Requirements Specification

03/05/2022

Team BioSphere

Project Sponsors: Jenna M. Keany, Dr. Christopher Doughty
Team Mentor: Melissa D. Rose
Team Members: Matthew Nemmer, Brandon Warman, Teng Ao, D'Yanni Bigham

Version 3.0

By signing below, the clients of the Biomapper project hereby agree that the requirements listed in this document are (a) sufficient and cover all desired functionality, (b) are the only responsibilities of team BioSphere, and (c) are final unless a new version of the document is created and agreed upon.

Table of Contents

Introduction

The loss of tropical forests is a cause for global concern given that they play a vital role in large-scale environmental processes. By absorbing vast quantities of CO2, tropical forests provide oxygen and help stabilize the Earth's climate. Furthermore, tropical forests help to maintain the world's water cycle through transpiration, generating clouds that travel all over the world. As for biodiversity, "tropical forests contain over 30 million species of plants and animals... half of Earth's wildlife and at least two-thirds of its plant species!" [1]. These facts make it clear that tropical forests need to be preserved and protected.

The importance of environmental protection cannot be overstated. It is crucial that forest ecologists have access to and are able to interpret LIDAR-derived data products such as digital elevation models (DEM) and canopy height models (CHM). DEMs are a representation of the bare ground topographic surface model of the Earth and CHMs are a measurement of the height of trees, constructions, and other structures above the ground topography. DEM and CHM values are important for determining where specific species of plants and animals may be found and studied. The results of these studies, along with values of above ground biomass (AGB), help determine the environmental importance of the forest. AGB data provides an estimate of the carbon content of each area within a forest. Additionally, conclusions drawn by the ecologists allow policymakers and conservationists to better understand environmental changes and enact legislation to protect forests.

The aforementioned datasets can be obtained from LIDAR-equipped satellites, such as the International Space Station with its onboard GEDI sensor. However, processing this data and visualizing it is a non-trivial task. What forest ecologists need is an easy way to access the processed environmental data in real-time while conducting their fieldwork. Such a tool is what Dr. Christopher Doughty, the client and sponsor of this project, has tasked the team to create. He studies megafauna within the forests of Africa and works closely with ecologists working there. Their business is less focused on financial gains and more on learning about the environment and ensuring that it is protected. Currently, the geospatial data they use can be visualized using tools like Google Earth Engine (GEE). GEE is a web application that uses Javascript to host, edit, analyze, and visualize processes geospatial data within a web browser. However, it has its unique codebase language that is not applicable to every user and is difficult to use in the field. So forest ecologists are still in need of a simple tool for visualizing environmental data and relevant maps while in the field.

Problem Statement

As Figure 2.1 shows, the raw LIDAR data obtained from satellites must be processed and visualized so that others can make use of it. This is difficult to do as forest ecologists want the data in a simple format like a color-coded map, but the raw LIDAR data captures the environment in 3D. The waveforms that LIDAR sensors create to depict the 3D structure of the planet are complex and not readily viewable. They carry a number of datasets, including the previously mentioned CHMs, DEMs, and AGB. Programming scripts must be written that take this information and turn it into formats that are more human-readable, such as images. This undertaking requires technical knowledge that typical ecologists are unlikely to possess.

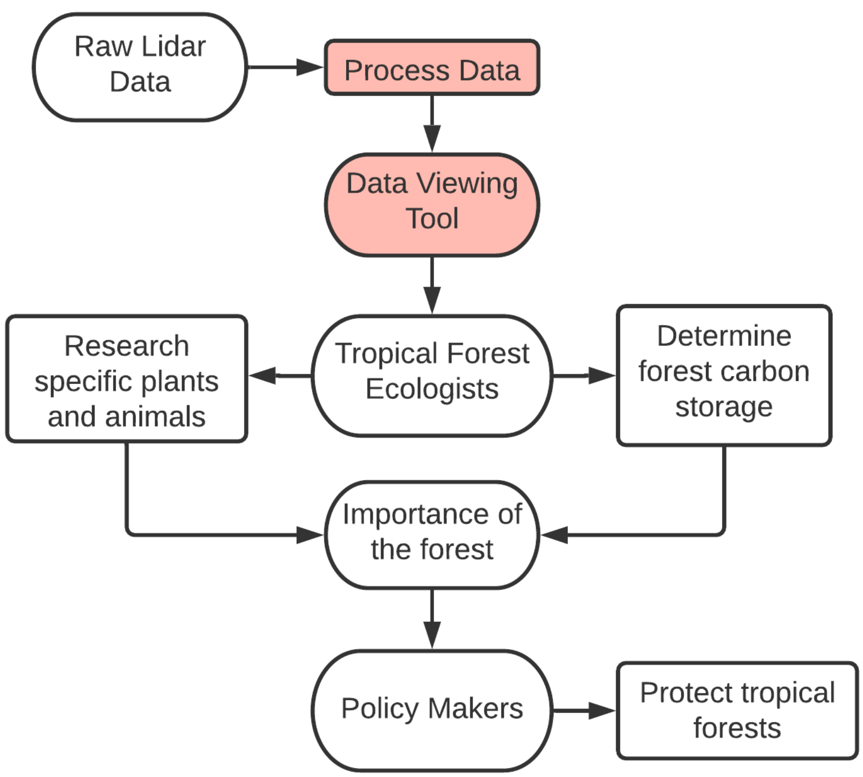

Figure 2.1

Luckily, the clients (as well as others) are capable of processing the raw data and making it available for others to use. They handle the difficult task of producing the datasets that are desperately needed to study, draw conclusions about, and protect tropical forests. This leaves the team with the task of creating a tool capable of displaying those datasets. The existing methods such as GEE or simply viewing data images are not sufficient. Ecologists need something that is more practical than GEE and provides more functionality than plain images. Ideally, it would be a tool they can take with them while conducting fieldwork.

Solution Vision

The final product will be a mobile application that is capable of displaying GEDI-derived data on a map. The application will primarily provide data such as canopy height, DEM values, and above-ground-biomass through an interactive heat map. The main features of the application will include:

- User will have the ability to scroll and zoom in on the map, allowing them to view any region of interest they desire
- User will have the ability to select the type of data to be displayed on the map
- User will be able to specify a range of values; highlighting areas within that range of input values
- User will be able to download map for offline use

In order to allow the user to find data for their region, the application will need access to their GPS. It will get the data from the server by using the coordinates as an index to retrieve the region From there the application will be able to populate the map with the data they need. Following this, the user can now scroll around the map and zoom in/out of specific areas.

If the user will be in an area that will not have an internet connection then they are able to download the data for a region beforehand. The user will be able to manually input the coordinates for a region and get the corresponding data. After that there will be an option for the user to download for offline use, if an internet connection is present.

The data populated on the map will come from a server which receives its data from a file system. Both the server and file system will be hosted on a web hosting service. The file system will contain canopy height, elevation, and above-ground-biomass for African regions.

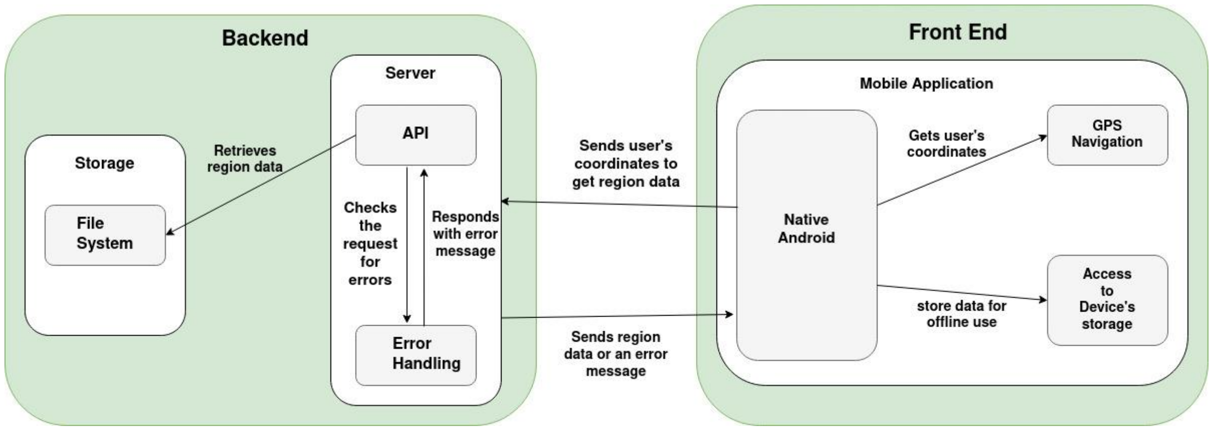

Figure 3.0: General outline of the solution

Project Requirements

Taking the team's solution vision into account, a broad list of project requirements have been synthesized. Each of these will be briefly described before expanding them out into multiple sub-requirements describing more clearly how the desired outcome will be achieved. This will result in a comprehensive list of functional, performance, and environmental requirements that cover all vital aspects of the project.

Domain-Level Requirements Overview

The mobile application will display a map that can be navigated

The most effective way to convey geographic information to the mobile app users is by implementing a map. It will use a heatmap, which is a "graphical representation of data that uses a system of color-coding to represent different values." [2]. The app users should be able to navigate the map via touch screen interactions and be able to specify the type of data they currently want to be displayed.

The mobile application will feature tools for locating areas of interest

The users of the mobile app will need a way to search for specific areas that may be interesting to them. To this end, the ability to specify a range of data values to be displayed will be implemented. Also, users will be able to touch anywhere on the map and be shown the data value associated with that location.

The mobile application will be capable of offline functionality

Since the app will likely be used in places where internet connection is not available, there needs to be a method for storing data on the device itself. It will be possible for the user to specify which area of the map will be stored locally for offline use. They will need to be able to further specify the resolution of the area downloaded. The app should inform the user if too large of a request has been made.

Text within the mobile application will be in either French or English
Many of the countries in Africa containing tropical forests include French as an official language. So, the text within the mobile app will accommodate both French and English readers. Upon loading for the first time, the user will be asked which language they prefer. After that, it will still be possible for the user to change the language within the mobile app's settings.

The mobile application will be capable of centering its map at a designated point
The mobile app will be capable of retrieving the user's location and centering the map at that point once the app is opened. Additionally, the user will be able to set a center point that can be used instead of their location. This allows the user to be elsewhere yet still be presented with the area they're interested in after opening the app.

Comprehensive Requirements List

Note that all requirements are by default functional requirements. Performance requirements are identified by a single asterisk (*) while environmental requirements are specified using double asterisks (**). A slow-motion camera or code for timing operations will be used to verify the speed related performance requirements.

The mobile application will display a map that can be navigated

- The user will be able to select the desired data to be displayed
    - canopy height
    - above-ground biomass
    - elevation
- A color-coded heat map will be used for displaying the data
    - *Stretch Goal:* Make it so the elevation data is displayed using a non-linear color palette to provide better contrast for lower values while still including high elevations
- The user will be able to scroll and zoom the map
    - As new locations are scrolled over, the corresponding map tiles will be loaded and displayed
    - As the map is zoomed in, more detailed map tiles covering less area will be loaded and displayed
    - As the map is zoomed out, less detailed map tiles that cover more area will be loaded and displayed

    - This map tile loading will be handled by the mapping software development kit utilized within the app
- *When new data is chosen or the map is scrolled or zoomed, the new data tiles will be loaded and displayed within 2 seconds
- *The map will be available for navigation within 7 seconds of opening the app
    - The previous two performance requirements are based on the typical speeds seen in the team's app prototype
- **The mobile app will be created for the Android platform
    - This is based on the clients owning Android devices, Android being the leading mobile operating system globally, and the costs associated with putting an app on the iOS App Store
- *Stretch Goal:* Additional map labels such as cities, roads, and country borders will be added to the map
- *Stretch Goal:* Create an iOS version of the app
    - The primary functionalities of the Android app will be, whenever possible, translated directly into code for the iOS app
- *Stretch Goal:* Datasets from the previous decade(s) can be displayed
    - The interval between datasets depends on what LIDAR technologies were in-use before GEDI was launched
    - Tools for stating changes in the forest between datasets will be implemented such as how much area or biomass was lost

The mobile application needs to feature tools for locating areas of interest

- The user can specify a range of data values and only areas with data within that range will be shown
    - A lower-bound can be input, an unspecified lower-bound means all values below the upper-bound are applicable
    - An upper-bound can be input, an unspecified upper-bound means all values above the lower-bound are applicable
    - Any pixels that correspond to data values not in the filtered range will be blacked out.
    - *The loading times of displaying an applicable area will vary, with an expected maximum time of 5 seconds
- *Priority Stretch Goal:* The user can select a point on the map and data for that point will be shown
    - *Data for a single point will be determined and shown within 1 second
- *Stretch Goal:* Include an "about page" that includes background information about the app and instructions for its us

The mobile application needs to be capable of offline functionality

- The user will be able to specify an area and resolution that is needed for offline use
    - The user will be able to select a region of interest center-point
        - This will be done by clicking the desired location on the map
        - Only one can be selected at a time
        - This point will be stored within the app's persistent files so that even after closing the app, it will still be available next time the app is opened
        - It will be possible for the user to remove this point
    - A radius in kilometers must also be given before downloading can begin
    - *Up to 4 gigabytes of map tiles will be able to be stored on the local device
        - This will be implemented by restricting the radius to a maximum
        - This maximum will be shown to the user
        - Inputting a radius that exceeds this value will prevent downloading
    - A message will be shown stating that WiFi is recommended for downloading otherwise cellular data will be used
    - A download for offline use will automatically delete the previously downloaded offline map data
- The user will be able to specify when the device is in offline mode
    - When in offline mode, the map data will be loaded only from local storage and no attempts will be made to load data from the online storage server
    - It is required that offline map data has been downloaded prior to this point to display any map data
- *Stretch Goal*: Make it so the user can download more than one dataset at a time
    - The user will be able to select specifically which datasets they want to download for the provided region of interest
    - The user can also specify which datasets they want to delete

Text within the mobile application will be in either French or English

- The ability to select the language will be implemented in one of two ways:
    - User will manually select the desired language
        - User can change the language within the app's settings
        - The language chosen by the user will be stored within the app's persistent files so that this choice will be "remembered"
    - The app will be designed to allow multi-language support
        - This means the app will change the text language to what the device is configured to
        - It is more scalable than the previous option, but does not allow the user to change the language for the app specifically

    - Regardless of which option is chosen, translations will need to be manually written within the app
    - A hybrid of the above two options is a possibility
- **French translations will be provided by an individual fluent in the language who is an associate of the clients
- *Stretch Goal:* Implement more languages
    - For each additional language added, the process will be the same as how French was implemented

The mobile application will be capable of centering its map at a designated point

- The user can select from two default center locations
    - The map can be centered at the user's current location
        - The mobile device's GPS system will be accessed to accomplish this
    - The user will be able to set a region of interest center point
        - This location is the exact same as the region of interest described above, under the requirements for offline functionality section
- If the user's location cannot be retrieved and no region of interest is set, the map will by default be centered to some arbitrary location
    - This will be within the tropical forests of Africa as that is currently the focus of the project
    - This location is currently at the latitude-longitude coordinate (0, 20)
- The map will be centered around this location automatically when the app is started

A backend server will be created for storing map data

- The mobile app will be able to request map data files from an online server
    - When new map tiles are needed (such as after scrolling or zooming the map), they will be obtained from the server (assuming the device has internet connection and is not in offline mode)
    - When an area is specified for offline use by the user, the corresponding map data will be downloaded from the server to the device's local storage
- The storage server will be capable of holding large datasets
    - *An estimated maximum of 100 gigabytes of data will be able to be stored
        - This current estimate may be refined as the project progresses
        - What regions are covered by the app, the zoom levels that are available, and the resolutions of the map tiles will impact this estimate
- The storage server will be secure
    - Only authenticated administrators will be capable of updating and adding additional stored datasets
        - Before the end of the project, the clients will be have been made authenticated admins

        - The clients will be given documentation for maintaining the server
    - The mobile app will not be modifying any data on the storage server, only accessing it

The map data will initially be retrieved from where it’s currently stored

- **The CHM, DEM, and AGB data will be obtained by exporting them from GEE
    - Any of these that are not already stored on GEE and publicly available will be provided by the clients
    - A script will be written in GEE that exports them in GeoTIFF (.tif) format
    - Files exported from GEE are sent the Google Drive of a team member or the client
- The GeoTIFF files will be downloaded by a team member or the client from their Google Drive and tiled using tiling software
    - **The map tiles will be 256x256 pixels
        - These are standard dimensions for map tiles
    - **The map tiles will be in PNG (.png) format with 32 bits of color depth
        - This is the format exported by the desired tiling tool (GDAL) and is accepted by popular mapping development kits such as Google Maps and OpenStreetMaps
    - The tiling process preserves the geolocation of the original data via the names of the images the the folders containing them
- The map tiles will then be transferred to the storage server
- This retrieval process is currently being done by team members, but will be taken over by clients after the project ends

Risk Analysis

The functional/performance requirements rely on components that are different in architecture and so there is room for risks, and they need to be acknowledged. A failure that happens in any aspect of the system will cause the whole system to fail. This is because all the components are dependent on one another to create the experience the user gets. Potential risks have been identified that the application could encounter. These descriptions below include both the likelihood of it happening and the severity of it.

Legacy Node.js Packages

Likelihood: Low

Severity: High

Node.js packages fall into two categories: active development and legacy (outdated). Legacy packages are discouraged since there is a chance for it to become incompatible with newer software. This means that the team is going to strive to use packages that are in active development. Having active packages will also help keep maintenance low on the server. The likelihood of this event happening is rather low because the team will be using popular packages to handle logic on the server. Popular packages that have many projects that depend on it typically do not become legacy. If the case arises where packages on the server become outdated then the mobile application will become unusable. This is because the packages are what keep the server actively listening for requests from the application. Meaning if the server cannot listen for requests then the application cannot display data to the user. The mitigation strategy for this is to find packages that are similar to the legacy ones being used, but are in active development. The team will then bring the application down for maintenance, so old packages can be swapped out for new ones.

Android OS Update

Likelihood: High

Severity: High

The downside of having a native Android application is that it must be compatible with the current version of the OS. This means the team will have to actively anticipate an update that could cause the application to break on the new OS update. A possible solution to this is to use updated software for the Android framework. Which means little code will need to be modified to work on the new update. The likelihood of an Android update happening is extremely likely. Major OS updates happen on an annual basis, so it is inevitable that one will occur. The severity of this is it will cause the application to not work on the new OS. The mitigation strategy is to bring the application down for maintenance. This will provide the team with time to update the application with new components from the Android framework for the new OS.

Incorrect Data Provided

Likelihood: Low

Severity: Low

Any time an application interfaces with a server for data, there is a chance there is miscommunication on one of the sides. One form of miscommunication is the application can send a request for data that doesn't exist on the server. The other form is the server can respond with incorrect data to the request. For example, if a user wants to see DEM values for a region but instead the server responds with above ground biomass values. The likelihood of incorrect data being provided to the user is low. This is because the mitigation strategy that the team will take is to have manual checks throughout the development process. The checks will be to ensure there is no miscommunication between the application and the server. It will be done by making test requests to the server and seeing how the server responds.

Unpredictable Web Hosting Service

Likelihood: Low

Severity: High

Because the server will be hosted through a third party hosting service. There are several things that can go wrong such as overwhelming the service with requests, outages, or services being disrupted. The likelihood of this happening is rather low as the company is rather proficient at preventing these kinds of events from happening. However, if such an event does happen the severity will be high. This is because the server will be down and won’t be able to communicate with the mobile application. The mobile application on the other hand will be unaffected by this, but it will be useless since it cannot display data to the user. The mitigation strategy for this is to have a backup server. The server will be on a small device such as a Raspberry Pi and because of that performance of the application will be drastically reduced because of limited resources. This is however a temporary solution until the web hosting service is back up.

Project Plan

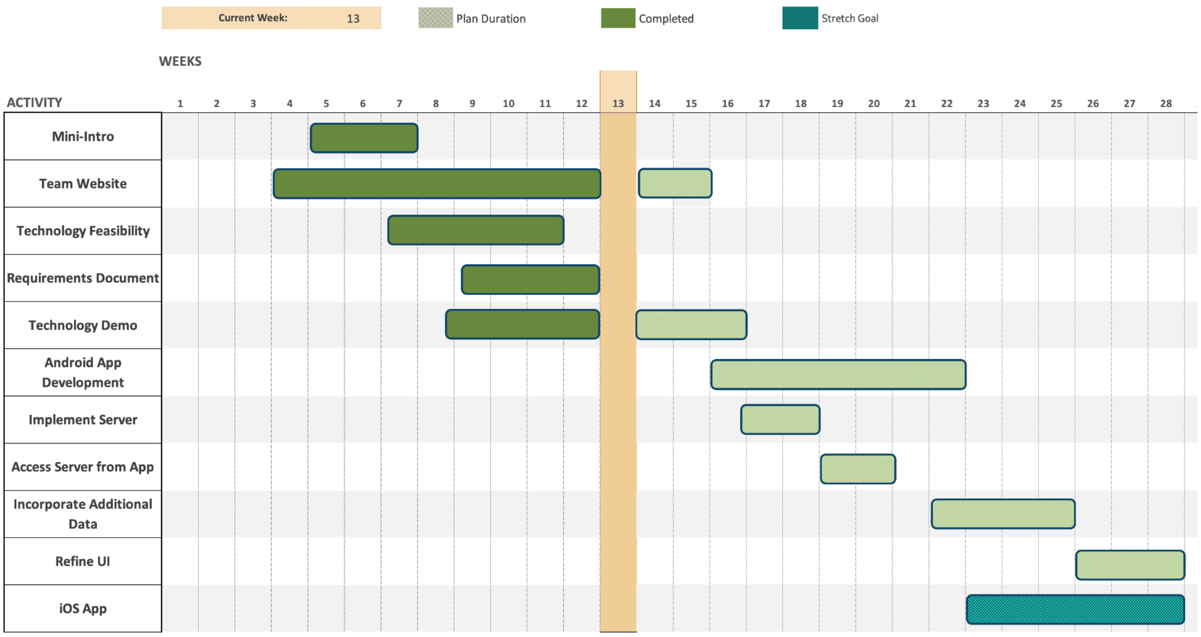

Figure 6.0: Gantt Chart for Development Plan

Including previous and ongoing work, the project has been divided into goals and have been assigned appropriate timeframes to be completed within. The plan is to continue development on the Android application over the next few months while simultaneously setting up the server for the data to be stored on. Data storage will be accomplished through an AWS server which allows for scalability in size. Once the server is functional and data is stored on it the team's next goal will be accessing the data stored on the server within the mobile application. When a functioning application is created, more data including above ground biomass and DEMs will be added to the server. Expanding on the mobile application by incorporating the new datasets to be displayed on the mobile application when toggled will require data to be added to the server and more development time. Towards the end of the mobile app development, a couple weeks will be set aside for refining the user interface, fixing bugs, and optimizing code. If time permits, a stretch goal the clients would like is development of an iOS application. This could be achieved by splitting the team into an iOS development group and a group to finish refining the Android application.

Conclusion

The loss of tropical forests is cause for global concern; this is due to the vast amounts of CO_2 absorbed by tropical forests. Crucial data that is derived from LIDAR imagery satellites such as digital elevation models and canopy height models can be used to interpret and provide insights into an ecosystem's health and biodiversity. Currently the problem with getting this data is the accessibility and complexity of the data. The solution envisioned by the clients will be a dynamic map which can display relevant map overlays to a user. An optional download feature will be included to aid researchers who are conducting field studies where internet connection is not guaranteed. The application will include GPS data to center on a user's location, the ability to filter data to a specified range, and access data stored on a server to reduce space complexity.

This document laid out the requirements for the mobile application and a plan on reaching the clients needs. By coming to an agreement with the clients through this document will mean both parties are in accordance with what is expected moving forward. Indicating that team Biosphere has done their due diligence in research and communication to achieve the correct outcome. With progress on a prototype mobile application that has a navigable map and acquisition of data from GEE stored on a server, the team is making great progress and will continue development in the upcoming months.

Glossary

Above Ground Biomass (AGB)
A computed dataset that provides an estimate of the carbon content of each area within a forest.

Canopy Height Model (CHM)
A measurement of the height of trees, constructions, and other structures above the ground topography.

Digital Elevation Model (DEM)
A representation of the bare ground topographic surface model of the Earth.

Global Ecosystem Dynamics Investigation (GEDI)
A high resolution LIDAR sensor aboard the International Space Station used to study Earth's forests and topography.

Google Earth Engine (GEE)
"A platform for scientific analysis, visualization, and storage of geospatial datasets and satellite imagery, for academic, non-profit, business and government users." [3].

Heatmap
"[A] graphical representation of data that uses a system of color-coding to represent different values." [2].

References

[1] "Why are rainforests important?". Rainforest Concern.
https://www.rainforestconcern.org/forest-facts/why-are-rainforests-important.

[2] "Heatmap". Optimizely and Fresh Egg.
https://www.optimizely.com/optimization-glossary/heatmap/.

[3] "Google Earth Engine FAQ". Google.
https://earthengine.google.com/faq/