MOON TREK TELESCOPE

California State University Los Angeles
Sponsor: Jet Propulsion Lab
Spring 2021

Nicolas

TEAM MEMBERS AND ROLES

Nicolas Ojeda
Team Lead
Backend (django) ,
Graphics(Threejs)

Alex Lamb
Communications Lead

Albert Ramirez
Lead Backend Developer

Dakota Townsend
Documentation Lead

Jacob Frausto
QA/Testing Lead

Kevin Aguilera
Development
(Graphics Model)

Gerard Rosario
Front End Graphics
Lead

Pavit Chawla
User Interface Lead

Elvira Sakalenka
Computer Vision Lead

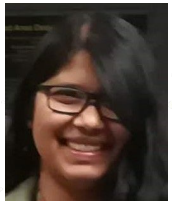
Nicolas

LIAISONS

Shan Malhotra

JPL

Jet Propulsion Laboratory
California Institute of Technology

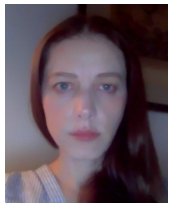


Natalie Gallegos

JPL

Jet Propulsion Laboratory
California Institute of Technology

Faculty Advisor



Weronika Cwir

CAL STATE LA

Nicolas

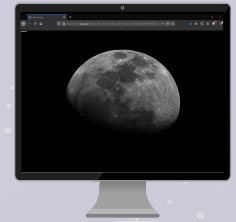
Agenda

1. Overview
2. Major requirements
3. Project technologies
4. General approach
5. Project challenges & solutions
6. Demo

Nicolas

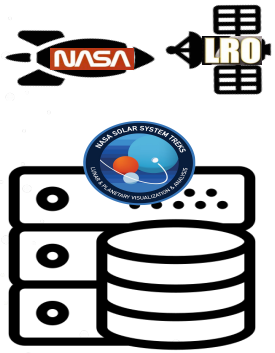
overview

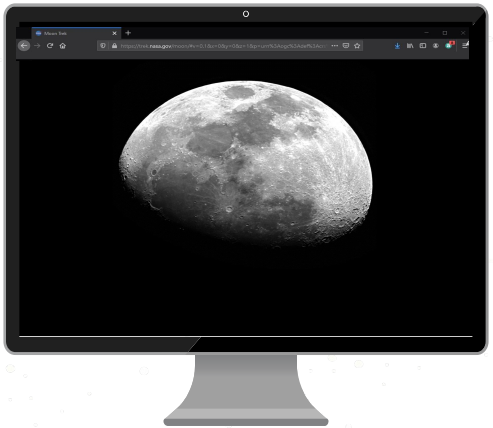
- Stream from telescope to web app
- Fetch data from JPL's Moon Trek
- Apply data to user's image

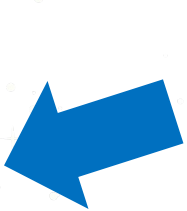


Nicolas

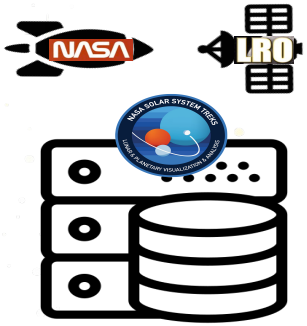
Moon Trek Telescope | overview

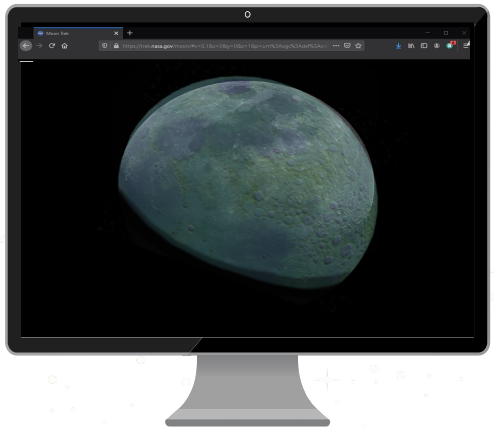


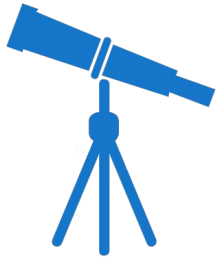


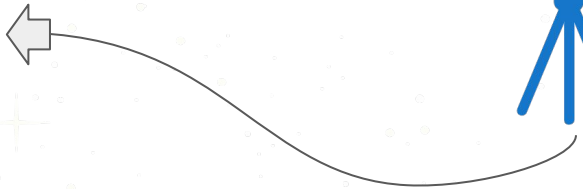


Moon Trek Telescope | overview



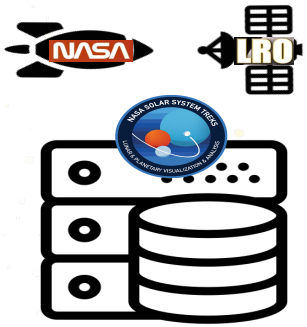


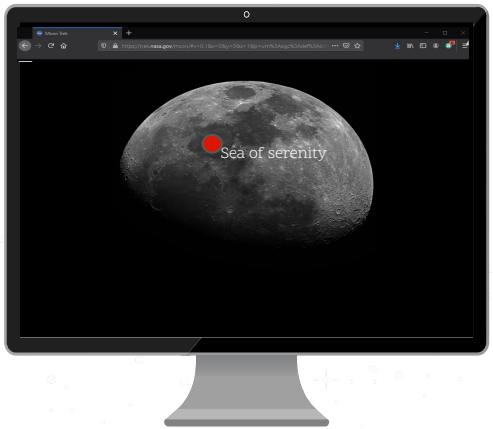


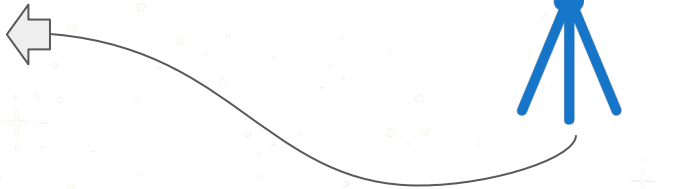


Feature:
Image captured with user's telescope with elevation overlay

Moon Trek Telescope | overview





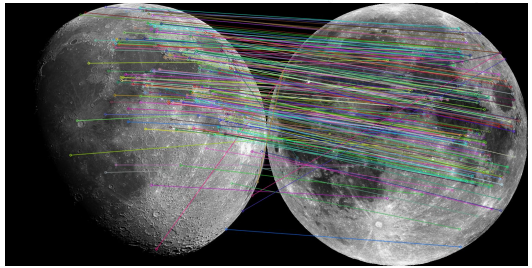


Feature:
Image captured with user's telescope with annotation of craters , etc.

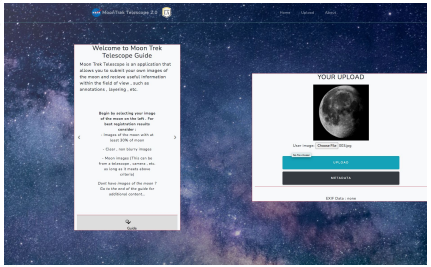
Major requirements

Nicolas

Moon Trek Telescope | Major Requirements

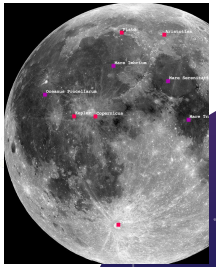


1. Route user's images to web app



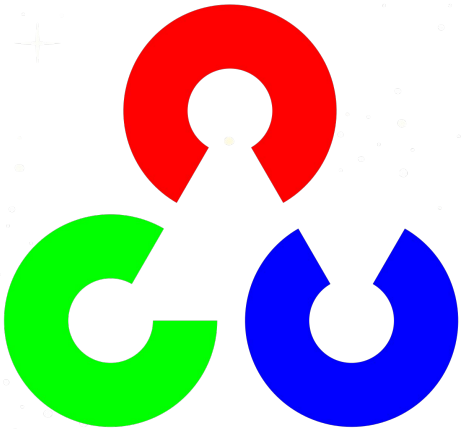
2. Register it to mosaic of the Moon

3. Annotate it with data

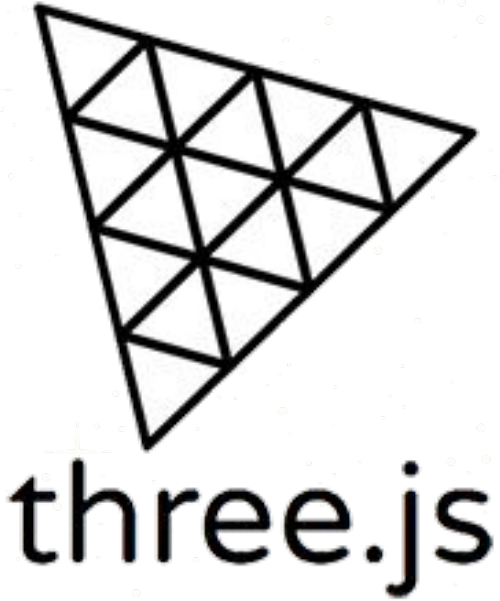


Technologies

Technologies | Backend/server side



Technologies | Front-end/client side



Approach

Image registration of source image to a reference image that correctly correlates with the Moon Trek Portal

Alex

Moon Trek Telescope | Approach

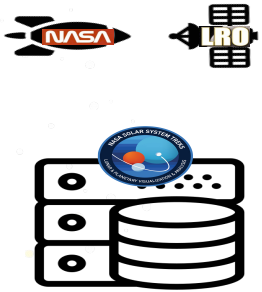


Image Registration

Reference Image

3d model

Goal is to exactly map the source view in the reference image .

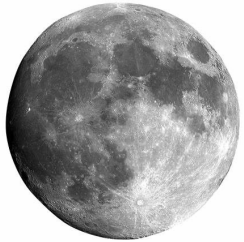
Source Image

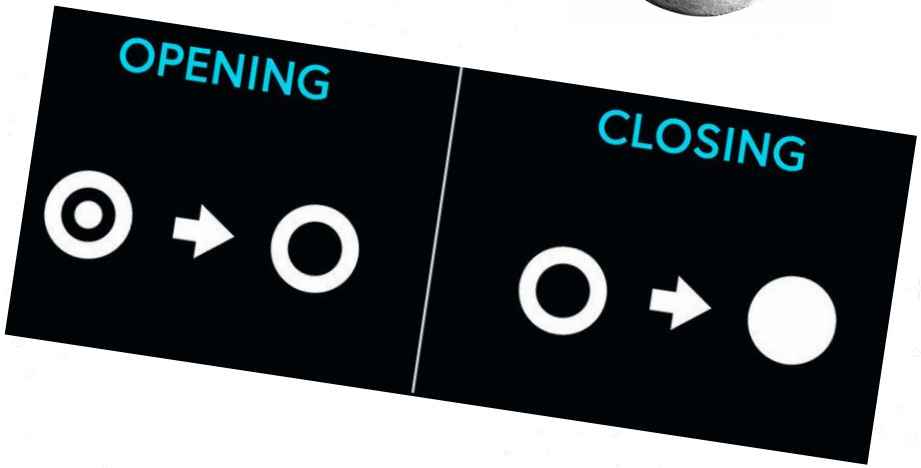
OpenCv - Circle Detection

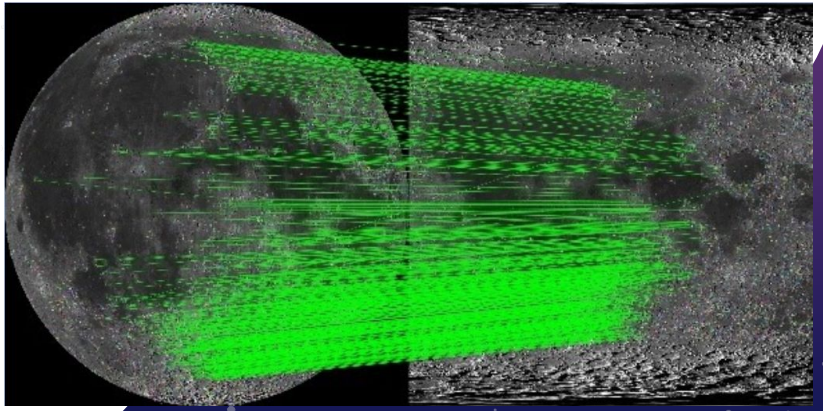
'time' and 'location' data

Alex

Moon Trek Telescope | Approach







Moon Trek Telescope | Approach

Request

```
http://54.157.167.17:5000/nearest-point/earth/moon/-118.173225/34.195109/2020-10-07T01:10:45
```

Response

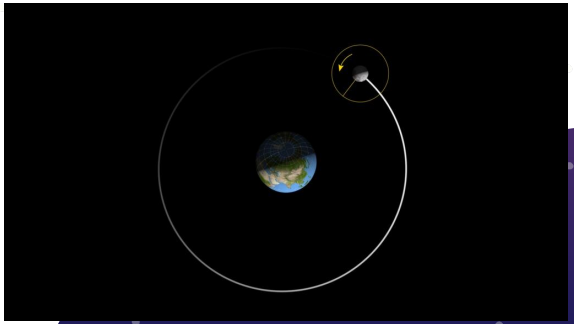

```
{
  "observer": "earth",
  "target": "moon",
  "altitude_km": 1737.4,
  "longitude": -4.551454259598997,
```


Request

```
http://54.157.167.17:5000/lat-to-rect/moon/earth/0/0/2019-10-07T01:10:45
```

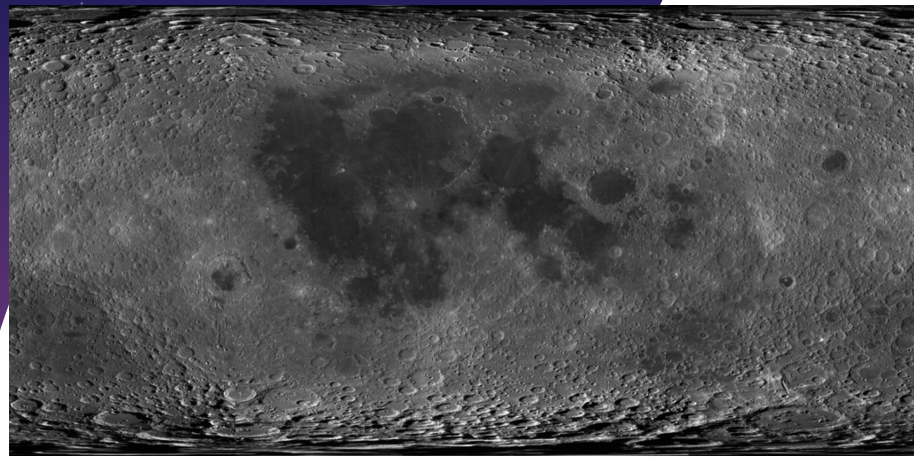
Response

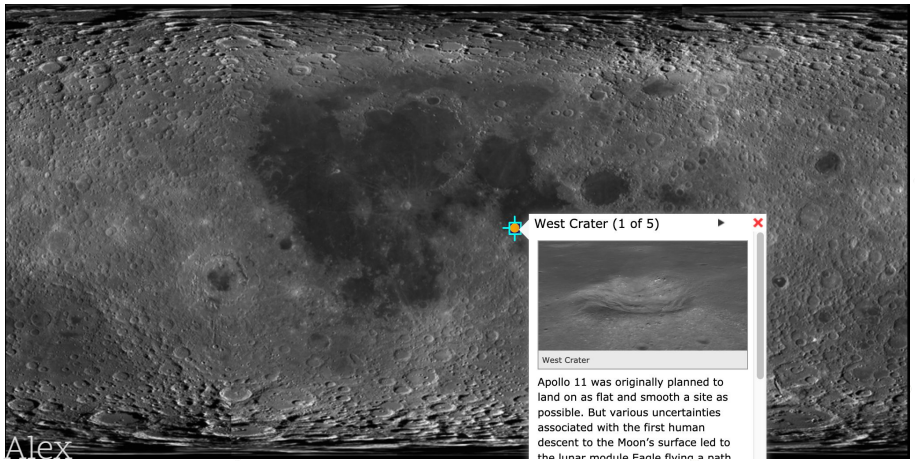

```
{
    "origin": "EARTH",
    "units": "km",
    "positions": {
        "moon": {
            "x": -12282.268170093246,
            "y": -368872.8597873927,
            "z": -147150.61159076623
        }
    }
}
```

Alex

Moon Trek Telescope | Approach





- Determine what is moon in source image
- Perform morphological transformations on source image
- Register source image with reference image
- Produce faithful longitude/latitude coordinates for source image
- Apply data layers to source image

Alex

Challenges/Solutions

Jacob

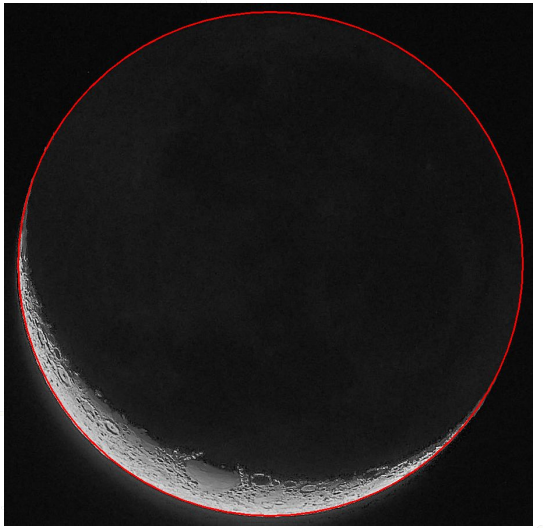
Challenges/Solutions | Circle Detection

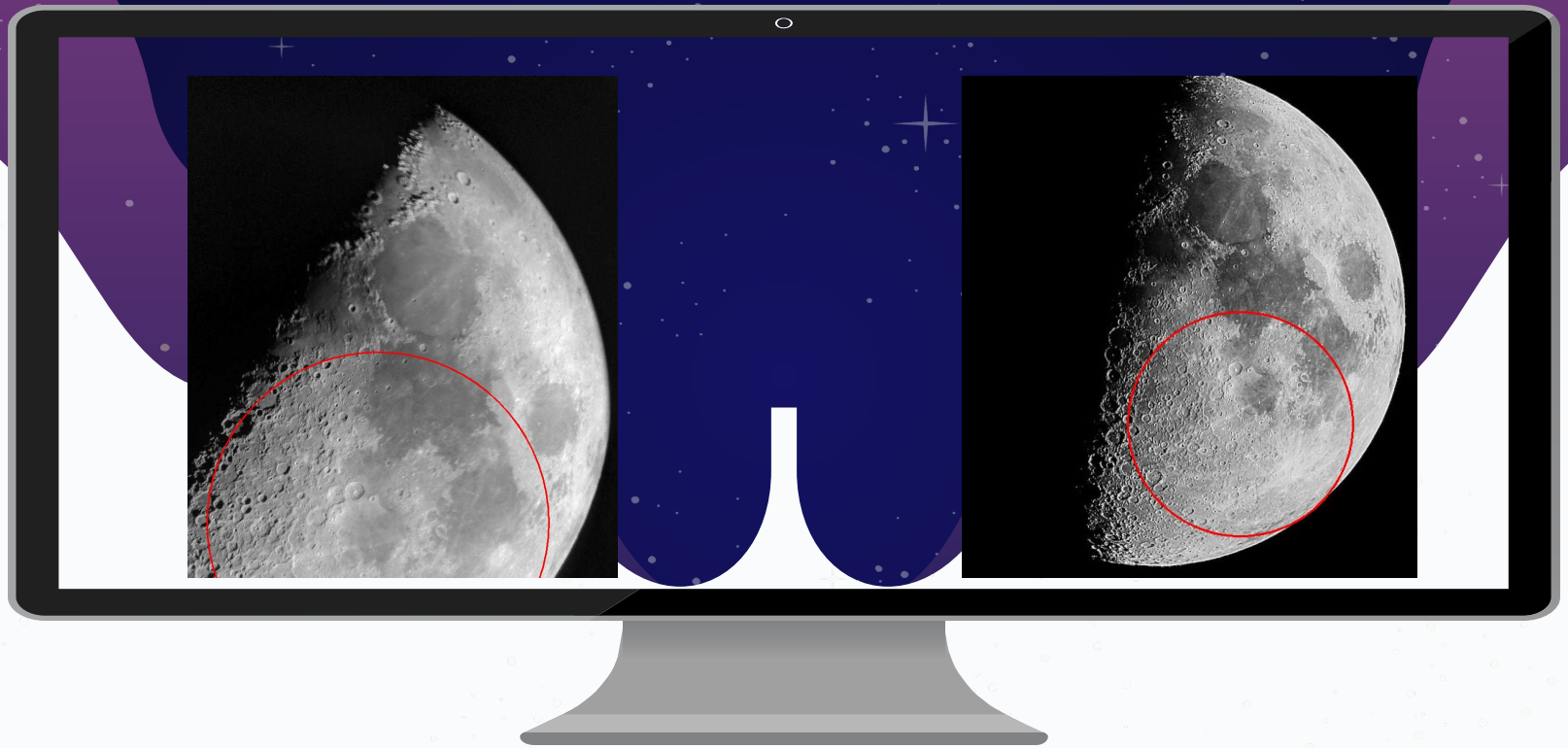
- OpenCV HoughCircles()
- Accuracy
- Optimization

Circle Detection Challenges

OpenCV HoughCircles()

- **image:** image for circle detection
- **detection method:** HOUGH_GRADIENT - method for circle detection
- **dp :** inverse ratio of resolution
- **min_dist:** minimum distance between detected centers
- **param_1:** Upper threshold for the internal canny edge detector
- **param_2:** Threshold for center detection
- **min_radius:** (unused) Minimum radius to be detected
- **max_radius:** (unused) Maximum radius to be detected





Circle Detection Errors

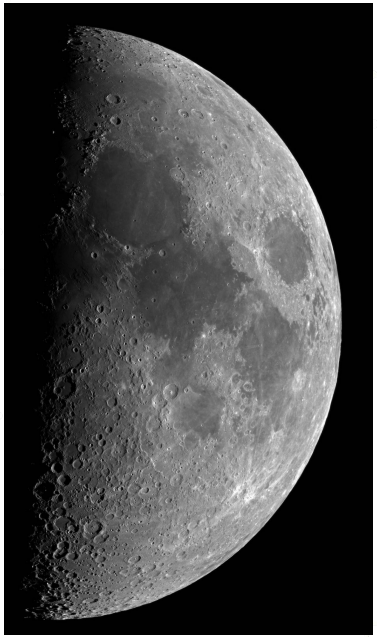
circle detection successes

Jacob

Circle Detection Challenges

Optimization

- Experimented with parameter values to increase accuracy
- Successful in decreasing runtime



Jacob

Challenges/solutions | Image registration

Image registration is an image processing technique which transforming different set of data (multiple photographs, data from different sensors, times, depths, or viewpoints) into one coordinate system.

Example of data:

User image

Reference image

Image registration |opencv(computer vision)

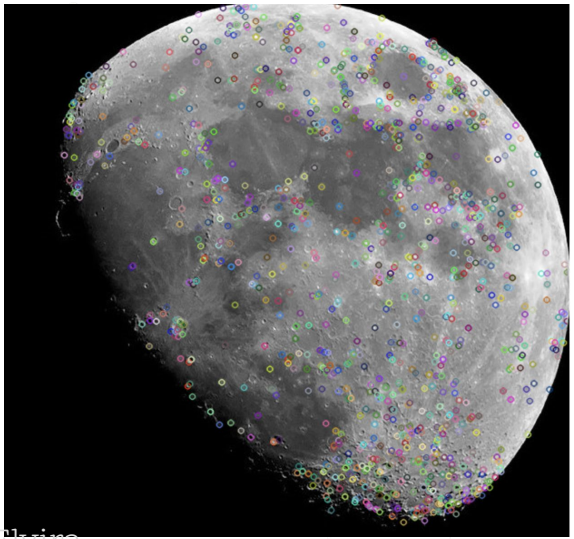
Sample of 108 images and our challenges :

- ORB (Oriented FAST and Rotated BRIEF)
- SIFT (Scale Invariant Feature Transform)
- BRISK (Binary Robust Invariant Scalable Keypoints)
- SURF (Speeded-up Robust Features)

Image registration | SIFT(Scale Invariant Feature Transform

General stages of automatic image registration:

- Feature detection and description



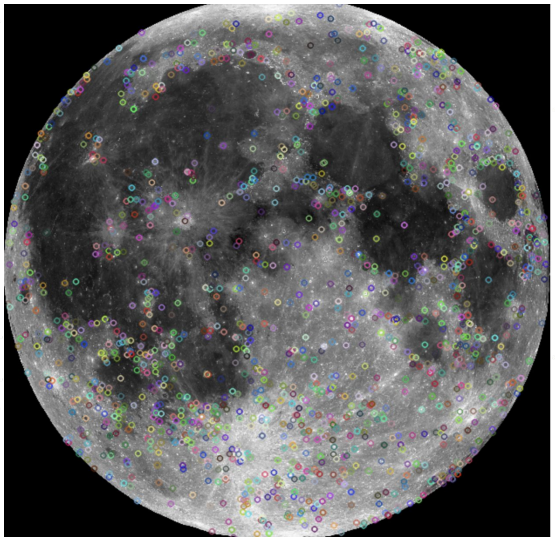


Image registration | SIFT(Scale Invariant Feature Transform

- Feature matching

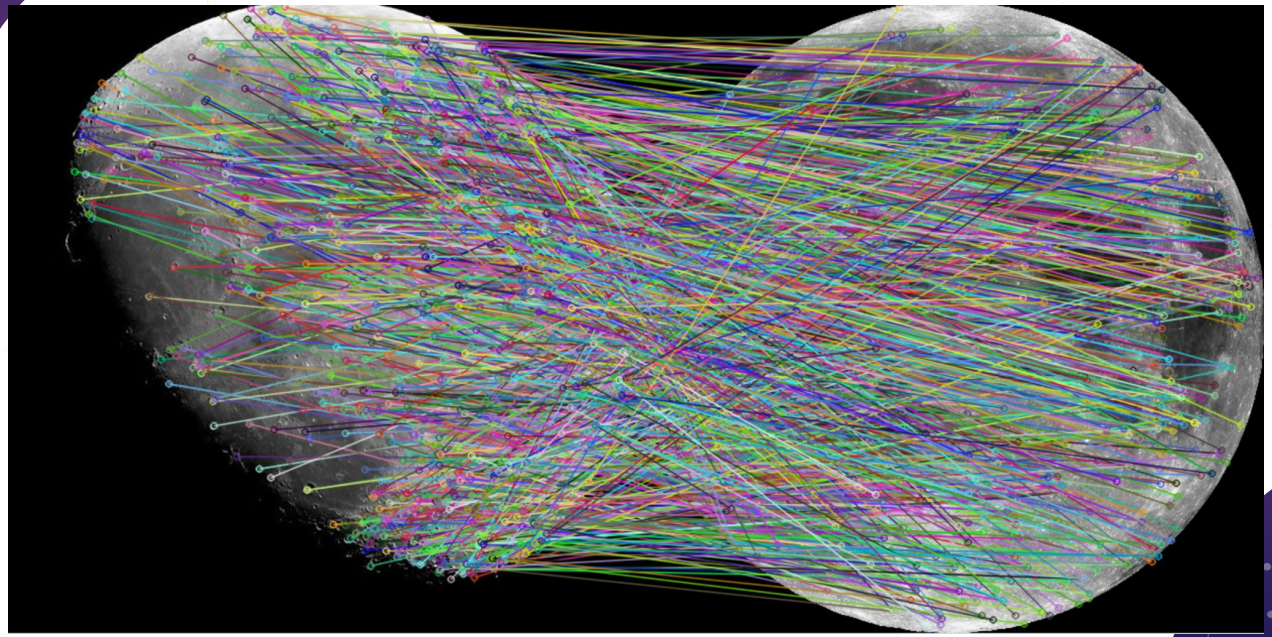


Image registration | SIFT(Scale Invariant Feature Transform

- Outlier rejection(The RANSAC - Random Sample Consensus)

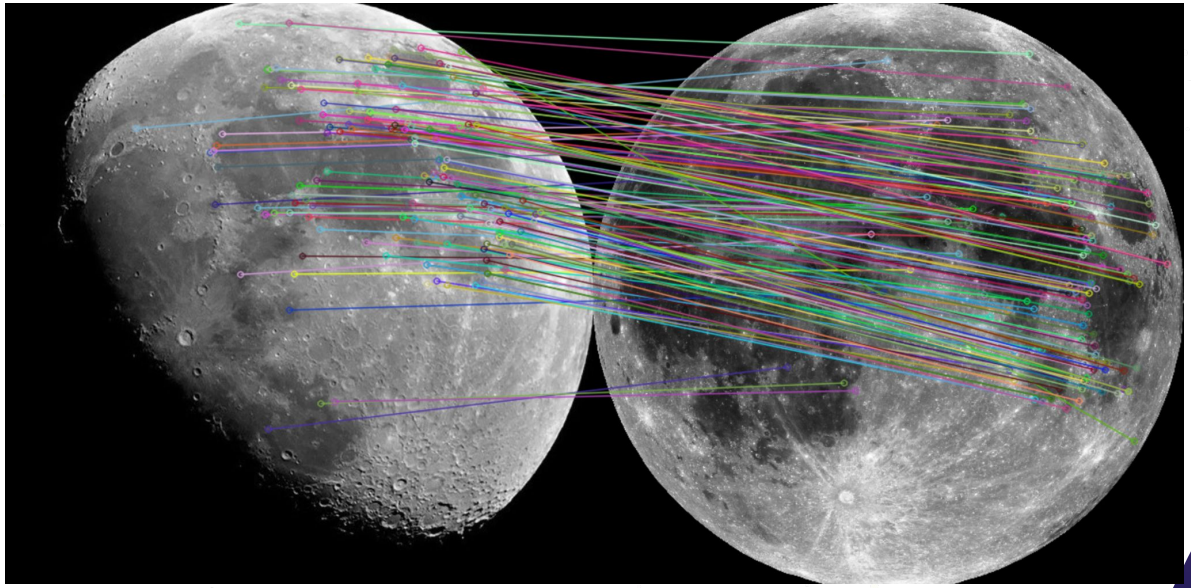
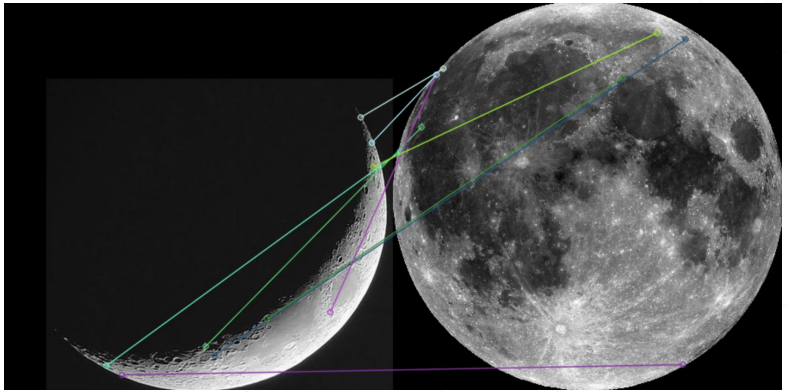


Image registration | SIFT(Scale Invariant Feature Transform

- Derivation of transformation function
- Image reconstruction

Image registration | SIFT(Scale Invariant Feature Transform

Results:
About 30 % images failed



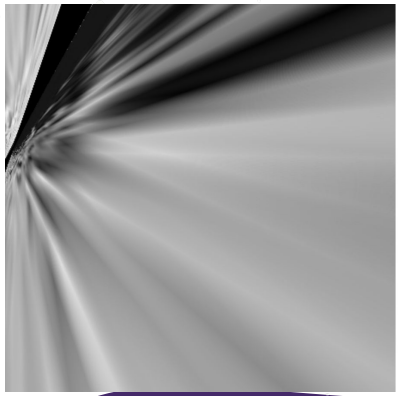
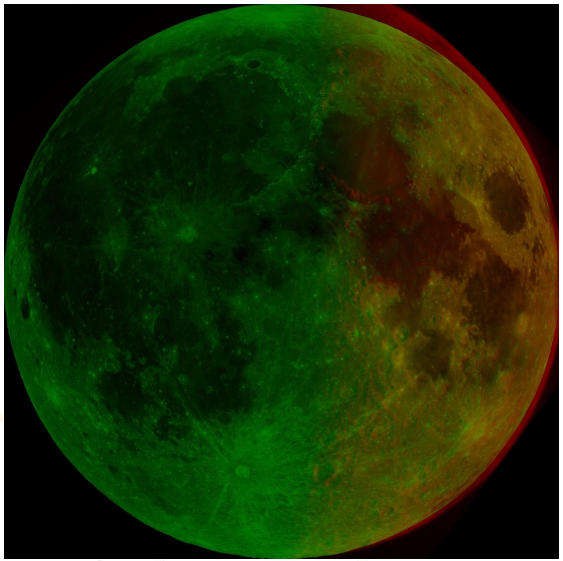
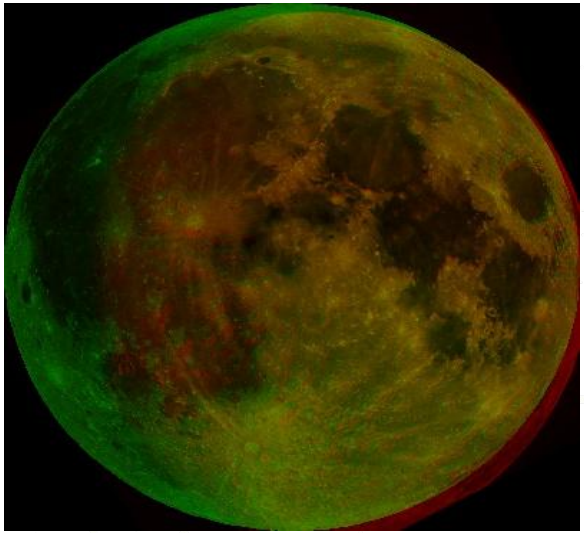


Image registration | SIFT(Scale Invariant Feature Transform

About 70% successful image registration



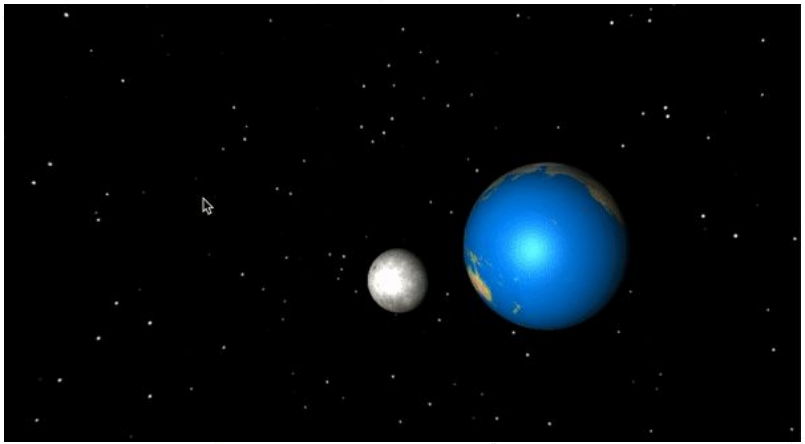


Challenges/Solutions | 3D Model Purpose

- Match telescope view with 3D model
- To help calculate nearest point of what's visible through telescope
- Important for image registration

Kevin

Challenges/Solutions | 3D Model Challenges



- Original 3D model plans
  - Lunar phase : Shape of the Moon due to sunlight
- Three.js
  - New language
  - Complex understanding

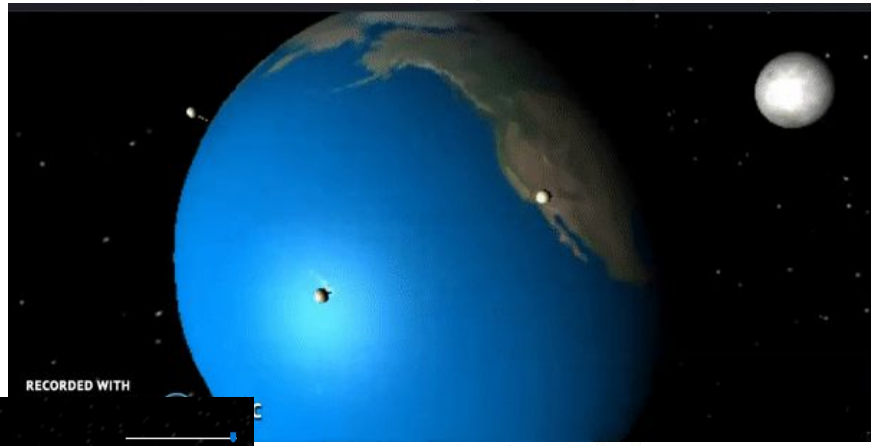
three.js

Challenges/Solutions | 3D Model Changes

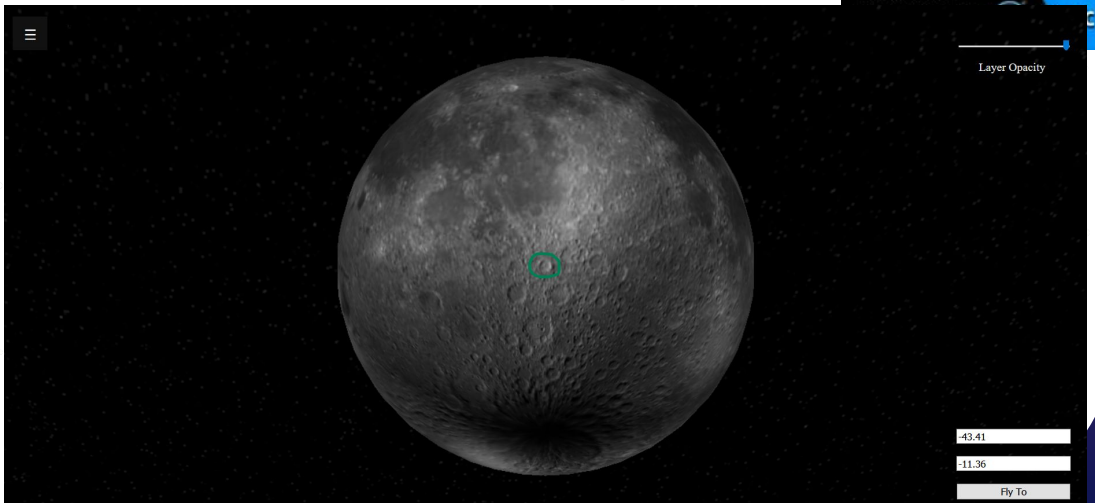
- Suitable substitute for the 3D model
- Use of existing image of the Moon called the Globe LRO REF
- New 3D model with user experience

Kevin

Challenges/Solutions | 3D Model Fly To

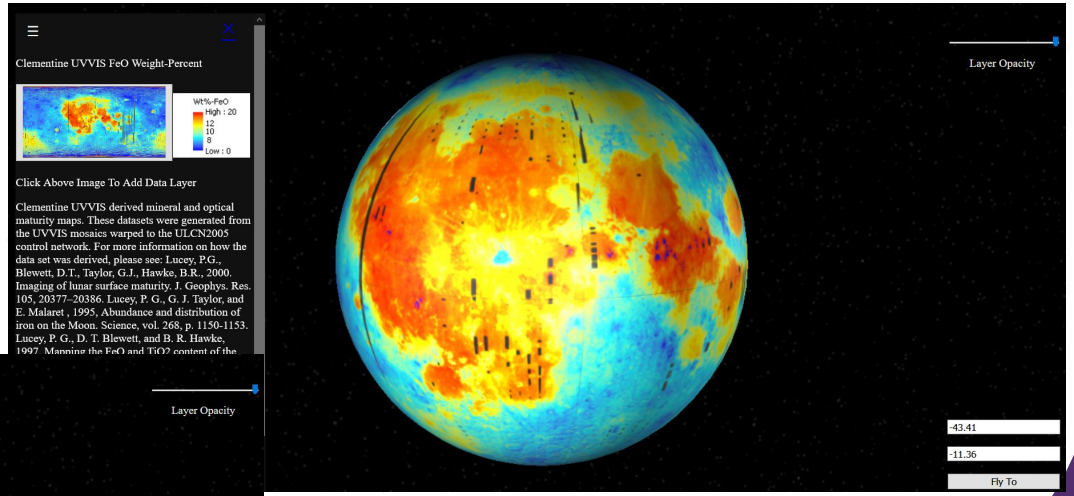


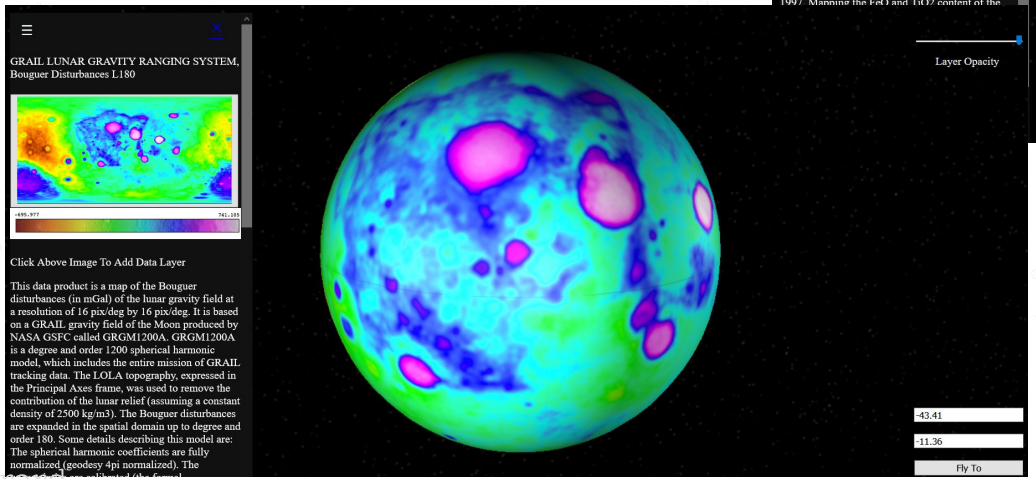
- Points on sphere
- For accuracy
- Original function
  - New function.



Challenges/Solutions | 3D Model Layers

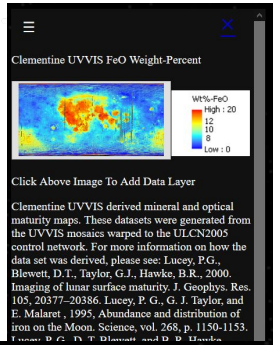
- Adding annotations
- User Interaction
- Implementation with OpenLayers
  - With Threejs

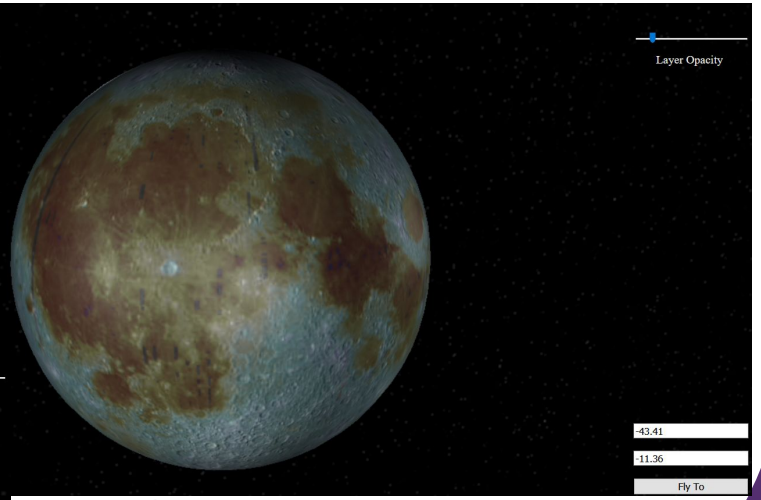


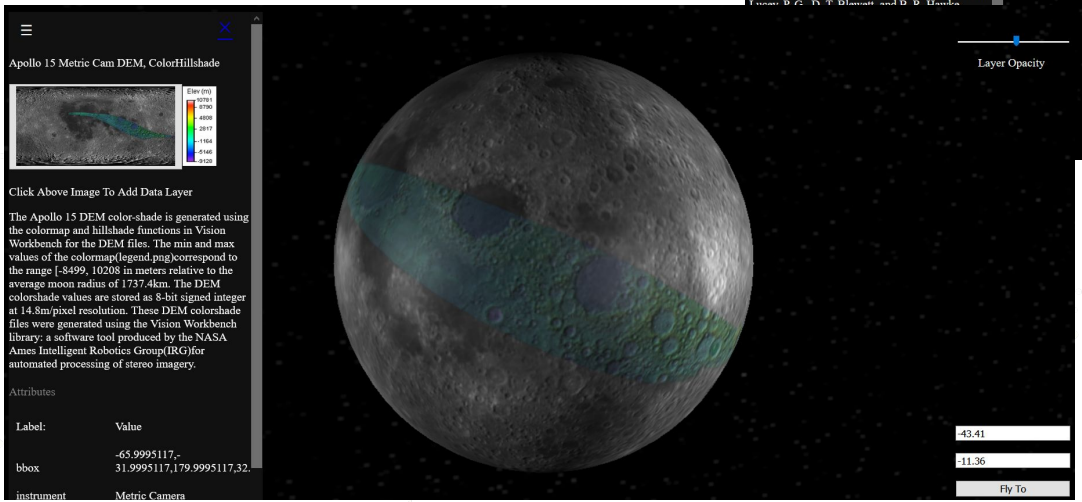


Challenges/Solutions | 3D Model Opacity

- A way to check the annotation
- Level can be designated
- Implementation in OpenLayer

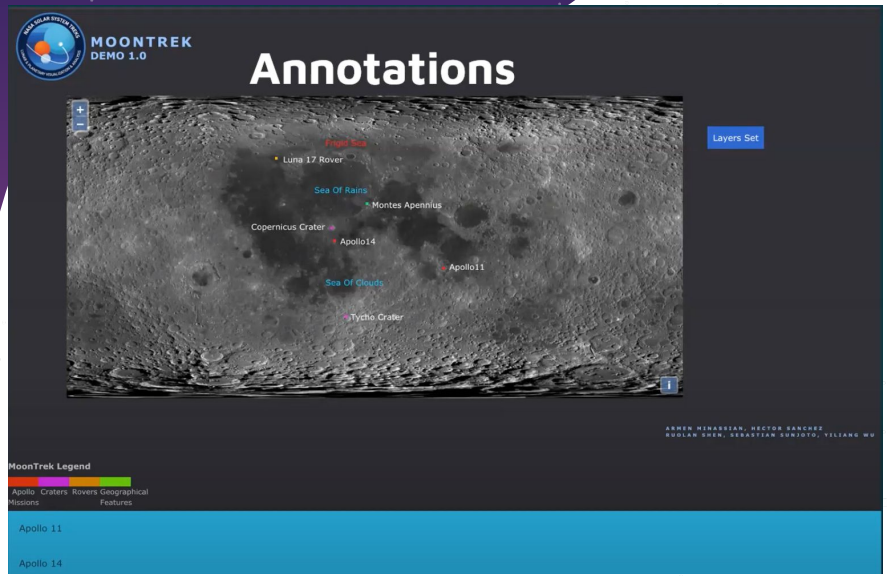




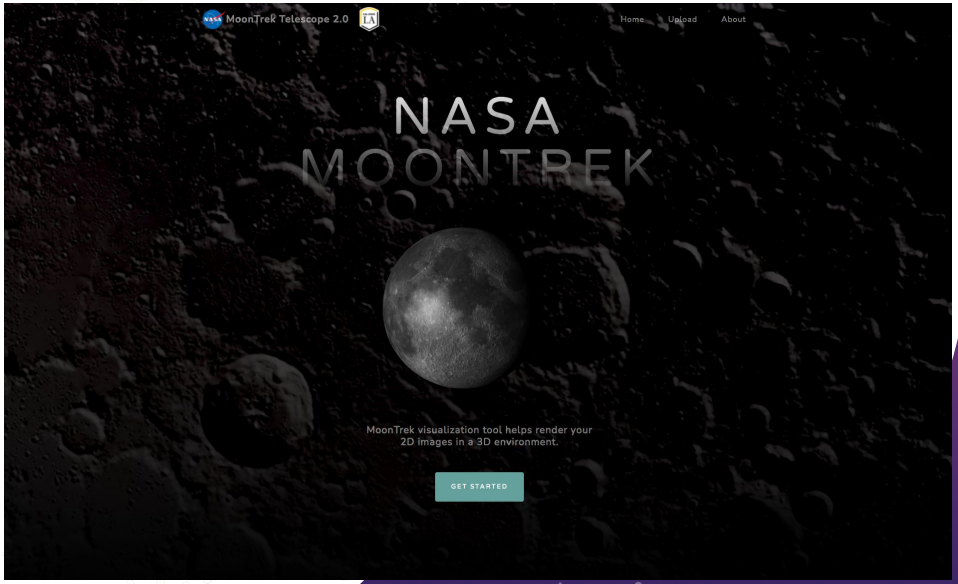


Gerard

Challenges/Solutions | New Interface



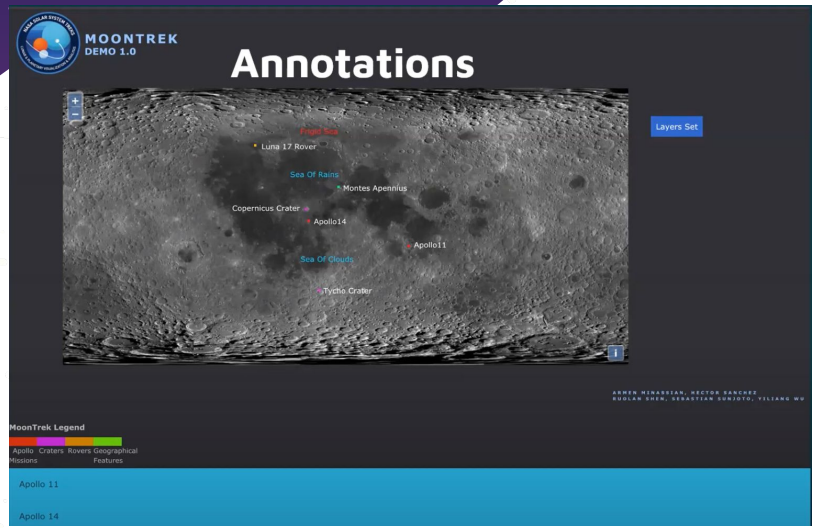
OLD: not straightforward,
What do you do to begin?



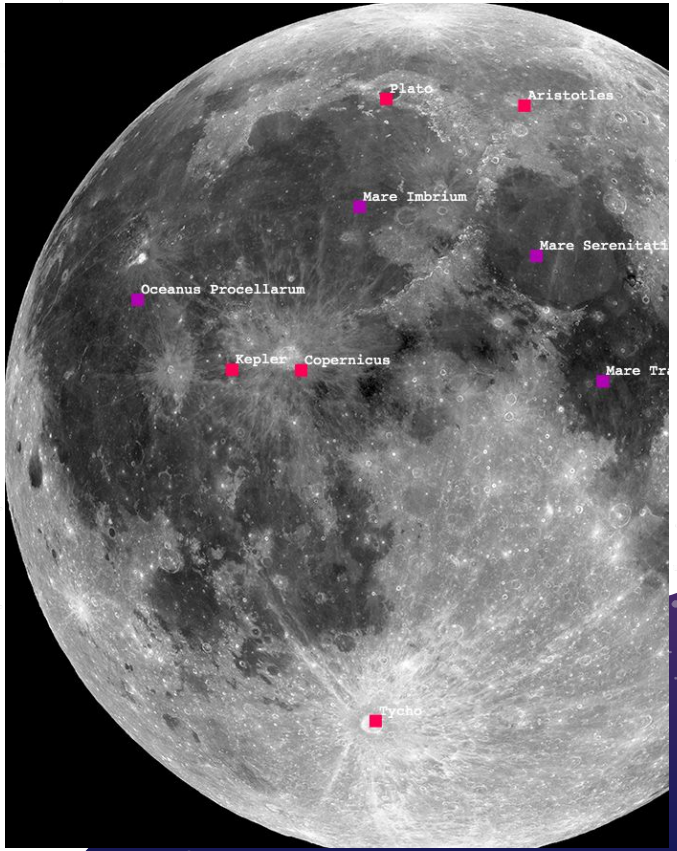
NEW:
More intuitive
Cleaner
Sleeker

Pavit

UI | Registration & Annotations



OLD



NEW

MoonTrek Telescope 2.0

Home Upload About

Welcome to Moon Trek Telescope Guide

Moon Trek Telescope is an application that allows you to submit your own images of the moon and recieve useful information within the field of view , such as annotations , layering , etc.

Begin by selecting your image of the moon on the left . For best registration results consider :

- Images of the moon with at least 30% of moon
- Clear , non blurry images
- Moon images (This can be from a telescope , camera , etc. as long as it meets above criteria)

Dont have images of the moon ? Go to the end of the guide for additional content...

Guide

YOUR UPLOAD

User image: Choose File 003.jpg

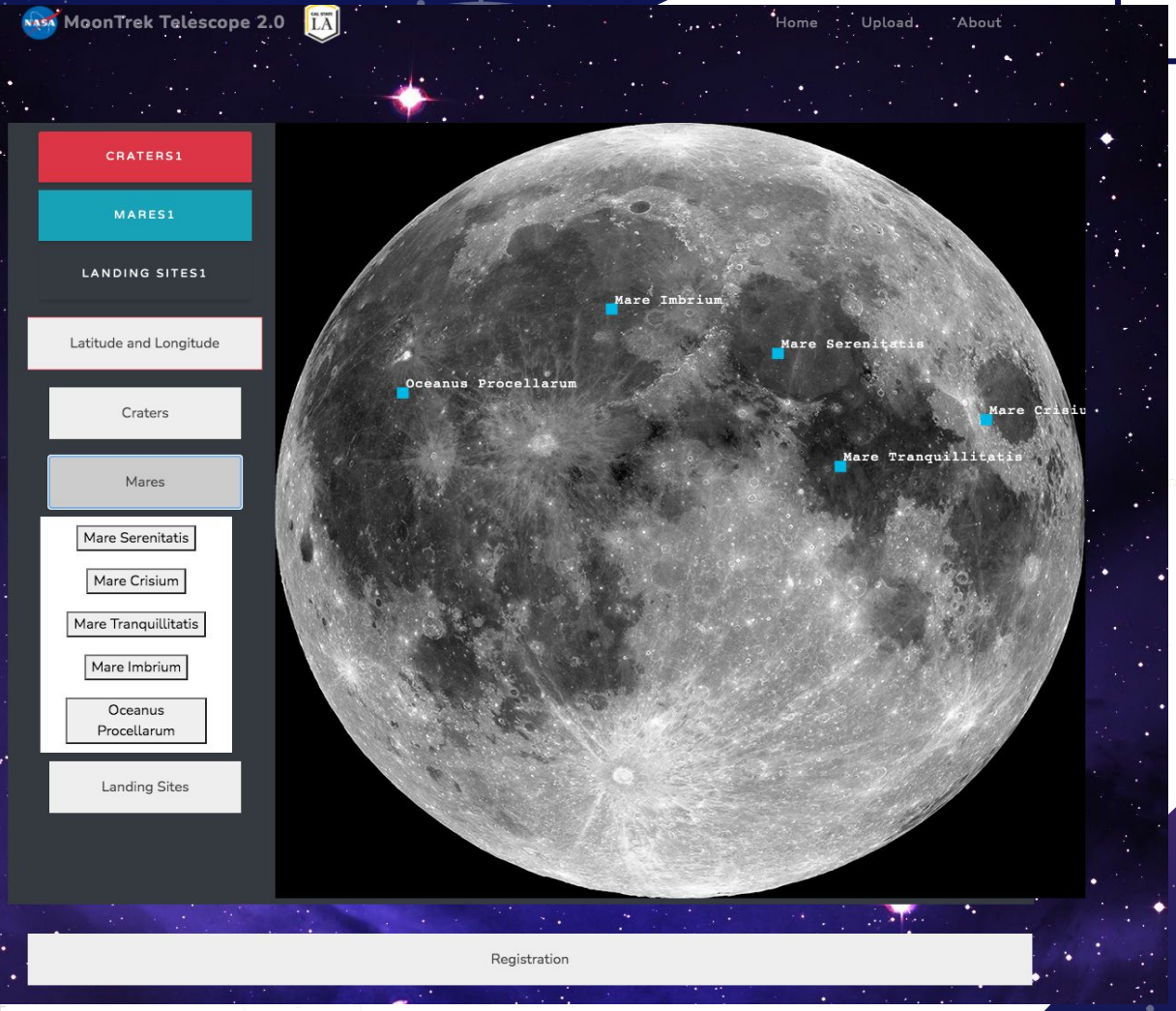
No file chosen

UPLOAD

METADATA

EXIF Data : none

Albert



- Craters
- Landing Sites
- Mares
- Easy to Read Info

- Annotated

Challenges/Solutions | User Interface

- Working with static images
- Simplified version
- Oriented towards a more casual user
- Moving away from a cluttered and complicated interface
- More interactive iteration of MoonTrek


demo