# MaRRS Lab UAS Photogrammetry Data Scripts Protocol

Creator: Clara Bird ([clara.birdferrer@gmail.com](mailto:clara.birdferrer@gmail.com))

Updated: 7/24/2019

## In this folder you will find…

Here is an overview of what the folders within the FlightDataScripts folder contain.

FlightDataScripts – This is the folder that contains all the scripts

DJI\_AltFromMetadata – This tool extracts the altitude from the image metadata of a DJI drone

Images (folder)

dji\_altitude.py

exiftool (application file)

read\_me.txt

ImageFlightLogData – This tool extracts the flight log data for each image in a Flight folder

File\_structure.txt

Instructions.txt

Perday\_imgdata.py

TimeCorrectionScript – This tool calculates the time correction factor from camera to GPS

Images (folder)

GPStime\_images.csv

Instructions.txt

TimeCorrectionCalculator.py

Imglist\_matchup.py – This script pulls the flight log data for the total length images.

gui\_collating\_alloptions.py – This script populates the image list with the GUI measurements.

Renaming\_code.py – This script adds prefixes to all the files in a folder.

## Extracting data for photogrammetry

TimeCorrectionScript , ImageFlightLogData, and matchup were created to be able to pull the altitude data for each image from the flight log using the time stamp of each image. To do this we first need to run the TimeCorrectionScipt to calculate the time difference between the camera time and the GPS time. Then we run ImageFlightLogData on every flight day (ex. 181224\_L) to pull the data for every image. After the images to be used for photogrammetry have been selected then matchup is used to pull the flight log data for just those images we want. Refer to the file structure below.

180226\_L 180226\_A

180227\_L 180227\_A

* Flight\_01 Flight\_01
* Flight\_02 Flight\_02
  + Flight\_Logs Flight\_Logs
    - 18022701\_L\_F2.GPX SYNLog-18-18-50\_27-02-2018.CSV
    - 180227\_L\_F2.TXT
  + KML Laser\_Altimeter
    - 18022700\_L\_F2.KML LA180227\_F2\_A.CSV
  + Sony Sony
    - 180227\_L\_F2\_DSC01262.ARW 180227\_L\_F2\_DSC01262.ARW
    - 180227\_L\_F2\_DSC01262.JPG 180227\_L\_F2\_DSC01262.JPG

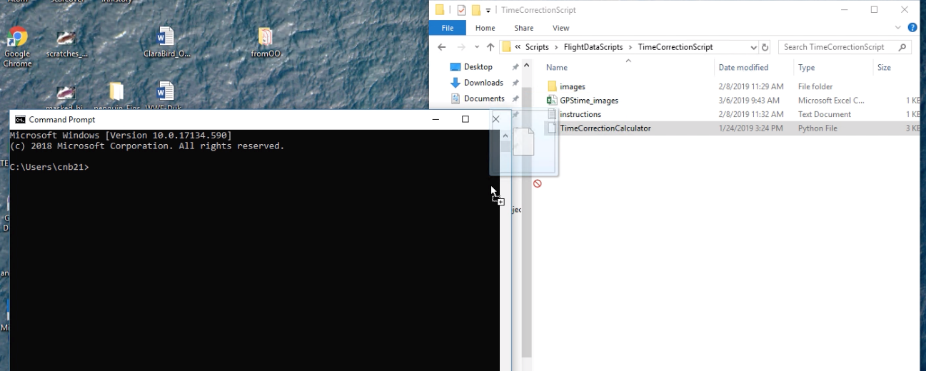
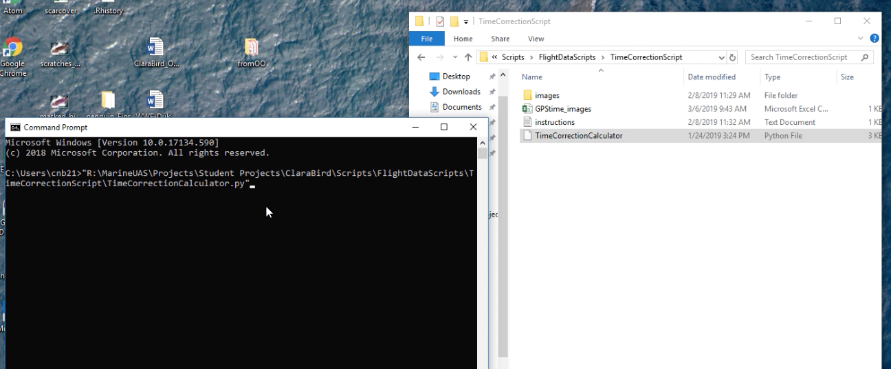
### TimeCorrectionScript Instructions

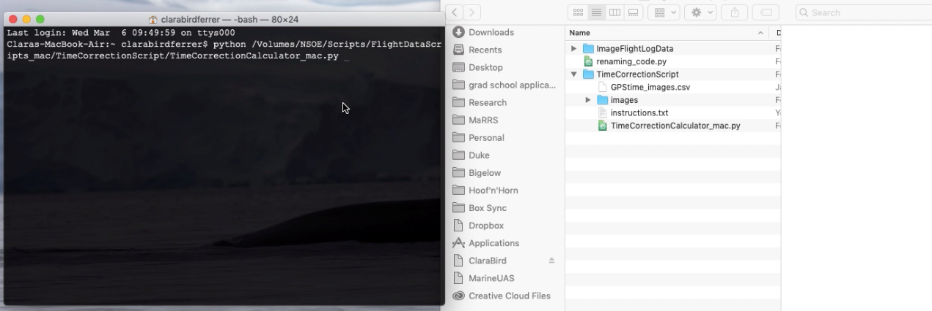
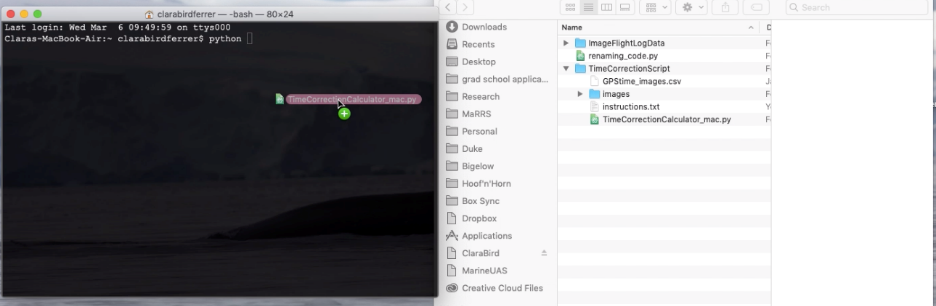
This script can be run from anywhere.

1. Copy and paste the GPS image (picture of the bad elf on the ipad/phone) for each individual flight into the ‘images’ folder. The information for each of these images will go in the .csv
2. Fill in the *GPStime\_images.csv* with the Flight, ImageName, and GPStime. It should look like this.

|  |  |  |
| --- | --- | --- |
| Flight | ImageName | GPStime |
| 181216\_L\_F2 | 181216\_L\_F2\_DSC01862.JPG | 13:25:08 |

1. Now it’s time to run the script!
   1. First you need to import easygui. Open terminal and type “pip install easygui”.
   2. Windows: open a command prompt window, drag and drop the script from the folder into the command line, then hit enter.



* 1. Mac: open a terminal window, type python, then drag and drop the script from the folder into the terminal window, then hit enter.
     1. Note: if you run python from something like Canopy on mac then open a Canopy Terminal window (Tools > Canopy Terminal). You can also run it from the Canopy editor.

1. A window will open asking you **to select the Time Corrections folder** where everything is saved. Once you select the folder the tool will just keep running.
2. When the script is done it will create a .csv in the same folder called TimeCorrections.csv, you need to copy and paste this .csv into the folder that contains all of the date\_aircraft folder before running the next tool.

### Image Flight Log Data Instructions

If you don’t have the right file structure the script will not work.

#### File Structure example (grey indicates that the folder name doesn’t matter, red indicates part of the setup)

DroneData (I will refer to this folder in the instructions but it can be named whatever you want)

181224\_A

Flight\_01

Flight\_Logs

Sony

Laser\_Altimeter

TimeCorrections.csv

PhotogrammetryMatchup

1. Create a folder within the DroneData folder named **PhotogrammetryMatchup**. The script will save a copy of the output .csv in this folder which is setup for the matchup script.
2. Make sure that the TimeCorrections.csv has been copied into the DroneData folder.
3. Running the script
   1. Drag and drop the script into the command prompt or terminal window and hit enter
      1. Unlike the other scripts this script does not need to be saved in any specific folder.
   2. A window will open asking you to select the **date\_aircraft folder**
   3. Another window will ask for the **aircraft type** (right now this is either alta or lemhex)
   4. You will need to repeat this for the script in each date\_aircraft folder.

* The script will output a .csv called ImageAltitudes\_YYMMDD\_Aircraft.csv (ex. ImageAltitudes\_181216\_L.csv) to the date\_aircraft folder and to the PhotogrammetryMatchup folder.

### Matchup Script Instructions

This script will pull the data for the images you have chosen for photogrammetry from the ImageMeasurements .csv.

#### File Structure Example(grey indicates that the folder name doesn’t matter, red indicates part of the setup)

DroneData

PhotogrammetryMatchup

ImageMeasurements\_181224\_L.csv

ImageMeasurements\_181216\_L.csv

ImageList.csv

1. Create the ImageList.csv, it should look like this. Column headers should be spelled and capitalized just like this.

|  |  |  |  |
| --- | --- | --- | --- |
| Individual | Animal\_ID | Img\_Category | Image |
| Mn180609\_L\_F2-01 | Mn180609\_L\_F2-01 | TL.1 | 180609\_L\_F2\_DSC00375.JPG |
|  | Mn180609\_L\_F2-01 | TL.2 | 180609\_L\_F2\_DSC00413.JPG |
|  | Mn180609\_L\_F2-01 | TL.3 | 180609\_L\_F2\_DSC00423.JPG |
|  | Mn180609\_L\_F2-01 | TL.4 |  |
|  | Mn180609\_L\_F2-01 | TL.5 |  |
|  | Mn180609\_L\_F2-01 | R-BH.1 |  |
|  | Mn180609\_L\_F2-01 | R-BH.2 |  |
|  | Mn180609\_L\_F2-01 | Fluke.1 |  |

1. Drag and drop the script into a command prompt/terminal window and hit enter (see the instructions for the TimeCorrectionScript for more detail on how to do this).
2. A window will open asking you to select the **PhotogrammetryMatchup** folder you want the script to run on.
3. A second window will open asking you to enter the **name of the Image List** file that is in the Photogrammetry Matchup folder. Make sure to include the **.csv** when entering the name.

### The altitude measurements

* BaroAlt: the barometer altitude or ‘ImgTimeNotInTable’ if the timestamp is missing or ‘badbarofile’ if the file is incorrectly formatted or empty.
* LaserAlt: the laser altitude or ‘ImgTimeNotInTable’
* Time130: if LaserAlt is 130 (error) then the script will look 5 seconds above and below the time stamp, if it finds a non-error value within the window then it will use that value for LaserAlt and Time130 will show the timestamp of the LaserAlt value that was used.

### GUI Matchup Script Instructions

This script will pull the data from all the GUI output .csvs into the image list.

This script combines all the output csvs and makes one large data frame that contains the image, animal ID, and all measurements. If you used different names for measurements these will just appear as extra columns with empty cell values for places where you didn’t use that measurement name.

*\*If you measure an angle make sure to include angle in the name of the measurement\**

**Note:** if there’s a repeated Object Name in a GUI output csv (like two rows of Total Length) the script will end and give you a message telling you to get rid of the duplicate. If you want both instances of the measurement included, then just alter the name slightly (for example: Total Length\_1).

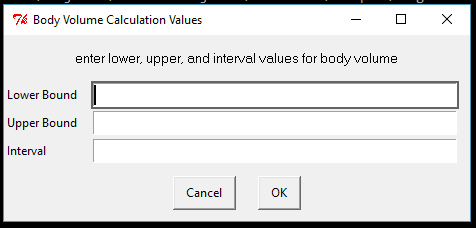
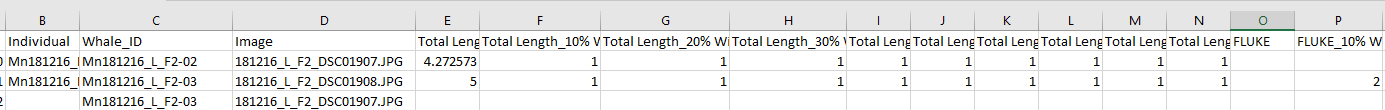
1. Create the ImageList.csv, see Matchup Script Instructions above for an example
2. Make sure that all the GUI output .csvs are either saved in one folder or saved in the IDs folder for each animal with a folder containing the ID folders for each individual.
   1. So either Or (if it’s in red it’s GUI output)

**GUIcsvs** **Individuals**

180227\_A\_F2\_DSC01262.CSV Mn180227\_A\_F2-01

180227\_A\_F4\_DSC01766.CSV 180227\_A\_F2\_DSC01262.JPG

180228\_A\_F1\_DSC01555.CSV 180227\_A\_F2\_DSC01262.CSV

1. Drag and drop the script into the command prompt/terminal window (see instructions for the TimeCorrectionScript for more detail) and hit enter
   1. This script does not need to be saved anywhere specific
2. A window will open asking if you have all the csvs in **one folder or in individual folders**.
3. A window will open asking if you want the **safety net**. The safety net means that the script will ask you to enter a sheet with the focal length per flight and the altitude per image and **it will recalculate the measurements using those values**. This could be useful if you were measuring a lot of images and didn’t want to take the time to manually enter the altitude and focal length.
4. A window will open asking if you want **body volume** to be calculated. Only click yes if you measured widths for Total Length. If you click yes another window will open asking for details.
   1. If you clicked yes a window will open asking you to enter **the lower and upper bounds** you want for body volume and the **interval**. So for example if you wanted volume calculated from 20%-80% you would enter 20 as the lower bound and 80 as the upper bound. The interval is the interval of total length that you want the segments to be calculated in. So, if you entered 5, the script would calculate body volume using intervals of 5%. Make sure that the interval is not less that the increments that you measured, so if you measured 10% increments 5% increments would not work.
5. A window will open asking if you want the output to only contain certain individuals. This is useful if you just want the measurements for a few of the individuals in the folder.
   1. If you choose yes then a window will open asking for the csv containing the list.
   2. Just one column, with the header spelled and capitalized like in this example.
6. A window will open asking you to enter the **name** you want for your **output**.
7. A window will open asking you to select the **folder where all the .csvs are saved** (‘GUIcsvs’ above) or the Individuals folder (‘Individuals’ above).
8. Another window will open asking you to select the folder where you want the **output to be saved.**
9. If you said that you wanted the safety net, then two more windows will open. If not, the script will start running.
   1. The first will ask for the csv containing **the altitudes per image**.
      1. Required columns: Image and Altitude (spelled and capitalized just like that).
      2. It’s up to you if you want to use the Barometer or Laster altitude
   2. The second will ask for the csv containing the **focal length per flight.**
      1. Make sure to spell and capitalize the same way as in the example table.
      2. Ex)
10. The script will save the output to the folder you selected in step 5.
    1.  The output will look like this (for example)

## DJI Altitude from Metadata Instructions

This script can be run from the folder that its in. Currently its written to be run using windows. DJI saves information about the flight as metadata in the image but the format is a little weird so this script will extract it.

1. Copy all the images you want the altitude for into the images folder.
2. Open the dji\_altitude.py file in python WIN or IDLE and run the code from there.
   1. To open, right click on the script file and select either “edit with PYTHONWIN” or “edit with IDLE”. Once the window is open then find the button to run the module/script.

* The tool will output a .csv named metadata\_altitudes.csv into the DJI\_AltFromMetadata folder, this csv will have the image name and the altitude.

Make sure that the exiftool application is in the folder, the script uses it.

## Renaming Code Instructions

This script contains lines of code that can be run in command prompt to rename/add a prefix to all the files within a folder. It was written to add the date\_aircraft\_flight prefix to all the images.

1. Open the script by right clicking and select one of the edit with \_\_\_ options (any of them is fine).
2. There are comments in the code telling you what each of the options does.
3. Once you have decided which one you want to use, edit the code so that the file path in the first line is correct and the prefix in the second line is correct. I’ve included an example below with the file path and the prefix colored. This example adds the prefix to images.

cd /D R:\MarineUAS\Projects\Client Projects\LTER\LTER 2018\LTER2018 - updated format\180125\_A\Flight\_01\Sony

for %a in (DSC\*) do ren "%a" "180125\_A\_F1\_%a"

1. Once you have edited the lines of code, copy and paste the two lines into a command prompt window and hit enter.
   1. Dragging and dropping the script file won’t work. This script contains command prompt commands that should be copied and pasted or typed directly into command prompt.