Navigation Based Application with Augmented Reality and Accessibility

Sebastião Rocha
Escola Superior de
Tecnologia, IPCB
Campus da Talagueira
Castelo Branco, PT
sebasti3d@hotmail.com

Arminda Lopes
Escola Superior de
Tecnologia, IPCB &
ITI/LARSyS
Campus da Talagueira
Castelo Branco, PT
aglopes@ipcb.pt

CHI 2020 Extended Abstracts, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6819-3/20/04.
DOI: https://doi.org/10.1145/3334480.3383004

Abstract
The implementation of adaptative applications for navigation and ubiquitous interfaces for accessibility defines a new era on spatial computing; especially on solutions that understand reacting to the environment, delivering navigation capabilities with high accuracy and utility for users. Real-time key information and awareness is presented in augmented reality (AR) by using interactive visual elements and auditive instructions. This paper describes a solution running on smartphones by using spatial computing capabilities for positional determination, tracking the user movements over the environment space. Besides the user-friendly minimalistic visual interface, the system is also able to instruct, guiding a completely blind person from a start position into a certain destination, avoiding obstacles. The communication is completely independent and uses voice recognition for accessibility capabilities.

Author Keywords
Accessibility; spatial computing; mobile applications; augmented reality; navigation

CSS Concepts
• **Human-centered computing → Human computer interaction (HCI);** Mixed / augmented reality; Natural language interfaces; • **Human centered computing;** Ubiquitous and mobile computing systems and tools

Google and the Google logo are registered trademarks of Google LLC, used with permission.

Figure 1 - Google Live View enables navigation on cities helping the users on reaching a certain destination by using AR elements, figure courtesy Google and is published at [6]

Introduction

Complex and large-scale building structures can be challenging to navigate and reach specific location spots, especially for outside people, that are not used to the building structure. This problem may occur on shopping malls, airports, museums, government institutions, conferences and hotels, affecting everyone one way or another. Navigation-based applications like the one described on this paper intend to help, guiding users on reaching a certain goal. Recent advances on technological areas related to device hardware capabilities, computer vision, spatial computing, artificial intelligence, and the arrival of accessible platforms for AR development (ARKit [1] and ARCore [2]) lead into a dawn of immersive and spatial aware applications [3]. The goal of uncover and explain key features and technical details about this solution is to advance and spread relevant information related to spatial computing, natural and ubiquitous interfaces, pathfinding algorithms, accessibility and optimization.

Related work

In order to get a better perspective about the solution proposed for personal navigation-based applications with AR features, it is important to list and describe related solutions. In author´s opinion this list includes the best and most promising ones (Table 1). Dent Reality [4] is a complete service suite sharing many features with the paper´s solution, but uses Apple´s indoor mapping algorithms [5] (Wi-Fi based) instead of a visual system. This service allows to map large scale infrastructures, providing navigation instructions using AR and interactive 2D maps. Google Live View [6] is a recently presented feature for the well-known Google Maps service [7], this add-on provides AR elements combined with the traditional interactive 2D maps and navigation directions (Fig. 1). Google combines the traditional global positioning system (GPS) with a visual positioning service (VPS) what works perfectly since Google has already digitalized several cities around the world, providing the needed reference data for the computer vision system. ARWay [8] also provides indoor navigation instructions using AR and is based on a visual system to calculate the device position inside the infrastructure. MobiDev [9, 10] is another alternative that provides indoors navigation using AR, for position determination the system uses GPS, beacons and 2D image triggers spread around the infrastructure. Microsoft [11] have released several research prototypes for indoor navigation using Wi-Fi signal, in the video it is possible to visualize the system outputting the device position on a 2D map, the solution appears to lack on pathfinding navigation and AR elements since they are not covered. This shows that either small or bigger companies like Google, Apple and Microsoft are developing or have already released custom solutions to address this sector, emphasizing the value of this type of applications.

Service	Position determination	Navigation capabilities	AR enabled
Dent Reality	Wi-Fi	Yes	Yes
Google Live View	GPS and VPS (vision based)	Yes	Yes
ARWay	Vision based	Yes	Yes
MobiDev	GPS, beacons, 2D images	Yes	Yes
Microsoft prototype	Wi-Fi	No	No

Table 1 – Related solutions comparison by feature

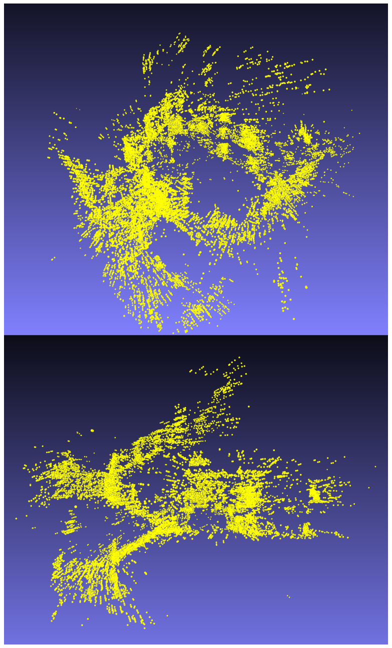

Figure 2 - Mapping results for a stairway environment chunk inside MeshLab software [26], multiple chunks are combined in order to construct large scale AR maps

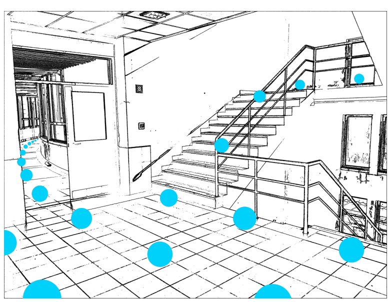

Figure 3 – Waypoint nodes inside a map (blue spheres over the floor)

Solution

The implemented solution covers the mapping and creation of digital maps (digital twins [12]) related to the physical structure where the application will work. This process introduces the term of AR Cloud, a digital platform where the real world is used as a visual anchor for digital information. Applications become spatial aware triggering and displaying specific rich AR elements on top of real-world positions [13, 14].

AR Cloud

The AR Cloud is an upcoming digital platform used to build complete persistence and multi-user shared AR experiences [15, 16, 17, 18]. This platform consists on a dynamic updated 3D map of the real-world (digital twin) used as a reference to place digital contents over specific real-world positions. Developers and investors already argue this platform will become the single most important software infrastructure in the computing industry, where location-based information is anchored into real-world positions with millimetric precision. Developers will be able to position and pin any type of information on specific places over the world, such as images, text, 3D objects or notification triggers. Instead of using traditional GPS coordinates and Bluetooth signals to fix content into a certain position [19], which translates on low precision and limited functionality, the AR Cloud will use visual features to track and figure out the position and orientation of the device compared to real-world reference [20, 21, 22].

Mapping process

As refereed before, physical structures need to be captured into a digital form, usually a group of complex and elaborated point clouds (chunks or clusters of point cloud data) used as reference. Point clouds can be acquired using several methods, ranging from LIDAR systems [23], time-of-fligh (ToF) camera sensors [24] or simply by using smartphone camera sensors (RGB) widely available nowadays. This academic project uses the smartphone camera method, providing a suitable result for an accessible price tag. During the mapping phase, the system uses the sensors present on the device (camera and the inertial measurement (IMU) [25]) combined with the device motion, extracting the point cloud result in real time. Fig. 2 shows a point cloud chunk related to a slice (stairway) of an AR map (entire building block), inside MeshLab software [26].

Contextual elements on a map

After capturing multiple point cloud chunks using an iPhone X and iPad Pro, the result information is transferred into a development computer. The point cloud chunks are then assembled, transformed, optimized and even cleaned for better tracking performance, addressing possible errors captured during the mapping phase, related to movement accumulated drift and poor visibility due to camera sensor limitations. These point cloud data chunks are used as an extensive anchor for that environment in order to place digital contents associated to the application. The current solution uses two types of objects, waypoint nodes and destination nodes.

Waypoint nodes

Waypoint nodes illustrated on Fig. 3 are used to describe walkable positions, where the user is able to navigate. This combined with a custom designed A* algorithm enables accurate pathfinding navigation from a start node into a destination node. Currently a waypoint node only stores a 3D vector position, related to the x, y and z coordinates inside the digital map.

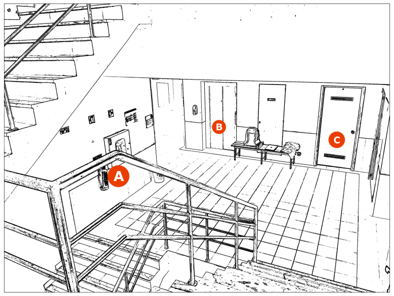

Figure 4 - Destination nodes inside a map (red spheres), A is related to a fire extinguisher, B is an elevator and C is a storage room

Figure 5 - Some of the default icons used (elevator, bedroom, store, kitchen/restaurant, exit/emergency, trash can)

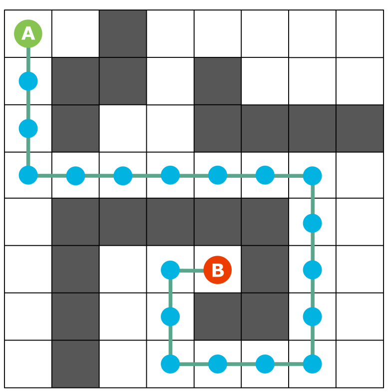

Figure 6 - A* pathfinding case example, the figure shows a path from point A to point B, middle points are waypoint nodes. The algorithm uses a grid map (white squares) and excludes obstacles (gray squares)

DESTINATION NODES

Destination nodes illustrated on Fig. 4 are related to marked points of interest (POI) inside the environment, usually they consist on rooms and spots, or even specific objects, like products in a supermarket, providing even more detail into the digital maps. The destination nodes store a position in space as a 3D vector position like the waypoint nodes. Additionally, destination nodes also store a name label, a tag value for algorithm searching purposes, and finally an image data (Fig. 5). The associated image data is usually an icon that provides better understanding for the final user, as it can be helpful on a shopping center, where each store will be better identified using the brand icon instead of using just the brand name.

Storing and accessing the digital maps

After assembling the digital map, composed by several chunks for better performance, it is essential to create a way of distributing and retrieving the map over the internet for any device at any time, meaning that anyone will be able to download and use the digital map for that specific infrastructure. Currently the digital map data is stored on a cloud database, where it can be easily accessed when triggered by the application. There are several ways to trigger the usage of a certain map, the most basic includes using an identifier (2D image or QR code [27]) that triggers the download of a specific map, however this requires having multiple identifiers located over the building structure. A much clever way and completely invisible for the user was implemented using location identifiers (ID) [28]. Location IDs are used to trigger and load specific map chunks, each location ID is associated with the Wi-Fi´s service set identifier (SSID) [29]. Since Wi-Fi networks are stable and static in space, this method associates the point cloud chunks to specific Wi-Fi network access points (APs) [30], in case Wi-Fi is not available (outdoors) the system uses GPS coordinates instead, providing less accuracy but very similar results.

Loading a map

When a chunk is triggered, the application will check if it has already the associated map data (chunks plus contextual elements) stored locally on disk. If not, the application will request and download the map from the cloud database, storing the data locally, avoiding the need of downloading the same map again in the future, saving resources and network bandwidth. A version check is also enabled, comparing the local version to the cloud version (latest), if the local version is outdated this will force the download of the newest version. After a map is loaded into memory, the system will now try to relocalize it on the correct position. The process of relocalize a map chunk means the application will try to correctly place it over the real-world environment, this process uses the device camera to scan the environment and try to find similar feature points between the current view and the reference data (chunk). This process can take a big performance hit, especially with bigger map chunks and/or with less capable hardware (low-end and mid-range smartphones). It is expected that the application relocalizes with success after some seconds, populating the environment with all contextual elements inside the building structure. The final result is a seamless experience requiring no interactions from the user.

User navigation algorithm

Once the system loads and relocalizes the digital map related to the environment around the user, the application is now capable of calculating the best path

$$f(n) = g(n) + h(n)$$

Equation 1 - A* heuristic function

$$h = sqrt(\Delta x + \Delta y + \Delta z)^2$$

$$\Delta x = (n.x - destination.x)^2$$

$$\Delta y = (n.y - destination.y)^2$$

$$\Delta z = (n.z - destination.z)^2$$

Equation 2 – 3D Euclidean distance used as approximated h cost

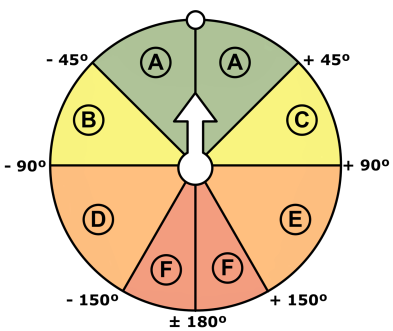

Figure 7 - Navigation instructions, on the area identified as A, users will be instructed to walk forward, on B and C areas users will be instructed to turn on a specific direction (left or right), on D and E areas users will be instructed to turn quite on a specific direction (left or right) and finally on the F areas users will be asked to turn back, meaning the node is on the opposite direction

from a start node to a destination node. In this case, the shortest path between the two nodes (Fig. 6). This is an essential functionality required for navigation-based applications using a custom-made A* pathfinding algorithm, that results in enhanced performance even for larger maps.

A* PATHFINDING ALGORITHM

When triggered, the A* algorithm creates a tree of paths originating from the start node and extends those paths one at a time until it eventually reaches the destination node having the smallest cost (meaning the least distance travelled) [31, 32, 33, 34, 35, 36, 37]. At each iteration, the algorithm uses the following heuristic function (Equation 1) in order to select which of the available options will it prioritize to extend and search. This way the algorithm selects the node that has the lowest f value, meaning the cost of selecting that node (n) outperforms the other available options. This method prioritizes the best paths, resulting in less computational cycles leading to a better overall performance. The heuristic function sums the g and h values, where g is the cost of the path from the start node to the current node (n), and h is the estimated cost of the cheapest path from the current node (n) into the destination node. There are several ways to calculate the h cost value, by using exact methods or approximated methods. Exact methods provide slightly better results when compared to approximated methods, but they are very demanding to calculate which leads into slower performance and heavy battery consumption. By using approximated methods, it is possible to get a near real-time calculation of h cost value without using so many processing (computational cycles) and resources (system memory), and the results will be almost as good as using the exact methods. There are several techniques of calculating the approximated h cost, currently the algorithm uses the 3D Euclidean distance from node n to the destination node, seen on Equation 2 to calculate the approximated h cost value. On each iteration, the algorithm stores on the current node its predecessor node, this way it is possible to determinate the sequence of steps inside the calculated path. The A* algorithm will finish with success, resulting in an output list containing the start node, a couple of nodes in between and end up with the destination node, the application will now be able to know how to instruct the user from the start node to the destination node (goal).

Navigation instructions

One of the main features present on the current system is the ability to continuously track the position and orientation of the device by using 6-DoF [38] SLAM [39] tracking provided by ARKit and ARCore platforms. By understanding and knowing the user position and the path required to reach a certain destination, it is possible to guide the user over the calculated path, providing real-time instructions based on the user motion (walking and turning), adapting the path continuously over time. Fig. 7 illustrates the possible instructions used to guide the user, there are two types of movement outputs, the move action related to walk forward and the turn action used when the user needs to turn on a specific direction. These instructions will be triggered dependent on the orientation angle of the user (center arrow in Fig. 7) related to the target node (white circle on top of Fig. 7). For a better interface, the system also uses haptic feedback, triggering the vibration motor of the device when the calculated angle related to the target node is near zero (user is facing the correct direction).

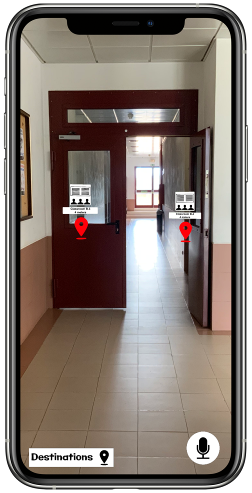

Figure 8 – Main interface screen, this screen shows near POI using visual AR elements

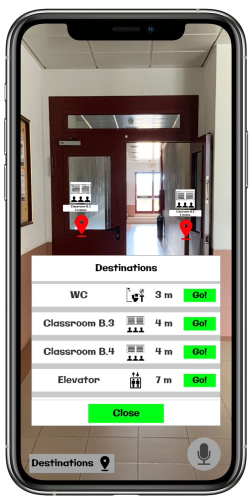

Figure 9 – List available destinations interface screen

Interface

As mentioned before the application system requires a well-planned interface for better accessibility and user-friendliness. Therefore, the application will have a base interface composed of visual elements drawn on the screen where the user is able to tap and control (Fig. 8), nicknamed as a traditional interface with AR elements. Additionally, the application will also enable an auditive interface that works seamlessly, succeeding on replacing the traditional visual interface, but in this case, it is specifically designed for blind and low vision users. This natural interface has voice recognition and uses auditive instructions for communication feedback. Both interfaces (visual and audio) will work in parallel and enable the same level of features, resulting in an adaptative multi-modal interface for the final user.

To keep the application simple and effective two main actions are available (List available destinations and Go to destination), but more actions can be added easily in the future dependent on the client needs, support for more languages can also be added easily, at the moment only English and Portuguese are being used.

List available destinations

Users will be able to know about available destinations inside the digital map (building structure), by tapping on the "Destinations" button located on the lower left (Fig. 8) or simply by saying something related to, "List destinations."; "Tell me about destinations." or "Where can I go?", this phrases will trigger the action. In either case, the system will list available destinations around the user, via visual elements by drawing a list containing the destinations (Fig. 9) or by outputting the destination names via audio for the user.

Go to destination

Users can select a destination from the available destinations list (Fig. 9), or they can say something related to, "I want to go to the [destination name]." or "Lead me to [destination name].". When the application receives a valid destination, the user navigation algorithm will be triggered calculating the best path and instructing the user on how to get there (Fig. 10).

Figure 10 - Navigation interface, the application draws the path using AR elements and dictates navigation instructions

Conclusions

This paper demonstrates the usefulness of spatial computing applications, with the use of ubiquitous interfaces leaving a positive impact on user's everyday life, special on blind persons. The presented solution is a low budget navigational system in its early phases of design and implementation, intending to draw a path for future applications and ecosystems related to these themes, specially the use and implementation of AR cloud platforms. It is essential to continue testing and improving the design and accessibility of this type of applications as it requires time and resources to analyse and study the behaviour and needs of users.

References

[1] Apple, "ARKit," 2019. [Online]. Available: https://developer.apple.com/arkit/.

[2] Google, "ARCore," 2019. [Online]. Available: https://developers.google.com/ar.

[3] M. Miesnieks, "Dawn of the AR Cloud," 2018. [Online]. Available: https://medium.com/6d-ai/dawn-of-the-ar-cloud-1b31eb4b52ac.

[4] Dent Reality, "Dent Reality," 2019. [Online]. Available: https://www.dentreality.com.

[5] Apple, "Indoor Mapping Data Format," 2019. [Online]. Available: https://register.apple.com/resources/imdf/.

[6] Google, "Introducing Live View, the new augmented reality feature in Google Maps," 2019. [Online]. Available: https://support.google.com/maps/thread/11554255?hl=en.

[7] Google, "Google Maps," 2019. [Online]. Available: https://www.google.com/maps.

[8] ARWay, "ARWay," 2019. [Online]. Available: https://www.arway.app.

[9] Mobidev, "How Augmented Reality-based Indoor Navigation Works," 2019. [Online]. Available: https://mobidev.biz/blog/augmented-reality-indoor-navigation-app-developement-arkit#1.

[10] Mobidev, "ARKit based AR indoor navigation in corporate campus," 2018. [Online]. Available: https://youtu.be/VmROm6nbElA.

[11] Microsoft, "Indoor Navigation," 2016. [Online]. Available: https://youtu.be/yLTV_ZOFlWc.

[12] Wikipedia, "Digital Twin," [Online]. Available: https://en.wikipedia.org/wiki/Digital_twin. [Accessed 2020].

[13] J. Boyle, "First steps into an AR Cloud: Relocalization," 2018. [Online]. Available: https://medium.com/@3d_jb/first-steps-into-an-ar-cloud-relocalization-58d04ea93aa.

[14] J. Boyle, "Sharing in AR, Today and Tomorrow," 2018. [Online]. Available: https://medium.com/@3d_jb/sharing-in-ar-today-and-tomorrow-da285881cd65.

[15] A. Strange, "The Future of the AR Cloud — a Thousand Walled Gardens Bloom," 2019. [Online]. Available: https://next.reality.news/news/future-ar-cloud-thousand-walled-gardens-bloom-0196249/.

[16] K. Kelly, "AR Will Spark the Next Big Tech Platform—Call It Mirrorworld," 2019. [Online]. Available: https://www.wired.com/story/mirrorworld-ar-next-big-tech-platform/.

[17] O. Inbar, "Mirrorworld v. AR Cloud or: How I Learned to Stop Worrying and Love the Spatial Future," 2019. [Online]. Available: https://medium.com/super-ventures-blog/mirrorworld-v-ar-cloud-or-how-i-learned-to-stop-worrying-and-love-the-spatial-future-59de8fe8538f.

[18] T. Parisi, "Building the Mirrorworld," 2019. [Online]. Available: https://medium.com/@tonyparisi/building-the-mirrorworld-f1f978c3eae6.

[19] J. Boyle, "Global Relocalization: A Better GPS," 2018. [Online]. Available:

https://medium.com/@3d_jb/global-relocalization-a-better-gps-b6b98d2db20c.

[20] O. Inbar, "AR and Blockchain: A Match Made in The AR Cloud," 2018. [Online]. Available: https://medium.com/super-ventures-blog/ar-and-blockchain-a-match-made-in-the-ar-cloud-7b10c52faddb.

[21] O. Inbar, "The AR Cloud is Making it Rain!," 2018. [Online]. Available: https://medium.com/@oriinbar/the-ar-cloud-is-making-it-rain-66c4c0f164f9.

[22] M. Miesnieks, "Why AR Apps will struggle for engagement without the AR Cloud," 2017. [Online]. Available: https://medium.com/6d-ai/why-is-arkit-almost-useless-without-the-arcloud-6ee1e7affc65.

[23] Wikipedia, "Lidar," 2019. [Online]. Available: https://en.wikipedia.org/wiki/Lidar.

[24] Wikipedia, "Time-of-flight camera," 2019. [Online]. Available: https://en.wikipedia.org/wiki/Time-of-flight_camera.

[25] Wikipedia, "Inertial measurement unit," 2019. [Online]. Available: https://en.wikipedia.org/wiki/Inertial_measurement_unit.

[26] MeshLab, "MeshLab," 2019. [Online]. Available: http://www.meshlab.net.

[27] Wikipedia, "QR code," [Online]. Available: https://en.wikipedia.org/wiki/QR_code. [Accessed 2020].

[28] 6D.ai, "AR Cloud Guide," 2019. [Online]. Available: https://developer.6d.ai/user/dashboard/?view=ar_cloud_guide.

[29] Lifewire, "What Is a Service Set Identifier (SSID)?," 2019. [Online]. Available: https://www.lifewire.com/definition-of-service-set-identifier-816547.

[30] Wikipedia, "Wireless access point," [Online]. Available: https://en.wikipedia.org/wiki/Wireless_access_point. [Accessed 2020].

[31] GeeksforGeeks, "A* Search Algorithm," 2017. [Online]. Available: https://www.geeksforgeeks.org/a-search-algorithm/.

[32] Wikipedia, "A* search algorithm," [Online]. Available: https://en.wikipedia.org/wiki/A*_search_algorithm.

[33] GeeksforGeeks, "A* Search Algorithm | GeeksforGeeks," 2017. [Online]. Available: https://youtu.be/vP5TkF0xJgI.

[34] Computerphile, "A* (A Star) Search Algorithm - Computerphile," 2017. [Online]. Available: https://youtu.be/ySN5Wnu88nE.

[35] A. Patel, "A* Pathfinding," 2019. [Online]. Available: http://theory.stanford.edu/~amitp/GameProgramming/.

[36] Amit Patel, "Introduction to the A* Algorithm," 2019. [Online]. Available: https://www.redblobgames.com/pathfinding/a-star/introduction.html.

[37] S. Lague, "A* Pathfinding (E01: algorithm

explanation),” 2014. [Online]. Available: https://youtu.be/-L-WgKMFuhE. [Accessed 2019].

[38] Wikipedia, “Six degrees of freedom,” [Online]. Available: https://en.wikipedia.org/wiki/Six_degrees_of_freedom. [Accessed 2019].

[39] Wikipedia, “Simultaneous localization and mapping,” 2019. [Online]. Available: https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping.