

# *Focus* (Ver.2017-06) Manual

Allison Hsiang

2016-09-06

## I Introduction

The *focus* module of the *AutoMorph* software package (developed by Pincelli Hull and team [1]) is used to generate extended depth of field (EDF) images from a series of z-stack images via focus stacking, using either Zerene Stacker [2] or FIJI [3]. For the FIJI option, the StackFocuser plugin [4] is used. Zerene Stacker is commercially available, and produces the highest quality EDFs in our experience; however, the non-proprietary FIJI option is provided for users unwilling or unable to obtain a Zerene Stacker license. Zerene Stacker does allow users to download a 30-day free license, which allows users to test out the performance of the software before committing.

## II Technical Specification

*Focus* is designed to work with the output generated by *segment*, the *AutoMorph* module for image segmentation. It takes as input a series of folders, each containing z-stack images of an individual light-colored object on a dark-colored background. As output, *focus* generates three folders: a folder named 'focused' that contains all individual object EDFs, with the metadata labels created by *segment*; a folder named 'focused\_unlabeled' that contains all individual object EDFs, without the metadata labels created by *segment*; and a folder named 'stripped' containing the z-stack images for all identified objects without the metadata labels (used for downstream processing, namely as input for the *run3dmorph* module of *AutoMorph*). In addition, *focus* will compress and archive the original *segment* output z-stack images to conserve space. If the user is using the Zerene Stacker version of *focus*, a LOG file and XML file detailing Zerene Stacker's parameter settings will also be written to file. Figure 1 shows a overview of the *segment* and *focus* pipeline.

*Focus*, like all *AutoMorph* modules, is run using the command line. On Mac OSX, you must install the GNU Coreutils command line tools (more information

[here](#)) in order to use the essential UNIX commands (*e.g.*, 'ls' and cat') and run the *run3dmorph* binary executable. If you are unfamiliar with the command line, we recommend searching for introductory tutorials online and familiarizing yourself before diving in ([here are some suggestions](#)). A good golden rule when dealing with the command line as a beginner is: never input a command if you don't know exactly what it will do!

## III Installation

### III.1 Prerequisites

*Focus* is written in Python and currently is only compatible with Python 2.7. Compatibility with Python 3.x is under development for future release. The complete list of prerequisites is:

- Python v.2.7
- Zerene Stacker (recommended) or FIJI
- ImageMagick

#### III.1.1 FIJI

In order to run *focus* using FIJI, the user must have an installation of FIJI that is accessible from the path via one of the following commands:

- Mac: 'ImageJ-macosx'
- Linux: 'ImageJ-linux32' or 'ImageJ-linux64'
- PC: Not yet supported (under development)

On Mac OSX, this can be achieved by creating a symbolic link to the FIJI executable in the /usr/bin/local folder. Assuming the user has installed FIJI, this can be done by opening a Terminal window and entering:

```
$ ln -s /IPATH/Fiji.app/Contents/MacOS/ImageJ-macosx /usr/local/bin
```

where IPATH is the installation location of FIJI, for example /Applications. You will now be able to use FIJI from anywhere in your system, so *focus* will be able to call FIJI as necessary. Note that you should use absolute paths when creating symbolic links, not relative ones.

The Stack Focuser plugin [4] must also be installed. The class file can be downloaded [here](#). Once downloaded, the plugin can be installed by opening FIJI and selecting the menu option **Plugins -> Install Plugin...** and then selecting the class file.

### III.1.2 ImageMagick

ImageMagick can be downloaded and installed using the installers and directions available [here](#). Alternatively, if you have MacPorts or Homebrew on your system, ImageMagick can be installed using the following commands:

```
$ sudo port install ImageMagick
```

or

```
$ brew install ImageMagick
```

On newer versions of Mac OS X, you may need to return ownership of `/usr/local` back to your user account with the following command:

```
$ sudo chown -R $(whoami) /usr/local
```

## III.2 Setup

Once you have downloaded the *AutoMorph* software package, you will find the *focus* executable in the 'focus' folder. Within this folder you will also find a configuration file (`focus.cfg`) that must be changed to match the user's system and preferences; details on this can be found in section V.

We recommend adding the *focus* executable to your path, so that *focus* can be called from anywhere in your system. To do this on Mac OSX, open the Terminal program (located at `/Applications/Utilities/Terminal`), and type the following command at the prompt:

```
$ ln -s AMPATH/AutoMorph/focus/focus /usr/local/bin
```

where AMPATH is the location of your *AutoMorph* installation. For example, if *AutoMorph* is located in `/Applications`, the full command would be:

```
$ ln -s /Applications/AutoMorph/focus/focus /usr/local/bin
```

You can now use *focus* from any location on your computer.

## IV Quick Run

Once *focus* is installed, it can be run via the command line using the following command (assuming the *focus* executable is in your path):

```
$ focus <path to input directory>
```

Unlike the other *AutoMorph* modules, *focus* does not use a control file; rather, the user supplies the path to the directory that contains the individual object z-stacks (*i.e.*, the path to the 'final' folder generated by *segment*). Note, however,

that the *focus* configuration file must be properly set before running *focus* (see section IV).

*Focus* can take some optional arguments, namely:

- -v, --verbose: turns on verbose mode
- -i, --interactive: runs focusing software in interactive mode (note that this greatly slows down the performance speed of *focus*)
- --reset: reverts the input directory to pre-focused state
- --clean: removes the z.stack directory if a tar.gz archive version of the directory exists

In general, the average user will not need these optional arguments, and they are included here merely for documentation completion's sake.

## V Configuration File

The configuration file tells *focus* what program to use to generate the EDF and sets the necessary parameters based on this choice. The default configuration file is located at AutoMorph/focus/focus.cfg. *Focus* will look for focus.cfg in the directory on which *focus* is being called first; if it cannot find focus.cfg there, it will default to using the focus.cfg file in the installation directory. Thus, if the user needs to use customized settings for a single sample, they can simply make a new focus.cfg file without tweaking the entire installation of *focus*. focus.cfg should be edited using a plain-text editor such as TextWrangler for Mac OSX. Example configuration files can be found in /AutoMorph/focus/example.cfg.

A list of the parameters in the configuration file follows:

### V.1 Focus Parameters

**software:** The software to be used for focus stacking and generating the EDF. The two options are **zerene** for Zerene Stacker and **fiji** for FIJI. The default software is Zerene Stacker.

### V.2 ImageJ Parameters

No additional parameters are required if *focus* is run using *FIJI*.

### V.3 Zerene Stacker Parameters

These parameters only need to be set if the **Software** parameter is set to **zerene**.

**zerene\_dir**: The full path to the location of the user's Zerene Stacker installation. On Mac OSX, the default location is `/Applications/ZereneStacker.app`.

**system\_memory\_MB**: The amount of system memory in MB available to allocate to Zerene Stacker's operation. A minimum of 4400 MB is suggested.

**temp\_dir**: The path to the temp directory for Zerene Stacker; on most systems, this should be `/tmp`.

**headless**: The command to use for running Zerene Stacker in headless mode (*i.e.*, without a graphical user interface (GUI)). On Mac OSX, this should be set to `Xvfb`; on Linux systems, this should be set to:

```
\textbf{xvfb-run --auto-servernum --server-num=1
```

or whatever an appropriate server number is, depending on your setup. To disable headless mode, set this parameter as nothing.

## VI Hands-On Example Run

An example set of z-stack images for testing *focus* can be downloaded on Zenodo [here](#). A mini dataset containing only two z-stacks is also available in AutoMorph/example\_datasets (GitHub). For the following tutorial, we assume the user is using the larger dataset from Zenodo.

The configuration file for this tutorial can be found in AutoMorph/example\_cfg/focus\_4sq.cfg. This tutorial assumes that the user is using FIJI to run *focus*. The user does not need to change the configuration file itself in any way.

An example configuration file for running the example with Zerene Stacker is also included (AutoMorph/example\_cfg/focus\_4sq\_zs.cfg). If using Zerene Stacker, this tutorial assumes that the user is on a Mac OS X system with at least 8 GB of memory, and that Zerene Stacker is installed at `/Applications/ZereneStacker.app/`. If the user meets these requirements, they do not need to change the configuration file `focus.cfg` at all; otherwise, the user must adjust the configuration file as necessary (see section V).

This tutorial also assumes that *AutoMorph* is installed at `/Applications`, that the user has added the `focus` folder to the path, and that the example dataset is located in `/Downloads`.

1. Copy the configuration file (either `focus_4sq.cfg` if using FIJI or `focus_4sq_zs.cfg` if using Zerene Stacker) into `/Downloads/focus_4sq_example_stacks`. Rename it to `focus.cfg`.
2. At the command line prompt, enter:

```
$ focus /Downloads/focus\_4sq\_example\_stacks
```

That's it! While running, *focus* will output messages updating its progress. When complete, the command line prompt will reappear. The output of *focus* will be nested within the input folder (in this case, `/Downloads/focus_4sq_example_stacks`).

## VII Troubleshooting

If the user is using *focus* directory on successful output from *segment*, they are unlikely to run into any problems regarding input files. The most likely source of error while running *focus* is Zerene Stacker and appropriate configuration file settings. Zerene Stacker was not built to run via the command line, and our in-house experience suggests that it is prone to error, especially if multiple instances of it are being run at the same time. If issues arise, we recommend considering the following:

- Check that Zerene Stacker is up to date (the 'update me' dialog box can cause Zerene Stacker to hang even when headless mode is engaged).
- If using the 30-day free trial version of Zerene Stacker, check that this trial has not expired.
- Check that the configuration file is properly set for the focus stacking software being used.
- If using *focus* in Zerene Stacker mode on a multi-user system, make sure that the Zerene Stacker license is accessible to all users who need it

## References

- [1] *AutoMorph* (<https://github.com/HullLab/AutoMorph>)
- [2] *Zerene Stacker*, Zerene Stacker LLC (<http://zerenesystems.com/cms/stacker>)
- [3] Schindelin J, Arganda-Carreras I, Frise E *et al.* (2012) Fiji: an open-source platform for biological-image analysis. *Nature Methods*. **9**(7):676-682.
- [4] Umorin, M. (2002) Stack Focuser (<https://imagej.nih.gov/ij/plugins/stack-focuser.html>)

- [5] Hsiang AY, Nelson K, Elder LE, Sibert EC, Kahanamoku SS, Burke JE, Kelly A, Liu Y, Hull PM. AutoMorph: Accelerating community morphometrics with 2D and 3D image processing and shape extraction. **Methods in Ecology and Evolution**. *In revision*.

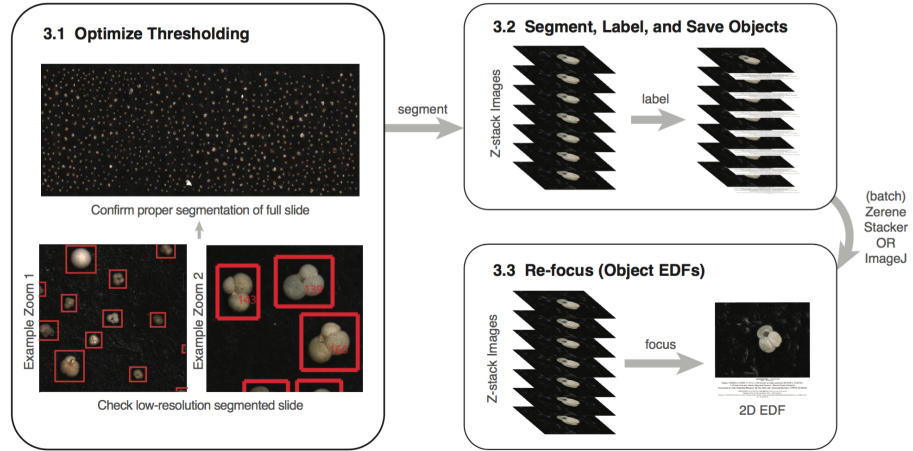


Figure 1: Overview of the image processing pipeline of the *segment* and *focus* modules of *AutoMorph*.