Lim, S., Madasamy, K., Manoharan, R., &Verma, A. (2022). Peer–Peer Communication Using Novel Slice Handover Algorithm for 5G Wireless Networks. Journal of Sensor and Actuator Networks, 11(4), 82.

[26] pp. 1-5, 2021,doi: 10.1109/RTUCON53541.2021.9711733.

[27] J. Li, X. He and J. Li, "2D LiDAR and camera fusion in 3D modelling of indoor environment", National Aerospace and Electronics Conference (NAECON), 2015.

[28] M. Parulekar, V. Padte, T. Shah, K. Shroff and R. Shetty, "Automatic vehicle navigation using Dijkstra's Algorithm," 2013 International Con- ference on Advances in Technology and Engineering (ICATE), pp. 1-5, 2013,doi: 10.1109/ICAdTE.2013.6524721.

[29] G. M. Patel, A. Singh, T. Bhala, A. Jora and D. Chandna, "A Comparative Study on Shortest Path Visualization using Artificial Intelligence," 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 1-4, 2021,doi: 10.1109/ICRITO51393.2021.9596550.

[30] M. A. Djojo and K. Karyono, "Computational load analysis of Dijkstra, A*, and Floyd-Warshall algorithms in mesh network," 2013 Interna- tional Conference on Robotics, Biomimetics, Intelligent Computational Systems, Jogjakarta, Indonesia, pp. 104-108, 2013,doi: 10.1109/RO- BIONETICS.2013.6743587.

[31] J. Pak, J. Kim, Y. Park and H. I. Son, "Field Evaluation of Path-Planning Algorithms for Autonomous Mobile Robot in Smart Farms,"in IEEE Access, vol. 10, pp. 60253-60266, 2022, doi: 10.1109/AC- CESS.2022.3181131.