Tracking Pipeline On Nikon Ti2 Confocal Microscope

Danish Islam
Zhen Lab

Last Updated: June 2024

0. Introduction

Several students at the Zhen Lab worked on creating a completely custom calcium tracking pipeline starting from recording the data on the confocal microscope, to processing the data with CaTracker and then performing artifact correction. This guide focuses on the steps of the pipeline I developed that come before CaTracker, which is essentially generating the data!

There are 5 steps: setting up your experiment with Micro-Manager, using the Confocal Automated Tracking System to track your sample, using the Recording program to save the data to disk, the Compression program to create HDF5 files and the HDF5 Video Splitter program to crop the videos as you please to give as input to CaTracker.

I hope to make this guide as easy-to-read and concise as possible, so if you have any questions feel free to ask Danish or Hongruo on Slack or email. If you have technical issues like reports of a bug, you can contact Danish.

1. Micro-Manager

Micro-Manager is a software that was developed by researchers at University of California, Berkely. You must start this program and start the live feed through this because our tracking program uses its API Pycro-Manager to retrieve the image data.

First, make sure you started the Nikon Ti2 Eclipse microscope. Then when you start Micro-Manager, the window will appear as it does in Figure 1 when you start it up. There will be preset configurations for experiment conditions you can use on the right but if you want to create your own, you can ask Hongruo how to. Now that this is set up, simply press **Live** to start your live video feed.

Wesley and Hongruo will be the best guides on how to set up the dual splitter light conditions and the rest of the experiment here. Make sure your sample is centered in the video feed.

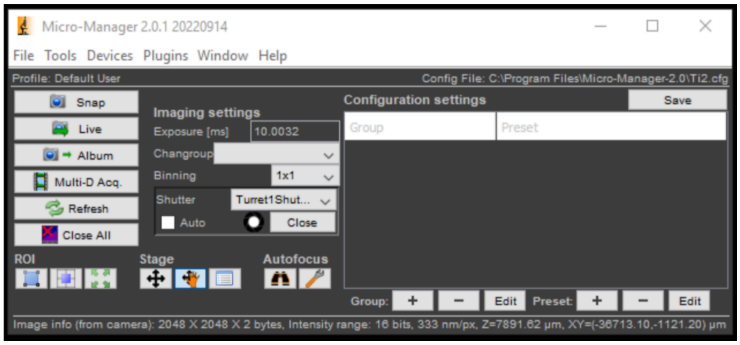

Figure 1. The starting menu of Micro-Manager.

2. Confocal Automated Tracking System

With the Micro-Manager stream running, you can now start tracking your sample. Open up the following directory in PyCharm to run the program: **C:\Users\Nikon\PycharmProjects \Ti2-Tracking-Application.** If the Micro-Manager live stream is on, then it will open a window that looks like Figure 2. Sometimes Micro-Manager has an error in initially connecting, so don't worry if it takes two or three attempts at starting the program.

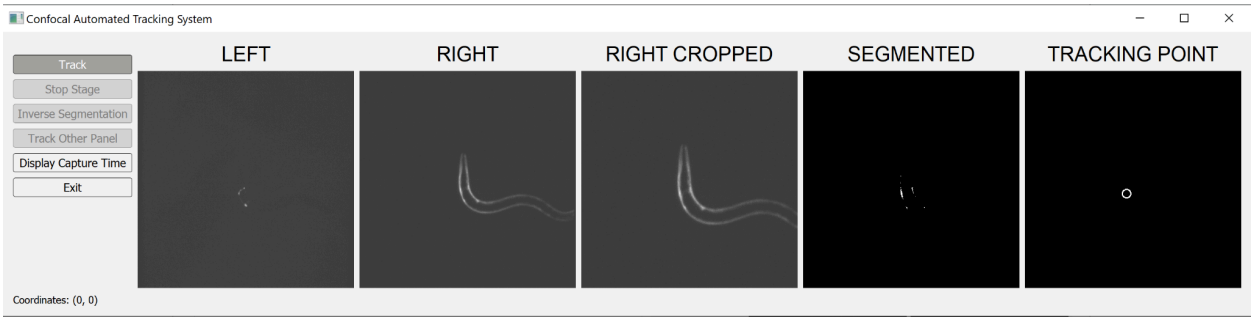

Figure 2. A screenshot of the Confocal Automated Tracking System GUI.

There are five panels showing, in order, the video feed of the left channel, the video feed of the right channel, the zoomed-in feed of the channel you choose to track, the segmentation on the zoomed-in feed and a feed of the current tracking point.

At the bottom it will show you the XY stage coordinates and on the left are your buttons. If you are satisfied with the segmentation, you can press the **Track** button and it will automatically follow your sample with velocity-control using a proportional controller. Please make sure you don't move the stage through the joystick here or else it will result in losing control of the stage. If you do lose control of the stage, stop **Track** and press **Stop Stage**. If it still does not work you have to restart the microscope.

There are other options such as **Inverse Segmentation** for flipping the binary thresholding, **Track Other Panel** to switch what side you want to track and **Display Capture Time** for information on the frame rate in the terminal.

3. Recording Program

If tracking is working fine, you can now begin recording. We created a separate application for recording to optimize tracking performance while also allowing for recording without the tracking program. This also depends on the Micro-Manager live stream. Click on the **Recording Application** on the desktop to open up the window that looks like Figure 3.

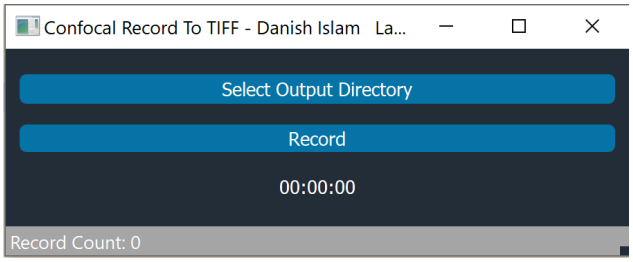

Figure 3. An illustration of the Recording Program.

The use of this program is simple. Press **Select Output Directory** to choose where you want your files to get saved. Then press **Record** and it will save each frame as a TIFF image in a folder within your output directory. It will also output a text file called **recording_log_<date>_<timeofday>.txt**. This shows the stage coordinates and time of photo for each frame.

4. Compression Program

Now you should be finished with the live recording steps, you can do the rest of these steps “offline” without the microscope. Now that you have the folder of TIFF images, you want to assemble them into a compressed video as an HDF5 file. Since these video recordings are very high definition, they can take up alot of space which is why we compress the data to conserve as much space as possible on our hard drives.

Please keep in mind, there is a trade-off between the level of compression and the time it takes for processing. Higher compression levels will take longer to process, but this is preferred in the lab, so it's best to run these compressions in the background! To get started click on the **Compression App** on the desktop to open up a window that looks like Figure 4.

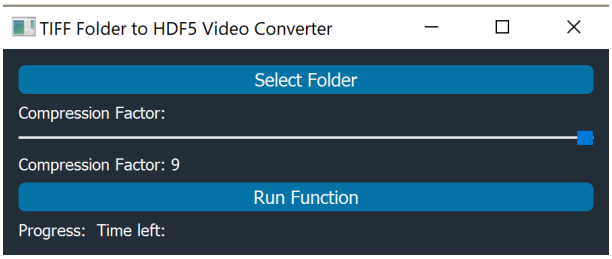

Figure 4. The GUI of the Compression App.

First, you can select the folder with **Select Folder** where you have your raw data recorded from Section 4. Then, use the bar to choose your compression factor with 1 being the lowest and 9 being the highest (and preferred by the lab). Once you press **Run Function** it will begin to compile the HDF5 file in a folder called **output** in the parent directory of your input folder. In the terminal it will show you the compression time for each frame and you can come back to see if it's done according to your best estimate.

5. HDF5 Video Splitter

Now that you have your HDF5 video created, you may find you want to split it up so that you can feed it into CaTracker in bite-sized chunks. You can use the video splitter command line program by simply typing in hdf5_splitter into your terminal. The program will run as shown below in Figure 5.

```
====================================
Zhen Lab HDF5 Video Splitter 🔬🥽
A command line program by Danish Islam
Detected you are on a Windows environment
====================================

1. Please input a valid path to the HDF5 video you want to divide up.
Enter here: D:\Zhen Lab\CaTracker Paper Data\video25012024-828_1.h5
Confirmed to be a valid HDF5 file existing on your computer

2. Loading in your video, please wait...
Loaded in the video. It has 1363 frames and takes 2.662 GB w/o compression

3. Please enter a list of tuples representing the coverage of each sub video e.g. (1,5) (6,10)
Enter here: (0,250)

4. Generating the sub-videos in the same directory as the input path
Created a sub-video from frame 0 to 250! Took 111.36 seconds.
```

Figure 5. Demonstration of the Video Splitter program.

It should work for Windows, MacOS and Linux environments. First, it will ask you to enter a valid path to your HDF5 file. Once it is confirmed it is a valid file, it will take some time to load in your video and it reports how many frames the video has. In the third step, you can split the video by creating subsets based on the start and end frames. If you are working with the highest compressed HDF5 videos it will take a bit of time because it has to make sure the output is compressed as well.

6. Acknowledgments

Special thanks to PhD candidate Hongruo Zhang for helping me set up this pipeline since May 2023. Also, a huge thanks to Dr. Mei Zhen for her extensive support in creating this pipeline. I also had a great time participating in the bigger calcium tracking pipeline project with Dr. Wesley Hung, Huayin Luo, Mohammad Haddadnia and Shuyu Van Kerjwick. Last but not least, a big thanks to the entire Zhen Lab team for creating a warm and supportive work environment.