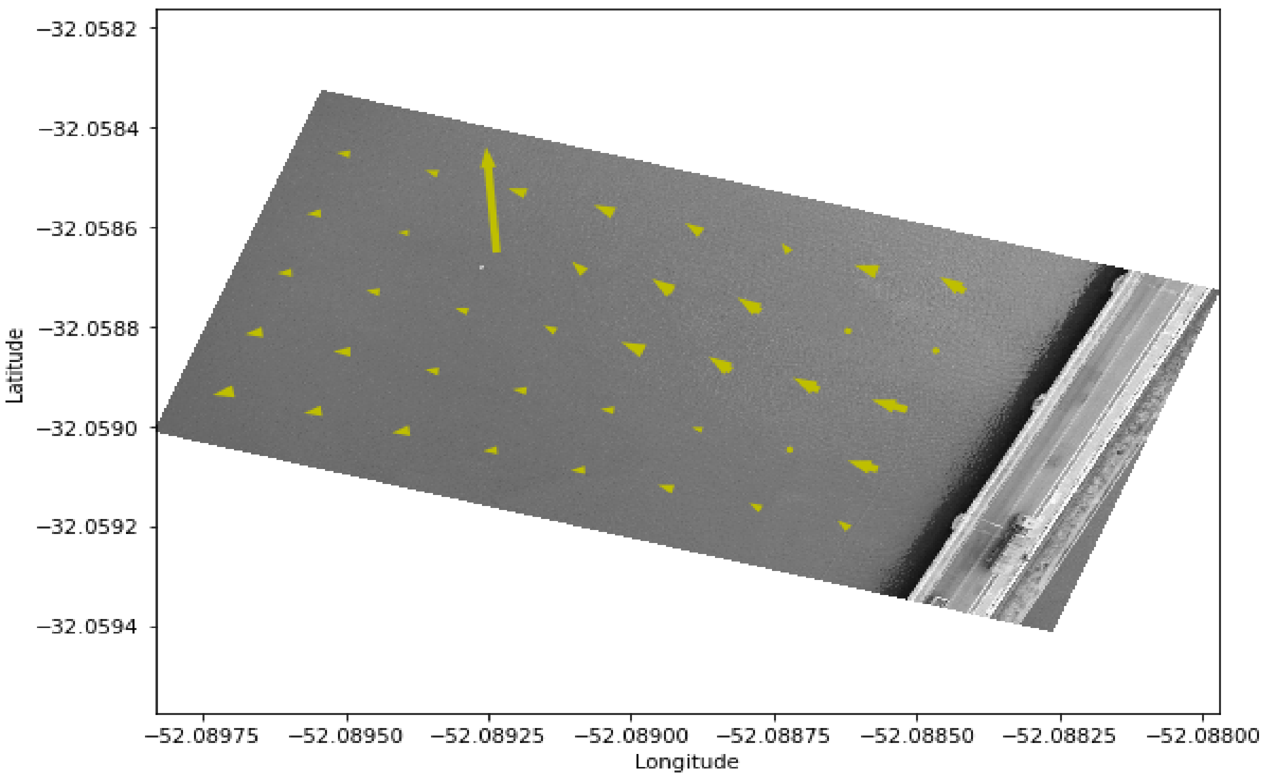
PyDroneCurrents Manual



Flavia Macedo

Laboratório de Oceanografia Costeira e Estuarina - Locoste

Instituto de Oceanografia - IO

Universidade Federal do Rio Grande - FURG

1. **Introduction**

This manual was developed to support the use of the PyDroneCurrents function package, which aims to calculate the speed of surface currents through drone images. The method used to extract currents from images is based on the wave dispersion relationship and the present algorithms are adapted from routines developed by Carrasco, 2019 found on the page: https://github.com/RubenCarrascoAlvarez/CopterCurrents.

This tool package contains this manual which briefly explains how to use the package, a model video in MP4 format to test the package, and a reference notebook in .ipynb format to run in the JupyterLab environment indicating how to use each function of the package. to generate results.

To obtain surface currents using this method, it is first necessary to record a video of the water surface (Example attached). To be recorded, this video requires some prerequisites, which are:

- The drone camera must contain the stabilization gimbal;
- The drone must film perpendicular to the water surface, in a nadir position (90°);
- During filming, the drone must remain completely static;
- Film for at least 10 seconds;
- The waves must be visible in the image;

Before obtaining the images, it is recommended to calibrate the camera, which is an optional process to run the package, since the notebook contains reference values for camera calibration parameters. For a study where there is a need for greater precision, proper calibration becomes important.

1.1 Camera calibration:

To perform current analysis through images it is necessary to obtain the camera calibration parameters, since it is through these values that it is possible to generate the pixel values in meters. These parameters are obtained through the camera calibration process, and are of great importance for the purpose of removing lens distortion.

There is more than one way to calibrate a camera, one of which is through the camera's field of view (FOV). This method consists of obtaining the camera calibration parameters by obtaining an image containing an object of known size in meters. In general, this parameter ends up varying little for camera and resolution, thus interfering little with the results. Therefore, if the camera calibration is not performed, a pre-existing calibration file can be used as long as it is from the same camera model and resolution. Calibration file values were sent with the package to be used if necessary.

2. Processing

2.1 Prerequisites:

For the PyDroneCurrents package to work, you need:

- Video:
An example video is available at the link: https://drive.google.com/drive/folders/1I0Oq4Maw_ynuvm1Yo4OrjmdujcLtownm?usp=drive_link

– Python:
The algorithm developed to measure currents is read in a Python environment, which is an open-source language, and can therefore be downloaded for free via the link: https://www.python.org/downloads/.

– JupyterLab (Optional):

Environment for working in Python by creating a notebook that allows easy sharing.

– Python packages: numpy, matplotlib and OpenCV:

Some Python packages are necessary for code processing, two of them are very common packages, numpy and matplotlib, and in addition to these, OpenCV is necessary, which is a library developed in the C++ language that allows image manipulation. One of the options for installing these packages is using 'pip install' (https://pypi.org/project/opencv-python/).

– MediaInfo (Optional):

To obtain video metadata, the MediaInfo program is an option to extract.

2.2 Running PyDroneCurrents in steps

First step:

In a Python/Jupyter-Lab environment, open the reference notebook 'Run_currents_PyDroneCurrents' and first, import the previously installed packages in which you will perform the operations and the PyDroneCurrents package that are already described in the first cell.To use the package, it must be in the same folder as the reference notebook being analyzed. If not, you can place the path where the package is located through the sys library in the option highlighted as a comment in the cell containing the libraries.

In the second cell, for the code to work, it is necessary to edit some variables present, the first variable to enter is the path where the video that will be analyzed in the 'video_fname' variable is located (Line 2), make sure the name is correct as in the model.

The second variable to be edited is where the video metadata is contained. This metadata can be acquired through specific programs that read it, in this case, the MediaInfo program was used (Download link: https://mediaarea.net/en/MediaInfo). Place each variable obtained in its location within CamPos_ST (Line 6).

The variable ‘time_limits’ corresponds to the time of the video in seconds that you want to analyze, with the first number being the starting time and the second the ending time. For current analysis purposes, 10 seconds is enough to obtain the speeds.

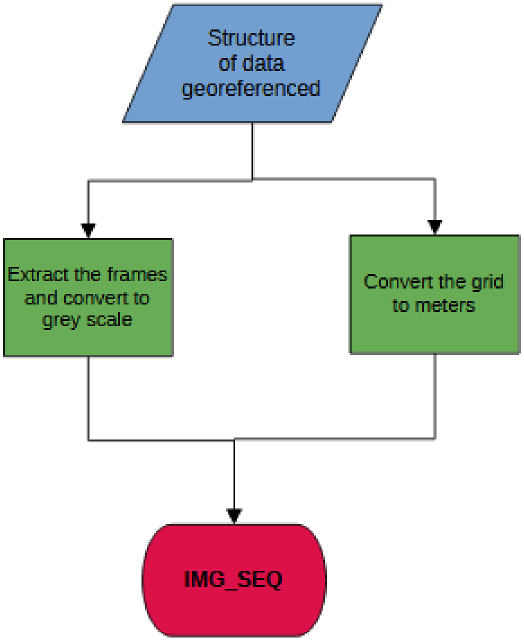
The value of ‘offset_homewater_Z‘ represents the distance from sea level. This parameter is important to be able to obtain the real distance in meters that each pixel in the image is equivalent to. This parameter informs the total altitude that the drone is above sea level.

The file_cal variable contains the camera calibration parameters that are already registered in the code used for the camera of a Phantom 4 Pro v2 drone.

After editing all necessary variables, execute the second cell. This first step will record all the data necessary to create a georeferenced data structure. To create this structure, run the next cell containing the function that does this.

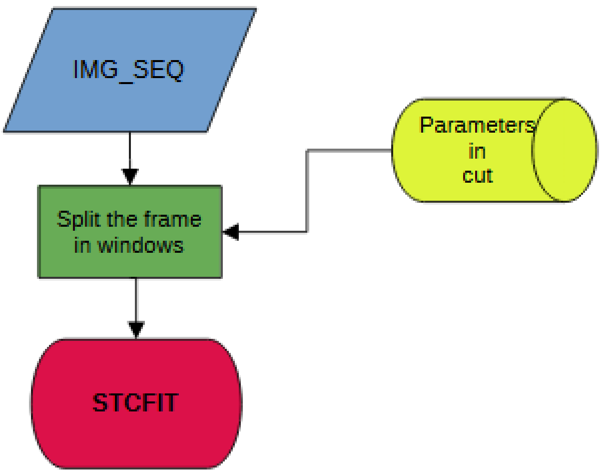
Second step:

In the fourth cell, the 'run_georeference_struct' function is responsible for creating a sequence of images extracted from the video containing all frames. Depending on computational capacity, it is not possible to analyze all frames of the video, so it is important to define at the beginning the total time you want to analyze the video. In addition to extracting the frames, the function also passes the frames to gray scale and extracts pixel size parameters in meters, thus generating an image grid in meters, with the origin being the center of the camera.



Third step:

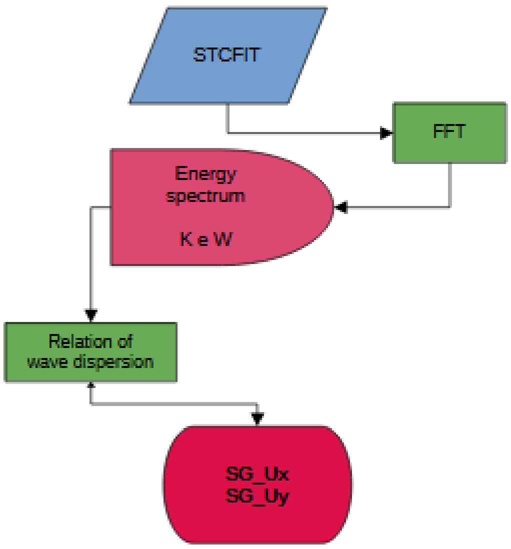
In this step, each frame is divided into windows to run the current analysis. The number, size and overlap of these windows can be edited in the code. At this stage it is also possible to edit some variables in the code according to the environment you want to analyze, such as the limits of wave number, radial frequency and minimum and maximum speeds.



Fourth step

During the fourth stage is where the FFT (Fast Fourier Transform) process takes place. FFT is an algorithm responsible for calculating the fast Fourier transform, which allows the signal to be converted from the spatial domain to the frequency domain. It is through this method that it is possible to convert the pixel intensity values contained in an image into its frequency and convert it into a wave number. Furthermore, this process is carried out in 3D, for this the algorithm used is the 3D FFT, with the radial wave frequency being calculated over the video time.

Once these parameters are obtained, with the wave dispersion relationship it is possible to obtain the surface currents that are acting in the environment at the time of filming. This step is carried out through two attempts, where the second attempt generates vectors with a greater precision. The current values u and v are generated according to the best adjustment based on the limits introduced for the algorithm, with the criterion for the best choice being the calculated signal-to-noise ratio (SNR).



Fifth step:

This last step is where the currents generated in the previous step are plotted. It is possible to view the currents through the camera's reference frame or georeferenced in relation to geographic-north.

The function that plots according to the camera reference makes it possible to visualize the current vectors through the camera grid in the position in which the images were obtained.

To obtain the currents according to their direction in space, it is necessary to georeference the vectors and points contained in the image grid. First, the image grid is transformed, which is in meters, with its origin being the center of the camera into geographic coordinates (latitude and longitude). Afterwards, the vector rotation process is carried out based on the theory of Euler angles.

Remove vectors generated in land areas:

To remove image vectors that are generated in areas that do not have a water surface, a mask is placed where the generated values are located within the mask area will note be valid. For this procedure, points that form a polygon are digitized and for this example the points are stored in a file with ‘csv’ extension, attached to this material.

3. References

Stresser M, Carrasco R, Horstmann J (2017) Video-Based Estimation of Surface Currents Using a Low-Cost Quadcopter. Geosci and Remote Sens Lett 14(11): 2027-2031.