

UAS Data Management Standard Operating Procedure

Pre-requisites

Before making data available for upload, the following information must be defined in the wheatgenetics database:

1. experiment table:

An entry must be made for each experiment for which flight data is to be collected.

The `experiment_id` must be composed only of alphanumeric characters and optionally a hyphen-minus separator. The hyphen-minus character (Unicode character 002D) can be used as a separator within the `experiment_id`. No other characters are allowed.

2. `htp_instrument` table:

An entry must be made for each camera (`sensor_id`) that is used for collecting flight data.

The `sensor_id` must be composed only of alphanumeric characters and optionally a hyphen-minus separator. The hyphen-minus character (Unicode character 002D) can be used as a separator within the `sensor_id`. No other characters are allowed.

Database updates can be made by users with access to the wheatgenetics database and who are familiar with the sensor types to be added.

Definitions

Flight Data Set- a flight data set contains all images and log files produced by a single flight. There may be multiple flights per experiment.

Note: For Micasense, a new flight folder is created whenever the camera is powered up and will have a name of the form `nnnnSET`. For example `Flight1=0000SET`, `Flight2 = 0001SET` and so on. Within each flight folder, a subfolder containing the images is created which has a name of the form `nnn`. The first folder will be named `000`. If more than 999 images are stored, a second image folder (`001`) is created . Two log files are also created: `paramlog.dat` and `diag.dat`. Each `nnnnSET` is considered to be a single flight. In addition to the Micasense files, there will also be a UAV log file (i.e. the DJI log file) for each flight.

Data Submission

Micasense RedEdge Data Sets

1. Create a flight folder for each flight data set on the field laptop that is used for data collection named according to the standard naming convention for flight data sets:

```
<dateyyyymmdd>_<location>_<experiments>_<camera_type>_  
<planned_elevation>_<lens_angle>_<image_type>_<flight_number>
```

Examples: 20180404_18ASH_BYD0BYD2_Rededge_20m_-90_Still_Flight1

2. Make sure that the following data files are present for each flight data set:
 - a. One or more nnnnSET folders containing
 - i. Images
 - ii. diag.dat
 - iii. paramlog.dat
 - b. (DJI)log file with a naming format yyyy-mm-dd_hh-mm-sss_v2.csv.
Example: 2017-03-02_11-14-39_v2.csv

3. Upload each flight data set folder to the Beocat uav_incoming folder:
/bulk/jpoland/images/staging/uav_incoming

DJI X5R Video Data Sets

1. Create a flight folder for each flight data set on the field laptop that is used for data collection named according to the standard naming convention for flight data sets:

```
<dateyyyymmdd>_<location>_<experiments>_<camera_type>_  
<planned_elevation>_<lens_angle>_<image_type>_<flight_number>
```

Examples: 20180508_18ASH_AM3_X5R_6m_-60_video_Flight1

2. Make sure that the following data files are present for each flight data set:
 - a. One folder per range containing (.dng) images with the following folder naming format DJI_<camera_id>_<range>_<dateyyyymmdd>

Examples: DJI_A01733_C001_20180508
DJI_A01733_C002_20180508

- b. DJI log file with a naming format yyyy-mm-dd_hh-mm-sss_v2.csv.
There should be only one log file per flight.

Example: 2017-03-02_11-14-39_v2.csv

- c. One video file per range (.MP4)

Examples: DJI_0001.MP4
DJI_0002.MP4

3. Upload each flight data set folder to the Beocat uav_incoming folder:
/bulk/jpoland/images/staging/X5R_staging

Data Archival – Micasense RedEdge Data Sets

1. Run the program to check and organize the flight data folder by typing the command in a terminal window:

```
python collate_micasense_flight_data.py
```

2. Move the renamed flight data set folders into the uav_staging folder on Beocat: /bulk/jpoland/images/staging/uav_staging

The uav_staging folder will be polled automatically (nominally) once a day to check for the presence of new data sets that need to be archived.

Each data set found in the uav_staging folder will be processed automatically. Note that the exact processing steps are camera platform dependent.

3. Verify that all required file types are present in the folder:
 - a. Image sub-folders
 - b. UAV log file
 - c. Camera-specific log files e.g. diag.dat and paramlog.dat for Micasense data sets
4. Extract the image date/time and GPS position metadata from the Micasense (.tif) image EXIF.

5. Rename each image according to the image naming convention:

```
<camera_manufacturer>_<sensor_id>_<date>_<time>_<original_image_name>
```

6. Store the image metadata for each data set in wheatgenetics.uas_images (one row per image.)
7. Store the flight summary information for each data set in the wheatgenetics.uas_run table (one row per flight data set.)
8. Move each processed data set from the uav_staging folder to the archive folder (/bulk/jpoland/images/uas).

9. **Future:** Generate an image of the polygon defining the area covered by each flight, superimposed on a polygon, which defines the area covered by the associated experiment, for QC review. Store the PDF image file in the uav_processed folder. The name of the image file should be of the following format:

<date>_<start_time>_<experiment>_<planned_elevation>_<camera_type>_<camera_angle>_<image_format>

The QC image file may cover multiple flights for the same experiment.

Note: The UAV operator should ideally perform the QC review in order to make sure that all areas included in the mission plan have been covered and there are no missing areas.

Data Archival – DJI X5R Data Sets

TBD.

Error Scenarios

1. A solution is required for determining whether the file folders in the uav_staging folder have been completely and correctly copied into that folder.

Proposed solution: Before the flight data folder is copied into the uav_staging folder, generate a manifest (text) file which is a list of all files, sub-folders and their image files, contained in the flight data set folder, and store it in the flight data set folder. This can be generated using the Cygwin Linux command:

```
find -L <flight_folder_name> > manifest.txt
```

When the folder is pre-processed, the archive process will check that all files listed in the manifest file are present in the flight data set folder.

2. A solution is required to support remote upload of data from locations where Internet connectivity is poor.

Appendices Naming Conventions

Data Set Folder Hierarchy

The folder/file hierarchy for a specific flight data set will be:

- Flight Data Set Folder
 - Image Sub-Folder 1
 - Image 1
 - Image 2
 - Image n
 - Image Sub-Folder 2
 - Image 1
 - Image 2
 - Image n
 - Image Sub-Folder 3
 - Image 1
 - Image 2
 - Image n
 - Log file type 1
 - Log file n

Note: The exact set of raw data files preserved will be camera system dependent.

Flight Data Set Folder Name

Each flight data set should be named according to the following format:

```
<date>_<start_time>_<planned_elevation>_<camera_type>_<camera_angle>_<image_format>_<flight_sequence_number>
```

Note that the experiment, planned_elevation, camera_type, and camera angle are user inputs.

date: yyyyymmdd e.g. 20170701. The date of the flight.

start_time: hhmmss e.g. 11:30:25 the start time of the flight. Initially, this will be entered manually. Ultimately, it will be derived from the start time of the associated flight log.

planned_elevation: nnm planned flight altitude in meters e.g. 15m

camera_type: Valid types are:

- X5R
- X3
- X3NIR
- MRE (for the Micasense RedEdge),
- FLIR
- NANO

Note: Other types will be added as new camera platforms are deployed.

camera angle: numeric value in degrees. (Note that this will always be a negative value so there is no need for a minus sign.)

image_format: v=video s=still

flight_sequence_number: A number starting at 1 and incrementing by 1 which is assigned whenever there are multiple flights with the same combination of date, experiment, planned_elevation, and camera_angle.

Examples:

```
20170701_113031_15m_X5R_70_v_1  
20170701_114015_15m_X5R_70_v_2
```

```
20170702_120321_30m_X3_90_s_1
```

```
20170703_133345_15m_MRE_70_s_1
```

```
20170704_153618_15m_FLIR_90_s_1
```

Flight Data Set File Folder Contents

1. One or more sub-folders containing images. (All image sub-folders that are produced natively by the different cameras will be moved up directly under the top level data set folder, i.e. any sub-folder hierarchy will be eliminated)

2. Camera-Specific Log files - Log files will retain their original names.

Image Sub Folder Name

Image sub-folders will be named according to the following format:

```
<camera manufacturer>_<sensor_id>_<sub-folder number>_<date>
```

camera_manufacturer: Valid types are:

- DJI
- FLIR - FLIR Systems
- MICA - Micasense
- HW - Headwall

sensor_id: - A valid sensor_id obtained from the htp_instrument database table. The sensor_id must be composed only of alphanumeric characters and optionally a hyphen-minus separator. The hyphen-minus character (Unicode character 002D) can be used as a separator within the sensor_id. No other characters are allowed.

sub-folder number: The original lowest-level sub-folder that contains image files. For example, Micasense sub-folder numbers will be 000,001...nnn.

date: yyyyymmdd e.g. 20170701 The date of the flight.

Note: Other types will be added as new camera platforms are deployed.

Examples:

```
DJI_A01733_C001_20170701
DJI_A01733_C002_20170701
DJI_A01733_C003_20170701

MRE_1620061_C000_20170701
MRE_1620061_C001_20170701
```

Image File Name

Most camera image files will need to be renamed in order to make them unique. All parts of the new image file name should be derived from the EXIF associated with the image.

Image files should be renamed according to the following format

```
<camera_manufacturer>_<sensor_id>_<image_date>_<image_time>_<original_image_name>
```

camera_manufacturer: Valid types are:

- DJI
- MICA -Micasense
- FLIR -FLIR Systems
- HW -Headwall

image_date: yyyyymmdd e.g. 20170701

image_time: hhmmss e.g. 11:30:25

original_image_name: The original name assigned by the camera system e.g. IMG_001_1.tif

Examples:

```
DJI_A01733_20170701_HHMMSS_IMG_000001.dng
MRE_1620061_20170701_HHMMSS_IMG_0001_1.tif
```