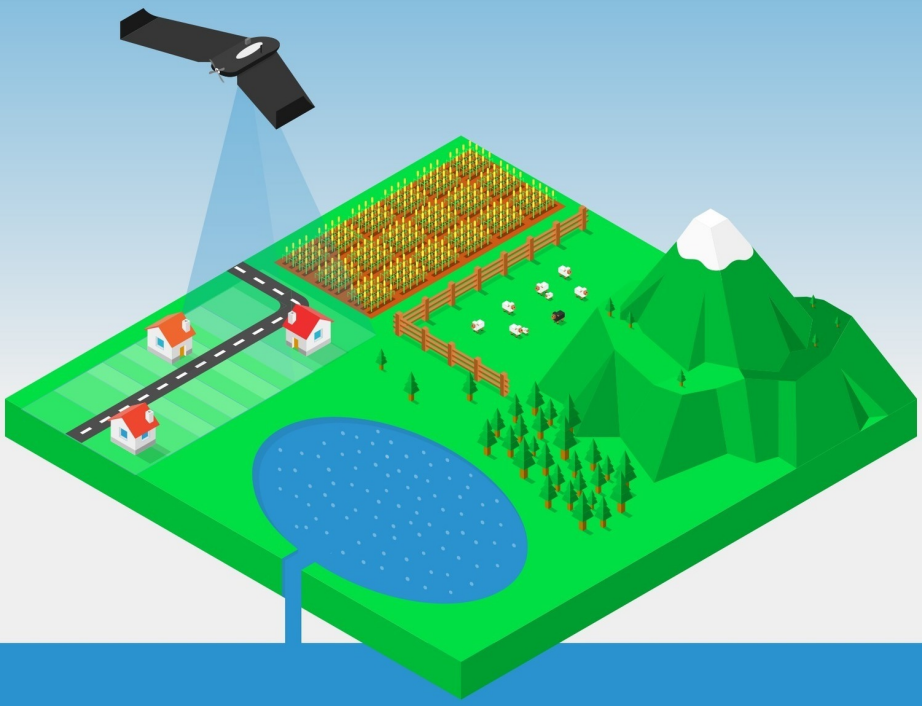
The Ultimate Guide to Drone Mapping
Using Free and Open Source Software



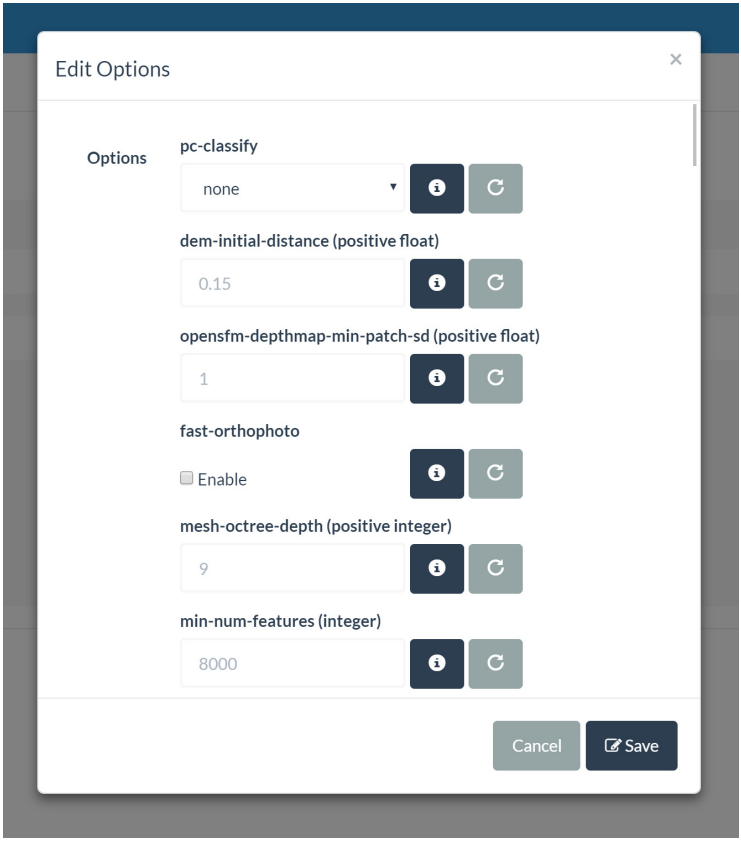
OpenDroneMap
The Missing Guide

Piero Toffanin

10

Task Options in Depth

There are several components involved in the data processing pipeline. Each component has several "knobs" or "adjustable settings" that influence the output. The software exposes a subset of these available knobs through various options. When creating a task, a user can choose to tweak one or more options to change the behavior of the pipeline.



Options as shown on WebODM when creating a task

If the list seems overwhelming, just remember that this is subset of all possible options that could be available from the various components of the data pipeline! This should raise some curiosity. Hidden features and processing capabilities could be hiding in the source code of OpenDroneMap, in the form of an option not yet exposed! The software exposes only those options that seem to have the biggest impact on results, or those

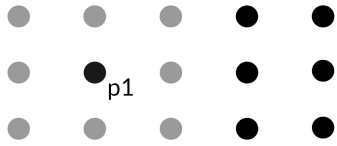
necessary to handle different workflows. But many, many more options, under the hood, remain unexposed in order to keep their number somewhat manageable.

Tuning options is more art than science. There's no clear guidelines on how to tune options to achieve optimal results. That's mostly because the "best" options for a certain dataset do not automatically transfer over to another. Given that there's a big variety of possible scenes, cameras, mission planning strategies, it would be immensely time consuming to write an exhaustive guide. Plus, the tools are evolving quickly, so by the time such guide would be written, it would already be obsolete. But fear not!

This chapter is about understanding in details what each option does. By learning about each individual option, you'll be able to quickly improve your results, explain why certain models turn out the way they do and know what to tweak if the results don't turn out the way you want.

A few of these options might be missing from the graphical interface and might available only from the command line. This is because sometimes the option does not make sense in the context of the interface workflow, or it's simply not supported.

Feel free to jump around and use this chapter as a reference. As the software gets better, some of these options might disappear from future versions and new ones will be introduced. The list below is taken from the software as of March 20th 2019. I will try my best to keep up with updates to the software and plan for an updated edition of the book after the book is published. In alphabetical order:



Dots represent approximate image locations, extracted from EXIF tags. When the matcher-neighbors is set to 8, only the 8 nearest neighbors (highlighted in gray) are considered for matching with image p1.

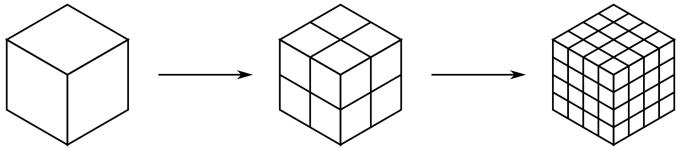
For datasets with lots of overlap, it can be beneficial to increase this value since it's likely that valid matches will not be taken into consideration and decrease the accuracy of the reconstruction. It can be set to zero o disable it. If no location information is embedded in the EXIF tags of the images, this option is disabled. This option works in conjunction with the —matcher-distance option.

mesh-octree-depth

When it comes to generate 3D models, this is probably the most important option. It specifies a key variable for the Screened Poisson Reconstruction[4] algorithm, which is responsible for generating a mesh from the point cloud. The details of the algorithm are fascinating, but probably outside the scope of this chapter. For the curious ones, the best description the author could find is available on Wikipedia under the "Surface Reconstruction" section: https://en.wikipedia.org/wiki/Poisson%27s_equation

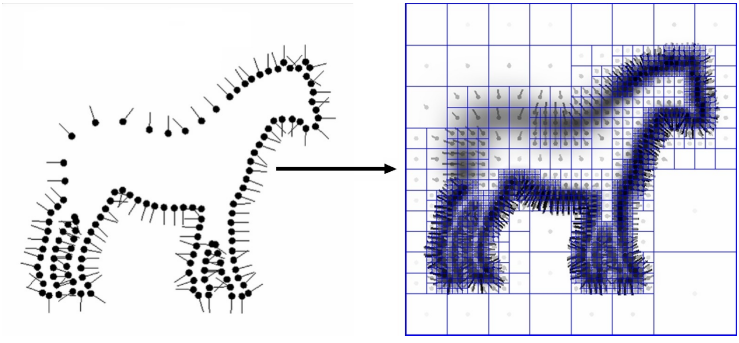
[4] Screened Poisson Reconstruction: watertight surfaces from oriented point sets. http://www.cs.jhu.edu/~misha/MyPapers/ToG13.pdf

To understand how this option affects the output, it helps to visually understand the concept of an octree. First, octree just means "eight-tree" (okta is "eight" in Greek). Why eight? Because at each level (or "depth") of the tree, each box (or "node" or "branch") of the tree is divided in eight parts. At the first level there's only one branch. At the second level there's 8. At the third there's 64 and so forth.



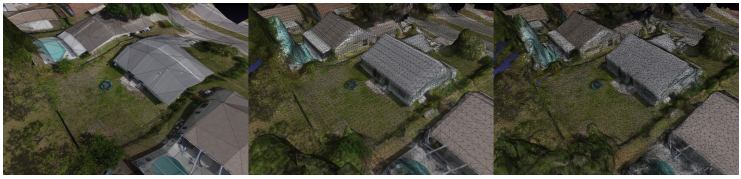
An octree with depth 1, 2 and 3.

Lower depths in an octree allow finer details to be captured.



Points and resulting octree
Image courtesy of M. Kazhdan
http://www.cs.jhu.edu/~misha/Code/

The practical aspect of this option is that the higher the value, the finer the resulting mesh will be, at the trade-off of exponentially longer run-time and memory usage. The default value of 9 works well for a lot of different cases. Flat areas can benefit from lower values (6-8) and urban areas can improve by setting this value higher (10-12). When increasing this option, —mesh-size should also be increased as finer meshes require more triangles.



From left to right: mesh-octree-depth 6, 9 and 11 and mesh-size 10000, 400000 and 1000000
Notice the house structure.

mesh-point-weight

Similarly to —mesh-octree-depth, this option specifies a key variable for the Screened Poisson Reconstruction algorithm, which is responsible for generating a mesh from the point cloud. It affects the mesh by giving more importance to the location of the points. In practical terms, higher values can help create higher fidelity models, but can also lead to the generation of artifacts if excessive noise is present. In general the default value works fairly well.

mesh-point-weight set to 0, 5 and 20. Notice the lack of edges in the left image and the excessive bumps in the right image.

mesh-samples

Similarly to —mesh-octree-depth, this option specifies a key variable for the Screened Poisson Reconstruction algorithm, which is responsible for generating a mesh from the point cloud. It specifies how many points should fall within a node of the octree during its construction. In practical terms, this value should be tweaked between 1 and 20 to improve the smoothness of a model. If there's noise in the point cloud, this value should be increased. If there's little or no noise in the point cloud, this value should be set to 1 (the default).

mesh-samples set to 1, 20 and 100. Ideal values for this option are between 1 and 20.