United States Geological Survey

Interactive Point Visualization

Requirement Specification Document

Prepared by:	Erin Bailey, Curtis Bilbrey, Alex Farmer, Tim Velgos
Prepared for:	**Dr. Omar Badreddin**
Date Submitted:	November 25th, 2014
Project Sponsor:	Jay Laura
Faculty Mentor:	Dr. James Palmer

Last Updated:	**November 25th, 2014**
Creation Date:	**October 30th, 2014**

Table of Contents

1. Executive Summary

The Interactive Point Visualization (Geographical PointViz) project provides the ability to store large amounts of data from the Kaguya Spectral Profiler in raw binary form, and convert it into usable latitude/longitude coordinates along with related attributes. Once stored, the project will then be able to display a configurable range of the data for a user to use, which will be adjusted in the web application for each user.

The storage of data in this project must be dynamic because of the fact that different coordinate pairs possess different amounts of attribute data. Continuous visible and near infrared reflectance spectra will be used in order to determine the type and breakdown of minerals on the lunar surface. With that in mind, a traditional SQL database would not allow for seamless additions of attributes because of the data limits, therefore MongoDB has been selected with its NoSQL structure in mind.

The purpose of this project is to store and visualize the large amount of raw geological data that USGS has acquired from the Kaguya Spectral Profiler, so that scientists and researchers may draw gain knowledge and draw conclusions based on the data that is provided to them. This data corresponds to readings taken from the surface of the moon that focus on analyzing reflectivity along different measured wavelengths.

The web interface for the project will allow users to understand the data that has been acquired, by representing it with visual mapping technology, or by exporting the data fitting the range designated by the user.

Completion of this project will require a backend composed of a databse to interact with and store all of the coordinate data and additional attributes. The front end will be created using visualization tools hosted as an interactive web application.

Storing and visualizing data are the main concerns of this project, and the limits of our project scope. Once fully developed, the project will support the storage, query, and visualization of the spatially enabled data that is provided by the United States Geological Survey.

2. Introduction

This document is designed to specify the functional and non-functional requirements of the Geographical PointViz web application in a clear and consistent manner.

2.1. Document Purpose

This document serves as a detailed description of the requirements related to the Geographical PointViz web application as agreed upon by the developers and the customers. This document is not intended to be a general help and troubleshooting document, however the explanation of each feature and option may assist the user in using the application.

This document may be updated in the future to accommodate additional updates in order to continue maintaining the current prototype.

2.2. Intended Audience

The stakeholder(s) of this document is any person, group, or organization that has an interest or concern in the Geographical PointViz web application requirements. Specifically, this document is designed for the project team and functional groups that must determine how to implement the application in order to adhere to the customer's requirements.

2.3. Project Background

The Kaguya Spectral Profiler is a spectrometer that obtains continuous reflectance spectra in order to analyze minerals and their location on the surface of the Moon. The data is obtained at a high spectral resolution as well as a high spatial resolution. The Kaguya Spectral Profiler is the first device to conduct continuous spectral observations of the Moon in the visible to near infrared region.

Until the Kaguya Spectral Profiler, observations with historical orbiters have been limited in spectral coverage, with detailed information restricted only to lunar surface mineral composition. The Kaguya Spectral Profiler performs global spectroscopic mapping of the Moon, which is necessary to determine the mineral content of the lunar surface.

The Geographical PointViz web application must serve as an interactive interface between the user and the points that are obtained from the Kaguya Spectral Profiler. The goal of the web application is to visualize the exploration of data analysis that is an essential component of a large Big Data analysis framework.

2.4. Business Goals/Objectives to be achieved

The goals of this document are to outline the overall objectives for the development of The Geographical PointViz web application

2.4.1: The Geographical PointViz web application must be considered a reliable tool that USGS may use for analyzing data gathered from the Kaguya Spectral Profiler. USGS currently does not have a tool to efficiently analyze this data. This application will allow them to further understand and share the data that they have been collecting.

2.4.2: The Geographical PointViz web application must take the raw data in binary format from the Kaguya Spectral Profiler and display it in a format that users may understand and use. This is completed through a Python script that has been provided by USGS.

2.4.3: The completed Geographical PointViz web application will be an interactive web interface that allows the user to choose the amount of data, and which attributes they wish to view, which will be supported by the MongoDB back-end.

2.4.4: The Geographical PointViz web application will be constructed in Python and will pull data from a MongoDB environment so that USGS may easily maintain and update the project once it is finished.

2.4.5: Researchers, scholars, and scientists will be able to further their research based on the data acquired from the Kaguya Spectral Profiler.

2.4.6: The Geographical PointViz web application will open up the possibility for other satellites or profilers to contribute attribute data that may be linked shared geological data coordinates.

2.5. Benefits/Rationale

The main purpose of our project, and the largest benefit of our project, is to provide a prototype for USGS as a tool for sifting through large amounts of geographical data and visualizing it to users. USGS does not currently have a tool that fulfills this purpose, and this prototype will help assist in further development and exploration using their large quantities of data.

Another benefit of our project is the use of Python and MongoDB. USGS supports and encourages Python as a preferred coding language, and MongoDB as a database environment. Constructing our web application using these two technologies allow USGS to continue where we leave off for further development.

Our web application will serve as a prototype in order to encourage future data sharing between other telescopes and spectral profilers.

2.6. Stakeholders

The primary stakeholder for our project is USGS, more specifically our sponsor Jay Laura. The other stakeholder would be our C.S. department, and Dr. Palmer for his interest in our project as our faculty mentor.

Both Jay and Dr. Palmer have been crucial to the developmental process, and both have been frequently referenced in the process.

2.7. Dependencies on existing systems

The project depends on MongoDB, Python compilation, and the data received from the Kaguya Spectral Profiler. For this project, our team will be provided with uninterrupted access to MongoDB, Python compilation, and data from the Kaguya Spectral Profiler.

2.8. Description of Development Methodologies

The Geographical PointViz development process utilizes an Agile SCRUM-based methodology. This development process consists of weekly sprints hosted by our faculty mentor and scrum master, Dr. Palmer, where the progress of each week is discussed and suggestions may be made. The development process also includes a text-based backlog where the developers and mentor may update tasks and separate completed tasks from tasks in progress.

3. Requirements Scope

The scope of this document includes the deliverables as well as defining boundaries for the development of the prototype.

3.1. In Scope

Storing data in the MongoDB implementation is half of the scope for this project.

The second half of the project scope is taking the data and creating a visual representation for the amount of data specified.

3.2. Out of Scope

Beyond the scope of our project, is the process of how the profiler is gathering the data from the Moon. The accuracy of this data is not questioned for the scope of the project.

We also consider any research or work completed using the results of the visualization to be beyond the scope of the project. The project will return the data as it is stored in the database, and cannot be held responsible for inaccurate results in research having gained information from the Geographical PointViz web application.

4. Functional Requirements

This document provides a list of the functions and uses of the Geographical PointViz web application and its components.

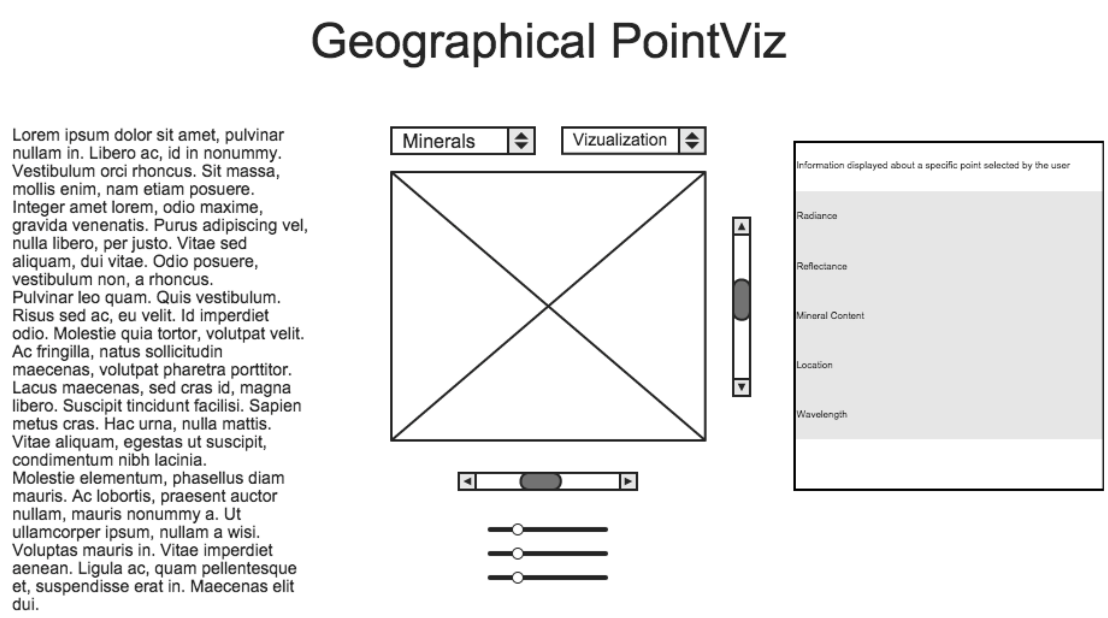

Figure 1.1

4.1. Adjust Settings

4.1.1. 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.

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

4.1.3. 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.

4.2. Viewing an Image

4.2.1. 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.

4.2.1.1. 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.

4.2.2. 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.

4.2.2.1. Layer Attributes: Selecting a layer must create a display of all attributes associated with that layer.

4.2.3. Specific Hexagon Selection: The user must be able to select aspecific hexagon in a bivariate hexbin map to get a more detailed description of what the user is looking at.

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

4.3. Query Data

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

4.3.1.1. 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.

4.3.1.2.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.

4.3.2.Brush Components: The interface must contain three brush components.

4.3.2.1.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.

4.3.2.2.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.

4.3.2.3.Reflectivity Brush Component: The interface must contain a range slider for reflectivity where the user may define a range of reflectivity that they wish to query.

4.3.3.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.

4.3.4.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.

5. Non-Functional requirements

5.1.Scalability

Defines the ability at which the Geographical PointViz web application can gracefully meet the demand of stress caused by increased use.

5.1.1.MongoDB: This product must be build with MongoDB so that it may be maintained and expanded once the original developers have finished development.

5.1.2.Python: This product must be built using the Python programming language so that new developers that resume this project will be able to continue development once the original developers have left.

5.1.3.Database Schema: The schema of the database must be built such that similar data from other satellites may be added without requiring the schema to be altered.

5.1.4.Palette Functionality: The product must be built with palette functionality so that a user may add or remove palettes to adjust the query criteria.

5.1.4.1.Example: An example of this could be the reflectivity palette which would only display attributes relating to reflectivity.

5.1.5.Palette Implementation: Palettes should be easy to implement and separate from the rest of the system such that if a new palette needs to be added in the future, the user is not required to edit core functionality.

5.2.Usability

Defines how efficiently the user can use, learn, or control the web application.

5.2.1.Reduction of Visual Clutter

5.2.1.1.Filtering: This product must utilize filtering of point data in a way so that a minimal data set is used to represent all point data for a given image.

5.2.1.2.Simple: This must be achieved so the product may be as visually simple as possible.

5.2.1.3.Split: The data will be made visually simple by a filtering process where the data is split into a grid where only subsets of data within each grid component are used to visualize data.

5.2.2.User Interface and Human Factors

5.2.2.1.Intended Users: The intended users are researchers and scientists, or others with a scientific background to understand the processed data.

5.2.2.2.User Training: The interface should be intuitive and easy to understand, and no additional resources should be needed beside the help document.

5.2.2.3.Browser Support: The PointViz web application will promise support for IE9, usability with other browsers is not assured.

6. Conclusion

As we continue the implementation of this project, we come closer to completing a full deliverable of everything that is specified by our final requirements document. Some of the concluding details are still in progress, and we plan to finalize those with our sponsor, as well as our faculty mentor, before next semester.

One of the issues we faced during our implementation was a conflict in communication between ourselves and our sponsor's requirements, versus our explanation of these requirements to our faculty mentor. We've since then resolved these issues, however, it did push back our actual implementation quite a bit because we were concerned that we were not working towards the correct solutions.

However, we have had a lot of success while working on this project. We are utilizing more accurate documentation, explanatory processes, and prototype mock ups in order to make sure that communication issues will no longer inhibit our ability to move forward with our project. We've taken advantage of SCRUM techniques, such as backlogging our progress, as well as addressing impediments.

As we continue to move forward with our project, we expect to continue dealing with both successes and setbacks. So far we have learned that communication is a key component in working with a client, and is necessary for obtaining the oversight of a faculty mentor. We are also gaining new skills with MongoDB with every sprint we complete.