6-2022

Poster: Indoor Navigation for Visually Impaired People with Vertex Colored Graphs

Pei Du
Portland State University, pdu@pdx.edu

Nirupama Bulusu
Portland State University, nbulusu@pdx.edu

Citation Details

Du, P. and Bulusu, N. (2022). Poster: Indoor Navigation for Visually Impaired People with Vertex Colored Graphs. In The 20th Annual International Conference on Mobile Systems, Applications and Services (MobiSys ’22), June 25–July 1, 2022, Portland, OR, USA. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3498361.353879

Poster: Indoor Navigation for Visually Impaired People with Vertex Colored Graphs

Pei Du*
Portland State University
Portland, Oregon, USA
pdu@pdx.edu

Nirupama Bulusu
Portland State University
Portland, Oregon, USA
nbulusu@pdx.edu

ABSTRACT

Visually impaired people face many daily encumbrances. Traditional visual enhancements do not suffice to navigate indoor environments. In this paper, we explore path finding algorithms such as Dijkstra and A* combined with graph coloring to find a safest and shortest path for visual impaired people to navigate indoors. Our mobile application is based on a database which stores the locations of several spots in the building and their corresponding label. Visual impaired people select the start and destination when they want to find their way, and our mobile application will show the appropriate path which guarantees their safety.

CCS CONCEPTS

• **Human-centered computing → Accessibility**; *Ubiquitous and mobile computing*; Interaction design.

KEYWORDS

graph coloring, shortest path finding algorithm

ACM Reference Format:
Pei Du and Nirupama Bulusu. 2022. Poster: Indoor Navigation for Visually Impaired People with Vertex Colored Graphs. In *The 20th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '22), June 25–July 1, 2022, Portland, OR, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3498361.3538797

1 INTRODUCTION

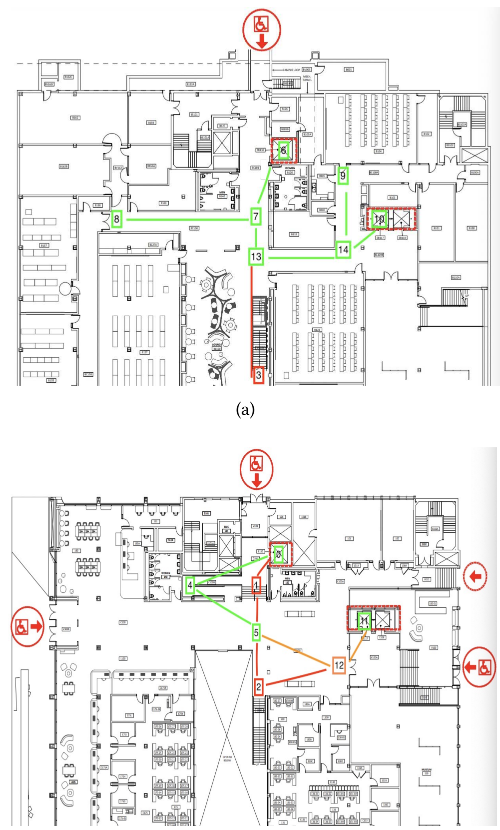
Visual impairment is a pervasive disability which consists of limited vision, blurry vision, light sensitivity and blind spots[1]. According to CDC, nearly 21 million people in United States have low vision problems, encountering daily encumbrances in activities such as indoor wayfinding, relying traditionally on audio and visual guidance [2–9]. However, these techniques do not always help them find the safest and shortest way to their destination. We build our wayfinding application based on graph theory [10] and our previous work [11]. A volunteer can use our system to automatically collect data for Assisted Navigation Systems, replacing human effort with deep learning, augmented reality and indoor positioning technologies.

Our application depends on the database that the volunteer collects. We consider the locations such as doors, elevators, stairs as graph nodes. Based on the labels in the database, we classify them as safe, cautious, and unsafe and mark them with different colors. And then we apply shortest path algorithms colored graphs to find the safest and shortest way for vision impaired people.

The contributions of our paper is to design an algorithm which uses Dijkstra and A* algorithms on colored graph to find the safest and shortest way between several floors. We next describe our work in graph coloring (Section 2).

MobiSys '22, June 25–July 1, 2022, Portland, OR, USA

ACM ISBN 978-1-4503-9185-6/22/06.
https://doi.org/10.1145/3498361.3538797



(a)

(b)

Figure 1: Graph Coloring (a) Basement (b) First floor

2 GRAPH COLORING

Graph Coloring is a step which needs to be accomplished before we run indoor wayfinding application. We use the floor plan of a specific building and consider it as a graph and classify the nodes as three categories, safe nodes, nodes with caution, and unsafe nodes.

The colors assigned to each category are green, orange and red respectively. For visually impaired people, it is very dangerous for them to use stairs to reach a different floor. Hence the stairs should be always be marked as red. In contrast, the elevators are very friendly to them, so we mark it as green. For some obstacles, such as desks which may have potential hazard or interrupt the user, we mark them as orange. If the user cannot find a path with green nodes, the application will show a path with a mixture of green and orange nodes, and the users should proceed with caution. In Figure 1, we assign a color to different spots and connect them with weighted edge based on the distance calculated from the positions.

We create two graphs G_1, G_2 to represent two floors and remove red nodes first to get G_{1r}, G_{2r}. Then, we also need to remove orange nodes G_{1o}, G_{2o}. In order to guarantee the safety of the user, G_{1o}, G_{2o} have higher priority than G_{1r}, G_{2r}. Only when the algorithm cannot find a path in G_{1o}, G_{2o}, we need to choose a path with orange nodes and use G_{1r}, G_{2r}. For the floor the user starts their path, we find the shortest path from the start point to each elevator, save the path and total weights. Similarly, for the floor where the user's destination lies, we find the shortest path from each elevator to the destination. We concatenate a final route with same connecting point (i.e. the elevator) and compare the total weights of each route and return the shortest route. Algorithm 1 shows the pseudo code of safest path finding algorithm. We test our algorithm on Android Studio, search our way from node 0 to node 9 and we attach the screenshot (see Figure 2) to show the safest and shortest path we find for the graph in Figure 1.

Algorithm 1 Safest Path Finding Algorithm Between Two Floors

Require: $V_{start} \in G_1(E_1, V_1), V_{end} \in G_2(E_2, V_2)$
Ensure: $(V_{start}...V_{elevator}...V_{end})$

Remove Red Nodes in G_1, G_2, get G_{1r}, G_{2r}
Remove Orange Nodes in G_{1r}, G_{2r}, get G_{1o}, G_{2o}
for $V_i \in V_{elevator}$ *in* G_{1o} **do**
    Run Dijkstra or A* on G_1o from V_{start} to V_i
    if Shortest path found **then**
        save path
    else if No path found **then**
        Run Dijkstra or A* on G_1r from V_{start} to V_i
        save path
    end if
end for
for $V_i \in V_{elevator}$ *in* G_{2o} **do**
    Run Dijkstra or A* on G_2o from V_i to V_{end}
    if Shortest path found **then**
        save path
    else if No path found **then**
        Run Dijkstra or A* on G_2r from V_i to V_{end}
    end if
end for
Concatenate path based on same $V_{elevator}$ and compare all the paths
return shortest path $(V_{start}...V_{elevator}...V_{end})$

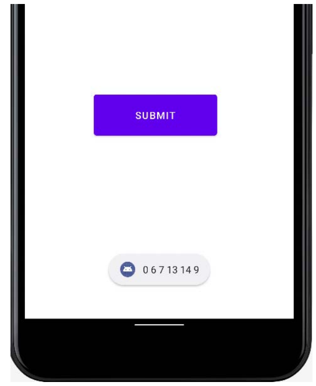


Figure 2: Test Result

3 CONCLUSION

We have proposed an algorithm for visually impaired people to help them find their way safely and quickly in the indoor environment. Our method doesn't need any installations for sensors or visible marks, only one volunteer can collect positions and labels of nodes by using automated AR-based annotation tool and then color the graph built from the floor plan. The users can hold a mobile device to guide their way to destination. We just finished the design of the algorithm and test it in the emulator. In the future, we have to calibrate the position and use AR tools or audio guidance to help them find their way in the real environment.

REFERENCES

[1] AOA (American Optometric Association). Common types of low vision. 2015.
[2] Yuhang Zhao, Sarit Szpiro, and Shiri Azenkot. Forsee: A customizable head-mounted vision enhancement system for people with low vision. pages 239–249. International ACM SIGACESS Conference on Computers and Accessibility, 2015.
[3] Yuhang Zhao, Sarit Szpiro, Lei Shi, and Shiri Azenkot. Designing and evaluating a customizable head-mounted vision enhancement system for people with low vision. ACM Transactions on Accessible Computing, Dec 2019.
[4] Yuhang Zhao, Sarit Szpiro, Jonathan Knighten, and Shiri Azenkot. Cuesee: Exploring visual cues for people with low vision to facilitate a visual search task. pages 73–84. ACM UbiComp, Sep 2016.
[5] Yuhang Zhao, Michele Hu, Shafeka Hashash, and Shiri Azenkot. Understanding low vision people's visual perception on commercial augmented reality glasses. Number 4170-4181. ACM CHI, 2017.
[6] Yuhang Zhao, Elizabeth Kupferstein, Brenda V. Castro, Steven Feiner, and Shiri Azenkot. Designing ar visualizations to facilitate stair navigation for people with low vision. UIST 2019, Aug 2019.
[7] Yuhang Zhao, Elizabeth Kupferstein, Hathaitorn Rojnirun, Leah Findlater, and Shiri Azenkot. The effectiveness of visual and audio wayfinding guidance on smartglasses for people with low vision. CHI, 2020.
[8] Dragan Ahmetovic, Cole Gleason, Kris M. Kitani, Hironobu Takagi, and Chieko Asakawa. Navcog: Turn-by-turn smartphone navigation assistant for people with visual impairments or blindness. In *Proceedings of the 13th International Web for All Conference*, W4A '16, New York, NY, USA, 2016. Association for Computing Machinery.
[9] Manaswi Saha, Alexander J. Fiannaca, Melanie Kneisel, Edward Cutrell, and Meredith Ringel Morris. Closing the gap: Designing for the last-few-meters wayfinding problem for people with visual impairments. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '19, page 222–235, New York, NY, USA, 2019. Association for Computing Machinery.
[10] Mikio Kano and Xueliang Li. Monochromatic and heterochromatic subgraphs in edge-colored graphs - a survey. *Graph. Comb.*, 24(4):237–263, sep 2008.
[11] Pei Du and Nirupama Bulusu. An automated ar-based annotation tool for indoor navigation for visually impaired people. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '21, New York, NY, USA, 2021. Association for Computing Machinery.