High Resolution Image Viewer for 360 Images

OpenStreetMap

Xinyi (Cynthia) Fan
Google Summer of Code 2022

04.19.2022

Overview

JOSM is an extensive editing tool for OpenStreetMap (OSM) geodata created in Java. It provides a variety of advanced features that cannot be found on the OSM default editor such as presets, different imagery tools, and data validation and filtering. Among these, we are interested in one feature of its geotagged image viewer to view 360 images in JOSM. People can either view their own 360 images by opening in JOSM or using Mapillary images in imagery to open targeted 360 images. However, currently the feature does not support converting the 360 image into flat images while the resolution of the 360 image produced is pretty low and the display of the detail for the image is not efficient.

Goals

1. Implement a high-resolution image viewer for 360 images while providing the feature to zoom in at different zoom levels and display details as required.
2. Implement the GUI in JOSM for users to select between different resolution image viewers.

Specifications

Image Viewer

We will focus on creating an image viewer that uses CPU in JOSM to generate high resolution images.

Specifically, with the inspiration from GIMP, we can modify our code implementation of *mouseDragged()* and *getRotation()* into several features (either only use mouse control or use mouse and keyboard control) that could be included in our image view:

Selection of region: the input of mouse control (click and drag) to specify the region that needs the projection; Another try of section will overwrite the current selection (already have part of it and needs modification)

Tilt: click icon in the toolbox list or input using mouse control (shift+click and drag) to tilt the image;

Spin: click icon in the toolbox list or input using mouse control (Ctrl+click and drag) to rotate the region;

Zoom: the input of mouse control (Alt+click and drag) to zoom in to a certain level of the image (already have)

Also, with testing the geotagged image viewer, one possible feature that can be modified is the zoom-in/out feature done by the trackpad on Macbook. The drags on the trackpad are not quite functionable so we will also try to resolve this problem in this image viewer.

Once the selection is done, we will prompt our users to confirm the region of the section selected.

After these specifications are done, we will be ready to target that area of our input 360/panoramic image and convert it into a high resolution flat image and display it in the JOSM image viewer as a flat image.

GUI

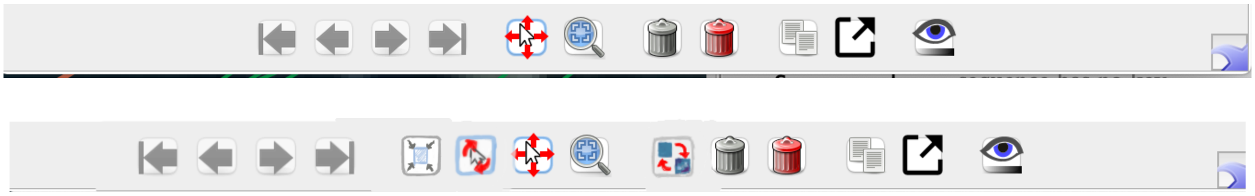
Under the imagery feature of JOSM, we will be able to implement several GUI tools that will help the user make their selection and modification.

Switch mode button: this will allow users to pick whether they want the normal 360 viewer or a high-resolution image viewer to a specific area of that image.

Toolbox feature: This includes the new spin and tilt feature we hope to incorporate into the image viewer to allow more variation of modification. Several new icons will be added to the toolbox.

Confirmation of region: For converting the 360 image into the flat image, we need a prompt to either do the conversion of the selected region or a confirmation to let the user secure the region selected for the conversion.

A sample drawing of the feature (before and after) can be found below:



In this drawing, the added icons from left to right are: selected regions, tilt/spin, and the conversion between low resolution and high resolution images.

In this case, we will successfully create an upgraded version of the image viewer in JOSM.

Timeline

I. June 13-17

Familiarize myself with the workflow of converting a 360 image and the code support behind every step. Specifically, familiarize with the code *ImageProjectionRegistry.java, Equirectangular.java, IImageviewer.java* and *Perspective.java.*

II. June 20-24

Work through the GIMP implementation and get familiar with the key idea behind it. Start to adapt the feature into codes.

III. June 27-July 1

Implement our new image viewer using the GIMP feature with CPU and without the help of GPU in codes.

IV. July 4-8

Implement the mouse control feature for our image viewer and test with image inputs.

V. July 11-15

Implement the converting procedure of turning the 360 image into equirectangular images.

VI. July 18-22

Test and debug the image viewer and make sure it is working properly.

VII. July 25-29

Start to implement the GUI feature of adding an option choosing feature of the image viewer. Work through the existing code and get familiarize with it.

VIII. August 1-5

Implement the GUI feature for image viewer in JOSM.

IX. August 8-12

Test and debug the whole procedure to improve any corner cases.

X. August 15-19

Catch-up week if previous procedure is late or exploration week for identify anything in addition to the current feature

XI. August 22-26

Summing up the work done and start to write the documentation for the feature implemented

Personal Information

Detailed information and implementation please refer to https://github.com/cyfan11/GSoC-proposal

Name: Xinyi (Cynthia) Fan

OSM account name: fanxy11

Current occupation: Senior Undergraduate at Davidson College in Davidson, NC, USA; Incoming graduate student at Carnegie Mellon University in Pittsburg, PA, USA.

Website/Github: https://cyfan11.github.io/page/

Involvement in OSM: I mapped the data and updated the routing information near and in Davidson College both using the OSM default method and JOSM.

Programming skills: proficient in Java with projects and upper-level courses taken; proficient in other languages including Python, C, C++; Experience with Javascript, HTML.

Previous and current Computer Programming projects: Refer to Github for a sample code. *TCP/IP routing and built a VPN forwarder with encryptor and decryptor *Partnered with a local nonprofit to help them build a website using R Shiny and Javascript to calculate and show the desired demographic data from user input *Built an AI that plays the space pinball and wins the game using reinforcement learning in python and unity.

Plans for the summer: I do not have any other commitments during this summer.

Understanding: I went through the procedure of loading a 360 image in JOSM and went through the codes provided to view the image and turn the image into equirectangular.

Adaptation: I identified the parts of the code that could be used to adapt into other implementations. This could be found in my timeline description and the comments in Equirectangular.java.

Brainstorming: I went through the panorama projection project of GIMP and is able to translate the idea into codes and ideas of implementation.