Open Drone Map:

Structure-from-Motion Workflow

Valorie Marie, Andrii Zaiats, Ryan Wickersham, Trevor Caughlin

This document shows a workflow to complete image processing and photogrammetry, also known as structure-from-motion. To process our projects, we used the WebODM versions 2.8.4 to 2.8.8 Native Windows Application. The first part of the workflow follows the flow for an RGB sensor to extract products such as an orthomosaic, digital elevation model, and 3-dimensional point cloud. The workflow assumes the use of ground control points collected in the field and visible in the imagery, which allows for an accurate geographical spatial reference. The second section covers multispectral sensor data, including how the processing differs from the RGB workflow.

The third section has all the Plant Health Options available in post-processing in WebODM, such as Normalized Difference Vegetation Index (NDVI) and Green Leaf Index (GLI), to name a couple. Finally, the last two sections have additional resources that have been gathered to assist in troubleshooting and some WebODM Community Pages that have been useful in sorting out some issues along the way.

This document includes the following sections:

RGB Data

The imagery used in the red-green-blue (RGB) processing was acquired using a DJI Mavic 2 Pro (SZ DJI Technology Co., Ltd., Nanshan, Shenzhen, China) with an RGB Hasselblad 20MP sensor in a raw (.DNG) format and later converted to .JPG format for WedODM processing. All ground control points (GCPs) were collected using Real-Time Kinematics (RTK) GPS system.

- We used Topcon HiPer V (Topcon Positioning Systems, Inc.), containing a stationary (base) and mobile (rover) RTK receivers to collect the points. We ran the base station for at least 2.5 hours and post-processed using the Online Positioning User Service (OPUS) approximately two weeks after the date of data collection. Each GCP was collected for 60 seconds using the rover at 2 m above ground and post-processed using the corrected base station location. We used location data in the WGS84 horizontal datum, and Ellipsoid height for vertical datum.

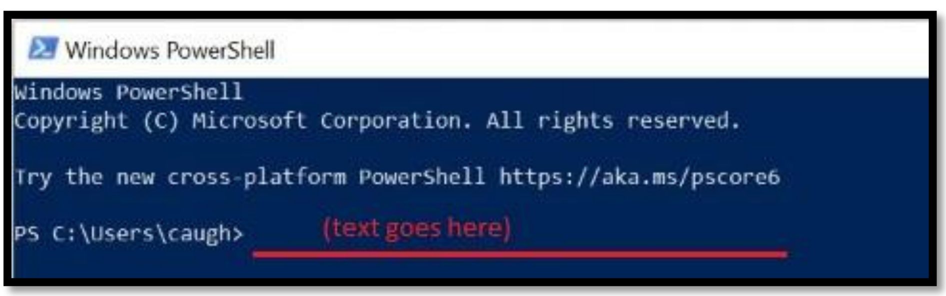
RGB Image Preparation (JPGs)

1. Images must be in an acceptable format for WebODM before we begin (JPG, PNG, BMP, TIF).

2. To convert from .dng (or .DNG) files to .JPG., open the application 'Windows PowerShell.'

3. Copy and paste the line below into Windows PowerShell, replacing the path name with the pathway to the .dng images. If this fails, rename the folders used in the pathway using underscores instead of spaces. We used ImageMagick open software to convert the images to .JPG format (The Imagemagick Development Team, 2021). If the execution is successful, copies of the original image files will appear in the same directory with .JPG file extensions.

   a. `magick mogrify -format jpg -quality 100 D:\Name\Location_Images\*.DNG`



4. Right-click one of the new JPGs, go to properties, then details, to see whether metadata transferred. If metadata did not transfer, there will be no GPS attached to each photo. You will need to copy the metadata from the .dng images to your newly created .JPG images. We used ExifTool to write the metadata from .JPG to .dng files (https://exiftool.org/) (Harvey 2020). Copy and paste the line below into Windows PowerShell. Replace the path name

with the pathway to all your .dng and .JPG images. **Keep images in the same folder until this step is done.**

a. `exiftool -tagsfromfile %d%f.dng -all:all -overwrite_original -ext jpg D:\Name\Location_Images\`

5. Verify that the final .JPG files contain all metadata by going to properties and then details.

6. Next, copy all the JPGs to a separate folder (labeled appropriately), or delete the original DNG files if they are adequately stored elsewhere.

Ground Control Points (GCPs)

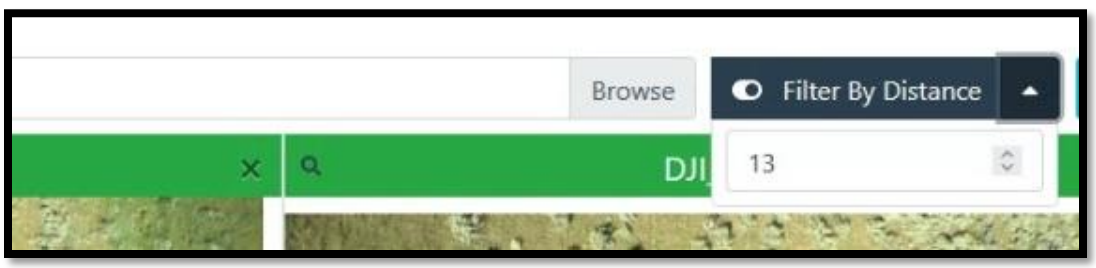
7. We need the .csv file to prep the gcp.txt file which can be used in WebODM. Prepare the ground control points list by opening the GCP .csv file with the GPS locations of the ground control points laid out before the flight. If applicable, remove extra lines such as the base and plant rows. The .csv file must have four columns: label, latitude, longitude, and elevation. Save.

8. Ensure the Lat and Long are in decimal degrees (dd.ddddd), not degrees, minutes, and seconds (DMS) coordinates. If they need to be converted, this website is an option: https://www.latlong.net/degrees-minutes-seconds-to-decimal-degrees

9. We used GCPEditor Pro to prepare the gcp input file (UAV4GEO, 2022, https://uav4geo.com). Open the WebODM Desktop Application and go to the GCP Editor Pro tab. This will allow you to load the .csv file and images, and to place markers in the photos. Click browse, navigate to the GCP .csv file and select. Check that the coordinate reference system (CRS) is correct. Most of the time, it will be WGS84 (i.e., epsg:4326), but if the GPS data is in a different CRS it needs to be specified before you move to the next step. Click 'Go to next step' in the bottom right corner.

10. Click on a GCP marker on the map or from the list on the left that you want to place first, then click 'Tag'.

11. At the top, click 'Filter By Distance' and type 10 meters to narrow down the photos to load and mark. Next, click 'Browse', navigate to where the newly created JPGs are, and select all the images. This can take a few minutes to load the images the first time, but once they are loaded in you do not have to do this step for every GCP. If you do not have ten images available to mark, slowly increase the 'Filter By Distance' to get more images. It is not recommended to go too large a distance because the chance of the GCP showing up in the image is vastly reduced, and the chances of picking up another GCP goes up. We found a 20 meter threshold useful for our UAS platform with flight altitudes at 40 m above ground, but a different flight mission may require adjusting this threshold.



12. Hold shift and scroll in to see the GCP. Click the center to place the marker as accurately on the center of the GCP as possible. The header is blue if there is NOT a marker placed on the image and turns green when one has been placed. If there is no GCP in the image, you can exit that photo by clicking on the 'X' in the upper-right corner of the image. Place 5-10 markers/images per GCP. If more than one GCP appears in a single image, mark the correct GCP each time. Then, click "Save Changes" in the bottom right corner.
    a. The first image closest to the GCP should have the GCP closer to the middle of the image. Scroll in slowly to look for it. Alternatively, you can open that same image from File Explorer to look for the GCP. Using colored objects next to ground markers, or uniquely shaped/imprinted ground markers, may make this task easier.

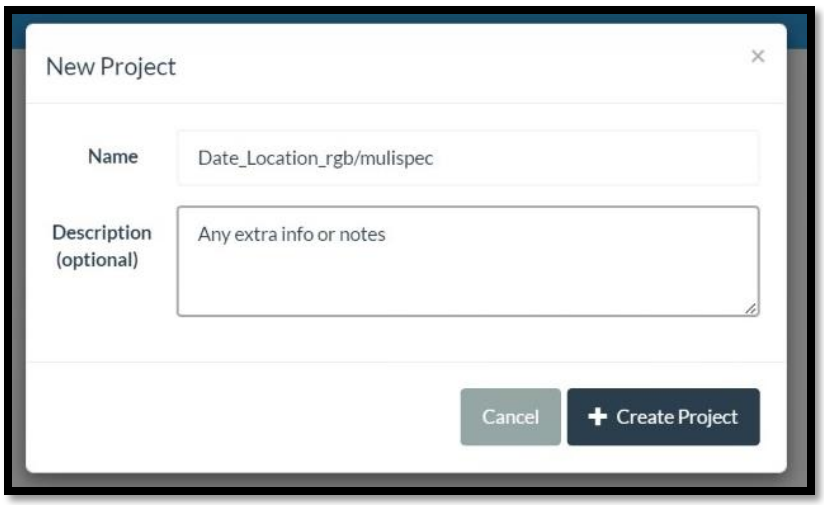
13. Continue this for all the GCPs. Then 'Go to the next step' in the bottom right corner.

14. Download GCP_list.txt. Rename it with the flight date, location, and type of imagery (e.g., YYYYMMDD_siteName_rgb_gcp_list.txt). **Put this file in the same folder as all the .JPG images created in Step 6 of the RGB Image Preparation**.

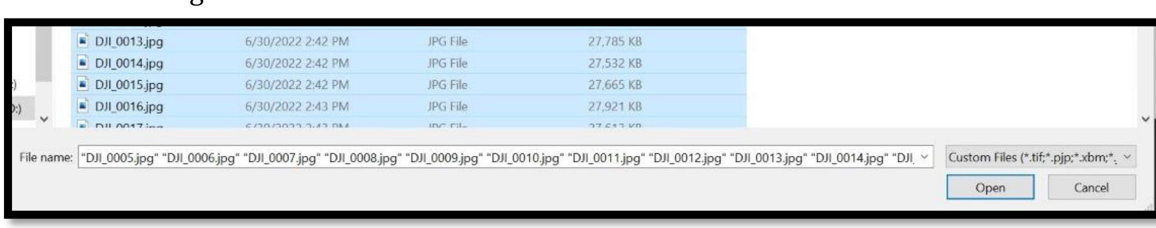
Open Drone Map (WebODM)

15. Open the WebODM application. We used natively installed 'WebODM' on Windows, but the interface will be similar regardless of the platform or the type of installation.

16. Add Project, Title the project. >> Next section: leave the name as what it auto-populates.



17. Select Images and GCP >> Navigate to the folder with the JPG images (Step 5 in RGB Image Processing) and the txt file with the GCP list (Step 14 of the GCP section). **You will need to use the drop-down to show all files, not just the images.** Then select all the images and the txt file that is now not hidden (keyboard shortcut: Ctrl+A). You can check whether the ODM has the gcp file by verifying the number of files imported to the ODM: the number should be #images + 1.



18. RGB Processing Options and Parameters:

Note: some of the options may change depending on the version of the software. Full list of options and explanations found here: https://docs.opendronemap.org/arguments/ .

auto-boundary:true

camera-lens:brown

dem-resolution:1 – Based on the flight parameters and ground sampling distance (GSD).

depthmap-resolution:2400 – Several projects of similar UAS mission plans failed to reconstruct under 2400. After trial-and-error runs, we found that lower depth map resolution at 1900 did not significantly reduce the quality of the DSM, while at 1000 the change was noticeable.

dsm:true

feature-quality:ultra

mesh-octree-depth:12

mesh-size:400000

min-num-features:80000 – We maximized the mesh size and minimum number of features under our processing resources to maximize the accuracy of structural product.

optimize-disk-space:true

orthophoto-resolution:1 – Based on the flight parameters and ground sampling distance (GSD).

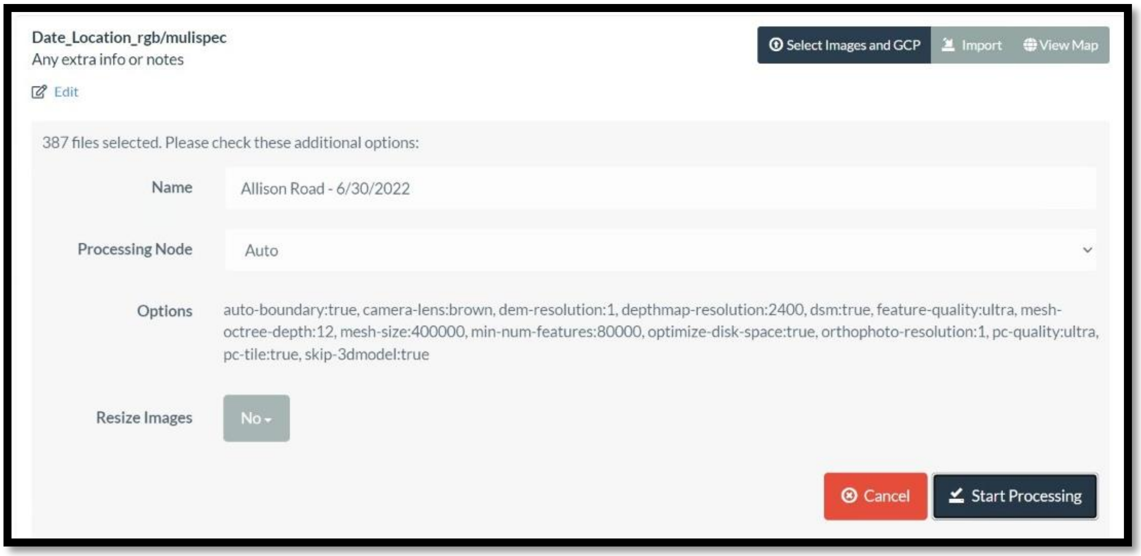
pc-quality:high – We did not find significant differences between the 'ultra' and 'high' options for our datasets, except the reduction in processing time under the 'high' option.

pc-sample:0.0005 – The resulting dense points from the parameters above are incredibly dense. We attempted to slightly thin out the dense cloud to avoid point replicates and more efficient processing.

pc-tile:true

skip-3dmodel:true – Optimize processing resources since the 3D model was not the goal of the UAS project.

19. Review >> Start Processing.
    a. This process can take several hours to a couple of days, depending on the machine's capabilities, storage settings, memory space, and the size of the project to be processed.

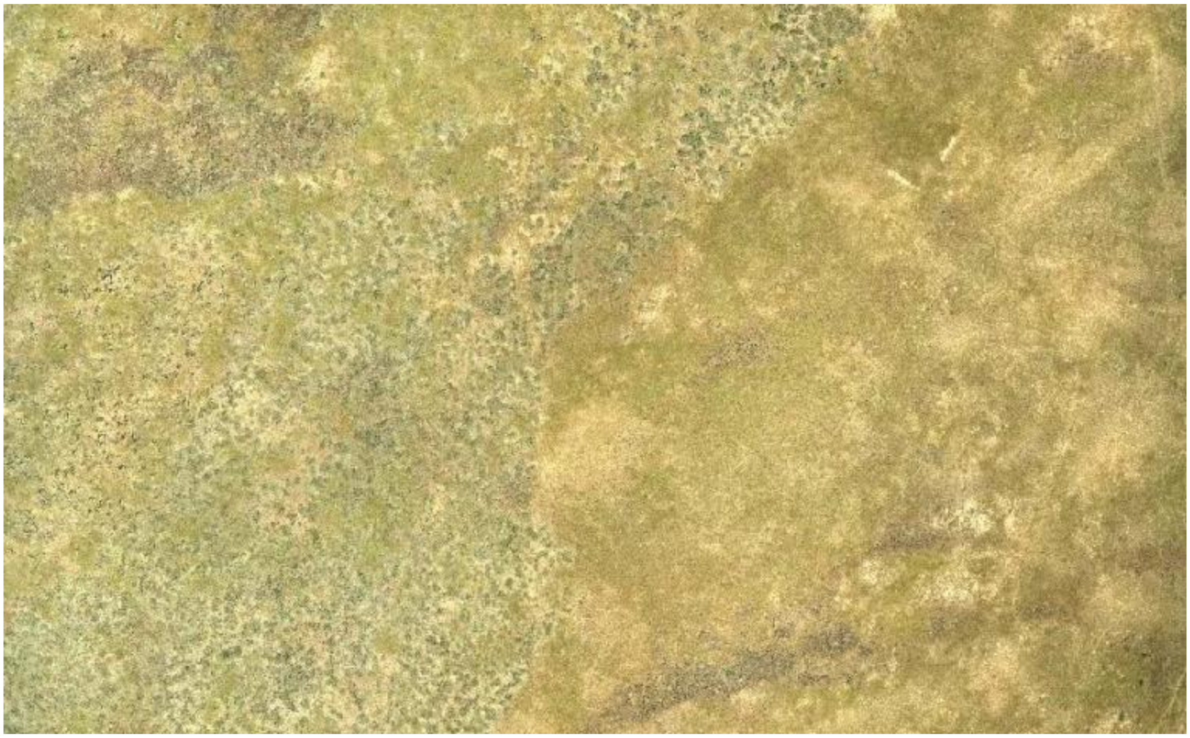


20. Download Assets >> **Point Cloud (LAS)** >> move to the correct products folder.
    a. This step, downloading the point cloud separately and assembled, will save time and work later instead of merging all the tiles.

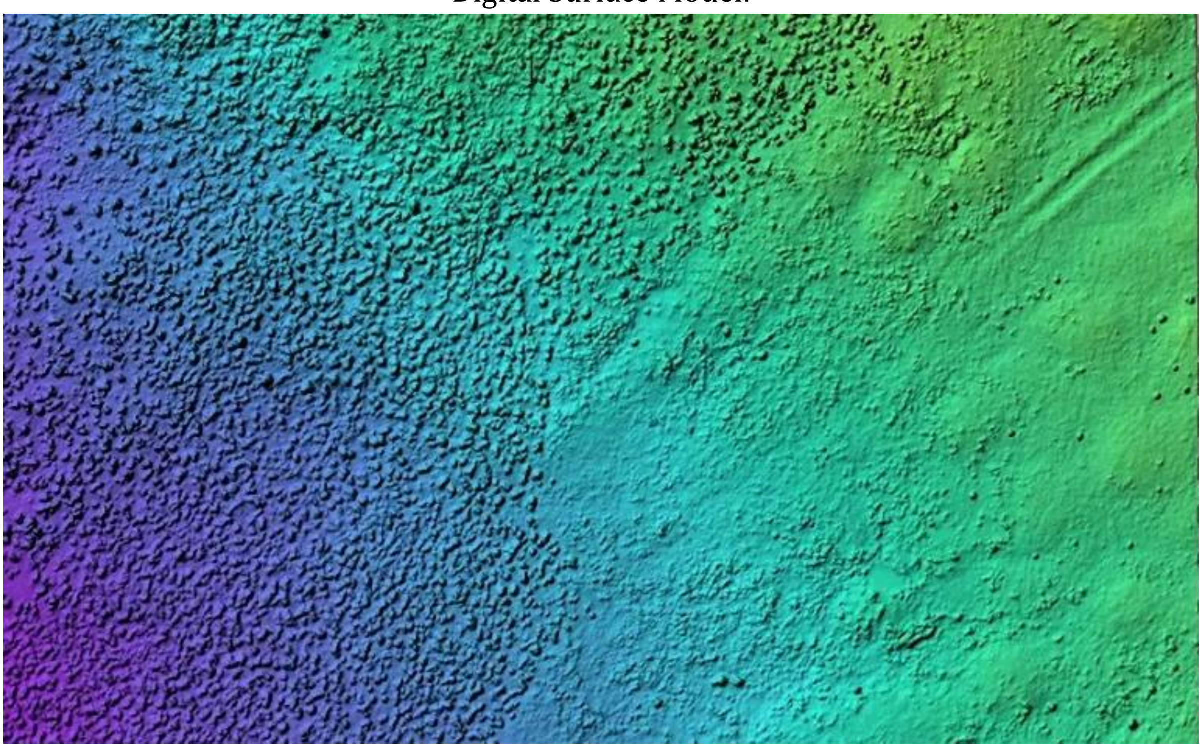
21. Downloading "**All Assets**" will give a folder with more detailed information >> move to the correct products folder
    a. This folder should provide all other needed products without downloading them individually.
    b. (entwine_pointcloud, odm_dem, odm_georeferencing, odm_orthophoto, odm_report, cameras.json, images.json, log.json, task_output.txt)

RGB Product Examples

Orthomosaic:



Digital Surface Model:



Dense Point Cloud:

Multispectral Data

All multispectral imagery was collected using a DJI Phantom 4 Multispectral (5 bands) drone (SZ DJI Technology Co., Ltd., Nanshan, Shenzhen, China), and all ground control points (GCPs) were collected using Real-Time Kinematics (RTK) with a TopCon base and rover.

Image Label Corrections for Large Datasets

Depending on the dataset's size, you may find that some labels for the images have been reused. In our case, we would have images labeled "DJI_0011" and "DJI_0011 (2)" to indicate that they are different images with the same name. Unfortunately, this caused some errors further down the line, so we found it best to change these labels beforehand. We renamed the image files following the following scheme: 'DJI_0011.TIF' -> 'DJI_10011.TIF', 'DJI_0011 (2).TIF' -> 'DJI_20011.TIF'.

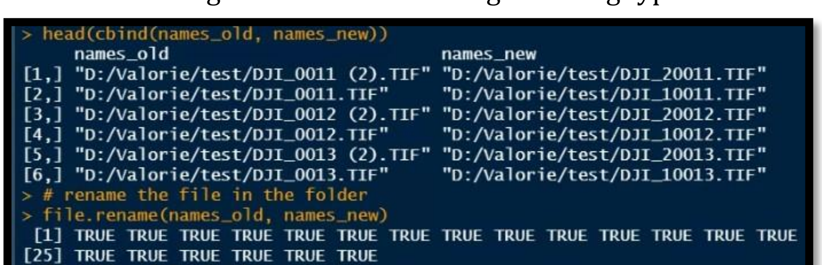
1. Open RStudio, copy and paste the function code from: https://github.com/andriizayac/uas_data_preprocess/blob/main/utils/rename_p4m_fn.R. Alternatively, you can run:
```
devtools::source_url("https://github.com/andriizayac/uas_data_pre
process/blob/main/utils/rename_p4m_fn.R?raw=TRUE")
```

2. To rename the files, execute the following commands in Rstudio:
```
library(tidyverse)
path <- 'path/to/multispectral_image_files/'
f <- list.fiels(path)
f_new <- rename_p4m(f)

names_old <- paste0(path, f)
names_new <- paste0(path, f_new)

# check the output
head(cbind(names_old, names_new))
# rename files in the folder
file.rename(names_old, names_new))
```

3. The screenshot below illustrates the change in the naming pattern of the image files. This method only works if the images have the same image labeling type as ours.

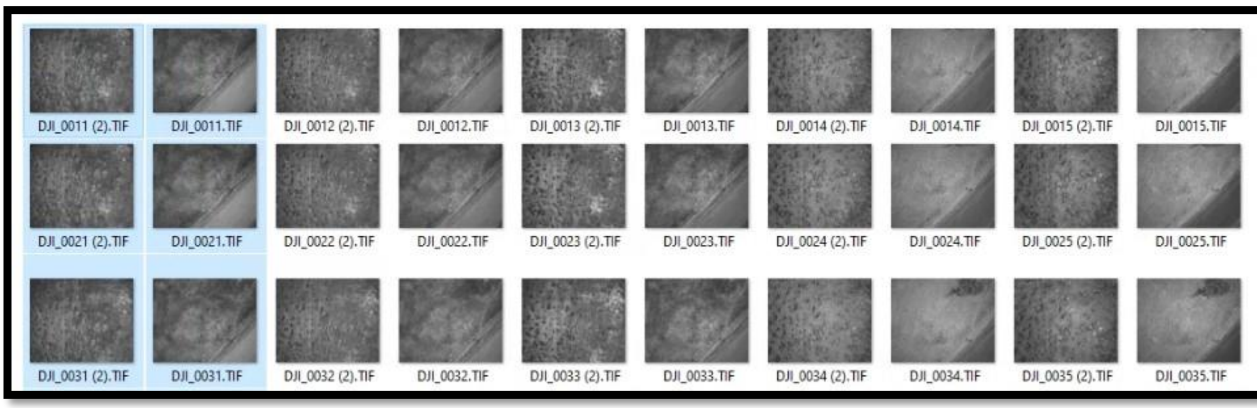


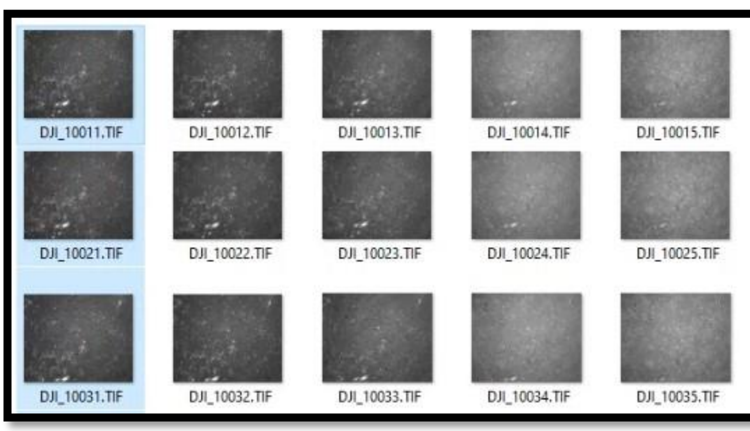
```
> head(cbind(names_old, names_new))
     names_old                           names_new
[1,] "D:/Valorie/test/DJI_0011 (2).TIF" "D:/Valorie/test/DJI_20011.TIF"
[2,] "D:/Valorie/test/DJI_0011.TIF"     "D:/Valorie/test/DJI_10011.TIF"
[3,] "D:/Valorie/test/DJI_0012 (2).TIF" "D:/Valorie/test/DJI_20012.TIF"
[4,] "D:/Valorie/test/DJI_0012.TIF"     "D:/Valorie/test/DJI_10012.TIF"
[5,] "D:/Valorie/test/DJI_0013 (2).TIF" "D:/Valorie/test/DJI_20013.TIF"
[6,] "D:/Valorie/test/DJI_0013.TIF"     "D:/Valorie/test/DJI_10013.TIF"
> # rename the file in the folder
> file.rename(names_old, names_new)
 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[25] TRUE TRUE TRUE TRUE TRUE TRUE
```

Multispectral Image Preparation (TIFs)

4. GCP Editor Pro currently will not use the TIF images from the multispectral flight. So, we will have to copy the images and the metadata (GPS) to use them in this program. The GCP.txt file we get from this process is all that we need. We will use the text file with the original TIF images in WebODM. We only need to place GCPs for Band 1, so we will select only those images to copy.

5. To select for Band 1 images, show image thumbnails and resize the window so it ends with the final band (5) as shown in the screenshots below. The first is if there are multiple images with the same name and have not been renamed yet. The first is after renaming all the images or if your data set did not have enough images to require duplicate names be used. Once they are all selected, copy them into a new separate folder labeled Band 1.





6. To convert from .TIF files to .JPG., open the application 'Windows PowerShell.'

7. This line of code will make a .JPG copy of the .TIF images, but all the metadata will be blank. Copy and paste the line below into Windows PowerShell, replacing the path name with the

pathway to the .TIF images. If this fails, rename the folders used in the pathway using underscores instead of spaces.

a. `magick mogrify -format jpg -quality 100 "D:\Name\Location\Multispectral_Images_Band1\*1.TIF"`

8. Leave the new .JPG and the old .TIF images in the same folder together until done with this step. The following line of code will copy metadata from the .TIF file to the new .JPG file. Copy and paste into Windows Powershell, replacing the path name with the same pathway.
   a. `exiftool -tagsfromfile %d%f.TIF -all:all -overwrite_original -ext jpg D:\Name\Location\Multispectral_Images_Band1\`

9. Move the JPGs into their separate folder labeled accordingly.

*****This is the end of image preparation for GCP Editor Pro.*****

10. Images must also be calibrated because Open Drone Map does not have a place to add the calibration panel images. We will use python code and the onboard sensor to calibrate the original TIF imagery.

11. Create an empty folder alongside (not inside) the folder with the original TIF images. Use the same name and add '_refl' at the end. Example: multispec_images <and> multispec_images_refl

12. We implemented image calibration using the on-board downwelling sensor (DLS) following the workflow based on the Micasense protocol, adapted to DJI Phantom 4 Multispectral (https://github.com/gdslab/p4m). In the following script, the first pathway is where it will take the original images from, and the second pathway is where the edited copies will go. To run the radiometric correction scripts, open WindowsPowerShell (or Command Prompt) from Anaconda and run the following commands:
   a. `conda activate P4M`
   b. `python p4m/raw2ref.py D:/Location/to/original/imagery/multispectral_images/ D:/Location/to/empty/folder/multispectral_images_refl/`

Ground Control Points

13. Using GCP Editor Pro, load in the .csv file with the GPS for the GCPs, and load in the images from the folder with the JPGs of just Band 1 from Multispectral steps 4-9. Then, follow the RGB steps 7-14 from above.

14. After placing GCP markers over the JPGs, download the GCP.txt file.

15. We had to use JPGs just to place markers, but we will use the original TIF files to process. Open the text file, go to Edit -> Replace, Find what -> [jpg], Replace with -> [TIF], Replace All, and Save.

16. Rename the document for specificity of site, date, and multispec. **Place this gcp.txt file in the folder with the original .TIF images.**

Open Drone Map (WebODM)

17. OpenDroneMap had an update in 2020 to accept TIFFs for multispectral and thermal data processing. Follow the RGB steps 15-21 from above with the options adjustment as follows.

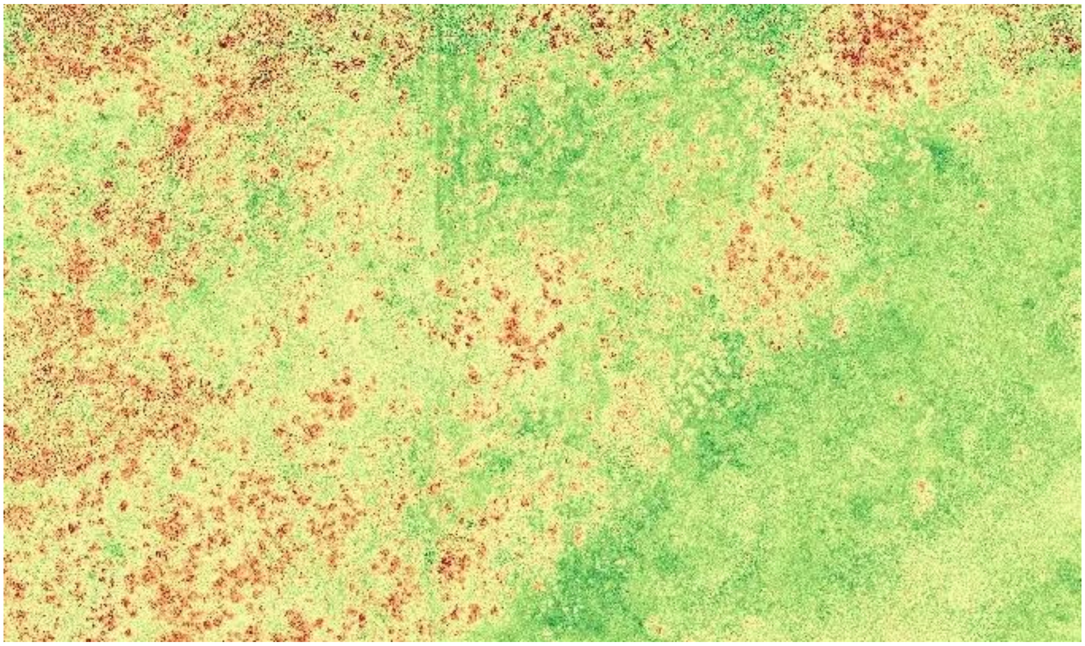
18. Options: (This is with a known ground resolution of 2cm. Replace with your resolution or remove if unknown.)
    a. auto-boundary:true, orthophoto-resolution:2, skip-3dmodel:true
        i. We do not need to pass the -radiometric-calibration: camera parameter because we have already calibrated the images reflectance prior to uploading to WebODM.

19. Download All Assets and the full combined point cloud as a .LAS file.

Multispectral Product Examples

Orthomosaic:

NDVI:



Plant Health Options (in WebODM)

Algorithm:

NDVI (Red) - Normalized Difference Vegetation Index: shows the amount of green vegetation. (N - R) / (N + R)

NDVI (Blue) - Normalized Difference Vegetation Index: shows the amount of green vegetation. (N - B) / (N + B)

ENDVI - Enhanced Normalized Difference Vegetation Index: is like NDVI, but uses Blue and Green bands instead of only Red to isolate plant health. ((N + G) - (2 * B)) / ((N + G) + (2 * B))

vNDVI - Visible NDVI: is an un-normalized index for RGB sensors using constants derived from citrus, grape, and sugarcane crop data. 0.5268 * ((R ** -0.1294) * (G ** 0.3389) * (B ** -0.3118))

VARI - Visual Atmospheric Resistance Index: shows the areas of vegetation. (G - R) / (G + R - B)

EXG - Excess Green Index: (derived from only the RGB bands) emphasizes the greenness of leafy crops such as potatoes. (2 * G) - (R + B)

TGI - Triangular Greenness Index: (derived from only the RGB bands) performs similarly to EXG but with improvements over certain environments. (G - 0.39) * (R - 0.61) * B

BAI - Burn Area Index: highlights burned land in the red to near-infrared spectrum. 1.0 / (((0.1 - R) ** 2) + ((0.06 - N) ** 2))

GLI - Green Leaf Index: shows greens leaves and stems. ((G * 2) - R - B) / ((G*2) + R + B)

GNDVI - Green Normalized Difference Vegetation Index: is similar to NDVI, but measures the green spectrum instead of red. (N - G) / (N + G)

GRVI - Green Ratio Vegetation Index: is sensitive to photosynthetic rates in forests. N / G

SAVI - Soil-Adjusted Vegetation Index: is similar to NDVI but attempts to remove the effects of soil areas using an adjustment factor (0.5). (1.5 * (N - R)) / (N + R + 0.5)

MNLI - Modified Non-Linear Index: improves the Non-Linear Index algorithm to account for soil areas. ((N ** 2 - R) * 1.5) / (N ** 2 + R + 0.5)

MSR - Modified Simple Ratio: is an improvement of the Simple Ratio (SR) index to be more sensitive to vegetation. ((N / R) - 1) / (sqrt(N / R) + 1)

RDVI - Renormalized Difference Vegetation Index: uses the difference between near-IR and red, plus NDVI to show areas of healthy vegetation. (N - R) / sqrt(N + R)

TDVI - Transformed Difference Vegetation Index: highlights vegetation cover in urban environments. 1.5 * ((N - R) / sqrt(N ** 2 + R + 0.5)

OSAVI - Optimized Soil Adjusted Vegetation Index: is based on SAVI, but tends to work better in areas with little vegetation where soil is visible. (N - R) / (N + R + 0.16)

LAI - Leaf Area Index: estimates foliage areas and predicts crop yields. 3.618 * (2.5 * (N - R) / (N + 6*R - 7.5*B + 1)) * 0.118

EVI - Enhanced Vegetation Index: is useful in areas where NDVI might saturate, by using blue wavelengths to correct soil signals. 2.5 * (N - R) / (N + 6*R - 7.5*B +1)

Additional Resources

https://github.com/OpenDroneMap/WebODM
https://docs.webodm.org/
https://odmbook.com/
https://courses.gisopencourseware.org/mod/book/view.php?id=500
- course & tutorial

https://courses.gisopencourseware.org/course/index.php
- open-source GIS courses

https://www.opendronemap.org/2020/01/choosing-good-opendronemap-parameters/
- Choosing OpenDroneMap parameters

https://community.opendronemap.org/t/mission-planning-for-heavily-forested-80-acres-on-slope/5664/2
- A page from the ODM forum with other useful resources and as a place to ask questions

https://github.com/OpenDroneMap
- WebODM section has helpful info and a link to the official documentation

WebODM Community Pages

Mission Planning:
https://community.opendronemap.org/t/mission-planning-for-heavily-forested-80-acres-on-slope/5664/2

Bowl Shaped Point Cloud:
https://community.opendronemap.org/t/problem-with-webodm-generated-dsm-dtm/1209/11

DEM not created:
https://community.opendronemap.org/t/orthophoto-surface-model-terrain-model-buttons-not-showing/572/2

Large Dataset:
https://community.opendronemap.org/t/large-dataset-need-advice-to-try-to-make-it-finish-processing/5252/3

Split-merge / Large dataset / time to merge:
https://community.opendronemap.org/t/webodm-clusterodm-and-gpus-oh-my/11479/9

Metashape and ODM comparison for parameter changes:
https://community.opendronemap.org/t/dem-orthophoto-fine-details-missing-when-compared-to-metashape/6401/5

Multispectral support:
https://www.opendronemap.org/2020/02/odm-0-9-8-adds-multispectral-16bit-tiffs-support-and-moar/
https://docs.opendronemap.org/multispectral/
https://community.opendronemap.org/t/creating-orthophoto-from-multispectral-images/4428
https://community.opendronemap.org/t/multispectral-photo-getting-started-question/3807
https://community.opendronemap.org/t/multispectral-processing-help-please/6341/4

References

ImageMagick Studio LLC. (2023). ImageMagick. Retrieved from https://imagemagick.org

Harvey, Phil. *ExifTool by Phil Harvey. Read, Write and Edit Meta Information*. 2020.

https://github.com/gdslab/p4m

https://github.com/andriizayac/uas_data_preprocess/blob/main/image_preprocess.ipynb

https://github.com/andriizayac/uas_data_preprocess/blob/main/utils/rename_p4m_fn.R