**International Journal of Data Informatics and Intelligent computing (IJDIIC)**
Vol. 2, No. 1, March 2023, pp. 70~77
DOI: 10.59461/ijdiic.v2i1.53

# Towards Designing a Planet Walk Simulation in a Controlled Environment

**Justus Selwyn[1], Prasanna Ranjith Christodoss[2]**
[1]Department of Computer Science, John Brown University, Siloam Springs, USA
[2]Department of Information Technology, University of Technology and Applied Sciences-Shinas, Oman.

**Article Info**

***Article history:***

Received February 15, 2023
Revised March 17, 2023
Accepted March 20, 2023

***Keywords:***

Knowledge Representation
Cyber Security
Malware Attack
Visualization

**ABSTRACT**

Planet Walk Simulation involves several parametric readings like physical human body conditions, geographic and atmospheric conditions, etc. In this work, we have set goals including system state monitoring, terrain mapping, and user navigation. Specific objectives are displaying the user's telemetry, elevating their spatial awareness through short range object detection, and displaying their location relative to origin using wi-fi routers. The experimental results showed navigation motion paths for astronauts and identify obstacles in the path with the help of LiDAR and Hololens.

***Corresponding Author:***

Justus Selwyn
Department of Computer Science
John Brown University
Siloam Springs, AR
USA
Email: jselwyn@jbu.edu

## 1. INTRODUCTION

Developing a user interface utilizing a Head-Mounted Display and peripheral devices to simulate a planet walk. Tasks include system state monitoring, terrain mapping, and navigation for both transport and search and rescue. Specifically, our goals include developing a device that can navigate a user to a given location within a margin of five percent of the distance traveled, display oxygen levels, heart rate of the user, and battery level of the system, build and display a topographical map of the area surrounding the individual subject. A planet walk is a simulation of the solar system that allows participants to physically walk the distances between the planets, to scale, in a controlled environment. It is an interactive educational tool that allows people to gain a better understanding of the size and scale of our solar system, the distances between the planets, and the characteristics of each planet. The planet walk simulation can be an effective way to engage people of all ages and backgrounds in science education. It provides a unique and interactive experience that can spark interest and curiosity about space and astronomy, and help people gain a deeper understanding of our place in the universe.

Based on these goals, several objectives have been formulated: utilize/create peripheral devices that gather useful information for the software, utilize/create peripheral devices that are compatible with the HoloLens, utilize holograms to point out obstacles to the user, utilize peripherals, triangulation, and holograms to navigate a user to a desired location, and create an intuitive user interface that has easy access to necessary applications.

Table 1. Program FMEA

| Programmatic System | Primary Sensor(s) | Additional Failure Modes (Refer to “Sensor FMEA” for more information.) | Failure Consequences | Consequence Rating (Chance * Consequence) |
|---|---|---|---|---|
| **Short/Long Rang Navigation** | Navigation Beacons<br>Range Finders | Distance Calculation Errors<br>Angle Calculation Errors<br>Visual Failure | User Led to wrong location<br>User does not know where they are | Moderate * High |
| **Oxygen Display ^ System Energy Level** | Primary Life Support System | Visual Failure<br>Interpretation Error | User does not know remaining 02 levels.<br>User not aware of an offline system | Low * High |
| **Heart Rate Display** | ECG Patch | Visual Failure<br>Interpretation Error | Vitals are not known | Low * Low |
| **Object Detection** | LiDAR | Visual Failure<br>Interpretation Error | Dependence upon user eyesight in limited light.<br>Contact with obstacles. unexpected terrain collision. | Moderate * High |

Table 1. Sensor FMEA

| Sensor | Information Given | Failure Modes |
|---|---|---|
| Navigation Beacons | Distance from beacons (to be computed to vectors) | Out of signal range<br>Loss of Power<br>Destruction of Antenna |
| Range Finders | Distance to pinpoint location (Secondary Distance Measurement) | Out of range, out of power<br>Secondary object blocking visual of target object |
| ECG Patch | Heart Rate of user | Loss of Power |
| LiDAR | Mapping of Local Terrain | Loss of Power<br>Lasers not working on moon |
| Primary Life Support System | System Power State<br>System 02 Levels | Loss of Power<br>Internal Sensor Failure |

### 1.1. Platform and Operating Engine

To develop the programmatic functions aligned with the goals previously stated, an operating platform has to be identified and finalized. Out of a month-long research, four main languages or frameworks were identified: Unity render engine, Unreal engine, any Javascript programming framework, or a hybrid engine using OpenXR. The capabilities that we prioritized were compatibility with the HoloLens 2, motion controller, and voice input [1]. Additional features needed are the libraries for Mixed Reality Tool Kits and World Locking as they will be required for developing a HUD display relative to where the user is looking.

### 1.2. I/O terminals

The core problem addressed in this work includes Object Detection, Search and Rescue, Vitals Display, and Navigation. These would require the construction of several I/O devices and integration of those with the HoloLens and the other scripts. I/O terminals would be primarily used to collect data and information from the surrounding and also the vitals of the user. The initial design of these peripherals includes navigation beacons through a 3-point Wi-Fi mesh, a Bluetooth-enabled or ad-hoc wifi based fitness device, a LiDAR sensor, an LED flashlight bulb, and a processor.

The first task to dissect was developing a way to describe a location and the user's relation to that location. This task was chosen first because both Search & Rescue and Navigation would be dependent upon it [2]. As for design, the present constraint is the absence of global positioning systems (GPS) or a polar north by which to navigate. Since established means of navigation are not possible, the focus is turned towards the implementation of position vectors. Given a user's position relative to two other locations, the angle and distance to another given coordinate can be calculated. One note in this approach is that the distance measured is a straight-line path, potentially cutting through local terrain. The result for the user is the assumption that the actual distance to be covered will be greater than that displayed on the screen. Likewise, they will be responsible for mapping themselves around impassible terrain.

Another crucial aspect of such navigation is the detection of objects immediate to their surroundings. This design takes an input stream of pixels from a Lidar sensor attached to the HMD. With this stream, an object detection algorithm is applied to identify the clusters of objects or masses that impede the course of the user. Given a location relative to the LiDAR's understanding of the objects' proximity, sprites are then overlaid onto these objects, which were visible in the display of the HMD.

Furthermore, displaying vitals to the user is a notable task. The ECG-patch will be used to give data to the HoloLens on heart rhythms of the wearer. This data would then be portrayed in a visual format in an overlay on the HoloLens screen so that the user can be aware of the status of their heart. Another vital aspect of the spacesuit is the life support system, which would include the power source as well as the user's oxygen supply [3]. Connecting this system to the HoloLens would then allow for the programmatic display of both the amount of oxygen and the battery level to the user.

### 1.3. Heads-Up Display

The heads-up display of the project (HUD) will be the means by which the user interacts with the processed information. A fundamental consideration for the user interface was maintaining ease of use. As computers are not intuitive by nature, the way we decided to appeal to the general audience was through familiarity. Therefore, the different subtasks for this work are presented as widgets on a toolbar. This toolbar is accessed by pressing a home key in the bottom left corner of the display. The reason behind the home key is to make the application synonymous with hitting the Windows key on a Microsoft computer. Clicking the Windows button causes applications to appear, that upon selection, takes the user to the desired program.

## 2. DESIGN PROPOSAL

The proposed four main application panels are: object identification, navigation, mapping, and in-hand widget menu[4]. The widget menu, which is similar in style to the Microsoft Windows taskbar, will serve as the home to the many operations[5]. From this bar menu, any of the other application panels can be opened and maintained on the screen. Concerning the vital telemetry readings, an application panel will have the battery levels, heart rate, the amount of oxygen left in the dress of the navigator, and the profile name of the individual. As for navigation, typical earth-based navigation systems such as the standard global positioning system and magnetic north will not function on the other planets.

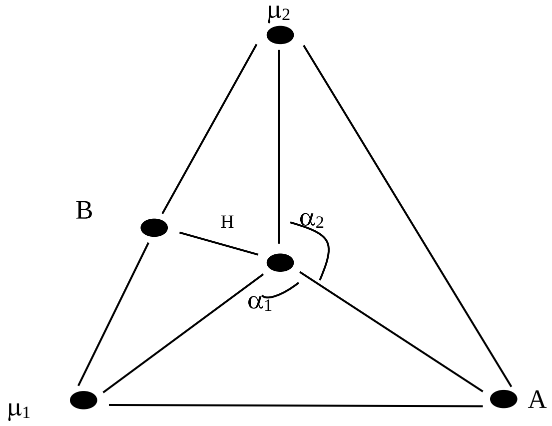

Figure 1. Positioning the Beacons

A location is dependently calculated upon a user's relative distance to the object or to a referenced location. Traditionally, location vectors are represented as x, y, and z coordinates. The initial plan was to move to coordinates based purely on the x and y domains, assuming that the change in height would be marginal for the planet’s extra-vehicular activity. However, it is found that elevation changes vary to the planet’s gravity. The resulting plan is to assume a standard delta z by which the program will refine distance measurements. Research and study are still being done to maintain accuracy in regions of higher and lower elevations[6].

The design will use wifi-based laser beacons to measure the user's distance from select land points. There will be three immobile beacons; home, A, and B. Beacon A will be set 100 meters from the house, and Beacon B will be set 10 meters from the home location, as shown in Figure 1.

As for the points of the triangle, the user x is represented by “ux”, home base by “H”, beacon A as “A”, and beacon B as “B”. The distances measured by beacons are the following:

H := length of side ux H
A := length of side ux A
B := length of side ux B
C:= length of side AH

Given side lengths H, A, and C, the triangle AH ux is fully defined. From these side lengths, alpha, the angle between C and H will be calculated using the law of cosines[7]. At this point, the program will display to the user H, the distance from home, and alpha, the angle relative to control beacon A from home. However, the user's position is not fully defined, as the user may be on one of two sides of the line AH. The completion of the definition is found through the line uxB. If B is less than H, the user is said to be "above" the line AH. Likewise, if B is greater than H, the user is below the line AH. The program gives the user their position in terms of A, alpha, and a Boolean above (set to false when below) from the array of distances between beacons.

As shown in Figure 2, the navigation will appear as a distance to the target and an arrow indicating the direction to the target location respective to the astronaut.

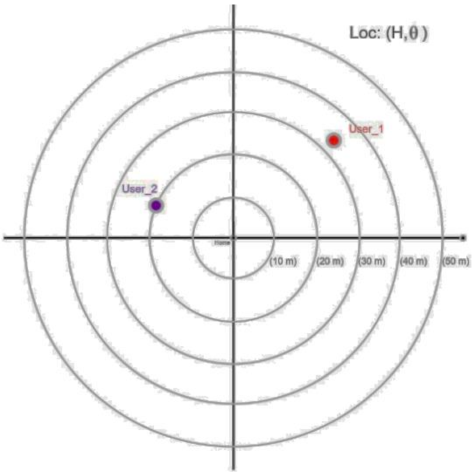

Figure 2. Navigation Appears

These visualizations will be encased in a hologram panel incorporated into the heads-up display (HUD). For the mapping application, the plan is to have a panel that contains a topographical map with both the user's location and a possible target location or distress signal. This map will be generated using a LiDAR attached to the HoloLens. Each panel will have a default position but should be capable of resizing and moving around[8].

Along with this universal capability, all the holograms will be connected and share an opacity level. This shared opacity level will help the user to maintain visibility on the move and remove holograms from view when necessary. The main form of interaction with the software and the HoloLens will be hand motion and voice communication. Although not mentioned in the above applications, object detection is still a crucial part of the design. The default threshold for this detection would be to pick up all sizes of objects, but the user can specify the targeted object size.

The initial idea is to have an optional slider in the settings to specify the desired size of the object. This application would be active by default, ensuring the user has the maximum amount of safety. The first step to accomplish this task is to utilize data from the LiDAR attached to the HoloLens. After gathering the data, holograms of the LiDAR data can be generated and placed on the obstacles in the field of view. These holograms will showcase the depth and size of the object in question[9]. A goal for this task is to possibly utilize the World Locking Tools that Unity has access to regarding mixed reality projects. These tools could be very useful in helping anchor holograms to their correct positions.

The last aspect of the design challenge is being able to execute a search and rescue operation. The intended plan for this scenario would be to retrieve communication from a distressed crew member. This task aims to have navigation beacons that can ping each other to alert nearby crew members and give a distance to the astronaut in peril. Each astronaut would have an antenna that could ping and receive pings. After closing the message, any navigation data will be uploaded to the navigation panel for the astronaut to act upon.

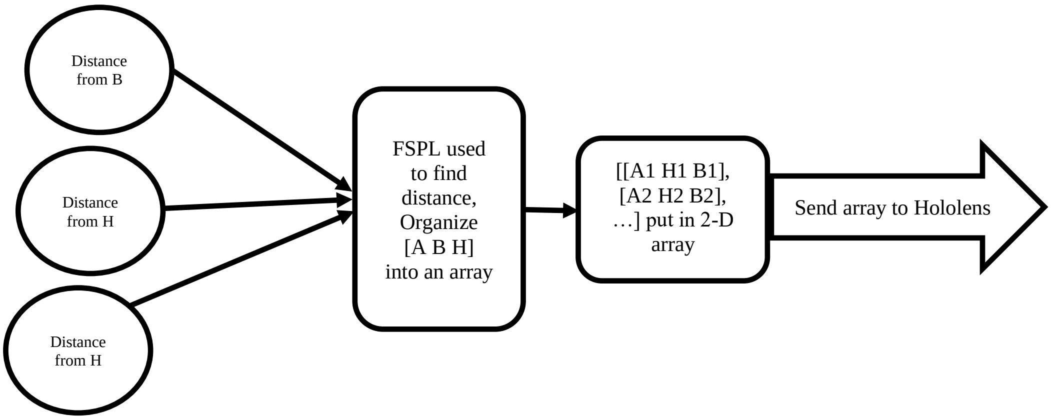

Figure 3. Wi-Fi Navigation and Data processing

Figure 3 shows that LiDAR’s terrain mapping using light and the ability to display and indicate distances and heights of impending obstacles is extremely useful for astronaut navigation and search and rescue [10]. The LiDAR will be connected to the Raspberry Pi via USB and its data will be sent in an array of data points for the Pi to store and then send to the HoloLens to display a scan of the surroundings to the astronaut[11]. It will have a max range of about twelve meters and readings above this data will be cut out for accuracy. The LiDAR will be positioned on the HoloLens right above the lens on top of the projector box so that it gets the most accurate readings from the astronaut’s perspective.

## 3. CONCEPT OF OPERATION

When starting the HUD application, the user will be presented with a widget menu upon hand rotation with palm up. As stated previously, this menu will contain application thumbnails to click on, which will start up the respective application. Widgets ease access in starting up other applications tied to the design challenge, minimizing the time needed to open the Windows home screen. Once the menu has appeared, the user can select an application. As a sample scenario, consider the user selecting the suit and vitals application. This selection would create a hologram popup in the top left corner of the field of view. The popup would contain:

- Helpful information, including the first and last name of the user.
- The oxygen level as a percentage and as time.
- The battery level as a percentage, time till empty, and visualization.
- The user's heart rate as a visual heart beating and as a numerical value in beats per minute.
- The oxygen saturation level as a percentage.

Figure 3 setting up the suit and vitals stream, the user selects the navigation application. Since this is the first time starting up the nav app, the user is redirected to the location application. This app is provided to store locations for easy access to prevent the user from keeping track of coordinates of different landmarks or destinations. Knowing this, the user updates the list with three new locations: the lander, a navigation beacon placed approximately 100 m away from the lander, and a landmark specified by Mission Control. Going back to the navigation app, the user can now select a location to navigate to. The user chooses the landmark, and the navigate app disappears and is replaced by a panel in the top right corner. This panel houses the landmark's coordinates, an arrow pointing to it, and the astronaut's distance from it. The arrow rotates as the user moves around, updating the direction and distance respective to the landmark updates.

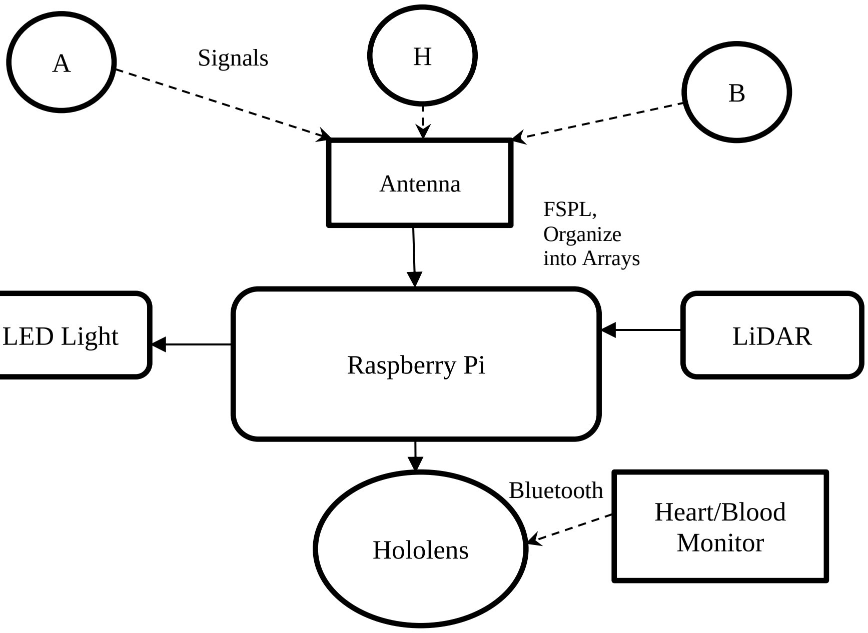

Figure 3. Operational Schematic

Upon setting up the navigation, the user decides to map out the area on route to the landmark. The user selects the terrain mapping thumbnail to pull up a map in the lower left-hand corner. This map is blank initially but will soon fill up through data gathered by the LiDAR. Another feature the user has access to during traversal is object detection. Detection is enabled by default, so detection should begin on startup. When detecting objects, the LiDAR will send data of positions of potential obstacles. The program will utilize these positions to generate holograms to help the user visualize obstacles with proper depth. The user can then take appropriate action to avoid the displayed obstacles.

Consider a potential search and rescue scenario the user is met with an alerting sound accompanied by an urgent message before finishing the trip to the landmark. A distress signal has come in from a fellow astronaut signaling that they are in trouble. The announcement states that the astronaut is wounded and in need of rescue assistance.

After closing the message, the HoloLens updates the navigation panel with a new location, changing each field to navigate the user to their distressed crewmate correctly. During the trip to the distress signal location, the user can constantly communicate with their crewmate through beacon-to-beacon messaging. A messaging app is provided that can either be typed out with a virtual keyboard or spoken word for hands-free mode. The program can then send these messages to other beacons.

## 4. TESTING PROCEDURE

There are several aspects to be tested at the end. The first aspect of the electrical system to test is the LiDAR Object Detection system (LOD). The LOD designed to detect objects on the Hololens, but the main testing and responsibility of the electrical side of the LOD is to gather the data on the from the LiDAR, organize and send it to the Hololens. Ideally this process would be automated, but the initial testing phase is done manually.

The second aspect in the WiFi Navigation System (WNS). The WNS will take the signal strength of the WiFi mesh point and convert that to a distance using a Free Space Path Loss equation (FSPL). This will also send the data to the Hololens using an API and will hopefully be automated, but it will have to be tested manually first, one mesh point at a time. Finally, the health monitoring system would be tested. This aspect was discontinued and will be left for the next team to potentially implement, but it would be tested manually as well by trying to extract the information from a fitness band to a raspberry pi.

The testing procedure for the LiDAR started by first scanning a chair or other object with the LiDAR and saving the scan as a .csv file. This file then had to be processed to only include the x,y,z plot coordinates in the file and separate them out into individual columns and cells. Then these readings were converted to a 3D dot plot display on the Hololens. This was done several times with a table and with a furniture to test the depth perception and specificity of the plot. This is one of the preferred approaches discussed in[12].

The testing procedure was centered around the FSPL equation 1 which is able to convert signal strength (dBm – decibel meters) to distance in meters.

Distance for RSSI (dBm) = 10^((Measured Power – (-RSSI)/(10 * N)) (1)

The measured power is a constant that is dependent on the Wi-Fi system being used. For each system it is different as it is based on the RSSI value at one meter away from the system. The RSSI is the measured value for whatever distance you are trying to measure. N is a constant that depends on the range of the Wi-Fi system and is somewhere between 2-4, in this case it is 2. For this system the Measured power was -28 dBm. This equation and the WNS was then tested at several different distances. It had to sit for a little while at each distance in order to let the pi adjust to the new distance. Only one mesh point was used to measure the distances just to ensure consistency and the system was tested at 1m, 3m, and 4m (based on the constraints of wired devices and how far they can stretch from each other. The results for the distances are relatively inaccurate and get more and more inaccurate the farther away the device gets. This is most likely due to the range limitations of the pi and the Wi-Fi mesh points.

### 4.1. Design Changes and Prototype

The main improvement that would be made would be to automate the process of sending readings of the current coordinates of the user with respect to the three meshes from the LiDAR to the Hololens or being able to store and process the values in the processor, and then send the location to the Hololens. This did not happen in this current design but the concept of object detection and how it would work is shown on the Hololens with the sample readings that were obtained from it.

The RSSI readings can be especially difficult to work with because they're somewhat inconsistent and it takes a long time for the processor to adjust to the new distance the mesh point is moved to. The other problem is other Wi-Fi signals that get picked up from the processor that get in the way of reading all three mesh points at once which hurts the chances of triangulation.

This system is able to still send information and readings to the Hololens but only manually and by being uploaded from the emulator to the Hololens. The navigation system can do much of what was desired originally. This is the most fully developed system of the originally desired systems. The processor can find the Wi-Fi point that's intended, obtain the RSSI, and then find the distance from the RSSI.

## 5. CONCLUSION

As for future work on this project there are many concrete possibilities. There are those that build upon existing Unity features such as object detection within the Lidar Display, and making the display feel more like a HUD. Others could take systems such as vitals and object detection and implement live data. And finally, there are those such as connecting the R-Pi to the HoloLens, and Search and Rescue that will have to be pioneered by the next group. Extension of this research can be mainly dedicated to the automation of the LOD system and finding a way to implement it to automatically scan while attached to the Hololens in conjunction with the Raspberry Pi. The other main research subject should be focusing on the WNS and automating that with three separate points. There also needs to be a way to filter out other Wi-Fi systems and not take them into consideration. Stand-alone beacons that ping every few seconds or so may be a good idea for a more practical usage of the Hololens. The health monitoring system should also be researched but may need to be more specific in the devices chosen in this system. A fitbit style band may not be the proper option in the future and more basic devices may be needed, such as a simple heart rate monitor.

## REFERENCES

[1] Minoufekr, M. et al. (2019) "Modelling of CNC machine tools for augmented reality assistance applications using Microsoft hololens," Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics [Preprint]. Available at: https://doi.org/10.5220/0007920806270636.

[2] Newbury, R. et al. (2022) "Visualizing Robot Intent for Object handovers with augmented reality," 2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN) [Preprint]. Available at: https://doi.org/10.1109/ro-man53752.2022.9900524.

[3] Carius, L. et al. (2022) "Cloud-based cross-platform collaborative augmented reality in flutter," Frontiers in Virtual Reality, 3. Available at: https://doi.org/10.3389/frvir.2022.1021932.

[4] Soviany, P. and Ionescu, R.T. (2018) "Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction," 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) [Preprint]. Available at: https://doi.org/10.1109/synasc.2018.00041.

[5] Redmon, J. et al. (2016) "You only look once: Unified, real-time object detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Preprint]. Available at: https://doi.org/10.1109/cvpr.2016.91.

[6] Long, S., He, X. and Yao, C. (2020) "Scene text detection and recognition: The Deep Learning Era," International Journal of Computer Vision, 129(1), pp. 161–184. Available at: https://doi.org/10.1007/s11263-020-01369-0.

[7] Fu, Z. et al. (2019) “Foreground gating and background refining network for surveillance object detection,” IEEE Transactions on Image Processing, 28(12), pp. 6077–6090. Available at: https://doi.org/10.1109/tip.2019.2922095.

[8] Li, B. et al. (2022) “Equalized focal loss for dense long-tailed object detection,” 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) [Preprint]. Available at: https://doi.org/10.1109/cvpr52688.2022.00686.

[9] Sharma, L. and Lohan, N. (2019) “Performance analysis of moving object detection using BGS techniques in visual surveillance,” International Journal of Spatio-Temporal Data Science, 1(1), p. 22. Available at: https://doi.org/10.1504/ijstds.2019.097607.

[10] Park, S., Bokijonov, S. and Choi, Y. (2021) “Review of Microsoft hololens applications over the past five years,” Applied Sciences, 11(16), p. 7259. Available at: https://doi.org/10.3390/app11167259.

[11] Walker, M. et al. (2018) “Communicating robot motion intent with augmented reality,” Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction [Preprint]. Available at: https://doi.org/10.1145/3171221.3171253.

[12] Vaquero-Melchor, D., Bernardos, A.M. and Bergesio, L. (2020) “Sara: A Microservice-based architecture for cross-platform collaborative augmented reality,” Applied Sciences, 10(6), p. 2074. Available at: https://doi.org/10.3390/app10062074.

## BIOGRAPHIES OF AUTHORS

**Justus Selwyn** is working as a professor of computer science at John Brown University, USA. He received his Ph.D. in computer science in 2011 with specializations in software and knowledge engineering. He has also worked with software companies in India and has designed and developed several software applications for his clients. His research works are published in reputed international journals and are presented at international conferences in Germany, Israel, and in the US. His research spectrum widely covers areas like software engineering, knowledge engineering, machine learning, bigdata testing, web services, software engineering as a service, and knowledge-based artificial intelligence. He can be contacted at email: jselwyn@jbu.edu

**C. Prasanna Ranjith** Presently a Teaching Faculty in the department of IT at University of Technology and Applied Sciences-Shinas, Oman since October 2011. During this period, I have served as Quality Assurance Coordinator of IT Department from January 2012 to September 2013 ensuring best practices in teaching and learning. Now being the Chairperson of Department Research Committee from September 2016, I hold the responsibility of planning and implementing various Workshops, Seminars and Conferences towards professional Research and development. Obtained Ph.D. in Computer Science from Bharathidasan University, Trichy, India. My research interests centreon improving the performance of routing algorithms in networked computer systems and in Ad Hoc Networks, mainly through the application of Genetic Algorithm. He can be contacted at email: prasanna.christodoss@shct.edu.om