Lunar Exploration using Augmented Reality (LEAR)

David Brown, Sammy Lee, Teodor Malendyvch, and Yongsheng Xu

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

College of Optics and Photonics (CREOL), University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The NASA SUITS challenge of 2021 ask for the displays of the helmet to be brought into the modern era. The LEAR team choose to use the Hololens2 with an additional optical heartrate sensor. The integration of computer vision is used to create a larger contrast in images. A nearest point algorithm is used to help navigate astronauts from one point to another. Science sampling uses the Hololens2 holography library to create a non-offensive way to collect data and samples. The biometrics of the astronaut are displayed using similar technology along with the heart rate monitor as a secondary source of information. These inclusions bring the helmet display of the astronaut to the modern era.**

***Index Terms* — NASA SUITS challenge, CLAHE, Pulse Oximetry, UI Design**

I. Introduction

NASA's annual SUITS (Spacesuit User Interface Technology for Students) challenge was issued on August 31, 2020. In this iteration, the focus of the project was to bring the space helmet to the modern era using head mounted displays (HMD) for the upcoming Artemis Missions. The objectives of this challenge were divided into three categories, navigation, illumination, EVA system states and science sampling. The goals of the navigation objective were to navigate from an area to another selected point of interest and to return to the lunar lander. For illumination, it was asked that we consider a high contrast and low light level situation. For the EVA system states, the astronaut's biometrics should be available to the astronaut through the HMD and monitored for anomalies. In the science sampling, the teams were asked to be able to display instructions, interact with tools, locate the correct site, take pictures, and help collect samples. The LEAR team decided on using the HoloLens2 for our HMD. It was chosen based on its built-in processor and lack of additional peripherals. An external pulse oximeter was integrated to collect heart rate. A temperature sensor was also integrated externally to feed information into the HMD.

II. Requirements for NASA

The EVA task instructions shall be displayed. The astronaut must be able to always access the status of the spacesuit. The astronaut should be able to communicate with ground control at all times. A caution and warning system must be implemented to inform astronauts of spacesuit anomalies. An astronaut must be able to continue task on hand seamlessly in cases of interruption. The user interface shall not permanently impede the astronaut's ability to perform. All hand gestures must be operable with the EVA gloved hand, like a heavy ski glove. The user interface will take field notes for lunar sampling. The astronaut should know its location and how to navigate from one point to another at all times. For peripheral devices, the device shall communicate with the HMD. Any removable components on the peripherals must have a tethered attachment point. All tools must be operable with the EVA gloved hand. Peripheral devices must not have holes to trap fingers. There shall be no sharp edges. Pinch points should be minimized.

III. Hardware System

Mainly, our hardware contains 3 parts: microcontroller, wireless communication module, and oximeter, while the temperature sensor is embedded onto the wireless communication module. Simply, our temperature sensor reads data from the environment, proceed to the microcontroller, and microcontroller prints out temperature, and the process for oximeter is similar, it reads a voltage when our finger is put onto the IR LED, and process it to the microcontroller and we will get a heart rate after calculation, after these two process, the value of temperature and heart rate will be transferred to HoloLens by HC-05 Bluetooth module, then value will be print on the designed area.

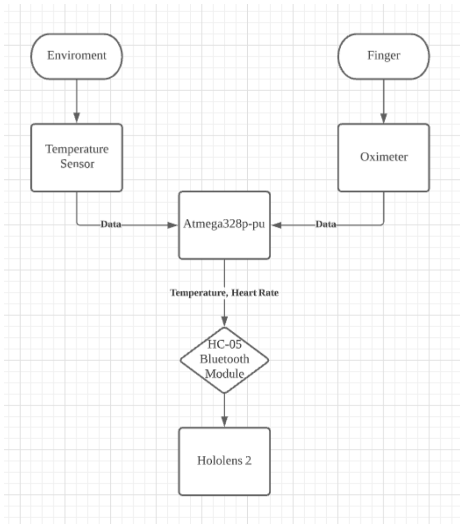


Figure 1: Hardware System Flowchart

Microcontroller is the heart of the project, after comparing MSP430, Raspberry Pi and Arduino, we decided to choose Arduino Uno as our reference MCU, the Arduino platform is a open-source, well-developed, and widely-used in many small projects, on the Arduino UNO, there is an Atmega328P-PU microcontroller, so we choose it as our microcontroller, it's a low-power CMOS 8-bit AVR microcontroller widely used in many projects. It has many digital communication peripherals: 1-UART, 2-SPI, 1-I2C, and several Capture/Compare/PWM Peripherals: 1 Input Capture, 1 CCP, 6 PWM. What's more, it's the same MCU used on Arduino UNO, after a process named "Burn the Bootloader", we can use Arduino IDE to program with it using C/C++, then upload the program intro the chip and use it like Arduino. Since there are a lot of projects based on Arduino, so we choose our sensors based on existing projects, it can help us overcome challenges.

MCU	Atmega328p-pu
Flush Memory Size	32 KB
CPU Speed	20 MHz
SRAM	2 KB
EEPROM	1 KB
Pin Count	32 Pin
I/O Pins	23
Temperature	-40 to 85 °C
Operating Voltage	1.8 to 5.5 V

Table 1: Atmega328p-pu Datasheet

A. Bluetooth

The HC-05 Bluetooth module is a well-developed 2.4GHz Bluetooth device follows the IEEE 802.15.1 standard for RF transmission, it supports for multiple communication protocols: UART, USB, SPI; operate between 4 to 6 volts, there are 6 pins on the module, 4 of them will be used: VCC, GND, TX, RX.

B. Temperature Sensor

A Dallas DS18S20 Temperature Sensor is built in the wireless communication PCB, it provides high accuracy (±0.5 °C) with fairly low cost, the operating voltage of the sensor is 3.0V to 5.0V, which works for the Arduino. In this project, we have multiple sensors to connected to Arduino, so this sensor has some advantages, it only requires one digital pin of the Arduino for communication which saves pin for the project, and has a unique 64-bit serial code, which can be connected to one Arduino pin in 1-Wire with other sensors and has no error code. It has a 3-pin connection, VCC, DQ, GND; pin DQ is for data output, to get the correct and stable temperature value, a 4.7 K ohm pull-up resistor is needed, which should be put parallel to the DQ and VCC. Since in this project, one-wire structure is applied, while coding, Dallas Temperature Library should be included.

C. Transceiver Module

There are two modules for RF transmit and receive, FS1000A and nRF24L01. FS1000A is a generic 433 MHz wireless RF module works at low transmission frequency, low power consumption, maximum transfer rate is 10 KB/S, can work between 3.5V to 12V. For nRF24L01, it is a 2.4 GHz Wireless IC, can transfer data at 2 MB/S with 1 MHz bandwidth, it operates between 1.9V to 3.6V while consuming low power.

D. Zigbee Wireless Communication Protocol

The XBee ZigBee wireless communication module can transmit data further than Bluetooth but at a lower rate. It follows the IEEE 802.15.4 standard for RF transmission at 2.4 GHz, it operates at 1.8V to 3.6V, and can transfer data at 500 KB/S maximal.

IV. PCB Design

A. Microcontroller PCB

In order to control the hardware system and connect to HoloLens, a microcontroller PCB is designed. On this PCB, an Atmega328p-pu is the heart, with 2 voltage regulator circuits built-in, LM7805 and LM317 for 5V and 3.3V respectively. In case to connect all pins and for future improvements, there are 32 pins in total with 6 analog pins built-in, so it can connect almost everything would use in a senior design project, for example, originally, the Bluetooth module should be connected to TX and RX pins, but it will be conflict to the oximeter, which is connected to analog pins, but there is another pins can be used, so we just change pins to transfer data. Also, an Atmega328p-pu processor is working on C/C++ environment, and its design for Arduino, to make the project easier, it's better to use Arduino IDE directly, so it's required to burn the bootloader on a brand new Atmega328p-pu before solder onto the PCB. After that, we can connect our microcontroller PCB with a USB to TTL converter with 5 pins: 3.3V, 5V, RX, TX, GND, then this module can use the Arduino IDE and all its libraries directly, and upload program into the microchip.

B. Wireless Communication PCB

For the wireless communication PCB in this project, we considered three ways to transfer data wirelessly, RF, Bluetooth and WIFI, so there are four wireless communication modules and one temperature sensor module. In this project, Bluetooth is tested to be the best way for data transfer to HoloLens. For the temperature sensor, there is a 4.7 K ohm pull-up resistor connected between Data pin and VCC pin, without it, the print out will be some weird number.

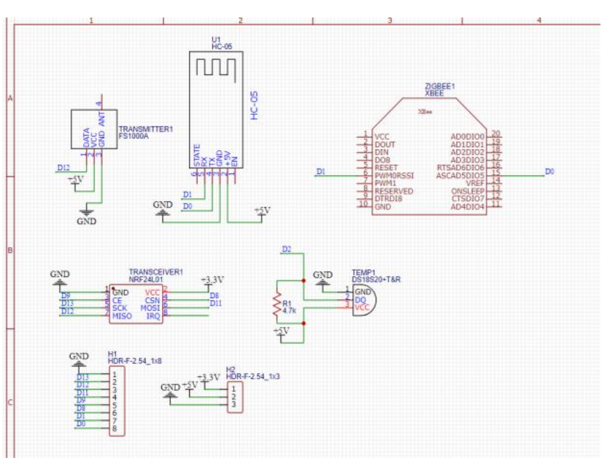


Figure 2: Wireless Communication PCB Schematic

V. Pulse Oximetry

A. Photoplethysmography

Wrist-mounted optical heart rate monitors utilize photoplethysmography to measure vital health information. It is a relatively low-cost and non-invasive method. This technique shines short-wave low-intensity infrared light through the surface of the skin. The light is absorbed by bones, veins, arteries, and skin tissue. [1]

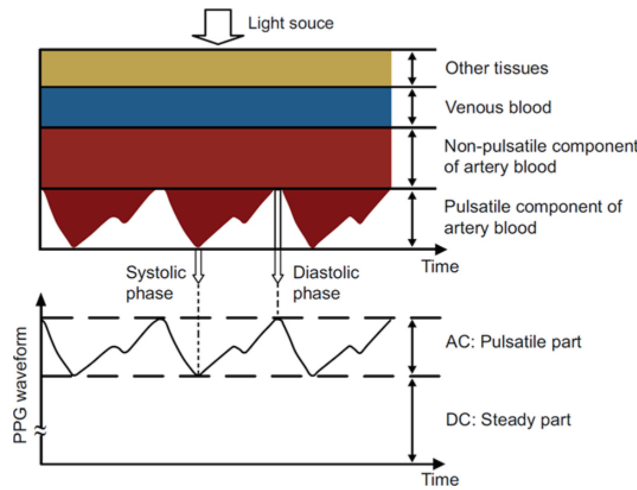


Figure 3: Useful diagram to demonstrate the AC and DC parts of a PPG signal.

Fig. 3 displays the process of photoplethysmography. The detected waveform has two parts. The DC steady state signal comes from the absorption of skin tissue as well as any other non-pulsating part of the epidermis. The AC part of the signal comes from the pulsating blood. By measuring the time difference between two crests of the AC signal, we can calculate the heart rate of the user.

Additionally, photoplethysmography can be utilized to measure the oxygen saturation (SPO2) of the blood. SPO2 is the ratio of oxygen-saturated versus total hemoglobin. The measurement of SPO2, also known as pulse oximetry, is dependent on the absorption spectrum of oxyhemoglobin (HbO2) and deoxyhemoglobin (Hb) in the visible and infrared spectral regions. Utilizing two LED sources in the sensitive 800-1000 range and one in the less sensitive 600-700 nm, two photoplethysmography signals can be measured as displayed in Fig 4. [2]

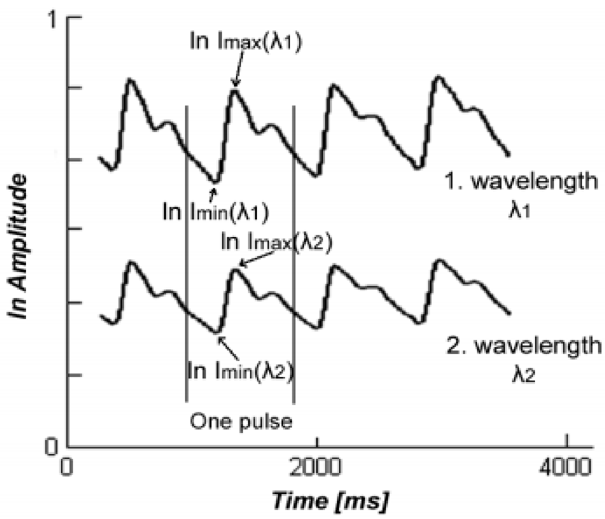


Figure 4: Reflected signal of two wavelengths measured by the photodiode.

From the max and min of both wavelength signals, we can calculate the R rate:

(1)

$$R = \frac{\ln \frac{I_{max}(\lambda_1)}{I_{min}(\lambda_1)}}{\ln \frac{I_{max}(\lambda_2)}{I_{min}(\lambda_2)}} \qquad (1)$$

Deduced from the Lambert-Beer Law, this equation along with experimental results can yield a calibration curve.

B. Design

In the design of the pulse oximeter, a few considerations were made.

To differentiate signals, the red LED and IR LED cannot be turned on at the same time. A switching input was designed to determine which of the LEDs would turn on. The MCU from which the switching input comes would then distinguish which LED is on and identify the signal as such.

The signal from the photodiode is processed through a passive high-pass filter with a cutoff frequency of 0.7 Hz, and through an active low-pass filter/non-inverting amplifier with a cutoff frequency of 33.8 Hz and a gain of ~8. These elements clean and amplify the signal, which is sent to be analyzed by the MCU.

The placement and orientation of the LEDs with relation to the photodiode was paramount in the acquisition in the signal. From experimentation, the ideal distance between each LED and the photodiode was determined to be ~7 mm. Due to the nature of the pulse oximeter, that distance is dependent on the person, how they will be wearing the wrist-mounted monitor, and any actions they will be performing such as standing still, walking, running, etc.

C. Experimentation

With the design presented, arterial blood signals were measured using an oscilloscope and displayed in Fig 5.

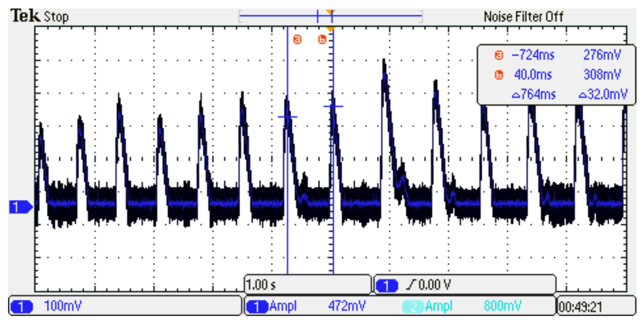


Figure 5: Measured PPG signal.

To determine the accuracy of our design, an off-the-shelf oximeter was utilized. Four waveforms were averaged to determine the heart rate in beats per minute and then compared to values from the standard oximeter. The results can be shown in Table 2.

Heart Rate Values	Sample Size	Average Error
70-75	6	2.63
76-80	31	2.58
81-85	19	4.55
86-90	26	2.27
91-95	18	2.49
96-100	2	4.05
101-105	1	4.76

Table 2: Heart Rate versus Average Error.

From the data, there does not seem to be a reasonable correlation between heart rate values and accuracy. The resolution of heart rate pulse detection is ~30 milliseconds.

Measuring the oxygen saturation requires calibration with the off-the shelf oximeter. The R value was calculated from alternating the IR and red LEDs and was mapped to the corresponding SpO_2 value. The results can be seen in Fig 6. Unfortunately, our calibration was limited due to the limited number of human subjects, which can be seen in the narrow scope of the calibration.

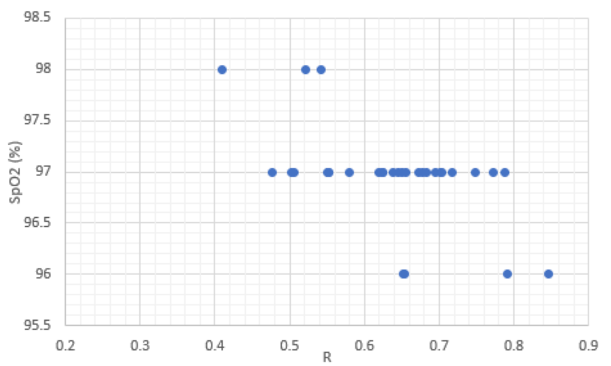


Figure 6: Calibration Curve. It is very limited by our subject pool. As a result, the SpO2 measurements are not as precise.

A modern pulse oximeter measures the SpO_2 with less than 2% error. Worse accuracy is also accepted by clinical staff due to the 4% dispersion of the blood oxygen saturation level between healthy human subjects. The oxygen saturation measurement (SpO_2) was compared against an off-the-shelf pulse oximeter. With a trial of 25 measurements, the error rate was found to be 1.98%. The SpO_2 readings were typically lower than the off-the-shelf oximeter. That is probably due to our incomplete calibration.

VI. HOLOLENS UI/SOFTWARE

A. Design Description

A flowchart detailing the overall structure underpinning the software of our proposed SUITS User Interface (UI) located in Figure 7. There are six total applications, each responsible for fulfilling a primary objective of the SUITS Design Challenge. Differentiation of each application is based on the color-coded legend. Application interfaces between two different applications are represented by two rhombus shaped boxes containing the communicating applications. Further detail on the logical structure of the Sampling, EVA State, and Illumination Applications can be found, in detail, in following sections of this proposal.

Also pictured is the communication protocol used to send data from our external sensors, which collect and send data in real-time through a microprocessing unit. The microprocessing unit then communicates wireless using bluetooth, sending the collected data to the HoloLens. This is represented in the diagram by the bluetooth symbol and the hardware/software interface. The data is collected by the HoloLens through a bluetooth socket, processed into a decipherable format, and fed into the application to allow the external environment to control the flow of information through the application.

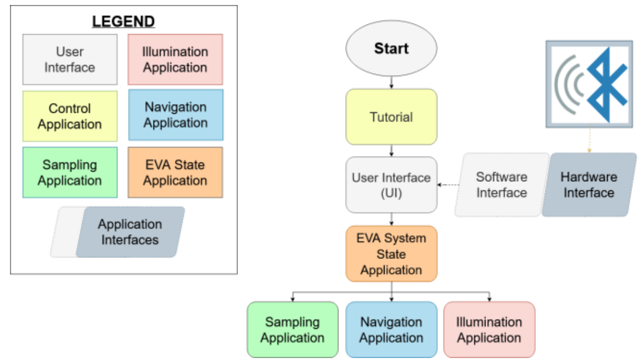


Figure 7: Software Flowchart Diagram

To accomplish the outlined objectives the software was divided into two parts: a backend and frontend. The backend is responsible for reading and writing the application data at runtime. Our backend codebase was written using a combination of C++ and C#, built in Visual Studio 2019 to run on ARM64, which outputs our proposed applications both Dynamic Link Libraries (DLL) and Windows Metadata (WinMD) files.

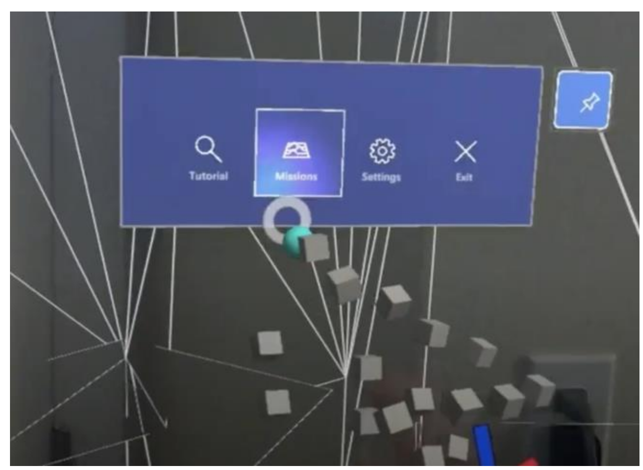


Figure 8: Opening Menu

The backend libraries (DLL and WinMD files) are used by the frontend to initialize, start, and obtain data at runtime. We used the cross-platform game engine Unity to build our frontend User Interface (UI) which, as outlined in our UI design, responds to the user by calling the backend libraries to process data and update the UI. The UI contains a set of holographic objects, referred to as “Game Objects” in Unity. We used the Microsoft Mixed Reality Toolkit (MRTK), an additional library which can be easily imported into Unity, to create each holographic game object.

When the UI is initialized on the HoloLens, the Start() function is called by the UI menu Game Object, which calls the Mission Control class from the backend libraries, passes it to the Mission Control Game Object at “home coordinates”, and displays the first menu for the user which contains four options (Tutorial, Missions, Settings, Exit). The Mission Control Game Object is used by the Unity UI to poll the backend for data.

For instance, on initialization, the Mission Control Game Object loads a JSON file containing details about the user’s mission (mission description, target coordinates, tasks, instructions, maps, etc.) into the “Mission Notebook” Game Object. When the user clicks “Mission Briefing” or “Start Mission” from the second UI menu, the Mission Notebook Game Object is displayed containing all the mission details nested within individual tabs on the notebook.

B. EVA system state

The EVA System State application (Fig. 7, orange blocks) always runs and monitors both astronaut and suit vitals (heart rate, suit oxygen levels, cooling water, etc.) in real-time to maintain astronaut safety. These measurements are displayed on the UI in an unobtrusive manner to avoid restricting the astronaut’s field of vision or destroying astronaut concentration. If an abnormality is measured, the astronaut will be warned on the UI. If the abnormality does not correct itself or is compromising the health of the astronaut, the astronaut will be warned to abandon the mission and return to the lander.

When the astronaut marks coordinates for a mission, the application interfaces with the Sampling/Navigation applications to calculate approximate distance, travel time, and whether current suit vitals (oxygen, etc.) will last the round-trip. The astronaut will be warned if the mission would be considered dangerous or taxing on resources.

C. Navigation

Astronauts are guided to marked locations of interest using the Navigation application (Fig. 7, blue). The application will download NASA Digital Terrain Models for the 2km - 10km range surrounding the coordinates. The “shortest path” will be calculated using GPS heightmaps and guide user based on starting location.

The distance and approximate travel time are calculated along with a directional path from the astronaut's current location to the marked location, which will display on the UI as an overhead directional arrow for easy guidance. Terrain models are used to render a 2D topological map of the terrain surrounding the marked location. These maps can be referenced by the astronaut and viewed within the HoloLens 2 visual field.

When the astronaut starts their mission, a virtual "tether" will attach the astronaut to their starting coordinates, keeping track of the astronaut's directional movement through the terrain. This will act as a way for the astronaut to record discrepancies between the map renderings virtually compared to the actual terrain, a source of field notes for recording actual paths around the terrain, and an easy way for the astronaut to navigate back to the Lander.

D. Geological Sampling

Missions to geological sites of interest are done using the Sampling application (Fig 7., green). Information pertaining to each individual site such as coordinates, tools to use, list of instructions for mission, field notes, and goals are downloaded from mission control and listed within the UI via JSON format.

Once the astronaut has been briefed on the mission, they will precede to the geological site. The distance to the location, approximate time of travel, and rendered 2D maps are automatically displayed through an interface between the Sampling and Navigation applications.

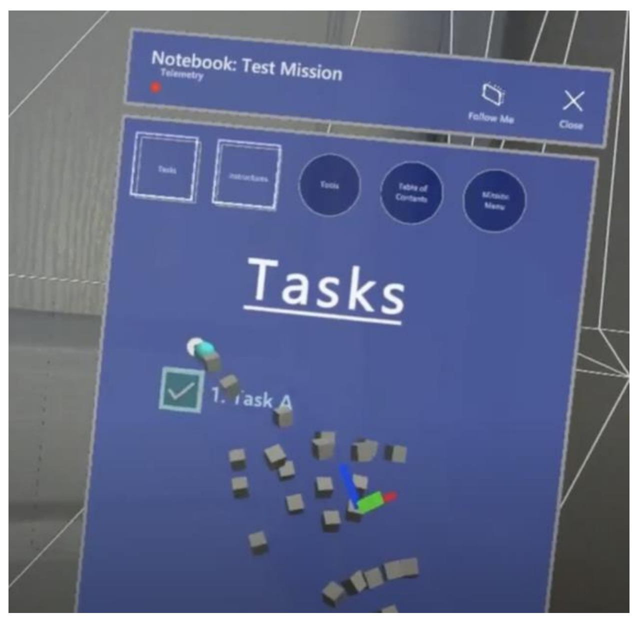


Figure 9: Task Assistance Template

VII. ILLUMINATION

Initial consideration on low level situation were solved through hardware. The team initially thought of using a system like Night Vision Goggles or LIDAR. These solutions were dismissed due to Bulk and adjustability in the former and cost of the latter. The team then chose to use Image and Video Processing through the built-in camera of the HoloLens2 to give more information to the astronaut. The initial tests were created in python using the OpenCV library. These initial tests were run using a high-definition image of at least 2k resolution. This image was then processed and recorded. The processing technique used included, gamma transforms, log transforms, thresholding, and histogram techniques. The team chose not to use more advanced recreation techniques or back propagation techniques based on processing times and processing power requirements. From inspection we concluded that histogram techniques were the best suited for an overall enhancement in contrast.

Figure 10: Example of base image and a histogram equalized image.

The main histogram technique used is Contrast Limited Histogram Equalization (CLAHE). This technique involves using a clipping limit value, which is based on a cumulative distribution function. It equalizes the histogram an area of values based on the selected grid size. The histogram Equalization technique uses a distribution of all the pixel values and tries to equalize them such that no value appears at a dominating amount. [3] We can see how the technique works when we compare the histogram of the base image and the histogram of the equalized image. In Fig. 11 we can see that there is a large collection of values in the lower end from the value 0 to 25. This gives us the dark image we see in the Fig.1 as the base. When we equalize the histogram the image values shift from having close to 4000 instances of a single value in the lower range to having around 1400 instances of a single value. The higher end of the range represents the white in the image. This can be seen in the histogram equalized image in Fig.12.

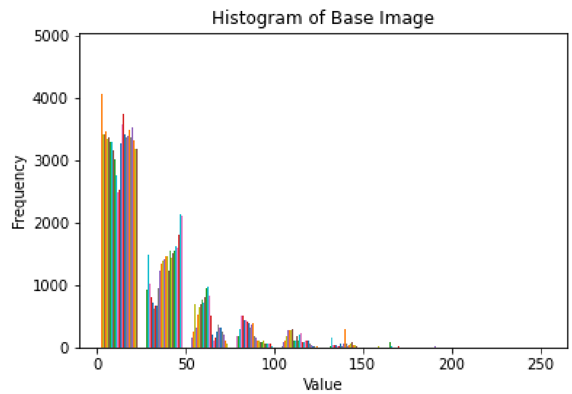


Figure 11: Histogram of the base image

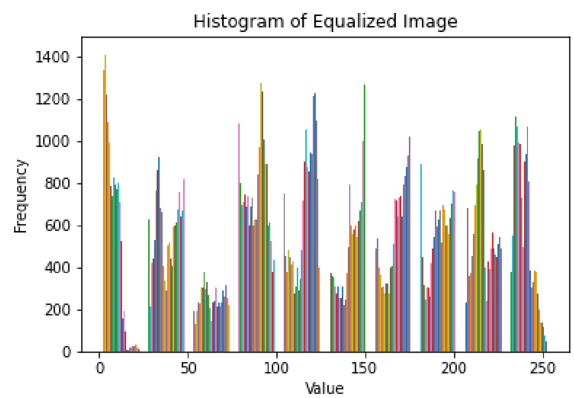


Figure 12: Histogram of a histogram equalized image.

When using OpenCV in python, the frames were not properly displaying. This could be caused by a lack of graphical user interface. The processing was then moved to using C# due to native compatibility with the HoloLens environment. In the C# environment video processing was then tested using a .Net wrapper of the OpenCV library called Emgu CV. For video, the main objective is to show a processed video to give the astronaut more information of its surroundings, like rear view camera for a car. The main requirements for the video processing being the processed video is at least 24 to 30 frames per second (fps). The fps requirement is a standard for film and television. It was chosen such that the astronaut does not need much time to adjust to it. CLAHE was then implemented into a C# program and the processing time was measured.

Trial	Measure Time (ms)
1	2.20
2	2.08
3	2.94
4	2.16
5	2.10
6	1.83
7	2.57

Table 3: Measured Processing Time of CLAHE

For 30 fps, the processing time of the algorithm must be less than 33 ms. In our measurements, the highest processing time occurred when the program was changing parameters and first applying the algorithm at 3.11 ms. The fastest time was around 1.82 ms. These runtimes tell us that the restriction on fps is based on the capturing device.

VII. CONCLUSION

All of the aforementioned material in this paper describes our attempt in satisfying the requirements of the NASA SUITS Challenge. While some requirements have not been met fully, we believe that this project was valuable not only to NASA, but to us as well.

ACKNOWLEDGEMENT

The authors wish to acknowledge the assistance and support of Rafaela Frota, who served as a consultant on this project.

Additionally, we extend thanks to the NASA SUITS team for putting on this competition and providing us with a Hololens II loan. Specifically, we offer our gratitude to Angelica Garcia, our mentor, and Dr. David Hagan, our faculty advisor for their expertise and guidance.

We are grateful to be sponsored by the Florida Space Grant Consortium and George Jackson Foundation, which has provided most of our funding and support.

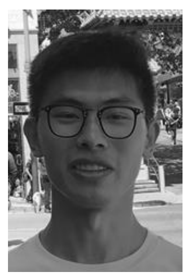
We offer our thanks and gratitude to the professors who have so kindly agreed to review our project.

BIOGRAPHY

Teodor Malendevych will receive his Bachelor’s of Photonic Science and Engineering from CREOL in May of 2021. Teodor is continuing his graduate studies with the IMLEX Master’s program in September 2021.

David Brown is graduating with a Bachelor’s in Computer Engineering in May of 2021. David’s career goals lie in bioinformatics and bridging the gap between computers and biology and he is pursuing jobs in those fields.

Sammy Lee plans to graduate with a Bachelor’s of Science in Photonic Science and Engineering in May of 2021. His current interest lies in computer vision and machine learning and plans to pursue those interest through industry or education.



Yongsheng Xu is a 23-year-old senior in Electrical Engineering at the University of Central Florida and will be receiving his Bachelors of Science in Electrical Engineering in May 2021. He is interested in design electronics that can make life easier.

REFERENCES

1. Cheriyedath, S. (2019, February 27). Photoplethysmography (PPG). Retrieved November 13, 2020, from https://www.news-medical.net/health/Photoplethysmography-(PPG).aspx
2. Stubán, N., & Masatsugu, N. (2008). Non-invasive calibration method for pulse oximeters. Periodica Polytechnica Electrical Engineering, 52(1-2), 91. doi:10.3311/pp.ee.2008-1-2.11.
3. K. Zuiderveld: Contrast Limited Adaptive Histogram Equalization. In: P. Heckbert: Graphics Gems IV, Academic Press 1994, ISBN 0-12-336155-9