

A Set of Tracking Car Scheduling Management System

Minghao Tan, Yubin Zheng, Weihua Li

Department of Electrical Engineering and Automation, Jinan University, Zhuhai, China

Email: tanminghao@stu2018.jnu.edu.cn

How to cite this paper: Tan, M.H., Zheng, Y.B. and Li, W.H (2021) A Set of Tracking Car Scheduling Management System. *Journal of Transportation Technologies*, 11, 660-668.

<https://doi.org/10.4236/jtts.2021.114041>

Received: August 13, 2021

Accepted: September 19, 2021

Published: September 22, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

With the rapid improvement of China's economic level and the rapid development of e-commerce, the demand for logistics warehousing, which is one of the most important links in the logistics transportation system, has also greatly increased. The dispatching management system applied in the warehouse has a wide range of applications in the logistics storage system. Improving the operational efficiency of the dispatching management system can effectively improve the efficiency of the automated warehouse system and reduce the cost of logistics transportation. Therefore, this paper designs a set of dispatching systems with the tracking car as the control target and realizes the positioning and path planning functions of the tracking car in the system. At present, the tracking car can completely receive the commands sent by the system, accurately follow the established path. At the same time, the system software can also obtain the video images taken by the camera on the car and play them smoothly.

Keywords

Scheduling System, Tracking Car, Indoor Positioning

1. Introduction

According to the data released by China Federation of Logistics and Purchasing, the total amount of social logistics in China was 298.0 trillion yuan in 2019 and 300.1 trillion yuan in 2020 [1]. Warehousing is one of the most important links in the logistics transportation system. China's logistics costs of warehousing costs are relatively high, accounting for about 30% of the total logistics costs. We can reduce the cost of warehousing to improve the transportation efficiency of social logistics, and at the same time, logistics enterprises can gain an advantage in the fierce market competition. But at present, the small and medium-sized enterprise warehouse management has a low degree of information, storage

management lack of flexibility and other problems, Therefore this design is based on the actual situation that the current storage equipment is relatively backward. Aiming at the inefficient traditional warehouse management method, we take the automatic tracking and obstacle avoidance trolley as the control object and combine the Internet of Things technology and indoor wireless positioning technology to design and produce a set of dispatch control systems. Warehouse scheduling management system has a wide range of applications in logistics warehousing systems. The improvement of its system efficiency is of great significance for improving the efficiency and accuracy of automated warehouse systems and reducing the total cost of logistics transportation [2].

2. Overall System Architecture

The overall architecture of the warehouse dispatching management system is shown in **Figure 1**. The system consists of a tracking car, wireless positioning system, and dispatching system software. The tracking car uses the Raspberry Pi as the main controller, and the infrared tracking module and the motor drive module are controlled by the Raspberry Pi control program to realize the tracking of the car. The wireless positioning system is composed of a positioning base station, a positioning server, and a positioning tag loaded on the car to realize the real-time positioning of the car. The dispatching system software makes use of A* algorithm to plan the path in the drawn warehouse map. After the planning result is sent to Raspberry Pi, the car is controlled to drive along the designated path. After connecting to the same LAN with the wireless positioning system, the dispatching system software can present the location label, that is, the real-time coordinate position of the trolley on the map. After the Raspberry Pi is connected to a USB camera, the dispatching system software accesses the video captured by the camera by connecting to a video server.

3. Tracking Car Design

The tracking car uses Raspberry Pi 3B+ as the main controller. It features 1 G

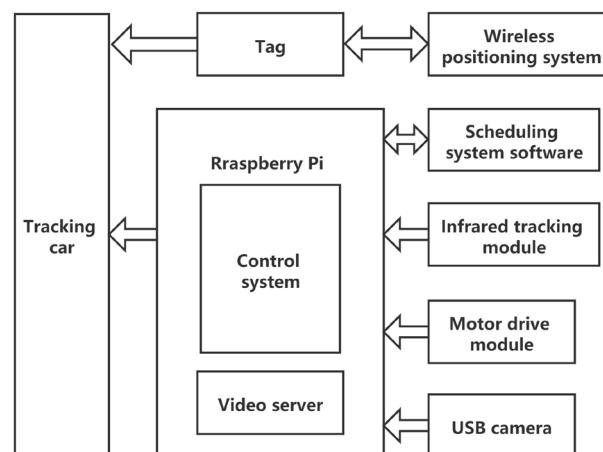


Figure 1. Overall structure of warehouse dispatching management system.

running memory, MicroSD as the system memory hard disk, integrated 2.4 GHz/5 GHz dual-band WiFi and Bluetooth 4.2, 4 USB ports and Gigabit Ethernet interface. It can meet the requirements of collecting sensor data, controlling motor, compressing and transmitting image, and establishing real-time communication with dispatching system software. Infrared tracking module uses reflective infrared sensor to transmit and receive the transmitted infrared, and determines whether there are obstacles in a certain distance ahead according to the time difference between transmitting and receiving [3]. The specific obstacle avoidance process of tracking is shown in **Figure 2**. The motor drive module uses double H-bridges, which can drive two DC motors at the same time. It uses photoelectric coupling to isolate the input signal and is equipped with undervoltage protection and electrostatic discharge circuit, which is anti-interference and stable and reliable. The camera uses USB camera based on UVC protocol standard, and the output image format is YUYV digital image format.

4. Wireless Location System

4.1. System Design

The wireless positioning system is composed of a positioning base station, a positioning server and a positioning tag carried on the car. First, use the server-side software to configure the wireless network card of the positioning base station to ensure that the server-side software and the positioning base station are connected to the same network. Then fix the base station in the four corners of the warehouse, and obtain the position of the positioning tag on the trolley through UWB technology and TOF positioning. After the location server and scheduling system software are connected to the same LAN, the real-time location is presented in the scheduling system software, and the location information is stored in the MySQL database.

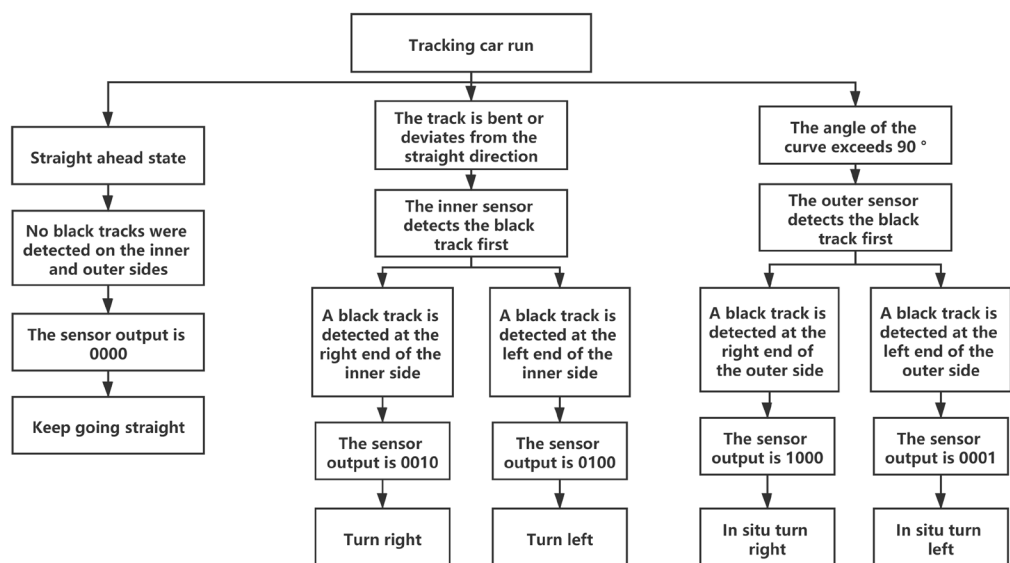


Figure 2. Tracking obstacle avoidance process.

4.2. Positioning Principles

Figure 3 shows the schematic diagram of calculating the label position with TOF positioning method. The calculation steps are as follows [4]:

- 1) Gets the distance between the tag and the base station.

$$D = C \times (T_2 - T_1)$$

where D is the distance between the label and the base station, C is the speed of the pulse, that is, the speed of light, T_2 is the time of receiving the pulse radio wave, T_1 is the time of sending the pulse radio wave.

- 2) The four base station coordinates $(x_i, y_i, z_i) (i=1,2,3,4)$ are used as the center of the circle, and the distance $d_i (i=1,2,3,4)$ is used as the radius. The tag position $l_{Tag}(x, y, z)$ is the coordinate of the intersection of the four circles.

The following ternary quadratic equation can be obtained:

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2 = d_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2 = d_2^2 \\ (x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2 = d_3^2 \\ (x_4 - x)^2 + (y_4 - y)^2 + (z_4 - z)^2 = d_4^2 \end{cases}$$

- 3) The equation is transformed and simplified to:

$$A l_{Tag} = B$$

Among them:

$$A = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ x_4 - x_1 & y_4 - y_1 & z_4 - z_1 \end{bmatrix}, \quad l_{Tag} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$B = \frac{1}{2} \begin{bmatrix} (x_2^2 + y_2^2 + z_2^2) - (x_1^2 + y_1^2 + z_1^2) + d_1^2 - d_2^2 \\ (x_3^2 + y_3^2 + z_3^2) - (x_1^2 + y_1^2 + z_1^2) + d_1^2 - d_3^2 \\ (x_4^2 + y_4^2 + z_4^2) - (x_1^2 + y_1^2 + z_1^2) + d_1^2 - d_4^2 \end{bmatrix}$$

- 4) Get the position coordinates of the positioning tag:

$$l_{Tag} = A^{-1} B$$

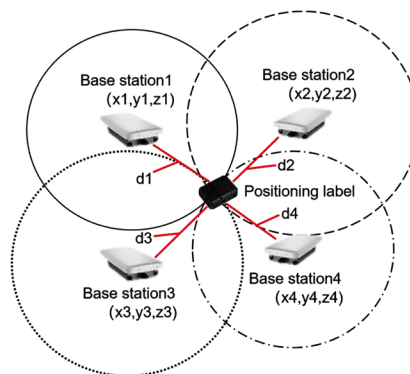


Figure 3. Schematic diagram of calculating the label position.

4.3. Location Test

The vehicle positioning function test is as follows. Before using the positioning function, you need to activate the Qingyanxun wireless positioning system. After the Qingyanxun wireless positioning system is activated, the positioning server will be activated, and the positioning information obtained by the positioning system will be uploaded to the positioning system server. Then start the dispatch management system, the system will automatically connect to the wireless positioning system when it starts, and then click the system map button to enter the map interface. The test result is shown in **Figure 4**. The black dots in the figure are map nodes and the blue lines are map routes. The green dot indicates the positioning point of the trolley, and its coordinates are displayed below it.

5. Scheduling System Software

5.1. Software Design

The scheduling system software is a WinForm program written in C++/CLI language. First of all, we need to connect the Raspberry Pi, the wireless positioning system and the dispatching system software to the same local area network by setting the IP address to ensure normal communication between the various parts. Then the components in the form should be able to realize the real-time display of information such as the position and coordinates of the trolley, the load capacity and the battery power, the real-time playback of the images obtained by the camera, and the path planning of the trolley. The form interface design is shown in **Figure 5**.

5.2. Path Planning Algorithm

The drawn warehouse map is transformed into the grid map represented by

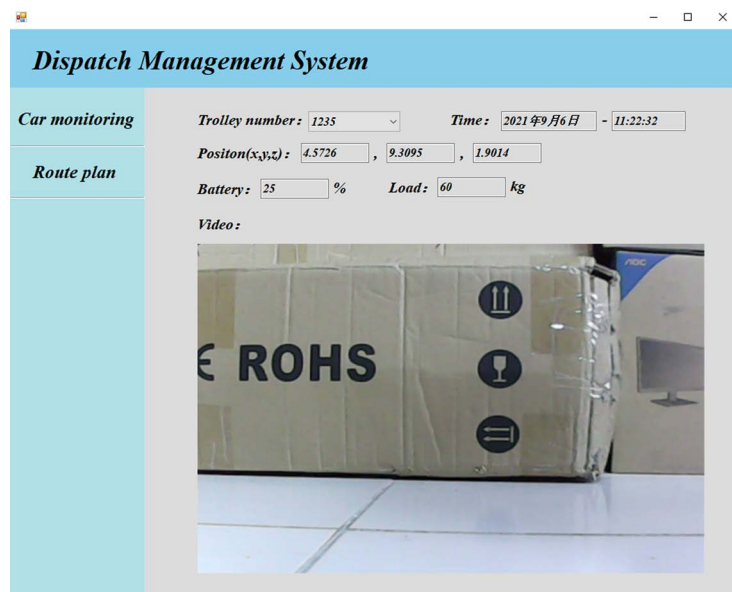


Figure 4. System positioning function test.

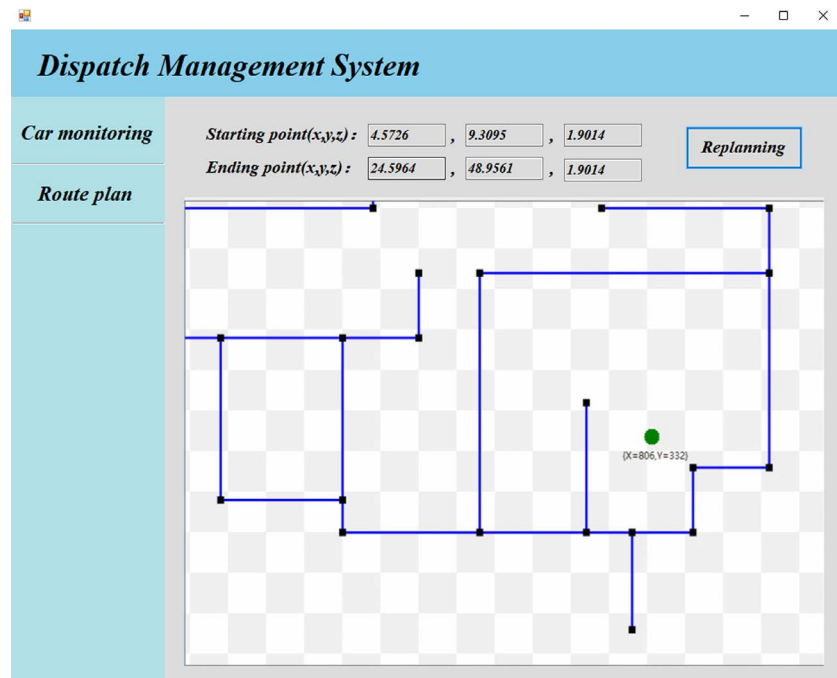


Figure 5. Dispatching system software interface.

points and lines. After the start and end points are determined on the path planning interface of the software, the path planning is carried out through A* algorithm. The algorithm steps are as follows [5]:

1) Create two tables, named Open List and Close List respectively. The Open List is composed of undiscovered nodes, arranged in ascending order according to the value of the evaluation function value F . $F = G + H$, where G is the actual path cost from the starting node to the current node, and H is the estimated path cost from the current node to the target node. The Close List consists of the nodes that have been searched.

2) Add the start point to the Open List.

3) Review Open List:

a) If the Open List is empty, the path planning fails; if it is not empty, take the node with the smallest F value as the current node to be processed, and then move it to the Close List;

b) If the current node is the end point, the path planning is successful, and go to step (5).

4) The 8 adjacent squares of the node are checked one by one in the priority order of the horizontal, vertical, and diagonal directions. If it is unreachable or already in the Close List, it is ignored. Otherwise, do the following:

a) If it is not in the Open List, add it to the Open List and set the current grid as its parent node;

b) If it is already in the Open List, use the F value as a reference to check whether the new path is better. If it is better, set its parent node as the current node, and recalculate its G and F values;

c) After checking the 8 adjacent squares, go to step (3).

5) Starting from the end point, each square moves along the parent node to the starting point, forming a path.

Right-click on any node in the map interface, select “Set as start point” or “Set as end point” in the pop-up menu, and then place the coordinates of the selected point on the map. After setting the start and end points, click the “Set Path” button in the upper right corner, and the system will automatically calculate and generate a path. The test result is shown in the figure below. The red trajectory is the path generated by the system as shown in **Figures 6-8**.

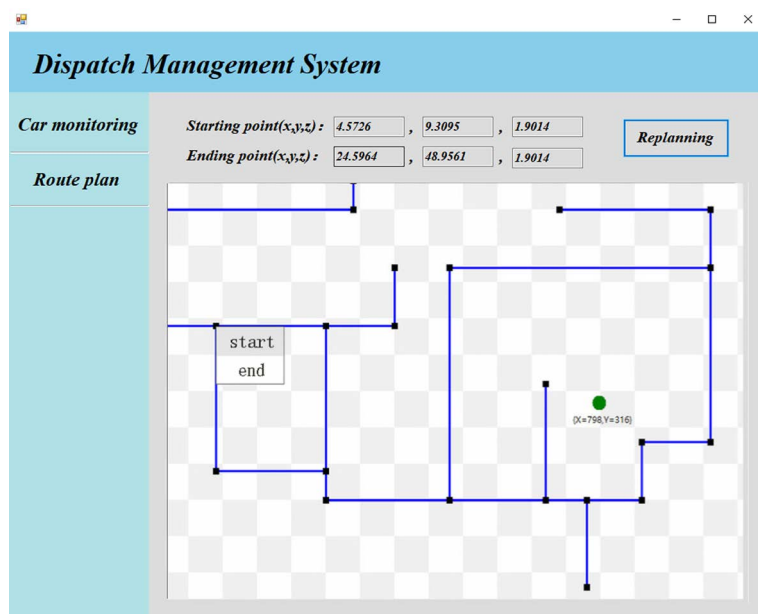


Figure 6. Setting the starting point.

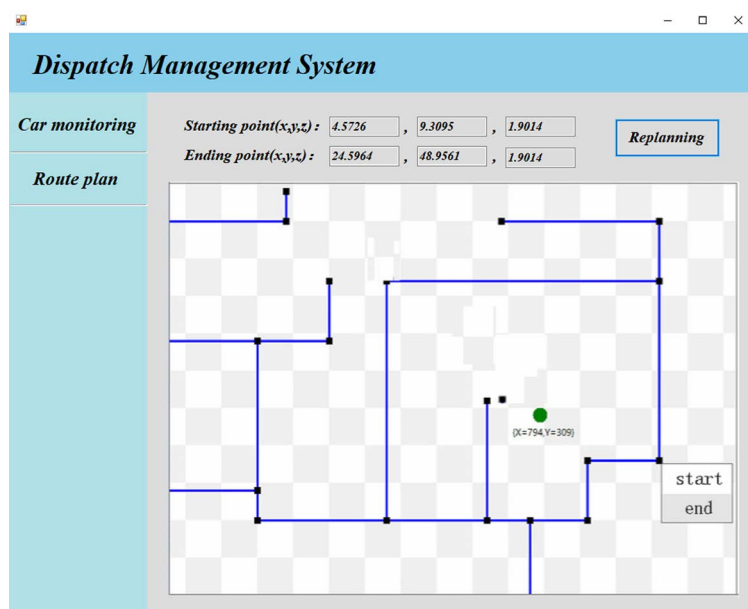


Figure 7. Setting the end point.

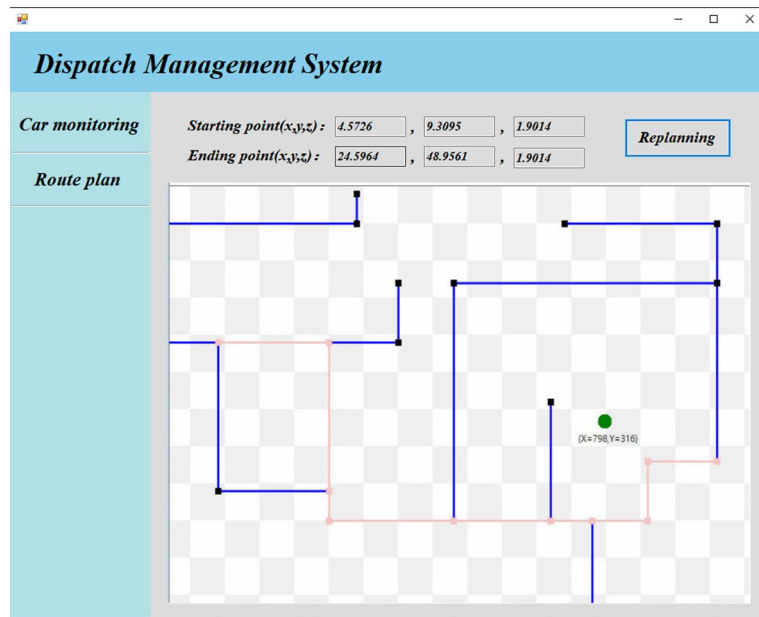


Figure 8. The system generates the corresponding path.

6. Conclusion

The traditional logistics and warehousing industry needs to consume a lot of labor resources, and manual operations are required in the packing and handling of goods. Compared with machines, it is inefficient, error-prone, and management is more confusing. Therefore, in view of the shortcomings of the current warehouse management system, this design takes the automatic tracking and obstacle avoidance trolley as the control goal and develops a dispatch management system, which finally completes the wireless positioning and path planning of the dispatch system and the control of the trolley through wireless communication. The function realizes fast and convenient warehousing and transportation trolley scheduling management, which can improve the efficiency of the warehousing management system, and is of great significance for reducing social logistics costs.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Hu, H. and Meng, Y. (2021) Analysis of Logistics Operation Situation in 2020. *Logistics Research*, 1-4.
- [2] Jiang, W. (2017) Research and Development of the Operation Efficiency Optimization Technology of Intelligent Three-Dimensional Storage System. Master's Thesis, Nanjing University of Science and Technology, Nanjing, China.
- [3] Li, J., Liu, X.X. and Xu, B. (2014) Design of Intelligent Tracking Vehicle Control System. *Automation Application*, 72-73+83.

- [4] Wang, X. (2021) Research on Indoor Target Tracking and Positioning Technology Based on UWB. Thesis, Shandong University, Jinan, China.
- [5] Gu, C. (2014) The Application of Improved A* Algorithm in Robot Path Planning. *Electronic Design Engineering*, **22**, 96-98+102.