DOI: 10.54254/2753-8818/18/20230390

Automatic dining car system based on A* algorithm

Sirui Li[1,3,+], **Senjie Yang**[2,+]

[1]School of Advanced Technology, Xi'an Jiaotong Liverpool University, Suzhou,215028, China
[2]School of Engineering, University of Glasgow, Glasgow, G128QQ, United Kingdom

[3]17625739025@163.com
+These authors contributed equally to this work and should be considered co-first authors.

Abstract. To solve the problem of selling dining and getting the most profit in a certain time by two devices, this project is divided into two parts, one is the detection and the other is route planning, these two parts are combined to find the nearest suitable selling position and plan the optimal path from the current position to that point. Firstly, achieve the function of detecting the center points of the densely populated areas based on the work of tracing the areas' contours and the center point of the object by OpenCV, then select the nearest point. Then find out whether the point is on the road, if yes, the point will be the endpoint, if not, use BFS (Breadth-first search) to find the nearest point of the selected point which is on the road. Finally, use the A* algorithm to plan an optimal route between the current position of the car and the endpoint. Furthermore, the heuristic function is used to optimize the algorithm to achieve maximum profit.

Keywords: mobile robot, OpenCV image processing, path optimization, A* algorithm.

1. Introduction

In the recent few years as the epidemic spreads around the world, the technology of automatic cars has been widely used in many areas which can deliver or sell a product without any contact with others. It is mainly to plan an optimal path from the initial position to the certain target position to deliver products in the shortest time [1]. In this project, detection and route planning are two essential parts. When achieving the detection part mainly use the computer vision by OpenCV, which is to find the red areas on thermodynamic diagrams of the map. Tracing the contours of the red areas that are densely populated, and find the center point of each area, choose the nearest one towards the start warehouse. To achieve the route planning part, clearing the start point and endpoint is the most basic condition. Thus, the system will judge whether the endpoint is on the road or not by BFS before planning the route and confirming the final endpoint. After ensuring the endpoint will plan a route between the initial warehouse and the endpoint, the A* algorithm is chosen to achieve the planning which is much more efficient than other methods. Moreover, there are some ways to optimize the A* algorithm, the bidirectional A* algorithm and heuristic algorithm are tried to get the route in the optimal period, and the heuristic algorithm has better performance. All these functions are visualized by Tkinter and run and show the result on a vector map. The article will achieve the optimal time and profit delivery system based on the technologies above.

DOI: 10.54254/2753-8818/18/20230390

2. Method

2.1. Overview of the project

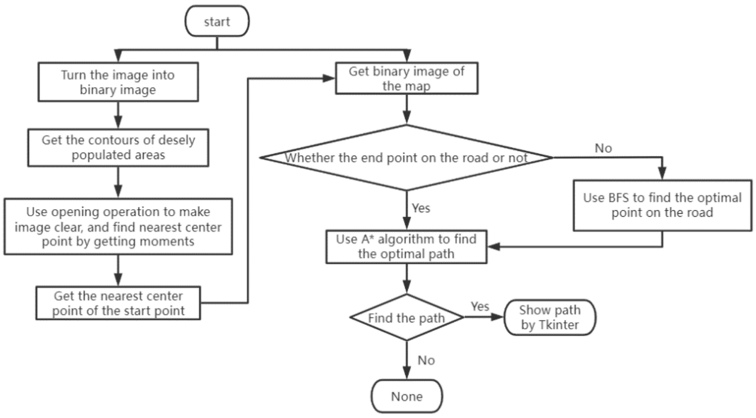

Figure 1. The flow diagram of the whole project.

Figure 1 is the flow diagram of the whole project.

2.2. Detection

2.2.1. Get contours of red areas

In the project, the first step is to get the contours of the red areas on the thermodynamic diagram which means these areas are densely populated so that products can be sold in the most crowded place to achieve maximum profit. This function will use the OpenCV library to firstly turn the image into a binary image and recognize the red areas and let the rest of the pixels do the same. Then get the contour image with lots of noisy points, the result needs to be operated by the opening operation later to make it clearer [2]. The final result is shown in Figure 2.

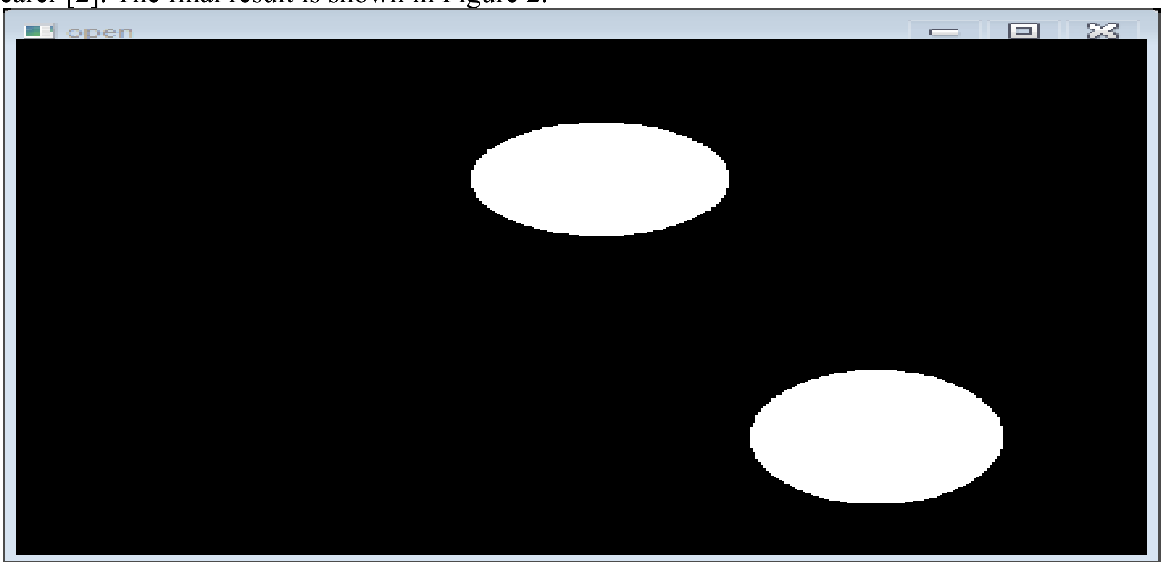

Figure 2. The image of contours after disposed of.

2.2.2. Get the nearest center point

After that still use OpenCV to get the moment of the contours and calculate the center point of each moment, the center points of all the crowded areas can be get [3].

The next step is to calculate the Euclidean distance from each center point to the start point, the formula is:

$$x = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

and find the point which is the nearest far from the start end and write the coordinate (620,240) on the map. The coordination in Figure 3 is the nearest point the detection part got.

Finally, pass the parameter of the coordinate of the nearest center point to the Route Planning part as the end point in the algorithm.

Figure 3. The coordinate of the center point.

2.3. *Route planning*

Firstly, get a vector image of the map and turned the image into a binary map to make it easier for the algorithm to run.

2.3.1. *BFS*

Receiving the coordinate of the endpoint, we must judge whether the endpoint is on the road or not. If the point is on the road this point will still be the endpoint, but if the point is not on the road the project will use BFS (Breadth First Searching) to find the nearest point on the road to find the optimal path, and the car will recognize this point as the endpoint [4].

To achieve the BFS algorithm, let the endpoint be the center point of the BFS and search the point around the endpoint layer by layer until the road is found, and this point on the road will turn into the new endpoint and be sent to the A* method part. The text in Figure 4 is the result of BFS, it displayed that "the nearest point on the road of the endpoint (240,620) is (240,621)"

```
the nearest point on the road of the end point (240,620) is (240,621)
```

Figure 4. The result of BFS.

2.3.2. *A* algorithm and optimization of it*

The A* algorithm is used to find the right path in this part, a heuristic method that can find the shortest path quickly by set a weight to parameters [5].

Firstly build a tree, so that the root can be traced back when looking for the aim point. Next search the neighborhood near the minF point and calculate the cost f, then push the neighborhood into Openlist[6].

Finally, comparing whole points in Openlist and finding the point that possesses minimum F, then let the point become a new minF point and search its neighborhood. Repeat steps 2 and 3 until the target point is found.

$$F = K_g \times G + K_h \times H \tag{2}$$

Use the K_g and K_h to improve the performance of the algorithm. If we increase the parameter of K_h , the searches tend to expand in the end and would be faster. However, its outcome would be longer than other A* methods [7]. In contrast, if the kg parameter is increased, the search tends to expand at the beginning. It is possible to consume more time to search and output a shorter path.

2.4. Visualization by Tkinter

Use OpenCV to calculate the data and use one of its functions to show how the computer finds the outcome. The depict below are steps that how to visualize the path.

Firstly, copy the road map as ShowMap to protect the original road map and create a black picture as TraceMap. Because the roads' width in the road map only single pixel length, we need to use the Mathematical morphology to widen the road to be seen obviously, hence we use the erode algorithm to widen the road. Erosion is the morphological dual to dilation. It is the morphological transformation which combines two sets using the vector subtraction of set elements [8].

Next, when getting a point and push it into OpenList, the point in the TraceMap will be labeled into white.

Then, since the picture is too large to display on the screen, resize the image to the correct size. Before resizing the photo, the line in TraceMap will be widen to see the path more clearly. The Erosion is chosen in morphology algorithm, which can expand the graph without changing shape.

Finally, combine the TraceMap and RoadMap into a picture, then resize it to the correct scale and show it on the screen.

3. Result

3.1. Input

There are two pictures we need to input into the algorithm to calculate the result and a graph to show the result on the screen.

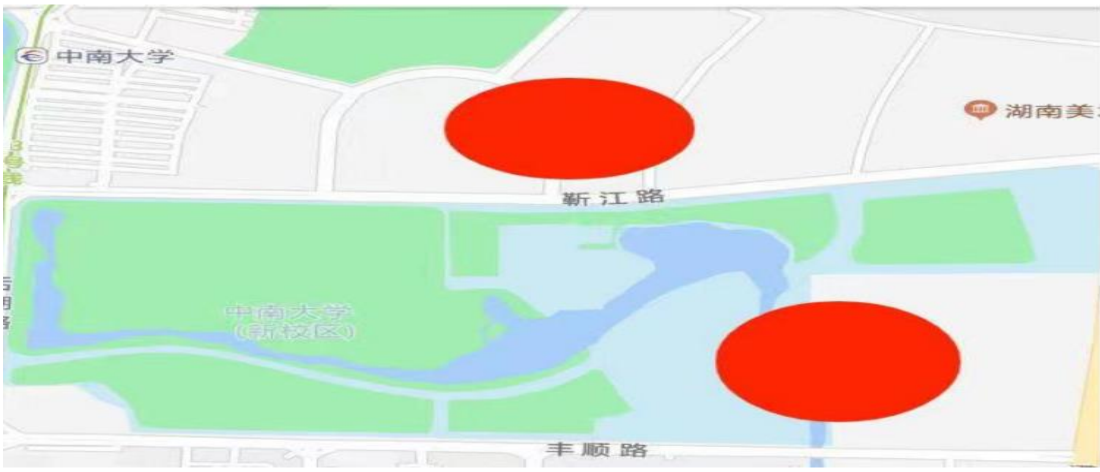

Figure 5. The income of heat map of population density.

Figure 5 shows the population density. Using the specific algorithm can calculate the location of the center point. A thermodynamic diagram interface can also be used to get other data as input.

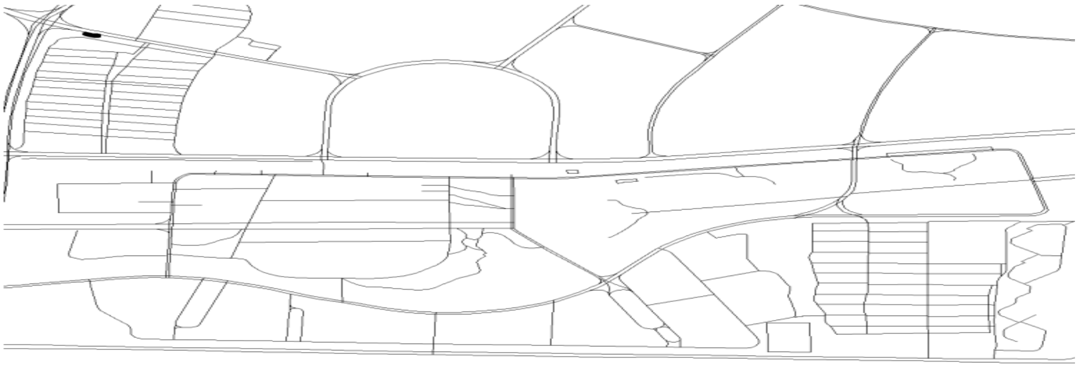

Figure 6. The input of RoadMap [9].

Figure 6 is the picture that is the most important among these images. The data are used to identify where the road is, where the vehicle can access it, and where it is inaccessible. Download its original vector map on the website OpenStreetMap. Then transfer it into PNG.

Figure 5 and Figure 6 are used to calculate the center point and the road, and the following part is to show the result more clearly.

Figure 7. The input of Navigation Map [10].

Figure 7 is the layer that helps the viewer visualize the path on the real map, making the outcome more clearly. The map can be get from some GIS websites.

3.2. *Outcome*

Figure 8. The result of the program.

The picture shows the outcome of the project and how the vehicle organizes its routine in the real world.

The blue line in Figure 8 is the path calculated by this system, the black lines are the road map, and the background is the navigation map.

Figure 9. The result of the Time Cost.

Figure 9 shows the time cost of the algorithm used in this project and initial Road's image resolution is 1201 × 886 imply that it is enough to apply to the real-world problem.

4. Conclusion

Overall, this project has solved the problem that using two devices to get maximum profit in a certain time without any contact. It combined the detection part and the route planning part. However, there are some functions that should be added to the project but failed to finish at this stage. To be specific, schedule the battery uses. If the battery is insufficient for reaching the next point, the program will choose the nearest warehouse to recharge the vehicle's battery. Furthermore, to detect the right way the car could access in the real world, for example, to plan the road when the street prohibits vehicle steering or when it is a one-way street. There are still many jobs that we could do to push the program forward.

Acknowledgement

Sirui Li and Senjie Yang contributed equally to this work and should be considered co-first authors.

References

[1] Jing.Z.(2020) Arriving of the food truck.J.Auto Observation, 03:117.
[2] Shuai.G, (2012). Image denoising based on mathematical morphology. D, TP391.41
[3] Bin,H, Wenzhi. Z, Yuancheng.D, Hongqiang.T,(2018) Method of objects'color and contour recognition based on OpenCV. Modern Electronics Technique. [J].37:96-97
[4] E. Žunić, A. Djedović and B. Žunić, (2016) Software solution for optimal planning of sales persons work based on Depth-First Search and Breadth-First Search algorithms. Electronics and Microelectronics (MIPRO), pp. 1248-1253
[5] Mingliang.Y, Ning L, (2022) Study on Mobile Robot Path Planning based on Improved A* Algorithm. Mechanical Science and Technology for Aerospace Engineering. J. 41: 795-800
[6] Daokui.Q, Zhenjun.D, Dianguo.X, (2008) Research on Path Planning for a Mobile Robot, Robot, J. 30(2):97-101
[7] Xiao.Z, Zheng.W, Chengkan.H, Yanwei.Z, (2018) Mobile Robot Path Planning Based on an Improved A* Algorithm. ROBOT. 40
[8] R. M. Haralick, S. R. Sternberg and X. Zhuang, "Image Analysis Using Mathematical Morphology," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 4, pp. 532-550, July 1987, doi: 10.1109/TPAMI.1987.4767941.
[9] OpenStreetMap[EB]/[OL]https://master.apis.dev.openstreetmap.org/#map=15/28.1504/112.9427 10/07/2022
[10] BaiduMaps[EB]/[OL]https://map.baidu.com/@12573521.69543578,3248407.054473594,17.06z 10/07/2022