

Pix4D iOS developer exercise

Overview

The goal of this exercise is to create and fly a mission with a drone simulator and then download and show the captured images in the app.

The exercise consists of 3 tasks which can be done independently and in the desired order. Further development should be made in Swift and where needed in Objective C / C++.

It is highly encouraged to write code adhering to best practices, handle possible errors and write automated tests.

Please also provide a short explanation of your implementation.

A skeleton Xcode project and several frameworks are provided:

P4DDroneInterface:

- abstract classes and data structures for interacting with a generic drone

P4DDroneInterfaceSimulator:

- a concrete implementation of a simulated drone

P4DFlightPlanner:

- utility functions for creating a flight plan

Objective

1. Drone mission creation

Add code to the method **createDroneMissionWithCapturePlan:cameraParameters:** to create a drone grid mission according to the given capture plan and camera parameters.

Methods from **P4DFlightPlanner GeoUtils** can be used to compute geo locations of the waypoints so that the correct image overlap is achieved.

A simple implementation is provided which creates a flight plan on the edges of the given rectangle. This will however not respect the given overlap parameters and must be changed.

Requirements:

- The lines should be evenly distributed in the rectangle.
- Each line should have a photo at the start and end.
- The resulting front and side overlap should not be smaller than the given requirements, but can be higher.
- The taken photos should be oriented in the direction of flight.
- Correct results should be generated if the parameters are changed.

On the right side is an explanation for image front and side overlap.

For the given parameters, the final result should be similar to the image below left.

Resulting Mission

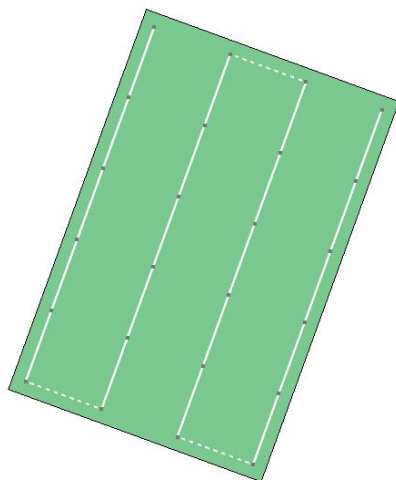
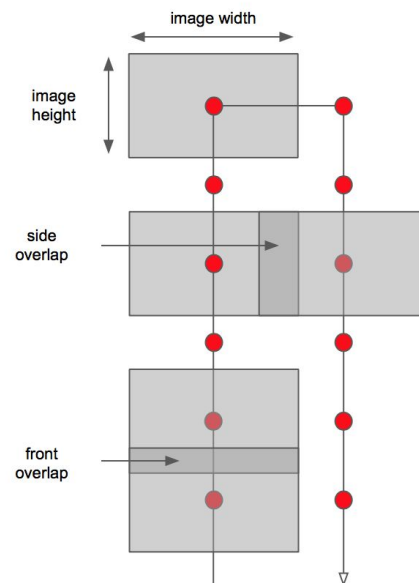


Image Overlap



2. Display information on the map

Show the various capture plan and mission elements on the map:

- capture area as a rectangle overlay with alpha
- mission waypoints as dots
- mission waypoints connected as a line
- taken photos as images with correct rotation using the provided photo_marker.png
- flight path of the drone as it moves as a poly line

Colors should be chosen so that all elements are clearly visible on the map.

3. Download and show images from the drone

After a mission is finished, use the **P4DCamera downloadMedia:data:completion** method to download the captured images and save to the app documents directory.

During the mission, the drone will report which images it has taken through the **photoTakenCallback** block. The info should be saved by the app and then passed to the downloadMedia method.

The **photoTakenCallback** is not always reliable and it can happen that it will not always be called as expected or can have a considerable delay.

Add UI as needed, to show a dialog with progress of the image download and a new view where the images can be seen, for example as a collection view.