Planetary Surface Flyover Movie Generator

Team Members: Chris Omlor, Shawn Anderson, Angel Jimenez
Hieu Phan, Khang Lam
Faculty Advisor: Dr. Kang, Richard Cross
JPL Liaisons: George Chang, Emily Law, Shan Malhotra
Department of Computer Science
College of Engineering, Computer Science, and Technology
California State University, Los Angeles

Objective

Develop an application that generates a user-defined flyover movie from the planetary Trek portals (http://marstrek.jpl.nasa.gov, http://vestatrek.jpl.nasa.gov). Application allows users to specify a designated camera path. Each point of camera path contains information for camera position(latitude, longitude, elevation), camera orientation(roll, pitch, yaw), camera viewing angle, and data layer being displayed. Application allows user to interactively add information through a web interface and generates a movie by rendering and animating scenes from the camera's perspective.
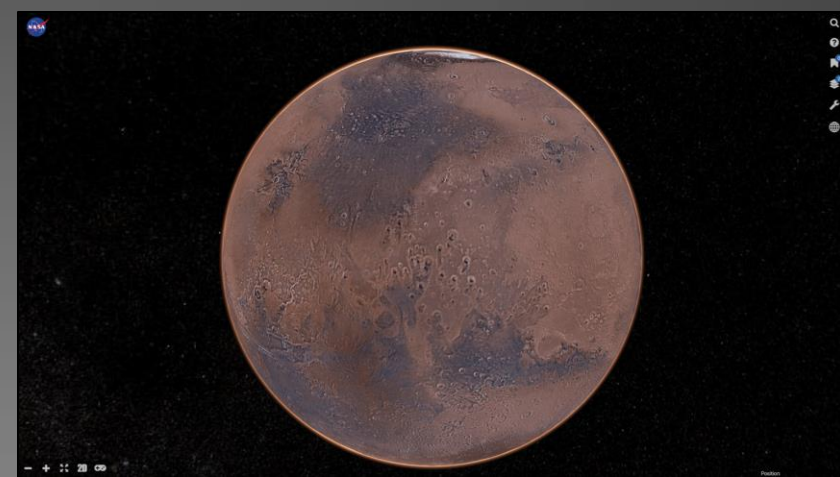
System Overview

The user can interact with the user interface that we have developed using Cesium, a javascript library that is also used for marstrek. The user interface can generate a JSON file by using Interactive mode which allows the user create the camera path or Batch Mode which lets the user upload a JSON file. The JSON file can either be downloaded by the user to be used at a later time or sent to the server to generate a movie. When Preview or Create Movie is clicked, the sun data will be requested from the server to be included in the file. Then the JSON is sent over as a JSON array through a post request to the server with point information, sun data, and user information. When the server receives the JSON file over post, the server will parse out the user's email and also create a unique ID corresponding to this specific job and stores both the ID and the user's email in a database. After Job Manager stores the data in the database it will create a subprocess and pass the video ID as well as the json array to the job program driver. Preprocessing creates and splits the scene from parsed JSON data. Split Jobs divides the work into rendering tasks using Blender. Finally Combine Scene joins the scenes in the movie that will be delivered to the user by downloadable link sent to their email.
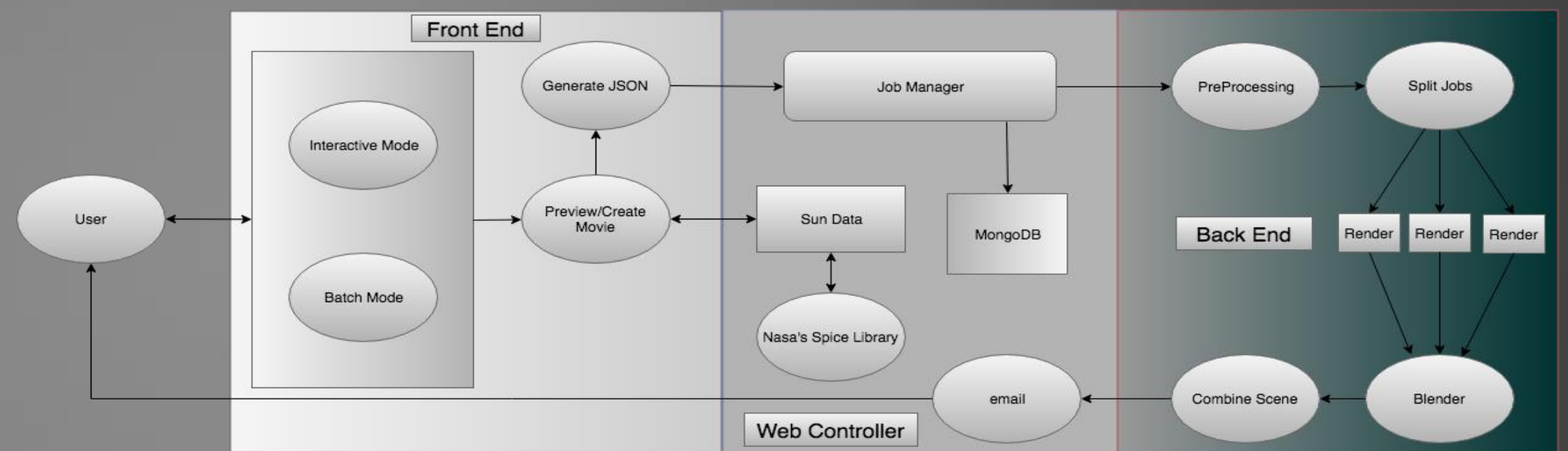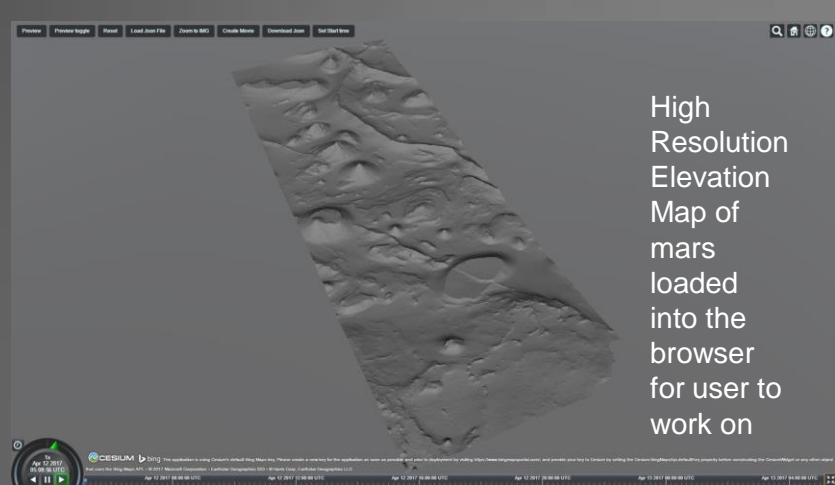

Font-end: User Interface for path selection

Augmenting Mars Trek
- Accept the user's camera path via 3D User Interface implemented using Cesium
- Allow easy integration with JPL's Mars Trek site

High Resolution Elevation Map of mars loaded into the browser for user to work on

3D Model Display
- 3D model of mars was only a small section, orientation and rotation of the terrain was a challenge to make it match the planetocentric coordinates of the globe.

Path Selection
- Users can create a polyline of editable points for a camera path
- Points are stored in a Linked List
- JSON is generated to be downloaded or sent to the back end to create movie.

View of User Interface with blue camera path and edit box

First person view
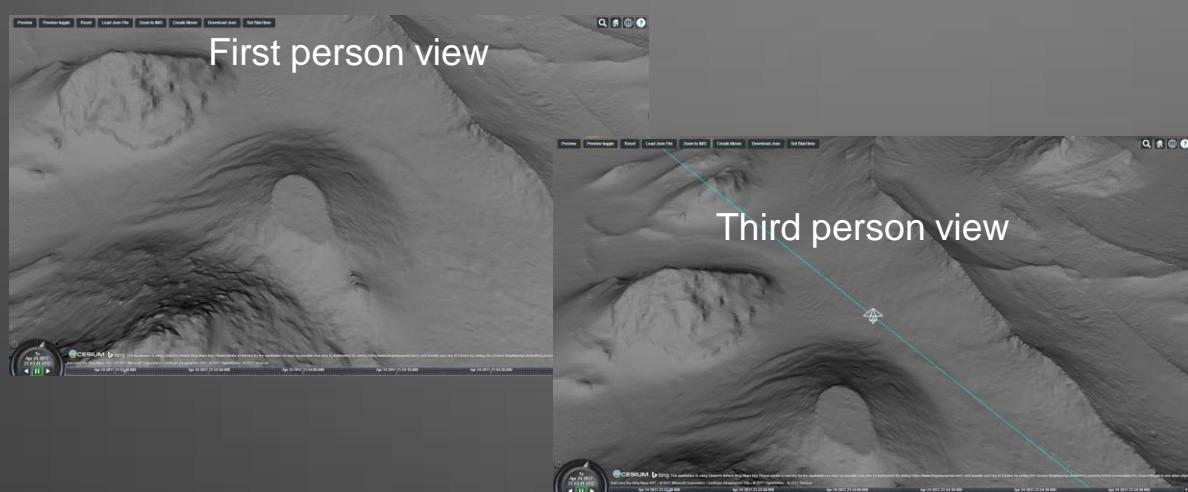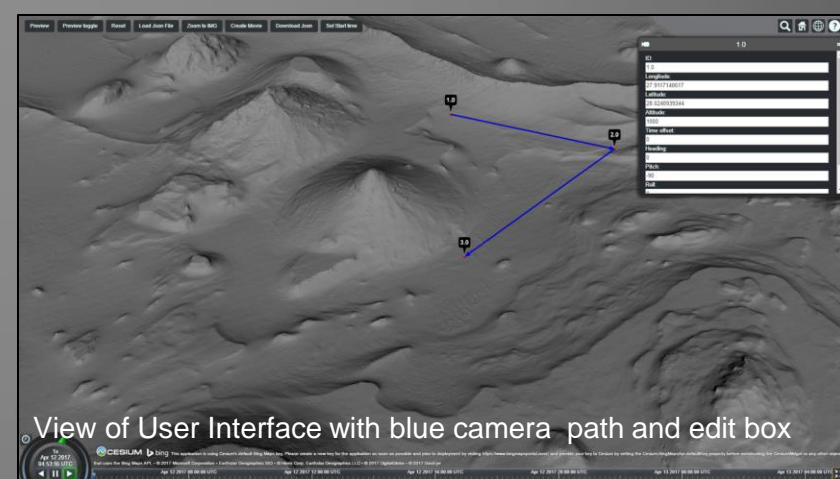
Third person view

Preview Movie
- 2 view modes: First person & Third person views
- Render a quick overview of the targeted movie in low resolution before confirmation
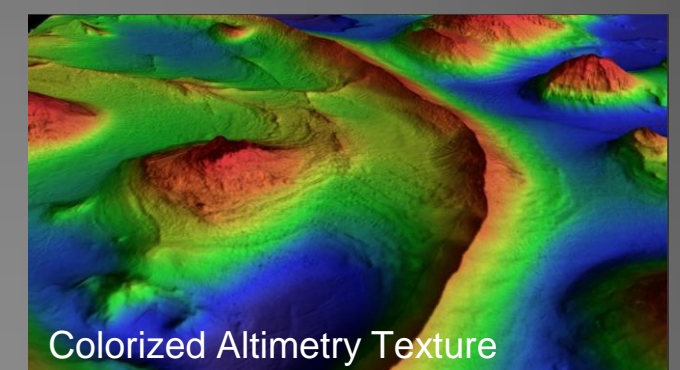
Back-end: Movie generator using Blender instances

Scene Creation
- Automated scene creation using the Blender API and Python
- Camera and Lighting placement and orientation
- Scene Optimization Preprocessing for quality and speed

Shaded Relief Texture
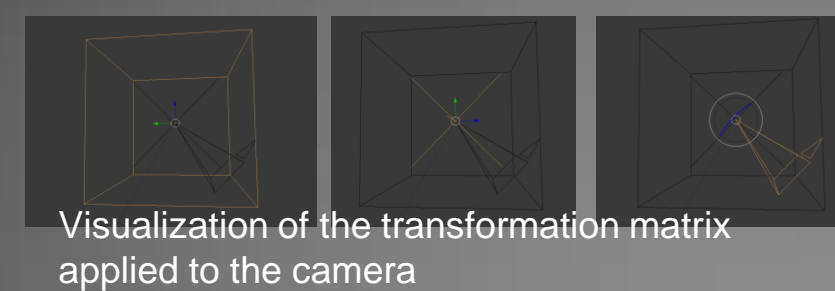
Visualization of the transformation matrix applied to the camera

Colorized Altimetry Texture

Efficient Rendering
- Distributed Rendering: Split rendering jobs using Hadoop map-reduce. Preprocess each render job to include minimal required surface area per process
- In each job, reduce the polygon count on unused terrain
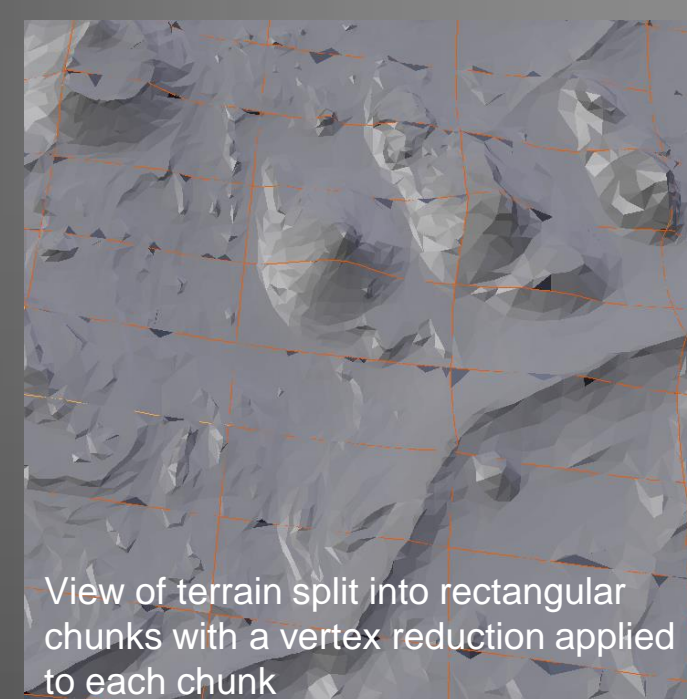
View of terrain split into rectangular chunks with a vertex reduction applied to each chunk
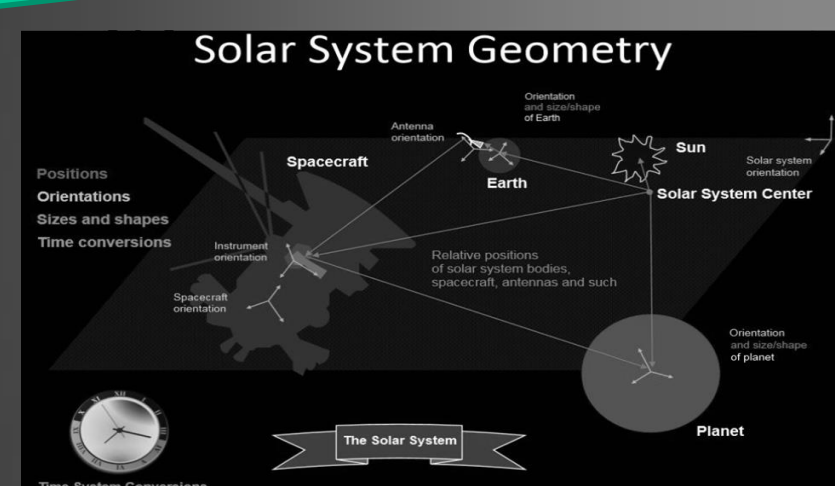
Challenges
- Handling multiple coordinate systems
- Large data size (each mesh size is about 500 MB)
- Creating scientific lighting
- Mesh/scene splitting
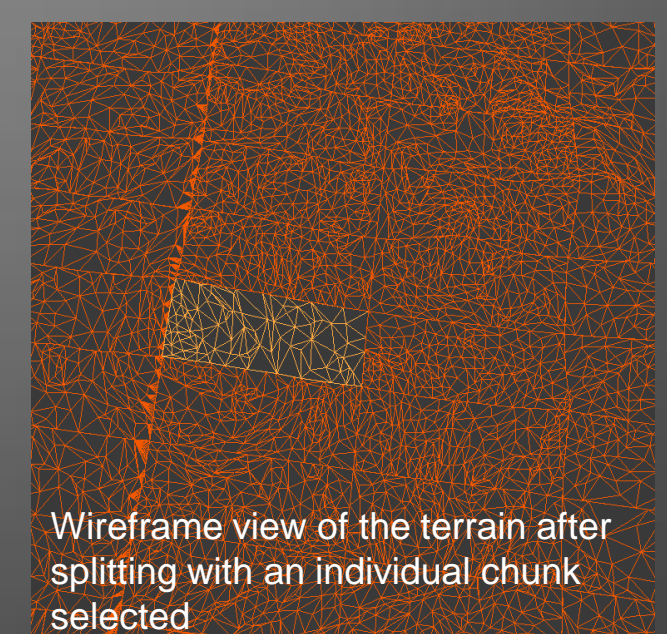
Wireframe view of the terrain after splitting with an individual chunk selected

Web Controller

Solar System Geometry

Sun Information using Spiceypy
- Sun angles and sun vector are scientifically calculated for each unique point the user creates at a given point in time.

Conclusion

- Designed a system architecture to efficiently generate a movie from a user defined camera path and real Mars DEM (Digital Elevation Map) files
- Created a Cesium-based UI for User to create a path and integrated with the existing Mars Trek
- Give user a preview of path generated
- Get accurate sun data on each point
- Created a Server that is able to listen to the UI when ready to start the sub-process
- Fully Automated System generating a 3D scene that renders a movie file from parsed JSON file
- Delivered a rendered video as output to the user

Future Enhancements
- Implementing a more intuitive way to change altitude
- Path creation from a polynomial expression
- Further improvements on render times