Run3dmorph (Ver.2016.10.25) Manual

Allison Hsiang

2016-11-22

I Introduction

Run3dmorph is the 3D-data extraction module of the *AutoMorph* software package developed by Pincelli Hull and team [1]. *AutoMorph* is available for download on GitHub. The *run3dmorph* module was written by Allison Hsiang and is described briefly in Hsiang *et al.* [2] and in detail in Hsiang *et al.* [3].

Run3dmorph is designed to work directly with the output of *segment* and *focus*, the image segmentation and batch focus stacking modules of *AutoMorph*. The basic pipeline of *run3dmorph* (Fig. 1) is as follows: using z-stacks of individual objects extracted by *segment*, *run3dmorph* writes a series of ImageJ [4] macros that call the Stack Focuser plugin [5] and generates height maps for each object (Fig. 1a). The height maps code relative height using greyscale pixel values ranging from 0 to 255. To remove background noise, the 2D outline of the object is extracted using the EDFs generated by *focus*; the background is then deleted via element-wise multiplication (Fig. 1b). The cleaned-up height map (Fig. 1c) is then converted to a 3D mesh: by correlating the height map greyscale values with the height of each focus plane in the original z-stack using the z-step distance between plane, the real-world height of each pixel (as limited by z-step distance resolution) can be determined (Fig. 1d). Residual noise in the mesh is removed using a custom sliding neighborhood filter with a user-defined $n \times n$ kernel (where n is a positive odd integer). Finally, the extracted mesh is outputted as OBJ and OFF files, and the surface area and volume are calculated. The surface area and volume of the hidden backs of the objects are also calculated using a series of three idealized shapes (Fig. 2). For more details on these steps and algorithms, see Hsiang *et al.* [3].

Run3dmorph, 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!

Run3dmorph is run using a plain-text input file, referred to here as a 'control file'. Explanations for the user-controlled variables in the control file, and how to set them, are presented in section IV. By default, *run3dmorph* will output the x,y,z-coordinates of extracted meshes as CSV, OBJ, and OFF files. It will also output the calculated volume and surface area of each extracted mesh, along with total volume and surface area estimates using each of the three idealized back shapes (Fig. 2).

Optionally, the user can have *run3dmorph* output 3D PDFs, which contain: the metadata label for each object; the extracted length and height of each object; the operative grid size and base unit used for mesh extraction; the 3D mesh embedded as a rotatable, zoomable object (viewable using Adobe Acrobat/Reader); and the original 2D EDF image of each object. The 3D PDF allows for easy checking of the fidelity of the 3D mesh extraction, but requires the non-trivial installation of many dependencies (see section II). If the user has difficulty getting the 3D PDF to build, we recommend checking the extracted 3D meshes in a mesh-viewer program such as MeshLab.

II Installation

II.1 Prerequisites

Run3dmorph currently requires MATLAB (version R2014b or higher) and Python (version 2.7); it only runs on Unix-like systems, including Mac OSX. Removal of dependence on proprietary software, as well as PC compatibility, is currently in development. MATLAB must be available in your system's path and callable via the command 'matlab'. On Mac OSX, this can be achieved by creating a symbolic link to the MATLAB binary executable:

```
ln -s /IPATH/MATLAB_VERSION.app/bin/matlab /usr/local/bin/matlab
```

where IPATH is the installation location of MATLAB, for example /Applications, and VERSION is your MATLAB installation version number, for example R2016b. After entering your system password when prompted, you will now be able to use MATLAB from anywhere in your system, so *run3dmorph* will be able to call MATLAB as necessary.

Python must be in your system's path and callable via the command 'python'. This should be the default behavior with a standard installation of Python on a Unix-like system. Python comes preinstalled on Mac OSX; this can be verified (and the version number checked) by typing 'python' into the command line in Terminal.

Run3dmorph also requires an installation of ImageJ or FIJI [4] (recommended) that is callable from the command line via the one of the following commands:

- Mac: 'ImageJ-macosx'
- Linux: 'ImageJ-linux32' OR 'ImageJ-linux64'

On Mac OSX, this can be achieved by creating a symbolic link to the ImageJ/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/ImageJ-macosx
```

where IPATH is the installation location of FIJI, for example /Applications. After entering your system password when prompted, you will now be able to use FIJI from anywhere in your system, so *run3dmorph* will be able to call FIJI as necessary. If you elect to install ImageJ instead of FIJI, note that the ImageJ executable is found at ImageJ.app/Contents/MacOS/JavaApplicationStub. This should still be linked symbolically to /usr/local/bin/ImageJ-macosx as above; *run3dmorph* does not recognize the JavaApplicationStub command when running ImageJ. For this reason we recommend the user install FIJI instead of ImageJ to use with *run3dmorph*.

The Stack Focuser plugin [5] must also be installed. The class file can be downloaded here; this file should be placed in the plugins folder in the user's ImageJ or FIJI installation (for FIJI on Mac OSX, this folder can be accessed by right-clicking the FIJI application icon and clicking 'Show Package Contents').

Finally, if the user is intending to use the 3D PDF functionality of *run3dmorph*, an installation of LaTeX is required. LaTeX must be callable from the command line via the command 'pdflatex'. This should be the default behavior when installing LaTeX with the default settings. The required STY file for the required LaTeX package, media9, is included in /AutoMorph/run3dmorph.

II.2 Setup

Once you have downloaded the *AutoMorph* software package, you will find *run3dmorph* in the 'run3dmorph' folder. The 'bin' folder contains the *run3dmorph* executable, while the 'src' folder contains the MATLAB and Python source code and a default control file (run3dmorph_control_file_v.2016.09.21.txt), in which the user sets *run3dmorph*'s various parameters.

You must add the *run3dmorph* 'bin' and 'src' folders to your path, so that *run3dmorph* 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:

```
nano ~/.bash_profile
```

The nano text editor will open the .bash_profile, and you will likely see paths set by other programs (*e.g.*, Python). Move to the bottom of the of your .bash_profile file (do not change anything that's already there unless you know what you're doing!) and then type:

```
#Setting PATH for run3dmorph
PATH="R3DBINPATH:${PATH}"
export PATH
PATH="R3DSRCPATH:${PATH}"
export PATH
```

where R3DBINPATH and R3DSRCPATH are the full paths to the *run3dmorph* 'bin' and 'src' folders, respectively. For instance, if *run3dmorph* is located at /Applications/AutoMorph/run3dmorph, the full entry would be:

```
#Setting PATH for run3dmorph
PATH="/Applications/AutoMorph/run3dmorph/bin:${PATH}"
export PATH
PATH="/Applications/AutoMorph/run3dmorph/src:${PATH}"
export PATH
```

Once this is done, exit nano (control + X), hit 'Y' when prompted to save, and then hit 'enter' to accept the original file name to write. You can now run *run3dmorph* from anywhere in your system (note that you may need to restart your shell for this to take effect).

III Quick Run

Once *run3dmorph* is installed, it can be run via the command line using the following command (assuming the *run3dmorph* 'bin' and 'src' folders are in your path):

```
run3dmorph <path to control file>
```

Run3dmorph can also take an optional argument, 'reset', which will delete the output 'morph3d' folder if it exists in the currently specified output directory and rerun the entire pipeline. This is activated via the following command:

```
run3dmorph <path to control file> reset
```

While running, *run3dmorph* will output status messages on its progress. It will also indicate when a mesh cannot be extracted for a particular object (this is usually due to inappropriate image processing parameters in the *run2dmorph* outline extraction/background deletion step, namely **Intensity Range In**, **Intensity Range Out**, **Gamma**, and **Threshold Adjustment**).

Depending on the number of objects being processed, *run3dmorph* can take a long time to run (for reference, our in-house microfossil images take 3-5 minutes to process per object). If *run3dmorph* is stopped before it is compete, it will restart from where it left off as long as the morph3d output folder has been untouched. To restart *run3dmorph*, simply run *run3dmorph* with the original control file again.

IV Control File and Parameters

The control file serves as the means by which the user supplies the necessary parameters for *run3dmorph* to operate. A default version of the control file can be found in AutoMorph/run3dmorph/src. Within the control file, each parameter is briefly described. If the parameter requires user input, it is marked as such; otherwise, the default value of the parameter is listed (although the user can, of course, change these as they see fit). It is recommended that the user read through the entire control file before using *run3dmorph*. The control file should be edited using a plain text editor (*e.g.*, TextWrangler for Mac OSX) to avoid interpretation issues and must use Unix (LF) encoding.

For parameters that are not required, the default value can be used by setting the parameter equal to '[]', as such:

```
output_dir = []
```

A listing of the parameters for *run3dmorph* follows, with required parameters marked with a *:

IV.1 Global Parameters

> ***Path to Run3dmorph Software**: the full path (*not* relative) to the 'src' folder of the user's *run3dmorph* installation. For example, if *AutoMorph* is installed in /Applications, this parameter would be set as **/Applications/AutoMorph/run3dmorph/src**.

***Path to Run2dmorph Software**: the full path (*not* relative) to the 'src' folder of the user's *run2dmorph* installation. For example, if *AutoMorph* is installed in /Applications, this parameter would be set as **/Applications/AutoMorph/run2dmorph/src**.

***Focused Directory**: the full path (*not* relative) to the folder containing the 'final' directory created by *focus* (*not* the path to the 'final' directory itself; *e.g.*, **/Applications/AutoMorph/focus/examples/4sq** is correct, but **/Applications/AutoMorph/focus/examples/4sq/final** is not).

***Output Directory**: the full path (*not* relative) to the location where *run3dmorph*'s output should be saved. *Run3dmorph* will create a folder named 'morph3d' in this location, within which all output will be saved.

***Sample ID**: a name designating the identity of the current run.

***Macro Mode**: a boolean specifying whether macro-object mode should be activated. This should be set to True when the input images were captured using a camera (*e.g.*, a DSLR camera) and False when the input images were captured using a microscope. When set to True, ImageJ/FIJI will run the StackReg plugin [6] using the 'scaled rotation' method to align z-slices before generating the height map, and then invert the greyscale values in the height map (a necessary step based on in-house experience dealing with camera-generated images).

IV.2 ImageJ/FIJI Parameters

Kernel Size (StackFocuser): an odd integer n that sets a $n \times n$ kernel that is used by Stack Focuser to generate a heightmap using a sliding neighborhood filter. Larger kernel sizes correspond to larger patches over which pixel values are averaged, *i.e.*, lower detail. Images with large amounts of noise should use larger kernel values. The default value is 11.

FIJI Architecture: can be set to either 32 or 64, corresponding to the 32-/64-bit versions of ImageJ/FIJI. This is only relevant for Linux systems and should be set as 'None' on Mac OSX systems, like so:

```
fiji_architecture = None
```

IV.3 Mesh Extraction Variables

***Unit**: the base unit used for pixel conversion, *e.g.*, microns.

***Pixel Calibration Factor**: a number designating the conversion factor for the pixel size in the input images. The height and width of the input images will be scaled using this conversion factor such that a 1×1 pixel = 1×1 base unit (as specified under **unit**).

***Number of Slices**: the number of slices in each individual image z-stack.

***Z-Step Size**: the distance between each z-slice in the input z-stacks, in base units (as specified under **unit**).

Kernel Size (Outlier Filter): a positive odd integer designating the neighborhood size (in pixels) for the outlier filter that cleans noise the 3D mesh (see Fig. 1d). Use a larger kernel size for larger patches of noise (though note that processing time increases significantly as kernel size increases). Default value is 19; for our in-house foraminifera images, we find a kernel size of around 45 works well.

Downsampling Grid Size: a positive integer designating the x,y-grid size (in pixels) to be used if the user wishes to downsample the 3D mesh coordinates. The default value is 1, which corresponds to a coordinate being written for every pixel in the object.

IV.4 Run2dmorph Parameters

These parameters are used when using *run2dmorph* to extract 2D outlines to delete background noise (see Fig. 1b). Users are advised to determine the optimal values for these parameters using *run2dmorph* before running *run3dmorph*.

Intensity Range In: a range specified in the format [$\boldsymbol{low_{in}}$ $\boldsymbol{high_{in}}$] that gives the input intensity range for the gamma filtering step. This is generally used to increase the contrast between the background and the object by lightening darker values in the input image by mapping the $\boldsymbol{low_{in}}$ value to the $\boldsymbol{low_{out}}$ value and the $\boldsymbol{high_{in}}$ value to the $\boldsymbol{high_{out}}$ value. More information can be found in the documentation for the *imadjust* function of the MATLAB image processing toolbox. The default range is [0 0.2].

Intensity Range Out: a range specified in the format [$\boldsymbol{low_{out}}$ $\boldsymbol{high_{out}}$] that gives the output intensity range for the gamma filtering step. The default range is [0 1].

Gamma: a number that specifies the gamma value for the gamma filtering step. Gamma values <1 result in a concave-down gamma correction curve (*i.e.*, the intensity range mapping is weighted toward brighter output values); gamma values >1 result in a concave-up gamma correction curve (*i.e.*, the intensity range mapping is weighted toward darker output values); and gamma = 1 results in a linear mapping (Fig. 2). The default value is 2.

Threshold Adjustment: a number that specifies an adjustment value that will be added to the threshold value that is automatically determined by the MATLAB function *im2bw* during conversion of the image from greyscale to black and white. A higher threshold adjustment value corresponds to higher tolerance during the conversion (*i.e.*, a broader, lighter range of grey will be considered white in the final black and white image). Note that while this adjustment value can be negative, the *total* threshold adjustment value cannot be negative. The default value is 0.

Noise Removal Limit: a number that specifies the size (in percentage of total image size) below which isolated objects will be removed from the image during the noise removal step. For instance, a noise removal limit of 0.05 will result in all objects smaller than 5% of the total image size being removed. The default value is 0.05.

IV.5 3D PDF Parameters

***Build PDF**: a boolean specifying whether *run3dmorph* should generate files for 3D PDF generation (*i.e.*, IDTF, U3D, and LaTeX files).

***Run LaTeX**: a boolean specifying whether *run3dmorph* should run LaTeX to generate 3D PDFs (should be set to False if **Build PDF** is set to False).

V Hands-On Example Run

An example 'final' output directory created by *focus* that is ready to be processed using *run3dmorph* can be downloaded on Zenodo here. If the user completed the hands-on tutorial from the *focus* manual, they may also choose to use the files outputted then for this tutorial. The example control file is located at /AutoMorph/run3dmorph/examples/run3dmorph_control_4sq_example.txt. If the user's installation of *AutoMorph* is in /Applications, then this example control file can be used directly without any changes (unless the user wishes to turn off 3D PDF creation). Otherwise, the user must change the **Path to Run3dmorph Software**, **Path to Run2dmorph Software**, **Focused Directory**, and **Output Directory** parameters using a plain-text editor.

Step-by-step instructions – which assume that the user has added *run3dmorph* to their system's path as specified in section II.2 – follow:

1. At the command line prompt, navigate to the folder containing the control file:

```
cd /Applications/AutoMorph/run3dmorph/examples
```

2. Start *run3dmorph* by entering the following command:

```
run3dmorph run3dmorph_control_4sq_example.txt
```

Run3dmorph will proceed, beginning with height map generation. Status update messages will be printed to the screen as *run3dmorph* runs. When extraction is complete, the command line prompt will reappear. The output files will be located in: /Applications/AutoMorph/run3dmorph/examples/4sq/morph3d.

VI Troubleshooting

After adding the 'bin' and 'src' folders of *run3dmorph* to your path, you may receive 'permission denied' error messages when trying to use *run3dmorph*. If this is the case, run the following commands in the command line to make the listed files executable (assuming *AutoMorph* is installed in /Applications):

```
chmod +x /Applications/AutoMorph/run3dmorph/bin/run3dmorph
chmod +x /Applications/AutoMorph/run3dmorph/src/batch3dmorph.py
chmod +x /Application/AutoMorph/run3dmorph/src/batchIJSF.py
```

This may also happen with the IDTFConverter executables, which are used during the 3D PDF creation stage. If this occurs, make the IDTFConverter executables executable using the chmod +x command as above. For example, on Mac OSX, assuming *AutoMorph* is installed in /Applications, you would run:

```
chmod +x /Applications/AutoMorph/run3dmorph/src/mesh2pdf/bin/maci/IDTFConverter
```

Additional issues may arise due to invalid control file specifications and/or inappropriate image processing parameter values (as in *run2dmorph*). In general, if *run3dmorph* encounters an error before outline extraction begins, something is likely wrong with the control file formatting or encoding. In this case, the user is advised to:

- Check that the encoding of the control file is set to Unix (LF) and that the control file is in plain text format
- Check that the paths specified under **Global Parameters** are correct
- Check that the parameters are formatted correctly; in particular, a common mistake is specifying parameters inside brackets (*e.g.*, specifying **Kernel Size (Outlier Filter)** as **[45]** instead of **45**). The only exceptions to this are the **Intensity Range In** and **Intensity Range Out** parameters, which *should* be specified within brackets as a range (*e.g.*, **[0 0.2]** instead of **0 0.2**).

If *run3dmorph* runs but declares it cannot extract meshes for an excessive number of objects, the problem likely lies in the *run2dmorph* image processing parameters; in this case, the user is recommended to first optimize these parameters by running *run2dmorph* separately on 5-10 test images before returning to run *run3dmorph* on the full sample.

References

[1] *AutoMorph* (https://github.com/HullLab/AutoMorph)

[2] Hsiang AY, Nelson K, Dobbins B, Elder LE, Liu Y, Hull PM. AutoMorph: Accelerating community morphometrics with 2D and 3D image processing and shape extraction. *Methods in Ecology and Evolution. In prep.*

[3] Hsiang AY, Elder LE, Hull PM. (2016) Toward a morphological metric of assemblage dynamics in the fossil record: a test case using planktonic foraminifera. *Philosophical Transactions of the Royal Society B.* **371**:20150227 (doi:10.1098/rstb.2015.0227)

[4] 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.

[5] Umorin, M. (2002) Stack Focuser (https://imagej.nih.gov/ij/plugins/stack-focuser.html)

[6] Thvenaz, P. (2011) StackReg (http://bigwww.epfl.ch/thevenaz/stackreg/)

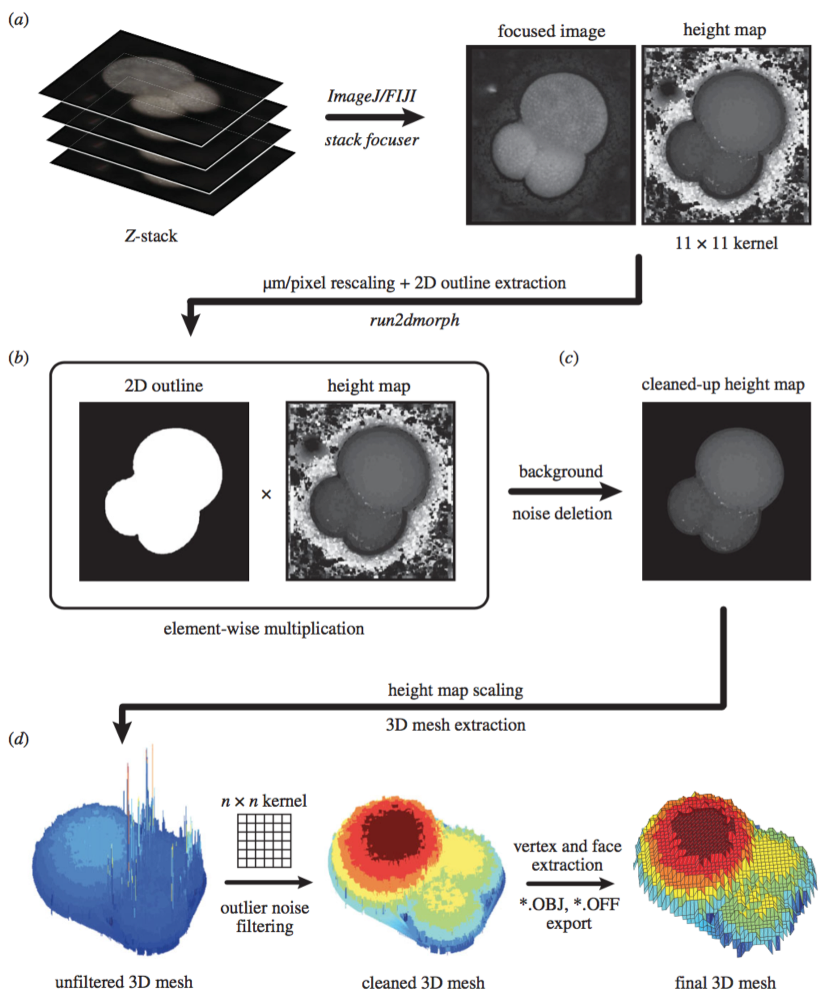

Figure 1: Visual pipeline of 3D mesh extraction using *run3dmorph*. From Hsiang *et al.* [3].

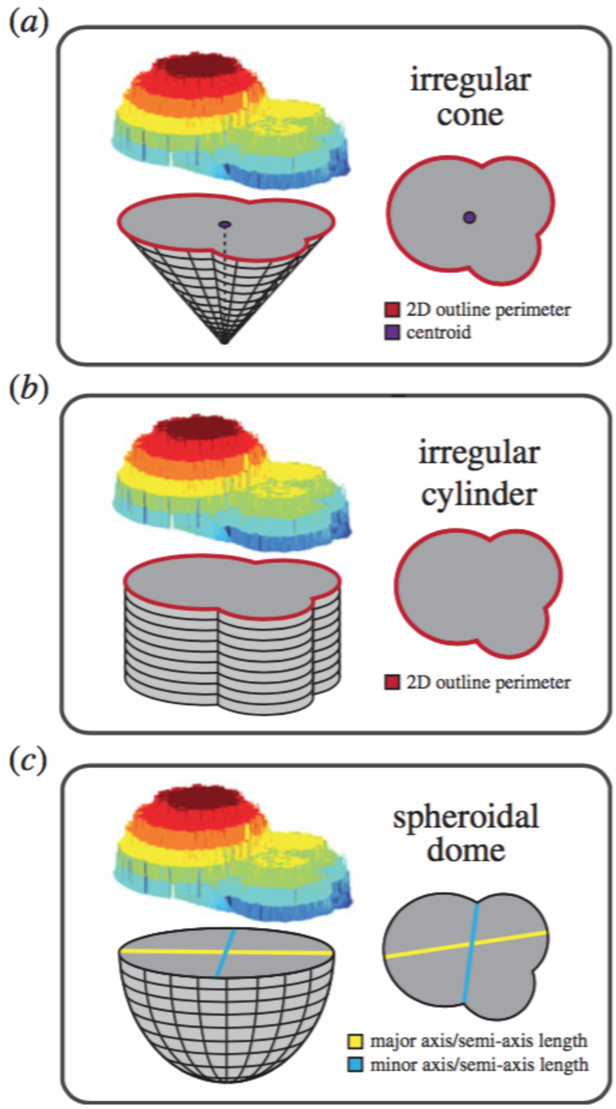

Figure 2: Idealized back shapes, used by *run3dmorph* to estimate surface area and volume of the entire object. From Hsiang *et al.* [3].