

Indoor optimal path planning based on Dijkstra Algorithm

Yicheng Xu^{1,a}, Zhigang Wen^{1,2,b}, Xiaoying Zhang¹

¹ School of Electronic Engineering, Beijing University of Posts and
Telecommunications, Beijing, China

² Institute of Sensing Technology and Business, BUPT, Wuxi, China
^a997103460@qq.com, ^bzwen@bupt.edu.cn

Keywords: pedestrian guidance, Dijkstra algorithm, path smoothing, user preferences

Abstract. Providing optimal path information and the personal navigation path is the open issue for the pedestrian guided system. In order to solve the problems existing in the pedestrian guided system nowadays, such as the jagged paths which probable exist in the shortest path, and the lack of the considerations of different users' preferences, this paper firstly proposed a shortest path planning scheme for pedestrian, which is based on Dijkstra algorithm. Then some measures have been done which can be used to smooth the jagged paths on the basis of shortest path. At last, considering the users' preferences generated by collecting and analyzing users' behavioral information, we optimize the paths in advance. Above all, this paper mainly comes up with an indoor personalized pedestrian guidance method with path smoothing function.

Introduction

Along with the increasingly complexity with internal layout of large-scale buildings, and the increasing time that people spend indoors, it has a great practical significance to provide users with efficient and comfortable indoor navigation path planning. At present, there are mainly two problems with indoor pedestrian navigation systems: (1) For now most path planning methods use shortest distance or least time cost as the main standard [1]. But it is not applicable in the situation of indoor environment that a large number of short distance sections may lead to the appearance of jagged paths. (2) Most of the existing path planning methods don't take the different preferences and demands to the road condition of users into consideration, such as the choice between stairs and the elevator.

To solve the above problems, this article proposes a pedestrian induction method, which provides users with the shortest path from source to destinations firstly, then generates the smooth path that eliminates the jaggies of the former generated path, and gives out the personalized indoor navigation path combined with different user preferences at last.

The shortest path planning

The Choice of Basic Algorithm.

There are a lot kinds of algorithms to solve the problem of monophyletic shortest path, which are mainly divided into the traditional shortest path algorithm and the heuristic search algorithm. According to the research, analysis and comparison between all kinds of path planning algorithm, we conclude that Dijkstra algorithm and A* algorithm is one of the best ways to calculate the shortest path between two points. Dijkstra algorithm is recognized as a classic algorithm used to weight the shortest path, and also the most perfect in theory, the most widely used algorithm, which calculate the shortest path of the nonnegative weights network. The algorithm can guarantee the global optimality of the shortest path, but the disadvantage is the low efficiency and the high time complexity, which is up to $O(n^2)$. A* algorithm is introduced into heuristic search. It takes the location information into consideration in the process of path search, and extends node and path to the destination in priority, which improves the efficiency of the algorithm greatly. However, the efficiency advantage of the A* algorithm can be displayed only when the distance of the path is

over thousands of meters, while the disadvantage is obvious because of its` implementation complexity and no guarantees that the optimal solution is obtained. In view of the pedestrian guidance system, the complexity of the Dijkstra algorithm is acceptable. And most of the path distance is short in the indoor path planning, so this article choose Dijkstra algorithm as the basic algorithm of shortest path planning.

Dijkstra Algorithm Principle.

Dijkstra algorithm is a label-set algorithm, which divides network nodes into three types: not labeled nodes, temporary labeled nodes and marked nodes. Firstly, we initialize all the nodes to become the not labeled nodes, then transfer the nodes that connect to the shortest path nodes into temporary labeled nodes in the process of searching, and mark the nodes, which are the nearest nodes to the source point and selected from the temporary labeled nodes, into marked nodes each cycle, until we find the target nodes or all nodes become the labeled nodes [2].

Smoothing path planning

Design Idea.

In view of the jagged path problem, the design idea of this article is as follow: On the basis of finding the shortest path, calculate the first N shortest paths in turn, where the path is shorter than the latter one. Then determine whether the joint point is the turning point according to the slope comparison between the two paths, and figure out the number of the turning points. At last, through comparing and selecting out the path that has the least number of the turning points in the first N shortest paths, which also is the smooth path that has the shortest distance, we realize the jagged path smoothing.

Algorithm Design.

As mentioned above, Dijkstra algorithm is a classic algorithm to solve the narrow shortest path problem. From the research, it has been found that finding the first N shortest path can be converted to the problem that finding the special shortest path in the subgraph after deleting one side from the original path. The basic idea is: if we have first s-1 shortest paths r_1, r_2, \dots, r_{s-1} , and remained paths r_s, \dots, r_M are in the path set of the subgraph, it is obvious that r_s is the shortest path in the subgraph. After figuring out the No. s shortest path, use some subgraphs to replace the subgraph of the No. s shortest path, and the new set of the subgraphs is the set that we need to figure out the No. (s+1) shortest path [5].

The specific steps of smoothing path algorithm are as follow:

Step.1 Initialization

We define that the Paths array stores all the edges in a path, the CutEdge array stores all the removing edges of subgraph, and the mapping PathList stores every path`s name.

Step.2 Find the Shortest Path

Use the Dijkstra algorithm to figure out the shortest path between two points, and output the ID number of each node on the shortest path. Then put the every edge of the shortest path into the Path array, while making the path name "path 0" in the mapping PathList.

Step.3 Figure out the First N Less-Shortest Path Cyclically

Delete all the edges when the CutEdge array is not empty. Take out the No. j edge of the last shortest path and delete it. Use the remained edges as the subgraph to calculate the shortest path. After having a number of j paths, the new Less-Shortest path is the shortest one among them. Put this path into the PathList array, and new deleting edge into the CutEdge array. Then we finish one cycle. After doing above repeatedly N times, we can get first N Less-Shortest path.

Step.4 Compare the Number of the Turning Points

The turning point is the joint point that the slope is not the same on both sides when comparing the two adjacent edges. After comparing the turning point number of each path, the most smoothing path is the one that has the least turning point number.

Use of the Algorithm.

As is shown in picture 1, it is a weighted directed graph where v_i ($i = 1, 2, \dots, 11$) represents the node, and the number near the edge represents the weight of the edge. We set the $N=3$, and use the

above algorithm to plan the path. As the result shows, the first three shortest path from v_1 to v_{11} are $V_1V_5V_9V_{10}V_{11}$, $V_1V_8V_9V_{10}V_{11}$, $V_1V_5V_6V_7V_{11}$, the distance of each path are 41, 43, 45, and the most smooth shortest path is $V_1V_5V_6V_7V_{11}$.

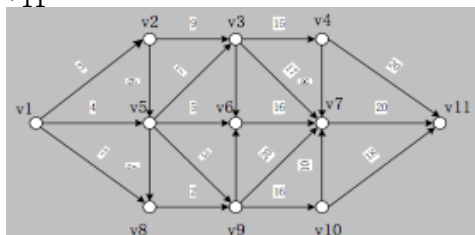


Figure 1. Weighted Directed Graph

From the experiment result, we can see that the algorithm can smooth the jagged path effectively, and get the most smooth and shortest path.

Research of user preference

Extraction of User Preference Data.

User preferences can also be referred as the user's interests that its information is relatively stable and requirement is relatively long-term [4]. When in the complex indoor environment, different users have the different requirements for roads under the same scene. And same user's preference may be same under different scenarios. So getting the user's preference factors by location, data acquisition and data analysis, and adding the preference factors as weights into road information, can help to generate the personalized indoor navigation path for different users.

Data Collection.

The data acquisition of model user preferences is combined of explicit sampling and implicit sampling two ways.

The explicit sampling includes two methods: user basic information sampling method and subjective assignment method. The former one gets the user's background information by asking the user directly, and the latter is a more mature and commonly used decision method, requiring the user to input certain preference information. The subjective assignment information that this model needs is the importance of the four factors: clear identification (λ_1), smooth degree (λ_2), distance (λ_3) and commodity type (λ_4). The information plays an auxiliary role in the analysis of user preferences, at the same time can partly solve the system problem of "cold start" due to the unsettlement of user preference model in the early years of the system.

The way of implicit sampling is as follow: first get the user's location and behavior information from client equipment, and sent the information to the background server to generate user's location information for data analysis, then obtain the user preference information and model the user preference. This article collects the times that user appears at the different characteristic nodes and the retention time two parameters as the user preference information.

Examples of application

The application example chooses the indoor mall as implementation scenario. Users are divided into two types of ordinary user and cart-using user. And we mainly consider the elevator type, commodity area type and the degree of jogged road as factors when generates the planning paths.

Different commodity type has the different attribute values A_i , and starting point is set to O, end to D, for user type set to G. The value of N is determined by the $\frac{\lambda_2}{\lambda_3}$ of subjective assignment method and Value Selection of N Table (Table 1).

Table 1. Value Selection of N

N	Smoothness	Distance	S/D
1	0	1	0
2	0.3	0.7	<1
3	0.5	0.5	1
4	0.7	0.3	>1
5	1	0	∞

The specific path planning algorithm is as follows (We take floor (O) = floor (D) as an example) :

Switch

Case G = cart-using user

Delete corridor whose width is below the cut-off line

Automatically set the subjective information: smooth degree > commodity type > distance

Calculate the weights of different attributes

Case G = ordinary user

Read the subjective assignment information

Calculate the weights of different attributes

Reduce the edge weights λ_1 connected with obvious signs

Read the user preference information, determine the user goods preference type K, and retrieve the commodity type attribute

While $A_i=K$

Reduce the edge weights λ_4 where the type is A_i

If is the preference shop

Reduce the edge weights λ_4 connected with the shop

Calculate $\frac{\lambda_2}{\lambda_3}$ and set N according to Table 4

Calculate the path

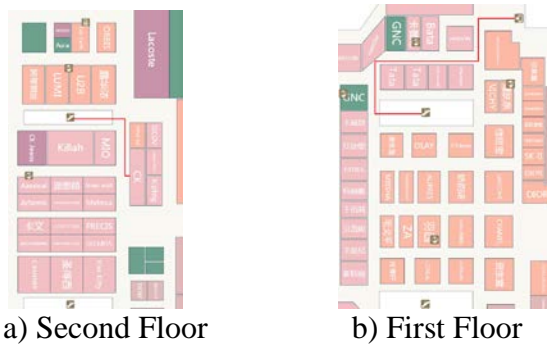


Figure 2. Indoor Induced Instance

Take Beijing Juntai mall as an example. The user type is ordinary user, subjective information is commodity type > distance > smooth degree, the user preference shop is GNC which concludes from the analysis of the client data, and there is no elevator preference. User needs to walk from second floor to the north gate exports on the first floor. According to user's information, the induction information is shown in figure 2.

As we can see from above, firstly, we don't need to delete the road factors. Secondly, user's starting point and destination is not in the same floor, so we need to choose the nearest elevator as the middle node. Because there is no user preference shop on the second floor, we plan the shortest path directly. The GNC is user's preferring shop on the first floor, and the commodity type is more important than the path distance, therefore the planning path from the elevator to the export on the first floor includes the GNC brand information rather than the shortest path.

It can be concluded that when under the indoor scenarios, the path planning based on the user preferences can provide user with personalized path planning scheme to improve the user experience.

Conclusion

According to the characteristics of complex indoor environment, this article comes up with a pedestrian induction method that is based on Dijkstra algorithm, which realizes the effective path planning between two points, and solves the problems of the jagged path smoothing. In addition, considering the user's personal preference, we also make the user preference as weight information, so as to realize the personalized path planning.

Acknowledgments

The work presented in this paper was supported by the National Great Science Specific Project (Grants No. 2014ZX03002002-004) and National Natural Science Foundation of China (Grants No. NSFC-61471067).

References

- [1] Patrick M. Dudas, Mahsa Ghafourian, Hassan A. Karimi. ONALIN: Ontology and Algorithm for Indoor Routing [C]. *Mobile Data Management: Systems, Services and Middleware*,2009, 720 ~ 725.
- [2] Jiang Fenghui, li Shujun, Jiang Fengjiao,etc. The Dijkstra Algorithm Improved and its Application in the Traffic Navigation System Based on GIS. *Geomatics & Spatial Information Technology*,2011,34(4):129~130.
- [3] Wang Fengbo, Cui Xianguo, Wang Ling, etc. Research and Implementation of Driving Navigation Path based on Dijkstra Algorithm. *Science and Technology of Western China*,2011,10(34):26~27.
- [4] Craig Silverstein , Monika Henzinger, Hannes Marais , et a l. Analysis of a very large Web search engine query log[C]. New York: In *SIGIR Forum* , 1999, 33(1): 6~12.
- [5] CHAI Deng-feng, ZHANG Deng-rong. Algorithm and its application to N shortest path problem. *Journal of Zhejiang University (Engineering Science)*, 2002, 36(5): 532-533.