Research on Personalized Indoor Routing Algorithm

Weijun Bian, Yucheng Guo*, Qizhi Qiu
School of Computer Science and Technology
Wuhan University of Technology, Wuhan, China
bwj1990032368@gmail.com; ycheng.g@gmail.com; 736019272@qq.com

Abstract—As the mature GPS positioning technology cannot work well in indoor environment, there is no mature indoor navigation system for civil use. Besides indoor maps and indoor positioning technology, indoor routing algorithm is an important part of indoor navigation technology. This paper first discusses the problem of cross-storey in buildings and gives a solution. Unlike outdoor routing, shortest path is usually not the best path indoors, a personalized path considering user preference and interest can be better. To achieve personalized routing, this paper comes up with a way to model and acquire user preference. On this basis, this paper improves traditional A* algorithm by considering user preference and then gives an example to show the advantage of the personalized A* algorithm. The results show that the improved algorithm can find a better path by considering path length and user preference synthetically.

Keywords—indoor navigation; routing algorithm; user preference; personalized.

I. Introduction

With the development of the society, increasingly, there are more and more buildings with large scale and complex internal structure and the demands for the indoor navigation service are also increasingly urgent. Although in the past few years, electronic maps and navigation software had made great processes, the research on indoor navigation still progressed at a slow pace. The mature GPS positioning navigation system can not work indoors because it can not receive signals. Because there are bottlenecks in the indoor positioning technology, most studies on indoor navigation are focusing on the robotic navigation and there are no mature civil indoor navigation products by far[1-5]. Meanwhile, in the era of network, information and digital things, personalized technologies have been used in various fields. Because there are bottlenecks in the indoor positioning technology, this technology has not been well used in the indoor navigation field. However, it will be meaningful to use personalized indoor navigation according to users' interests in today's high-paced life. Personalized indoor navigation service should not only be based on the technologies of indoor maps and indoor navigation, but also be supported by the personalized indoor routing algorithm. This is also the focus of this paper.

II. Cross-storey problem

The problem that the routing algorithm needs to solve is to find the path between the two points in the graph and this path is usually the shortest path between these two points. The common algorithms are Dijkstra algorithm, Floyd algorithm, Bellman-Ford algorithm, A* algorithm and so on. These algorithms can be well used in the outdoor routing. However, these algorithms cannot be directly used in the indoor routing because there are so many storeys in the building [6].

a. First of all, we should consider the circumstance that we compose all the needed graphs to make a big connected graph and at that time the topological structure of the graph need to keep changing. By doing this will spend much more time in changing the topological structure which will make the operational efficiency become low and the coding will be very difficult, too.

b. We can connect all the storeys through the stairs to make a bigger connected graph. In this connected graph we can inquire any shortest path between two points without changing topological structure of the graph. However, when using the blind search algorithms such as the Dijkstra algorithm, the execution times of the algorithm will be greatly increased.

c. When considering the stairs as one side of the connected graph, the weight will not be well determined. When the weight is too small, while the heuristic algorithms such as the A* algorithm are searching paths, they will take the first path it finds as the path without considering other paths which is obviously unreasonable. On the other hand, when the weight is too big, the algorithms will not to choose the stairs, which will make it finding no paths.

We've considered that when users are going to different floors, they will usually use the same stair, especially when users are using shuttle elevators. Therefore, this is our solution to the cross-storey problem.

(a) To find the path between the position of the user and one of the stair in the building.

(b) To find the path between the corresponding stair in the target floor and the target position of the user.

(c) Take the paths (a) and (b) have found as a feasible path.

(d) Repeat (a) to (c) and find the feasible paths for every stair.

(e) Compared all the feasible paths, which have been found before and then choose the best one.

Therefore, the cross-storey problems have become the routing problems between two points. After getting all the paths when users choose every stair, we can get a best path by comparing these.

III. Personalized routing algorithm

Traditional routing algorithm is used to find the shortest path between two points and it is fully applicable for the outdoor navigation routing, which can arrive at the destination as quickly as it can. For some indoor places such as the airports, the train stations, the traditional shortest routing can be also very effectual. However, for the more common

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.60

scenarios in people's daily life, the shortest path doesn't mean the best path. For example, when users are in a big shopping mall, if there are much more places that the users interested in on one path, he can pass through these places when he choose this path, sometimes he can even find other more places that he interests in. This will not only save users' time in another way but also enhance the user experience of the indoor navigation. Even though this path is not the shortest one, it is a better one[7]. Therefore, it will be meaningful to use personalized indoor navigation to find a best path for users according to users' interests.

A. Preference modeling

Preference modeling is the process that generalizes the measurable modeling from the information about users' preference and behaviors. The accurate descriptions about users' preference cannot be modeling. Preference modeling is not the common description about users, but a kind of users' formalized description, which faces to algorithms and have specific data structure [8].

In this paper, we choose to use the notation, which is based on the vector space model. It is a way, which uses the vector in the key word vector space to express the user models, and it depends on the particular cases. The main basis is to be determined by the places in the building that the users are interested in. if users are in a shopping mall, the subject terms of their interests may be determined as entertainment, food, sport and so on. So users' interests can be expressed as a vector of subject terms: **U**={k_1, k_2, k_3,...k_n}, k_n means the weight of the nth subject term. The weight will choose a number between 0 and 1 according to users' preference. On the other hand, the contents on each position are expressed by an n-dimensional vector to express the content features of the position. By calculating the similarities of these two vectors, we can know the preference of users. The calculation formula of the similarity is as follow, among them, **m** expresses the preference feature vector of users and **n** expresses the content feature vector of the position:

$$sim = \frac{\vec{m} \bullet \vec{n}}{|\vec{m}||\vec{n}|} \tag{1}$$

B. Preference acquisition

The acquisition of users' preference is based on the communication between system and users. There are two kinds of personalized feedbacks: the implicit feedback and the explicit feedback. The explicit feedback needs users to take part in it directly which means that users should provide his information by themselves and give evaluations to the present program and system in order to get the preference. The implicit feedback doesn't require users to provide their information and all the tracks are done automatically by the system. Implicit feedback can be got by tracking users' browsing histories and checking the users' behavior log.

This paper uses the notation of the vector space model to express users' preference. We use user action logs to inspect the positions that users used to take as the destinations because the influences on the present position information are not so big. We can suppose that the destinations are $\mathbf{l_1}$, $\mathbf{l_2}$ ..., $\mathbf{l_n}$,the times that take the places as destinations are $\mathbf{t_1}$, $\mathbf{t_2}$ ..., $\mathbf{t_n}$, we can get users' preference vector by using the weighted sum of the destination vector. If $\mathbf{T}=\mathbf{t_1}+\mathbf{t_2}+...+\mathbf{t_n}$, then the calculation formula of users' preference is as follow:

$$\vec{m} = \sum_{i=1}^{n}\left(\frac{t_n}{T}\vec{l}_n\right) \tag{2}$$

When users use the system at the first time, they cannot get the preference vector because they don't take any places as the destination. At this moment users can get the preference vector by answering the multiple-choice questions that offered by the system. For example, if the choices of one-subject terms are hate, fair, like, enjoy, we can ensure that the weight of this subject term is 0, 0.4, 0.6, 0.8 according to users' choices. Users will get a preliminary user preference vector **s** after answering every question. As users' behaviors come into being, that is to say, to navigate by taking some positions as destinations, we can get a preference vector **m'** by the using the ways explained before according to these positions. Then we can get the present preference vector by using the weighted sum of the two vectors, which is $\mathbf{m}=\mathbf{w_1 s}+\mathbf{w_2 m'}$. Weight $\mathbf{w_1}$ and $\mathbf{w_2}$ will be determined by the specific circumstances. Because the preliminary user preference vector **s** can not reflect the users' preference and interests well, when the times of navigation reach some scale, we can totally use **m'** to determine the user preference and at this time **m**=**m'**.

C. The personalized A* algorithm

This paper uses the improved A* algorithm to solve the routing problems between two points. The A* algorithm has defined the evaluation function **F**=**G**+**H** in order to find the shortest path. Among which **G** expresses the consumptions which have been produced between the start point and the right now point, and H expresses the forecast consumption which will be produced between the right now point to the final point [9]. H is a forecast value and has many ways to be defined. This paper chooses to calculate the distance from the present coordinate (**x**, **y**) to the target location coordinate ($\mathbf{x_1}$,$\mathbf{y_1}$). The calculation formula is as follow:

$$H = \sqrt{(x-x_1)^2 + (y-y_1)^2} \tag{3}$$

In order to find a personalized routing, this paper has improved the evaluation function. The value of **G** has changed from the consumptions which have been produced between the start point and the right now point to the consumptions which is related to users' preference value, that is :

$$G = \sum\left((1-sim_i)D_i\right) \tag{4}$$

$\mathbf{D_i}$ expresses the consumption weight of this path while $\mathbf{sim_i}$ expresses the cosine similarity between the content feature vectors of this path's destination and users' preference features vector. When the destination of this path is just at the target position, **sim** is 0, which means that there's no need to consider the similarity between destination and users' preference. To begin the algorithm, two arrays should be built. The CLOSED array saves the nodes, which have been inspected; The OPEN array saves the nodes, which have not been inspected. The pseudo code of the algorithm is as follows:

Algorithm 1 Personalized A* algorithm

```
while(OPEN!=NULL)
//when OPEN is empty,the path to target place doesn't exist
{
n=node which has the lowest F in OPEN;
if(n==TargetNode)//find the path to the target place
break;
for(each adjacent node x of n){
if(x is in OPEN){
compute F' and G' of x //compute new G and F of node x
if(G' < G){//this path is better
parent node of x=n;//set n to the parent node of x
G=G';F=F';//refresh values of G and F for node x
}//end if
}//end if
if(x is in CLOSED)
continue;
if(x is not in both){
parent node of x=n;//set n to the parent node of x
compute G and F of x;
Insert x into OPEN;
}//end if
}//end for
remove n from OPEN;
insert n into CLOSED;
}//end while
```

D. Evaluation of the algorithm

In the connected graph showed as Fig 1, we can assume that the estimate value H between $\mathbf{L_i}$(i=0,1...,6) and the end point is respectively 7、6、5、4、3、2、1 and the similarity $\mathbf{sim_i}$(i=1,2...,6) between destination and users' preference is respectively 0.6、0.4、0.4、0.7、0.4、0.5. The weight of the path length has been marked on the figure.

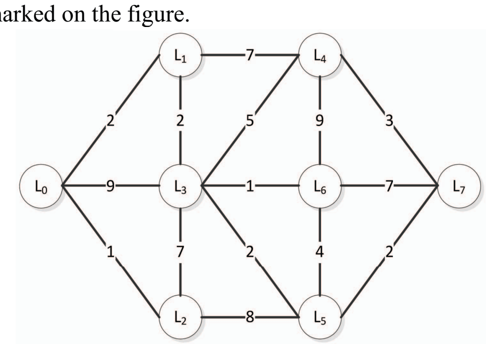

Fig. 1. Connected Graph

When using the traditional A* algorithm, the path we can find is L0->L2->L5->L7, this path is the shortest path and the sum of the length weight is 11. While using the personalized A* algorithm, the sum of the length weight is 12 which is greater than the shortest path. However, because during the way users will pass through the L1 and L4 which have high similarities to user preference, if considering the path length with weights of user preference we can get the sum as 5.9 which is less than 7.4, the sum of the path which traditional A* algorithm finds. Therefore, the path found by the personalized A* algorithm is considered as a more coincident path to users' preference.

This kind of algorithm has considered both path length and users' preference. The algorithm will search the target position because the **H** in the estimating function is the distance between the user's position and the target position. We need to consider two extremes. When their cosine similarities are similar to each other, the algorithm will choose the same path which the traditional A* algorithm will choose. If the differences of the consumption value were little and the differences of the sim value were big, the algorithm will choose the path, which is more suitable for users' preference. For other situations, the algorithm will consider both the path length and the preference in order to choose the best path.

IV. Conclusion

This paper has studied the personalized indoor routing algorithm. First of all, the paper puts forward the cross-storey problem in the indoor routing and provides the solution to the problem. By doing this, the paper has turned the cross-storey problems into routing problems between two points. Second, the paper models the users' preference and uses space vector to express users' preference and position's content. On this basis, the paper improves the A* algorithm in order to make the routing more personalized and finally reach the aim of using personalized indoor routing.

References

[1]Lei Cao. Study of the Location and Navigation Services in Complex Indoor Scenes Based on Android Mobile Computing Platform[D]. Wuhan University of Technology. 2013

[2]YuCheng Guo, Lei Cao. Study of the Location and Navigati on Services in Complex Indoor Scenes Based on Android Mob ile Computing Platform[C]. DCABES 2012,Guilin,2012:91-93.

[3]Han Changyong, JAEGEOL Y, GYEYOUNG I. Implement ation of Web Services for ILBS[J]. Communication in Comput er and Information Science, 2011(261):415-422.

[4]HUANG Haosheng, GARTNER G. A Survey of Mobile ind oor navigation systems[J]. Lecture Notes in Geoinformation an d Cartography, 2013(3):305-319.

[5]Jain, M. ,Bangalore, Rahul ,Tolety, S. . A study on Indoor n avigation techniques using smartphones[J]. Advances in Comp ting, Communications and Informatics(ICACCI), 2013 Interati onal Conference on 22-25 Aug. 2013:1113-1118.

[6]Shaohua, Luo. Research on the Navigation Algorithm[D].B eijing University of Posts and Telecommunicaions.2011.

[7]Dudas, P. M. ,Ghafourian, M. ,Karimi, H. A. .ONALIN: On tology and Algorithm for Indoor Routing[J]. Mobile Data Man agement:Systems,Services and Middleware, 2009.MDM '09. T enth International Conference on 18-20 May 2009:720-725.

[8]]Li Chun, Zhu Jianmin, Yejian, Zhou Jiayin. Survey on rese arch inpersonalization service[J]. Application Research of Co mputers, 2009,26(11):4001-4009.

[9]Liu Hao, Bao Yuanlu. A* Algorithm for Finding the Optim al Path on Vector Maps[J].Computer Simulation,2008(4):253-257.

* Yucheng Guo, Corresponding Author