Focus Stacking OME-TIFF Microscopy Images

Alec Soronow
Aahwan Koirala
BME 160

January 28, 2022

Abstract

Microscopy is a necessity in biology and its wide range of applications have diversified the format of microscopy data. These formats ensure microscopes to give their end user useful data. A common technique is split and image into focal planes as a z-stack. Having separated focus points is useful for allowing the inspection of every in focus object for a given specimen. However, for humans processing this data it is often desirable to have the entire image in focus as one plane, a method known as focus stacking. A range of proprietary microscope softwares provides solutions for this but an open-source general algorithm is hard to find. We propose a python program run on the Anaconda distribution that uses machine learning feature detection, combined with traditional histogram analysis to quickly process microscopy images. The open-source solution is intended to be widely compatible across environments and support the open microscopy format TIFFs that most microscopes can export.

1 Introduction

Analysis of microscopic images of given specimens is an integral part of biology. Modern microscopes can take images at varying focal depths to ensure they capture the specimen in focus. These planes (z-stacks) need to be scanned through by a human to find what looks best in focus. Some microscopes have an auto-focusing routine, but many researchers prefer multi-depth approaches to prevent data loss due to equipment error. The Euiseok Kim lab at the University of California, Santa Cruz uses multi-depth imaging routinely in its research. Scans can contain up to 5 different focal depths, and these need to be analyzed to count all the cells in an image. Compounding the issue is that there are typically multiple channels for a given scan, each with its z-stacks. That process adds considerable time to quantify the results of experiments. Focus stacking can take all those focal planes and select the in-focus elements, condensing them to a single image for ease of use.

2 Design and implementation

The algorithm was inspired by a stack overflow post [2], and a more straightforward implementation by Charles McGuinness [1]. We used the OpenCV library for its breadth of tools. The exported format of the images is a multi-channel OME-TIFF file. We take that file and read its raw byte data as grayscale values for each z-stack in each channel. A feature detector ORB finds the features in every z-stack of each channel. Within each channel, the features align all the images to the topmost z-stack. Next, one of the following edge detection algorithms, Laplacian of Gaussian, Sobel Gradient, or Canny, calculates each image's in-focus regions. Each algorithm has varying

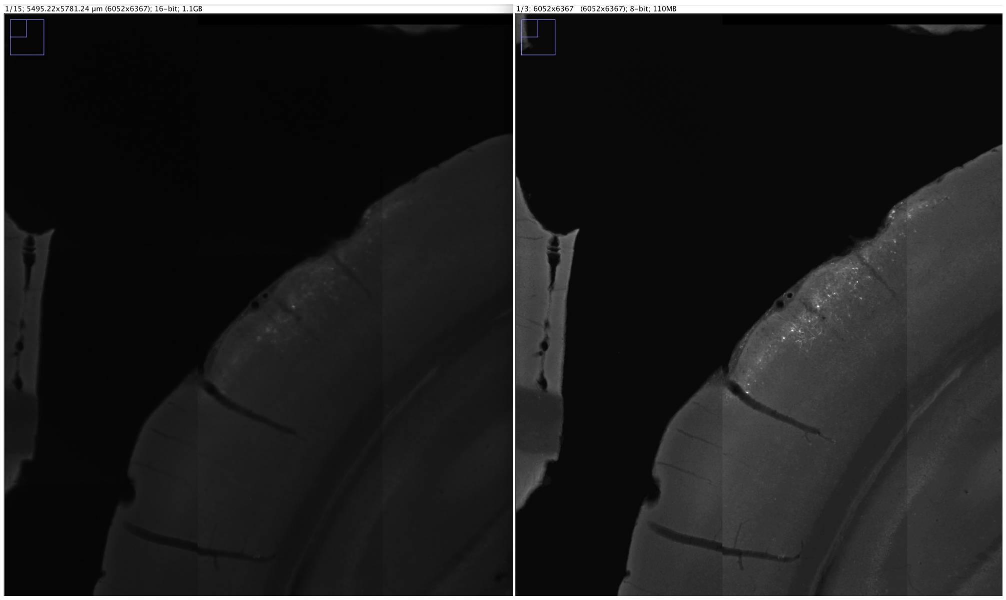

Figure 1: Left: Unfocused top most z-stack of an OME-TIFF. Right: Focus stacked image with increased brightness. Note the original had 15 z-stacks, 5 per channel, and the output has one image per channel. Conversion to 8-bit grayscale data is necessary input images into ORB.

effectiveness depending on the image, so we let the user decide which to use. Finally, the edges construct a bit-wise mask that can construct a final image from all the in focus elements of each z-stack. The result is an image that contains all the in-focus elements of the stack of that channel. We provide sample output in Figure 1. This method loads every image in memory, so it is not ideal for huge files. Down sampled images should provide plenty of data for cell counting and quantification and are ideal for this program.

3 Conclusion and future work

The program performs well for files less than 4GB, and it quite speedy. Currently, it only supports OME-TIFF in the ZCXY or CZXY configuration. In the future, adding support for more OME-TIFF formats and preserving metadata would be significant improvements. ORB is also not the best feature detector out there. A better feature detector or even a custom trainable one tailored to the data set could enhance results.

References

[1] Charles McGuinness. *focusstack*. May 3, 2015. URL: `https://github.com/cmcguinness/focusstack` (visited on 02/21/2021).

[2] *What are some common focus stacking algorithms?* May 16, 2014. URL: `https://stackoverflow.com/questions/15911783/what-are-some-common-focus-stacking-algorithms` (visited on 02/21/2021).