Plant Attribute Extraction

Campbell Motter

Attribute Modification & Storage

4 December 2022

Table of Contents

List of Figures

1. Subsystem Overview

1.1. Subsystem Description

My subsystem was Attribute Modification and Storage. The subsystem contains four major parts; multiple files, merging data, zipping data, and special cases. Multiple selections was the first requirement of this subsystem. I was tasked with allowing the user to generate plant attributes for multiple orthomosaic/canopy height model(CHM) files. The results from this attribute generation would all be downloadable in one zip package and with one process of generating results. Second, merging data was the next requirement. For users who select multiple files to be generated, there will be a merged CSV and SHP file containing the attribute data for each orthomosaic. A merged file will be created for each plant attribute. This allows the user to analyze and compare data much easier across different dates. The third requirement was the downloadable zip package. This package contains a parent zip file called resultsZip. Within that zip file there will be a zip file for each selected attribute which contains the individual results for each orthomosaic as well as the merged files. Finally, the special cases was a personal addition to my subsystem. These special cases include no CHM selected, and when the number of input orthomosaic files is more than the input CHM files. For no CHM, I needed to allow users to generate Canopy Cover(CC) and Excess Greeness(EXG) without a CHM file being input. The mismatching special case allows users to generate Canopy Height(CH) and Canopy Volume(CV) only for the input CHM files. Therefore if the user selects three orthomosaics and only one CHM file, CH and CV results will only be calculated for that one orthomosaic and CHM pair. This concludes the requirements of the Attribute Modification and Storage subsystem.

1.2. Current State of the Subsystem

My subsystem has been fully tested and validated. My subsystem correctly does all of these tasks listed above and can correctly generate results for all possible cases given by the users selected input

1.3. Future Improvements

There are a few possible improvements to my subsystem. One of which is threading for each of the input orthomosaics. This will allow for all of the results to be generated simultaneously. Plant attribute generation is a lengthy process, so this will save lots of time for high volume inputs. Another improvement I can think of is adding a progress bar viewable by the user. This will let the user know exactly how long their results generation will take. It is always better to keep the user informed on exactly what's going on and how much of their precious time will be spent.

2. Validation

Validating my subsystem includes four steps. First was to ensure that the Javascript was receiving all of the selected files and correctly turning the array into a comma separated string for a dynamic number of Orthomosaic and Canopy Height Model files. The second step was to ensure that the files for attribute generation were being completed. Thirdly was ensuring that the files were correctly zipped together. Finally, the fourth step was ensuring that each requirement of my subsystem was fully functional and giving the user accurate information.

2.1. Multiple Files

In order to ensure that multiple files were being selected through the GUI and passed to the python, I printed the selected Orthomosaic and Canopy Height Model variables to the inspect element console through every step of the process. I also did the same after the partitioning of the comma separated string to ensure none of the information was corrupted or incidentally changed when partitioning with the delimiters. The final validation would be seeing that each orthomosaic has viewable results within the resultsZip folder.

2.2. Merging data

In order to validate the merging data requirement I had to validate for both CSV and SHP files. The best way to validate this was making sure that a merged data file was created during each possible attribute generation. It was necessary to view each of these files and make sure the merged file was populated with all the selected orthomosaics as well as containing the same results that you would see in each respective individual file. The last validation for this requirement was making sure a merged file was not created when only one orthomosaic was selected.

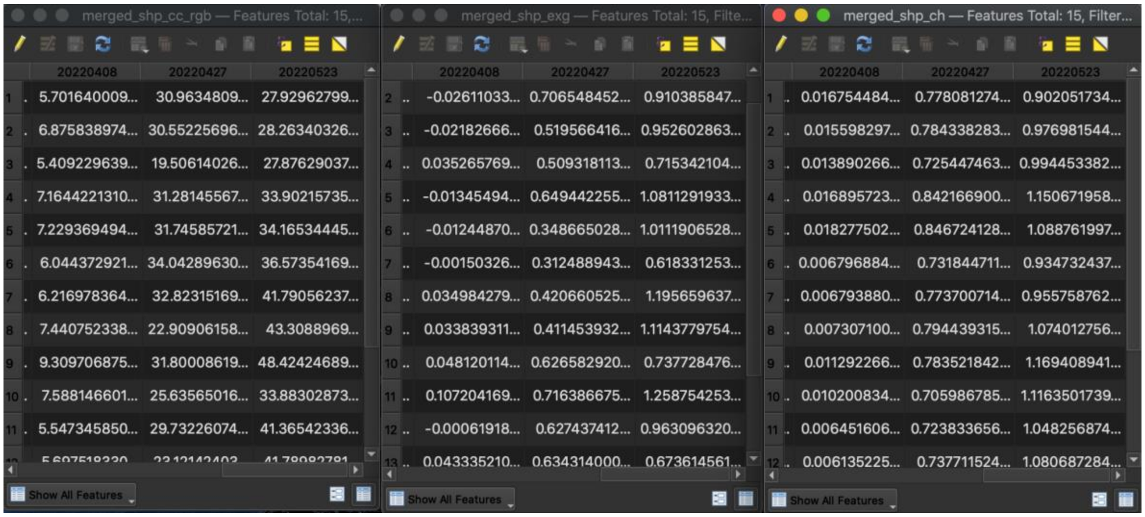

Figure 1: Merged SHP attribute table

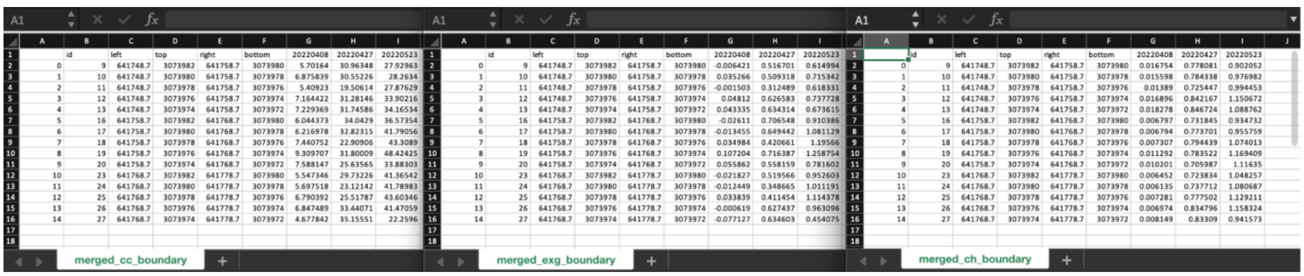

Figure 2: Merged CSV file

2.3. File Zipping

Validating this subsystem was a bit harder than some of the others. This is because I needed to be able to fully download results in order to see if the files were zipped correctly. Another reason was because if the resultsZip wasn't found, there were about five different places the file path could have been incorrect. The file paths were also quite lengthy making it hard to identify any small errors. I validated it by making sure the zip file was downloadable for each possible input case, ensuring that all the selected results were contained within the zip file.

```
ubuntu@bhub:/var/www/html/uas_data/download/product/2022_Corpus_Christi_Cotton/2
0220523_cc_p4r_parking_mosaic_clipped_2022_cc_corn_boundary_clipped$ ls
cc_boundary  ch_boundary  exg_boundary  results  resultsZip.zip
```

Figure 3: Attribute folders

```
ubuntu@bhub:/var/www/html/uas_data/download/product/2022_Corpus_Christi_Cotton/2
0220523_cc_p4r_parking_mosaic_clipped_2022_cc_corn_boundary_clipped/results$ ls
ccZip.zip  chZip.zip  exgZip.zip
```

Figure 4: ResultsZip zip file

2.4. Special cases

This portion wasn't the hardest to validate but definitely the most time consuming. In order to validate I needed to make sure that no CHM input wouldn't impede result generation for CC and EXG attributes. This was validated by checking that attribute results were made and placed in the resultsZip. This is checked by using Putty to check inside the SSH directory. The next special case was mismatching orthomosaics and CHM inputs. In order to validate this I tested that each possible mismatch case would correctly generate results and would contain all orthomosaic results for CC and EXG attributes, and CH and CV were generated for only the orthomosaics that had a matching CHM input file.

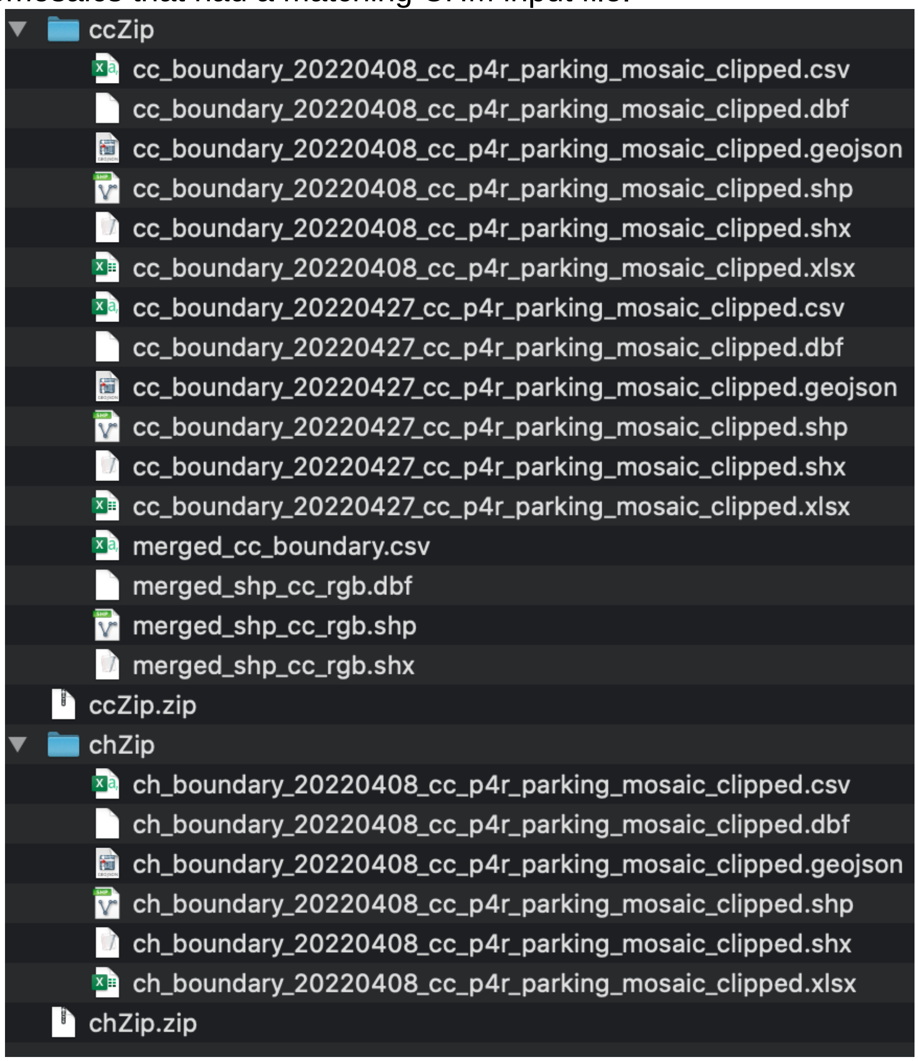

Figure 5: Mismatching orthomosaic/CHM

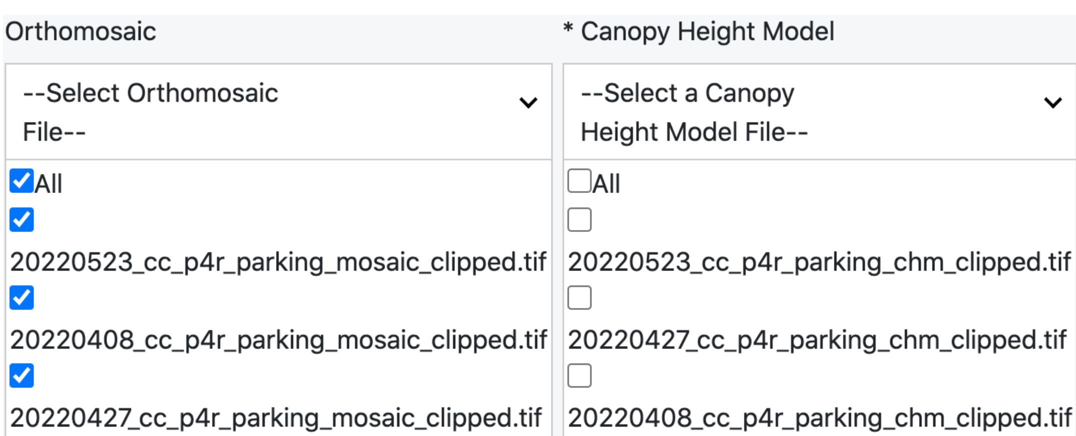

Figure 6: No CHM web input

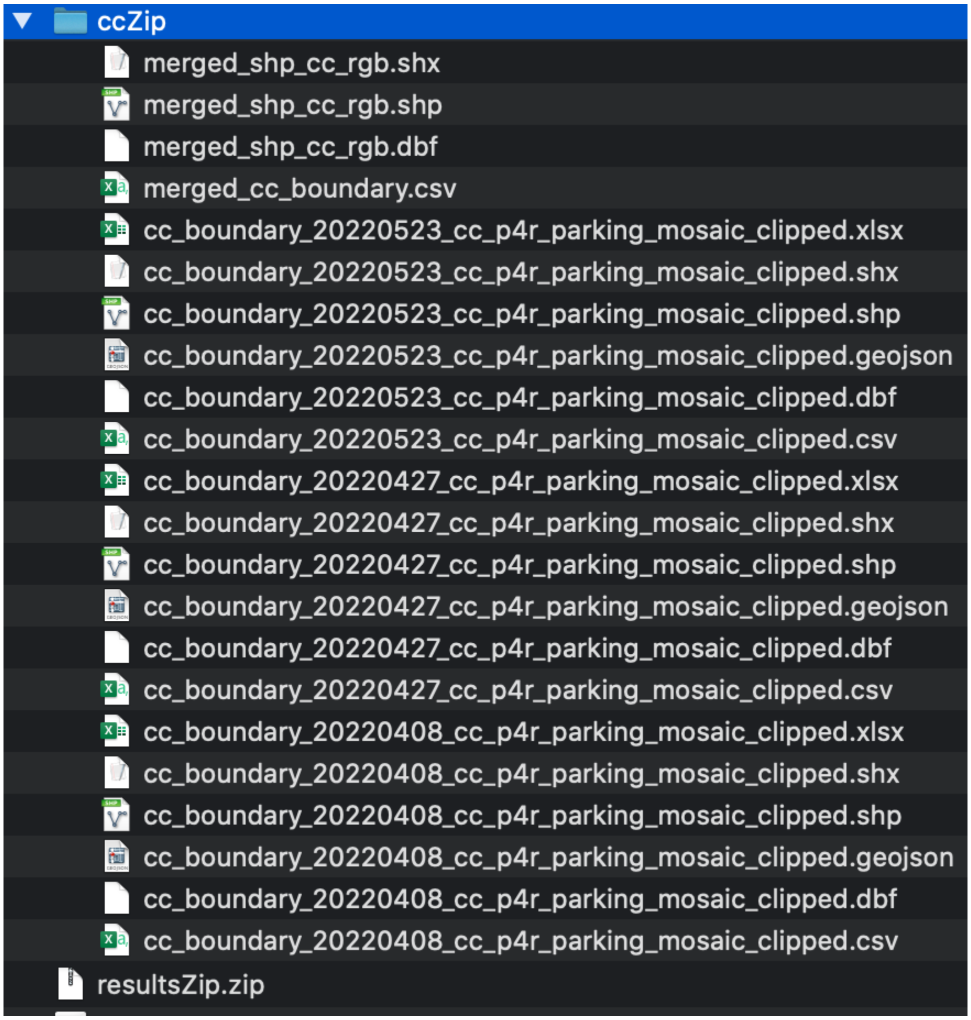

Figure 7: No CHM results folder view

2.5 Test Cases:

FULL SYSTEM TEST CASES:

Corpus:

- Mismatching Orthomosaic/CHM
  - [408,427,523] ,[408,427,523](P)
  - [408,427,523], [408,427](P)
  - [408,427,523], [408,523](P)
  - [408,427,523], [427,523](P)
  - [408,427,523], [408](P)
  - [408,427,523], [427](P)
  - [408,427,523], [523](P)
  - [408,427], [408,427](P)
  - [408,523], [408,523](P)
  - [408,427], [408,427](P)
  - [408,427], [408](P)
  - [408,427], [427](P)
  - [408,523], [408](P)
  - [408,523], [523](P)
  - [427,523], [427](P)
  - [427,523], [523](P)
  - [408], [408](P)
  - [427], [427](P)
  - [523], [523](P)
- Mix of Orthomosaics and CHMs:
  - [408,427,523] ,[408,427,523] {CC, EXG, CH}
  - [408,427,523] ,[408,427,523] {CH}
  - [408,427,523] {CC, EXG}
  - [408,427,523] {CC}
  - [408,427,523] {EXG}
  - [408,427], [408,427] {CC, EXG, CH}
  - [408,427], [408,427] {CH}
  - [408,427] {CC, EXG}

Figure 8: Corpus Christi Test Cases

- Mix of Orthomosaics and CHMs:
  - [408,427,523] ,[408,427,523] {CC, EXG, CH}
  - [408,427,523] ,[408,427,523] {CH}
  - [408,427,523] {CC, EXG}
  - [408,427,523] {CC}
  - [408,427,523] {EXG}
  - [408,427], [408,427] {CC, EXG, CH}
  - [408,427], [408,427] {CH}
  - [408,427] {CC, EXG}
  - [408,427] {CC}
  - [408,427] {EXG}
  - [408,523], [408,523] {CC, EXG, CH}
  - [408,523], [408,523] {CH}
  - [408,523] {CC, EXG}
  - [408,523] {CC}
  - [408,523] {EXG}
  - [408], [408] {CC, EXG, CH}
  - [408], [408] {CH}
  - [408] {CC, EXG}
  - [408] {CC}
  - [408] {CC}
  - [427], [427] {CC, EXG, CH}
  - [427], [427] {CH}
  - [427] {CC, EXG}
  - [427] {CC}
  - [427] {CC}
  - [523], [523] {CC, EXG, CH}
  - [523], [523] {CH}
  - [523] {CC, EXG}
  - [523] {CC}
  - [523] {CC}

Figure 9: Mismatch Test cases

- [427], [427] {CH}
- [427] {CC, EXG}
- [427] {CC}
- [427] {CC}
- [523], [523] {CC, EXG, CH}
- [523], [523] {CH}
- [523] {CC, EXG}
- [523] {CC}
- [523] {CC}

Amarillo:

- Standard:
  - [408], [408] {CC, EXG, CH}
  - [408], [408] {CH}
  - [408], {CC, EXG}
  - [408] {CC}
  - [408] {EXG}
- Unnecessary CHM:
  - [408], [408] {CC, EXG}
  - [408], [408] {CC}
  - [408], [408] {EXG}

Success/Fail Scenarios:

- Success:
  - (Previously listed test cases)
- Fail:
  - Selecting Canopy Height or Volume without selecting a CHM
  - Error reading image file
  - Error reading the SHP file
  - Project/Orthomosaic results do not exist
  - The specified canopy attributes do not exist

Figure 10: Amarillo test cases and fail cases

3. Github Repository:

https://github.com/RonBatista/ECEN404_Capstone

Appendix A: Acronyms and Abbreviations

Below is a list of common acronyms and abbreviations:

CSV - Comma-separated values
PHP - Hypertext Preprocessor (coding language and file type)
GUI - Graphical user interface
SHP - Shapefile
CC - Canopy Cover
EXG - Excess Greeness
CH - Canopy Height
CV - Canopy Volume
SSH - Secure Shell
CHM - Canopy Height Model

Appendix B: Definition of Terms

- Orthomosaic: stitching together multiple images, correcting distortion and color balancing to produce a seamless mosaic dataset.
- Threading: running the code concurrently using a coding method called threading