Unmanned Aerial Vehicle Remote Sensing Tools

Chris Reudenbach

2018-02-27

Contents

This version of the document was produced on *Di Feb 27 2018* using `uavRst` version **0.3.2**

Installation

The `uavRst` package needs the `library("devtools")`:

```
devtools::install_github("gisma/uavRst")
```

Some remarks about UAV flying

The basic idea is to provide an easy to use workflow for controlling rtf UAVs from planning and flying autonoumous surveys up to the derivation and postclassification of the data. The focus is set on retrieving your own data for answering your own questions keeping in mind the specific advantages and disadvantages of UAV derived data. RTF UAV#s provide very high spatial and very low spectral resolutions. Additionally the point cloud is only genereated by visible surface structures.

The central script for survey planning is `makeFP`. It creates either intermediate flight control files for the dji phantom x UAVs or ready to upload control files for MAVLINK conform flightcontrollers like the Pixhawk as used in the 3DR Solo uav.

The dji control files are designed for using with the propietary litchi flight control app exchange format, while the 3DR Solo files are using the MAVLINK common message set, that is used by the PixHawk flight controller family.

DJI Phantom family

The reason using DJI is their absolute straightforward usage. Everybody can fly with a DJI but the price ist a hermetically closed system. Only the litchi app provides additionally to a cloud based mission planer an offline/standalone interface to upload a csv formated waypoint file for autonomous flights to the Phantom.

PixHawk/3DR Solo:

The open uav community is focussed on the PixHawk autopilot unit and the Mission Planner software. It is well documented and serveral APIs are provided. Nevertheless a terrain following autonomous flight planning tool is not available. murcs creates static implementation of the MAV format that is ready to be uploaded directly on the Pixhawk controller using the upload2Solo function.

Survey Area

To define a flight area you have to provide either 4 Points (or 3 lines). You may take more complex vectors like a multi point polygon, but only the first 4 coordinates x1, x2, x3 and x4 (for the launching position) are used in exactly this order. If you take a rectangle the 4th corner coordinate will be the launching point!

The concept is looking like the following sketch.

```
 x2------x3           x2-------x1
| a                  /
|                   /
|   x4             / x4
|  /             / /
x1/             x3/
```

This coordinates the length of the line and the angle are used to calculate extend and paralells of the flightplan according to the flight altitude, overlap etc. Note the flight direction depends on the order of the points. If the flightPlanMode is equal tracks.

The result look like this.

```
#--#  #-->             #--#  #
|  |  |               /  /  /
|  |  |              /  /  /
|  |  |             /  /  /
#  #--#         <--#  #--#
```

If flightPlanMode is equal waypoints the result is an equal spatial distribution of waypoints:

```
#--#  #-->             #--#  #
|  |  |               /  /  /
#  #  #              #  #  #
|  |  |             /  /  /
#  #--#         <--#  #--#
```

waypoints is optimal for autonoumous flights under calm conditions in complex terrain because the camara takes a picture at every waypoint track is optimal for relatively plain areas and automatically triggered picture capturing Note: Automatically picture capturing in a time interval works only within the range of the remote control. because the the uav needs a trigger signal for taking pictures.

Terrain Following flightplan

The followSurface switch is used to adapt the fixed flight altitude into a terrain following flight altitude.

NOTE: You have to be aware that the DJI uav is calibrating the altitude at the launch position in the field! So you need either a correct coordinate altitude or a high resolution DEM to get a good! estimation of the lauch position and altitude. You must choose a clearly defined and reliable launching position both in the map and the field. If you fail I made the experience that the aircraft probably will hit the terrain…

How it works. Let us assume a defined flightaltitude of 50 m. According to the launching point altitude the uav will act like the following sketch shows:

```
 ............... x-(uav)-x ........... uav started at 30 m altitude results in
                                          a "real" flight altitude of 30m + 50m => 80m



                 ___60m____
                |          |
       30m _x__|           |
      ____|                |___
  ___|                         |____


                ___60m____
    ..........|          |............ uav started at 0 m altitude results in
           ___|          |___          "real" flight altitude of 50m above 0m
      ____|                  |
  ___|                       |__x__ 0m
```

To avoid negative impacts from the P3 auto calibration, the launch altitude is used to correct the flight altitude according to: maximumAltitude_of_surveyArea + altitude_of_launchposition So the adapted flight altitude is always seen as the flight altitude above the highest terrain altitude:

“‘

……………………………… real altitude of uav 110 m

```
           ___60m____
          |          |
     30m _x_|        |___
    ____|                |
 ___|                    |______
```

```
To get a fixed scale flight the launch altitude is used to correct the flight altitude according to maximumAl
```

```
           ..........
          |          |
      ....|          |....
   ....|     ___60m____    |
```

….| | | |……. real altitude of uav 50m 30m x| | / /
| |x___ 0m

“‘

Usecase EPSG:3031

```
<Layer>

</Layer>
```

First the primary arguments:

- ResourceURL
- format

parameter format for use in preview

Now we need an appropriate structure for these parameters to make it less complicated to feed the projView funktion with this information. This is a simple nested list. If we fill it with the values it looks like this:

Some examples

First we need to load some packages and get data:

Please note you have to activate the layers using the layer control in the upper right corner.

For the NASA EarthData you may use the visEarthPole() function.

Final remarks

There are still a lot of limitations that need addressing:

In future releases I would like to

- get around the issue that you have to do a manual job in parameter sampling
- integrate a defined plugin concept for own function that provides correct parameter lists
- integrate a fully working localTile functionality
- …

I hope that you will find projvView as useful.

If you have any feedback, please don't hesitate to contact me.

Bug reports should be filed at https://github.com/environmentalinformatics-marburg/mapview/issues

Cheers

Chris