MoonTrek Telescope Augmented Reality: Project Report

Sponsored by:

NASA Jet Propulsion Laboratory

Faculty Advisor:

Weronika Cwir

Liaisons:

Natalie Gallegos & Shan Malhotra

Team Members:

Nadir Abdusemed, Jackson Bentley,

Jesus Cruz, Youssef Elzein,

Derek Guevara, Joe Hineno,

Rich Ho, Owen Ramirez,

Salman Sheikh, Alex Sherzai

12 May 2023

Table of Contents

1. Introduction:

1.1. 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.
- Previous code was implemented in Django and had more than 11,000 lines of code and was prone to bugs when updated. Changing to Vue allowed us to modularize the sites architecture and greatly reduce the code base

1.2 Design Principles

- The Principal objective was to correctly annotate the user image. In order to achieve this the following subtasks were created:
  - Coordinate mapping
  - 3D model of the Earth, Moon, and Sun
  - Create database of images to test registration model
  - Context-aware image registration
- Team roles were broken down as listed below. Team members were free to and did work outside of these roles when necessary
  - Coordinate Mapping - Jesus Cruz
  - 3D Modeling - Alex Sherzai, Jackson Bentley
  - Image Registration - Jesus Cruz, Joe Hineno
  - Image Database - Derek Guevara, Rich Ho, Nadir Abdusemed, Owen Ramirez
  - Project Leads - Youssef Elzein, Salman Sheikh
- Two sites were built independently with the same frameworks, MoonTrek AR and YourMoon.

- 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.

1.3. Design Benefits

- Updating to Vue allowed us to modularize the code base and greatly reduce the lines of code.
- Initial implementations of the 3D model were not producing correct results so a debugger was created(3 Body Debugger). This allowed us to track positions over time and correctly determine where errors were being introduced. Resulting in a the ability to create a context aware reference image
- The context aware reference image allowed us to perform context aware image registration which greatly improved the accuracy of the overlays that were applied to the user image.

1.4. Achievements

- Team Leads
  - Keep morale
  - Standups
  - Assign sprints goals
  - Host meetings
- Coordinate Mapping
  - Map from a flat surface to a sphere
  - Account for transformations due to rotations about any axis
- 3D Modeling
  - Create 3D models for Moon, Earth, and Sun
  - Plug-in coordinates for the position of models
  - Orient the moon correctly with the correct lighting
  - Set camera view based on nearest-point for Moon and Earth
- Context-Aware Image Registration
  - Create dynamic reference image to perform context-aware image registration
  - Improved accuracy of the transformation matrix for overlays
- Database
  - Create a database of test images for MoonTrek
  - Store image file and metadata of the image
  - Upload images on *YourMoon* web application

2. Related Technologies

2.1. 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 JPL's version.
- The previous implementation used a static image of the moon to perform image registration. This resulted in overlays that did not fully map onto the user image and did not properly align features. Points of interest were also not located in the correct location. The updated implementation now uses a dynamic image to perform context-aware image registration
- The User Interface's main approach was to allow a seamless and easy-to-use application for the user as a means of receiving information in regard 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.

2.2. Reused Products

- The program is coded in Python (3.7.X) and uses Django 3.1.7, a web framework to create the application alongside some CSS and HTML. The application uses OpenCV coded in Python, ThreeJS which uses Javascript, and some light Shell scripting.

3. System Architecture

3.1. Overview

- The architecture for MoonTrek can be broken down into five main factors: User Interface, Communication, 3D Model, and Registration.
- Our web application is built using the REST framework which allows us to separate server-side and client-side code. For the back end, we are using ExpressJs which is responsible for server-side processing and fetching data from other servers. For the front end, we are using VueJs which is the client that will be served to users on their initial request. The client will run in the user's web

browser and only makes requests to the ExpressJS backend for data and processing which cannot be done in the browser.

- The three dimensional model is responsible for generating accurate images of what the Moon should look like relative to a particular time and place on Earth. We use API endpoints that JPL set up to fetch the positions of Earth, Moon, and Sun at a particular time, along with the orientation of the Moon. We can then take textures of Earth and Moon, wrap them around spheres with their respective diameters, and then set their positions. When our model is made, it serves as an accurate representation of what the Moon looks like relative to the given location at the given time.
- When developing the three dimensional model which was meant to generate images of the Moon, we noticed that the images weren't matching what we expected. In order to analyze how the bodies were moving over time in our model, we had to create the Three Body Debugger. This tool works similarly to the three dimensional model but updates the positions of the bodies every second. We can analyze their movements either hour by hour or day by day. Had we not developed this tool, we likely would not have noticed that the model issue was coming from the position data which we got from JPL's API endpoints.
- 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.
- YourMoon Overview: We needed a database of user captured Moon images to verify the accuracy of the registration techniques. This was not available anywhere online, so we decided to create our own database, which led us to the creation of a second, complimentary, website called YourMoon. YourMoon allows users to upload a photo they took of the Moon, manually crop it so that any external noise is eliminated, and upload it to our database.

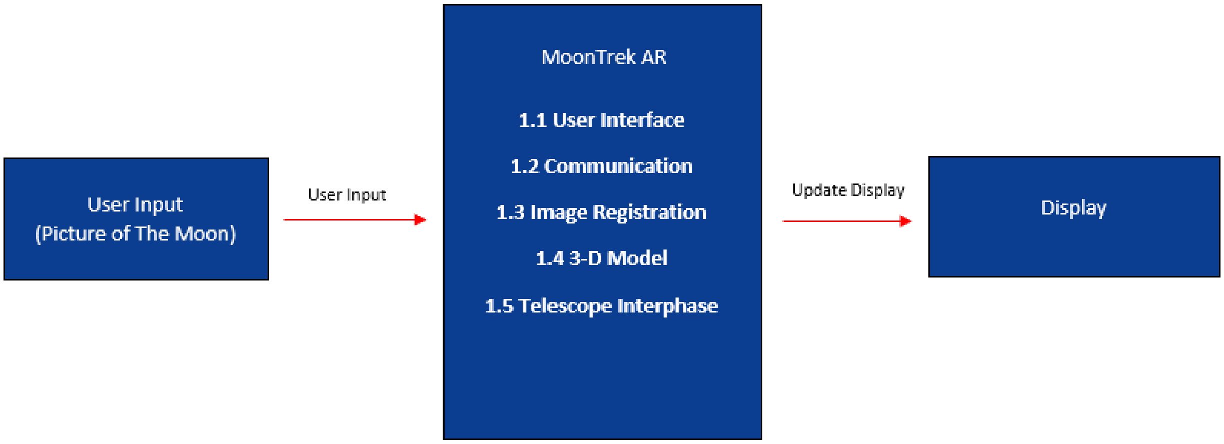

Figure 1: MoonTrek Data Flow Diagram 0 (DFD 0)

3.2. Data Flow

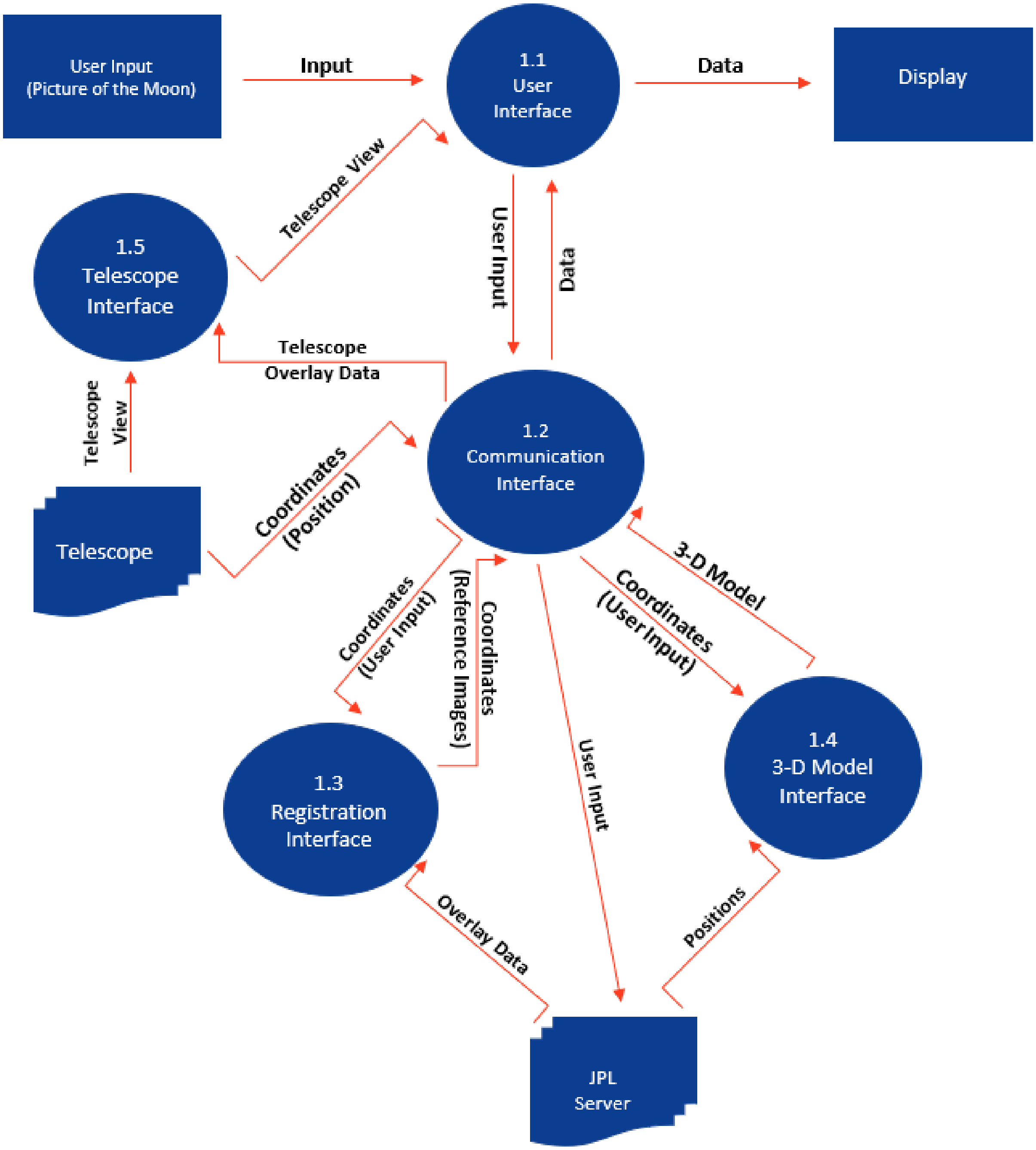

Figure 2: MoonTrek Data Flow Diagram 1 (DFD 1)

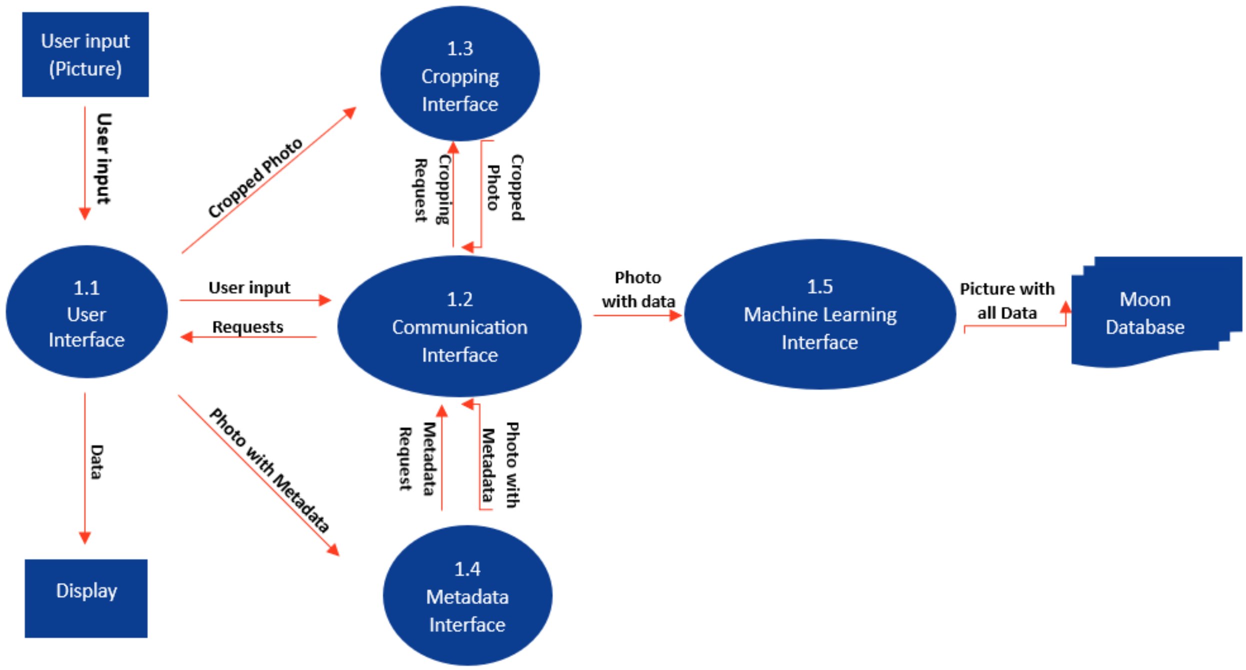

Figure 4: Your Moon Data Flow Diagram (DFD 1)

3.3. Implementation

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

- Coordinate mapping:
  - Mapped Selegraphic coordinates from a flat surface to a sphere
  - Account for rotations about all 3 axis(yaw, pitch, roll)
  - Given pixel location with relation to the center of the image, rotations about each axis return the Selenographic coordinate
- 3D Modeling:
  - Generated 3D models representing the Moon, Earth, and Sun systems.
  - Set the position of models in 3D space through the use of XYZ vectors provided by JPL's Planet Vector Search API.
  - Set the position of the 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.
- Image Registration:
  - Improved circle detection for images
  - The 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
  - Context-aware image registration
  - Correctly position overlays on user images
- Image Database:
  - Create an About page to show a description of the project and team
  - Updated UI on every page of MoonTrek: Home, About, Upload, Connect
  - Made application more fluid and presentable

4. Conclusions

4.1. Results

The results build upon the team's previous MoonTrek and add features such as a circle detection, 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.

4.2. Future Work

- Implement all overlays available on the MoonTrek website.
- Implement a complete telescope for computer communication, and implement a live feed display for Connect page to be able to view the telescope point of view on MoonTrek.
- Test models extensively with larger database of images

YourMoon

- Extract Exif data from the moon image that the user selects.
- Add better cybersecurity.
- Have users log in and see the pictures they have uploaded.

Website/Server Security

- Use a Web Application Firewall (WAF)
- Implement Two-Factor Authentication (2FA)
- Use Encryption for SQL Data
- Implement Role-Based Access Control (RBAC)

User Engagement

- Enhance user experience
- See how the moon looked from user's location

- See how the moon looked on your birthday
- See how the moon looked a century ago
- Standard dataset for computer vision researcher

5. 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/. 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/.
- Nasa.gov, 2016, 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.
- *Revolution Imager*, https://www.revolutionimager.com/.
- "Three.js – JavaScript 3D Library." Threejs.org, 2019, threejs.org/.