An Augmented Reality Guidance and Operations System to Support the Artemis Program and Future EVAs

Najya Ahsan, Michael Andersen, Peter Baldwin, Jasmine Brown, Naiya Chapman-Weems, Chantil Hunt Estevez, William Hyland, Blessing Leonard, John Manlucu, Michael Vandi, Claudia Yee, Giovanni Vincenti[1], Greg Walsh[2]
University of Baltimore, Baltimore, MD, 21201

As technology becomes more pervasive in today's society connecting nearly every element of one's life, it is essential to think of ways to apply the same paradigm to Space exploration and the spacesuit in particular. The solution presented in this paper describes ARGOS, a guidance and operations system that utilizes an augmented reality device to support an astronaut during an EVA. This system serves as a model for future training and operational environments, describing how visualization and interaction methodologies can be coupled with distributed computing and ad-hoc networking to bring flexible cloud-computing-like features to a future generation spacesuit. The manuscript describes the system and preliminary tests.

Nomenclature

AR	=	Augmented Reality
ARGOS	=	Augmented Reality Guidance and Operations System
DCU	=	Display and Control Unit
EVA	=	Extra-Vehicular Activity
HMD	=	Head-Mounted Device
IVA	=	Intra-Vehicular Activity
MCC	=	Mission Control Center
ML1	=	Magic Leap One
OCTaVIA	=	Operations Control, Translation, and Visual Interface Assistant
RCA	=	Remote Control Application
UI	=	User Interface
xEMU	=	Exploration Extravehicular Activity Mobility Unit

I. Introduction

NASA SUITS presents a mission to students that has not been undertaken in over 40 years; returning to the Moon. Since Eugene Cernan, Harrison Schmitt, Ronald Evans and the Apollo 17 mission in December 1972, no person has returned to the surface of the Moon. However, with the Artemis Mission planned for 2024, that is soon to change. In order to prepare for the challenges of modern and future lunar missions, NASA has decided to leverage augmented reality technology to aid astronauts to accomplish their lunar assignments and field work.

In order to develop a system designed to aid users through complex and possibly dangerous tasks, we propose the Augmented Reality Guidance and Operations System (ARGOS) which utilizes the Magic Leap One Augmented Reality Headset (ML1). Using the ML1, ARGOS presents a minimal, yet effective user interface designed to display vital information, provide instructions, and reduce the cognitive load for the user.

We accomplish these tasks through the introduction of 3 main elements in our system: Voice Commands/Interaction, Remote Control Application, and the ARGOS-specific Operations Control, Translation, and Visual Interface Assistant, also known as OCTaVIA. The goal of ARGOS is to create a system that will ensure a user's ability to complete the given task with the utmost safety and efficiency, despite any communication failures that may occur. Therefore, the system is designed with both an online and offline mode to ensure that functionality

[1] Associate Professor, Science Information Arts and Technologies, gvincenti@ubalt.edu
[2] Associate Professor, Science Information Arts and Technologies, gwalsh@ubalt.edu

persists despite any disruptions in communication. Lunar missions present a unique and dangerous set of challenges, and with the updated design of ARGOS, astronauts will have an available and usable system built specifically to help reduce cognitive load, aid completion of tasks, and increase safety.

II. Mission Objectives

The NASA SUITS Design Challenge set several requirements, aimed at developing "a user interface for an HMD in an augmented/mixed reality device which assists astronauts with their responsibilities during a lunar mission by providing instructions and other vital information for EVAs in a non-obstructive way."[1] The provided scenario included three main modes of operations: translation to the worksite, scientific sampling, and rover repair. Each scenario then has a series of universal mandates, which include constant communication with Mission Control Center and an IVA Operator, the interfacing with the suit's control systems as well as telemetry stream, and the ability for the UI to quickly adapt to conditions of extreme brightness or darkness.

A. ARGOS

In order to adapt to the demands of a lunar mission, the primary purpose of ARGOS centers on a single concept: to make EVA missions more efficient and effective. In order to accomplish this main goal, the systems integrated within ARGOS work as a cohesive unit, each contributing to the success and efficiency of the others. User interface, interaction design, communications, and OCTaVIA (our RAID One inspired ad-hoc network and assistant) will work in unison to reduce cognitive load for the user. In order to meet the needs of astronauts and mission requirements, the UI architecture supports multi-modal functionality that include a work mode and navigation/exploration mode, and a universal menu containing suit vitals and data collection tools that are accessible from any active mode. Below, we examine how each component within the system contributes to the goal of bringing maximum efficiency to the field.

III. EVA Operations

The user interface of ARGOS, which is a head mounted display (HMD), utilizes a minimalistic display to present relevant data to the user while keeping the field of view unobstructed. Vitals information and consumables, such as H_2O, O_2, Battery, Pressure, and Total Mission Time, will be constantly present to the user, as shown in Figure 1. H_2O, O_2, Battery, and Pressure, are presented within the user's field of view in the form of bars in the left-hand corner. These bars will use a color gradation system to indicate the state of each value, with green indicating a healthy status, yellow indicating a degrading status, and red indicating low or warning status. By formatting the bars in this way, the HMD presents the information to the user in a way that is easy to consume and process. This element is crucial in reducing cognitive load for the user.[2]

Figure 1: Main menu that lets the EVA operator select the task to complete.

In order to maintain an unobtrusive design, the HMD utilizes two versions of telemetry display: minimized and maximized. For instance, the HMD will always display the primary values for telemetry data related to H_20, O_2, Battery, and Pressure while minimized. However, at any given point the user can maximize these values to show an expanded view containing all the categories' sub-values, such as back-up battery. This allows the HMD to strike the right balance between providing all information to the user and keeping the field of view open and free for the user.

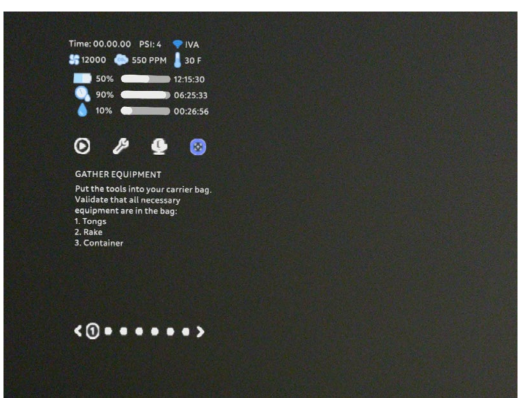

Figure 2: Sample stepwise procedure associated with the Scientific Sampling mode.

Outside of presenting telemetry data to the user, the HMD must also provide the user with detailed instructions for the task at hand. In order to keep in line with the minimalist design of the user interface, the HMD uses an independent, doc-able, virtual manual. The user may select a location to spawn or create the manual, bringing the manual into view, as shown in Figure 2. Then the manual will remain stationary and unaffected by head motions while the user moves about the space, thereby keeping the field of view clear. The user can adjust the position of the manual incrementally through hand or voice interaction, as well as replace the manual entirely in relation to the new user position. By keeping the manual's position independent from the information displayed within the HMD, the user is able to reference the documentation at their own pace, while keeping their field of view unobstructed. The manual will include detailed step-by-step instructions of the task at hand, a user-friendly paging system, and embedded animations for each step. These three elements coalesce to create a clear and concise manual that appeals to different types of learners. By including text instructions in tandem with visual aids, ARGOS HMD ensures that the instructions are accessible to all users, even those with little or no experience with the task at hand.

One of the most significant challenges when designing augmented reality displays is the absence of control over the user's lighting conditions. This challenge is magnified when considering the extreme conditions of the Moon, where the environment can either be well-illuminated or completely shadowed. Due to artificial or natural sources, augmented reality user interfaces should work as intended in any lighting conditions. For this reason, ARGOS' user interface will incorporate appropriate levels of contrast for every lighting context to make sure it stands out in both bright and dark environments. To accomplish this goal ARGOS incorporates a light and dark mode palette swap for the elements of the UI. We intermittently took a picture of the field of view and analyzed it to get indicators like brightness and total internal reflection, and then did some projections to get the total luminous intensity in lux. We used lux in place of candela mainly because lux primarily measures light exposure and not necessarily it's intensity. This allowed us to eliminate unnecessary variables, which made calculations easier. When lux values are above 80,000 the UI automatically switches to light mode and when it drops below 80,000 the UI switches to dark mode.

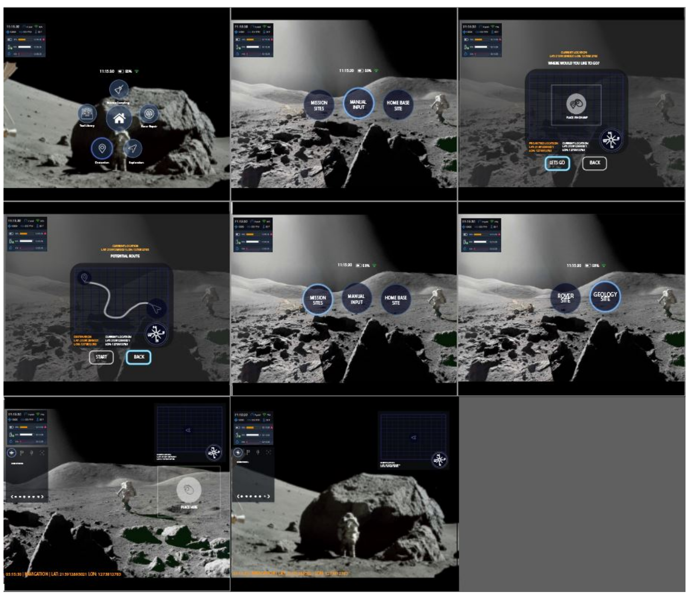

Figure 3: Render of the translation mode using waypoints and/or breadcrumbs.

The last scenario that is part of our UI is the Translation mode, rendered in Figure 3. The system guides the EVA operator by using different methods, including waypoints and breadcrumbs. In the case of waypoints, the system would utilize beacons that have been previously set by others or architected through some type of GPS-like coordinate system. In the case of breadcrumbs, the EVA operator will utilize equipment to disseminate "breadcrumbs" along the path, which will serve as a temporary waypoint for the way back as well as communications. This last type of wayfinding is part of the responsibilities of OCTaVIA, described later in the paper.

B. Interaction Design

In order to interface with our HMD, ARGOS has implemented a multi-modal approach to interaction. The multi-modal implementation gives the user multiple ways to interact with the HMD, while simultaneously creating a failsafe system in case one mode of interaction becomes unusable. By including multiple forms of interaction, the ARGOS HMD ensures there will always be a way to interact with the user interface despite any complications that may arise. With the aim of creating a robust interaction system, ARGOS supplies the user with 4 modes of interaction: Gesture Mapping, Eye Tracking, Voice Commands, and the ARGOS-specific RCA, which will be discussed in the section dedicated to IVA/MCC Operations.

1. Gesture Mapping

One of the strongest features of ML1 is its ability to recognize, map, and interpret hand gestures. With eight unique recognizable hand gestures, the Magic Leap far outperforms the original HoloLens. Therefore, ARGOS leverages the ML1's capability to create a robust form of gesture-based interaction. Furthermore, by selecting specific hand gestures used by the system, ARGOS avoids over-burdening the user's cognitive load. Since there are physiological limitations to how humans process new experiences and recall information from memory,[3] ARGOS' selective use of gestures ensures that the gesture-based form of interaction is intuitive, easy to remember, and does not overwhelm the user or distract them from the mission.

2. Eye Tracking

When considering the realistic nature of completing scientific surveying lunar missions, one must consider restraints that may arise when completing the mission. For instance, if the user's hands are occupied with tools throughout the survey, a gesture-based system quickly becomes cumbersome. Therefore, the ARGOS HMD implements eye tracking as an additional form of interaction. Using a gaze and hold method of input validation, the HMD allows the user to interact with elements of the user interface without the use of the hands. This makes interaction while holding tools much more efficient, removing the need to put down tools in an effort to interact with the HMD.

3. Voice Commands

Previous iterations of ARGOS relied heavily on hand gestures as the primary form of interaction between user and the HMD. However, through testing and feedback we have ascertained that hand gestures may not be the most effective way to interact with our user interface. Due to the weight and cumbersome nature of the spacesuit, frequent raising of the arms to interact with the UI can quickly lead to fatigue. In order to alleviate this constraint, ARGOS will incorporate voice commands as the primary form of interaction between user and interface. This will allow for more streamlined interaction, lessening the cognitive load, as well as reducing the physical strain the suit has on the user's body, allowing them comfort and longevity while on difficult missions.

In addition, voice commands allow for an effective method of completing field notes. Without access to a keyboard, methods of note taking such as gestures or eye tracking prove slow and laborious. However, with the ability to capture field notes through speech to text, the user can take notes while conducting the geological survey. Thus, increasing efficiency by removing the need to start and stop the survey or place down tools to take notes.

IV. IVA/MCC Operations

A key component to ARGOS is the integration of the Remote-Control Application. The RCA will work in tandem with the ML1 to aid users throughout lunar missions. Through the RAID One inspired network, OCTaVIA, the RCA will connect to the Magic Leap from within the lunar base. The RCA grants the user within the lunar base the ability to see live vitals (as shown in Figure 4), interact with the HMD on the Magic Leap, and send relevant documents to the Magic Leap user in the field. The RCA gives astronauts in the field the ability to completely outsource their cognitive load to an on-base user, allowing the RCA user to navigate and control the user interface while the field user remains free to focus on the mission. The RCA was a key element to the design of the first version of ARGOS, and with further iteration, the RCA provides a vital element to the HMD. This reduces the mental overload for the user, while simultaneously providing aid throughout the mission by incorporating real time assistance from users within the lunar base. This also eliminates the communication delay currently present in Moon to Earth communications.

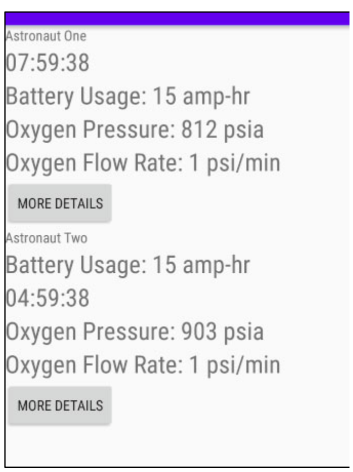

Figure 4: EVA operators telemetry summary view.

An additional functionality of the RCA allows ARGOS to connect to external entities, such as the rover, in order to streamline interactivity between user and rover. Pairing the RCA to the rover allows for diagnostic, location, and telemetry data of the rover to be passed to the IVA/MCC operator. With the rover data, the IVA/MCC operator can aid with rover maintenance, navigation, and monitoring of vital levels. Overall, connecting the RCA to the rover provides the IVA/MCC operator with all relevant rover information needed to make the mission safe and more effective.

V. Communications and OCTaVIA

Although the ML1 has cutting edge features out of the box, the headset still has limitations such as software restrictions to lower-level hardware access, an essential (but growing) documentation, and a hardware that is optimized for AR/MR functions rather than general information processing. Due to this, ARGOS has employed an Operations Control, Translation, and Visual Interface Assistant, otherwise known as OCTaVIA. OCTaVIA is a RAID 1 inspired system with a communications infrastructure that was designed for disaster recovery and computational load balancing.[4] By employing OCTaVIA's services, ARGOS is able to work around the limitations that the hardware currently presents. OCTaVIA is comprised of 8 Raspberry Pis that operate as nodes within a Kubernetes infrastructure. These nodes operate independently from one another and have the capabilities to send and receive information to and from the HMD on the ML1. The nodes will be deployed between sites during translation to destinations. This means that the user is capable of placing a node within a desired area to broadcast the ad-hoc network, making each node an access point for the network. Each node verifies connectivity to at least one other node at all times.

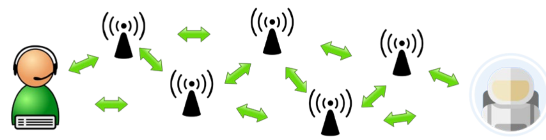

Figure 5: Overview of OCTaVIA's connectivity.

A dynamic connection between the nodes, depicted in Figure 5, allows for the physical infrastructure of the network to be changed on the fly. For example, if the node that the HMD is currently connected to fails or has a poor signal, the computational balancing will dynamically move to the next available and most efficient node within the astronaut's range. OCTaVIA's nodes are also responsible for database storage for lunar requisitions and location tracking. This type of connectivity can also be leveraged to provide wayfinding in unknown territories, by implementing a breadcrumb-inspired navigation system using Bluetooth.[5]

The system architecture of each node is relatively simple, as shown in Figure 6. We used 8 Raspberry Pi 3+ systems, and replaced the operating system with Ubuntu Server OS. This allowed us to run 64-bit versions of all the software required, instead of the 32-bit version which was a limitation of Raspbian OS for our devices. Kubernetes is then used for load-balancing among nodes, and the execution of any software is tied to Docker containers. These containers allow us to package an application of any type to run in this virtual system, allowing IVA operators and MCC to easily deploy new functionalities onto OCTaVIA without limitations imposed by the operating system or the software that is already present on the nodes.

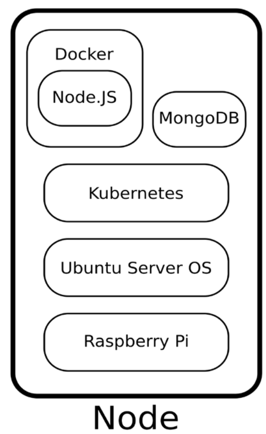

Figure 6: System architecture for each node.

This architecture was chosen so that we can alleviate the responsibility that is typically assigned to a single computer, typically referred to as the server, and spread the ability of processing the same type of requests to any connected node.[6] In our case, this is possible by allowing each node to process requests through the Node.JS server, which in turn either processes the information itself or refers to the MongoDB service. We chose MongoDB because it is much easier to mirror instances of the document sets among nodes, while retaining the read/write ability of each single node. This would be more difficult to implement on a device such as the Raspberry Pi with relational databases, since they typically require either a large infrastructure for data mirroring, or a server-style configuration with one read/write node, and then mirror nodes that only allow for read operations. A relational database setup would be a hinderance in scenarios where the server node cannot be reached, and the EVA operator needs to store information on the assistive OCTaVIA node that is closest to them.

The components that are currently running on each node are an API for telemetry, one for exchanging messages among connected operators, and one to save field notes. Other functionalities that are currently under development for OCTaVIA are voice recognition (given the limitations of the ML1 API), and object recognition to facilitate the EVA operator's task in contextualizing the UI for the appropriate operations.

VI. Preliminary Tests

The system that is described in this paper is complex, and has several moving parts. Since each has been studied relatively well, but they have never been put together in this particular way and with the mission requirements that we described earlier, we have much testing to do. We started with usability tests, since the main purpose of this system is to provide the ability for an EVA operator to perform a variety of tasks. We also looked into the physical limits of the system (in particular the ML1 headset and OCTaVIA's nodes). Preliminary tests included power usage and thermal behavior of the nodes, in order to minimize power consumption and optimize the performance of each component. In this section we present preliminary results that will be further discussed in later publications.

C. Usability Tests

The usability test procedure lasted approximately 30-40 minutes and was divided into three portions: ML1 demo, usability test, and exit interview. For each test, there were two facilitators and one participant. One facilitator guided the user for certain portions of the procedure and the second facilitator documented observations of the user's behaviors during the test, with zero interference.

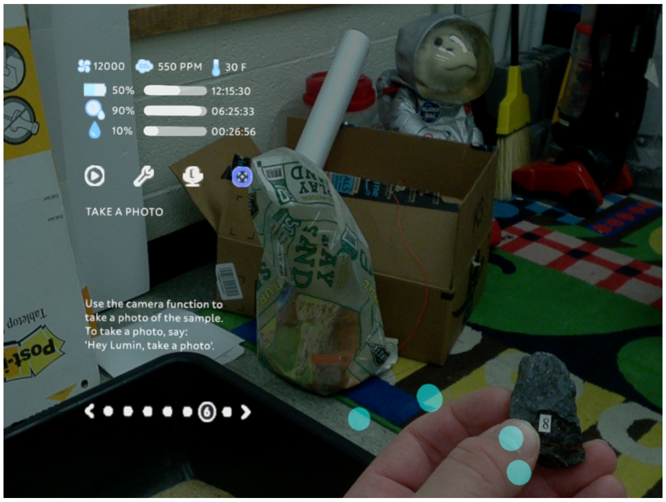

Figure 7: Sample rock collection during testing.

Each participant was asked if they needed to adjust the headset with prescription lenses. Once the participant put on the ML1, the facilitator demonstrated the hand gestures needed to navigate through the system. After the hand gesture demo, the facilitator asked the user to open the main menu and say out-loud what menu option they would select. The user was subsequently instructed to select Science sampling, and encouraged to complete the rest of the instructions on their own. Figure 7 shows a snapshot of the sample taken by the participant using the voice-activated feature. The facilitator only communicated with the participant to encourage them to think-aloud during the test. Minimal facilitator interference was practiced ensuring users were primarily utilizing ARGOS to accomplish their task.

The research team utilized a two-phased usability test of 11 participants consisting of task-based observation and post-activity interviews. The results can be broken down into the following categories: information architecture, interaction design, visual design, and situational awareness.[8]

We examined information architecture though the lens of information hierarchy and content design. These findings resulted in examining the main menu, the instructional content, and pagination of information. Through observation, we found that 36% of participants read through all the instructions before starting the process and there was a 75% average success rate with making the proper menu selection. During the interview, 72% of participants expressed positive feedback about the step-by-step content. These results revealed that users were not over stimulated by the information presented in ARGOS, and suggested more information would improve their experience.

In this study, interaction design is defined as interactions such as hand gestures and voice command used by participants to complete their tasks. After reviewing our observational data, we found that 36% of participants struggled with balancing hand gestures and performing the real-world task. Another finding was that 27% of participants held hand gestures after the system reacted. In the interviews, 27% of participants mentioned that the ML1 system registered unintended hand gestures while 45% found the voice interaction for the camera and audio recording to be intuitive. Interaction related results revealed that users experienced difficulty when using hand gestures for interaction while showing no signs of struggle when practicing voice commands. From an interaction design perspective, a potential cause for the issues with hand gestures is the interference of utilizing hand motions to execute physical tasks and use of hand gestures for controlling the AR system.

In order to assess the factors of visual design, we looked at the UI components which included layout, 3d models, color, and typography. The most commented upon visual design and UI component was the progress tracker. All of the participants mentioned the progress tracker in some way while completing the task. Conversely, only 27% of participants mentioned the progress tracker as useful in the post interview. Participants did not think the interface was obtrusive (81%) and only 27% of participants reported difficulty in reading the text. Most interesting, only 18% of participants reported that the 3D models were helpful. Overall, the visual design was received well, as nine participants expressed the visual interface design did not obtrude the FOV and were able to complete all tasks without visual interference. An area of improvement includes legibility issues of the instructions, as three participants reported difficulty reading the instructions. The cause could be due to scale, color contrast issues, or vision impairment.

In terms of situational awareness, 36% of participants showed difficulty transitioning awareness from virtual to physical tasks. However, only 9% of participants reported difficulty orienting to AR. The results of this study revealed that users may have difficulty switching context between virtual elements and the physical world. Four participants did not anticipate completing the scenario in the real-world. When tasked to gather rock sampling tools, participants were unable to locate the tools because they assumed the tools would be virtual-based. One participant expected the sample site to also appear virtually and attempted to rake the floor instead of the sandbox. When participants came to the realization the instructions referred to the real world, it was apparent their demeanor changed and immediately understood how to complete the task. After this recognition, all participants understood the boundary between physical and virtual elements and were able to complete the scenario.

The findings from this exploratory study prove the value of using an augmented HMD to assist beginner users in procedural tasks. This study contributes the findings related to information architecture, visual design, interaction design, and context-switching to the augmented reality industry. Additionally, this study contributes to the design fiction method of prototyping concepts for emerging technologies. These discoveries should be considered when designing a wearable augmented reality interface for procedural methods. It is encouraged to prototype by utilizing traditional prototyping tools in order to discover successes and failures in design with greater efficiency. It is safe to assume the ARGOS interface design was successful in supporting novice users, as all participants reported they felt educated on how to collect rock samples. The results from this research affirm the effective use of Nielson's foundational design heuristics and Gestalt principles. Additional discoveries from this research that should be considered when designing a digital interface for HMD wearable augmented reality experiences include spatial awareness, cognitive overload, and context switching. Given all participants vocalized they would use an augmented reality system in the future for advanced procedural tasks, future steps for advancing ARGOS involves integrating a complex task in order to stress-test the existing design.

This research has revealed an interest in using augmented reality HMD for procedural tasks. Wearable augmented reality is considered an emerging technology in an industry with room for advancement. Thus, the future of the augmented reality industry holds opportunities for hardware, software, and prototyping tools that may increase in availability. It is recommended that designers, developers, and researchers continue to iterate, experiment, and contribute research findings as this technology progresses and increases in accessibility.

The phenomenon of augmenting reality through virtual elements to enhance the physical world is progression towards amplifying human perception and enabling human cognition. The addition of other emerging technologies such as including real-time comprehension and object recognition could contribute to enhancing reality. Real-time comprehension and object recognition would allow the system to learn and adapt to the user's situation, and dynamically provide a solution to the existing task. These technologies are parallel to the enhancing user reality through data – real-time adaptive information integrated into the physical world to amplify human cognition. Ultimately, any advancement of augmented reality is designed to enhance the physical world to enable individuals to learn and take action.

D. EVA Power Consumption Tests

To implement a mixed reality system, we need to have a computational platform that is capable of generating and managing virtual objects in the real world. The platform must also be capable of tracking all the virtual objects generated in the real world space. Next, we need to display these objects via some sort of medium. In many cases these media are in the form of headsets and mobile hand-held devices. It should be noted that other functionality like hearing and touch can also be augmented. Displaying these objects is not the only criteria, we need to align the objects correctly in the field of view, and alignment requires converting the size of virtual objects relative to the render space. Wearable peripherals like remote controls also allow users to interact with the augmented world. In cases where the mixed reality device needs to communicate with other devices, wireless networking is needed to facilitate that functionality. These many requirements pose some significant limitations to the use of mixed reality systems. One of such limitations is augmenting digital objects outdoors including immersive GPS navigation and annotation of relevant information on buildings and stores. The number of peripherals required for outdoor augmented reality is bothersome and accurate tracking outdoors is also daunting.[7] The mixed reality device has less control over the outdoor environment and there are very few resources like sensors to track every single activity going on. In mixed reality systems, there is a direct proportion between perfecting the technology and battery life. As mixed reality platforms continue to improve, the power consumption on these devices also increases. The technical problems as it relates to energy consumption in both augmented and mixed reality are quite similar. Both technologies require a significantly large update rate which requires making some trade-offs with battery capacity. We sought out to identify the factors that contribute to energy consumption, so we could identify innovative methods to minimize energy consumption.

Upon examining the elements that contribute most to energy consumption, two main factors were identified:

1. The computations a system performs
2. The amount of data being processed

If these problems are not addressed at the pre-production and manufacturing level, developers would have to seek other methods to mitigate these two factors. For this study, we focused on two factors of power consumption that can be easily addressed at the post-production level. It is important to note that the methods highlighted here are only applicable to localized mixed reality systems and only at the post-production level. These methods were also experimented on a particular case study with ARGOS.

To observe the power consumption of EVA with regards to the two power-consuming factors mentioned above, we isolated the telemetry feature and experimented with the effects of regulating the impact of the two factors. The telemetry feature is essentially intended to retrieve a list of telemetry data from a server and then proceeds to sort the data received in order of priority and displays the sorted data to the astronaut on the HMD. We separated this feature into steps that allows our tests to adjust the number of computations performed and the amount of data being processed by the EVA system. We implemented a latency layered architecture with OCTaVIA using web sockets which allows EVA to only receive telemetry data when they are available without having to unnecessarily send intermittent requests which may not return any new data using a conventional server-client communication mode of representational state transfer (REST). A REST architecture requires the client (EVA) to make a request to establish a connection with the server (OCTaVIA). When the connection is established, the server returns a response and the client processes that response. This structure can be generalized to a two-step process on the client side: 1) Send a request, 2) Get a response.

These two steps will be repeated each time the client wants to receive updated information. With a web socket architecture, connection is established only once with the server allowing a free flow of responses from the server without the client explicitly making requests. Apart from the initial connection, a web socket can be generalized to only one step on the client side: 1) Get a response. This reduces the total number of high-level computations required to receive updated telemetry information by approximately half.

To regulate the amount of data being processed on EVA, the telemetry sorting task was designed to be performed either on the EVA system or on OCTAVIA. In addition, different sorting algorithms were tested to examine their impact on EVA power consumption. The sorting algorithms tested were grouped into two main categories: optimal algorithms and suboptimal algorithms.

The criteria for grouping sorting algorithms are based on the currently known asymptotic runtime analysis of algorithms in Big-O notation. For this study, we consider the median time complexity as $O(n \log(n))$. Algorithms with a runtime faster than or equal to $O(n \log n)$, such as Merge Sort and Quick Sort, were put in the optimal category. Algorithms with a runtime slower than $O(n \log(n))$ like Insertion Sort and Selection Sort were put in the sub optimal category.

Table 1: Overview of the conditions for each simulation.

Simulation	Conditions	
	Latency Layered Architecture	*Optimal Algorithms*
Experimenal 1	Yes	Yes
Experimental 2	Yes	No
Experimental 3	No	Yes
Control	No	No

The tests were performed by running 4 simulations in three intervals of 15 minutes each. Table 1 reports the summary of the simulations, where the first three alternated the conditions of implementing both a latency layered architecture and optimal algorithms or implementing either of them, and the fourth served as control implementing no latency layered architecture and no optimal algorithms. The total power consumed by each simulation was collected based on the average difference of the battery percentage at the beginning and end of the simulation.

Table 2: Summary of the results and projected estimated runtime.

Simulation	Average usage over 15 min	Usage per minute	Saving over Control	Estimated Runtime
Experimenal 1	6.3%	0.42%	26%	238 min
Experimental 2	7.0%	0.47%	18%	213 min
Experimental 3	6.3%	0.42%	26%	238 min
Control	8.6%	0.57%	---	175 min

After running all tests, we obtained the results summarized in Table 2. Simulation 1, testing a combination of a latency layered architecture and optimal algorithms had an average power consumption of 6.3% power. Simulation 2, testing a combination of a latency layered architecture and suboptimal algorithms had an average power consumption of 7%. Simulation 3, testing a combination of no latency layered architecture and optimal algorithms had an average power consumption of 6.3%. Finally, the Control Simulation had an average power consumption of 8.6%.

The results elicited supported the hypothesis that the use of efficient algorithms and latency layered architecture does contribute to saving energy consumption in localized mixed reality systems. The most significant differences were between Simulations 1, 3 and the Control Simulation. The total power consumed in Simulation 2 was closer to Simulations 1, 3 than to the Control Simulation. However, the amount of energy optimized cannot be accurately computed based on the readings recorded. We continue working with the ML1 and become familiar with its architecture, however we do not yet have a firm hypothesis regarding possible reasons behind the results.

Results were recorded purely from the battery percentage report on the magic leap one dashboard so it did not take into account the approximation process used by the Lumin Operating System on which the device runs. Therefore, these readings may not have been an accurate measurement of the amount of energy consumed. Furthermore, battery temperatures at all instances were ignored and that also may or may not have affected the accuracy of readings.

VII. Conclusions and Future Directions

The system described in this paper, along with the preliminary results reported here, support that our group has identified a system that we can use for training and to model the design and implementation of future augmented reality-enabled systems for space exploration. The development of this system will continue, especially by adding more supportive features that will enable the EVA operator to integrate more AI-based features, to be supported by OCTaVIA. We will also continue studying the dynamics and performance of each node, as well as the entire cluster, in this serverless computing model. We will also work on translating this system into more immediately necessary fields, such as disaster recovery as well as training for procedural tasks associated with mechanical repairs. Lastly, it is worth noting that this infrastructure has been used to increase the engagement of computer programming students in the higher education institution associated with this group.[9] We will continue utilizing ARGOS as a research testbed as well as a way to engage undergraduate and students through a hands-on educational approach. The platform described in this paper is currently being extended to meet the requirements of SUITS 2021, and future publications will describe its operations in more detail.

Acknowledgments

The work reported in this manuscript was in part funded by the University of Baltimore, the Turner Student Research Awards, the University of Baltimore Foundation, and the Maryland Space Grant Consortium.

References

The following pages are intended to provide examples of different reference types. When using the Word version of this template to enter references, select the “references” style to automatically format your references. If you are using a print or PDF version of this document, all references should be in 9-point font, with reference numbers inserted in superscript immediately before the corresponding reference. You are not required to indicate the type of reference; different types are shown here for illustrative purposes only.

[1] NASA SUITS. Retrieved 2019, from https://microgravityuniversity.jsc.nasa.gov/nasasuits.cfm

[2] Harley, A. “Visibility of System Status,” Nielsen Norman Group. 2018

[3] Baddeley, A. (1994). “The magical number seven: Still magic after all these years?,” *Psychological Review*, Vol. 101, No. 2, 1994, pp. 353–356.

[4] Pogkas, N., Karastergios, G. E., Antonopoulos, C. P., Koubias, S., & Papadopoulos, G. “Architecture design and implementation of an ad-hoc network for disaster relief operations,” *IEEE Transactions on Industrial Informatics*, Vol. 3, No. 1, 2007, pp. 63-72.

[5] Mendelson, E. U.S. Patent No. 9,204,251. Washington, DC: U.S. Patent and Trademark Office., 2015.

[6] Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. “The rise of serverless computing,” *Communications of the ACM*, Vol. 62, No. 12, 2019, pp. 44-54.

[7] Azuma, R. T. “The challenge of making augmented reality work outdoors,” *Mixed reality: Merging real and virtual worlds*, 1999, pp. 379-390.

[8] Yee, C., “Wearable augmented reality in procedural tasks: Designing an interface used to deliver step-by-step instructions to support novice users in unfamiliar tasks,” M.Sc. Thesis, University of Baltimore, Baltimore, MD, 2020.

[9] Vincenti, G. “Engaging IT Students through the NASA SUITS Design Challenge: An Experience Report,” *Proceedings of the 20th Annual SIG Conference on Information Technology Education*, 2019, pp. 22-27. 2019.