Intelligent Control and Automation, 2020, 11, 33-46
https://www.scirp.org/journal/ica
ISSN Online: 2153-0661
ISSN Print: 2153-0653

Business Plan for Autonomous Delivery Robot

Yangyang Li

Brookline High School, Massachusetts, USA
Email: 21liy@brooklinek12.org

How to cite this paper: Li, Y.Y. (2020) Business Plan for Autonomous Delivery Robot. *Intelligent Control and Automation*, **11**, 33-46.
https://doi.org/10.4236/ica.2020.112004

Received: April 21, 2020
Accepted: May 25, 2020
Published: May 28, 2020

Abstract

This paper introduces an autonomous robot (AR) cart to execute the last mile delivery task. We use navigation and intelligent avoidance algorithms to plan the path of the automatic robot. When AR encounters a new unrecognizable terrain, it will give control to the customer who can control the AR on its mobile app and navigate to the specified destination. We have initially designed an autonomous delivery robot with the cost of 2774 dollars.

Keywords

Autonomous Robot, Online Order, Delivery Within-Community, Intelligent Algorithm

1. Introduction

With the advance of technologies, the cost of an LiDAR continues to fall and the network communication speed becomes faster and faster, which makes the realization of autonomous robot delivery possible. Drone transportation is very common now [1]. In the military field, there are already smart cars that automatically evade the enemy and transport goods [2]. The real-world application of automation has been discussed since the late 20th century. According to the National Highway Traffic Safety Administration (NHTSA) of the United States, self-driving cars are categorized into 5 levels: Level 0—No Automation, Level 1—Function Specific Automation, Level 2—Combined Function Automation, Level 3—Limited Self Driving Automation, Level 4—Full Self Driving Automation. As the technology continues to develop, self-driving cars are expected to be real-world practical in another 5 - 10 years. Autonomous guided vehicles have found numerous applications in the industries, hospitals, army, etc. [3] [4].

Due the thrive of Internet and the surge of online order, solution to the last mile delivery has become a more and more urgent issue. "This is a white-hot market for jobs, particularly in grocery," said Bill Lewis, director of the retail

practice at Alix Partners. "The online demand for grocery is now about $20 billion for same-day delivery annually, and that's projected to [hit] about $80 billion over the next four years. There's a hot market for individual contractors and services to supply the labor for those deliveries." Form the mainstream perspective, using autonomous robot cart (as shown in Figure 1) to execute the last mile delivery task can bring a considerable sum of profits. For short-range transportation robots used in the community, there is little related research, which is our research object.

Our proposal of Autonomous Robot (referred as AR) is a level 3 autonomous robot cart. We use positioning methods [5] [6] [7] [8] and avoidance algorithms [9] [10] [11] [12] to plan the path of the automatic robot. In most of the time, it can self-navigate to a designated location. Navigation control in these situations is mostly done using the image captured by the camera. As the images captured are usually correlated with noises, a suitable boundary detection algorithm was developed to get the accuracy and robustness against the different noises. The AR would replace the conventional way of transferring materials within an allocated space. It will ensure faster, safer and dependable way of delivering materials. However, to reinsure the delivery will be delivered to the customer, when AR encounters new and unrecognizable terrain, it will pass its control to the customer. As such customer can control the AR on their mobile app and navigate it to come to their appointed destinations.

2. Model Protocol

The system architecture of AR consists of three main components (as shown in Figure 2).

2.1. Physical Delivery Unit

This is a physical layer of the AR system, which is an autonomous robot cart with abilities to transport freight with light level of manned control. The components of a robot unit include but not limits to Controllers, GPS receiver, Magnetometer, Delivery carriage and motors.

Figure 1. A kind of autonomous robot cart.

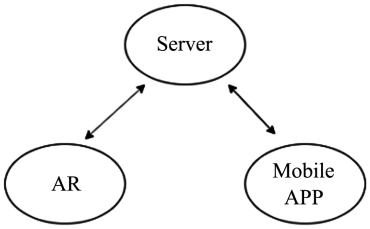


Figure 2. The system architecture of AR.

2.2. Mobile Application

Customers can place orders via our app on their mobiles. The mobile app also provides them ability to monitor and control the assigned AR for their orders. The app can consistently acquire the location of the AR and livestreams the real time traffic captured by the cameras installed on the AR. This application is developed in both IOS version and Android Version.

2.3. Cloud Based Server

The cloud base server is essential for human-robot interaction in terms of communication between AR and customers. The cloud-based server enables the long-distance real-time monitoring and two-way data transmission by using the network communication.

3. System Features

Our AR is equipped with 9 cameras and LiDAR sensors which can afford it a 360 degree of view of its surrounding as well as live-streaming the image back to the server; Crawler type wheels which enable it to travel through snow/raining field without slipping, as well as going up stair; LED screen installed on top which is used to display QR code to customers to unlock the storing trunk; image-processing sensor that can encode with OpenCV for image recognition. With the hardware and installed AI system, it can realize following features:

- Real-time Localization
- Live Streaming
- Switch Manual Control
- Path Planning in Dynamic Environment
- Emergency Kill switch
- QR code locked Delivery cart
- Customized Map for simpler usage

3.1. Real-Time Localization

The GPS module constantly acquired the current location of the AR and send the location information to the server and the server updates it to the mobile app for customers to synchronize the delivery process (as shown in Figure 3). This is a security feature as the customers can take necessary action if any unexpected problems happened to the AR, such as the BOT is trapped among pedestrians or going off-course.

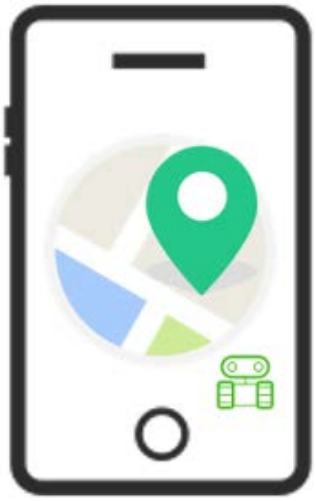


Figure 3. The GPS module.

3.2. Live Streaming

The Live Video Streaming function realization is done by the Raspberry Pi 3. By combining Dataplicity, Wormhole and cameras, Raspberry Pi 3 can encode the video using the hardware chip in H264 format. Customers can enable/turn off the livestream function on the app. The procedural of live streaming is shown as Figure 4. The server requirement is fulfilled using Dataplicity, which creates a temporary website called Wormhole. The Wormhole links are embedded in the application for live streaming.

3.3. Complete Manual Control

This is a security feature to deal with unexpected situations, when the autonomous system cannot be relied on. In those situations, a notification will be sent to the mobile app and ask the permission of customers to take over the control. If permission is granted, the mobile app can remote control the AR via viewing the live stream and navigate the AR through Google map.

3.4. Path Planning in Dynamic Environment

3.4.1. Local Planning

In global planning, the map information is provided by Google map which is already known, while in local planning, the environment and traffic cannot be known in advance. As such, the AR needs to pre-build a map of the current environment by the captured image (as shown in Figure 5).

Recently, lots of efficiency path planning algorithms on local dynamic environment has been established, such as Ant Colony algorithm, Particle Swarm algorithm and Bacterial Foraging Optimization. Each of these methods has its own pros and cons. Among these, Bacterial Foraging Optimization is relatively efficient and has the most applicable scope. Also, BFO reduces the cost of computing power by avoiding most of the computation complexity, which makes it fit in small sized robots. The BFO working flowchart is shown as Figure 6.

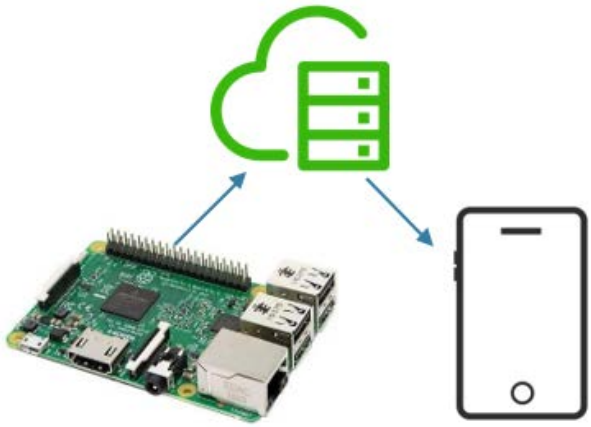


Figure 4. The procedural of live streaming.

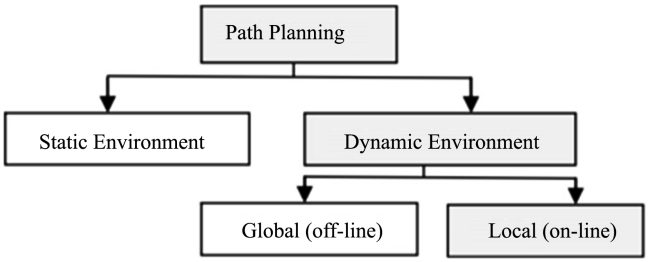


Figure 5. The current environment by the captured image.

The initial location of AR, the destination and the position of obstacles will be input as parameters. The final output will be a shortest pathway without crashing with any obstacles.

Step 1: ($S=1,2,\cdots,n$) virtual particles are generated according to the detected positions of obstacles with radius R around the AR's position. Each particle position would be defined as $P_s(t)$ at any given time t, the position after a small period of time is calculated as $P_s(t+dt)=P_s(t)+R\left(\Delta(t)/\left\|\Delta(t)\right\|\right)$

$$J_{\text{obstacle}}=H_{\text{obstacle}}*\exp\left(-W_{\text{obstacle}}\left\|\theta_i(t)-P_0(t)\right\|^2\right) \tag{1}$$

If $\left\|P_0(t)-C(t)\right\|^2\le\beta$

$$J_{\text{obstacle}}=0 \tag{2}$$

where Δt and $\left\|\Delta t\right\|$ is defined as a unit length of a random vector used to defined the moving direction of the particle and magnitude of it, respectively.

Step 2: During this step, our AR has met with the moving obstacles, a repellant Gaussian cost function is assigned to each position of detected obstacle to find out the best particle: where Hobstacle and Wobstacle are the constants defining height and weight of the repellant, $P_0(t)$ is the position of the obstacle detected by sensors, β is the sensor range.

$$J_{\text{goal}}=-H_{\text{goal}}*\exp\left(-W_{\text{goal}}\left(\left\|\theta_i(t)-P_G(t)\right\|^2\right)\right) \tag{3}$$

$$J=J_{\text{obstacle}}+J_{\text{goal}} \tag{4}$$

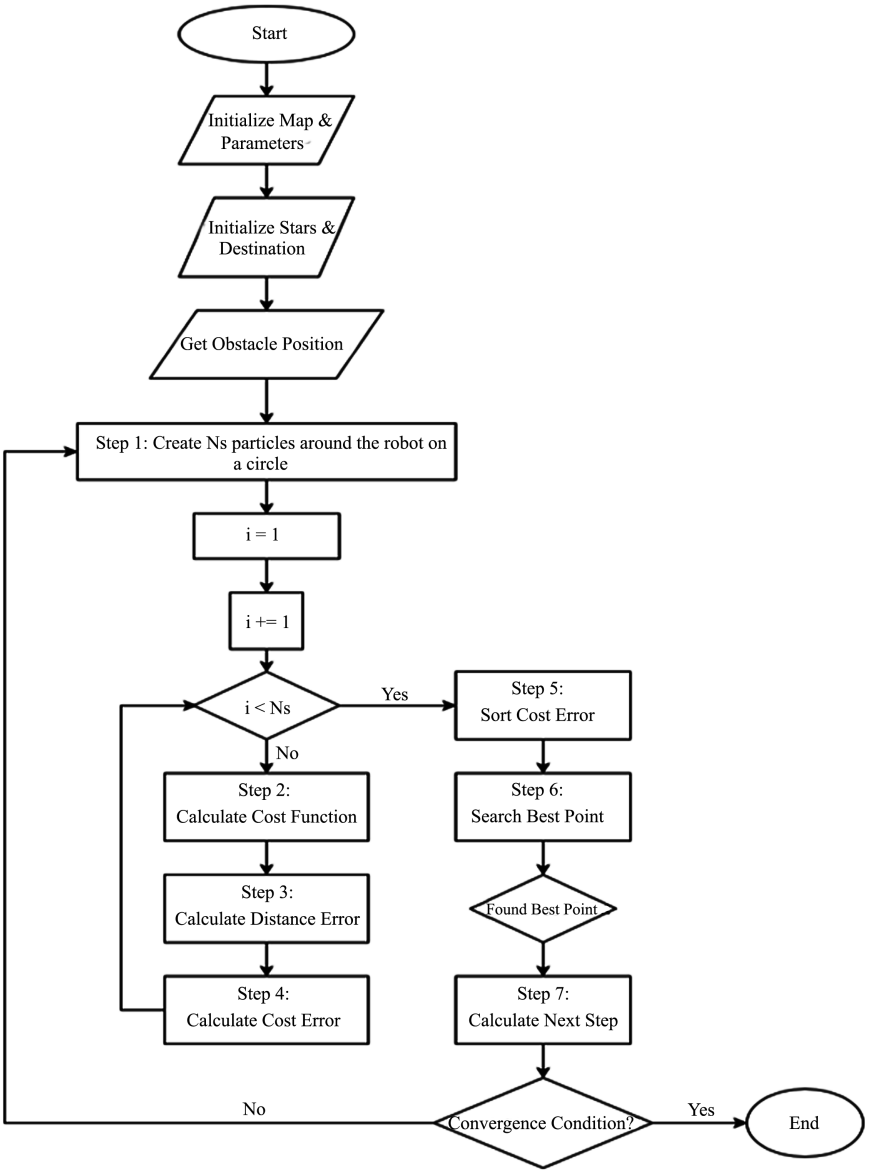


Figure 6. The flowchart of BFO.

Variables are defined in the same manner as (3). The total cost function is calculated by (4).

Step 3 & 4.

The best particle is decided by the distance error and cos function error to the target, which is calculated by (5) and the cost function error is calculated by (6). where $d_s(t) = \left\| P_s(t) - P_G(t) \right\|^2$.

$$e_s^d(t) = d_s(t+dt) - d_s(t) \tag{5}$$

$$e_s^J(t) = J\left(P_s(t+dt)\right) - J\left(P_s(t)\right) \tag{6}$$

Step 5 & Step 6:

At step 5, all particles are sort in ascending order of cost error, such that the particle with the lowest distance error will be in the top, which is the best particle

we try to find.

Step 7:

After the best particle has been found, add the particle to the desired path and replaced the current position of AR with the new position. Continue repeating this process till the convergence condition is satisfied. (Simulation result is shown as Figure 7).

The template is designed so that author affiliations are not repeated each time for multiple authors of the same affiliation. Please keep your affiliations as succinct as possible (for example, do NOT post your job titles, positions, academic degrees, zip codes, names of building/street/district/province/state, etc.). This template was designed for two affiliations.

1) For author/s of only one affiliation: To change the default, adjust the template as follows.

- Selection: Highlight all author and affiliation lines.
- Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select "1 Column" from the selection palette.
- Deletion: Delete the author and affiliation lines for the second affiliation.

2) For author/s of more than two affiliations: To change the default, adjust the template as follows.

- Selection: Highlight all author and affiliation lines.
- Change number of columns: Select the "Columns" icon from the MS Word Standard toolbar and then select "1 Column" from the selection palette.
- Highlight author and affiliation lines of affiliation 1 and copy this selection.
- Formatting: Insert one hard return immediately after the last character of the last affiliation line. Then paste down the copy of affiliation 1. Repeat as necessary for each additional affiliation.

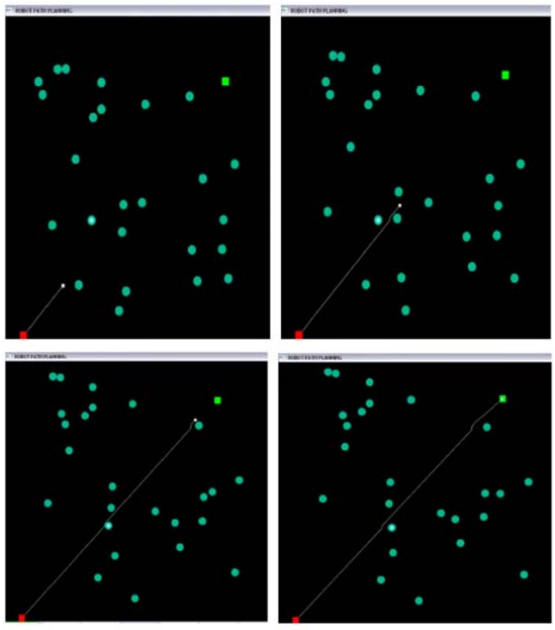


Figure 7. The simulation result.

3.4.2. Obstacle Recognition and Avoidance with Ultrasonic Sensors and LiDAR

LiDAR is essential for the realization of automatic obstacle avoidance, which is the core mission of our AR. The algorithms we use focus on solving sensor error problems and improving algorithm accuracy. The scanning areas of the laser are divided into emergency, accurate and fuzzy. If the MR detects an obstacle in these layers, the corresponding algorithm can be used to perform the obstacle avoidance behavior.

The AR is modeled as a kinematic unicycle and has three degrees of freedom, x_r, y_r, and θ_r, which moves on the xy-plane and rotates about z-axis with an angular velocity ω in Figure 8. The kinematics of the wheeled robot is depicted by (7).

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta_r) & 0 \\ \sin(\theta_r) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{7}$$

As shown in Figure 8, ω_L and ω_R are used to represent the angular velocities of the left and the right wheels, respectively. The linear and angular velocity of the robot is given by (8).

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{b} & -\frac{r}{b} \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} \tag{8}$$

where the results of the joint are effected by left and right wheels. Where r is the constant radius of the wheels, and b denotes the constant distance between the two wheels.

We used Livox Mid-40/Mid-100 for obstacles detection. The scanning angle resolution is 0.5˚ and the scanning area is 190˚. The number of its direction vector is 361. In Figure 9, dk is set to the return value of LIDAR in each vector direction, indicating the distance between the obstacle point and MR in any vector

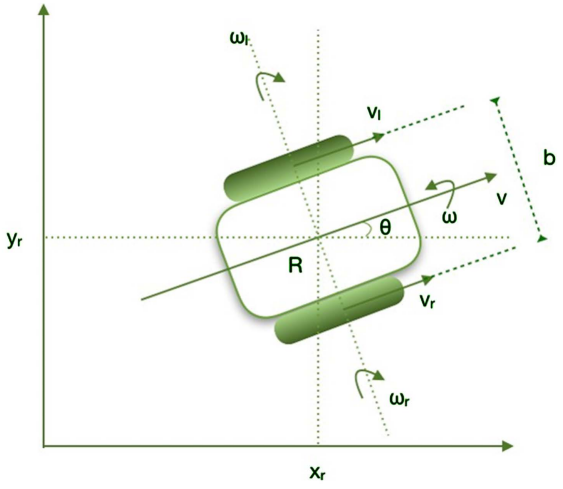


Figure 8. Where MED is the median filtering.

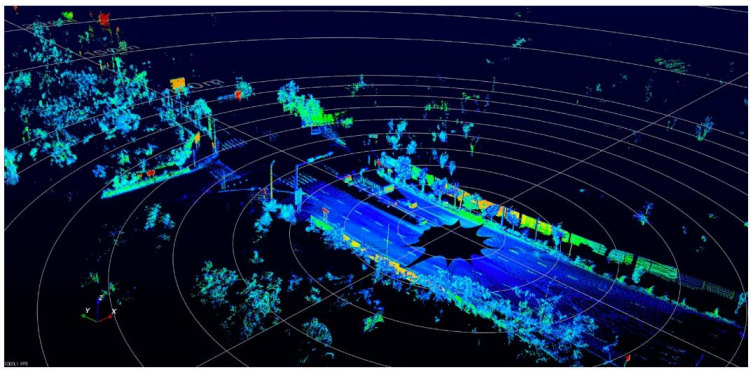


Figure 9. Where MED is the median filtering.

direction. Due to environmental effect and MR actions, there is an error in scanning data. Therefore, it is necessary to filter the raw data to first eliminate all unreliable data and isolated data from the scan results. We use the median filtering method to remove noise and improve the efficiency of the algorithm. The basic principle of the median filter is to replace the point with a median near the point:

1) Median Filtering

Consider the input vector $X=\{X_{-N},\cdots,X_{-1},\cdots,X_N\}$ and the non-negative integer weight, W0. Then, the output Y of the center weighted median filter is given as

$$Y=\{X_{-N},\cdots,X_{-1},W_0\diamond X_0,\cdots,X_N\} \tag{9}$$

When the weight in (9) is extended to the positive real value, the output is calculated as follows. First, the elements of the input vector X is sorted in ascending order. In the sorted sequence, let the i^{th} largest element be $X_{(i)}$.

Then, the output is equal to:

$$X\left(\frac{2N+W_\theta}{2}+1\right) \tag{10}$$

where $X\left(\frac{2N+W_\theta}{2}+1\right)$ means the maximum integer which does not exceed.

2) Mean Shift Clustering Algorithm

$$\hat{f}_h,K(x)=\frac{c_{k,d}}{nh^d}\sum_{i=1}^{n}k\left(\frac{\|x-x_i\|^2}{h}\right) \tag{11}$$

where $k(x)$ is the profile of kernel K so that $K(X)=c_{k,d}k\left(\|x\|^2\right)$; where $c_{k,d}$ is a normalization constant. When the derivative of $k(x)$ exits, $g(x)=-k_0(x)$ can be used as a profile to define a new kernel $G(x)=c_{g,d}g\left(\|x\|^2\right)$ with normalization constant $c_{g,d}$.

Take the gradient of (11), the following equation can be obtained,

$$m_{h,G}(x)=C\frac{\hat{\nabla}f_{h,k}(x)}{\hat{f}_{h,G}(x)} \tag{12}$$

Since $C = 1/2h2c$, then

$$m_{h,G}(x) = \frac{\sum_{i=1}^{n} x_i g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)} - x \tag{13}$$

Therefore, the mean shift vector computed with kernel G at location x is

$$x^{(k+1)} = x^k + m_{h,G}\left(x^k\right), k = 1, 2, \cdots \tag{14}$$

3.4.3. Obstacle Recognition

The measurement range of LIDAR is divided into (as shown in **Figure 10(a)**) emergency obstacle-avoidance layer and accurate obstacle-avoidance layer.

As shown in **Figure 10(b)**, d_k is set to return value of LIDAR in every vector direction, referring to the distance between an obstacle point and the AR in any vector direction. $d^{\max}$ represents the maximum effective distance in the scanning area. $\Delta d \approx \left|d_{k+1} - d_k\right|$ is the distance between two adjacent obstacles.

3.4.4. Complete Manual Control

When the MR is in the emergency obstacle avoidance layer, it needs to be quickly and simultaneously reflected to avoid obstacles as soon as possible. Therefore, the emergency obstacle avoidance algorithm should reduce the amount of calculation, increase the responding speed and efficiency, etc. If obstacles are detected within the emergency obstacle avoidance layer, the AR will move in the opposite direction of the obstacle until the unknown obstacle outside the emergency obstacle avoidance area. Accurate obstacle avoidance algorithms will be implemented quickly.

Set the two endpoints of expanded obstacles as $A(x_a, y_a)$ and $B(x_b, y_b)$. Applying the vector gradient algorithm, we can decide whether the line OG intersects with the line AB at first (as shown in **Figure 11**).

Line AB: $A'x + B'y + C' = 0$ (15)

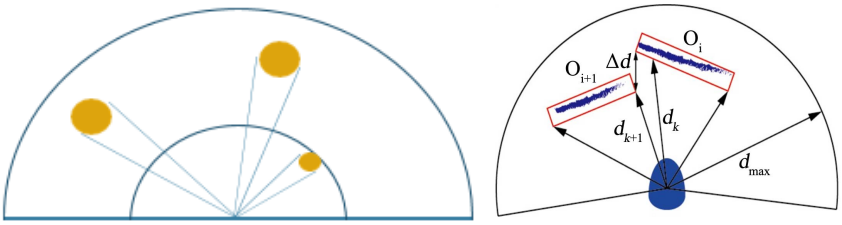
Line OG: $A''x + B''y + C'' = 0$ (16)

1) The distance between point A and line OG is

$d_a = \left(A'x_a + B'y_a + C'\right) \Big/ \sqrt{A'^2 + B'^2}$.

2) The distance between point B and line OG is

$d_b = \left(A''x_b + B''y_b + C''\right) \Big/ \sqrt{A''^2 + B''^2}$, AB intersects OG.



(a) (b)

Figure 10. The outer layer is accurate layer and the inside is emergency layer.

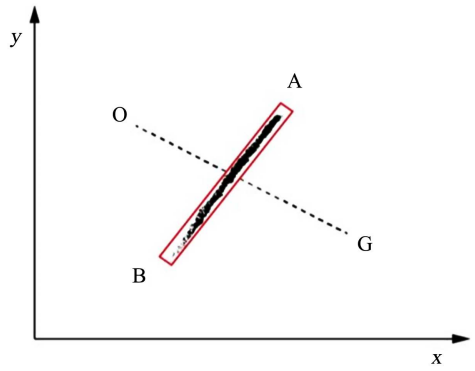


Figure 11. The line OG intersects with the line AB.

If $d_a \cdot d_b \leq 0$, AB intersects OG.

If $d_a \cdot d_b > 0$, AB doesn't intersect with OG.

If there is no obstacle intersected with expected trajectory, the output point is the current position of the AR, and AR can move along the trajectory as planned;

If there are obstacles intersected with expected trajectory, the distance between the feature points of the current obstacle and the goal can be calculated. The point of shorter distance can be selected as the next visible point of MR. MR can repeat the above procedure until reaching the target.

4. Different Weather Conditions

A road detection method, called dimensionality reduction deep belief neural network (DRDBNN), is proposed for drivable road detection. Due to the dimensionality reduction ability of the DRDBNN, it detects the drivable road area in a short time for controlling the robot in real-time. A feed-forward neural network is used to control the robot for the boundary following navigation using evolved neural controller (ENC). The robot detects road junction area and navigates throughout the road, except in road junction, using calibrated camera and ENC. In road junction, it takes turning decision using Google Maps data, thus reaching the destination.

4.1. Emergency Kill Switch

Safety is considered as a higher priority task for any system. There can be situations may arise, where the power to the BOT needs to be completely shut down. There is a dedicated physical button, placed secretly, for cutting off the power to motors in order to bring the BOT to complete stop, if any malfunction is noticed.

4.2. QR Code Unlocked Delivery Cart

To ensure the delivery will not be mistaken, the storage trunk of AR will be locked after the freight is put in. When AR arrived at the appointed location, customer needs to scan the QR code on its LED screen installed on top to unlock

the trunk via mobile app.

4.3. Customized Map for Simpler Usage

The mobile app is built using a custom map of the area where the BOT is to be delivered. The map is customized on Google Maps. Customized waypoints are placed, highlighting relevant destinations and marking relevant points for better user interactivity.

5. Presentation Drawing (Draft)

See Figure 12.

6. Budget

See Table 1.

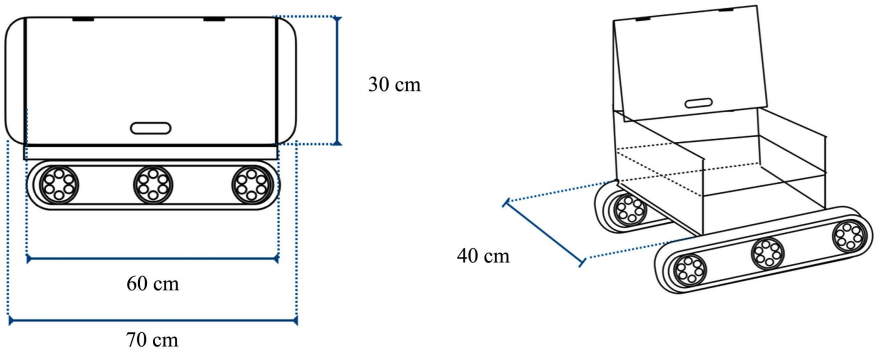


Figure 12. The model virtual map.

Table 1. The cost of an AR unit (estimated roughly).

Description	Quantity	Unit Price	Cost
Motor holder	6	US$ 50	US$ 300
Motor driver board	1	US$ 90	US$ 90
Crawler type motor	2	US$ 50	US$ 100
Battery	10	US$ 30	US$ 300
Dupont line	1	US$ 10	US$ 10
Raspberry pi 3B+	1	US$ 35	US$ 35
Arduino	1	US$ 70	US$ 70
Camera	9	US$ 120	US$ 1080
Livox Mid-40/Mid-100	1	US$ 599	US$ 599
LED screen	2	US$ 20	US$ 40
Magnetometer	1	US$ 30	US$ 30
GPS modules	1	US$ 20	US$ 20
MUVision Sensor 3	1	US$ 100	US$ 100
Total		US$ 2774	

7. Conclusion

The surge of online food order creates a great demand for high-efficiency within-in-community delivery. Therefore, we develop a supervised-autonomous robot cart for within-community delivery, by applying path-planning and obstacle avoidance algorithms and live streaming video manual control with the cost of 2774 dollars. Combined with the intelligent algorithms, AR can be a great business plan for high-efficiency-within-community delivery.

Acknowledgements

Thanks to Professor for giving great support to our work. The intelligent algorithm learning does help us a lot.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Vundurthy, P.B. and Sridharan, K. (2019) Protecting an Autonomous Delivery Agent against a Vision-Guided Adversary: Algorithms and Experimental Results. *IEEE Transactions on Industrial Informatics*, **16**, 5667-5679. https://doi.org/10.1109/TII.2019.2958818

[2] Kornatowski, P.M., Feroskhan, M., Stewart, W.J., *et al.* (2020) Downside Up: Rethinking Parcel Position for Aerial Delivery. *IEEE Robotics and Automation Letters*, **5**, 4297-4304. https://doi.org/10.1109/LRA.2020.2993768

[3] Dhariwal, A., Sukhatme, G.S. and Requicha, A.A.G. (2004) Bacterium-Inspired Robots for Environmental Monitoring. *Proceedings of the IEEE International Conference on Robotics & Automation*, New Orleans, LA, 26 April-1 May 2004, 1496-1443. https://doi.org/10.1109/ROBOT.2004.1308026

[4] Du, X. and Tan, K.K. (2016) Comprehensive and Practical Vision System for Self-Driving Vehicle Lane-Level Localization. *IEEE Transactions on Image Processing*, **25**, 2075-2088. https://doi.org/10.1109/TIP.2016.2539683

[5] Kümmerle, R., *et al.* (2015) Autonomous Robot Navigation in Highly Populated Pedestrian Zones. *Journal of Field Robotics*, **32**, 565-589. https://doi.org/10.1002/rob.21534

[6] Liang, J.-H. and Lee, C.-H. (2015) Efficient Collision-Free Path-Planning of Multiple Mobile Robots System Using Efficient Artificial Bee Colony Algorithm. *Advances in Engineering Software*, **79**, 47-56. https://doi.org/10.1016/j.advengsoft.2014.09.006

[7] Elsobeiey, M. (2015) Precise Point Positioning Using Triple-Frequency GPS Measurements. *The Journal of Navigation*, **68**, 480-492. https://doi.org/10.1017/S0373463314000824

[8] Prabhu, S.S., *et al.* (2018) GPS Controlled Autonomous Bot for Unmanned Delivery. 2018 *IEEE International Conference on Recent Trends in Electrical, Control and Communication* (*RTECC*), Malaysia, 20-22 March 2018, 128-132. https://doi.org/10.1109/RTECC.2018.8625677

[9] Fang, C. and Zhao, J. (2010) New Dynamic Obstacle Avoidance Algorithm with

Hybrid Index Based on Gradient Projection Method. *Journal of Mechanical Engineering*, **46**, 30-37. https://doi.org/10.3901/JME.2010.19.030

[10] Pil, H., Jeonghyun, B., Baehoon, C. and Euntai, K. (2015) A Novel Part-Based Approach to Mean-Shift Algorithm for Visual Tracking. *International Journal of Control, Automation and Systems*, **13**, 443-453. https://doi.org/10.1007/s12555-013-0483-0

[11] Wang, Y., *et al.* (2017) Obstacle Avoidance Strategy and Implementation for Unmanned Ground Vehicle Using LIDAR. *SAE International Journal of Commercial Vehicles*, **10**, 50-55. https://doi.org/10.4271/2017-01-0118

[12] Wu, P., *et al.* (2015) A Novel Algorithm of Autonomous Obstacle-Avoidance for Mobile Robot Based on LIDAR Data. 2015 *IEEE International Conference on Robotics and Biomimetics* (*ROBIO*), Zhuhai, 6-9 December 2015, 2377-2382. https://doi.org/10.1109/ROBIO.2015.7419694