Mixed-Reality Platform for Coordination and Situation Awareness in Smart Buildings

Elias Mahfoud, Weichao Wang, and Aidong Lu

University of North Carolina, Charlotte NC 28173, USA
{emahfoud, weichao.wang, aidong.lu}@uncc.edu

Abstract. Coordination and situation awareness are amongst the most important aspects of collaborative analysis in smart buildings. They are especially useful for emergency responses such as firefighters, police, and military soldiers. Currently, the communication between command centers and crews are often performed via voice, cameras, and possibly hand-held devices; which offer limited and in-efficient solutions. This work investigates a mixed-reality platform to improve the coordination and situation awareness for multiple users performing real-time operations in smart buildings. Our platform provides a flexible architecture to synchronize crew locations in buildings and vital information across the team and command center in real-time. We have also developed several immersive interaction functions to support efficient exchange of useful visual information. The case study and example results demonstrate the advantages of our immersive approach for on-site collaboration in real physical environment.

Keywords: Immersive Analytics · Mixed Reality · IoT · Smart Building.

1 INTRODUCTION

Since 1968 by Ivan Sutherland [21], extensive research has been conducted on the field of augmented reality (AR) [2, 5, 12, 24]. However, the technology was often used in research or prototype systems instead of practical applications. Only until recently, wearable technology has started to achieve tremendous advancements in many aspects [6]. Such advancements enabled the use of AR in a myriad of fields, such as gaming, education, architecture, facilities management, emergency response and military operations, and many others [18, 19]. 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 [17]. For example, emergency responses 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.

Specific to smart buildings, Modern Building Automation Systems (BAS) depend heavily on data collected from numerous sensors. The data is typically needed for configuring HVAC systems in facilities management in order to optimize energy consumption while regulating temperature and humidity. By translating such functionality to MR, technicians can immediately see data related to specific sensors while navigating a building, without the need to look up a floor plan and manually map their locations. Sensors in hidden places (such as the ceiling) can easily be highlighting in MR, which would be more challenging when looking up their location on a 2D floor plan. MR has a tremendous potential to improve the efficiency of analysis and diagnosis.

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 is designed for both command center and individual crews, utilizing mixed-reality devices to assist indoor operations such as regular maintenances or emergent evacuations. We provide a flexible server-client architecture supporting two-way communication, which synchronizes crew locations and vital information in real-time. We have also designed several important functions for crews to access real-time information and perform investigation on-site conveniently. Our examples and case study demonstrate the prototype system implemented for the latest Microsoft HoloLens and the MR effects that blends virtual information and the real physical environment.

2 Related Work

We review the latest work from the fields of augmented reality, immersive analytics, and coordination mechanisms 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. [8] 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 [23] 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. [10]

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 [11], 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 Immersive Analytics

A number of recent studies on immersive analytics provide favorable results for stereoscopic techniques. For example, Alper et al. [1] presented stereoscopic highlighting to help answer accessibility and adjacency queries when interacting with a node-link diagram, and the evaluation results showed that they could significantly enhance graph visualizations for certain use cases. Ware and Mitchell [25] studied the perception of variations of 3D node-link diagrams and showed that stereoscopy reduced errors and response time in a very high resolution stereoscopic display.Similarly, Greffard et al. compared 3D stereoscopy with 2D visualization and 3D monoscopy, and found that stereoscopy outperforms both 2D and 3D monoscopy [9]. The effectiveness of immersive analytics still needs to be explored. The differences depend on the approaches as well as the applications [7, 14]. Systems that combine different devices have also been developed for visualization and visual analytics [13, 3, 4].

2.3 Coordination Mechanisms

Many researchers have addressed collaboration between users in immersive, controlled and small environments; which is very suitable for virtual reality. For example, an augmented-reality collaboration environment was presented by [22], where a stereoscopic display was projected onto a see-through display, allowing multiple users to simultaneously view the same spatially-aligned model. And recently, a collaboration service based on Microsoft HoloLens was published by Object Theory [16]. 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.

3 System Design and Architecture

3.1 Overview

Our design of the MR platform consists of a server component used by the dispatcher at the command center and individual interface supported by the latest Microsoft HoloLens for crews to perform operations in the field.

- The command center can overview comprehensive information of building and all crews, provide instructions to individual crews.
- The crews can synchronize their locations in the building, perform on-site investigation and operation, and share information with others.

Our system architecture supports smooth communication between the command center and crews, including the real-time locations of the crews and various building information. Operatives at the command 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 dispatcher can also navigate through the building with accurate 3D CAD models and real-time locations and gaze directions of crew members, and create detailed target locations and paths for the crews in the field to follow. The following describes two important components of our system respectively.

3.2 Support for Multiple Connected Devices

Our prototype system is built using the Unity3D engine. The networking component uses UNET, Unity's high-level networking API (HLAPI). This API is server-centric. It requires server-client communication and does not support peer-to-peer communication. The HLAPI determines which objects can be controlled by which clients using authority. While the HLAPI offers many automatic synchronization features, it assumes that the server and all crews share the same coordinate system; however, due to the nature of HoloLens's inside-out tracking, each HoloLens device has its own independent coordinate system, shown in fig. 1. The different coordinate systems 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 crews. Several approaches can be followed to establish a common coordinate system. While HoloLens support 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. As shown in fig. 2, our CAD model includes an object with the tag 'Origin'.

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 HoloLens device starts

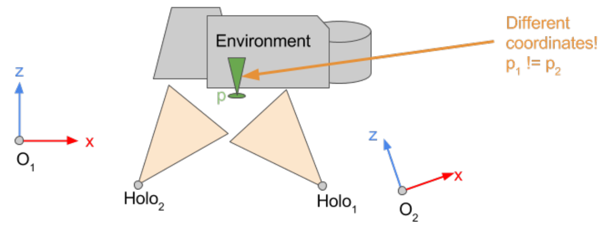


Fig. 1. The challenges of having different coordinate systems for each crew and the physical environment.

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 HoloLens world position is reset to $(0, 0, 0)$, while the rotation is only reset around the Y axis since HoloLens 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, shown in fig. 3. Let T_{cam} be the transform of the HoloLens 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 easily computed by simply computing a quaternion q that rotates from Z'_{img} to Z'_{cam} as follows:

$$q_{xyz} = Z'_{img} \times Z'cam \tag{1}$$

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

Notice 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

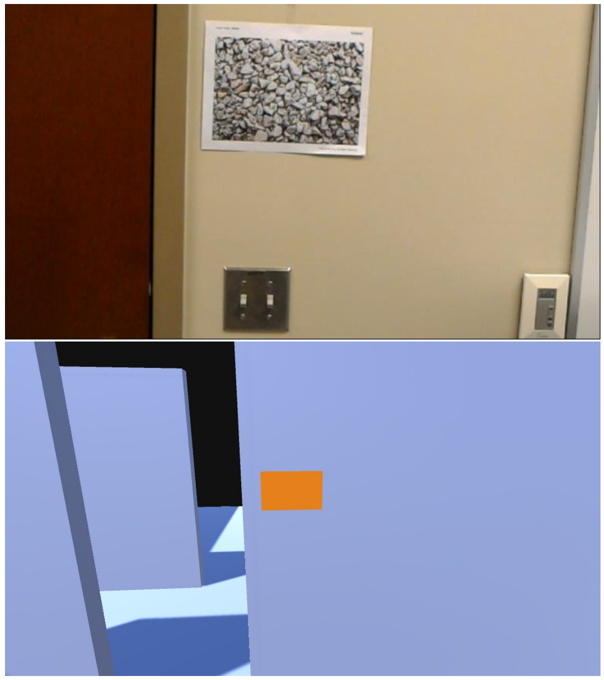


Fig. 2. Vuforia marker and Origin object at the corresponding point. The 'Stones' Vuforia marker and the Origin object at the corresponding point.

model. The resulting final offset is sent back to the client to be used to translate coordinates between it local coordinate system and the common coordinate system. The workflow is demonstrated in fig. 4.

3.3 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.

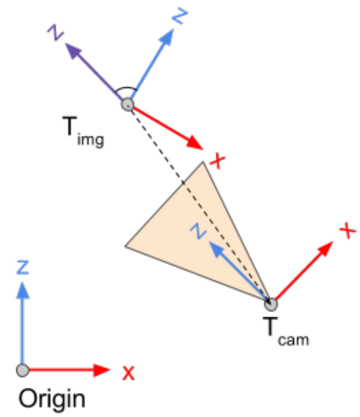


Fig. 3. Computing the initial camera offset in the common coordinate system. The position is the offset between the camera transform, T_{cam}, and the image target transform, T_{img}. The rotation is the angle between the projections of the Z vectors of these transforms on a horizontal plane.

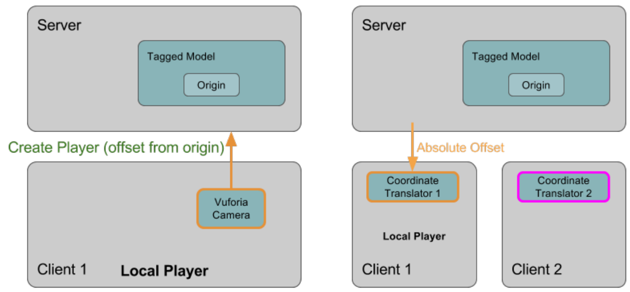
It is also worth noting that the custom network transform also interpolates the position and rotation between received coordinates to achieve smooth animation. This custom network transform is used to synchronize the location of all individual objects (all characters representing operatives and all targets).

4 Immersive Interaction and Analysis

We provide several important functions to support the basic operations of command center and crews in the field.

4.1 Avatar for Representing Crew Locations and Directions

To support understanding of the situations of crews in the field, we create avatars to represent individual crews with a character that mimics the accurate head movement in real-time. The synchronized position and rotation are applied to the character head after interpolation, achieving smooth animation of the head. To make the characters look closer to a person instead of floating heads, we also add a body section which maintains an upright position while the head is moving. This is accomplished with a unity behavior attached to the body. The script has, as a parameter, an offset to determine how far below the head the body should be (to simulate a neck). The script moves the body to that location below the head, and it only rotates it vertically to face the same direction the head is looking. The result simulates character movement more naturally and is much more expressive than what only a floating head can be.



(a) (b)

Fig. 4. Offset computation workflow. The workflow for computing the initial client transform in the common coordinate system. **a)** shows client1 before it connects to the server. The offline scene is loaded and the Vuforia camera is active. When the image target is located, a request is sent to the server to connect the client, and use the calculated offset. **b)** shows the server replying to client1 with its absolute offset, which will be used to translate coordinates for that client. Notice that client2, which is already connected, has a different translator (translator2), since client2 has a different initial offset.

4.2 Creating Targets for Marking Important locations

We often need to mark important locations in a building, such as places with reported accidents. Targets can be created by dispatchers on the command server, or by crews in the field, illustrated in fig. 5. In both cases, the server is responsible for creating and controlling the target instances. The dispatcher can left-click anywhere on the 3D CAD model to create a target. The target's location is synchronized across clients using the same custom network transform used for the characters.

For crews to create targets, they issue a voice command ("Create Target"), which sends a command to create the target. The server raycasts the operatives head direction and creates a target at the hit point with the model.

4.3 Creating Paths for Detailed Instructions

To provide detailed instructions to the crews, dispatchers at the command center can create paths on the server. This may represent safe paths for crews to follow or paths with all required locations to check. Right-clicking on the 3D CAD model initializes a path instance on the server (which is not synchronized yet). Subsequent left clicks add points to that path, until another right click is issued, which terminates the path and synchronizes it across the network clients (fig. 6).

4.4 World in Miniature for On-Site Exploration

The system also supports a world-in-miniature (WIM) view to help improve crews' situational awareness. Stoakley et al. explored the use of a WIM in VR

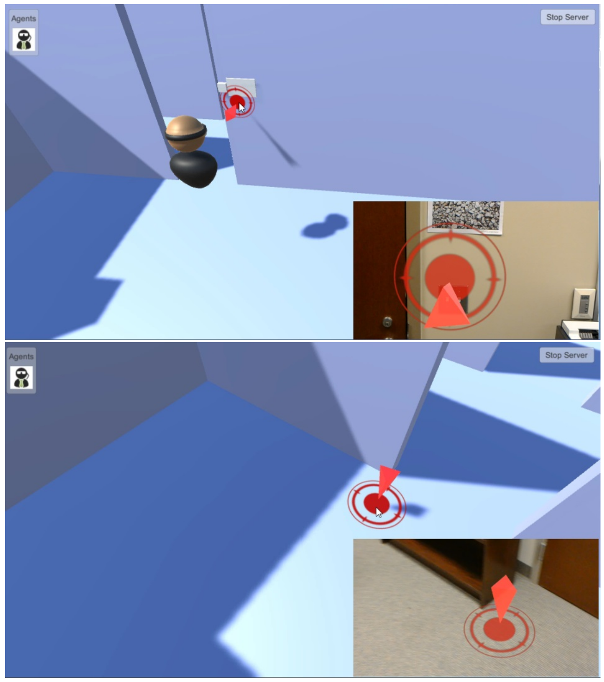


Fig. 5. Targets shown on the server and in mixed-reality. The server view is shown as the background and the HoloLens view is shown on the corner.

[20], and concluded that WIMs offer many advantages and are intuitively used by users. As shown in fig. 7, our WIM shows real-time positions and rotations of crews in the field, as well as positions of all targets and paths. It utilizes HoloLens's hand tracking to show it hovering over the crew'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 crews 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.

Using a Particle System to Improve Performance Several performance issues arise when dealing with the WIM. Simply displaying individual instances to represent all the crews as well as all the targets pauses a significant performance bottleneck. Since it is important to see crews' 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

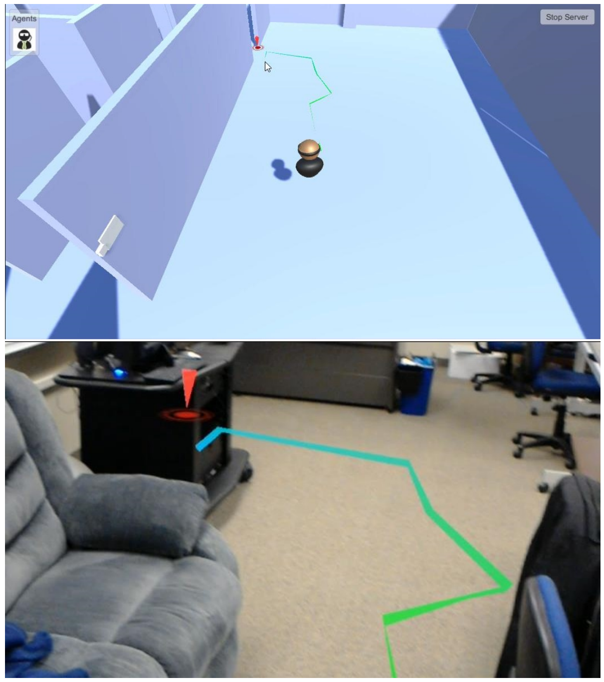


Fig. 6. Paths shown on the server and in mixed-reality.

hit. This performance optimization has been adopted from the adjacency matrix project.

Tracking Objects 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 use 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 location in the frame before the WIM is updated, which yields more accurate positions.

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

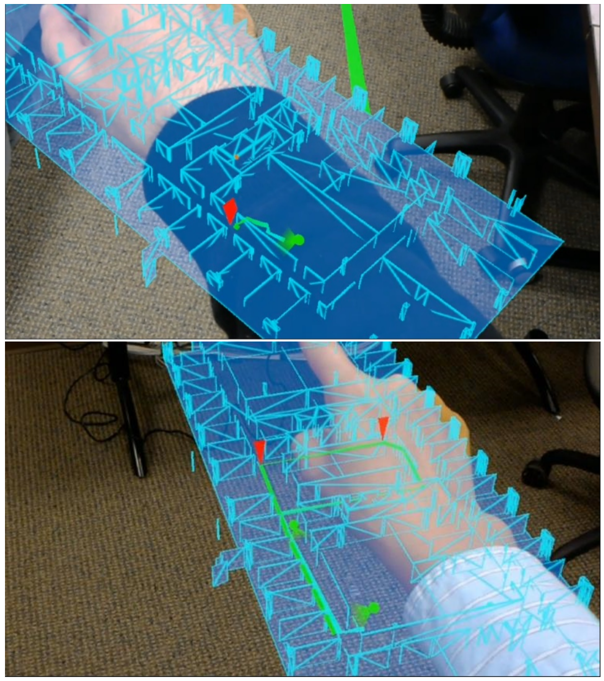


Fig. 7. World-in-miniature supports the crews in the field to understand their environment and important locations of targets and other team members.

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 Example Results and Case Study

We have tested our prototype system in one academic building on campus. Figure 9 demonstrates a sequence of interaction performed by a team of two users.

As shown in the large background images, one crew stays inside a room, holding the WIM with hand gesture and tracking the action of the other crew. The second crew follows the pre-assigned path (green to blue) and investigates the building. Wherever there is an issue, the second crew can use hand gesture to mark the location either in the physical environment directly or on the WIM. The command center and both crews can share all the information updated in real-time. Throughout such a long sequence of study, the second user has traversed around 50 meters in the building. All crew locations, targets, and paths are synchronized in real-time accurately.

Fig. 8. Immersive analytics of IoT data for smart buildings.

Results from our previous work can be combined to achieve more advanced immersive analytics [15]. We support on-site investigation of a variety of building information (such as temperature, humidity, pressure, and CO2) as well as the resident information (such as traffic patterns). The crews can effectively assess such data and perform analysis at the site.

6 CONCLUSION AND FUTURE WORK

This paper presents a team coordination system which utilizes networked HoloLens devices connected to a server in a command center. The system improves team effectiveness in the field by allowing visual communication in MR between crews and the command center. Situational awareness is greatly enhanced with real-time synchronization of operative locations and the use of a WIM. We have also explored the strengths and limitations of current mixed-reality technology. The system is expected to be improved to adapt newer MR technology and eventually serve as a robust platform to build extensive coordination applications in ubiquitous environments.

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.

Acknowledgement

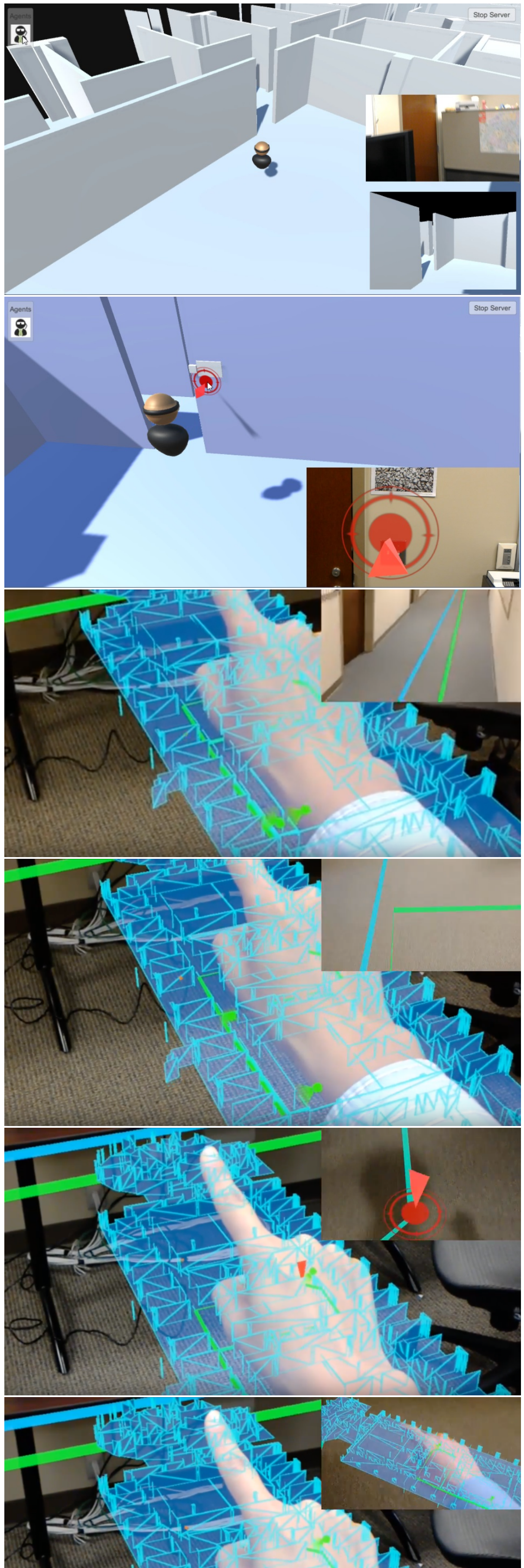
This work was supported by the National Science Foundation under Grant Nos. 1564039, 1629913, and 1840080.

References

1. Alper, B., Hollerer, T., Kuchera-Morin, J., Forbes, A.: Stereoscopic highlighting: 2d graph visualization on stereo displays. IEEE Transactions on Visualization and Computer Graphics **17**(12), 2325–2333 (2011)
2. Azuma, R.T.: A survey of augmented reality. Presence: Teleoper. Virtual Environ. **6**(4), 355–385 (Aug 1997)
3. Badam, S.K., Elmqvist, N.: Visfer: Camera-based visual data transfer for cross-device visualization. Information Visualization p. 1473871617725907 (2017). https://doi.org/10.1177/1473871617725907
4. Ball, R., North, C.: Effects of tiled high-resolution display on basic visualization and navigation tasks. In: CHI '05 Extended Abstracts on Human Factors in Computing Systems. pp. 1196–1199. CHI EA '05, ACM, New York, NY, USA (2005). https://doi.org/10.1145/1056808.1056875, http://doi.acm.org/10.1145/1056808.1056875
5. Billinghurst, M., Clark, A., Lee, G.: A survey of augmented reality. Found. Trends Hum.-Comput. Interact. **8**(2-3), 73–272 (Mar 2015)
6. Chen, H., Lee, A.S., Swift, M., Tang, J.C.: 3d collaboration method over hololens^{TM} and skype^{TM} end points. In: Proceedings of the 3rd International Workshop on Immersive Media Experiences. pp. 27–30. ACM (2015)

7. Cordeil, M., Dwyer, T., Klein, K., Laha, B., Marriot, K., Thomas, B.H.: Immersive collaborative analysis of network connectivity: Cave-style or head-mounted display? IEEE Transactions on Visualization and Computer Graphics **PP**(99), 1–1 (2016)
8. Feiner, S., MacIntyre, B., Hollerer, T., Webster, A.: 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. pp. 74–81. ISWC '97, IEEE Computer Society, Washington, DC, USA (1997), http://dl.acm.org/citation.cfm?id=851036.856454
9. Greffard, N., Picarougne, F., Kuntz, P.: Beyond the classical monoscopic 3d in graph analytics: An experimental study of the impact of stereoscopy. In: 2014 IEEE VIS International Workshop on 3DVis (3DVis). pp. 19–24 (Nov 2014). https://doi.org/10.1109/3DVis.2014.7160095
10. Huang, W., Sun, M., Li, S.: A 3d gis-based interactive registration mechanism for outdoor augmented reality system. Expert Systems with Applications **55**, 48 – 58 (2016). https://doi.org/http://dx.doi.org/10.1016/j.eswa.2016.01.037, http://www.sciencedirect.com/science/article/pii/S0957417416000609
11. Irizarry, J., Gheisari, M., Williams, G., Walker, B.N.: Infospot: A mobile augmented reality method for accessing building information through a situation awareness approach. Automation in Construction **33**, 11 – 23 (2013). https://doi.org/http://dx.doi.org/10.1016/j.autcon.2012.09.002, http://www.sciencedirect.com/science/article/pii/S0926580512001513, augmented Reality in Architecture, Engineering, and Construction
12. Kim, K., Billinghurst, M., Bruder, G., Duh, H.B., Welch, G.F.: Revisiting trends in augmented reality research: A review of the 2nd decade of ismar (20082017). IEEE Transactions on Visualization and Computer Graphics **24**(11), 2947–2962 (Nov 2018). https://doi.org/10.1109/TVCG.2018.2868591
13. Kister, U., Klamka, K., Tominski, C., Dachselt, R.: Grasp: Combining spatiallyaware mobile devices and a display wall for graph visualization and interaction. Computer Graphics Forum **36**(3), 503–514 (2007). https://doi.org/10.1111/cgf.13206
14. Kwon, O.H., Muelder, C., Lee, K., Ma, K.L.: A study of layout, rendering, and interaction methods for immersive graph visualization. IEEE Transactions on Visualization and Computer Graphics **22**(7), 1802–1815 (2016)
15. Mahfoud, E., Wegba, K., Li, Y., Han, H., Lu, A.: Immersive visualization for abnormal detection in heterogeneous data for on-site decision making. In: Interactive Visual Analytics and Visualization for Decision Making - Making Sense of Big Data (01 2018)
16. ObjectTheory: A mixed-reality collaboration service. objecttheory.com/work (2015), accessed: 2018-10-29
17. Roberts, J.C., Ritsos, P.D., Badam, S.K., Brodbeck, D., Kennedy, J., Elmqvist, N.: Visualization beyond the desktop–the next big thing. IEEE Computer Graphics and Applications **34**(6), 26–34 (2014)
18. Roux, W.L.: The use of augmented reality in command and control situation awareness. Scientia Militaria - South African Journal of Military Studies **38**(1) (2011), http://scientiamilitaria.journals.ac.za/pub/article/view/82
19. Santos, M.E.C., Chen, A., Taketomi, T., Yamamoto, G., Miyazaki, J., Kato, H.: Augmented reality learning experiences: Survey of prototype design and evaluation. IEEE Transactions on Learning Technologies **7**(1), 38–56 (2014)

20. Stoakley, R., Conway, M.J., Pausch, R.: Virtual reality on a wim: Interactive worlds in miniature. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 265–272. CHI '95, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1995). https://doi.org/10.1145/223904.223938, http://dx.doi.org/10.1145/223904.223938
21. Sutherland, I.E.: A head-mounted three dimensional display. In: Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I. pp. 757–764. AFIPS '68 (Fall, part I), ACM, New York, NY, USA (1968). https://doi.org/10.1145/1476589.1476686, http://doi.acm.org/10.1145/1476589.1476686
22. Szalavári, Z., Schmalstieg, D., Fuhrmann, A., Gervautz, M.: "studierstube": An environment for collaboration in augmented reality. Virtual Reality **3**(1), 37–48 (1998). https://doi.org/10.1007/BF01409796, http://dx.doi.org/10.1007/BF01409796
23. Tache, R., Abeykoon, H.A., Karunanayaka, K.T., Kumarasinghe, J.P., Roth, G., Fernando, O.N.N., Cheok, A.D.: Command center: Authoring tool to supervise augmented reality session. In: 2012 IEEE Virtual Reality Workshops (VRW). pp. 99–100. IEEE (2012)
24. Thangarajah, A., Wu, J., Madon, B., Chowdhury, A.K.: Vision-based registration for augmented reality-a short survey. In: 2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA). pp. 463–468 (2015)
25. Ware, C., Mitchell, P.: Visualizing graphs in three dimensions. ACM Trans. Appl. Percept. **5**(1), 2:1–2:15 (Jan 2008). https://doi.org/10.1145/1279640.1279642, http://doi.acm.org/10.1145/1279640.1279642


Agents
Stop Server
Agents
Stop Server