Maya Archaeology 2017

<u>Engineering Team</u>

Quentin Gautier
Eric Lo
Nikko Bouck
Connie Du
Kenneth Criddle
Andres Gomez
Danbing Zhu
Shashank Uppoor
Tawfic Rabbani

Abstract

Current archaeological site scanning techniques (SFM and LIDAR) are extremely slow to setup and implement. The technology that will revolutionize their 3D modeling projects is here, however, a practical solution has yet to be developed. This project aims to resolve the limitations of previous scanning tools by compartmentalizing hardware in a sleek, portable, and unified design. Our product creates a lightweight and mobile solution that streamlines the data capturing process and facilitates real time data validation to model these sites. The system implements a dramatically redesigned form factor, allowing a single user to accomplish what was once a two-person job. This individual is able to acquire data using cutting edge RGBD scanning technologies processed through customized SLAM algorithms.

Introduction

The Maya were once a thriving civilization in the late 8th century. They made incredible innovations in irrigation, agriculture, architecture, and horology. In the early 9th century, the Maya suddenly vanished. Their populations died out and their culture all but disappeared. There are many theories about their decline: drought that lead to starvation, increased warfare among the city-states that lead to mass genocide, and an enlargement of the ruling class that demanded extravagant temples causing unbalance and conflict with the dwindling working class. The secret to their demise is hidden in the archaeological ruins that they left behind and each day, we lose more and more of that valuable history. Environmental conditions coupled with looting and damage by local populations are causing the irreparable degradation of these sites. If something is not done to facilitate the preservation of these lost Maya sites, we will forever lose the secrets that these historical treasure troves hold for us.

Our solution to ensure the conservation of these Maya archaeological sites is to digitally archive them through the use of 3D scanning technologies. Previous iterations of this solution employed antiquated technologies such as LIDAR and SFM that were incredibly time inefficient and a hassle to implement. The LIDAR (Light Detection and Ranging) system used a laser emitting device mounted on top of a tripod. The tripod had to be set up just right so a lengthy scan could be run, and this process was repeated every couple of feet. This was a cumbersome task and incurred a lot of overhead due to setup and teardown at each scan point. Similarly SFM (Structure from Motion) required lighting to be hung throughout the cave so a team of photographers to take hundreds of images of the same object from different angles to be stitched together later by a special algorithm. The main issue, other than the incredible time consumption, was the lack of real-time validation of the scans these scientists were collecting. It was impossible to know if the LIDAR or SFM scan result that was just gathered of the area was complete or not until much later when they returned to the labs and ran algorithms that had days to process the accumulated data. A new solution needed to be developed that solved these issues, and that's where Maya Archaeology 2017 comes in.

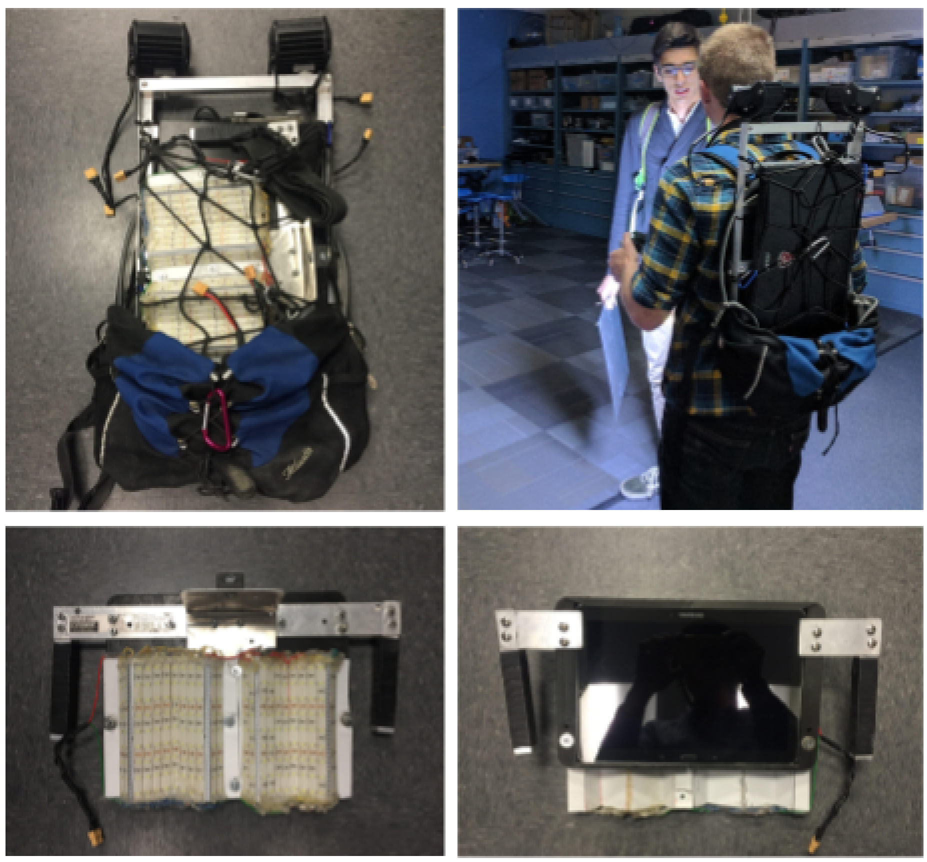
Given the clear obstacles in these past implementations of 3D mapping and scanning systems, this iteration of the Maya project focused mainly on scan efficiency. This was achieved by creating portable devices with real time feedback and onboard lighting systems with modularity in mind. The form factor of the scanning devices were designed to be operated by one person in order to reduce the amount of scientists needed to survey one region of a cave. This improves overall scanning time as a team of other scientists can explore and survey other areas of a particular archaeological site. Real time validation allows for scientists to evaluate if the current scan is complete. This reduces the amount of trips that need to be made back to a site in the case that a site doesn’t render properly back in the lab. It also allows an amature scanner to obtain detailed models that would otherwise require an advanced LIDAR technician or a professional SMF specialist. Finally onboard lighting reduces inconsistencies of lighting fixtures due to casting of shadows. This improves scan quality and speeds up setup time as the fixtures don't need to be set up and torn down for each scan. Outlined below is a detailed explanation of the steps taken to implement the system and the challenges that were solved to obtain these improvements.

Current Progress

We've built the prototype of a portable 3D scanning system that provides realtime feedback. It consists of a laptop, a backpack which carries the laptop and batteries used to power the whole system, and a tablet connected to a backpack mounted laptop. RGB-D camera sensors and an LED panel were mounted to the tablet for mobility and stable lighting purposes. The system is highly modularized. For example, user can easily choose a different algorithm to run as well as swap the laptop or the RGB-D camera with simple tools in the field. Currently this system is deployed in Maya archaeological sites in Guatemala.

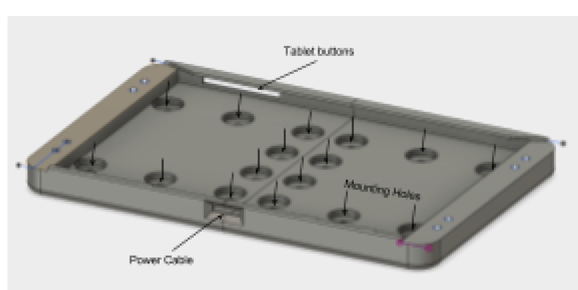
Backpack and Handheld Device

In designing the casing and mounting system for this device, we opted for a lightweight and versatile design. Below you can see the design schematics for the printed tablet mounts. The system was then assembled and scanned with as pictured to the right.



Tablet System: (top left) the system packaged up for quick transportation, (top right) tablet operated in SeaLab, (bottom left) rear of tablet with LED array and camera mounting system (bottom right) front of tabler with grip handles and lighting power cable

The case is designed to latch in the tablet by allocating it into place on the top facing side. This side is supported with an overhanging hinge that keeps the tablet from falling out of place. Additionally the top side has button access holes in order to easily operate the tablet and avoid having to unlatch it to power it on. The bottom facing side includes a hole for charging the device itself. On the back of the design you will note a lot of widely drilled holes. These are designed to support and contain the screws used to mount the cameras and handles to the back of the tablet case as well as attach the onboard lighting system.



Tablet Mount Model: featuring mounting holes, and ports for facilitating tablet use

Laptops and Tablet

We have three laptops available for our system. They are MSI, Dell XPS15 and Thinkpad W530. The Thinkpad W530 is used as a backup since it is pretty old and the hardwares are not as good as the other two. The table below summarizes the major specifications of the laptops.

Laptop	CPU	GPU	Memory
MSI	Intel i7	GTX 1060	16G
Dell XPS15	Intel i7	GTX 1050	16G
Thinkpad W530	Intel i7	Quadro K1000M	4G

All the laptops are running Ubuntu 16.04.

A Samsung tablet is used to connect to laptop in the backpack via VNC and remotely access the desktop.

Cameras

The system is compatible with all the RGB-D cameras we have now, including Kinect v1, Kinect v2, RealSense R200, RealSense SR300, RealSense ZR300.

SLAM Algorithms

SLAM (Simultaneous Localization and Mapping) is the computational problem of building map of the unknown environment while simultaneously keeping track of the agents (video camera) location with it. we want to shorten the time needed to scan a site by introducing SLAM algorithms. Aside from getting a 3D map directly from the SLAM algorithms, we can also use them to assist the data collection process of the technique called structure from motion (SFM). If we can take pictures for SLAM and SFM at the same time, SLAM can give user a rough idea on how good the already taken pictures can cover the target area. If the real-time feedback from the SLAM algorithm indicates that some parts of the reconstruction are not patched up, it's very likely that the same set of pictures will also leave a hole on the model generated by SFM. So the user should go back and take more pictures of that part.

We installed and tested 4 SLAM algorithms on our system. They are ElasticFusion, RTABMAP, RGBDSLAM v2 and ORB-SLAM2. All of them make use of RGB-D cameras. Except ElasticFusion, the other three are working with Robot Operating System (ROS). Using ROS makes it easier to share a data source among different processes. For example, the camera can publish the raw image data, then both a SLAM algorithm and a video recorder can subscribe the image data. In that way, we can have real-time feedback as well as the raw video. It will also be easier for future development to add in more features, like feeding IMU data to SLAM algorithm and make it more accurate.

Among the algorithms, ElasticFusion gets our special attention because it has almost all the desired features we want and yet is never tested in the field. RTABMAP and RGBDSLAM have already been tested by previous projects, but never on this hardware configuration, so they are also worth trying. We will discuss the algorithms individually in the following subsections.

ElasticFusion

ElasticFusion can construct dense surfel-based maps of room scale environments explored using an RGB-D camera in an incremental fashion without pose graph optimization or post processing steps. Surfels, or Surface Elements, are a powerful paradigm to efficiently render complex geometric objects at interactive frame rates. They comprise of depth, texture, color, normal information etc.

The features of the algorithm, which makes it suitable for our project, are as follows

- Performs better than existing dense SLAM algorithms when sensor makes movements of extended durations.
- When compared with algorithms that use pose graphs, ElasticFusion performs better when subjected to sporadic camera trajectories (camera paths that criss-cross / loop back on themselves), that is typical if a non-expert with a hand-held depth camera were to scan the environment.
- Emphasis is on the accuracy of reconstructed surface maps rather than camera trajectory tracking.

Results from literature

- Trajectory estimation performance is on par with or better than existing approaches.

System	kt0	kt1	kt2	kt3
DVO SLAM	0.104m	0.029m	0.191m	0.152m
RGB-D SLAM	0.026m	0.008m	0.018m	0.433m
MRSMap	0.204m	0.228m	0.189m	1.090m
Kintinuous	0.072m	**0.005m**	**0.010m**	0.355m
Frame-to-model	0.497m	0.009m	0.020m	0.243m
ElasticFusion	**0.009m**	0.009m	0.014m	**0.106m**

Table 1: Comparison of ATE RMSE on the evaluated synthetic datasets of Handa et al.

- Surface reconstruction results are superior to all other systems.

System	kt0	kt1	kt2	kt3
DVO SLAM	0.032m	0.061m	0.119m	0.053m
RGB-D SLAM	0.044m	0.032m	0.031m	0.167m
MRSMap	0.061m	0.140m	0.098m	0.248m
Kintinuous	0.011m	0.008m	0.009m	0.150m
Frame-to-model	0.098m	**0.007m**	0.011m	0.107m
ElasticFusion	**0.007m**	**0.007m**	**0.008m**	**0.028m**

Table 2: Comparison of surface reconstruction accuracy results on the evaluated synthetic datasets of Handa et al.

Note: Measurements are in ATE RMSE i.e. Root-mean-square of the Euclidean distances between all estimated camera poses and the ground truth poses.

- Execution time of the system increases with the number of surfels in the map (Average = 31ms per frame. Worst case = 45ms per frame).

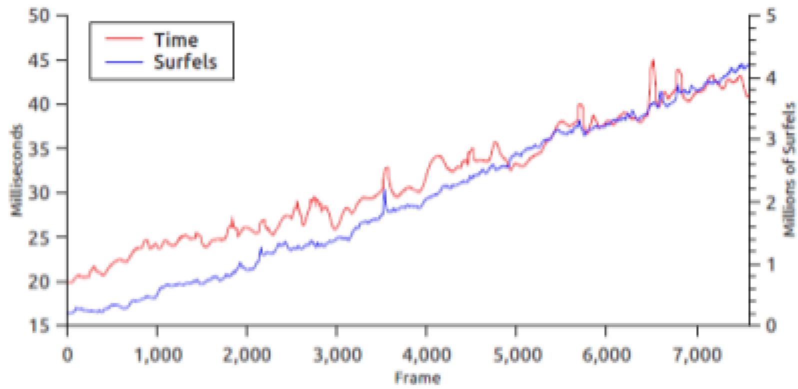


Figure 1: Frame time vs. number of surfels on the Hotel dataset.

For our experiments, we have used the open source version of the software obtained from ElasticFusion[1][2].

The implementation comes with a GUI, which you can see in Figure 2. The GUI makes it fairly easy to use the program. You can adjust the parameters, chose different drawing options, save the captured data and etc. And it gives the user real-time feedback of the meshed reconstruction. These all meet the requirements of this specific project.
The algorithm can do live capture/reconstruction as well as post-process of an existing dataset. The stored data can be viewed by Meshlab.

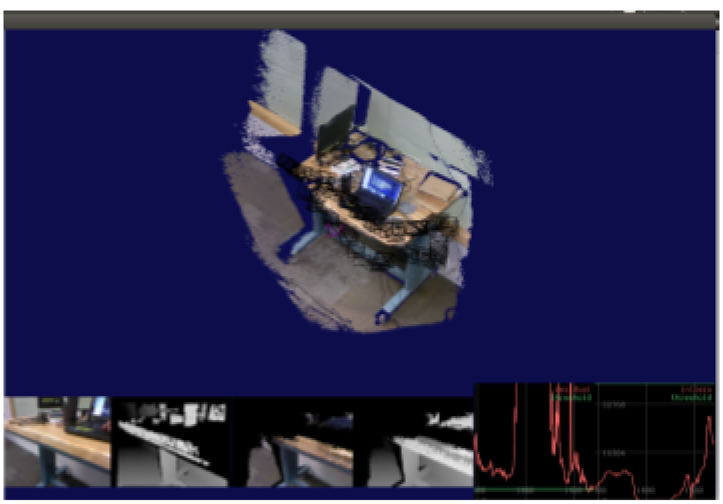


Figure 2: 3D scan of a table in SeaLab

This algorithm works with all the RGB-D cameras we have. Based on the experiments we have ran up to now, Kinect v1 has the best performance and a sample live scan of a table in SeaLab is shown to the left.

In conclusion, the capability of this algorithm can satisfy the requirements of this project. The drawback is its high demand of computational power/hardware. If we are going to use the Jetson board, then this algorithm may not be feasible. The first thought is although TX2 has a powerful GPU, the CPU could potentially be the bottleneck, because it has a max clock frequency of 2Ghz on 4 cpus where the more powerful i7 processors on the laptops we are using have a max clock frequency of 3.8GHZ . This cpu speed is important for some slam algorithms but not all. Another concern is TX2 uses ARM architecture. Thus, the installation might be different on Jetson, which may cause extra headache problems.

ORBSLAM2

This algorithm can be used for monocular, stereo and RGB-D cameras. It is able to compute in real-time the camera trajectory and a sparse 3D reconstruction of the scene in a wide variety of environments, ranging from small hand-held sequences of a desk to a car driven around several city blocks. It is able to close large loops and perform global relocalisation in real-time and from wide baselines. It includes an automatic and robust initialization from planar and non-planar scenes.

For our experiments, we have used the open source version of the software obtained from ORB_SLAM2[3].
The feature of this algorithm is: it tries to maintain as less points as possible and generates a very sparse point cloud. That make this algorithm more robust and faster than rest of the algorithms. The drawback is also obvious, it very hard to recognize the scene from the sparse point cloud. So it will cause trouble when we are trying to get real-time feedback of the reconstruction.

RGBDSLAMv2

This algorithm enables acquisition of colored 3D models of objects and indoor scenes with a handheld Kinect-style camera. The SLAM front-end is based on visual features ,such as SURF or SIFT, to match pairs of acquired images, and uses RANSAC to estimate the 3D transformation between them. The resulting camera pose graph is then optimized with the SLAM back-end HOG-Man.

For our experiments, we have used the open source version of the software obtained from RBGDSLAM[4].
This algorithm returns a dense point cloud to represent the reconstructed scene. In some sense, it is a combination of ORB-SLAM2 and ElasticFusion. It's not doing the online mesh so it is faster than ElasticFusion, yet it keeps enough points to give user a decent visual feedback.

RTAB-MAP / RTAB-Map_ros

RTAB-Map (Real-Time Appearance-Based Mapping) is an RGB-D Slam approach based on global loop detection. It can be used as both it's standalone version or can be run within ROS. RTAB-Map can be used to generate 3D point clouds of the environment and/or to create a 2D occupancy grid map for navigation.

For our experiments, we have used the open source version of the software obtained from RTAB-Map .
The performance of this algorithm is very similar to the one of RGBDSLAMv2.

Anza Borrego Expedition

Anza Borrego power system testing and data collection.

Cameras on Elastic Fusion:
The Elastic Fusion SLAM Algorithm had issues after 15m of scanning and would lose track of it's position, with all cameras when moving quickly. This would resulted in a mapping that was distorted and unusable. However when moving slowly Quentin was able to get a good results with the Kinect and ZR300.

The SR200 was very stable on Elastic Fusion with the device mounted lighting however this camera is only good at no more the 30 cm.

Conclusion:
We need to test more SLAM Algorithms and fine tune the thresholds for scanning the environment. We may also want to switch to a warm light spectrum and increase the lumen output mounted on the device.

Power System Results

Battery Life Analysis:
The laptops were able to last between 45 and 60 minutes while scanning on a full battery charge. The light system had ample power available however if we are able to extend the laptop life with external batteries we may need more batteries for the light system.

Lighting Review:
The Lights mounted on the device seemed help the algorithm stabilize.
The backpack mounted lights provided little benefit to the algorithm and may cause issues with the RGB due to the difference in light temperature.

Visual Feedback:
The wifi remote desktop over VNC was able to provide good feedback and the tablet had ample battery life. (Aprx. 8 hrs)

	Note: The VNC was unable to support full resolution. **Conclusion:** We need to increase the laptop operating life through an external power supply mounted in the backpack power bay.

Next Gen Model

For our next generation model we began designing a powerful, yet lightweight system. After some searching we came across the Nvidia Jetson TX2 which we believe will better solve many of our issues. Due to it's fast GPU we can improve render speeds with SLAM algorithms that are heavily dependent on GPU rather than CPU. The availability of the PCIE port allows for all raw data to be stored on an external source with a write speed of approximately 800MB/s. As for size, because of the dimensions it will be significantly smaller and lighter reducing the physical load the archaeologist must carry. as well as providing real time feedback. Another issue we wanted to address was power consumption. Our current system requires anywhere from 60W to 200W. Nvidia's tests state that the TX2 requires, at it's peak performance level, requires only 7.5W. This reduces the number of extra batteries that need to be carried, which further reduces the physical load.

Components List

Due to the high amount of data that the R200, ZR300, and SR300 RealSense cameras require, our system needs to be able to process large amounts of raw data and place it in persistent storage. As such, a finalized components list has been created looking at storage speed rates, computability, size, and weight as main critical points.

These components were initially decided upon due to their high performance rate, compatibility, and portability and thus were a good starting point for building the baseline capturing system.

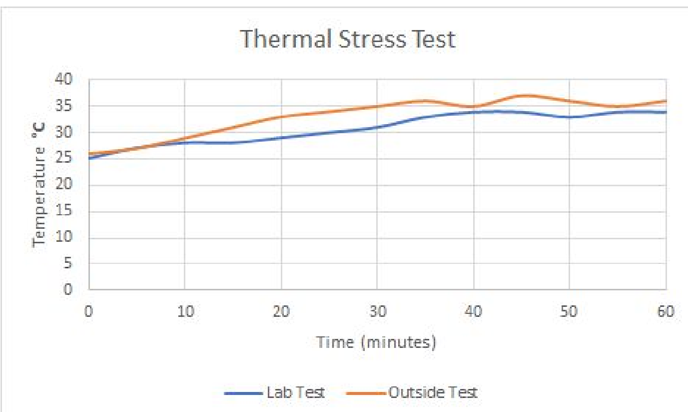
- Intel RealSense R200 RGBD Camera
- Intel RealSense SR300 RGBD Camera
- Intel RealSense ZR300 RGBD Camera
- NVIDIA Jetson TX2 Module
- Elecrow HDMI Display Monitor 7 Inch 1024X600 HD TFT LCD + Touch
- Intel SSD 600p Series (512GB, M.2 2280 80mm NVMe PCIe 3.0 x4, 3D1, TLC)
- Mailiya M.2 PCIe to PCIe 3.0 x4 Adapter
- uxcell PCI-E 4X Slot Riser Card Extension

System Design

In order to put all of the components in a portable easy to carry form factor, the team decided to proceed with a 3D printed case. This case will house all of the hardware listed above in order to accomplish the tasks of capturing the data obtained and processing it to produce real time feedback.

Thermal Capacity

After careful consideration and thought, the case design will be created with Acrylonitrile Butadiene Styrene (ABS) filament due to its particularly attractive properties. With a heat deflection of 83°C to 110°C (181.4°F - 230°F) and a strength to weight ratio of 79 kN-m/kg, we believe ABS is a suitable choice given the extreme temperatures the TX2 can theoretically achieve while running all cores for an extended periods of time. The *Thermal Stress Test* (shown on the next page), shows the TX2 doesn't reach temperatures higher than 37°C under a full stress test. We do however predict that the Guatemalan jungle will have a different ambient setting at around 40°C compared to the cooler 26°C temperatures here in San Diego. As such we tailored the design of the TX2 case to have accessible airflow and proper ventilation.



Power Supply

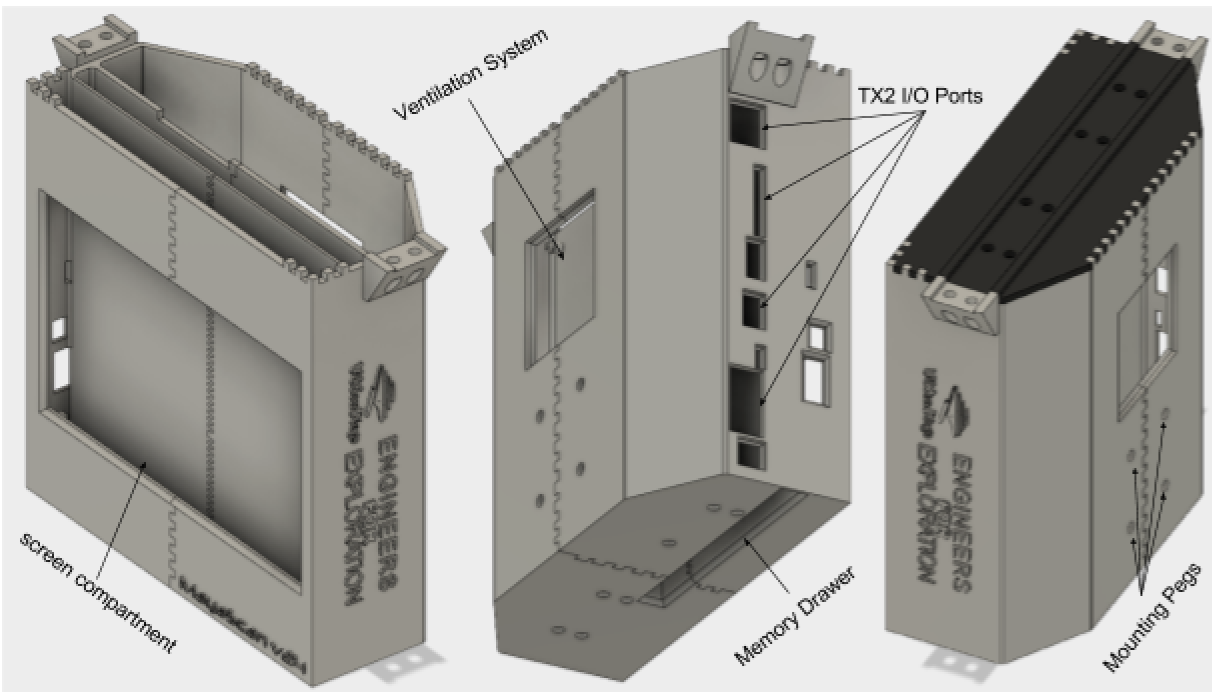
The idea of a portable and easy to carry form factor carried over into the power supply design as well. We need to be able to power the device for as long as possible to maximize efficiency and scan time, while using the smallest battery pack possible to minimize weight. The TX2 has a typical energy usage of about 7.5 watts under load. The original wall outlet power cord supplies 5.5V; 14.5A - 19.6 V; 4.74 A, and our goal was to simulate the top end spec as close as possible, so that the device would not be able to tell the difference between the battery pack and the wall outlet.

The issue with matching the wall outlet specs is that 19.6 V; 4.74A requires a fairly large battery (so large that it isn't legally allowed to be transported on public aircraft to Guatemala). Therefore, we had to change our plan to match the low end of the power cord spec (5.5V; 14.5A) using a step down converter from the MultiStar 5200mAh Battery. MultiStar outputs 16.5V to our desired 5.5V input and the device claimed able to maintain average performance of this low end voltage supply. After some stress tests with the battery pack, it was obvious that some of the peripheral connections were not supplying enough power and our camera repeatedly powered off.

Our power solution then reverted back to simulating the top end spec of the factory provided power supply at 19.6V using a single smaller battery (MultiStar 5200mAh Battery that can be flown on public airplanes with our expedition team) in series with a step up power converter. This unfortunately was a little bulkier than the battery and the step down converter, but it was successfully able to power the device and could even be used to power a laptop charger and the Tango tablet, which were other crucial components of the secondary scanning device.

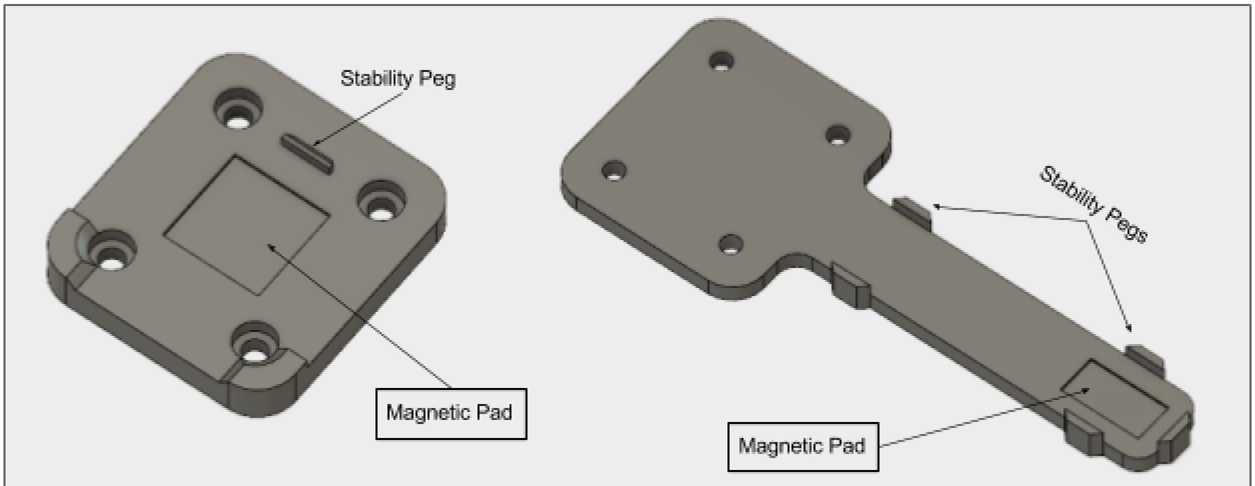
Finalized Model

Below you will find the baseline skeleton design for the case *MayaScan v24*. This case's purpose is to house all of the components for the TX2 based system including the memory modules, touchscreen, and the TX2 system itself. The final design has ports to house the multiple interfaces on the TX2 as well as the screen. The port holes were a little small, and as such needed to be widened. In doing so when refitting the TX2 on the case the Micro USB port came off of the board, resulting in the loss of that connection. The USB 3.0 port was physically modified, but still functional.



MayaScan v24: TX2 Components Case Cover: (left) Uncovered top of the case and screen compartment. (middle) TX2 exhaust vent, camera mounting pegs, memory drawer, and I/O ports. (right) TX2 Case with lid and handle mounting holes

The main case above is designed to interface with a large assortment of mounts used to attach each of the sensors provided in the Components List. The two cameras that needed 3D printed supports we the R200 and ZR300 cameras, as they would be more stable by attaching them through the magnets embedded in the back of the cameras and secured rotationally with stability pegs as pictured below. These mounts would then be connected through screws coming out of the back of the ***MayaScanv24*** figure above.



ZR300 Camera Mount (left) and R200 Camera Mount (right)

Past Milestones and Future Steps

Completed

Milestone 1 [May 2]

1. Jetson Handheld:
    a. Get rTAB running on TX2 and use it to create a 3D model of the hallways in CSE building. This is so we can get familiar with how the technology works and its quirks so that we can tackle different problems with a better perspective. [Tawfic]
    b. Build and test the power system for the Jetson TX2 [Andres]
        i. Should be able to power the system from just a battery
    c. Order and receive All Parts [Andres] [Nico]
        i. IMU
        ii. PCIE Hard drive
        iii. Jetson TX2 Backup
        iv. Lighting System (LED)
2. LightHouse System:
   Setup openVR on Linux with ROS and Vive ROS to set up the Lighthouse environment
3. General:
   Finalize all decisions such as which algorithms are to be used, how we are going to capture and store data, and how the system will work physically (power, bulk, weight, etc). Get a working system and test in various locations.

Milestone 2 [May 10]

1. Jetson Handheld:
    - 3D print Jetson case, and assemble system. [Andres]
    - 3D print Camera Mount and Imu Mount [Andres]

2. Lighthouse System:
    - Test lighthouse systems in various locations.
        - Map SeaLab in the dark
    - 3D print Camera Mount, Imu Mount, and lighthouse tracker [Andres]

Milestone 3 [May 12]

1. Assemble and lab test all parts of the Lighthouse system
    a. Measure the time it takes to capture 5’ x 5’ area

Milestone 4 [May 14]

1. Go to Anza Borrego to test in cave/tunnel environment
2. Make final adjustments/fix bugs before trip
3. Test previous 3D map versus new 3D map

Milestone 5 [May 19]

1. Assemble and lab test all parts of the Jetson system
    a. Measure the time it takes to capture 5’ x 5’ area
2. Quentin should have tuned the thresholds for the algorithms, using data collected from Anza Borrego mud cave expedition.
3. Take systems to Guatemala on Maya Trip.

Milestone 6 [June 1]

1. Complete course updates and requisites
2. Finish final presentation and documentation video
3. Finalize project (wiki) space

Conclusion

In summary, this iteration of the Maya project aimed to improve upon the complications created by previous implementations of 3D scanning archaeological sites. Some of these complications included setup time overhead when running LIDAR/SFM scans and lack of a real time feedback system. The systems that we’ve implemented this quarter aimed to satisfy these needs and as such reduce the amount of time taken to extract high quality scans from Maya archaeological sites.

E4E will continue to make improvements and work on this project in the coming years. Their current plan is to finish algorithm design of the TX2 device in an effort to remove the need for the backpack rig included with the Tango tablet device. The onboard lighting system will also continue to be perfected as the team learns more about the SLAM algorithm needs and how it interacts with the various types of cameras we can attach to the device. As always, a major improvement to the device would be to shrink any component to make it easier to operate and transport. One of the main focuses of future iterations will be to optimize the power supply to ensure that all components are fully functioning while keeping the weight and size to a minimum. Ideally, we will be able to send a team of scientists to an archaeological site with a fleet of wirelessly connected scanning devices, so an entire site could be scanned and stitched together in in just minutes.

References

[1] ElasticFusion: Real-Time Dense SLAM and Light Source Estimation, T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison and S. Leutenegger, IJRR '16

[2] ElasticFusion: Dense SLAM Without A Pose Graph, T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker and A. J. Davison, RSS '15

[3] ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras Mur-Artal et al.

[4] "3D Mapping with an RGB-D Camera", F. Endres et al.