The Intelligent Robot Path Planning and Navigation System Based on Openlayers3 Online Map

Liu Yang\textsuperscript{1,2}, Limin Yu\textsuperscript{1,4}, Sha Tao\textsuperscript{1,2}, Zhenghong Yang\textsuperscript{1,2}, Wanlin Gao\textsuperscript{1,2} and Yanzhao Ren\textsuperscript{2,3,*}

\textsuperscript{1} College of Information and Electrical Engineering, China Agricultural University, Beijing, China
\textsuperscript{2} Key Laboratory of Agricultural Informatization Standardization, Ministry of Agriculture and Rural Affairs, Beijing, China
\textsuperscript{3} College of Science, China Agricultural University, Beijing, China
\textsuperscript{4} Agriculture and Engineering University, Shandong, China
\textsuperscript{*} Email: xiaozhaochina@163.com

Abstract. The electronic map becomes more and more popular due to its convenience and powerful functions. This paper aims to use map information to build a system for path planning and navigation of an intelligent robot. It can provide a basis for field autonomous identification of crop diseases and insect pests with the robot as the carrier. The open source Openlayers is used as a tool to link online maps. And Dijkstra algorithm is used to obtain the shortest path. And then it can simulate a robot moving along the path by creating a moving point. In addition, it could accomplish a series of fundamental operations of controlling movement and the display of route details. The feasibility of the functions is verified by experiments and it lays a solid foundation for subsequent work.

Keywords. Map; openlayers; Dijkstra algorithm; robot; recognition.

1. Introduction
Today, with the rapid development of the Internet, people have higher and higher requirements for information. Electronic map has become an indispensable tool in people’s daily life [1]. Route query, route display, route detailed display and simulated navigation play an important role in daily travel life. Therefore, WebGIS comes into being. And path planning based on Openlayers not only greatly saves the cost and reduces the development difficulty, but also meets the basic functional requirements and brings better user’s experience.

China is among the countries with the earliest agricultural development in the world. Crop pests and diseases have become a major problem disrupting farmers. Different from the traditional method of detecting diseases, the method of using a robot as a carrier to autonomously detect the situation of crops (especially in the early stage of disease onset) can save humans from themselves.

2. Openlayers
The WebGIS system based on Openlayers as the client architecture is divided into three basic layers: display layer, service layer and data layer. At present, the domestic WebGIS development platform can be summarized into three parts. The first is ESRI ArcGIS system. The second is Openlayers (currently ol3), which has strong expansibility. The third is the online API based on ol or leaflet packaging [2]. As a lightweight open source WebGIS development framework, Openlayers has the characteristics of low
cost and simple development [3].

Openlayers3 is an open source WebGIS engine, which uses JavaScript, the latest HTML5 technology and CSS technology. In addition to supporting PC web side, it also supports mobile side [4].

Openlayers3 is a particularly popular version of Openlayers that supports multiple types of data sources. It includes free and commercial map tile services, such as OpenStreetMap, Bing, MapBox, Stamen, and any other XYZ source tile layer [5-7].

3. Creation an Environment Map
Using Openlayers3 to implement the path planning can be divided into three parts. First, create an environment map; second, realize the path planning of the robot; lastly, finish simulated navigation function. This paper uses online map as the environment map. When connecting online map, there must be a layer. A layer is a visual display of data in a resource.

Tile map is a commonly used layer type. The concept of slicing is to use a grid to cut a map into small squares of equal size. Now many maps use sliced maps, such as Baidu map, Amap, etc. So when you open the map, you will see that the map is displayed and loaded one by one when the network is not good enough. A request for a map image is presented in figure 1. Or through chrome's developer tools, you'll see requests for images in the network, as showed in figure 2.

Since the tile layer requests the image in a regular grid, it could allow the server to cache the image requests and return the cached results the next time the same region is visited. Therefor better performance can be realized. The map block reading function provided by Openlayers greatly improves the loading speed of maps [8].
4. Path Planning of the Intelligent Mobile Robot

4.1. Path Planning Function
Robot path planning is mainly composed of two parts. One is to generate the environment map, and the other is to determine the best path. In the actual operation of the intelligent mobile robot, a curve connecting the starting point and the terminal point by a certain strategy is called path planning. Path planning is a significant robotics technology. Good robot path planning technology can save working time and improve working efficiency; thus it can save human resources and reduce capital investment [9]. Taking China agricultural university as the experimental object, taking each intersection as the node and combining with Dijkstra algorithm, we can plan a shortest path from the starting point to the end point.

4.2. Dijkstra Algorithm
Dijkstra is an indispensable algorithm in robot path planning. Dijkstra algorithm is a typical single-source shortest path algorithm, which is utilized to calculate the shortest path from one node to all other nodes. The main feature is to extend outward from the starting point until it reaches the end point. Its time complexity is O(n^2).

Let G= (V, E) be a weighted directed graph, and divide the vertex set V into two groups. The first group is the vertex set that has solved the shortest path (represented by S), and the second group is the vertex set that has not determined the shortest path (represented by U). The exact process is as follows.

(1) The weight between every two points is expressed in the matrix, and a_ij is the weight between point i and j. An n by n matrix A is established.

\[
A = \begin{bmatrix}
a_{11} & \cdots & a_{1n} \\
\vdots & \ddots & \vdots \\
a_{n1} & \cdots & a_{nn}
\end{bmatrix}
\]

Among them, a_{ii}(i=j)=∞.

(2) By comparing the first column in A, the minimum value of the first column is a_{1k}, which is the weight between k point and the starting point and it is the minimum value among any other connected with the starting point.

(3) Then taking k as the starting point. It’s time to obtain the minimum value between k point and other points connected with k. After finding the point m, it will be outputted and matrix B can be got.

(4) Then taking m as the starting point, iterate according to the third step until the route passes through all points. And then, the shortest path is obtained [10].

Dijkstra algorithm is mainly used to calculate the shortest path between two known points. The computer autonomously plans the best path for the robot, making the operation simpler and practical.

4.3. Establishment of Database
Due to make full use of weight information, Dijkstra algorithm can calculate the shortest path between two given points. Taking China Agricultural University as the experimental object and each building object as the node, the coordinate points obtained are stored in the database. The weight is initialized according to whether there is a path between two nodes and the actual distance between nodes. Then Dijkstra algorithm completes the path planning of the robot by obtaining operational information from the database.

5. Simulation Navigation Function Realization
Once having the correct path, we can simulate the navigation of the robot (The robot is abstracted as a dot ignoring its size). Robot navigation is the process of monitoring and controlling the movement of a robot from one place to another. The navigation function is actually subdividing the path by some rule, as showed in figure 3. The robot’s position changes rapidly and continuously along the divided points. It looks like moving along a trajectory visually. The more points subdivided, the smoother the simulation will be.
When the robot starts to move, the position of the robot keeps moving along the subdivided dot, and the color of its path also changes, as showed in figure 4.

![Figure 3. Efficient partition of path information.](image)

![Figure 4. Simulate the robot motion with a moving point.](image)

6. Experience

6.1. Experiment 1: Correct Display of Path Information

Before connecting to the database, in order to test whether the nodes and paths can be displayed correctly by using Openlayers3, a path is randomly selected and each turning point of the path is taken as a node to observe whether the desired path can be obtained. The coordinates of each node of the path are showed in table 1 below (different accuracy may lead to errors):

| Table 1. Node coordinates. |
|----------------------------|
| Starting point | [116.3551,40.0018] |
| node 1         | [116.3550,40.0034] |
| node 2         | [116.3549,40.0035] |
| node 3         | [116.3519,40.0034] |
| node 4         | [116.3518,40.0052] |
| ending point   | [116.3506,40.00524] |

Through testing, it can be seen that openlayers3 can visually display the desired effect, as showed in figure 5.

![Figure 5. Display of path details.](image)
6.2. Experiment 2: Dijkstra Algorithm Realizes Shortest Path Planning

The purpose of using Dijkstra algorithm is to make the computer autonomously calculate a shortest path from the starting point to the terminal point rather than artificial choose. It can make the operation smoother and simple. By creating weights between the reachable nodes in the database, the computer can plan a shortest path.

In order to test whether Dijkstra algorithm can run correctly, taking experiment 1 as an example, the weight between every two adjacent nodes in the trajectory of figure 5 is stored in the database, and a small weight value is assigned. In addition, other nodes should be randomly selected and given a large weight value. After running, it is found that Dijkstra algorithm can achieve the same effect as experiment 1.

7. Conclusion

In addition to helping developers achieve the basic effects of map exploring (zoom in, pan out, zoom out, etc.) in the browser, Openlayers can even expand the existing operations and the supportive types of data to give them more functions.

Openlayers not only implements map data access in standard format, but also is able to integrate with other development methods easily. It enables Openlayers to run normally on current mainstream browsers without redeploying the new development environment. Its features of not relying on third-party API make Openlayers convenient and efficient for use.

References

[1] Farkas G 2017 Applicability of open-source web mapping libraries for building massive web GIS clients Journal of Geographical Systems 19 (4) 273-295.
[2] Wang Q, Lei Z and Zhang R 2017 Path planning of 2.5D map based on openlayers achieve Geomatics & Spatial Information Technology 40 (5) 97-100.
[3] Zhang W 2015 An implementation scheme of GIS graphics system based on openlayers Industry and Mine Automation 41 (9) 71-3.
[4] Huang C, Lei W and Yang C 2017 Ship collision risk identification system based on openlayers3 and Cassandra Computer Applications and Software 34 (9) 9-13.
[5] Lu Y 2018 A web publishing solution of geospatial information based on MapServer and OpenLayers Bulletin of Surveying and Mapping 495 (06) 137-141.
[6] Yang P and Zou S 2012 The research and development of WebGIS client based on OpenLayers Geomatics & Spatial Information Technology 35 (3) 131-3.
[7] Shen J 2016 Implementation of WebGIS based on ExtJS and OpenLayers China Computer & Communication (9) 56-7.
[8] He X 2017 Design and implementation of WebGIS geographic information service system based on GeoServer and OpenLayers ShanXi Science & Technology of Communications (04) 126-130.
[9] Du J, Zhang B, Chen H, Ren Q and Sun H 2018 Campus delivery scheme based on Dijkstra algorithm E-Commerce Letters 7 (3) 38-46.
[10] Jiang Y, Su H and Lu L 2015 Establishment and application of emergency logistics decision support platform based on OpenLayers Logistics Technology 34 (2) 275-277.