A Field Tool to Assess the Accuracy of Automated Drone Flights

Robert Lauder

Ulster University

Abstract

[The abstract should be one paragraph of between 150 and 250 words. It is not indented. Section titles, such as the word Abstract above, are not considered headings so they don’t use bold heading format. Instead, use the Section Title style. This style automatically starts your section on a new page, so you don’t have to add page breaks. Note that all of the styles for this template are available on the Home tab of the ribbon, in the Styles gallery.]

Keywords: [Click here to add keywords.]

Introduction

This tool has been created to quickly help drone mappers check the accuracy of their flights whilst still in the field so that they know whether they need to redo part or all of the flight before they leave site. To describe the script fully and give context to the scenarios in which I will be describing, I need to give a short introduction into the workflow of drone mapping.

Creating orthophotos with a drone requires aerial photos to be taken at set intervals along a calculated flight pattern. The automated flight and image capture is guided by the drones inbuilt GPS unit, this will also geotag all images taken. The interval and flight pattern will vary depending on the drones altitude, the camera specification, and the needs of the user. In general, the more complex a surface is, the more overlap is required to get an accurate representation of that surface, the user will also likely benefit from a double grid pattern, this provides more points that can be matched between images (tie points). There are many other factors to consider when creating a plan, like ground sample distance (the size of the pixels), speed, flight duration and topography but they are not in the scope of this project. The basic concept to be aware of this that the flight plan instructs the drone where to fly, how to fly and the exact positions in that flight that it must take photos. These positions can also be downloaded from the mission planner and saved as a KMZ file for preview in Google Earth. It is this flight plan KMZ as well as the actual images captured by the drone that will be used to assess the accuracy of the flight, by comparing the flight plan coordinates with the geotagged image coordinates. For most of this project the flight plan coordinates will be called PreCoordinates because they are coordinates from before the flight, and the drone image coordinates will be called PostCoordinates because they are the result of the flight. In an ideal world there would be no difference between the PreCoordinates and PostCoordinates, but that is not the case as you will see. Drone flyers are always at the mercy of the weather and even when conditions seem pleasant on the ground at your take-off location, the winds can be or can quickly change to much faster speeds at higher altitudes. Different drones have different capabilities when it comes to counteracting these winds, but it is common for them to deviate from the flight plan in such scenarios. If the drone deviates too far off the plan, it will exit the automated flight and ask the user to land the drone due to high winds. The user can instruct the drone to continue the flight plan from the last checkpoint (last accurate interval where a photo was taken), or manually return the drone to land where they can either wait for weather to improve otherwise abandon the flight completely. If the flight is continued from the last checkpoint there is a high change of duplicate images being taken as the drone repositions back onto the flight plan and retakes the images along a portion of the flight where it had previously gone astray, before continuing with the rest of the plan as normal. If the flight was completely abandoned, there will be fewer images captured than were scheduled in the flight plan.

These different situations need to be highlighted as they guide the flow control in the accuracy assessment portion of the script, in which the number of images and the number of points in the flight plan must align so that points [1, 2, 3] are compared against images [1, 2, 3] rather than [1, 3, 2], which would not give valid results.

Setup and Installation

A setup/installation section that describes how to install your code, including a list of the main dependencies. You should also include a link to your repository in this section – this is what I will use to evaluate your code for Part 2 of the Assessment. If you have developed your code with specific test data, you should clearly instruct the user where they can acquire that test data (or similar test data). You do not necessarily have to provide these datasets in your GitHub repository, but they should be easy for the user to track down.

Four sets of test data are included in the *Test\_Images* folder. Each one represents a different scenario that can happen when mapping. The only requirements for the script to work are that there are drone images in JPG/JPEG format, and there is a KMZ file containing the flight plan.

*1\_Standard*: the number of images match the flight plan. The entire flight was completed as planned.

*2\_Duplicate*: the number of images does not match the flight plan. Too many images, there are duplicate images.

*3\_AbandonedFlight*: the number of images does not match the flight plan. Not enough images, the flight plan was not completed. There are no duplicate images.

*4\_DuplicateAbandonedFlight*: the number of images does not match the flight plan. Not enough images, the flight plan was not completed. There are also duplicate images.

Open Anaconda, activate the environment that came with the repository then open the python terminal via that environment. Within each folder there is already a python file. Navigate to the directory of the required test then simply run the script.

Cd [directory]

Python [script].py

Methods

A methods section that clearly explains what your code does – if you’re performing a certain kind of analysis, this should explain the steps of the analysis and the theory behind it. This section should be written in the style of a methods section for a journal article or technical report.

Results

This will depend on the inputs given and the outputs requested.

Troubleshooting

A section that provides some troubleshooting advice in case things go wrong.

References

Last Name, F. M. (Year). Article Title. *Journal Title*, Pages From - To.

Last Name, F. M. (Year). *Book Title.* City Name: Publisher Name.