ECLIPSE MEGAMOVIE PROJECT

There's going to be a solar eclipse. Let's make a movie!

PROJECT OVERVIEW

The Eclipse Megamovie capstone project is focused on making several contributions to the larger Eclipse Megamovie project, run by the Google Making & Science team and scientists at UC Berkeley. These contributions are focused on two components of the Eclipse Megamovie project. The first of these is a web based eclipse simulator, which we built from scratch. The second is an eclipse image processor, which will run in Google Compute Engine. The image processor was built by adding to/modifying existing code that was open sourced by Google. This modified image processor code is available at https://github.com/lariszakrista/EMP.

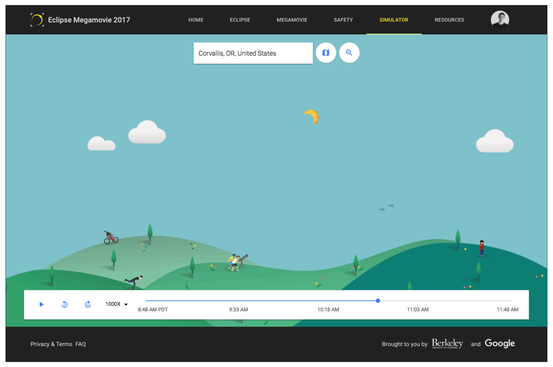


Figure 1

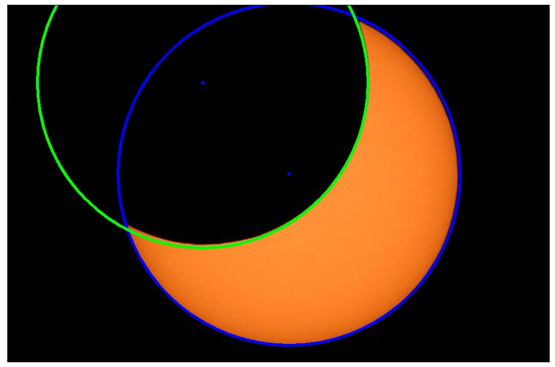


Figure 2

IMAGE PROCESSOR

The image processor is an application that analyzes photos of solar eclipses with the goal of determining whether or not a given photo shows an eclipse at totality, and if so, where the eclipse is in the photo. Figure 2 illustrates the results of running the image processor on an image of a partial eclipse – this is an image that would be discarded by the final image processor. The image processor will be used as a preprocessor in order to assemble user-submitted images into the final Eclipse Megamovie.

Development of the image processor includes creating additional tools to aid in this image processor development. One of these applications, `run_imgproc_test`, automatically pulls images from Google Cloud Storage, runs them through the image processor, and assembles the results into a nicely formatted HTML file. This HTML file is linked to the current version of the image processor source code and is uploaded to Google Cloud Storage. This makes it easy to share and reproduce image processor results. This application consists of a Python script and a calling Bash script.

The image processor itself is implemented as a C++ class. This class is designed to be inherited from, so its easy for developers to change small pieces of the image processor with minimal effort. It is easy to view and share the results of these changes using the aforementioned `run_imgproc_test` tool. OpenCV is used extensively by the image processor for its image processing functions.

The image processor remains under development by Google engineers. We hope our contributions have made their jobs easier.

ECLIPSE SIMULATOR

The eclipse simulator is a web application that allows users to visualize what the eclipse will look like at any location in the United States. It was officially released on the Eclipse Megamovie website this April – see Figure 1, or the live simulator at https://eclipsemega.movie/simulator. At the time of its launch, our simulator was the second most viewed page on the Eclipse Megamovie site, behind the landing page. Today it receives 25-50 daily views each lasting nearly 2 minutes!

The simulator is built using JavaScript, HTML, and CSS, and makes use of jQuery, Material Design Lite for the Googley UI, the Google Maps JavaScript API, and MeeusJs – an open source collection of astronomical computation functions, such as Sun/Moon position.

Although we did not implement the astronomical computations in the simulator, there were several interesting computational components we did have to implement ourselves. See the *Technical Deep Dive* section for an in-depth explanation of one of these.

Technical Deep Dive

SIMULATOR VIEW TRACKING

From a UX perspective it is highly desirable for the Sun to be relatively large in the simulator window, as shown in Figure 1. This Sun size (as shown) is in fact dramatically larger than the "real" Sun size for the field of view of the simulator.

There are two ways to achieve a larger Sun size like this:

a. *Increase the Sun's angular radius*
b. *Reduce the angular field of view displayed in the simulator window*

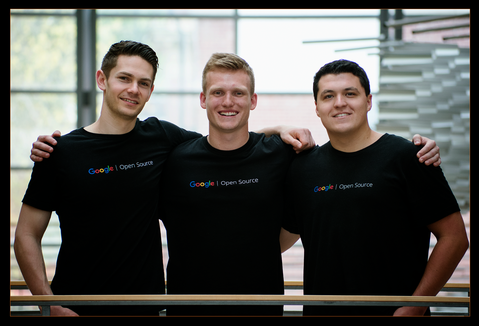
Unfortunately, both of these methods have significant drawbacks. Increasing the Sun's angular radius results in the appearance of a partial eclipse hours before the eclipse actually begins. And reducing the angular field of view results in there being a very short time window during which the Sun is visible in the simulator. In order to overcome these issues *and* maintain the Sun's large size, we had to get a bit creative.

Our final approach was to decrease the angular field of view of the simulator and *track the Sun's path through the sky*, to keep it visible in the simulator. Notice however, that tracking the Sun's path exactly results in the Sun remaining dead center in the simulator window. Therefore, instead of tracking the Sun's path exactly, we use a wider "reference" field of view and look at the Sun's position in this field of view at the beginning, middle, and end of the simulator time window. We then use this information to track the Sun such that at these 3 points in time, the Sun appears in the same position in our narrow field of view that it would be in the wider, reference field of view.

Notice that when tracking the Sun as described above, the amount of angular movement required to track the Sun from the beginning to the middle of the time window may be much different than that from the middle to the end of the time window. Therefore, linearly tracking the Sun could result in a sharp change in tracking speed at the middle of the time window. In order to avoid this and enable smooth tracking of the sun throughout the time window, we use Lagrange polynomials to create a smooth interpolating polynomial of these three points. The simulator then tracks by adjusting its position according to this smooth polynomial.

The Project

GROUP MEMBERS



Bret Lorimore: *lorimorb@oregonstate.edu*
Jacob Fenger: *fengerj@oregonstate.edu*
George Harder: *harderg@oregonstate.edu*
David Konerding (sponsor)

ECLIPSE MEGAMOVIE PROJECT

The primary goal of the Eclipse Megamovie Project is to produce a high definition, time-expanded video of the total solar eclipse that will cross North America from the northwest to the southeast on August 21, 2017. The Megamovie video will be pieced together from images collected by citizen scientists at various points along the eclipse path. This will provide continuous datasets that far exceed what any one person could capture from a single location, where the longest duration of totality possible would be under three minutes. The Eclipse Megamovie, by contrast, will be about two hours long (from https://eclipsemega.movie).

Google

Oregon State UNIVERSITY