Cross-Platform Immersive Visualization and Navigation with Augmented Reality

Akshay Ayyanchira
aayyanch@uncc.edu
University of North Carolina
Charlotte, NC, USA

Elias Mahfoud
emahfoud@uncc.edu
University of North Carolina
Charlotte, NC, USA

Weichao Wang
weichaowang@uncc.edu
University of North Carolina
Charlotte, NC, USA

Aidong Lu
aidong.lu@uncc.edu
University of North Carolina
Charlotte, NC, USA

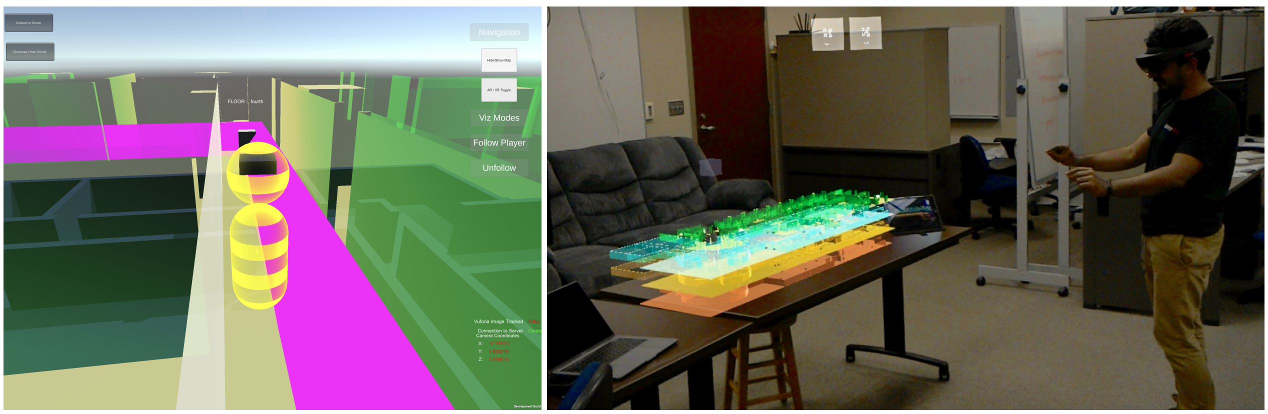


Figure 1: Immersive visualization and navigation with mobile devices (iPad) and AR HMDs (HoloLens)

ABSTRACT

Navigation and situation awareness are amongst the most important aspects of collaborative analysis in 3D environments. This paper investigates the latest technology of mixed reality to improve the team navigation through effective real-time communication. We develop a cross-platform collaboration system, supporting different types of devices and operating systems, including Microsoft HoloLens and iOS devices. Our system provides essential coordination and information sharing functions by leveraging on device sensors and Vuforia API of image marker to localize users inside a building. We provide a set of essential building navigation, visualization, and interaction methods to support joint tasks in the physical building environments among participants with mobile devices in real-time. We have performed a user study to evaluate different devices used in coordination tasks. Our results demonstrate the effects of immersive visualization for improving 3D navigation and coordination.

CCS CONCEPTS

• **Human-centered computing → Mixed / augmented reality**; *Visualization techniques.*

KEYWORDS

Immersive visualization, 3D Navigation, Smart Building, Augmented Reality

ACM Reference Format:
Akshay Ayyanchira, Elias Mahfoud, Weichao Wang, and Aidong Lu. 2021. Cross-Platform Immersive Visualization and Navigation with Augmented Reality. In *The 14th International Symposium on Visual Information Communication and Interaction (VINCI '21), September 6–8, 2021, Potsdam, Germany.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3481549.3481564

1 INTRODUCTION

Since 1968 by Ivan Sutherland [23], extensive research has been conducted on the field of augmented reality (AR) [1, 4, 15, 26]. However, the technology was often used in research or prototype systems instead of practical applications. Only until recent years, wearable technology has started to achieve tremendous advancements in many aspects [6]. Such advancements enabled the use of AR in a

VINCI '21, September 6–8, 2021, Potsdam, Germany

ACM ISBN 978-1-4503-8647-0/21/09…$15.00
https://doi.org/10.1145/3481549.3481564

myriad of fields, including gaming, education, architecture, facilities management, emergency response and military operations, and many others [19, 20]. The concept of mixed reality (MR) has also been proposed to represent advanced technologies for blending the physical world with the digital world.

The trend of MR can also be integrated with major improvements from other fields, especially smart cities, internet of things (IoT), and big data techniques. New technologies can be built from these interdisciplinary research fields and revolutionize how we work and collaborate in many professions [18]. For example, emergency responders such as firefighters and police can utilize all available information collected through smart devices and sensors to understand building situations in real-time. Advanced machine learning and data analysis algorithms can assist commanders to derive important decisions for operations in the field, such as if the crews should enter a room with potential hazards. The crews can also share information with others for more efficient collaboration when performing time-critical jobs on-site. Such techniques may significantly improve job performances, but requiring efficient data sharing and coordination solutions.

This work presents a mixed-reality platform which supports the coordination of multiple users and improves their situation awareness in smart buildings. Our MR platform utilizes mixed-reality devices including AR HMD HoloLens and iOS devices to assist indoor operations such as building navigation and search. We provide a flexible server-client architecture supporting two-way communication, which synchronizes user locations and vital information in real-time. We have also designed several building navigation, visualization and interaction functions for users to access real-time information and perform investigation on-site conveniently. Our examples and user study demonstrate that the prototype system supports the team coordination and navigation tasks and enhances the situational awareness. The results also suggest that such immersive functions are promising for challenging situations including operations in buildings with complex structures or time-critical navigation tasks.

The remaining paper is organized as follows. We first review the related work in Section 2. Section 3 presents our system architecture that uses a centralized server to connect cross-platform devices. Section 4 describes our immersive visualization and navigation system and Section 5 presents the interaction functions that enable users to perform several common navigation, sharing, and exploration tasks. Section 6 summarizes our user study and Section 7 concludes the paper with future work.

2 RELATED WORK

We review the latest work from the fields of augmented reality, collaboration, and cross-device techniques as important components of new human-computer interfaces for smart buildings.

2.1 Augmented Reality for Smart Buildings

While augmented reality has been proposed for over 20 years, its applications on smart buildings are not common due to the lack of suitable devices and services. Back to 1997, Feiner et al. [10] presented a system which used augmented reality to provide information about the university campus. Information was overlaid through a head-mounted see-through display and a hand-held device. The system used GPS to determine position, accelerometer to determine head pitch and roll, and a magnetometer to determine head yaw. Tache et al. introduced a similar system [25] in 2012 as a training tool. Their system reported operative locations in the field to the command center, and displayed that information on a calibrated CAD model. Another similar system was designed by Huang et al. [12] in 2016. This system, called ARGIS, introduced a precise registration method for displaying GIS data in augmented reality. The precise registration method required user input through a hand-held device to guide the calibration process and compensate for measurement errors in position and orientation. All of the aforementioned systems are designed to work outdoors and hence the reliance on GPS data.

For indoor systems, Irizarry et al. presented InfoSPOT [13], an AR system with available devices such as an iPad. The system displayed Building Information Model (BIM) to show infrastructure and other information overlaid in the camera view of an iPad. Predefined markers were placed into the environment and registered in the BIM database, along with their position and orientation. The user scanned a marker image that allowed them to identify which area they were located in and view that environment through the device. Most of these projects relied only on reading data from a centralized database. Some of them allowed for interactive data authoring from the server side, but operatives in the field were still unable to communicate back, other than reporting their location.

2.2 Collaboration in AR/VR Environments

Augmented reality and virtual reality have both been used to create various collaboration environments, which allow users to share and interact with virtual objects in real time. Immersive environments have been used to create unique collaborative experiences [1, 4, 14].

According to a recent survey [14], collaboration is an important topic in AR. Both co-located and remote collaboration systems have been developed. For example, an augmented-reality collaboration environment was presented in [24], where a stereoscopic display was projected onto a see-through display, allowing multiple users to simultaneously view the same spatially-aligned model. Benko et al. [3] supported multiple users to explore scaled and full-size representations of an archaeological dig. Nilsson et al. [17] presented a co-located AR system for supporting joint planning tasks by providing shared organization-specific views for police and military personnel. Dong and Kamat [8] introduced a tabletop fiducial AR system for co-located collaboration. Butscher et al. presented an approach of AR above the tabletop (ART) for collaborative analysis of multidimensional data [5]. More recently, a collaboration service based on Microsoft HoloLens was published by Object Theory [27]. It allowed multiple people to view and interact with models while seeing virtual avatars of each other, making it easy to point at objects and communicate remotely.

2.3 Cross-Device Techniques

AR developers face a proliferation of new platforms, devices, and frameworks, leading to new applications and techniques created with cross-device approaches. Similarly, ubiquitous analysis methods integrating different devices and frameworks were also proposed [9]. For example, the GraSp approach [16] demonstrated that spatially-aware mobile displays and a large display wall can be coupled to support graph visualization and interaction. Horak et al [11] presented the combination of smartwatches and a large interactive display to support visual data analysis. Butscher et al. [5] combined a touch-sensitive tabletop and AR headsets to visualize clusters, trends, and outliers in multidimensional data. Recently, Speicher et al [21] developed a taxonomy of AR system components and identified key challenges and opportunities in making them work together through a review of existing AR platforms and a survey of 30 AR designers, developers, and users.

Cross-device techniques often need to handle the networking and communication among multiple devices. For example, VisHive [7] constructed web-based visualization applications that can transparently connect multiple devices. Similarly, our approach contains a client-server networking component to stream data from server to connected clients and vice versa on press of a button.

3 CROSS-PLATFORM IMMERSIVE SYSTEM

3.1 System Architecture

The design of our system is to support multiple users to perform 3D navigation tasks through immersive systems running on different devices. As shown in Figure 2, mobile devices including HoloLens and iOS devices can communicate with each other to support joint tasks. Therefore, our prototype system consists of a server component allocated at a center and individual interfaces for clients supported by different devices. The clients can synchronize their locations in a building, perform on-site investigation and operation, and share information with others. The main purpose of the server is to gather information about the connected clients and distribute the information to them.

Our system architecture supports smooth communication between the center and clients, including the real-time locations of the clients and various building information. Operatives at the center can also issue instructions to describe changes in their environment, for example marking a location as a new target or requesting rally point at a location they see. The clients can also navigate through the building with accurate 3D CAD models and real-time locations and gaze directions of clients, and create detailed target locations and paths for the clients in the field to follow. The following describes four important components of our system respectively.

3.2 Support for Multiple Devices

Our prototype system is built using the Unity3D engine. While this is a general framework, our prototype system supports two types of devices. The following lists the libraries we use for each type of devices.

HoloLens – The HoloToolkit.Sharing library allows applications to span multiple devices, and enables holographic collaboration. HoloToolkit.Sharing enables multiple devices to communicate and stay in sync seamlessly in real time. Users can also collaborate with other users who may be in the same room or working remotely.

IOS devices – ARKit 2 allows developers to integrate shared experiences, persistent AR experiences tied to a specific location, object detection and image tracking to make AR apps even more dynamic. Shared experiences with ARKit 2 allow multiple users to play a game or collaborate on projects like home renovations.

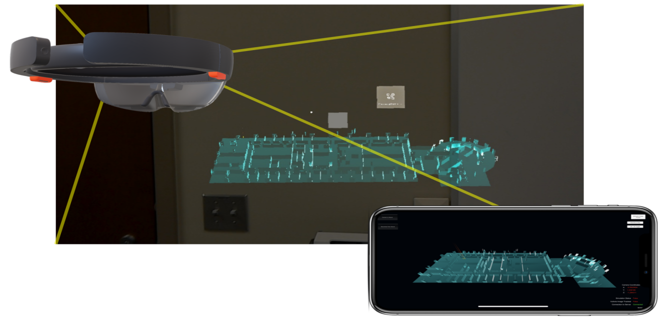


Figure 2: Application Design for Hololens and iOS devices being used in the immersive navigation system.

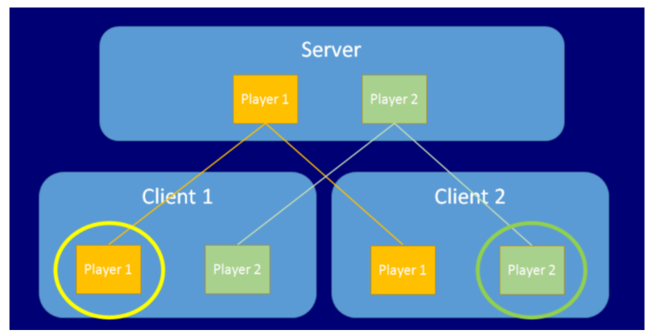


Figure 3: Unet Networking Architecture

3.3 Networking

Our networking component uses UNET, Unity's high-level networking API (HLAPI). Unet networking offers two primary modes to establish networked projects, peer to peer based projects or a single server. Figure 3 (referred from Unity's explanation Unet documentation) demonstrates how two clients connect with server and have their properties reflected on other client instances. The single server approach is used in this project to keep the components isolated and modular. Unet communicates with connected components using MessageBase Class. The project uses two such custom classes inherited from MessageBase class, one for client device status sent from client to server and the other for server to broadcast information to connected clients.

3.4 Coordination of Locations

While the system architecture offers many automatic synchronization features, it assumes that the server and all clients share the same coordinate system. However, the different location tracking mechanisms prevent direct use of UNET's location synchronization.

To resolve this problem, a common coordinate system must be used between the server and all connected clients. Several approaches can be followed to establish a common coordinate system.

While each device supports a spatial anchor created by the first connected client as the common coordinate system, it is not possible to associate the spatial anchor with a 3D CAD model. Instead, we choose a flexible approach through an external, predefined location that has a corresponding point in the CAD model as the common coordinate system.

Specifically, we use Vuforia to locate this external predefined point. Vuforia is an augmented reality library which uses computer vision to locate predefined images in 3D space. It offers strong integration with the Unity engine, and its latest version supports HoloLens devices. When a new device starts the application, the offline Unity scene is loaded. The scene has an active Vuforia camera configured to locate a predefined pattern (Vuforia target). When that target is located within the camera view, the event handler computes the camera offset and connects to the server, while also loading the online Unity scene.

Several factors impact coordinates when loading a new scene, affecting the way the camera offset is computed in the new coordinate system. First, the device world position is reset to $(0, 0, 0)$, while the rotation is only reset around the Y axis since each device can compute the correct X and Z rotations from the accelerometer. This means that to compute the initial camera coordinates in the common coordinate space, the position and rotation of the camera have to be computed differently. Let T_{cam} be the transform of the device camera in its local coordinate system when the image target is detected, and let T_{img} be the image target's transform in the local coordinate system. The position offset of the camera is computed as follows:

$$T_{offset} = T_{img}^{-1} T_{cam}$$

The translation component of T_{offset} is the initial camera position in the common coordinate system. To compute the initial camera rotation in the common coordinate system, only the rotation around the y axis is needed. Let Z'_{cam}, Z'_{img} be the projections of the Z vectors of T_{cam} and T_{img} onto a horizontal plane, respectively. The rotation around the Y axis can be calculated by simply computing a quaternion q that rotates from Z'_{img} to Z'_{cam} as follows:

$$q_{xyz} = Z'_{img} \times Z'cam \quad (1)$$

$$q_w = Z'_{img} . Z'_{cam} + \sqrt{len(Z'_{img})^2 len(Z'_{cam})^2} \quad (2)$$

Note that the square length of the vectors is used since it is faster to compute, then the square root of their multiplication is taken.

The computed transform is relative to the image target's transform in the CAD model. To compute the final initial offset for a crew, the server multiplies the relative offset by the corresponding image target's transform in the CAD model. The resulting final offset is sent back to the client to be used to translate coordinates between its local coordinate system and the common coordinate system.

3.5 Synchronizing Locations in Real-Time

After a client connects to the server and receives its absolute offset in the common coordinate system, synchronizing pose is relatively straightforward using UNET. It is not possible to directly use UNET's NetworkTransform to automatically synchronizes the pose of each player; however, it is possible to implement a custom network transform synchronization logic which utilizes lower level constructs such as syncvars. The custom network transform uses syncvars to synchronize local position and local rotation independently. It only sends changes to the server if they exceed a configurable threshold. Whenever the threshold is exceeded, the custom network transform converts the coordinates into server space and sends them using a command. The server then automatically synchronizes the changed syncvars across all clients. With a syncvar hook, clients are able to convert received coordinates into their local space to show objects at the correct locations.

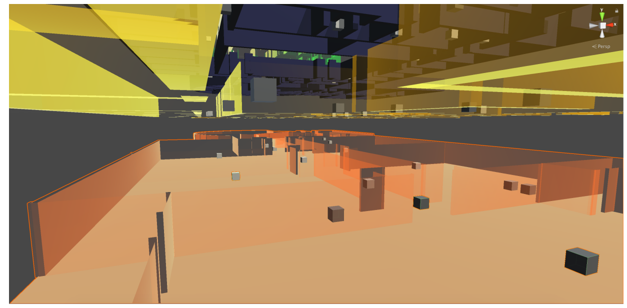


Figure 4: We add location points in a 3D Building structure to assist the navigation tasks.

It is also worth noting that the custom network transform also interpolates the position and rotation between received coordinates to achieve smooth animation. This custom network transform is used to synchronize the location of all individual objects (all characters representing operatives and all targets).

4 IMMERSIVE VISUALIZATION AND NAVIGATION

4.1 Immersive Building Navigation

To enhance the 3D navigation capability, we built a virtual navigation system. The functions are similar to basic GPS navigation systems. The navigation system is built with node-points spreading across entire building. Each node represents either a turn point, or a point of interest which can be an office, classroom, conference room, laboratory, or restroom etc. A node is modeled as a GameObject in Unity's hierarchy, placed inside the model of the building as shown in figure 4. As described below, there are 5 stages involved in navigation system.

4.1.1 Node Graph Generation. To make this application re-usable for new building models, we generate the node graph in a semi-automatic manner. Each building model has its own graph file, which is a text file that denotes the reachability from each node. For example, if there are three nodes in a building A, B and C, there will be three lines in the text file:

A C
B C
C A B

The first letter of each line denotes the starting node and the rest of the characters in the line tells if they are reachable from the starting node. Here C is reachable from A, and also B is reachable from C.

A and B cannot reach each other directly. A route graph is then automatically generated.

4.1.2 Finding the Shortest Path. Once the graph is generated, we also assign weights to graph. The weights of the edges are assigned automatically based on the 3D distances between the nodes, while the weights of the nodes are assigned manually so that special locations can be emphasized. A custom version of the breadth first search algorithm is used to find the shortest path from an assigned starting position to the destination.

4.1.3 Path Visualization. The path on HoloLens and iOS device is shown by drawing the path through each node inside the selected pathway. The visualization of path is achieved using Unity's Line Renderer component. An object of line renderer is kept global and reused for each new path drawn. The line renderer component requires the number of nodes that it is going to visit and the locations of each node in the 3D space. We use the node points created inside the building model for this purpose. There are functions designed to clear the path, redraw the path which is frequently used while switching back and forth from different visualization.

For the two types of devices, one difference is that the line renderer is redrawn in HoloLens. While the user moving around, the 3D building inside HoloLens is continuously changing its location based on user's position. Line Renderer does not stay intact with its parent component in Unity and hence appears drifting away from the model as user moves away. To avoid this, the line renderer on HoloLens is redrawn on each update.

4.2 Immersive Building Visualization

Based on device type, different visualization is implemented to best suite user experience. For iOS, the screen is the canvas and for HoloLens, the user's world space is utilized. Generally, we have explored four types of building visualizations, including 3 types for visualizing entire building structures and 1 type for AR based immersive navigation.

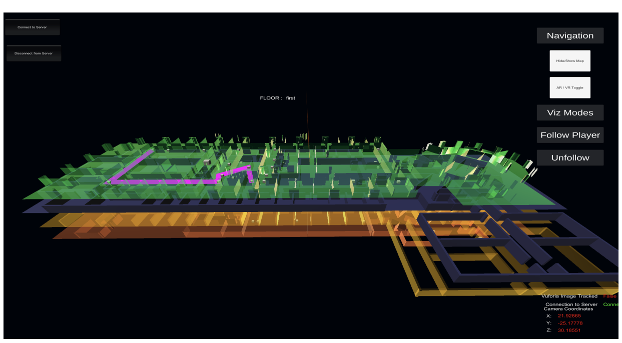


Figure 5: Isometric building visualization on iPad.

4.2.1 Isometric Visualization. Isometric visualization [2] represents the layered structure of a building as the accurate physical model. It is the model we use for navigation. The coordinates of all participants are received on this default visualization and then converted into other visualization types. The shader material used for floor model has high transparency to allow a see through effect from any view angle.

For HoloLens, the 3D building model is handled differently. Just like buttons, the maps were also assigned to face the user at all times. No matter where the user goes, the map always followed the user and anchored itself to specific place in front of user. The *TagAlong* component made sure that the transition is natural. Figure 5 shows an example of isometric visualization on iOS device and the right of Figure 1 shows an example on HoloLens.

4.2.2 Staircase Visualization. Staircase Visualization allows a user to have the benefits of 3D understanding of building like in isometric view but also allows the user to cover all areas without any obstruction in between. As shown in Figure 6 (left), since it is a layered staircase with small deformation from the original physical model, it is generally easy to recognize the floors and positions.

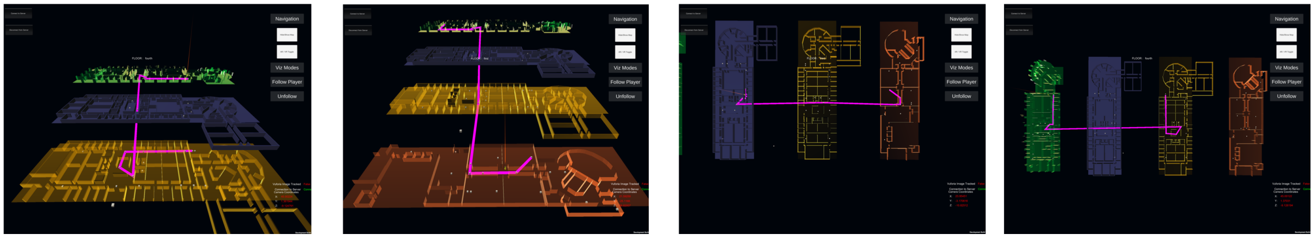
4.2.3 Orthogonal Visualization. Orthogonal Visualization [28] is basically the orthographic view of the model. As shown in Figure 6 (right), this view avoids the occlusion in 3D building model by displaying the floors side by side. It is more suitable to identify the spatial relations on one level of the model comparing to other visualization types.

4.2.4 AR Immersive View. The AR immersive view is designed based on the participants' view, similar to the first person view in the game design. For HoloLens users, the device captures the head position and direction automatically. We use the information to control the position and direction of camera, producing the first person view. Since the iOS devices do not have 3D manipulation functions as HoloLens, we use the device sensor data to control camera angles instead. This view produces the virtual building navigation and can be merged with the real physical environment during navigation. Figure 7 demonstrates the AR immersive view blended with real world scenario.

5 IMMERSIVE INTERACTION

We provide several important functions to support the interaction with our immersive systems on the two types of devices. All the visualization types support the touch controls to zoom and pan the map on iOS devices and hand gestures and voice commands on HoloLens.

The immersive visualization of floors are used as world-in-miniature (WIM) views to improve situational awareness of participants. Stoakley et al. explored the use of a WIM in VR [22], and concluded that WIMs offer many advantages and are intuitively used by users. Our WIM shows real-time positions and rotations of clients in the field, as well as positions of all targets and paths. It utilizes HoloLens' hand tracking to show it hovering over the client's hand when it is within view. The WIM can be rotated using the other hand to view it from a better angle. Finally, the WIM allows clients to create targets at locations that are not within their line of sight, by pointing their gaze point at the desired location and performing a tap gesture.



Staircase Visualization Orthogonal visualization

Figure 6: Staircase Visualization and orthogonal visualization on iPad.

Figure 7: Real world AR immersive view on iPad Pro.

5.1 Using a Particle System to Improve Performance

Several performance issues arise when dealing with the WIM. Simply displaying individual instances to represent all the clients as well as all the targets posts a significant performance bottleneck. Since it is important to see clients' positions and rotations at all times, it is necessary to use individual instances to represent them. Paths are simply lines so the same technique is used to render them on the WIM as in the full-size world. Targets, on the other hand, can be simplified to positions only. This allows the system to use a custom particle system containing all targets instead of an individual instance for each target. This approach is significantly faster and allows rendering hundreds of targets without a significant frame rate hit.

5.2 Tracking objects or participants on the WIM

To simplify tracking objects on the WIM, a unity behavior is developed to automate the process. It holds three parameters: a reference to the model used to represent the object on the WIM, a boolean to indicate whether a separate instance should be created or a particle system point, and a color for the particle system point. When the server instantiates the object on a client, this custom behavior adds the object to the WIMManager for tracking along with the appropriate parameters. Then in the LateUpdate function, the WIMRenderer iterates over tracked objects and uses the specified parameters to update the WIM. LateUpdate is used to guarantee that all objects have moved to their locations in the frame before the WIM is updated, which yields more accurate positions.

Along with different visualizations, we have added a feature of following a player. This feature allows clients to follow another client by selecting a dedicated button on screen. This feature draws shortest path to another client and keeps updating in real time as the clients get near or away from each other.

5.3 Hand Tracking

The WIM uses HoloLens's hand tracking functionality to show and manipulate the view. The HandTracker class registers for the tracking events fired by the HoloLens API. These events are SourceDetected (for when a new interaction source or hand is detected), SourceUpdated (for when a hand is moved), and SourceLost (for when a hand is not visible anymore). Each of these events reports the corresponding hand ID and world position (rotation is not tracked by HoloLens). Using these events, the hand tracker can determine how to manipulate the WIM object.

The hand tracker supports the use of two hands to manipulate the WIM. When the first hand is shown, the tracker saves its ID, activates the WIM, and moves it to the tracked position. Subsequent updates of the hand with that ID are used to update the WIM position. When a second hand appears (while the first hand is still tracked), the tracker saves its ID and its initial position. Subsequent updates of the hand with the second ID are used to compute the angle of rotation to rotate the WIM.

It is worth noting that HoloLens only starts tracking hands when it recognizes the beginning of a standard gesture, which is either an extended index finger or the beginning of a bloom gesture. An open palm or any other gesture does not trigger tracking events.

5.4 Voice Commands

Because the clients may wish to walk around the building model, to have different perspective, speech commands are implemented to make the building stay at one place or to follow the user on HoloLens. "Stay" command comes handy when a client wants to explore the building by not actually moving around the building whereas the "follow" command helps clients when walking inside the building by staying around the client. "Zoom in" and "zoom out" commands allow clients to zoom the model.

5.5 Navigation Interaction

We implemented different approaches for iOS and HoloLens. Showing destination list on iOS devices is accomplished using basic UI components like buttons, placed in Unity's canvas. The canvas is

Figure 8: User interface for selecting destination on an iOS Device.

displayed at start and hid when navigation button is clicked. The canvas system serves the purpose for 2D flat screen in all the cases. It provides visual cues that when cursor is over the button and has a 3D effect of it being pressed. For HoloLens, we use Holotoolkit, a package containing some reusable components and prefabs, to build the interface. We lay out buttons with components like Bill-Board, Sphere TagAlong and a custom Animator script to keep them near the user, facing towards the user at all times. Figures 1 and 8 demonstrate how buttons are displayed on iOS device and HoloLens respectively.

The building model is also controlled differently on these devices. While our iOS system incorporates multi-touch pan and zoom to allow model exploration, the HoloLens system uses its full capability of two hand manipulation to move, rotate and scale the entire model of building per the client's choice.

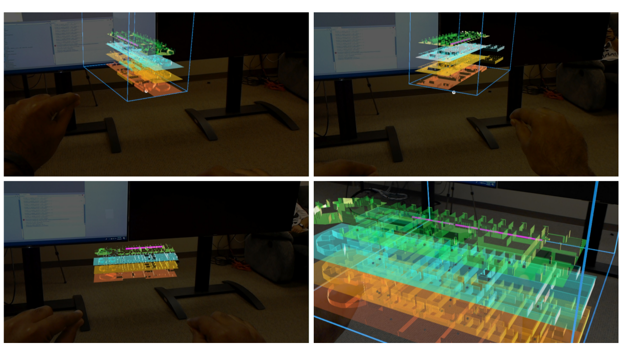


Figure 9: Rotation and Scale using Two Hand Manipulation

6 EVALUATION

To evaluate the effects of immersive navigation, we conducted a user study and acquired expert feedback. We were interested in the performance and experience of using immersive navigation in real life scenarios. Our hypothesis are from two aspects: (1) immersive system improves the navigation tasks, (2) the performance of different participants may vary among different devices.

6.1 Procedure

To understand overall effectiveness of the system, we have designed four tasks involving different navigation functionalities.

Task 1: AR Immersive Nav System. The participants were asked to go to specific locations in the building. During the process, participants were asked to switch between immersive view and default view.

Task 2: Viz Exploration. The participants were asked to go to specific locations in first floor from the fourth floor via stairs. During navigation, participants were asked to use any of the three visualization formats they feel comfortable with.

Task 3: Friend Finder. Group of two participants were considered in this task. Both participants registered the devices and then went to random locations inside the building. The task was to find each other out using our immersive system on iPhone or iPad. Participants were again asked to switch between any of the three visualization formats.

Task 4: Rescue Coordination. Group of two participants were considered in this task. One participant was asked to take iPhone or iPad, and the other participant was given a HoloLens. They both could talk to each other using cellular call. The two participants were asked to work together for one participant to reach a certain location on the map on fourth floor.

The tasks 1 and 2 were performed individually. The participants were given either iPhone or iPad. The devices were registered with image targets and connected to server. The tasks 3 and 4 required two participants to work together. To simplify the tasks, the participants were asked to roam on one floor only. The server kept track of all positions and rotations of participants throughout the session.

6.2 Participants

Eight participants were recruited for user study. All of them were students of ages from 24 to 28. Two of them had visited this building before. No participants had experience of HoloLens previously, therefore they were given HoloLens to learn basic operations before starting the user study.

6.3 Expert Feedback

To acquire the psychological aspect of our approach, we consulted feedback from a researcher in the field of Psychological Science. She performed the same set of tasks multiple times to understand how movement of a person could affect the tracking functions. She had also tested all the three types of devices - iPhone Xs Max, iPad Pro and HoloLens. Her review states, "this is an innovative project with the potential to facilitate collaborative spatial tasks (e.g., navigation, search-and-rescue)." We also received valuable feedback on improvements and areas to focus for future iterations, including different colour schemes to distinguish users and showing the heading of the player in the immersive system to clarify the user position/orientation.

6.4 Summary of Results

The tasks range from simple to complex from task 1 to task 4. Among which, complex tasks receive more positive feedback from participants. Specifically, in task 1, participants commented that immersive navigation could be useful, but not required; in task 2, majority participants preferred having immersive functions but could still be fine without it; in tasks 3 and 4, all the participants agreed the technique to be very useful. The comments from participants include – "Emergency Services like 911 to track down victim", "Finding items in big malls like Walmart or Lowe's or Costco", "Malls, Offices, parking areas" and "Shopping malls, Offices, multilevel parking lots."

7 CONCLUSION AND FUTURE WORK

This paper presents an immersive visualization and navigation system which utilizes networked devices across-platform, including popularly used mobile devices (iOS devices) and AR HMD (HoloLens) . Our approach improves the communication among clients in the physical building environments by providing a tool of real-time visual communication on devices that can be carried freely. We have developed a set of building navigation, visualization, and immersive interaction functions utilizing the available networking, iOS and HoloLens libraries, and provided the technical details of our system for reproduction. We have also performed a basic user study to evaluate the effects of immersive coordination and navigation, indicating that our approach supports the team coordination and navigation tasks and enhances the situational awareness. The results also suggest that such immersive functions are promising for challenging situations including complex building structures or time-critical navigation tasks.

In the future, we plan to continue to explore additional methods to further the effectiveness of our prototype system. First, the use of high-quality CAD models allows server-side automatic path computation, allowing dispatchers to quickly create paths by specifying start and end points, as well as operatives to request a path to a target using the WIM. Second, a video feed would be useful in some situations, but this requires improvements in both HoloLens and the Vuforia library (the current version of Vuforia does not support mixed-reality capture with spatial mapping). Third, coordinate-system drift occurring from large movements should be addressed by realigning whenever a known anchor point is encountered. We also plan to perform formal user study methods to evaluate the effectiveness of the presented functions. Our goal is to improve our system for important applications of smart cities, especially for emergency responses such as firefighters and police.

8 ACKNOWLEDGEMENT

This work was supported by NSF Awards 1564039, 1629913, 1661280, 1840080 and 1937010.

REFERENCES

[1] R. T. Azuma. A survey of augmented reality. *Presence: Teleoper. Virtual Environ.*, 6(4):355–385, Aug. 1997.
[2] D. Ben-Haim, G. Lappan, and R. T. Houang. Visualizing rectangular solids made of small cubes: Analyzing and effecting students' performance. *Educational Studies in Mathematics*, 16(4):389–409, Nov 1985.
[3] H. Benko, E. W. Ishak, and S. Feiner. Collaborative mixed reality visualization of an archaeological excavation. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 132–140, 2004.
[4] M. Billinghurst, A. Clark, and G. Lee. A survey of augmented reality. *Found. Trends Hum.-Comput. Interact.*, 8(2-3):73–272, Mar. 2015.
[5] S. Butscher, S. Hubenschmid, J. Müller, J. Fuchs, and H. Reiterer. Clusters, trends, and outliers: How immersive technologies can facilitate the collaborative analysis of multidimensional data. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 90:1–90:12, 2018.
[6] H. Chen, A. S. Lee, M. Swift, and J. C. Tang. 3d collaboration method over hololens^{TM} and skype^{TM} end points. In *Proceedings of the 3rd International Workshop on Immersive Media Experiences*, pages 27–30. ACM, 2015.
[7] Z. Cui, S. Sen, S. K. Badam, and N. Elmqvist. Vishive: Supporting web-based visualization through ad-hoc computational clusters of mobile devices. *Information Visualization*, 2018.
[8] S. Dong and V. R. Kamat. Collaborative visualization of simulated processes using tabletop fiducial augmented reality. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*, pages 828–837, 2011.
[9] N. Elmqvist and P. Irani. Ubiquitous analytics: Interacting with big data anywhere, anytime. *IEEE Computer*, 46(4):86–89, 2013.
[10] S. Feiner, B. MacIntyre, T. Hollerer, and A. Webster. A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment. In *Proceedings of the 1st IEEE International Symposium on Wearable Computers*, ISWC '97, pages 74–81, Washington, DC, USA, 1997. IEEE Computer Society.
[11] T. Horak, S. K. Badam, N. Elmqvist, and R. Dachselt. When david meets goliath: Combining smartwatches with a large vertical display for visual data exploration. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2018.
[12] W. Huang, M. Sun, and S. Li. A 3d gis-based interactive registration mechanism for outdoor augmented reality system. *Expert Systems with Applications*, 55:48 – 58, 2016.
[13] J. Irizarry, M. Gheisari, G. Williams, and B. N. Walker. Infospot: A mobile augmented reality method for accessing building information through a situation awareness approach. *Automation in Construction*, 33:11 – 23, 2013. Augmented Reality in Architecture, Engineering, and Construction.
[14] K. Kim, M. Billinghurst, G. Bruder, H. Been-Lirn Duh, and G. F. Welch. Revisiting trends in augmented reality research: A review of the 2nd decade of ismar (2008–2017). *IEEE Transactions on Visualization and Computer Graphics*, PP:1–1, 09 2018.
[15] K. Kim, M. Billinghurst, G. Bruder, H. B. Duh, and G. F. Welch. Revisiting trends in augmented reality research: A review of the 2nd decade of ismar (2008–2017). *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2947–2962, Nov 2018.
[16] U. Kister, K. Klamka, C. Tominski, and R. Dachselt. Grasp: Combining spatially-aware mobile devices and a display wall for graph visualization and interaction. *Comput. Graph. Forum*, 36(3):503–514, June 2017.
[17] S. Nilsson, B. J. E. Johansson, and A. Jonsson. Cross-organizational collaboration supported by augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1380–1392, 2011.
[18] J. C. Roberts, P. D. Ritsos, S. K. Badam, D. Brodbeck, J. Kennedy, and N. Elmqvist. Visualization beyond the desktop–the next big thing. *IEEE Computer Graphics and Applications*, 34(6):26–34, 2014.
[19] W. L. Roux. The use of augmented reality in command and control situation awareness. *Scientia Militaria - South African Journal of Military Studies*, 38(1), 2011.
[20] M. E. C. Santos, A. Chen, T. Taketomi, G. Yamamoto, J. Miyazaki, and H. Kato. Augmented reality learning experiences: Survey of prototype design and evaluation. *IEEE Transactions on Learning Technologies*, 7(1):38–56, 2014.
[21] M. Speicher, B. D. Hall, A. Yu, B. Zhang, H. Zhang, J. Nebeling, and M. Nebeling. Xd-ar: Challenges and opportunities in cross-device augmented reality application development. *Proc. ACM Hum.-Comput. Interact.*, 2(EICS):7:1–7:24, June 2018.
[22] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a wim: Interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 265–272, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
[23] I. E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I), pages 757–764, New York, NY, USA, 1968. ACM.
[24] Z. Szalavári, D. Schmalstieg, A. Fuhrmann, and M. Gervautz. "studierstube": An environment for collaboration in augmented reality. *Virtual Reality*, 3(1):37–48, 1998.
[25] R. Tache, H. A. Abeykoon, K. T. Karunanayaka, J. P. Kumarasinghe, G. Roth, O. N. N. Fernando, and A. D. Cheok. Command center: Authoring tool to supervise augmented reality session. In *2012 IEEE Virtual Reality Workshops (VRW)*, pages 99–100. IEEE, 2012.
[26] A. Thangarajah, J. Wu, B. Madon, and A. K. Chowdhury. Vision-based registration for augmented reality-a short survey. In *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 463–468, 2015.
[27] O. Theory. A mixed-reality collaboration service. http://objecttheory.com/work, 2015. Accessed: 2021-05-05.
[28] X. Wang, J. Lim, R. T. Collins, and A. Hanson. Automated texture extraction from multiple images to support site model refinement and visualization. In *The Fourth Int. Conf. in Central Europe on Computer Graphics and Visualization*, 1996.