United States Geological Survey

# Interactive Point Visualization

# Design Document

**Prepared by:** Erin Bailey, Curtis Bilbrey, Alex Farmer, Tim Velgos
**Prepared for:** Dr. James Dean Palmer
**Date Submitted:** February 5th, 2015

**Project Sponsor:** Jay Laura
**Faculty Mentor:** Dr. James Palmer

**Last Updated:** February 4th, 2015
**Creation Date:** December 3rd, 2014

## Table of Contents

# 1. INTRODUCTION

The purpose of this design document is to specify the architecture and design of the Geographical PointViz web application in a clear and consistent manner.
The architectural design outlined in this document builds upon the requirements listed in the requirements document.

The purpose behind the creation of the PointViz web application is because currently, USGS has no tools in-house or out to be able to efficiently analyze the sheer amounts of multi-dimensional data that is taken from the KSP.

Throughout this document we will be discussing the general concept of the Geographical PointViz web application and its purposes in the Systems Overview section. The next big step is to understand the general architecture including the design, description, and rationale, which we go into depth in the System Architecture section. From there, we continue on and move into the discussion of use of data in the creation of the web application as well as all the files, the number of records, the names, and each type of field. This information may be found in the Data Design section. We will conclude the document with a discussion on what our user interface will look like, as well as the functionality it will provide in the Human Design Interface section.

Please see Appendix A at the end of this document for questions regarding definitions, unfamiliar terms, or acronyms.

# 2. SYSTEM OVERVIEW

The Geographical PointViz web application is a project designed to interpret, store, and visualize geospatial data acquired from the Kaguya Spectral Profiler. The Kaguya Spectral Profiler takes images of the surface of the moon, and collects data on attributes such as reflectivity and radiance over a spectrum of wavelengths. The purpose of this project is to extract information from the images and store the data within a database, which may be queried by a web application to depict the information.

The end goal of the project is to allow all interested users to receive and visualize the findings from the Kaguya Spectral Profiler using the web application, and to be able to easily interpret and adjust the data that they see. The application will also be scalable to allow for additional satellites to contribute attribute data that can be tied to existing data connected by latitude/longitude coordinates.

# 3. SYSTEM ARCHITECTURE

In this section we discuss the general architecture of the Geographical PointViz web application including the design, description, and rationale.

## 3.1 Architectural Design

The Geographical PointViz application is comprised of four major components. The first component is the Kaguya Spectral Profiler, which creates the data as observed from the surface of the moon. The second component is the data that is then run through a python script on a USGS server that processes the raw data, and returns point and attribute values. This data is then entered into the third component, the database, which runs in a MongoDB environment to store data as documents inside of collections. The final component is the web interface, which displays specific data to the user upon request. These components are connected by queries that link the interface to the database, and the JSON that is returned to the interface.

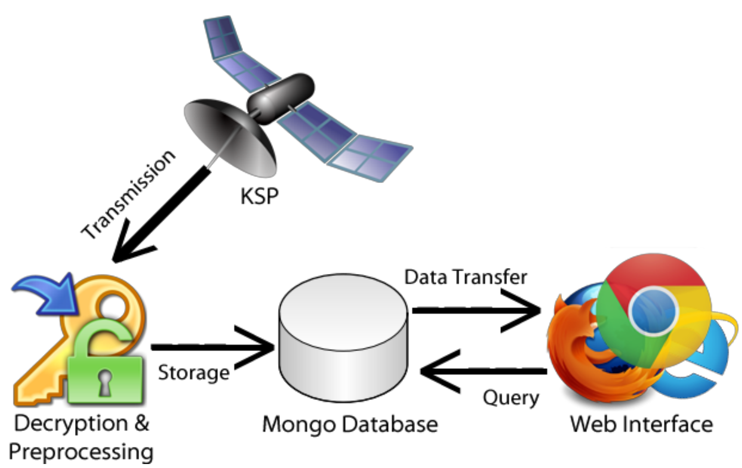


*Fig 3.1:Topology Diagram*

### 3.2 Design Rationale

For this project, are using Python, MongoDB, and the D3 JavaScript library as our main technologies. After hearing about the type of coordinate data that our sponsor wanted to use, and the attributes that we wanted associated with the data, we agreed that MongoDB would be the appropriate type of database to use due to the structure of its collections and associations. We then decided to use python, as recommended by our sponsor, with our web application to interface with the D3 framework that could provide graphic visualization for our interface.

We decided that for our specific purpose of large data storage, retrieval, and visualization, that the combination of MongoDB, Python, and D3 would be the ideal way to create our web application. We considered the power and visualization capabilities of ArcGIS, however in the end we discovered that it could not process the volume of data we wish to work with in an efficient time, and would not be considered to be user-friendly.

The Python script could be substituted for a different coding language, however it is both the preferred programming language of our sponsor, and it has support for interacting with MongoDB. With our sponsor preferring Python, it will allow them to expand the project in the future to best suit USGS.

MongoDB was decided as the best database for our needs as opposed to a SQL-oriented database, as the NoSQL functionality of MongoDB allows for the allocation and storage of our vast amount of data.

D3 is a relatively accessible JavaScript library that gives us graphing capabilities that work best with our point data. Other alternatives proved to react more poorly when given larger amounts of data to plot.

# 4. DATA DESIGN

The following diagram is a depiction of the data organizational structure. Within the database, there are many collections; One for each image. Within each collection, there are documents.

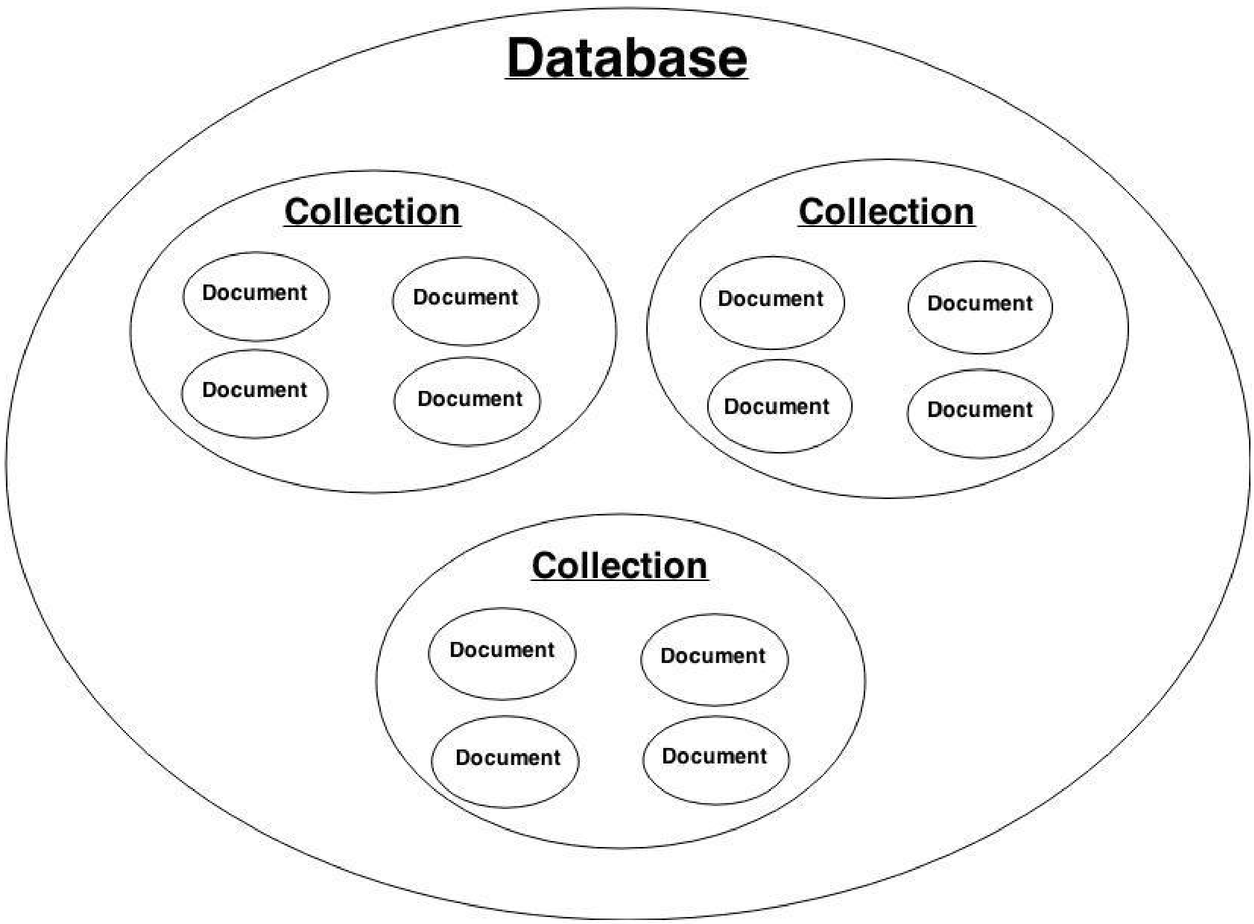


*Fig 4.1:Mongo Database Overview*

There are four document types. Point documents, there are one of these for every point associated with the image. A wavelength document, there is one for each image. A header document, there is one for each image. Also, an angles document, there is one for each image. Further descriptions of what is contained in each of them can be found in the section below.

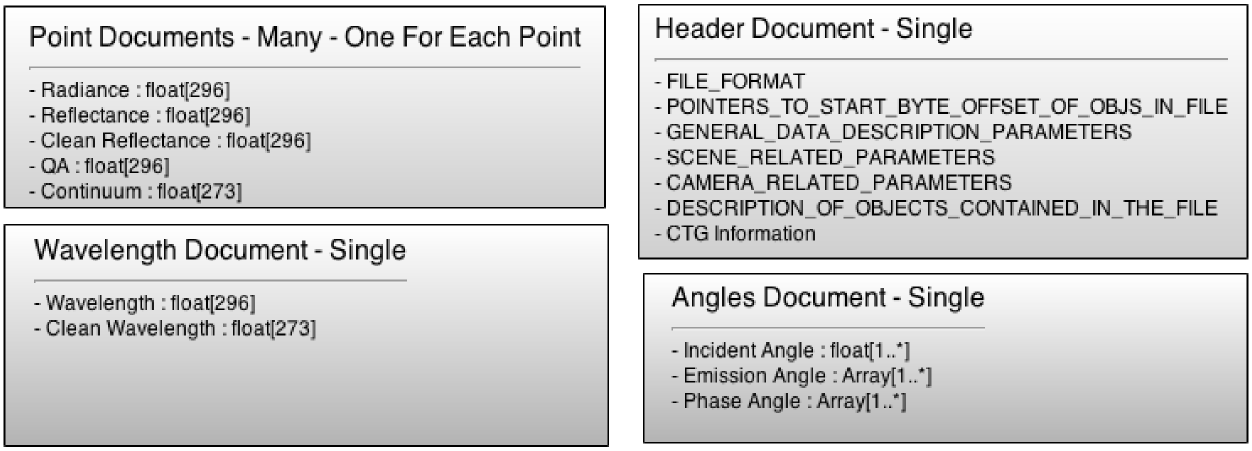


*Fig 4.1:UML Description of Documents Contained in Each Collection*

### 4.1 Data Description

The following descriptions discuss the meaning of the data in regards to how they are stored in the database.

**4.1.1** In MongoDB a database is comprised of many collections, where a collection can hold many JSON documents.

**4.1.2** The database represents each image as a collection.

- Each collection contains a wavelength document that represents a sample size of a continuous wavelength.
  - Each wavelength document contains a wavelength array that contains 296 floats that each represent a continuous wavelength in ascending order.
  - Each wavelength document contains a clean wavelength array that contains 273 floats that each represent a clean wavelength in ascending order.
- Each collection contains a document for each point.
  - Each point contains a radiance array which contains 296 floats, where each float represents the radiance level at the corresponding wavelength.
  - Each point contains a reflectance array which contains 296 floats, where each float represents the reflectance level at the corresponding wavelength.
  - Each point contains a clean reflectance array which contains 296 floats, where each float represents the clean reflectance at the corresponding clean wavelength.
  - Each point contains a quality assurance array, which contains 296 floats that correspond to both reflectance and radiance.

- Each point contains a continuum corrected reflectance array which contains 273 floats that correspond to the clean reflectance.
- Each collection contains an angles document, where the combination of the angles define that position of the KSP in relation to the moon when the image was taken.
  - Each angles document contains a list of phase angles where each angle is associated with a point.
  - Each angles document contains a list of emission angles where each angle is associated with a point.
  - Each angles document contains a list of incidence angles where each angle is associated with a point.
- Each collection contains a header document
  - Each header document contains the header information associated with each image.

# 5. HUMAN INTERFACE DESIGN

The interface used in the Geographical PointViz is represented as a web application that is interactive for the user. It is intended for researchers and scientists to be able to efficiently analyze the surface of the moon based on certain attributes. The user will be able to interact with a series of pictures, clickable points, sliders, and brush components.

## 5.1 Screen Images

Below is a mockup of what we intend our user interface to look like.

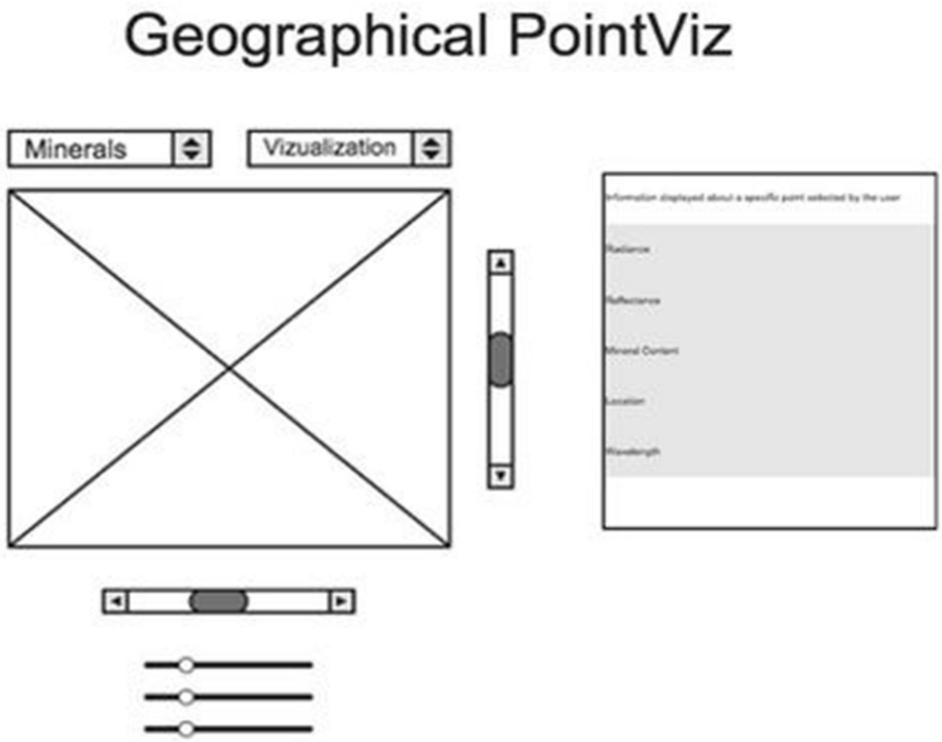


*Fig 5.1:User Interface Mockup*

**5.2 Functionality**

The function of this user interface is for users to interact with a map of the moon as well as individual points that are mapped to latitude/longitude coordinates based on the following functionality components.

**5.2.1 Viewing an Image**

The following features assist the users in viewing the point data.

- Select: Certain points on a map of points must be selectable in order to get a more detailed description of what the user is viewing.
- Associated Attributes: Selecting a point must create a display of all attributes associated with that point as well as a line graph depicting the wavelength associated with that point.
- Layer: The user must be able to select a specific layer on a heat map in order to get a more detailed description of what the user is seeing.
  - Layer Attributes: Selecting a layer must create a display of all attributes associated with that layer.
  - Specific Point Selection: The user must be able to select a specific point in the desired map to get a more detailed description of what the user is looking at.

- ○ Attribute Display: Selecting a point must pop out a display of all attributes associated with that point, as well as a line graph depicting the wavelength associated with that point.

**5.3 Adjust Settings**

These actions may be performed by the user to change the viewing area of the displayed data.

**5.3.1 Zoom In**: The user may zoom in on the current map they are viewing by selecting the button on the left hand side of the image.

**5.3.2 Zoom Out**: The user may zoom-out of the current map they are viewing by selecting the left hand side of the image.

**5.3.3 Panning**: The user must have the ability to pan through the image while it is zoomed in on. The user must also be able to hold a click and drag the image in the direction they wish to pan.

**5.4 Query Data**

These tools allow the user to issue the query commands to the database with given specifications set through menus and options.

**5.4.1 Drop-Down Menus**: The interface must contain two drop-down menus: the Mineral drop-down menu and the Visualization drop-down menu.

- Mineral Drop-Down Menu: The interface must contain a drop-down menu, where a user can select a mineral to query for. The user must then add that mineral to a list with an “add” button. The user can add as many minerals to this list as they wish.
- Visualization Drop-Down Menu: The interface must contain a second drop-down menu that allows the user to choose what kind of visualization they wish to see, including heat maps and bivariate hexbin maps.

**5.4.2 Brush Components**: The interface must contain three brush components.

- Wavelength Brush Component: The interface must contain a range slider for the wavelength, so that the user may define a range of wavelengths that they wish to query.
- Radiance Brush Component: The interface must contain a range slider for radiance where the user may define a range of radiance that they wish to query.
- Reflectivity Brush Component: The interface must contain a range slider fo reflectivity where the user may define a range of reflectivity that they wish to query.

**5.4.3 Brush Ranges**: The user must be able to define ranges for however many brushes they wish. For brushes not given ranges by the user, the entire range will be queried.

**5.4.4 Brush Warning**: If a range is not defined for any of the brushes, warning text will appear underneath the brushes they have selected. For brushes not given defined ranges by the user, the entire range will be queried.

# 6. CONCLUSION

The design and architectural decisions made for the Geographical PointViz web application have been based upon the needs of the user to have timely access to the data from the KSP in an efficient and uncomplicated manner. Through the components and technologies discussed earlier in the document, our web application will provide the much-needed link between the information gathered by the KSP and the researchers, scientists, and users who may then use the data as they wish.

As of right now, we have a solid foundation for the backend dealing with the database. As far as the front end of the user interface goes, we have a working prototype in D3 that takes in the arrays of attributes and maps them to coordinates. The user interface currently has the functionality to zoom in and out, pan, and interact with the points on the map. Moving forward we hope to connect the database to the web interface so that users can query from the application to the database. We will conduct user testing and refactor and optimize the project as a whole.

# APPENDIX A: ACRONYMS, ABBREVIATIONS, AND DEFINITIONS

| Term | Definition |
|---|---|
| JSON | JavaScript Object Notation |
| Radiance | The flux of radiation emitted per unit solid angle in a given direction by a unit area of a source. |
| Reflectivity | The property of reflecting light or radiation, especially |

| | |
|---|---|
| | reflectance as measured independently of the thickness of a material.. |
| Phase, Emission, Incidence Angles | Together these angles describe the position of satellite in relation to the object it's orbiting. |
| Spectrum | An array of entities, as light waves or particles,ordered in accordance with the magnitudes of common physical property, as wavelength or mass. |
| USGS | United States Geological Survey |
| Wavelength | The distance between successive crests of a wave, especially points in a sound wave or electromagnetic wave. |
| Clean data | Data that has been post-processed through mathematical adjustment functions given by USGS |
| Dirty data | Data that has had no adjustment, and has been taken directly from the KSP |

## APPENDIX B: Products and Tools

| Software/Tool | Version | Source | Description |
|---|---|---|---|
| MongoDB | 2.6.5 | http://www.mongodb.org/ | A document oriented NoSQL database. |
| D3 | 3.4.13 | http://d3js.org/ | A Javascript library used for manipulating documents based on data. |
| Python | 2.7.8 | https://www.python.org/ | An object oriented programming language. |
| JavaScript | 1.8.5 | N/A | |
| JSON | N/A | http://www.json.org/ | Javascript Object Notation is a lightweight data-interchange format. |