A FEASIBLE VOICE BASED INHOUSE NAVIGATION SYSTEM FOR THE VISUALLY IMPAIRED

Soham Roy

Kalinga Institute of Industrial Technology, Bhubaneswar, India

ABSTRACT

A proposal has been made to implement an inhouse navigation system for the visually impaired persons, which is costeffective, easy to implement and very easy and user-friendly and can be used by blind or partially blind people to go about the inside premises of an office or any building or floor having turns and obstacles and different locations.

KEYWORDS

RFID tag/reader, antenna, navigation, BFS, obstacles, messages, transmissions

1. INTRODUCTION

We are all familiar with google maps in the roads which has become a part of our day to day life. While they immensely help in the navigation the unknown streets that enables us reach our destination, an implementation of such an application with all the features involved is only possible to a global giant like google. But if we think of such navigation systems inside the building, for example inside of an office premises, just the thought of it feels to be swept aside as to why a normal person would require a navigation system inside the building, which has probably all the directions/signboards and can be familiarized in a few visits. But the same cannot be said for a visually impaired person, he would still be required instructions on the directions he needs to walk and the turns he needs to make and the obstacles that is in front of him to make him reach his destination. As building an app like google maps for each such office premises is out of question, a proposal has been made that is cheap and easy to implement and effective and aimed at the solution to this problem.

2. METHODOLOGY

We have come up with a solution to combine the capabilities of RFID reader/tags which are basically small sticker like equipment’s which can be attached to a wall or any surface and the tag can respond and can transmit message to an antenna/reader which comes in close proximity, the range of proximity can be set according to requirements.

Incorporating this into our solution, the tag will be integrated with the smartphone, which is a device that everyone uses, and a separate hardware device won’t be required. That will be used as an antenna and coming in proximity to the tag will transmit signal to the smartphone and which in turn will instruct directions to the visually impaired person typically in the form of like keep walking straight, turn right, left and soon.
We have categorized obstacles into two parts:

**Permanent obstacles** like office bays, walls and objects that cannot be moved which comes in the way of the user from the source to the destination and the avigation system has to bypass and guide the user through a viable path by providing instructions.

**Temporary obstacles** like chair, and other stuff that can be moved from the path, these kind of obstacles will be notified to the user by means of object detection algorithm and the smartphone can be used for this one too.

3. **PROPOSED SOLUTION**

For the scope of this paper we will focus only on the permanent obstacles (Point 1), as for Point 2 there are several state of the art solutions available.

For the sake of simplicity let’s assume that the user will be navigating one floor of an office building and the floor is a rectangular floor. And the user will be present at any location on the floor and will use the navigation system to go to any destination in the same floor for e.g. lobby, cafeteria, restroom etc.

So, in the rectangular floor we have placed RFID readers at strategic locations which can transmit signal to the user antenna and then after the RFID unique identifier matches, the instruction will be passed on to the user at the right moment. The floor has been considered from a bird’s eye view for better visualization purposes.

![Diagram](image.png)

Figure 1 is a snapshot of how this looks like, the $L_N$ are the position of the RFID tags which is guiding the user to the destination.

Let’s assume the floor is a 2D matrix and the RFID readers have been placed strategically at an intersection of a row and a column. Every permanent obstacle will be known by the 2D map of the
floor in the system from beforehand so when calculating the path, it can steer the user to a relevant path probably by directing him, to turn left/right.

If the algorithm selects the PATH 2 to the restroom here, so the command in this case will be

Go straight go straight go straight go straight you haveto take left on next turn left you have to turn right on next turn next turn turn right destination reached

Each of this instruction above will be passed on to the user one by one as he reaches to close proximity to each of the RFID readers. Here also the user is also being notified from before if he has to make a turn like right or left. And a failsafe mechanism has also exists if the user deviates from the path then the algorithm/RFID system will guide the user on the right track.

4. ALGORITHM / WORKING:

For our purpose we have assumed the floor is a 2D rectangular matrix of shape (50*90) and the RFID readers have been placed at every 10th intersection of a row and a column. every permanent obstacle will be known by the virtual map of the floor from beforehand and will be marked ranging from one tag to another. i.e if their exists an obstacle then the path encompassing the nearest 2 RFID tags will be blocked. the location of the major destination will also be provided in the floor map like desk no, lobby , cafeteria etc.

Figure 2. -the birds eye view of the floor, the yellow dots represent position of the RFID tags, the lines in green represents the permanent obstacles.

For the core navigation algorithm we have used Breadth First Search Algorithm and each node is a location of a RFID reader and the obstacles are marked in the map from beforehand i.e.prior to running the algorithm the navigation system will have a complete knowledge of the permanent obstacles in the map as well as the destinations.
During the start of the application the location of the user will be determined based on nearest RFID tag and the destination will be prompted from the user, after we have the set of current loc and destination loc, the navigation function using BFS as the core algorithm will be invoked and the list of the unique identifier of the RFID tags that will be on the path is determined and the steps or instructions to guide the user will also be determined and stored. Now as the user antenna attached to the smartphone matches and receives the signal from the nearest RFID readers, it matches with the RFID unique identifier to that of the imminent RFID UID in the list of the path determined(from the navigation algorithm), if it matches then the user is on the right track and the message of the next step(i.e go straight/left/right) is transmitted to the user or else this entire process is started from beginning to reroute the user in the right track.

**Pseudocode:**

```plaintext
Func start_application(destn=Blank):

Present_loc_coordinates = Determine_present_location() // from nearest matched RFID tag
If(destn == blank):
    Destn = ask_user_input()

Destn_coordinates = determine_destn_coordinates(Destn) // from mapped data

Points = determine_nav_path_coordinates(present_loc_coordinate, destn_loc_coordinate)
// this fn will return list of coordinates of the RFID tags in the path
Path_RFIDUid = Get_RFIDUids_for_path_determined(Points) // will return list of RFID Uids
// Which will come in the path of the user

Messages_List = Determine_messages(Path_RFIDUid) // this function will return a list of messages
// to be passed on to the user
While(destn_reached=False):
    Listen = detect_transmission()
    If Listen.id != RFID_in_list.head() // user deviated along the way
        Func_start_application(destn = Destn) // restart the application
        with
        // Destination
        as the parameter
        Else Listen.id == Path_RFIDUid.head() : Path_RFIDUid.pop()

        Transmit_msg(Messages_list.head())
        Messages_List.pop()          End of Func_start_application()
```
RESULTS:

This figure shows the source and destination of the user, the top left (0,0) is the source and the point (30,70) is the destination which is mapped as location “USER DESK” in the map. After running the Func_start_application() for this set of source/destination: We get RFID UIDList={ 0, 9, 18, 19, 20, 21, 22, 13, 14, 15, 16, 25, 34 }

At Step1: RFID UID=0 (which is also source location of the user marked in Figure 3)

For simplicity sake we assumed that the user face is turned towards the direction of movement.
Message transmitted="start moving straight "

At Step2:- RFID_UID_detected = 9 (matches RFID_UIDList = True)

Message transmitted="go straight and you have to take a left on the next turn"

At Step3:- RFID_UID_detected = 18 (matches RFID_UIDList = True)
Message transmitted=""Turn Left ""

At Step4:- RFID_UID_detected = 19 (matches RFID_UIDList = True)

Message transmitted=""go straight ""

At Step5:- RFID_UID_detected = 20 (matches RFID_UIDList = True)
Message transmitted="go straight"

At Step 6:- RFID_UID_detected = 21 (matches RFID_UIDList = True)

![Figure 9. step 6-RFIDUid-21](image)

Message transmitted="go straight and you have to take a left on the next turn"

At Step 7:- RFID_UID_detected = 22 (matches RFID_UIDList = True)

![Figure 10. step 7-RFIDUid-22](image)
Message transmitted="turn left and you have to take a right on the next turn"

At Step 8 :- RFID_UID_detected = 22 (matches RFID_UIDList = True)

![Figure 11. step 8-RFIDUid-22](image)

Message transmitted="turn right"

At Step 9:- RFID_UID_detected = 13 (matches RFID_UIDList = True)

![Figure 12. step 9-RFIDUid-13](image)
Message transmitted = "go straight"

At Step 10: RFID_UID_detected = 14 (matches RFID_UIDList = True)

![Figure 13. step 10-RFIDUid-14](image)

Message transmitted = "go straight"

At Step 11: RFID_UID_detected = 15 (matches RFID_UIDList = True)

![Figure 14. step 11-RFIDUid-15](image)
Message transmitted = “go straight and you have to take right on the next turn”

At Step12:
- RFID_UID_detected = 16 (matches RFID_UIDList = True)

At Step13:
- RFID_UID_detected = 25 (matches RFID_UIDList = True)
Message transmitted="go straight"

At Step14: - RFID_UID_detected = 34 (matches RFID_UIDList = True and this is destination location or destination RFID tag)

Figure 17. step 14-RFIDUid-34

Message transmitted= "Destination reached"

Figure 18s. Direction plot of alternate path
Step1 :RFIDUid 0- Message= “start moving straight”
Step2: RFIDUid 9 – Message =”go straight and you have to take a left on the next turn”
Step3 :RFIDUid 18- Message= “turn left”
Step4 :RFIDUid 19- Message= “go straight” Step5 : RFIDUid 20- Message= “go straight”
Step6 : RFIDUid 21- Message= “go straight and you have to take a right on the next turn”
Step7 : RFIDUid 22- Message= “turn right and you have to take a left on the next turn”
Step8 : RFIDUid 31- Message= “turn left and you have to take a right on the next turn”
Step9 : RFIDUid 32- Message= “turn right and you have to take a left on the next turn”
Step10 : RFIDUid 41- Message= “turnleft”
Step11:RFIDUid42-Message=“gostraightandyouhavetotakealeftonthenextturn”
Step12 : RFIDUid 43- Message= “turnleft”
Step13 :RFIDUid 34- Message= “destination reached”

An alternate path to reach the same destination:

Thusthemovingpositions oftheuserwereidentifiedbytheRFID/reader/antennasignaltransmissionswhenevertheycameincloseproximityanthentherightmessagewastransmittedtotheuser onthenextmovements stepstobetaken and ofcourse this will be coupled with object detection to notify the user of temporary objects which lie in its path so as a whole this will turn out to be a complete feasible, easy to implement voice enabled navigation system.

5. CONCLUSION

A novel arrangement was proposed on guiding an impaired user to a destination which is very accurate, efficient, feasible, cheap and easy for custom implementation in any physical space or inhouse premises. However, in the solution discussed above it is of utmost importance that the accurate virtual map be set up in memory resembling to the real location/floor map as closely as possible as the virtual map in the memory is the ultimate playground upon which this algorithm pivots. One also has to bear in mind of the strategic locations of the placement of the RFID tags on the floor especially at the corners and turns. There is no hard and fast rule of the correct distance between RFID tags as the algorithm is independent of the distance between the tags. It depends on the budget and the complexity of a map, the more a map is of complex nature with corners and turns and wide spaces, the more the number of RFID tags will be needed to fit the model. However, in our implementation and testing we had set up the entire system with a straight-line distance of 5 to 7 meters between two tags and that gave very good and accurate results of efficiency to cost output. The failsafe mechanism is also checked at every iteration of the algorithm and is triggered at the point of deviation when the user misses any instruction and strays from the determined path, which is also very fast to recalculate the path and a new batch of instructions. Even though in this case, the entire algorithm starts from the beginning it worked very fast on our tested 90*50 matrix without the user even realizing it. The complexity of the algorithm is O(N2) where N is the number of RFID tags used in the map, and it produces an optimal result owing to the fact that only a finite number of RFIDs will be used even to cover a large area. So, in a whole this a quick and complete solution to a problem which falls under a very complicated space like navigation. As a part of future work, we would like to test with complex and arbitrary shaped locations which would lead to more complicated maps and bring about more describable and elaborate messages along with other navigation features like traffic analysis etc. to encompass and address a wider problem in navigation space.
REFERENCES


AUTHOR

Soham Roy - Completed his Bachelors in computer science and engineering from Kalinga Institute of Industrial Technology in 2018. His primary research interest lies in Data Science AI/ML, Neural Networks and Algorithms.