A Cloud Computing Infrastructure to Support xEMUs and Future EVAs

John Manlucu, Najya Ahsan, Ruth Robinson, Michael Vandi, Kaitlyn Baker, Giovanni Vincenti[1]
The University of Baltimore, Baltimore, MD, 21201

To create a safer operational environment during EVAs, we propose OCTaVIA (Operations Control, Translation, and Visual Interface Assistance), a modular and robust system that incorporates industry-standard technologies used in cloud architectures. The main features include distributed data storage, processing and redundancy, a communications infrastructure designed for disaster recovery, computational load balancing, localization services, and passive safety mechanisms. We utilize a testbed composed of several Raspberry Pi and Jetson Nano devices placed on stationary posts within the EVA workspace, which operate as nodes in a Kubernetes infrastructure, a system that automatically deploys, manages, and scales containerized applications. The system offers distributed processing and communications through the Raspberry Pi devices, GPU-intensive computing devices with Jetson Nanos, and LiDAR-enhanced visual technology to support remote as well as automated monitoring of EVA worksites. OCTaVIA is part of ARGOS, a larger solution presented at ICES 2021 aimed at creating an information and communications management system that utilizes augmented reality to let astronauts interact with mission personnel and assets.

Nomenclature

AR	=	Augmented Reality
ARGOS	=	Augmented Reality Guidance and Operations System
DBMS	=	Database Management System
DHCP	=	Dynamic Host Configuration Protocol
EVA	=	Extravehicular Activity
HMD	=	Head-Mounted Display
ISaMS	=	Intelligent Sensing and Mapping System
IVA	=	Intravehicular Activity
JSON	=	JavaScript Object Notation
LiDAR	=	Light Detection and Ranging
MAE	=	Mobile Augmented Environment
MCC	=	Mission Control Center
NoSQL	=	Not Only Structured Query Language
OCTaVIA	=	Operations Control, Translation, and Visual Interface Assistance
PAM	=	Passive Activity Monitor
RCA	=	Remote Control Application
SSH	=	Secure Shell
STEM	=	Science, Technology, Engineering, and Math
xEMU	=	Exploration Extravehicular Mobility Unit

I. Introduction

EVAs are arguably one of the most dangerous parts of a mission[1]. With the advent of xEMUs, the safety and toolkit of an astronaut is greatly augmented; however, there are still significant limitations imposed by the environments that NASA plans to explore in the near future[2]. This paper explores how we can utilize cloud-like technologies and infrastructures to assist an astronaut's activities, augment the xEMU's capabilities, and ultimately create a safer operational environment.

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

OCTaVIA, or Operations Control, Translation, and Visual Interface Assistance, is a modular and robust system that incorporates industry standard technologies used in cloud architectures. The main architecture is inspired by a RAID-1 redundant data storage and transfer system[3], with a communications infrastructure designed for disaster recovery[4,5] and computational load balancing[6]. OCTaVIA also serves localization purposes, utilizing multiple methods to pinpoint an astronaut's position. We utilize a testbed composed of several Raspberry Pi and Jetson Nano devices placed on stationary posts within the EVA workspace, which operate as nodes in a Kubernetes infrastructure, a system that automatically deploys, manages, and scales applications. The system offers distributed processing and communications through the Raspberry Pi devices, as well as GPU-intensive computing devices with Jetson Nanos. The system also integrates LiDAR technology to support remote as well as automated monitoring of EVA worksites.

This manuscript describes OCTaVIA in detail focusing on its underlying IT architectures, and reports initial performance results of its location and distributed processing systems. We also report proof-of-concept data about our LiDAR-based systems aimed at terrain sensing and astronaut passive activity monitoring. The results support its viability as an infrastructure for an autonomous and resilient support system for astronauts during EVAs.

II. System Overview

ARGOS originated as a student-driven, faculty-supported project in response to the NASA SUITS (Spacesuit User Interface Technologies for Students) Design Challenge, one of NASA STEM's Artemis Student Challenges[7]. This system is now a platform that supports educational initiatives[8,9] as well as undergraduate[10,11,12], graduate[13,14], and faculty research.

ARGOS, depicted in Figure 1, allows for the interaction of multiple entities, such as IVA/MCC controllers, astronauts on EVAs, and other autonomous systems. The main communication infrastructure is provided by OCTaVIA, a series of nodes interconnected through a Wi-Fi network. The primary mode of interaction is through MAE, an augmented reality application that provides an HMD to the user. This system is a head-worn device that takes incoming data from OCTaVIA and xEMU space suits to provide information relative to the user's location, telemetry data, and hosts navigation, scientific sampling, emergency and remote-operation controls, and a warning/alert system.

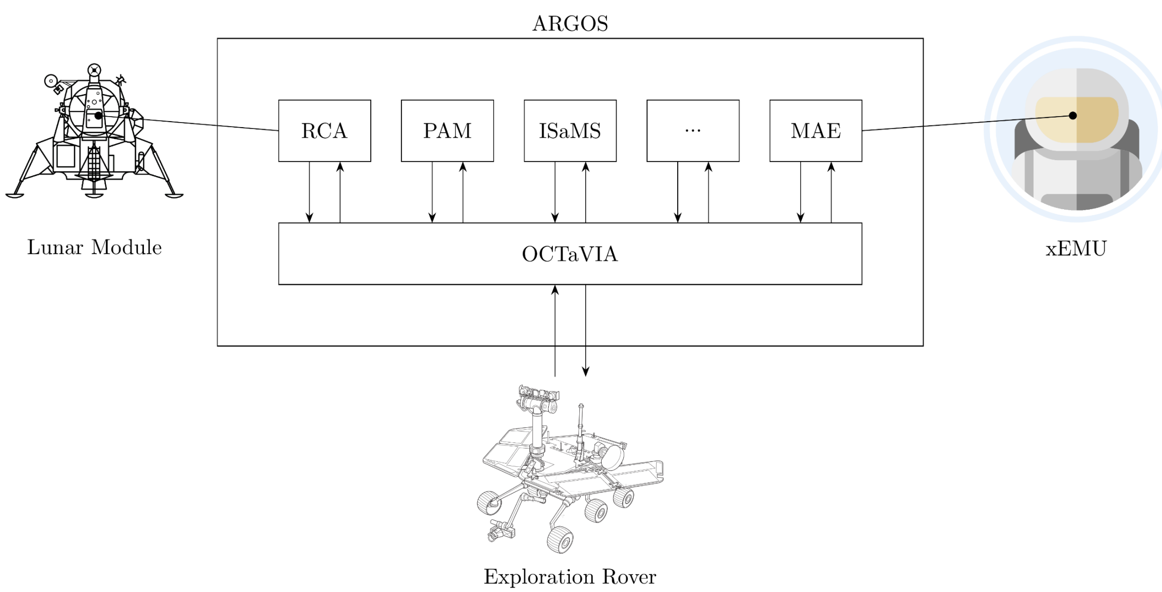

Figure 1: OV-1 depiction of ARGOS, including modules that extend OCTaVIA's capabilities.

Access to ARGOS and any of its features can be achieved by any device that can be connected wirelessly, such as an exploration rover or other equipment. As OCTaVIA is the backbone of ARGOS, and it consists of interconnected devices providing services, it can be easily expanded to include new features that are readily available throughout the entire system by adding them as simple services.

A. Mobile Augmented Environment

MAE is the Mobile Augmented Environment that can be integrated into an xEMU, bringing ARGOS into the spacesuit. The features described next are accessed through MAE and are an integral part of ARGOS. As MAE is perhaps the most apparent of all systems, we will introduce it briefly. OCTaVIA and any module attached to it, including some discussed later in this paper, will support the astronaut and will be accessible through MAE.

1. Input controls

ARGOS uses hands-free controls to operate the system. The main input types for these operations are gesture control and voice command. The HMD is preprogrammed to take in preset and customized gesture controls. A selection of these preset controls has been added to the AR system to help the user select elements and traverse through various windows and menu items using the designated hand gestures. Voice control has a series of preset commands that can be used to select elements, open and close windows, and issue commands related to navigation. ARGOS peeks for voice input and is activated when the phrase "Hey ARGOS" is recognized. After activation, the user can voice a command from the list of preset voice activated commands to enable the system to perform an activity.

Figure 2: Screenshot of what an astronaut sees in the HMD. The dark area in the center can be attributed to the low light in the environment when the screenshot was taken.

2. Telemetry

The telemetry values related to the astronaut's vitals and various life-supporting tanks and other equipment attached to the xEMU suit can be displayed in the top left corner of the user's view with the HMD, as shown in Figure 2. Additional telemetry of other astronauts on the same EVA can also be recorded by each device, allowing a secondary window to be displayed that allows the user to view other's telemetry data.

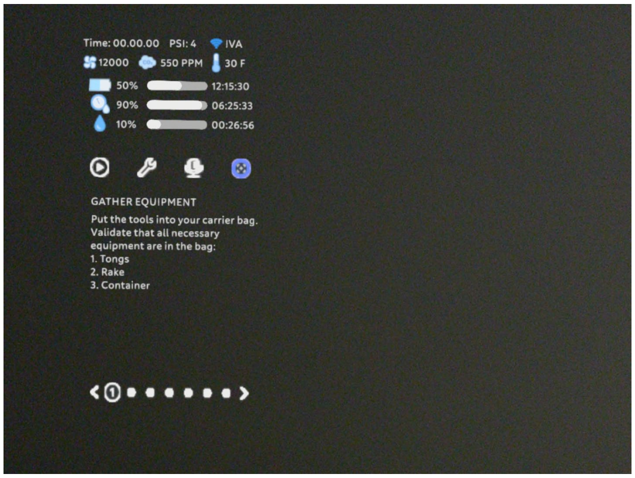

Figure 3: Procedure checklist in MAE.

3. Navigation

The upper-right quadrant in Figure MAE shows the minimap that gives an overview of the user's position. The central white arrow shows the position and orientation of the astronaut, while the orange pins show locations of interest. The dotted path shows the trajectory that the astronaut should take to reach the next waypoint. The list of waypoints is provided to MAE through OCTaVIA, and it can be managed by RCA, as reported later in this work.

4. Procedure Guidance and Scientific Sampling

The main goal of NASA SUITS is for student participants to create a system that allows EVA operators to easily access supportive features. In the years where our students participated, the main tasks were procedure guidance and scientific sampling. MAE supports these activities through a series of stepwise procedures that clearly identify what the astronaut has to perform and the tools necessary. Figure 3 shows the procedure guidance mode, where the astronaut can see the directions as well as an animated depiction of what needs to be done when following a procedure. In Figure 4 we can see an example of the user stepping through the soil sampling procedure[13].

5. Alerts and Warnings

Predetermined thresholds have been calculated to determine the normal, alert, and emergency levels of telemetry values. If any telemetry values surpass the normal threshold, an alert will notify the user of their current issue.

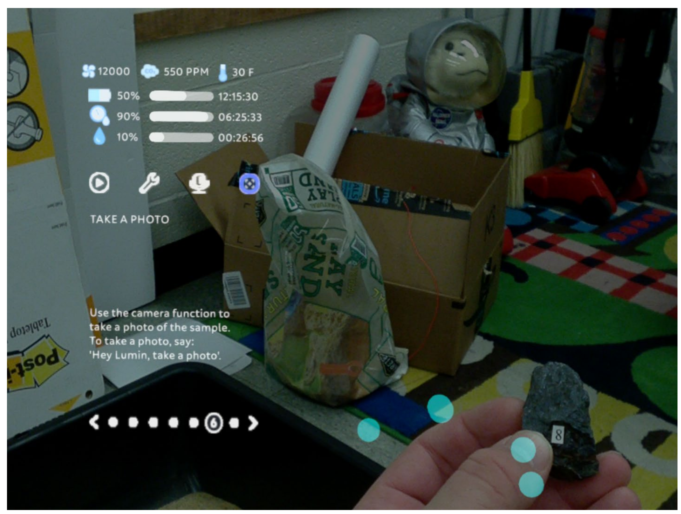

Figure 4: Scientific sampling scenario in MAE.

6. Emergency remote-operation system

In the event of an emergency, ARGOS can activate an emergency mode that allows RCA to activate guidance operations on the user's HMD. Upon activation of this system, the user can allow mission control to guide the user back to the lander without the need of any input controls or wait for others on the EVA team to arrive for assistance. Other members on the EVA mission are alerted and shown the location of the astronaut in distress. This will then calculate the distance and form a directional path to the astronaut in need in case assistance is required due to immobility issues.

OCTaVIA

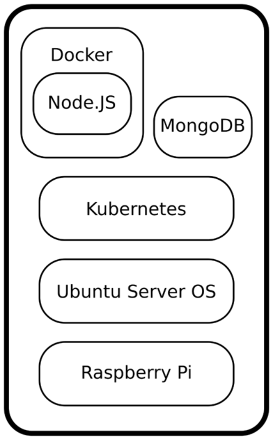

Figure 5: Overview of the internal architecture of each node.

OCTaVIA is inspired by the RAID (Redundant Array of Inexpensive Disks) data storage and transfer system, and in particular its Level-1 architecture, which focuses on mirroring of all disks to provide data redundancy in case of failure. In our case, the mirroring is not limited to data, but also includes the system's architecture and any applications running on it. Each node is an entire computing system with processing, storage, and communications capabilities, and is structured as reported in Figure 5. Each node is also a physical access point for new hardware, such as LiDAR cameras, that would augment the capabilities of ARGOS, as described later.

B. Networking

The main feature of OCTaVIA is the interconnectivity that each node can provide. This breadcrumb system, for which a scenario is reported in Figure 6, is highly adaptable to any scenario where communications can be made difficult by the layout. Even though it would be ideal to always pick sites with wide-open spaces, it may be necessary for an EVA operator to go beyond the line of sight. A system like OCTaVIA would then provide full connectivity, computational, and storage support to the operator, no matter how far from the base.

In this scenario, nodes are denser in the area that is farther from the base, in order to provide more support to the astronaut in case of system loss. In this example, we may also have a ridge blocking the path between N_5 and the rest of the nodes towards the xEMU, so the only channel of communication would be N_6-N_7. This situation may quickly change if we were to position another node within range of N_5 and N_9, so the network would then have added redundancy between the xEMU and the base.

This type of networking can be achieved with a wireless (mobile) ad-hoc network (or MANET), where each node establishes connections directly with any other device that is within range. This topology makes for the network to be self-configured and self-healing. This means that, should any node become unavailable, with enough density of nodes, others will pick up the communications and create a new path between any two nodes in OCTaVIA.

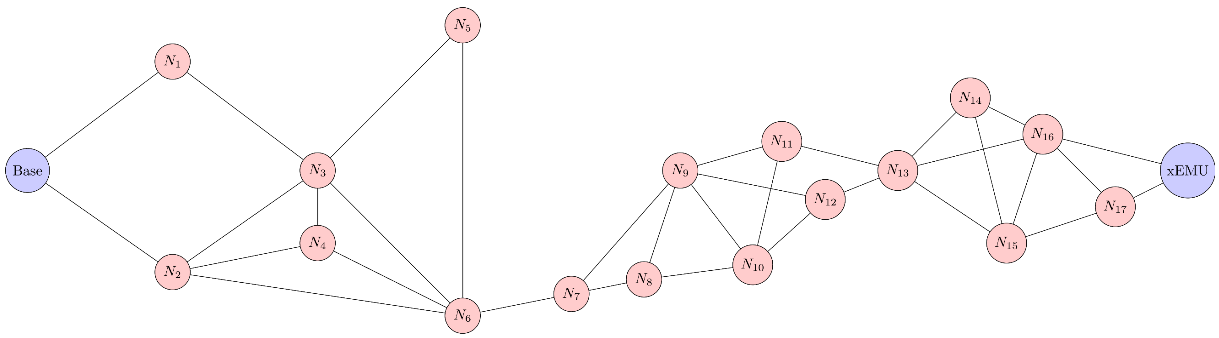

Figure 6: Sample deployment of OCTaVIA over a long-range exploration mission.

C. Types of application architectures

Any digital system will utilize software to turn its hardware features into fungible assets. Today's technologies allow for dynamic and adaptable systems that expand or shrink as necessary, running on a multitude of systems typically referred as "the cloud." As we created OCTaVIA, we had to decide on the basic structure of programs that run on it. In particular, the network topology and potential computational power offered by the nodes gives us much flexibility when choosing the architecture of any system. Examples in this section will reference the scenario depicted in Figure 6.

7. Standalone applications

The most basic structure of an application is typically considered standalone. This means that the application runs on a single system, thus the processing and data storage will be all local. This type of system is often utilized in older applications as well as in embedded systems. Although this type of configuration is necessary in some cases, such as in embedded systems, it is often accompanied by serious limitations.

As an example, we can think of a voice recognition system associated with the xEMU. In this case, should any component fail within the xEMU, the system will not be available to the astronaut. Another major drawback of a standalone application is that, when it is deployed on a battery-powered system, the application will rely solely on the battery system. If such system is shared among other electronic components, there may be the need to limit or halt its execution to ensure that other life-sustaining applications are running.

8. Multi-tiered applications

Modern applications gravitate towards the use of N-tier architectures, where a number of logically (and often physically) separate systems mange distinct roles that allow a user to interact with an application. The 3-tier system is the starting point, where the system is running on up to three different machines. The first device is responsible for the presentation tier, which serves as the user interface for the user. Then a server typically manages the application tier, which is responsible for the logic of the application. Lastly, another server is responsible for the data tier, which manages data storage and retrieval. The majority of modern applications utilize this paradigm, as web pages and mobile applications alike serve as the UI, while the logic and the data are handled remotely for either application type.

The main disadvantage of the multi-tiered application is that it typically requires several systems to run. For example, the presentation (or UI) layer may rely on a web browser, the application tier may rely on an application server such as JBoss, and the data tier may rely on a DBMS. Each of these can be bulky, adding significant overhead to the system that is running them, thus it is not advisable to run a multi-tiered application on a single system. It is, however, advantageous to run it on multiple systems.

A possible scenario for such system is that the application is running the presentation tier on the xEMU, and the application tier as well as the data tier at the base. This configuration will ensure that any major data processing associated with the application itself, as well as any data storage and retrieval, will be performed remotely, potentially saving resources on the xEMU such as battery power, limiting thermal output as well as task-switching. However, this scenario has drawbacks as well. If any of the communications between the xEMU and the Base are interrupted, the system will not be able to process the requests sent by the xEMU.

9. Tier distribution

As we safely split the application into multiple tiers, we can now utilize distributed systems where we allow each tier to run on more than one location. This is possible by utilizing appropriate management software, such as Kubernetes. As we take each tier and allow it to either process the business logic or the data management, we can now spread them to multiple nodes. This means that we are no longer limiting the processing to the Base, but we now have the ability of performing the computations and data operations at any node. This flexibility gives us the ability of offsetting the processing and data management responsibilities away from a single node, and building redundancy into the system, implementing an environment that can be characterized as serverless computing[15].

In this scenario, we can take into consideration a fully distributed system, where any node can serve for business logic computation or data processing, which is the goal of OCTaVIA. Looking back at a situation where there is no direct connection between the xEMU and the Base, for example if link N_6-N_7 were to be unavailable, the xEMU could still rely on the processing that is now taking place on any of the nodes that are reachable to it. By distributing both tier 2 and tier 3 to any node, we can also store the data in any node in a way that, should any node become unavailable, any other node can pick up the slack by offering the same processing environment as well as data assets and capabilities available on the rest of OCTaVIA.

D. Implementation Hardware and Software

OCTaVIA's original purpose of student engagement and participation in the NASA SUITS Design Challenge in ways relevant to our degree programs dictated the hardware choices that create the current system. The entire system is currently implemented using the following elements:

Hardware

- 8 Raspberry Pi 4 Model B, to create the main infrastructure of OCTaVIA
- 3 NVIDIA Jetson Nano, to process high-demand computational operations such as PAM
- 3 Magic Leap 1 Augmented Reality devices, which runs MAE
- 1 Microsoft HoloLens2 Augmented Reality device, which also runs MAE

- 2 Android tablets, which runs RCA
- 1 GPS receiver, to simulate a Lunar or Martian Positioning System

Software

- Ubuntu Server 20.04 LTS, as a base operating system
- Node.JS, as an application server
- MongoDB, as the original database server
- CouchDB, as a distribution-efficient alternative to MongoDB
- Python 3, as the primary language of development
- Custom-made applications and services, developed as necessary
- OpenPose, for the recognition of human figures
- Docker, for the containerization of applications
- Kubernetes, for the distribution, routing, and scaling of applications

To achieve a deployable serverless network for OCTaVIA, we followed a similar architecture to Microsoft's Serverless Distributed File System. This architecture allowed us to utilize all machines connected to the network and use them for processing power, rather than having one singular server that handles all the computations. OCTaVIA's network takes the next step. Instead of having a dedicated router for the network, we created DHCP servers on each node. We then set up a broadcasting ad-hoc network for each, eliminating the need for a router. Each node was assigned a static IP address to prevent IP conflict and continuous connectivity when the systems fail or reboot.

The DHCP server allows other external devices to connect and be assigned a leased IP address. This gives the user access to the network and remain in communications with any other connected device. In order to achieve this in a timely fashion, we initially used the local router to SSH and manually set static IP addresses for the nodes within the private network. This allowed us to manage the nodes directly, without the need for extra equipment. We maximized efficiency when developing for all the nodes at once during setup by utilizing the tool TMUX (terminal multiplexer) to SSH into multiple nodes at once and send mirrored commands to all systems.

10. Data Storage and DBMS Observations

Although it is possible to distribute database systems, it is usually not as easy as creating a distributed application. The main modification that we need to implement when distributing the application layer is to move away from assuming a stateful session. This means that every request must be handled as a singleton and not one request in a series. This can be done relatively easily by consistently sending the current application state along with the request, and the business logic will utilize what it receives in order to compute the response. This implies significant security concerns, however since OCTaVIA would operate in an environment with no malicious actors, such dangers would be nullified.

In the case of the database management system, however, statefulness is essential. We cannot re-upload all the historical data at every request because the current node may not have it available somewhere. For this reason, we have to rely on the DBMS to perform the distribution (including replication and updating) properly. Many systems are indeed capable of handling the distribution of data, however it is important to remember that most of the current technologies allow reads from any node but writes from a single, main node.

For example, assuming that the xEMU puts in a request for the transcript of the last five messages exchanged with MCC, then any node can serve it as the application is simply reading the database. However, if the xEMU is transmitting a communication that needs to be transcribed, then the speech recognition can happen at any node capable of handling it, for example N_{17}, but then it will contact the data tier to store the transcription and the original audio in the database. In case that the main Write node is any other, then we will run into a situation where we will create a bottleneck at the data storage level.

Some DBMS applications, such as CouchDB, allow for Read/Write at any node, whereas other more popular applications, such as MongoDB, utilize the single Write node. Currently we are utilize MongoDB for its simple integration with Node.JS through Mongoose, however we are planning on running stress tests on the system and identifying the real drawback of a single Write node and compare it to a fully distributed solution such as CouchDB. Moreover, we are currently staying away from relational DBMS applications because the enforcement of relationships would create a significant overhead in terms of storage time. It is possible to take the data from a NoSQL model and transpose it into a relational one through ETL (Extract-Transform-Load) operations, which can be done either in batch or on data streams at locations where the relationships are leveraged for queries, such as in cases for mission review and planning at MCC or at a Base prior to an EVA.

Modules Extending OCTaVIA's Operations

OCTaVIA can be easily extended by observing the main architecture described above. In particular, the distributed processing and storage can host many different types of customized services and extensions. We can also leverage either current hardware, such as the Wi-Fi and Bluetooth devices for the positioning system, or add new hardware, such as cameras for monitoring systems. From a practical standpoint, each new service that runs on OCTaVIA would simply require its own communications port. Once a port is dedicated to a particular service, the containerized nature of the application will take care of the integration with the computing environment and the distribution service will take care of the deployment and scaling of each container. In terms of peripherals, it is possible to add peripherals to the system by simply allowing it to connect to the network. Once it is on the network, all the services will be exposed and available to it. This section describes some of the modules that add features to OCTaVIA, to demonstrate its flexibility as well as chart the system's near future.

E. Positioning

NASA is working on a Cislunar Autonomous Positioning System Technology Operations and Navigation Experiment, or CAPSTONE[16]. Our basic assumption for OCTaVIA is that an accurate Lunar (or Martian) Positioning System may be in place by the time missions will deliver the first astronauts, however such system will likely need a backup. Very much like on Earth, where airplanes generally utilize GPS technologies to fly, they can also rely on radio-based navigation systems, such as VOR (Very high frequency Omni-directional Range), DME (Distance Measuring Equipment), and ILS (Instrument Landing System). For this reason, our system includes both GPS and radio-based positioning systems, such as Wi-Fi and Bluetooth[17]. This redundancy also allows us to utilize ARGOS indoors, where GPS signal is not available.

Given the hardware that we currently utilize, we decided to focus on Bluetooth rather than Wi-Fi, as this latter technology would require multiple antennas at each node[18]. We acknowledge that Bluetooth is particularly limited in range, and that Lunar exploration will require longer distances. However, this positioning system is a proof-of-concept for OCTaVIA and can be implemented using different technologies. Bluetooth is also more practical for ARGOS because the system is usually deployed in small, enclosed test environments, which make Bluetooth ideal for this scenario.

Bluetooth can be utilized for 2-dimensional as well as 3-dimensional tracking[19,20]. Moreover, the system can utilize complex triangulation and sensing systems, as well as an "out of the box" system that does not require much programming[21]. Currently, we opted for this second approach, where the triangulation is achieved by simply reading the RSSI (Received Signal Strength Indicator) value associated with the Bluetooth signal and that communications devices can easily detect to establish the quality of the transmission[22]. In order to establish a baseline for our indoor

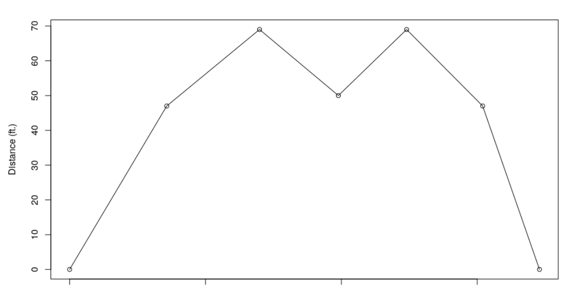

Figure 7: Distance from the stationary beacon.

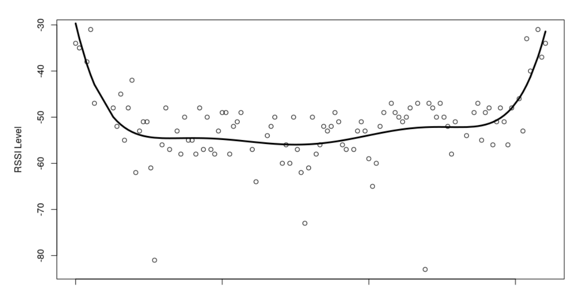

Figure 8: RSSI signal strength.

location system, we utilized a basketball court as a testing ground. The main receiver was placed at half-court on the south-side. The other receiver was carried around the bounds of the court to mimic an astronaut moving towards and away from a triangulation element.

The results, reported in Figure 7 and Figure 8, show the distance from the stationary beacon as the mobile one walks around the edges of the court (Figure 7). As the mobile beacon moves around, the RSSI value (Figure 8) fluctuates between -30 (stronger) to -80 (weaker). The figures show that, once the mobile beacon goes past 35-40 feet, the reading becomes relatively stable. This means that we can rely on the RSSI value with greater confidence as the static beacons are within 30-50 feet from the mobile one. Since we are using triangulation methods to estimate the position of the simulated astronaut (the mobile beacon), we can place stationaly Bluetooth devices throughout the work area at a distance of approximately 50-70 feet.

This work will be further developed as we wish to also incorporate Wi-Fi positioning as an alternative, in order to simulate longer ranges. Moreover, through empirical testing, we started looking at different levels of accuracy. In particular, we are interested in four levels:

- Level 0, the device on the astronaut can find at least three beacons within range
- Level 1, the device on the astronaut can get poor measurements (high standard deviation for a group of readings) from at least three beacons
- Level 2, the device on the astronaut can get somewhat accurate measurements (medium-low standard deviation for a group of readings) from at least three beacons
- Level 3, the device on the astronaut can get very accurate measurements (very low standard deviation for a group of readings) from at least three beacons

The multiplevel system would allow resources to detect the presence of astronauts in different work areas and optimize the computing resources dedicated to supporting the astronaut. For example, nodes that are farther away from areas where the astronaut is located may be reserved only for data storage purposes, providing redundancy with a relatively low power consumption. Areas that are actively utilized by astronauts should instead be fully functioning. A multi-level system may trigger the redirection of power and/or resources (and in particular load balancing for distributed processing systems), so that the astronaut will be able to rely on fast turn-around to any request.

F. Translation Planning

Another important part of this system is the planning and tracking, which is integrated in the RCA. This system can be utilized with Bluetooth positioning as well as GPS. During preliminary testing, we rely on GPS to minimize interferences associated with Bluetooth, and to have a redundant positioning system where our users and researchers can focus on the planning and tracking tool, rather than troubleshooting technical difficulties associated with interferences or malfunctions. The prototype is reported in Figure 9.

This system is implemented in Android, and runs on a 10” tablet. The prototype shows different elements:

- Lunar map with the current path (Figure 9 A)
    - The path reports the base in blue, intermediate waypoints in green, the current final waypoint in red, and the next candidate waypoint in pink
- List of preset waypoints (Figure 9 B)
    - Used to facilitate the inclusion of already-defined destinations
- Waypoint controls (Figure 9 C)
    - Used to set, edit, and remove waypoints
- Waypoint list (Figure 9 D)
    - A scrollable list of waypoints

The current screenshot does not include a view of the astronaut; however, it is possible to integrate the position recorded through OCTaVIA and visualize it in real-time on the application.

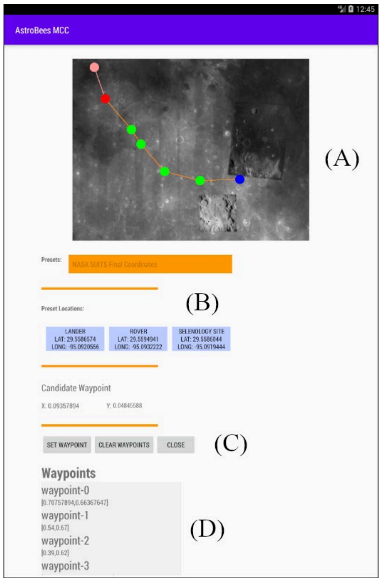

Figure 9: Prototype of the Navigation Planner and Tracker on the RCA.

G. ISaMS

Although satellite and historical imaging of the moon's surface have been captured, it is still a challenge for astronauts to achieve real-time navigational capabilities and obstacle avoidance during a spacewalk. By utilizing the scalable data storage capacity and computation power of OCTaVIA's network, our solution for navigation and obstacle avoidance is ISaMS, the Intelligent Sensing and Mapping System. This system is composed of two elements, one designated for mapping the astronaut's workspace prior to extravehicular activities and the other for terrain sensing of potential obstacles before and during extravehicular activities.

To first position ISaMS within the terrain, an astronaut or rover would have to set 'posts' within the workspace, to act as stationary high-vantage points for LiDAR image capturing. This will be integrated with the nodes of the kubernetes infrastructure and the Lunar Positioning System. The three systems will all be physically attached to the strategically-placed posts. The consideration for which posts to place the LiDAR cameras depend largely on image capturing range, physical obstacles on the terrain, power efficiency, and computational load balancing, which are later discussed in further detail.

Light detecting and ranging (LiDAR) technology is a remote sensing method that uses light in the form of pulsed lasers to measure variable distances to the land surface using the time it takes for light to depart to and return from the land surface. There are two types of Lidar, topographic (for land) and bathymetric (for water). Using topographic LiDAR, ISaMS will be programmed to map the surface and identify objects/land masses that are abnormally high or low elevation relative to its surrounding area. These masses will be considered obstacles in the astronaut workspace to some degree of accuracy. LiDAR is best suited for EVA due to its ability to produce high accuracy readings of depth and position for objects within a dim or dark visual space. In the anticipated Artemis III mission (expected 2024-2025), the Lunar south pole is of particular interest due to its occurrence of water ice within permanently shadowed craters known as cold traps. For this dangerous and dark terrain, having constant visual feedback of depth and positioning is crucial to the mission's success.

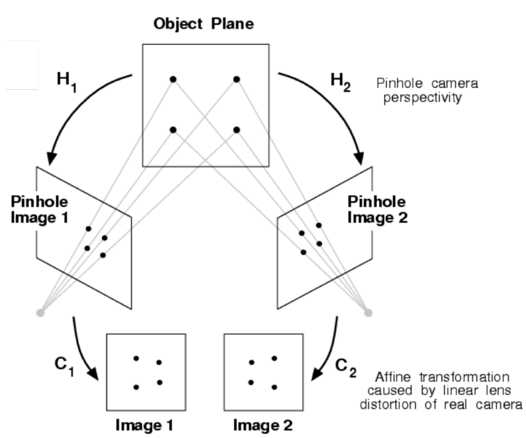

Figure 10: Visual representation of the process.

Upon placement of each post, the LiDAR cameras will initially map the terrain by treating each Lidar input as a 2D rigid plane with key identifying features (potential obstacles) existing on the plane. We will research and develop an algorithm that first performs pixel level image correlation, which is based on the statistical approach for finding the degree of similarity between two images at the pixel level. Two correlation methods, cross correlation in the frequency domain and fast Fourier transformation, are discussed in Tong et al[23]. Inspired by these methods, we intend to explore the alternatives with the highest computational efficiency.

Based on the pixels in the reference image (any of the LiDAR cameras) that have a high correlation with the pixels in the other camera images, we will identify homographies, or common features between the camera images. As seen in Figure 10, common features on the object plane from each camera perspective allow us to compute distance vectors between the camera projection (the pinhole image) and said object plane. This would create an alignment between the coordinate locations on each camera image via a homography matrix, which then directly translates to common coordinates on the object plane via distance vectors.

H. Passive Activity Monitor

Telemetry and communications help with ensuring that an EVA operator is constantly monitored, however the reality is that any gross body language related to distress situations may go unnoticed. For this reason, we propose a system called Passive Activity Monitor, or PAM, which will keep an eye out on any EVA operators while they are outside of life-preserving vehicles. PAM is an example of the many elements that can be integrated into OCTaVIA's architecture. This system will help monitoring operators in case astronauts should have trouble during a loss of communication. The help will come as a simple warning for minor issues, or automatically direct one astronaut to the location of the other in case of major problems.

PAM utilizes OpenPose[24], a system that analyzes videos and still images for posture. Preliminary tests show that the system can detect humans wearing street clothing, the xEMU, and the Orion Crew Survival System suit, producing the output reported in Figure 11.

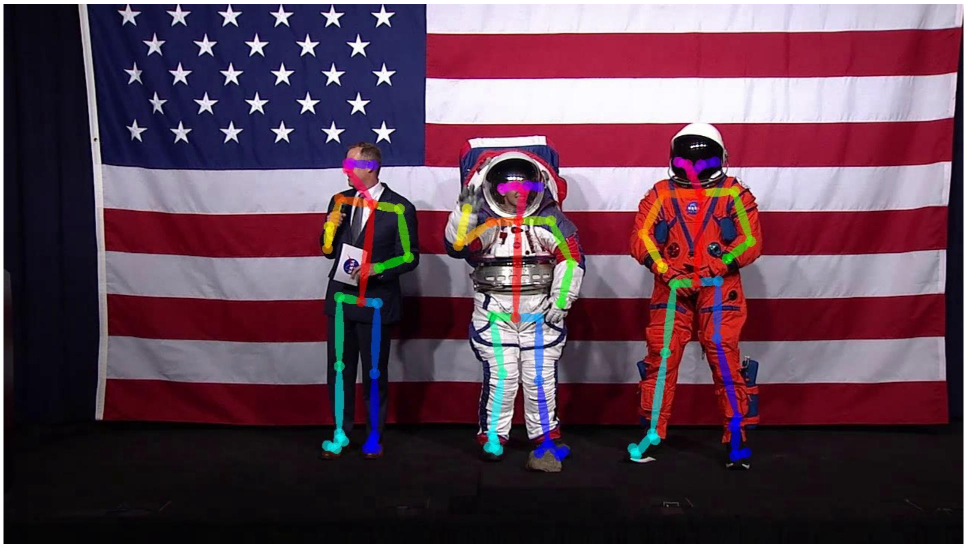

Figure 11: Output using the BODY-25 model in OpenPose, which detected humans wearing typical clothing (left), the xEMU (center), and the Orion Crew Survival System (right).

We wanted to assess whether the detection of a human wearing an xEMU was reliable enough, given that we utilized a pre-trained model available in OpenPose. For this reason, we took a short segment of footage from the unveiling of the xEMU composed of 1072 frames. This video segment shows people wearing regular clothing, the xEMU, and the Orion Crew Survival System. We only looked at the first two people, since the Orion Crew Survival System would not be utilized in an EVA. OpenPose offers three pre-trained models: MPI, COCO, and BODY-25[24].

Table 1: Summary of the actions and positions of a person wearing ordinary clothes.

Frame	Action	View	Keypoint Counts	Avg. Conf.
0-140	Arm motion while standing	Quarter right lateral / frontal	40 (23) 101 (24)	0.787
141-250	Rotation in place	Right lateral to frontal	43 (24) 67 (25)	0.805
251-1071	Standing in place with slight to moderate postural and arm movements	Frontal	1 (22) 28 (23) 450 (24) 342 (25)	0.801

The tests that we ran utilized the BODY-25 model, which is the most accurate and also the fastest. On this note, although OpenPose does not require specific hardware, the performance is significantly improved (approximately 6x to 10x faster) with GPU processing. To minimize latency in OCTaVIA, some of the nodes utilize Jetson Nano devices, which have a 128-core GPU built into them.

OpenPose returns a JSON (JavaScript Object Notation) file with the (x, y) coordinates of any keypoint it was able to detect as well as the level of confidence associated with each. In our case, each person could have at most 25 keypoints, and the confidence is in the range [0-1]. For this test, we looked at the action that each person was performing and we report the keypoint count as well as the confidence. In the case of a person wearing regular clothing, reported in Table 1, we can see that the system picks up the keypoints well. The system missed the most keypoints when the subject was not facing the camera (frames 0-140), but even then, it could detext at least 23 keypoints. As the subject turns (frames 141-250) and then eventually faces the camera (frames 251-1071) performing relatively slow motions, the system was able to pick up most keypoints with a high confidence.

We can utilize the results in Table 1 as control and compare them to the output of the system when it analyzed the subject wearing the xEMU, reported in Table 2. The actions were not the same, so the comparison cannot be considered a perfect match. However, the gross movements were approximately the same, expect for the walking (frames 0-65). We can observe that overall the confidence was lower, however the number of keypoints that were detected is high and comparable to the results from the control for the actions of rotation in place (frames 106-215 in Table Y) and frontal views with and without major motions of limbs. For this reason, we feel confident that we can utilize OpenPose with the default BODY-25 model as testbed for PAM.

Table 2: Summary of the actions and positions of a person wearing an xEMU.

Frame	Action	View	Keypoint Counts	Avg. Conf.
0-65	Walking	Left lateral	17 (22) 4 (23) 45 (24)	0.693
66-105	Arm motion while standing	Left lateral	2 (23) 38 (24)	0.675
106-215	Rotation in place	Left lateral to frontal	33 (24) 77 (25)	0.759
216-510	Standing in place with slight postural movements	Frontal	295 (25)	0.777
511-1071	Arm motion while standing with slight postural movements	Frontal	13 (24) 548 (25)	0.766

We intend to further leverage GPU systems by automating the recognition of actions based on the wireframe and keypoints[25]. This step will completely automate PAM by allowing it to detect each person's pose, but also their actions. Wireframe recognition may be utilized to detect distress poses, such as a person laying on the ground for more than a few seconds, or for pre-determined visual signals, such as extending limbs and assuming the shape of an X to signal the need for help, in case radio communications are down. Action recognition could enhance the system by detecting if certain positions are acceptable or not. For example, someone may kneel because they are not feeling well, or they may kneel because the mission requires them to pick up samples from the ground. Associating the automatic action recognition with the context of the mission will enhance the safety of the work site.

Future Works and Conclusions

The pilot tests of ARGOS and OCTaVIA have been very promising, however there are still many elements that need significant development. As the system is modular, we will first focus on the core elements of OCTaVIA. In particular, we will look at the network connectivity and its limitations, power usage of the nodes, and application load balancing. These elements will give a clear insight into how scalable is the system, and how much redundancy is necessary for the system to operate smoothly and without any delay to the EVA operator.

Next, we will focus on the different components that make up the current functionality, such as PAM. For this system, we are still learning about the abilities and limitations of the software that powers the keypoint detection. We are planning to perform a series of tests to evaluate its use with different image resolutions and object distances, and adding action detection abilities to automate the surveillance of the operator. Several other modules will be updated and added as necessary, since ARGOS is a framework that will still serve as the main infrastructure for student research associated with the NASA SUITS Design Challenge, as well as User Experience research in Augmented Reality through MAE.

The main limitation of this research group is associated with the hardware itself. We are working on reducing the physical footprint of the system as much as possible, but we do not have figures that specialize in hardware design and implementation. We are hopeful that building a rough profile of the processing capabilities of OCTaVIA as well as its power usage will give us enough information to reach out to other institutions that focus on hardware, and work on joining forces to create something that is portable and easy to deploy.

We believe that a fully distributed, flexible system will easily meet the needs of future EVAs in terms of availability, redundancy, and safety. This system has already proved its effectiveness in attracting students to STEM disciplines, contextualizing education, and inviting independent research. It is our sincere hope that its scientific and technical merits will also be evaluated and adapted into current and future developments of systems supporting EVAs and the Artemis program.

Acknowledgements

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

[1]Hansen, C., and Cassidy, C., "Mishap Investigation Board Summary of Extravehicular Activity 23: Lessons Learned from a Spacewalk Close Call," *Journal of Space Safety Engineering*, Vol. 1, No. 1, 2014, pp. 32-39.

[2]Alpert, B. K., and Johnson, B. J., "Extravehicular activity framework for exploration-2019," *Forty-Ninth International Conference on Environmental Systems*, ICES, 2019, pp. 1-30.

[3]Patterson, D. A., Gibson, G., and Katz, R. H., "A case for redundant arrays of inexpensive disks (RAID)," *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*, Vol. 1, ACM, 1988, pp. 109-116.

[4]Zussman, G., and Segall, A., "Energy efficient routing in ad hoc disaster recovery networks," *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 1, IEEE, 2003, pp. 682-691.

[5]Lien, Y.N., Jang, H.C., and Tsai, T.C., "A MANET based emergency communication and information system for catastrophic natural disasters," *Twenty-Ninth IEEE International Conference on Distributed Computing Systems Workshops*, Vol. 1, IEEE, 2009, pp. 412-417.

[6]Mahiddin, N. A., Sarkar, N. I., and Cusack, B., "Gateway load balancing and routing selection scheme of MANET in disaster scenario," *Second Asia-Pacific World Congress on Computer Science and Engineering*, Vol. 1, IEEE, 2015, pp. 1-7.

[7]Soto Medico, J. P., Gilbert-Wason, K. H., Hyland, W., Manlucu, J., Martinez, O., Paliashchuk, L., Ra, E., and Vincenti, G., "ARGOS: a platform for student engagement," *Twenty-First Annual Conference on Information Technology Education*, Vol. 1, ACM, 2020, pp. 298-298.

[8]Vincenti, G., "Engaging IT students through the NASA suits design challenge: an experience report," *Twentieth Annual SIG Conference on Information Technology Education*, Vol. 1, ACM, 2019, pp. 22-27.

[9]Vincenti, G., "Open challenges as a way to engage students: an experience report from three undergraduate courses," *Twenty-First Annual Conference on Information Technology Education*, Vol. 1, ACM, 2020, pp. 200-205.

[10]Ahsan, N., Andersen, M., Baldwin, P., Brown, J., Chapman-Weems, N., Hunt Estevez, C., Hyland, W., Leonard, B., Manlucu, J., Vandi, M., Yee, C., Vincenti, G., and Walsh, G., "An Augmented Reality Guidance and Operations System to Support the Artemis Program and Future EVAs," *Fiftieth International Conference on Environmental Systems*, ICES, 2021.

[11]Hunt Estevez, C., Jones, J., Shrestha, S., and Vincenti, G., "Serious Games in STEM: Online Collaborative Design of a Lunar Simulator," *International Conference on Human-Computer Interaction*, Vol. 1, Springer, 2021, pp. 223-235.

[12]Hunt Estevez, C., Jones, J., Shrestha, S., and Vincenti, G., "A Lunar Spacewalk Simulation to Support the Artemis Outreach Program and Promote Remote Public Engagement," *Fiftieth International Conference on Environmental Systems*, ICES, 2021.

[13]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, Interaction Design and Information Architecture, The University of Baltimore, Baltimore, MD, 2020.

[14]Crowther, N., "Remote Testing of AR HUDs for Lunar Exploration," M.Sc. Thesis, Interaction Design and Information Architecture, The University of Baltimore, Baltimore, MD, 2021.

[15]Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., ... and Suter, P., “Serverless computing: Current trends and open problems,” *Research Advances in Cloud Computing*, Springer, 2017, pp. 1-20.

[16]Cheetham, B., “Cislunar autonomous positioning system technology operations and navigation experiment (Capstone),” *ASCEND 2021*, AIAA, 2021, p. 4128.

[17]Liu, F., Jing L., Yuqing Y., Wang, W., Hu, D., Chen, P., and Niu, Q., “Survey on WiFi-based indoor positioning techniques,” *IET Communications*, Vol. 14, No. 9, 2020, pp. 1372-1383.

[18]Yang, C., and Shao, H. R., “WiFi-based indoor positioning,” *IEEE Communications Magazine*, Vol. 53, No. 3, 2015, pp. 150-157.

[19]Wang, Y., Yang, X., Zhao, Y., Liu, Y., and Cuthbert, L., “Bluetooth positioning using RSSI and triangulation methods.” *Tenth Consumer Communications and Networking Conference*, Vol. 1, IEEE, 2013, pp. 837-842.

[20]Park, H., Noh, J., and Cho, S., “Three-dimensional positioning system using Bluetooth low-energy beacons,” *International Journal of Distributed Sensor Networks*, Vol. 12, No. 10, 2016.

[21]Hallberg, J., Nilsson, M., and Synnes, K., “Positioning with bluetooth,” *Tenth International Conference on Telecommunications*, Vol. 2, IEEE, 2003, pp. 954-958.

[22]Escudero, C. J. (2005). In-building location using bluetooth.

[23]Tong, X., Ye, Z., Xu, Y., Gao, S., Xie, H., Du, Q., ... and Stilla, U., “Image registration with Fourier-based image correlation: A comprehensive review of developments and applications,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 12, No. 10, 2019, pp. 4062-4081.

[24]Cao, Z., Martinez, G. H., Simon, T., Wei, S. E., and Sheikh, Y. A., ”OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, No. 1, 2021, pp. 172-186.

[25]Yan, S., Xiong, Y., and Lin, D., “Spatial temporal graph convolutional networks for skeleton-based action recognition,” *Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI, 2018.