Requirements

Totality AweSun

Bret Lorimore, Jacob Fenger, George Harder

October 28th, 2016

CS 461 - Fall 2016

I. Eclipse Simulator Requirements

1) The simulator is a standalone JavaScript module that can easily be included on an existing webpage.
2) Users can select the location from which to simulate the eclipse. This can be entered at any point while using the simulator.
    a) Location can be entered as: lat/long, address, zip code, city name, state name.
    b) Initial simulator location can be programmatically set as initialization parameter
3) Users will be able to adjust the simulator time from 12 before the eclipse to 12 hours after it.
    a) Time can be advanced via a slider, or clickable buttons.
4) All simulator resources will load in <500ms given a 1-10 Mbps internet connection. Once this is achieved file sizes will be optimized to achieve the fastest load time possible.
5) Simulator will display the local time that is being displayed, e.g. there is a well defined time associated with the user selecting Corvallis, Oregon as their location and a simulator time of -3:13 (3 hours 13 minutes) before the eclipse. This time should be displayed on the simulator.
6) Solar/lunar placement will be scientifically accurate for the given location and time. This accuracy will apply to edge cases as well, where the location is say, not in the path of totality, or on the opposite side of the world as the eclipse. In the latter situation, the simulator would have to shift to a night time display during the eclipse.

II. Image Pre-Processor Requirements

1) The image pre-processor will be compatible with Ubuntu 16. 04 and will include a script that to install all dependencies/ build the binary.
2) The application will accept the following input as command line arguments:
    a) Required: image_list_file
        i) Absolute or relative (to the directory the binary was invoked from) path to file containing a list of image file names with no directory.
    b) Required: output_dir
        i) Directory to write output files to.

c) Optional: image_path_prefix

i) Absolute or relative (to the directory the binary was invoked from) path to prepend to each image filename in image_file_list. Defaults to ./.

3) The application will accept JPEG (.jpeg/.jpg) and PNG (. png) image files.

a) Images of an invalid format will be disregarded and an error message will be written to stderr.

4) Invalid JPEG and PNG files (e.g. cannot be opened by OpenCV, width/height equal to 0px, etc.) will be ignored and an error message will be written to stderr.

5) The application will classify the input images as being one of the following types:

a) FULL_DISK

i) Image of an unobscured solar disk.

b) TOTALITY

i) Image of a total solar eclipse.

c) CRESCENT

i) Image of a partially eclipsed sun, creating a crescent shape.

d) DIAMOND_RING

i) Image of a nearly fully eclipsed sun where there is one hot spot on the suns perimeter. This hot spot along with the suns perimeter have the shape of a diamond ring.

6) The preprocessor will sort the images into five separate bins based on the relative delta of the position of the center of the sun and the moon. This delta will be a signed value based on the suns position, i.e. if the moon is to the left of the sun (in the cropped/rotated image) the relative delta will be negative and conversely, if the moon is to the right of the sun the delta value will be positive.

7) The application will crop the images to a standard size with the sun centered in them.

8) The application will rotate DIAMOND_RING and CRESCENT type images so that they are aligned horizontally, as described by Krista et al.

9) The application will write the following output to the output_dir directory:

a) image_transformation.txt

i) File containing one line per image processed with the following values (comma separated):

A) processed_image

B) image_type

C) rot_angle

D) crop_topl_x

E) crop_topl_y

F) crop_botr_x

G) crop_botr_y

H) rel_center_offset - optional, only included for crescent type images

b) Pre-processed image files

i) images/*_pp.[png—jpeg—jpg]

10) The application will format all log messages as follows:

a) img_preproc:level:timestamp:message

i) Values of level: ERROR/WARNING/INFO/DEBUG

ii) Value of timestamp: current timestamp

iii) Value of message: specific logging message

David Konerding, Project Sponsor	Date
Bret Lorimore	Date
George Harder	Date
Jacob Fenger	Date