Utilizing multi-spectral data from drone imagery to create high resolution map products

Katherine Xu & Junpeng Fu

Data Gathering - Narrabri Trip











Process

1. Drone Photos
2. Go to moisture probes/ deciding extra points
3. Geometric data collection
4. Cut biomass
5. Weigh all biomass before/ after dehydrating

Task Allocation

Katherine:

Calibration (taking raw image and adjust for camera distortions/ colour correction based on radiance & irradiances)

Junpeng:

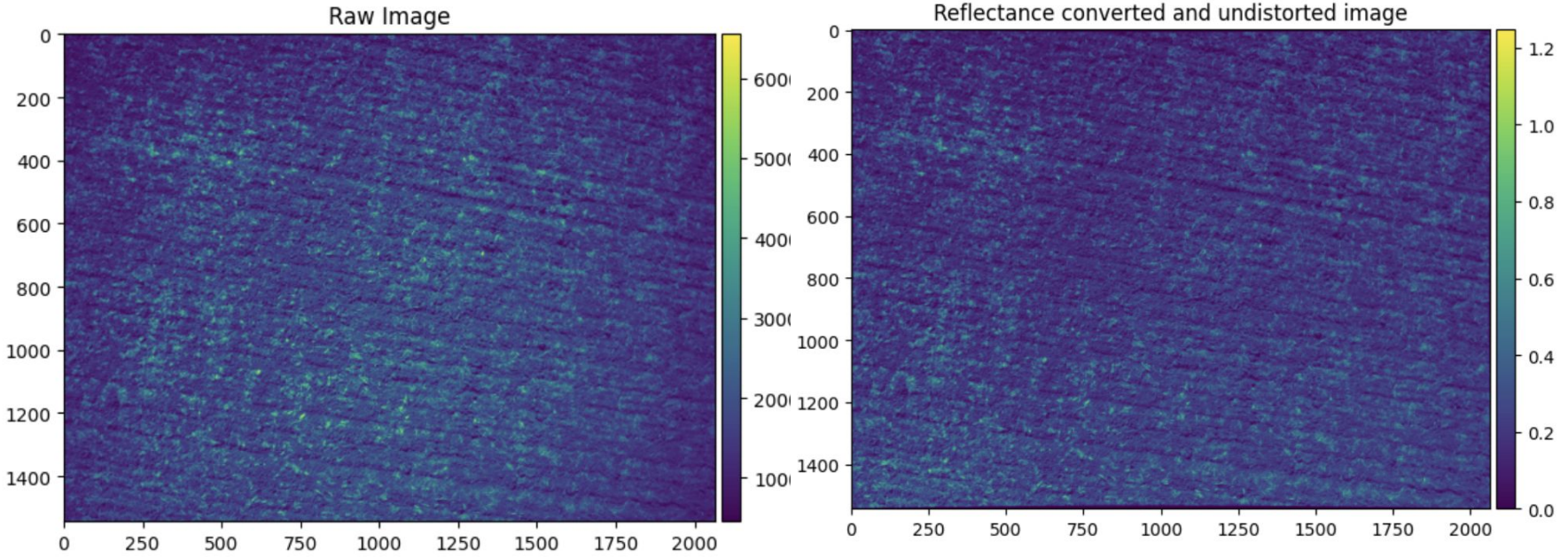
Stitch the calibrated images into a Orthomosaic map

Calibration

- Tutorial 1
  - Using the panel image to calibrate all images
- Tutorial 3
  - Using the DLS irradiance data and the radiance data reflected from the ground to calculate reflectance
  - Each capture has its own calibration reference

- Difficulties encountered
  - Environment setup (recommended OS: Windows)
  - Metadata writing
  - Image saving

Product


Raw Image
0
200
400
600
800
1000
1200
1400
0
250
500
750
1000
1250
1500
1750
2000
600
500
400
300
200
100
Reflectance converted and undistorted image
0
200
400
600
800
1000
1200
1400
0
250
500
750
1000
1250
1500
1750
2000
1.2
1.0
0.8
0.6
0.4
0.2
0.0

Batch processing

```
if not os.path.exists(outputPath):
    os.makedirs(outputPath)
if generateThumbnails and not os.path.exists(thumbnailPath):
    os.makedirs(thumbnailPath)

# Save out geojson data so we can open the image capture locations in our GIS
# with open(os.path.join(outputPath,'imageSet.json'),'w') as f:
#     f.write(str(geojson_data))

try:
    irradiance = panel_irradiance+[0]
except NameError:
    irradiance = None

start = datetime.datetime.now()
for i,capture in enumerate(imgset.captures):
#     outputFilename = capture.uuid+'.tif'
    outputFilename = capture.uuid
    thumbnailFilename = capture.uuid+'.jpg'
    fullOutputPath = os.path.join(outputPath, outputFilename)
    fullThumbnailPath= os.path.join(thumbnailPath, thumbnailFilename)
    if (not os.path.exists(fullOutputPath)) or overwrite:
        if(len(capture.images) == len(imgset.captures[0].images)):
            capture.create_aligned_capture(img_type, irradiance_list=irradiance, warp_matrices=warp_matrices)
#             capture.save_capture_as_stack(fullOutputPath)
            capture.save_capture_as_bands(fullOutputPath)
            if generateThumbnails:
                capture.save_capture_as_rgb(fullThumbnailPath)
    capture.clear_image_data()
    update_f2(float(i)/float(len(imgset.captures)))
update_f2(1.0)
end = datetime.datetime.now()

print("Saving time: {}".format(end-start))
```

Using the Batch Processing for each corrected image instead of stacks #121

New issue

Open **Prosperoux** opened this issue on Oct 14, 2020 · 12 comments

Prosperoux commented on Oct 14, 2020

Hello dear community,
I have followed correctly the tutorials in each steps. It works for a own single capture which is correct.
Now i try to use the batch processing, but i would like to get each image corrected in a loop, but not using the stacks.
How can i modify the code to get these results ? I am newbie in python and associated library, it's hard now to understand how to proceed.
Can someone help me ?
Thanks !

poynting commented on Oct 16, 2020 Contributor

It's possible what you're looking for is already proposed in the pull request here: #110

There were a couple of small issues with that which is why it has not been merged yet; maybe you can check out that branch and test if it does what you need?

1

Prosperoux commented on Oct 16, 2020 Author

Great ! Thanks. The modified capture.py with function capture.save_bands_in_separate_file from the code "Align images and save..." works.

Assignees
No one assigned

Labels
None yet

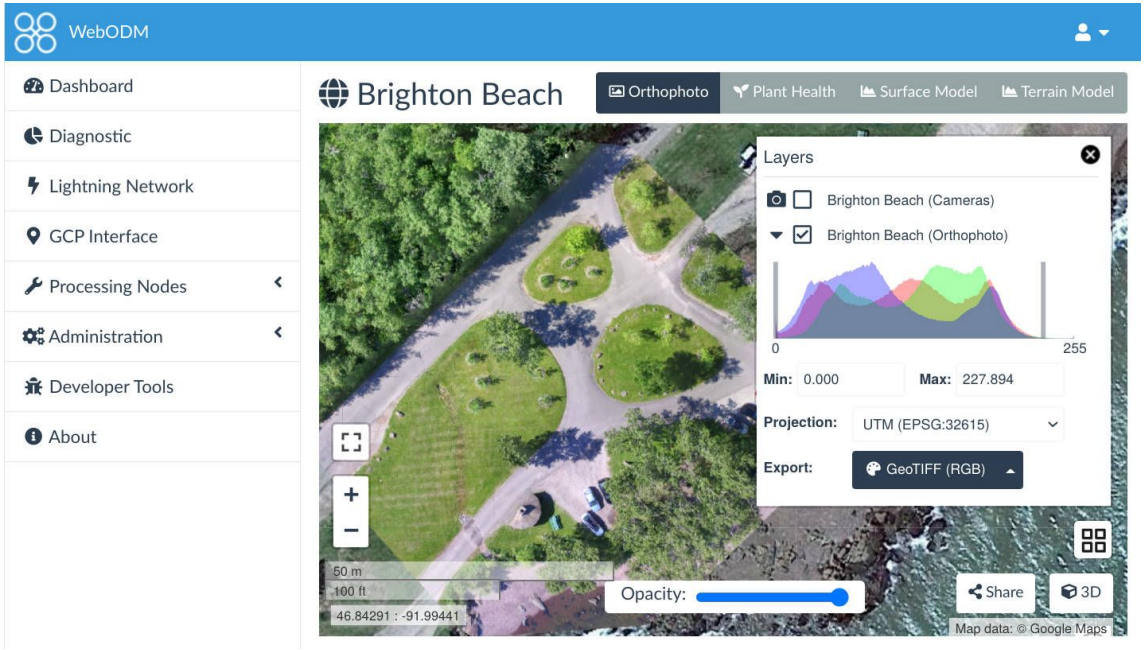
Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

4 participants

OpenDroneMap

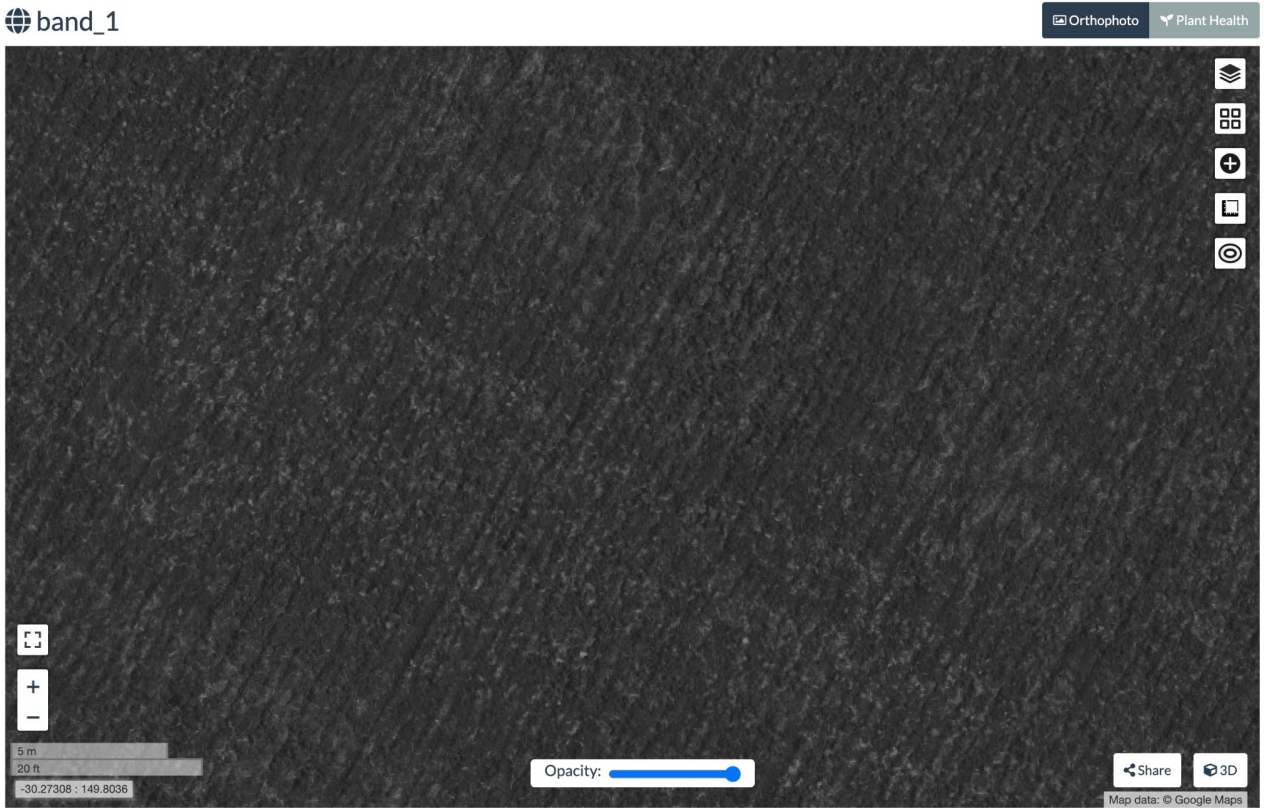


A user-friendly, commercial grade software for drone image processing. Generate georeferenced maps, point clouds, elevation models and textured 3D models from aerial images.

Calibrated band 1 (Testing dataset)



Calibrated band 1 (Testing dataset)



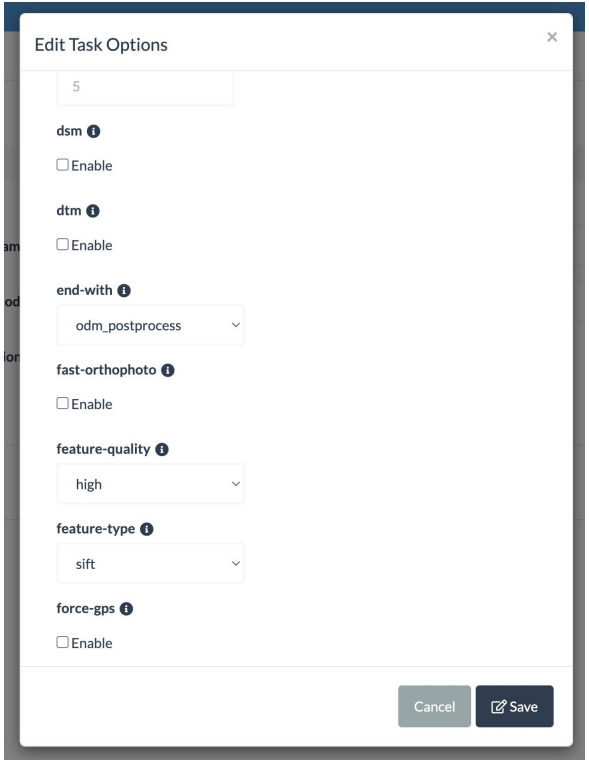
Calibrated band 5 (Testing dataset)

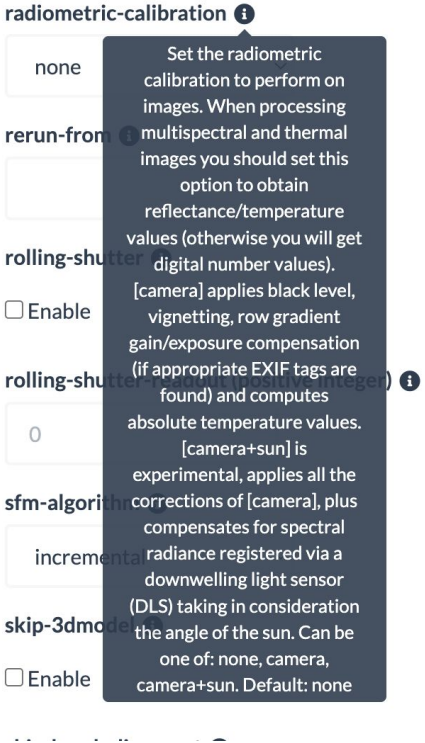


Uncalibrated (Final dataset)



Task options







Bringing it Together

Pipeline:

1. Calibration
    a. Setting up micasense environment
    b. Follow tutorials 1~3 to understand micasense library
    c. Follow “batch processing” notebook to process the images in batches
        i. The result image can be opened using the QGIS software (otherwise it’ll look pitch black)
2. Create map using OpenDroneMap
    a. Watch for disk space and RAM usage
    b. Understand all options to create the map

Further steps

1. Finish processing all data in the network drive
2. Stitch them into a map
3. Wrapping up everything with Rajitha on 7th March

Something to consider: Imagebreed

ImageBreed v10.1 Now Released

Launch your own instance of ImageBreed to manage your plant breeding field experiments, genotyping, aerial image phenotyping, and high-dimensional phenotyping (NIRS, transcriptomics, metabolomics)!

Please check the changelog for update information.

For a tutorials of pieces of the aerial image phenotyping system, follow the ImageBreed Channel.

Now integrated with OpenDroneMap for creating orthophotomosaics directly into the database!

Prerequisites

- You need Docker, docker-compose, and nginx installed on a machine with at least 8GB RAM and 50GB of disk. Though it is not necessary, ideally, the computer has >8 CPU cores, >96GB RAM, and >4TB disk space, and the disk space should be fast (Thunderbolt if NAS used).
- Check the speed of your disks and ensure a write speed >1 GB/s.

GitHub - nickmorales/imagebreed_dockerfile: The Dockerfile for ImageBreed