Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

SEEING THROUGH WALLS: REAL-TIME DIGITAL TWIN MODELING OF INDOOR SPACES

George Pu
Paul Wei
Amanda Aribe
James Boultinghouse
Nhi Dinh

Department of Computer &
Information Science & Engineering
University of Florida
432 Newell
Gainesville, FL 32611, US

Fang Xu
Jing Du

Department of Civil & Coastal
Engineering
University of Florida
365 Weil Hall
Gainesville, FL 32611, US

ABSTRACT

As the need for situational awareness rises in search and rescue tasks, systems for human spatial sensing augmentation in complex-built environments has become increasingly important. Sensor-based mapping and augmented reality (AR) techniques have been tested to provide enhanced visual assistance in indoor scenarios, such as ”seeing through walls” abilities for occluded areas. Scholars have examined various scanning techniques and UI/UX designs for AR-based occlusion visualization. However, evidence is still needed to show how scene capturing, 3D mapping, rendering and visualization can be executed integratively, enabling an instantaneous ”Digital Twins” modeling. This paper presents a robot-based system for mapping indoor environments and creating a near real-time virtual replica. A dynamically updated 3D map rendered with Unity generates egocentric ”x-ray vision” views of the occluded objects using AR headsets. A case study was performed in a lab room, resulting in virtual regeneration of surroundings supporting search and rescue in occluded areas.

1 INTRODUCTION

Enhancing the situation awareness of professional workers in complex built environments has been considered a critical contributor to the performance and safety of various industrial works, such as construction operations, facilities maintenance, and emergency responses etc. Especially, the ability to sense objects in occluded and confined areas, such as the ”seeing through walls” function, is critical to certain professionals such as first responders in search and rescue. Augmented reality (AR) and 3D mapping are widely used for interweaving virtual world with the real world, granting the user an exceptional visual capability (Sadik and Lam 2017). With the help of the technical improvements on scanning and Simultaneous Localization and Mapping (SLAM) algorithms, a 3D model of real world can be generated efficiently to support various uses in spatial sensing and understanding (Filipenko and Afanasyev 2018). RGB-D cameras and LiDAR are common tools for obtaining accurate spatial and visual information of the surrounding environment. By mounting the entire system on top of an unmanned ground vehicle (UGV), the map of an unfamiliar ground can be obtained and retrieved for visualization of a remote operator, or the telepresence. Although the development of aforementioned techniques enables the possibility of integrating computer vision and AR for visual telepresence, there are still limited cases reported realizing the entire workflow pipeline. Many problems remain, such as the compatibility between the sensors and SLAM, simulation fidelity, visibility in displaying systems, or interactivity of user interfaces (UIs) in AR devices (Makhataeva and Varol 2020). The gaps between individual systems maybe further expanded due to the high specialization of each.

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

The capacity, robustness, dexterity of the system relies on meticulously designed workflow, well-chosen components and smooth communication. This paper proposes a system integrating LiDAR, 3D SLAM, Unity engine, and AR to test an integrative workflow for geneating the ”seeing through wall” function in search and resuce tasks. An infrared LiDAR with 360° view and 100-meter range, and a RGB-D camera are mounted on top of a quadruped robot (i.e., a robotic dog) to scan a space behind the wall. The result is then sent to a ROS platform that executes a 3D SLAM to be stitched together to regenerate 3D models of the scanned space. The scanned map takes the form of a point could, which is a set of coordinates scattered in space. The point cloud is then passed on to Unity game engine for trimming and rendering. Finally, the resultant render will be transferred inside AR goggles (Microsoft HoloLens 2) to be visualizes alongside real-world scene. Orientation and distance of the generated virtual model can be adjusted and aligned according to the odometry and feature information. The UGV can be sent to the room adjacent to the operator, and as a result, the wall can be seen through. The remainder of this paper introduces the related works and the system.

2 RELATED WORKS

Efforts have been made to support the search and rescue in occluded areas. One track of research focuses on creating the spatial models based on the indoor localization estimating. In (Kumar et al. 2004), the authors developed a distrusted system for localization and alerts using radio tags with optical sensors. The position of a human or robot were determined with respect to these tags with a mean error less than 0.3 meters. A similar system is considered in (Faramondi et al. 2013). Here the user wears a vest with inertial sensors and magnetometers is used to estimate the user’s location. This estimate will drift over time as the error is unbounded in the long term. Radio frequency tags are used to reset the wearer’s position to provide room level accuracy.

Researchers have also been trying to visualize objects behind non-transparent physical obstacles with sensing and visualization techniques. For example, Radar waves are utilized to pass through walls and detect hidden people (Wang et al. 2012; Ralston et al. 2010) from the other side. WiFi signal can also be used to detect people behind walls since it won’t be blocked by most construction materials (Adib and Katabi 2013; Adib 2019). These methods rely on signals which are capable of penetrating the obstacles, and the resultant visual effects are based on speculation. Scholars are also interested in how to visualize the occluded objects in an intuitive way. Especially With the help of AR devices, rendered mesh can be aligned within a real-world perspective, and generate a seeing through effect artificially (Avery et al. 2009; Erat et al. 2018). Various types of perceptual backgrounds, depth perception protocols, and visualization metaphors can be used and measured display occluded objects (Livingston et al. 2013). A study in 2009 aimed to implement an AR X-ray vision system with several view modes, namely edge overlay, tunnel cut-out and detail, and overview modes, and dealing with visual occlusions such as buildings and walls (Avery et al. 2009). Another similar study implemented a visualization technique that dealt with human perception and visual saliency and compared the effectiveness of saliency-based X-ray and edge-overlay X-ray. They concluded that environmental effects and visual elements, such as weather, brightness, hue, saturation, and edge levels, affect overall performance of saliency-based and edge-overlay X-rays (Sandor et al. 2010).

Despite the advances on sensing and visualization for occluded areas, existing efforts have mostly been on methods for improving localization and tracking accuracy, or visualizing established model separately, instead of an integrative process from the scanning to UI/UX solutions. We propose an integrative process based on ground robots and a real-time 3D mapping and modeling workflow. By sending a robot to explore the other side, visual feed backs can be easily obtained and possibly fit in an AR device. This will actually give us a pair of eyes to see through the walls. It builds on scanning and 3D SLAM for mapping environments based on the estimated state of the robot from its equipped sensors (Cadena et al. 2016). Google Cartographer’s 2D SLAM algorithm, along with detecting loop closures and optimizing graphs, are tailored and improved to achieve real-time 3D mapping of large floors (Hess et al. 2016). The Robot

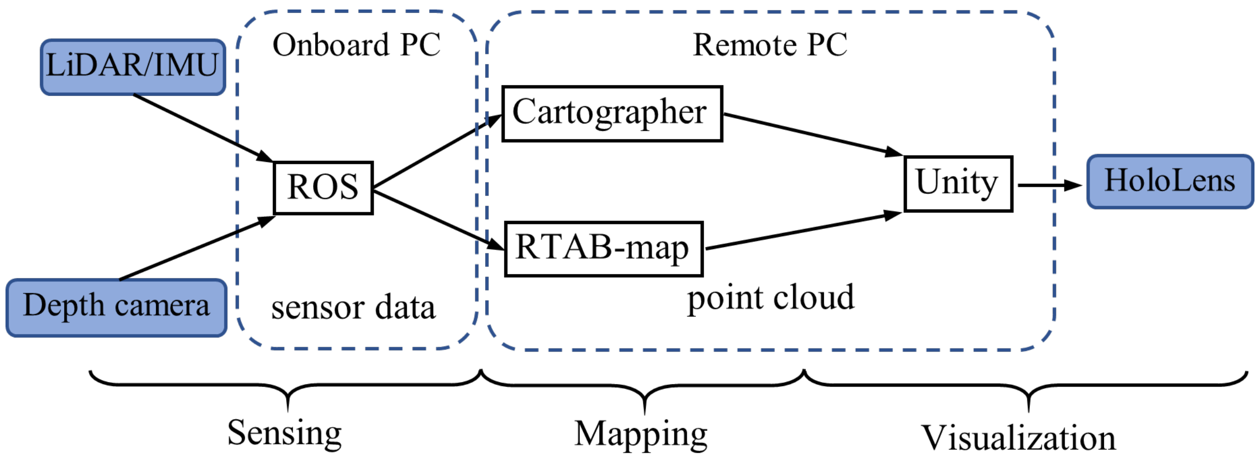

Figure 1: Overview of our "seeing through walls" system.

Operating System (ROS) framework is used as the platform for integrating functions, including the robotic functionality with roslaunch, and visualization and monitoring via RVIZ programs (Quigley et al. 2009).

3 SYSTEM DESIGN

Our system creates a workflow for 3D mapping of environments and exporting the resultant models. This workflow consists of three parts: scanning, SLAM for 3D mapping and modeling, and AR rendering. To evaluate the surrounding environment efficiently, two sensors are used in two consecutive steps. A LiDAR with 360-degree vision, long range and fast scanning speed is first engaged to scan the confined space and give a special feedback to the user. Then a depth camera mounted on the same UGV scans the surroundings and streams colored point cloud to the remote computer for detailed visualization in HoloLens.

For the LiDAR scanning, we integrate an IMU sensor for stabilizing the map. The SLAM algorithm for depth camera uses visual odometry and solely relies on the RGB-D data from the sensor. SLAM synthesizes a map of the environment from LiDAR point cloud scans and IMU pose information. We render the 3D point cloud model in Unity and transfer it to the Hololens 2 over WiFi. The point cloud models can be pushed to the AR goggles at regular intervals while data is being generated. Because the models can be very large, there is often not enough bandwidth to stream data from sensors or SLAM in true real-time and hence our solution only supports a pseudo real time refresh of the reconstructed scene model, at the frequency of one update per 10 seconds.

3.1 Scanning Platform

Our system uses Velodyne Puck LiDAR giving its fast responses and sufficient resolution in mobile scanning as well as a long range of 100 meters. Using a rotating 3D LiDAR reduces the time needed to quickly scan a room compared to a camera. In search and rescue scenarios, there may not be enough time to move the camera into various viewpoints. While our maps will be colorless, shape, distance, and landmark location can still be clearly communicated to users using monochromatic images.

For pose sensors, we are restricted to IMUs. 3D SLAM for LiDAR system requires IMU data to interpret the direction of gravity, which is important for matching the submaps while moving in space with 6 degrees of freedom. The IMU allows us to determine position and orientation along with any changes in these states without resorting to visual odometry. Because the IMU is the sole source of robot pose, it is important that our IMU is accurate and stable, without drifting over time. For this reason we choose the Xsens MTi-670 IMU, which has a low gyro bias stability. It can also connect to GNSS and incorporate GPS signals into

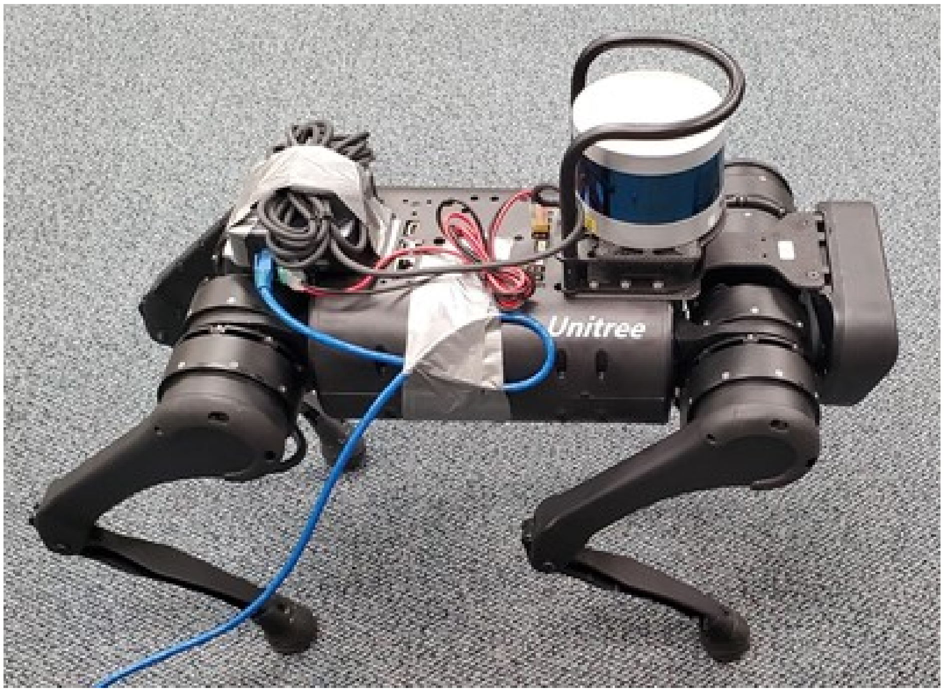

Figure 2: LiDAR scanning platform features Unitree A1 quadruped robot and Velodyne Puck sensor

its position estimate. Both sensors can be mounted on a mobile robot which moves independently of the user.
The sensors are mounted on the back of a quadruped robot. Stabilization of the UGV is vital for the mapping quality of the LiDAR system and especially for the depth camera system. A stabilized platform on the back of the UGV and its legged mobility ensures a smooth scanning operation regardless of indoor terrains. The robot has a height of 400 mm, a maximum tilting angle of 20 degrees, and a maximum moving speed at 3.3 m/s. The room of 513 Weil Hall at University of Florida (775 sqft) was navigated and scanned by the UGV. Then navigation was automated with the help of a Rapidly Exploring Random Tree (RRT) algorithm (Pérez-Hurtado et al. 2018).

3.2 Map Building

To obtain a precise map for the surrounding environment, SLAM algorithms is then implemented for its close loop capability. SLAM is the computation problem of estimating location and creating a map of the environment from observation data. Most commonly SLAM is formalized as a Maximum a posteriori (MAP) estimation problem (Cadena et al. 2016). Given some observations $Z = (Z_1, \ldots, Z_T)$, we want to estimate an unknown variable $X = (X_1, \ldots, X_T)$ consisting of the robot's state s_t and a map m_t. Each observation is a function of this hidden variable.

$$Z_t = h_t(X_t) + \varepsilon_t$$

We use MAP estimation to compute the variable value that maximizes the posterior $\mathbb{P}(X|Z)$. Assuming the noises are independent and normally distributed, this reduces to a non-linear least squares problem, which can be solved computationally using optimization algorithms like Gauss-Newton.

3.2.1 LiDAR mapping with Cartographer

There are many choices for SLAM algorithms with different methodologies designed for specific sensor combinations. Cartographer is chosen as our SLAM algorithm because it is compatible with a diverse

choices of hardware platforms (Hess et al. 2016). Common SLAM methods like OrbSLAM work with camera data only (Mur-Artal et al. 2015). Cartographer supports 2D and 3D mapping, without requiring IMU data for the former. It also integrates with ROS, a meta-operating system for Linux machines which helps orchestrate and record data among different nodes or processes (Quigley et al. 2009).
Cartographer, as an open-source SLAM algorithm, has proven itself with better trajectory building capability and precision over long distance, especially for indoor mapping without the help of GPS and visual odometry (Filipenko and Afanasyev 2018). But problems may occur due to its intrinsic property: the map building process is complete only after the software complete processing the scanned data and performed the close loop correction. As a result, the map obtained from LiDARs can only be extracted once after a while, and this is a trade-off for its precision. And the data streaming from LiDAR sensor is too dense to be transferred with wireless device. As a result, the LiDAR scan serves as a firsthand fast reflection of the environment.

3.2.2 Depth camera mapping with RTAB-map and Kinect

After the special information and dimension of the confined space is obtained with LiDAR scanner, the operator can then direct the UGV to observe and make detailed scan for close up objectives with the depth camera. SLAM algorithm is also applied for building the map with RGB-D data. The features of depth camera are complementary to that of the LiDAR sensor. Due to the depth measuring method, it can only work in a close range (2 m) for better imaging and mapping quality. I other words, this step requires special information of the unknown environment. But the capability of scanning the environment with visible spectrum grants the operator better understanding of the surrounding objectives and can help forming a depth-vision. On the other hand, the algorithm for depth camera SLAM can work in real-time, which also helps the operator to adjust the scanning path to gradually forming the map while controlling the UGV.
For this step, RTAB-map (Real-Time Appearance-Based Mapping) is chosen as the mapping algorithm to pair with Kinect 2 depth camera (Labbé and Michaud 2019). The obtained map in the form of point cloud is then processed to be transferred into Unity for visualization.

3.2.3 Rendering

For rendering and creating the simulation, the game engine Unity is used. Unity engine primarily accomplished three tasks:

1. Import the point cloud model which is in `Pointcloud2.msg` format.
2. Create particles for each of the points in the point cloud model according to a specified color and position.
3. Render the particles with proper orientation, size, and material.

The point cloud imported to Unity could be rendered with its predefined particle system. However, as the 3D map grows, the particle system will not have enough capacity to handle the rendering. To improve fps of rendered scene and memory usage, the VisualEffect (VFX) package was used. VFX provides rendering capability of large point set and better visual, which is highly compatible with this project.

To implement VFX in Unity, the High-Definition Render Pipeline is required, and the VFX graph was also used. The VFX graph is a user friendly tool for managing scripts and customizing VisualEffect avoiding shader programming. Figure 3 demonstrates the pipeline for rendering 3D point cloud with Unity VFX. Imported point cloud topic includes position and color information, which was split and stored with two 2D textures. The 2D textures was then passed on to VFX to initialize each particle. The rendering effects of each particle was then tuned within VFX. Due to the shortage of point density of the mapping software's online function, gaps between pints not be simply filled by inflating its size. Additional random

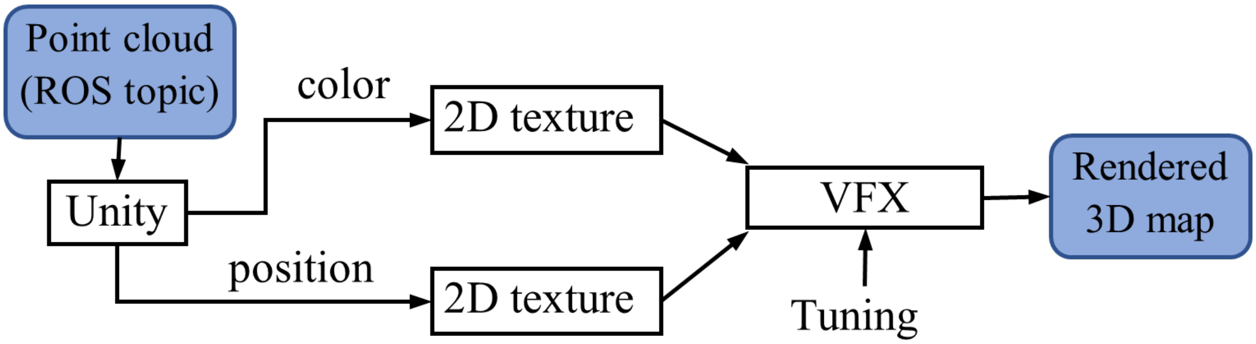

Figure 3: Utilizing Unity VFX for rendering 3D point cloud maps

points was then generated around each original ones for richer visual information. This system could successfully support rendering of sub-million amount of points with acceptable refresh rate.

3.2.4 AR for UI Integration

After the scene model is preprocessed in Unity, including adding the shaders and meshing, the AR is used to combine both real and virtual worlds into one new environment, allowing one to visualize the coexistence of physical and digital objects. In essence, a virtual environment overlays a physical world so that the user perceives both worlds simultaneously. For the system developed in this paper, two components are used, one for rendering and creating the simulation that would be viewed and the other for the actual visualization of the created simulation. For visualizing the created simulation, the Microsoft HoloLens 2 is used. The HoloLens 2 is an AR headset which can take hand movements as user input. Packages from the Mixed Reality Toolkit (MRTK) are required to support mixed reality development and deployment within Unity to a Microsoft mixed reality machine. The primary package that is imported into the Unity project contains the point cloud model extracted from the Mixed Reality Foundations package. The main feature needed is holographic emulation which allows the system to stream data into the HoloLens 2 over WiFi. This is required to achieve real-time visualization of the rendered models inside Unity.

Tracking is mainly done within Unity. Manual calibration is used by marking the starting position and orientation of the headset in physical space and moving the digital model of the headset to match the position and orientation of the headset. Once the environment is rendered and the simulation is initiated, the Unity positioning system is used to reference objects within both real and virtual space. Movement of the camera is tracked by the headset itself. The camera in Unity that correlates to the position, orientation, and size of the HoloLens 2 headset follows the physical movement of the HoloLens 2 itself, allowing for accurate representation of the movement and space around it. Within the HoloLens 2 itself, one would see a superimposed environment which was created in Unity over the physical space around them.

4 CASE STUDY

We tested our system inside a campus building. A laboratory room (775 sqft) in Weil Hall was scanned using a self-navigated ground robot that carried a Velodyne 16-channel LiDAR and a Microsoft Azure camera. Using the `velodyne_driver` ROS package and an Ethernet cable, 108 `PointCloud2` messages were recorded in a 48.3 MB `.bag` file every 10 seconds for reusablility. We used the modified `.launch`, `.lua`, and `urdf` files to perform SLAM on stationary data. Cartographer was fast and efficient enough to perform SLAM without trouble on even underpowered computer hardware in only a few seconds. Once the trajectory finished, Cartographer receives a signal to stop and write the state to a `.pbstream` file. Cartographer's assets writer is used to produce a `.xyz` point cloud file. After the overall spatial model was built with the LiDAR scanning data, the ground robot navigates to a target area driven by the SLAM self-navigation function (target point was defined by the operator). Then the Azure camera was used to

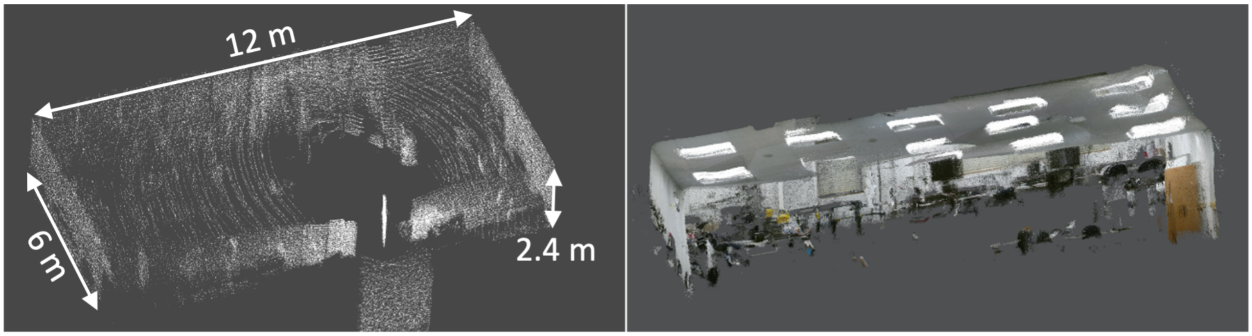

Figure 4: (a) LiDAR scanning result; (b) depth camera scan result.

collect more texture and color information for meshing. The point cloud model obtained from the LiDAR scanner and the depth camera were both transmitted to Unity and rendered before being displayed in the HoloLens 2 headset.

The benefits from scanning the room with a hierarchical workflow with bot the LiDAR scanner and the depth camera are multifold. First, LiDAR can complete scanning the entire room within seconds, which means a faster processing speed and more iteration frequency. It also provides a more precise and accurate reconstruction of the overall scene.The reconstructed map supports a fast SLAM for driving the navigation of the ground robot. Then, depth camera is used to collected richer information about the localities, including texture and color, which may be critical for the situational awareness of the human users. In this way, the benefits of the LiDAR scanner and the depth camera are leveraged while their limitations are mitigated.

The experience of using the HoloLens was smooth. Users were able to clearly distinguish between the superimposed 3D model from the actual environment. The model appeared transparent when taken as a screenshot in Figure 5. Because the HoloLen's screen was limited to a small box, the 3D model did not cover the entire field of view. As such, it created a natural "window" effect. Users could only see through wall in the center of their vision of but not on the periphery. A recorded video from the HoloLens can be found at https://youtu.be/3fJmmJ6iJgI.

We did not have difficulty judging distances to points in the 3D model. Because the 3D model was aligned virtually with the physical room, the visual cues used to judge distances to physical objects. If two distant objects—one real, one virtual—appear similarly sized, then both are the same distance. Virtual objects could occlude each other. Only the virtual objects in the user's immediate vicinity could be displayed. Because the 3D model was made up of points, the user could adjust the transparency of these points based on the user's relative position. Closer points could be made transparent so as to reduce visual clutter.

The LiDAR scanning result of a room sized 12m*6m*2.4m is shown in figure 4(a). Points marking the reflection of infrared laser beams reveal the shape of the room as well as objects inside. The points streamed from LiDAR was very dense. So, the scanning time was limited at 3 seconds, which is already enough to show the shape of large-scaled objects. LiDAR scan can be efficient and precise, because of its long coverage range. But the largest drawback is its incapability of sensing colored light.
The resultant point cloud is colorless, and hard for viewer to comprehend. In comparison, figure 4(b) shows a scan performed with RGB-D camera. The result from depth camera was also obtained with SLAM algorithm, but the stitching of sub-maps was based on matching featured points. The accumulated error leaves us s distorted map. In figure 5, two different results are compared. The HoloLens did a great job hovering these 3D models over the real-world scene, but the visualization was quite different. In figure 5(c), the 3D model could be perfectly aligned along the wall, but it is hard to tell the detail. On the other hand, 3D model result from depth camera, as shown in figure 5 (d), could help us understand objects behind the wall, but the objects were mostly dislocated.

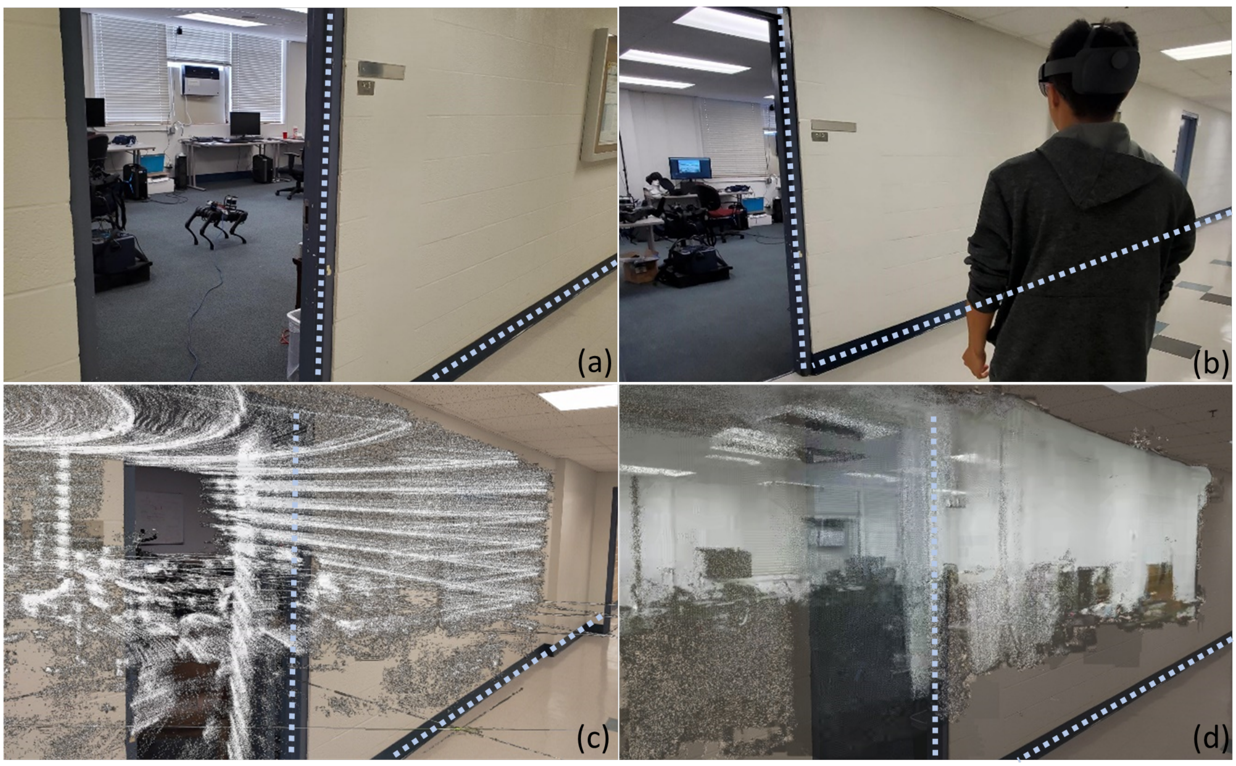

Figure 5: View in HoloLens presenting seeing through wall perspective. (a):UGV scanning the room (b):viewer outside the wall (c):using LiDAR scanned 3D map (d):using depth camera scanned 3D map

5 CONCLUSIONS

In this paper, a system for ”seeing through walls” vision was developed. The system utilizes robotic technology as well as LiDAR/depath camera and IMU sensors to create accurate 3D maps via 3D SLAM that can be visualized through mixed reality on the HoloLens 2. In addition, the system was developed to be pseudo real-time, meaning that the 3D map updates a small but defined time interval so that the user can see the map build up if the map has not be built before. In testing the beta system, it was seen the potential for the system as well as difficulties in future development such as automation with compatibility in mind as well as interface design. Even within this beta system, it was clear that the system developed showcases the potential for collaborative technologies in extending and augmenting human capabilities greater enhancing situational awareness. This work also contributes to the Digital Twins literature by testing the pseudo real-time modeling of the indoor space with potentially dynamic objects, such as moving people. It adds evidence pertaining to the applications of Digital Twins in enhancing situational awareness for emergency response and rapid inspections.

Future work would primarily focus on two parts: automation and interface. The system designed in this paper still requires a manually execution and configurations for AR view alignment. The future direction would be to automate the alignment steps via computer vision methods without any manual intervention. By simplifying the execution of the system to a limited number of steps, the system will be much more usable by a larger portion of people. The other future development direction is the interface. With the system process developed, it is important to tailor the interface so that it is intuitive to users. This would greatly aid in helping users identify crucial information in an efficient manner and would expand its applications to many scenarios.

REFERENCES

Adib, F. 2019. “SEEING WITH RADIO WI-FI-LIKE EQUIPMENT CAN SEE PEOPLE THROUGH WALLS, MEASURE THEIR HEART RATES, AND GAUGE EMOTIONS”. *IEEE Spectrum* 56(6):34–39.

Adib, F., and D. Katabi. 2013. “SEE THROUGH WALLS WITH WIFI!”. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, 75–86. Hong Kong, China.

Avery, B., C. Sandor, and B. H. Thomas. 2009. “IMPROVING SPATIAL PERCEPTION FOR AUGMENTED REALITY X-RAY VISION”. In *2009 IEEE Virtual Reality Conference*, 79–82. Lafayette, Louisiana: IEEE.

Cadena, C., L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard. 2016. “PAST, PRESENT, AND FUTURE OF SIMULTANEOUS LOCALIZATION AND MAPPING: TOWARDS THE ROBUST-PERCEPTION AGE”. *IEEE Transactions on Robotics* 32(6):1309–1332.

Erat, O., W. A. Isop, D. Kalkofen, and D. Schmalstieg. 2018. “DRONE-AUGMENTED HUMAN VISION: EXOCENTRIC CONTROL FOR DRONES EXPLORING HIDDEN AREAS”. *IEEE transactions on visualization and computer graphics* 24(4):1437–1446.

Faramondi, L., F. Inderst, F. Pascucci, R. Setola, and U. Delprato. 2013. “AN ENHANCED INDOOR POSITIONING SYSTEM FOR FIRST RESPONDERS”. In *International Conference on Indoor Positioning and Indoor Navigation*, 1–8. Montbeliard-Belfort, France: IEEE.

Filipenko, M., and I. Afanasyev. 2018. “COMPARISON OF VARIOUS SLAM SYSTEMS FOR MOBILE ROBOT IN AN INDOOR ENVIRONMENT”. In *2018 International Conference on Intelligent Systems (IS)*, 400–407. Funchal, Madeira, Portugal: IEEE.

Hess, W., D. Kohler, H. Rapp, and D. Andor. 2016. “REAL-TIME LOOP CLOSURE IN 2D LIDAR SLAM”. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 1271–1278. Stockholm, Sweden.

Kumar, V., D. Rus, and S. Singh. 2004. “ROBOT AND SENSOR NETWORKS FOR FIRST RESPONDERS”. *IEEE Pervasive Computing* 3(4):24–33.

Labbé, M., and F. Michaud. 2019. “RTAB-MAP AS AN OPEN-SOURCE LIDAR AND VISUAL SIMULTANEOUS LOCALIZATION AND MAPPING LIBRARY FOR LARGE-SCALE AND LONG-TERM ONLINE OPERATION”. *Journal of Field Robotics* 36(2):416–446.

Livingston, M. A., A. Dey, C. Sandor, and B. H. Thomas. 2013. “PURSUIT OF “X-RAY VISION” FOR AUGMENTED REALITY”. In *Human Factors in Augmented Reality Environments*, 67–107. Springer.

Makhataeva, Z., and H. A. Varol. 2020. “AUGMENTED REALITY FOR ROBOTICS: A REVIEW”. *Robotics* 9(2):21.

Mur-Artal, R., J. M. M. Montiel, and J. D. Tardos. 2015. "ORB-SLAM: A VERSATILE AND ACCURATE MONOCULAR SLAM SYSTEM". *IEEE Transactions on Robotics* 31(5):1147–1163.

Pérez-Hurtado, I., M. J. Pérez-Jiménez, G. Zhang, and D. Orellana-Martin. 2018. "SIMULATION OF RAPIDLY-EXPLORING RANDOM TREES IN MEMBRANE COMPUTING WITH P-LINGUA AND AUTOMATIC PROGRAMMING". *International Journal of Computers Communications & Control* 13(6):1007–1031.

Quigley, M., K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. 2009. "ROS: AN OPEN-SOURCE ROBOT OPERATING SYSTEM". In *ICRA Workshop on Open Source Software*, Volume 3, 5. Kobe, Japan.

Ralston, T. S., G. L. Charvat, and J. E. Peabody. 2010. "REAL-TIME THROUGH-WALL IMAGING USING AN ULTRA-WIDEBAND MULTIPLE-INPUT MULTIPLE-OUTPUT (MIMO) PHASED ARRAY RADAR SYSTEM". In *2010 IEEE international symposium on phased array systems and technology*, 551–558. Massachusetts, United States of America: IEEE.

Sadik, M. J., and M. C. Lam. 2017. "STEREOSCOPIC VISION MOBILE AUGMENTED REALITY SYSTEM ARCHITECTURE IN ASSEMBLY TASKS". *Journal of Engineering and Applied Sciences* 12:2098–2105.

Sandor, C., A. Cunningham, A. Dey, and V.-V. Mattila. 2010. "AN AUGMENTED REALITY X-RAY SYSTEM BASED ON VISUAL SALIENCY". In *2010 IEEE International Symposium on Mixed and Augmented Reality*, 27–36. Seoul, South Korea: IEEE.

Wang, F.-K., T.-S. Horng, K.-C. Peng, J.-K. Jau, J.-Y. Li, and C.-C. Chen. 2012. "SEEING THROUGH WALLS WITH A SELF-INJECTION-LOCKED RADAR TO DETECT HIDDEN PEOPLE". In *2012 IEEE/MTT-S International Microwave Symposium Digest*, 1–3. Montreal, Quebec, CANADA: IEEE.

AUTHOR BIOGRAPHIES

Fang Xu is a PhD student at the University of Florida. His research is now focusing on VR/AR, robotics and human computer interaction. His email address is `xufang@ufl.edu`.

George Pu is a senior at the University of Florida. His primary research interests are deep learning, computer vision, and robotics with publications in IROS and NeurIPS. His email address is `pu.george@ufl.edu`. His website is at https://www.georgerpu.github.io.

Paul Wei is a fourth-year undergraduate student at the University of Florida pursuing a Bachelor of Science in Computer Science with a minor in Innovation. His primary research interests are in human computer interaction, software development, and extended reality. His email address is `pwei1@ufl.edu`.

Amanda Aribe is a fourth-year undergraduate student at the University of Florida studying Computer Science in the College of Engineering with a minor in Materials Science and Engineering. Her current focus is on software development, and primarily works on backend development for healthcare systems. Her interests include AR/VR, development, software development, and bioinformatics. Her email address is `a.aribe@ufl.edu`.

James Boultinghouse is a senior undergraduate student at the University of Florida. While pursuing a Bachelor of Science in Computer Science, he will also be receiving a minor in Japanese language. His interests are in video game development and software security. His email address is `jboultinghouse@ufl.edu`.

Nhi Dinh is a fourth year undergraduate student studying at the University of Florida and is pursuing her baccalaureate degree in Computer Science while minoring in Electrical Engineering and Digital Arts and Sciences. Her interests are in software development, software security, and embedded systems. She is expecting to graduate in May 2021. Her email is `dinh.n@ufl.edu`.

Jing Du, Corresponding author, is an Associate Professor in the Department of Civil and Coastal Engineering at University of Florida. He holds an M.S. in Enterprise Management from Tianjin University and a PhD in Construction Management from Michigan State University. His research experiences focus on the development and test of the next generation Intelligent Information Systems (IIS) and Intelligent Physical Systems (IPS) for the design, engineering, construction and operations of built environments. His email address is `eric.du@essie.ufl.edu`. His website is https://faculty.eng.ufl.edu/ericdu/.