**MoonTrek Telescope
Augmented Reality :**

**Project Report**

Sponsored by:

NASA Jet Propulsion Laboratory

Faculty Advisor:

Cwir Weronika

Liaisons:

Natalie Gallagos & Shan Malhotra

Team Members:

Cindel Lopez, Sianez, Fu-Cheng Chuang,

Eduardo Cruz, Jingchao Feng,

Byron Garibay, Daniel Gonzalez,

Tony Hong Jr., Matthew Johnson,

Niloy Azad, Jonathan Navarrete

13 May 2022



**Table of Contents**

1. Introduction…………………………………………………………………………………………….. 3
	1. Background…………………………………………………………………………………… 3
	2. Design Principles…………………………………………………………………………… 4
	3. Design Benefits……………………………………………………………………………… 4
	4. Achievements………………………………………………………………………………... 4
2. Related Technologies………………………………………………………………………………... 5
	1. Existing Solutions………………………………………………………………………….. 5
	2. Reused Products……………………………………………………………………………. 7
3. System Architecture…………………………………………………………………………………. 7
	1. Overview……………………………………………………………………………………….. 7
	2. Data Flow……………………………………………………………………………………….8
	3. Implementation…………………………………………………………………………….. 9
4. Conclusions..……………………………………………………………………………………………. 10
	1. Results.…………………………………………………………………………………………. 10
	2. Future…………………………………………………………………………………………… 10
5. References.………………………………………………………………………………………………. 11
6. **Introduction:**
	1. **Background**
* Sponsored by NASA Jet Propulsion Laboratories and modeled after NASA MoonTrek, a web portal developed to help users explore and learn about the Moon. Our MoonTrek applications continue to serve the previous team’s purpose of bridging the gap between a user provided telescope image and a smartphone or laptop.
* MoonTrek Telescope Augmented Reality is slightly different from the well-known Augmented Reality, since modern Augmented Reality involves the implementation of objects and information on top of a live scene in a camera. The goal of MoonTrek Telescope Augmented Reality is to implement objects and information on a web application, while also providing educational information to our users.
* A general challenge faced by previous years was accessing the JPL’s telescope given to school, and with direct help from CSULA’s Specialized Equipment Technician, Jillian Tromp, the team was able to use a telescope.
	+ With the use of the telescope and Image Revolutioner, the team was able to video record and document the use of the telescope pointed at the Moon. This achievement showed that it was possible to do with precise calibration.
	+ A hurdle we faced was not being able to display the live feed directly to the MoonTrek connect page due to time constraint and our implementation. Our solution was to display a placeholder for a future team to implement the live feed display while still allowing the user to capture a photo from the Image Revolutioner and upload to the Connect or Upload page.
* Covid 19 and trying to meet in person was also an obstacle the advisor, liaisons, and team dealt with. Discord & Zoom were used to conduct meetings.
* There were a number of ‘technical debts’ which are defined as steps the previous team did in terms of applications and their versions, which left some issues with the current implementation.
	+ Red flag #1 the team encountered is called ‘Dependency Hell’, wherein the team could not use the latest version of the Python (3.11) and had to resort to using Python 3.7.X because openCV’s license had changed and in turn forced us to use older dependencies for the application to function.
	+ Red Flag #2 1x CSS file to hold all formatting and navigating it is rough.
	1. **Design Principles**
* The project is divided into 5 groups, and 2 members each.
	+ The Team Leads are there to organize and assist the other teams with the weekly stand-ups, and fill-in the roles needed.
	+ Telescope Integration team is focused on telescope hardware, a user defined and provided computer, and lastly the webpage to upload the image.
	+ The Image Registration team focused on improving the image detection of the Moon, comparing differences of changing the thresholds of the algorithm, and performing unit testing.
	+ The 3D Modeling team was working on the software side of tackling the Three-Body Problem in a 3D scenario.
	+ The User Interface/User Experience team focused on improving the navigation of the website and rearranging the aesthetic of the web application.
* The team was working on a contingent of leaving comments and leaving helpful information due the lack of comments from the previous team.
* The web application was tested within itself in isolated systems and virtual environments using Anaconda, until they were sufficiently unit tested to be merged.
* The application itself is simple and deliverable as a work in progress and later to be deployed at a later time, and independent of any workflow.

	1. **Design Benefits**
* By rectifying the previous team’s work, the team has documented and left sufficient explanation of works in progress to erode the technical debt.
* There were strides to focus in different departments of the program such as unit testing in Image Registration, UI/UX design on their personal computer, 3D Modeling focused on unit testing of a scenario, and telescope integration to successfully output the picture for Image Registration.
* Each team was partitioned to some knowledge of Django’s architecture, Python’s coding schema, Python’s OpenCV, Ascom Alpaca, Javascript, or knowledge of the JPL API Calls and it helped control the codebase.

	1. **Achievements**
* Team Leads
	+ Keep morale
	+ Stand ups
	+ Assign sprints goals
	+ Host meetings
* Telescope Integration
	+ Video Record
	+ Connect page ability to capture and upload
* 3D Modeling
	+ Create 3D models for Moon, Earth and Sun
	+ Plug-in coordinates for position of models
	+ Set camera view based on nearest-point for Moon and Earth
* Image Registration
	+ Improvements were made to circle detection and image cropping in the image registration portion.
	+ Recognition of test images improved from 70% to 89%.
* UI/UX
	+ Telescope Integration Connect page was implemented, as well as an informative About page
	+ Improvements made towards overall layout and aesthetic of application
	+ Improved documentation on source code, allowing for easier understanding of code for future use
1. **Related Technologies**
	1. **Existing Solutions:**
* JPL’s MoonTrek is focused more on learning about the Moon, while the team’s application focuses on the bigger picture of where the user is in relation to the Moon, Sun, and Earth. The annotated points of interest also touch upon like JPL’s version.
* Using an older version of OpenCV was mixed because it disabled the new features in the new Python. Later on OpenCV contrib was allowed to bypass the patent expiration. This means there needs to be an overhaul of the Image Registration side to incorporate the latest contribution and eliminate a majority of the technical debt.
* The approach to using a telescope, Nexstar 9.25 Telescope, was first learning how to use and view the Moon in a location. Then mounting a Revolution Imager to the eye piece and recording it on a computer. The approach was costly because no explanation was given, but until reading the manuals was when the team was able to put it together. Calibrating was the second step and recording the information on the Revolution Imager and installing an abandoned recording software. Then input the CD Key “VHS3G-NMLGG-HGGGE-82A42-DBMGD” into the software to control the camera. After ensuring the hardware is ready is when you can record, but a problem arose where the picture was always zoomed in. The quickest solution was to video record the Moon moving in space and stitch it in a freeze frame to provide an actual image. In turn this begins the 2nd step of then uploading the picture to the application.
* The main approach for the Telescope Integration of MoonTrek was connecting the telescope to the MoonTrek and viewing its point of view through the Image Revolutioner hardware. From here, we can capture or record what is displayed and the photo can be uploaded to MoonTrek from the Connect or Upload page.
* The Connect page we created is composed of a container with a placeholder image where the live feed would be in a future implementation. There are buttons to capture and save a screenshot of this placeholder. There is also an upload button to upload this saved image to MoonTrek for image registration, so the user can remain on the same page.
* The approach for positioning the 3D models involved the implementation of a Two-Body approach due to the complexity of the Three-Body Problem.
* Cross-Origin-Resource-Sharing(CORS) was a big obstacle when constructing the 3D models. When the web application tries to call image resources from a third party server, it returns a CORS error. The team overcame this obstacle by uploading source images into Github and turning them into raw images.
* The User Interface main approach was to allow a seamless and easy to use application for the user as a means of receiving information in regards to the uploaded image of the Moon. From here the user will be able to freely navigate throughout the application, always knowing their current page.
* The aforementioned Connect Page was fully implemented, as well as the About Page to allow the users to gain information in regards to the application and the team working on it. Alongside this, the overall aesthetic of the application was updated to fit with the MoonTrek theme as a whole.

	1. **Reused Products**
* The program is coded in Python (3.7.X), and uses Django 3.1.7, web framework to create the application alongside with some CSS and HTML. The application uses OpenCV coded in Python, ThreeJS which uses Javascript and some light Shell scripting.
1. **System Architecture**
	1. **Overview**

The architecture for MoonTrek can be broken down into five main factors: User Interface, Communication, Registration, 3D Model, and Telescope.

Here is a diagram (DFD level 0) that shows how this architecture works at a high level: The System Overview gives a brief description of how our application functions and will later discuss how the program is to be planned. This is meant to be read from a non-technical/ high-level user’s point of view.

* + - Build upon the previous MoonTrek Django-web application. The user shall capture an image of the Moon from their telescope or upload a corresponding image into the web application.
		- The user shall give the image to the web application; then, in turn, the web application provides points of interest on the Moon such as craters, maria, and landing sites.
		- The augmented reality portion of the project improves the data overlays, creating a 3D model of the Moon created by Jet Propulsion Lab's high-quality images and user-uploaded images of the Moon.
		- The team's objective is also to complete the telescope for computer communication, improve the accuracy of the image registration, and create a 3D model of the Sun, Moon, and Earth based on the images uploaded.
		- The same 3D model of Earth will also annotate where the picture was taken and its time.



* 1. **Data Flow**

The system comprises five modules; User Interface, Communication, Registration, 3D Model, and Telescope. The main modules doing the work are the communications and registration modules. The telescope image will be provided to these modules and execute crucial tasks.

Images from the telescope will be passed to the registration module, where all image registration procedures occur. The correct coordinates will be generated and given back to the communication module. The communication module will fetch overlay data with the correct produced coordinates implemented from the registration module. Then the data will be passed to the user interface and 3D model module to display it to the user’s view of the Moon. The telescope module will connect the application and send the uploaded photo from the live feed to the user interface.



* 1. **Implementation**

The project was split into multiple sections to allow for efficient development: Telescope Integration, User Interface, 3D Modeling, and Image Registration. Each section plays a key role in presenting the progression of the project.

* Telescope Integration:
	+ Implemented by creating a new webpage called Connect
	+ Connect page contains a place for the live feed/placeholder image
	+ Connect page contains buttons to screenshot live feed section and save
	+ Connect page also has a place to upload an image
* User Interface:
	+ Create About page to show description of project and team
	+ Updated UI on every page of MoonTrek: Home, About, Upload, Connect
	+ Made application more fluid and presentable
* 3D Modeling:
	+ Generated 3D models representing the Moon, Earth, and Sun system.
	+ Set position of models in 3D space through the use of xyz vectors provided by JPL’s Planet Vector Search API.
	+ Set position of camera based on the latitudinal coordinates provided by JPL’s Nearest Point API.
	+ Optimized code for toggling between data layer textures of the Moon 3D model.
	+ Added an HTML form where users can enter the timestamp and latitudinal coordinates of where the Moon photo was taken so that they could update the positions of the 3D models.
* Image Registration:
	+ Improved circle detection for images
	+ Increased success rate for processed images
	+ Identified correct coordinates of the moon
	+ Removed most of the unnecessary circles detected
	+ Improved detection of the moon edge
	+ Increase in accuracy of circle detection led to better image cropping
1. **Conclusions**

**4.1 Results**

The results builds upon the team’s previous MoonTrek and adds feature such as a prepared page for finish telescope integration, an About page to see previous teams working on the project, improving the Image registration algorithm, creating a 3D overlay to help astronomers understand the position of the Moon, Earth, and Sun, and helped benefit the user experience of the application.

The team cleared the previous team’s technical debt, and focused on creating a better base and understanding of MoonTrek’s features and modules.

**4.2 Future**

* Capture 3D model images from user view point and use the image as base image in image registration to increase success rate and accuracy.
* Extract Exif data from a Moon image in order to pass the information as parameters to the API used to position the 3D models. This would remove the need for users to manually enter the required information in a html form.
* Implement live feed display for Connect page to be able to view telescope point of view on MoonTrek.
* Look for new hardware to control the Revolution Imager and incorporate Ascom Alpaca to complete the link between the application and a user’s computer that is connected to a telescope.
1. **Resources**
* *Amazon.com: VHS to DVD 3.0 Deluxe [Old Version]*. https://www.amazon.com/VHS-DVD-3-0-Deluxe-VERSION/dp/B000NDTRG2.
* “ASCOM - Standards for Astronomy.” Ascom-Standards.org, [ascom-standards.org/](http://ascom-standards.org/). Accessed 10 May. 2022.
* “Closest-Point API | Planet-Vector API” JPL, <http://54.157.167.17:5000>
* “Django Documentation | Django Documentation | Django.” Docs.djangoproject.com, [docs.djangoproject.com/en/3.1/](http://docs.djangoproject.com/en/3.1/.).
* Nasa.gov, 2016, [trek.nasa.gov/tiles/apidoc/trekAPI.html?body=moon](http://trek.nasa.gov/tiles/apidoc/trekAPI.html?body=moon). Accessed 10 May. 2022.
* “Nexstar Evolution 9.25 Telescope.” *Celestron*, https://www.celestron.com/products/nexstar-evolution-925-telescope.
* “OpenCV: OpenCV-Python Tutorials.” Docs.opencv.org, [docs.opencv.org/master/d6/d00/tutorial\_py\_root.html](http://docs.opencv.org/master/d6/d00/tutorial_py_root.html).
* *Revolution Imager*, https://www.revolutionimager.com/.
* “Three.js – JavaScript 3D Library.” Threejs.org, 2019, [threejs.org/](http://threejs.org/).