PERCEPT Indoor Wayfinding for Blind and Visually Impaired Users: Navigation Instructions Algorithm and Validation Framework

Yang Tao, Linlin Ding, Sili Wang and Aura Ganz
Electrical and Computer Engineering Department, University of Massachusetts, Amherst, MA 01003, U.S.A.

Keywords: Navigation, Blind and Visually Impaired, Navigation Instruction Validation, Unity Game Engine.

Abstract: This paper introduces an algorithm that generates indoor navigation instructions for Blind and Visually Impaired (BVI) users as well as a validation framework that validates these instructions. The validation framework which utilizes Unity game engine incorporates: a) a virtual environment that mirrors the physical environment and b) a game avatar that traverses this virtual environment following action code sequences which correspond to the navigation instructions. A navigation instruction from a source to a destination is correct if the game avatar successfully reaches from the source to the destination in the virtual environment following this navigation instruction. We successfully tested the navigation instruction generation algorithm and the validation framework in a large two-story building with 66 landmarks and 1500 navigation instructions. To the best of our knowledge this is the first automated generation instructions algorithm for BVI users in indoor environments, and the first validation framework for navigation instructions in indoor environments. This paper is a significant step towards the development of a cost effective indoor wayfinding solution for BVI users.

1 INTRODUCTION

The World Health Organization reported that 285 million people are estimated to be visually impaired worldwide: 39 million are blind and 246 million have low vision (World Health Organization, 2016). According to the report for the National Health Interview Survey, 22.5 million adult Americans 18 and older reported experiencing vision loss (American Federation for the Blind, 2016).

Independent navigation through unfamiliar indoor spaces is beset with barriers for Blind and Visually Impaired (BVI) users. A task that is trivial and spontaneous for the visioned population has to be planned and coordinated with other individuals for BVI users. Although many improvements and aides are available to assist the visually impaired to lead an independent life, there has yet remain to be a challenge to develop a system that combines independence with scalability and affordability of indoor navigation.

There have been a number of research projects that aim to help BVI users navigate in unfamiliar indoor environments. The authors in (Idrees et al., 2015) introduce an indoor navigation application for BVI users that uses QR codes. In (Riehle et al., 2015) an inertial dead reckoning navigation system is introduced which provided real-time auditory guidance along mapped routes. NavCog (Ahmetovic et al., 2016) is a smartphone-based system that provides turn-by-turn navigation assistance based on real-time localization. StaNavi (Kim et al., 2016) is a navigation system that provides indoor directions utilizing Bluetooth Low Energy (BLE) technology and smartphone's built-in compass. The authors in (Rituerto et al., 2016) report a smartphone-based navigation aid which combines inertial sensing, computer vision and floor plan information to estimate the user's location. ISAB (Doush et al., 2016) is an integrated indoor navigation system utilizing a set of different communication technologies (WiFi, Bluetooth and radio-frequency identification) to help users reach destinations with a smartphone interface. The authors in (Basso et al., 2015) present a smartphone-based indoor localization system using a set of sensors (accelerometer, gyroscope, electronic compass). IMAGO (Jonas et al., 2015) is a vision-based indoor navigation system utilizing smartphone cameras.

PERCEPT system (Ganz et al., 2014), developed at the University of Massachusetts 5G Mobile Evolution Lab was developed with the cooperation of

Tao, Y., Ding, L., Wang, S. and Ganz, A.
PERCEPT Indoor Wayfinding for Blind and Visually Impaired Users: Navigation Instructions Algorithm and Validation Framework.
DOI: 10.5220/0006312001430149
In *Proceedings of the 3rd International Conference on Information and Communication Technologies for Ageing Well and e-Health (ICT4AWE 2017)*, pages 143-149
ISBN: 978-989-758-251-6

the Massachusetts Orientation and Mobility division from Massachusetts Commission for the Blind. PERCEPT is deployed at Arlington subway station in Boston, at Carroll Center for the Blind and at University of Massachusetts administration building. The system was successfully tested with over 50 BVI users. PERCEPT generates landmark based navigation instructions automatically for a specific indoor environment. To the best of our knowledge, PERCEPT is the only indoor navigation system for BVI users that can generate navigation instructions automatically. We will briefly introduce the algorithm in this paper.

Since the number of navigation instructions generated by the algorithm can be very large (e.g. for a building with 80 destinations we generate over 10,000 instructions), the validation of these instructions becomes a challenge. In this paper we introduce a navigation instructions validation tool using Unity game engine (Unity Technologies, 2016). To the best of our knowledge this is the first attempt to validate navigation instructions.

The rest of the paper is organized as follows: In Section 2 we introduce PERCEPT navigation instruction generation algorithm and in Section 3 we present a tool that validates these instructions. In Section 4, we provide a case study that shows the navigation instructions along with their validation process for a large indoor environment. Section 5 concludes the paper.

2 NAVIGATION INSTRUCTION GENERATION ALGORITHM

For paper completeness we briefly introduce PERCEPT system which was reported in (Ganz et al., 2014). In PERCEPT system the user carries a Smartphone (Android or iPhone) that runs PERCEPT application that provides landmark based navigation instructions helping the user navigate through indoor spaces to a chosen destination. PERCEPT includes three main modules: the vision free user interface using Android and iPhone accessibility features, the localization algorithm and the navigation instructions algorithm. For localization purpose we deploy in the indoor environment passive or active radio frequency sensors. The passive sensors use Near Field Communication (NFC) technology and are deployed at specific landmarks in the environment such as doors, elevators, escalators, etc. Using NFC technology the user can be localized at these landmarks only if he/she touches the sensor with the Smartphone. Assuming the user can be localized at each landmark, we introduced a landmark based navigation instructions algorithm.

The input to the navigation instruction generation algorithm includes the Blueprint and the landmarks which include doors, elevators, stairs etc. The output of the algorithm includes the navigation instructions.

The algorithm includes the following stages:

Stage 1: Generation of a directed graph in which the nodes represent the landmarks. Each edge has a weight that is proportional to the physical distance between the two nodes. However, if there is an obstacle between the nodes (landmarks) and/or its unsafe to traverse this edge, the weight of the edge is adjusted to a greater (infinite) value.

Stage 2: Generate navigation routes (shortest paths) between source and destination landmarks running Dijkstra's algorithm on the directed graph generated in Stage 1. Obstacles will be avoided when generating routes automatically since total weights of the routes involving obstacles are higher than the rest. Using the same logic, floor crossing preference, i.e. starts, elevator, escalator, set by users can also be accommodated by adjusting the weights of associated links.

Stage 3: Generate navigation instructions using the navigation routes generated in Stage 2. In collaboration with Orientation and Mobility instructors from Massachusetts Commission for the Blind we developed the following vocabulary that is used in the navigation instructions:

<u>Orientation Adjustment Vocabulary</u>

A. With the tag to your back
B. With the tag to your left/right side
C. Face the tag
D. With the (landmark) to your back
E. With the (landmark) to your left/right side
F. There is (landmark) to your certain direction

<u>Motion Vocabulary</u>

A. Turn right
B. Turn left
C. Continue (Walk straight)
D. Cross (hallway)
E. Walk past (the opening)
F. Walk across (landmark) to your certain direction
G. Go through (landmark)
H. Enter (landmark)
I. Reach (landmark) to your left/right side
J. Scan (NFC) tag to your left/right side

K. Press Prior/Next Instruction Button
L. Press End Journey Button

Travel-by Vocabulary
A. pass by obstacles or landmarks
B. hear certain sound/ smell in the route passing by a certain landmark
C. trail/ follow certain texture (on your left/right side)
D. keep certain landmark to the left/right side

Motion Stop Vocabulary
A. (Proceed) until reaching an intersecting hallway
B. (Proceed) until reaching an intersecting wall
C. (Proceed) until reaching an opening
D. (Proceed) until reaching a different texture (metal gate)
E. (Proceed) until reaching a certain number of doors/ openings/ recessed doorways
F. (Proceed) until passing a certain number of firedoors
G. (Proceed) until reaching a certain landmark
H. (Proceed) until the wall angles out

Floor-cross Vocabulary
A. using elevator – press a certain button for the destination floor
B. using stairs – go up/down certain number of flights of stairs

An example of the navigation instructions is provided in Section 4.

3 VALIDATION FRAMEWORK

In this section we introduce the navigation instruction validation tool (NIVT). NIVT is a new approach to automatically validate landmark-based navigation instructions. It evaluates the navigation instructions using a virtual reality environment by checking and ensuring that each path in the virtual environment can be traversed by following the corresponding navigation instructions generated by the algorithm introduced in the prior section.

NIVT was developed using Unity game engine which is used to create a broad range of interactive 2D and 3D virtual environments which makes it an ideal platform for virtualizing physical indoor environments. NIVT includes three major parts (see Figure 1): 1) a server database that includes the navigation instructions generated by the algorithm introduced in the previous section, 2) a virtual environment that represents the physical space which is traversed by the game avatar according to an action code sequence (see Section 3.1) and 3) an instructions parser which generates this action code sequence from the navigation instructions (see Section 3.2).

We describe below the virtual environment, the instruction parser and the testing process.

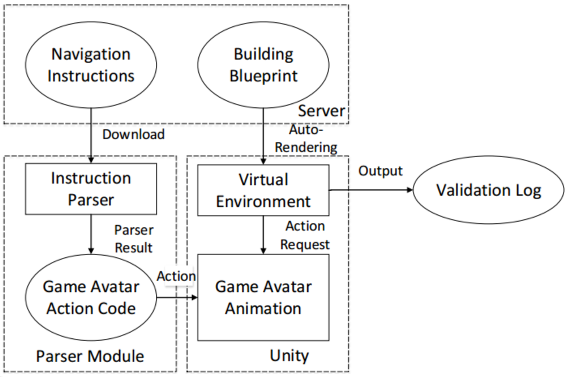

Figure 1: NIVT Architecture.

3.1 Virtual Environment

Using Unity game engine we generate the virtual game environment that represents the indoor physical environment floor plan. In this virtual environment, “tags” are used as markers to indicate landmarks, such as doors, elevators, stairs that are located in the physical space. Textures of landmarks like wood or metal are also rendered and represented in the virtual environment. To build a virtual environment we proceed as follows:

Step 1: Classify objects in target building’s CAD file by their different features (like wall, stairs, elevator, etc.), and save them in different layers;

Step 2: Utilize Microsoft .netDxf API (Microsoft .netDxf, 2016), a third-party library to read and write CAD file, to get 3D location of each objects in different layers;

Step 3: Create corresponding game objects in virtual reality environment with the exact same location in real building’s blueprint with .netDxf tool’s 3D location output;

Step 4: Add collider for each game objects and mark them with different tag according to their features;

Step 5: Download landmarks information from the web browser, convert their geometry location into Unity 3D location, and add landmarks in 3D world.

Figure 2 shows the floor plan of the UMass Campus Center (22,500 sqft), and Figure 3 shows the corresponding virtual game environment.

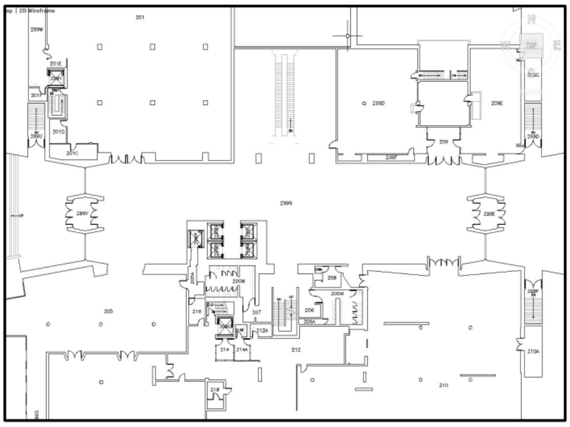

Figure 2: Campus Center Blueprint.

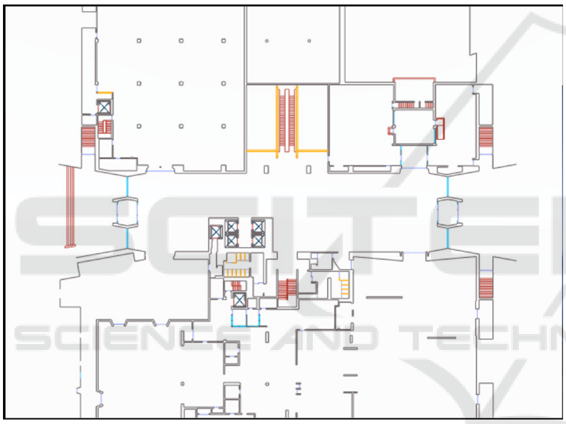

Figure 3: Campus Center Virtual Environment.

3.2 Parser Algorithm

The input to the parser are the instructions generated by the algorithm. For each instruction the parser generates executable action code sequence for the game avatar.

The parser includes the following steps:

Step 1: Retrieve a navigation instruction for a specific source-destination

Step 2: Split instruction sentences into different types. There is a container in the background, which divides all instruction sentences into different types according to the similarity of instructions. Table 1 lists some instruction types.

Step 3: Once the system figures out which type of instruction the current instruction is, it will extract the keyword and determine the action code to execute. These key words are crucial for the game avatar to locate landmarks inside the virtual environment.

A detailed example of the parser is provided in Section 4.

Table 1: Instruction Type and Animation.

Instruction Type	Instruction Sentence	Avatar Animation
Type 1	Turn right/ turn left	Turn right/ left
Type 2	Continue/ Walk straight	Continue/ Walk straight
Type 3	Trail the keyword until reach keyword	Ray cast from avatar to find the nearest landmark Trail the certain landmark to trigger next node
Type 4	Walk across/past the keyword	Walk straight to trigger next node
Type 5	With the keyword to your back/ left/right	Find closest node to avatar Turn avatar back/ left/right to it
Type 6	Walk to keyword direction	Ray cast from avatar to find the landmark at the provided direction Walk straight in that direction to trigger next node
Type 7	You have reached your destination	Finish current task Trigger next task

3.3 Testing

For each source-destination pair *(s,d),* the instruction parser translates the navigation instructions into an action code sequence which is used by the game avatar to traverse the virtual environment from source *s* to destination *d.* If the avatar reaches destination *d* following these actions, the navigation instructions for *(s,d)* are correct. Otherwise, the navigation instructions for *(s,d)* are marked as incorrect. After the navigation instructions are validated for all source-destination pairs we generate a validation log.

We used NIVT to validate all the navigation instructions generated for at the University of Massachusetts, Amherst, Campus Center Building.

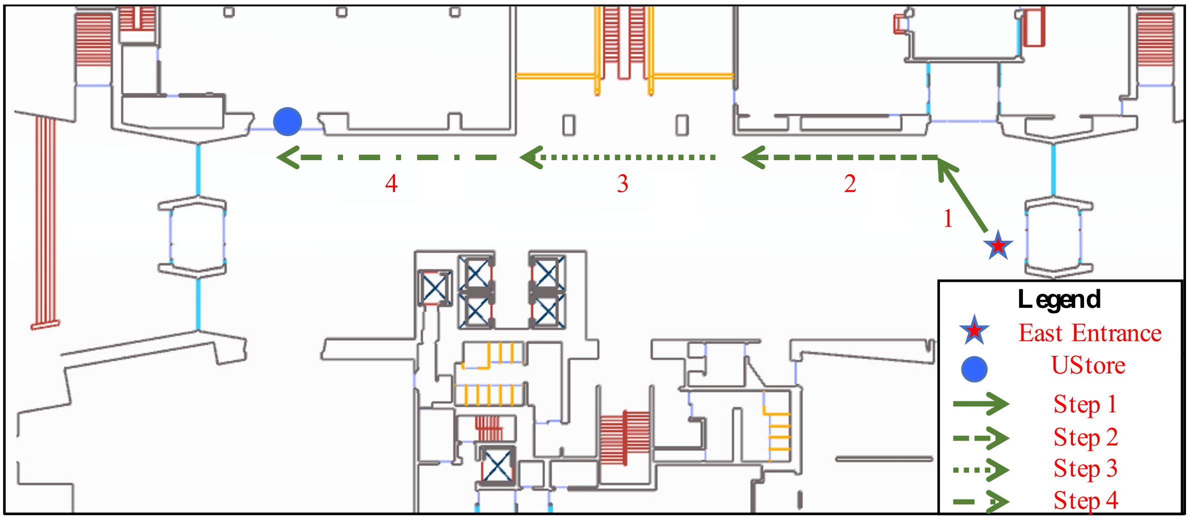

Figure 4: Campus Center Virtual Environment.

This is a two-story building with 66 landmarks and 1500 navigation instructions. NIVT has successfully validated all instructions. We have intentionally modified some of the instructions to render incorrect instructions and they were all have been discovered by NIVT. The test proves the validity of NIVT.

4 CASE STUDY

Now we will use an example to illustrate navigation instructions, how they are parsed by the instruction parser and how they are used by the game avatar to traverse the virtual environment that represents UMass Campus Center Building (See Figure 4). We assume that the source is the East Entrance and the destination is the UStore as marked in Figure 4. The instructions are presented to the user in multiple chunks of instructions. Next, we show step by step how to the system generates and validates each instructions chunk.

Step 1 (Chunk 1) – from East Entrance (Source Point) to Harvest Market

Instructions chunk 1 generated by the navigation algorithm: "*Your current location is: East Entrance. With the East Entrance to your back, there is Harvest Market to your 2 o'clock direction, walk across to the Harvest Market to your 2 o'clock direction, you will reach: Harvest Market. Press next Instruction Button.*"

Action code generated by the instruction parser: "back to East Entrance, walk to 2 o'clock, reach Harvest Market"

The corresponding game avatar traversal is shown in Figure 5. Following the action code, the game avatar will go from East Entrance to Harvest Market at 2 o'clock direction. Then the next chunk of instructions is presented.

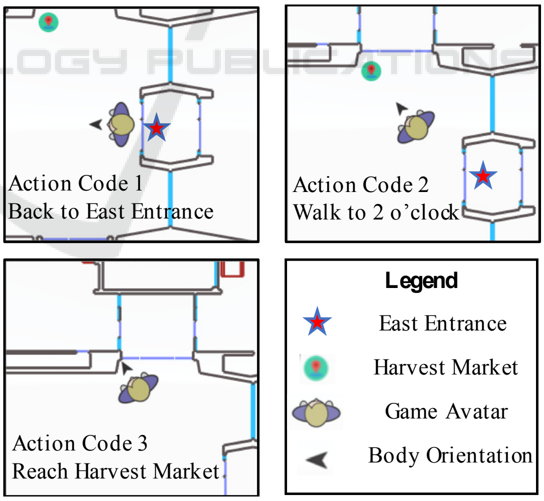

Figure 5: Step 1 Animation.

Step 2 (Chunk 2)– from Harvest Market to opening

Instructions chunk 2 generated by the navigation algorithm: *"Continue, trail the wall on your right side, until you reach an opening. Press next Instruction Button"*

Action code generated by the instruction parser: continue, trail right wall, reach opening

Game avatar traversal is shown in Figure 6. Following the action code, the game avatar will go from Harvest Market to an opening following the wall on its right side. Then the next chunk of instructions is presented.

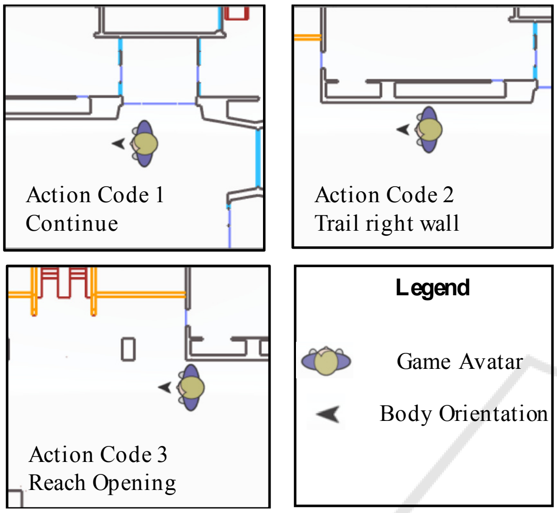

Figure 6: Step 2 Animation.

Step 3 (Chunk 3) – walk past the opening

Instructions chunk 3 generated by the navigation algorithm: *"With the wall to your right, walk past the opening. Press next instruction button"*

Action code generated by the instruction parser: right to wall, walk past opening, stop

Game avatar traversal is shown in Figure 7. Following the action code, the game avatar will walk past the opening and stop at that point. Then the next instructions chunk is presented.

Step 4 (Chunk 4) – from opening to UStore (Destination)

Instructions chunk 4 generated by the navigation algorithm: *"Continue, trail the wall on your right side, until you reach the first door, you will reach: UStore. Press finish button to end the journey."*

Action code generated by the instruction parser: continue, trail right wall, reach door

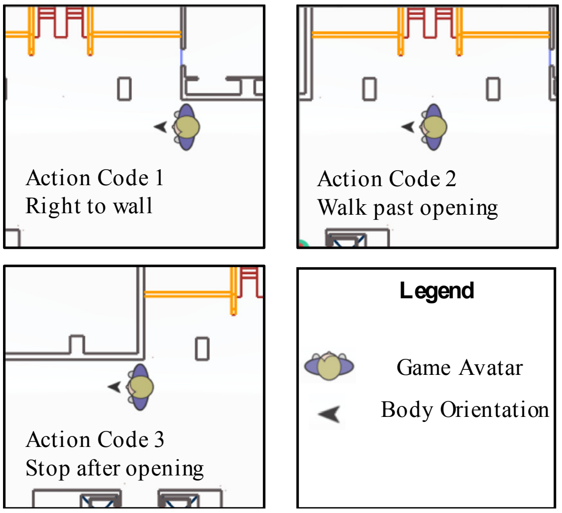

Figure 7: Step 3 Animation.

Game avatar traversal is shown in Figure 8. Following the action code, the game avatar will continue walking and follow the wall on its right side until it reaches a door. Then it will reach the destination, which is UStore.

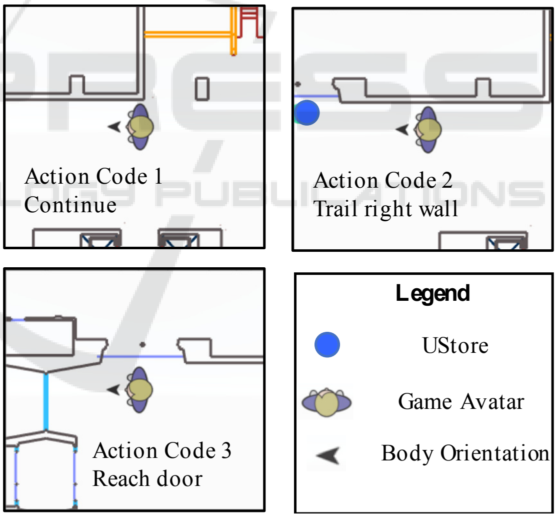

Figure 8: Step 4 Animation.

In this example we have shown that the game avatar has successfully navigated from the source (East Entrance) to the destination (UStore) following the instructions. These navigation instructions from the East Entrance to the UStore will be marked as correct in the validation log.

5 CONCLUSIONS

In this paper we introduced PERCEPT navigation instruction generation algorithm and a validation framework. We have tested the algorithm and validated the instructions in a large building for which we generated 1500 navigation instructions. The results encourage us to proceed to our next research agenda, which includes the development of tools that will enable us to automate PERCEPT deployment and testing in any indoor environment.

ACKNOWLEDGEMENTS

This project was supported in part by Grant 80424 from the Massachusetts Department of Transportation. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Massachusetts Department of Transportation.

REFERENCES

Ahmetovic, D., Gleason, C., Ruan, C., Kitani, K., Takagi, H., & Asakawa, C. (2016, September). NavCog: a navigational cognitive assistant for the blind. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'16)* (pp. 90-99). ACM.

American Fedration for the Blind. (2016). [Online]. Available at: http://www.afb.org/info/blindness-statistics/adults/facts-and-figures/235/. [Accessed 19 Nov. 2016].

Basso, S., Frigo, G., & Giorgi, G. (2015, May). A smartphone-based indoor localization system for visually impaired people. In *Medical Measurements and Applications (MeMeA), 2015 IEEE International Symposium on Medical Measurements and Applications* (pp. 543-548). IEEE.

Doush, I.A., Alshatnawi, S., Al-Tamimi, A.K., Alhasan, B. and Hamasha, S. (2016, June). ISAB: Integrated Indoor Navigation System for the Blind. *Interacting with Computers,* (2017) 29 (2), (pp. 181-202).

Ganz, A., Schafer, J. M., Tao, Y., Wilson, C., & Robertson, M. (2014, August). PERCEPT-II: Smartphone based indoor navigation system for the blind. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 3662-3665). IEEE.

Idrees, A., Iqbal, Z., & Ishfaq, M. (2015, June). An efficient indoor navigation technique to find optimal route for blinds using QR codes. In *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)* (pp. 690-695). IEEE.

Jonas, S.M., Sirazitdinova, E., Lensen, J., Kochanov, D., Mayzek, H., de Heus, T., Houben, R., Slijp, H. and Deserno, T.M. (2015). IMAGO: Image-guided navigation for visually impaired people. *Journal of Ambient Intelligence and Smart Environments*, 7(5), (pp.679-692).

Kim, J. E., Bessho, M., Kobayashi, S., Koshizuka, N., & Sakamura, K. (2016, April). Navigating visually impaired travelers in a large train station using smartphone and bluetooth low energy. *In Proceedings of the 31st Annual ACM Symposium on Applied Computing* (pp. 604-611). ACM.

Microsoft .netDxf. (2016). [Online]. Available at: https://netdxf.codeplex.com/. [Accessed 18 Feb. 2016].

Nakajima, M., & Haruyama, S. (2013). New indoor navigation system for visually impaired people using visible light communication. *EURASIP Journal on Wireless Communications and Networking*, 2013(1), (pp. 37).

Riehle, T. H., Anderson, S. M., Lichter, P. A., Whalen, W. E., & Giudice, N. A. (2013, July). Indoor inertial waypoint navigation for the blind. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 5187-5190). IEEE.

Rituerto, A., Fusco, G., & Coughlan, J. M. (2016, October). Towards a Sign-Based Indoor Navigation System for People with Visual Impairments. *In Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 287-288). ACM.

Tao, Y. (2015). Scalable and Vision Free User Interface Approaches for Indoor Navigation Systems for the Visually Impaired.

Unity Technologies. (2016). [Online]. Available at: https://unity3d.com/. [Accessed 19 Nov. 2016].

World Health Organization. (2016). [Online]. Available at: http://www.who.int/mediacentre/factsheets/fs282/en/. [Accessed 20 Nov. 2016].