1. **Copy untrimmed photos to laptop**

**Trim Photo Set**

Trim the photo set to include only photo from first to last waypoint, excluding launch/landing waypoints, can be done visually or aided by CV software.

**Visually**

1. Examine the photos carefully to locate the location where the copter reaches it's first waypoint and turns to head to the second.
2. Examine the photos carefully to locate the location where the copter reaches it's last waypoint and turns to head to landing location.
3. Copy the photos between first and last turn to a separate folder within project folder.

* Remove the **microSD** card from the front of the **Pixhawk** unit and plug the microSD card into a computer.
* Download **bin** **files** from the card just like a flash drive.  Bin files can then be **converted into useful log files in mission planner**. Convert .bin files to .log files. And convert to **gpx and kmz files**.
* Remember to return the microSD card to the Pixhawk before flying!

1. **Using PhotoScan to Create Ungeoreferenced 3D Point Clouds**

* Open **PhotoScan**
* Go to **Workflow** > Add **Photos**
  + Add the trimmed flight photos
* Go to **Workflow** > **Align** **Photos**
  + Use Accuracy: High
  + Pair Selection: Generic
* Click **OK** and let it run. This could take 8+ hours depending on the number of photos
* Once PhotoScan has completed, **save** the job to your project folder.
* Build **dense point cloud** (workflow -> dense point cloud)
  + Quality: Medium
  + Depth Filtering: Aggressive
* Build **Mesh** (workflow-> build mesh)
  + Surface type: Height Field
  + Source Data: dense cloud
  + Polygon Count: medium
  + Interpolation: Enabled
* Generate Orthophoto: Export 🡪 Orthophoto: Export JPEG
* **Export** the **point cloud**
  + Go to **File** > **Export** **Points**: and select a **PLY** file for export. Generate an **ASCII** **PLY** **file**, not a binary file, and **export colors**, but not normals
* **Export** the **cameras**
  + Go to **Tools** > **Export** **Cameras**
  + Generate an **XML** **file** or other file type, for example CHAN.
  + Export the point cloud as a **Bundler .OUT file**, says Yes or OK when prompted that some camera information will be lost, it is OK.

**ASK HECTOR WHAT TO DO NEXT:**

**Interpolate\_gps\_positions.py ? 🡪 gives you the cam\_xyz\_file that is used to create georeference point cloud in Photoscan**

**GPS\_camera\_telem\_interp.py ?**

Open Anaconda Command Prompt and type: python –m ecosynth.

* If you get local host error, go to **Windows Task Manager and end appropriate PID: Performance Tab 🡪 Resource Monitor (button at the bottom) 🡪 Network Tab (Listening Ports)**
* Open Anaconda Command Prompt again, and type ipython –m ecosynth
* If not opening, go to **C:\Users\jmueller\Documents\EcosynthSoftware** and **doubleclick on the setup python file. Then re-type** ipython –m ecosynth into Anaconda

In Ecosynth Aerial, click on Stage 3: Post-Processing and Analysis Tool:

* Output Directory
* Launch Elevation
* Log file (directly from copter) for example: 34.log (is a txt)
* Output file (from Photoscan)
* (Optional) Dense Ply File

**Key Outputs:**

* Georeferenced and Noise Filtered **PLY File**
* EcosynthBrowser-Ready PLY File
* Raster Images
* Helmert Transform Parameters
* Point Cloud Characterization Statistics

**Convert Telemetry, using Ecosynth Script**

Put **Ardu log file** (ex. 34.log) into same folder as **convert\_telemetry.py** script (received from Will in an email):

(1) Create a new folder and place the log file and convert\_telemetry.py script file in that folder

(2) Open the Anaconda Command Prompt

(3) Using the prompt, change into the folder where you've placed your log file and your convert\_telemetry.py script using the "dir" command to list folders and "cd folder\_name" to change into a given folder

(4) Using the prompt, type the following command in the window:  "ipython convert\_telemetry.py log\_name output\_name msl"

    Notes for step 4:

    the last three are arguments you are passing into the convert\_telemetry script that you should replace

**log\_name** - replace with the name and extension of your log file (e.g. "flight.log")

**output\_name** - replace with the name and extension of your new telemetry file (e.g. "telemetry.txt")

**msl** - replace with the mean sea level launch elevation value (in meters) of the UAV  (e.g. "50.6")

    no need to use quotes for any of the arguments - just make sure the last argument is a number

(5) You should be done after hitting enter - check your folder that the output file has been created in the same folder and if so, check that the file contains a series of lines with three numbers in each row

OUTPUT: Creates GPS\_position.txt, based in Arducopter .log file

<http://www.earthpoint.us/Convert.aspx>

Interpolate\_gps\_positions.py

(4) Using the prompt, type the following command in the window:  "ipython interpolate\_gps\_positions.py log\_name msl cam\_xyz\_output\_name"

Notes for step 4:

**log\_name** - replace with the name and extension of your log file (e.g. "flight.log")

**msl** - replace with the mean sea level launch elevation value (in meters) of the UAV  (e.g. "50.6")

**cam\_xyz\_output\_name** - replace with name and extension of your photoscan interpolated gps positions (e.g. "pscan\_interpolated.txt")

And then repeat step (5) where the output should look like:

# <label> <x> <y> <z>

a.JPG 364754.663290 4305831.506730 136.543925

...

where the photo name is the left-most datum and the lat, long, elevation are to the right

**In Ecosynth, go to Stage 3: Post-Processing and Analysis Tool:**

A Photoscan ground control file needs to be produced from the Arducopter .log file **and** the folder containing the trimmed photo set

Point Cloud Instructions:

Dense Point cloud. Instructions here: <http://www.agisoft.ru/tutorials/photoscan/05/>

**Ply to rhino Script:**

* 1. Save the ply file as a xyz file by right-clicking on the ply file, copying and pasting it, and then renaming the file extension to xyz
  2. Open new xyz file in Notepad ++ and take out the first 10 lines. Save
  3. Open Rhino 5 and type \_EditPythonScript
  4. Open script through File -> Open
  5. Run

1. Run the **Convert\_Telemetry script**. Probably in UTM, NAD 1983, Zone 19 projection. Save as a .csv with header and FIP column. Import into ArcMap, to determine how many rows to trim. Delete rows in xls. When done, delete FIP column, too. Save as GPS\_File.txt. Then TAKE OUT .txt extension!

**testing**

1. 