



PyZebrascope: An Open-Source Platform for Brain-Wide Neural Activity Imaging in Zebrafish

Rani Barbara^{1†}, Madhu Nagathihalli Kantharaju^{2,3†}, Ravid Haruvi¹, Kyle Harrington^{2,4*} and Takashi Kawashima^{1*}

¹Department of Brain Sciences, Weizmann Institute of Science, Rehovot, Israel, ²Max-Delbrück-Centrum for Molecular Medicine in the Helmholtz Association, Berlin, Germany, ³Humboldt University of Berlin, Berlin, Germany, ⁴Oak Ridge National Laboratory, Oak Ridge, TN, United States

OPEN ACCESS

Edited by:

Lilach Avitan,
Hebrew University of Jerusalem, Israel

Reviewed by:

Marco Lorenzo Dal Maschio,
University of Padua, Italy
Limor Freifeld,
Technion Israel Institute of
Technology, Israel

*Correspondence:

Kyle Harrington
kyle@kyleharrington.com
Takashi Kawashima
takashi.kawashima@weizmann.ac.il

[†]These authors have contributed
equally to this work

Specialty section:

This article was submitted to
Molecular and Cellular Pathology,
a section of the journal
Frontiers in Cell and Developmental
Biology

Received: 13 February 2022

Accepted: 18 April 2022

Published: 19 May 2022

Citation:

Barbara R, Nagathihalli Kantharaju M,
Haruvi R, Harrington K and
Kawashima T (2022) PyZebrascope:
An Open-Source Platform for Brain-
Wide Neural Activity Imaging
in Zebrafish.
Front. Cell Dev. Biol. 10:875044.
doi: 10.3389/fcell.2022.875044

Understanding how neurons interact across the brain to control animal behaviors is one of the central goals in neuroscience. Recent developments in fluorescent microscopy and genetically-encoded calcium indicators led to the establishment of whole-brain imaging methods in zebrafish, which record neural activity across a brain-wide volume with single-cell resolution. Pioneering studies of whole-brain imaging used custom light-sheet microscopes, and their operation relied on commercially developed and maintained software not available globally. Hence it has been challenging to disseminate and develop the technology in the research community. Here, we present PyZebrascope, an open-source Python platform designed for neural activity imaging in zebrafish using light-sheet microscopy. PyZebrascope has intuitive user interfaces and supports essential features for whole-brain imaging, such as two orthogonal excitation beams and eye damage prevention. Its camera module can handle image data throughput of up to 800 MB/s from camera acquisition to file writing while maintaining stable CPU and memory usage. Its modular architecture allows the inclusion of advanced algorithms for microscope control and image processing. As a proof of concept, we implemented a novel automatic algorithm for maximizing the image resolution in the brain by precisely aligning the excitation beams to the image focal plane. PyZebrascope enables whole-brain neural activity imaging in fish behaving in a virtual reality environment. Thus, PyZebrascope will help disseminate and develop light-sheet microscopy techniques in the neuroscience community and advance our understanding of whole-brain neural dynamics during animal behaviors.

Keywords: zebrafish, lightsheet microscopy, open source, software, image processing

INTRODUCTION

Animal behaviors occur through the collective actions of neurons distributed across the brain. Understanding such distributed neural dynamics in their entirety has been one of the central goals in neuroscience (Ahrens and Engert, 2015). Toward this goal, optical recording of neural activity at a brain-wide scale has become possible based on recent developments in genetically-encoded calcium indicators (Chen et al., 2013; Inoue et al., 2019) and volumetric fluorescence microscopy (Keller et al., 2008; Sofroniew et al., 2016; Voleti et al., 2019; Demas et al., 2021). Whole-brain neural activity imaging at cellular resolution was first demonstrated in larval zebrafish (Ahrens et al., 2013; Panier

et al., 2013; Freeman et al., 2014) among other vertebrate model organisms using digital scanned laser light-sheet microscopy (DSLM) (Keller et al., 2008). DSLM excites sample fluorescence in multiple voxels along the light cone of the excitation beam, and rapid scanning of the excitation beam enables fast volumetric scans with high spatial resolution and low light toxicity. These advantages of DSLM are best exploited in the optically transparent brain of larval zebrafish. DSLM enabled studies of whole-brain neural dynamics during visually-evoked swimming (Vladimirov et al., 2014; Chen et al., 2018), motor learning (Kawashima et al., 2016; Markov et al., 2021), learned helplessness (Mu et al., 2019), threat escape (Mancienne, 2021), and body posture change (Migault et al., 2018). It also revealed spontaneous noise dynamics across the brain (Ponce-Alvarez et al., 2018) and how those dynamics change during neural perturbations (Vladimirov et al., 2018; Yang, 2018) or administration of psychoactive reagents (Burgstaller et al., 2019). Further, it enabled high-speed voltage imaging on a single axial plane for recording membrane potential and spiking activity from a neural population during swimming in the midbrain (Abdelfattah et al., 2019) and the spinal cord (Böhm et al., 2022). These pioneering studies in zebrafish using DSLM expanded our understanding of diverse functionalities of the vertebrate brain.

Despite the above advantages of DSLM in achieving both high optical resolution and imaging speed compared to other large-scale volumetric imaging methods, such as structured illumination microscopy (Marques et al., 2020), multiphoton microscopy (Ahrens et al., 2012; Naumann et al., 2016; dal Maschio et al., 2017; Bruzzone, 2021), and light-field microscopy (Demas et al., 2021; Cong et al., 2017), it is still challenging to build a light-sheet microscope customized for zebrafish imaging and operate it for multiple experiments every day. These challenges come from the complexity of the microscope itself and its built-in parameters that the experimenter needs to manipulate. For example, light-sheet microscopes for whole-brain imaging in zebrafish (**Figure 1A**) typically consist of two excitation optical paths from the lateral and front sides of the fish and one optical detection path above the fish (Vladimirov et al., 2014; Migault et al., 2018; Markov et al., 2021; Mancienne, 2021) (**Figure 1B**). The fish is fixed in a water chamber and needs to be precisely maneuvered into focal points of the excitation and detection objectives. Moreover, it is necessary to prevent laser illumination into the eyes to secure fish's vision for behavioral tasks and prevent eye damage (**Figure 1B**). This configuration requires the experimenters to set at least ~20 parameters (camera exposure time per plane, number of planes per volume, the start and end positions for 2d motion for each excitation beam, light intensity and on/off timing for lasers, the start and end positions for detection objective motions, 3-dimensional positions for the fish chamber, and parameters for eye exclusion). The first studies (Ahrens et al., 2013; Vladimirov et al., 2014) on whole-brain imaging in zebrafish were made possible by using software custom-developed and maintained by a commercial entity, which charges high-priced service costs and does not provide service globally. This situation prevented the dissemination of the

technology in a flexible and customizable manner. Past progress of optical microscopy in neuroscience has been driven by open-source software (Pologruto et al., 2003; Voigt et al., 2022) for microscope control written in a programming language widely used in academics, such as ScanImage (Pologruto et al., 2003) for multiphoton microscopy. Hence, developing an open-source platform for light-sheet microscopy dedicated to neural activity imaging in zebrafish is necessary.

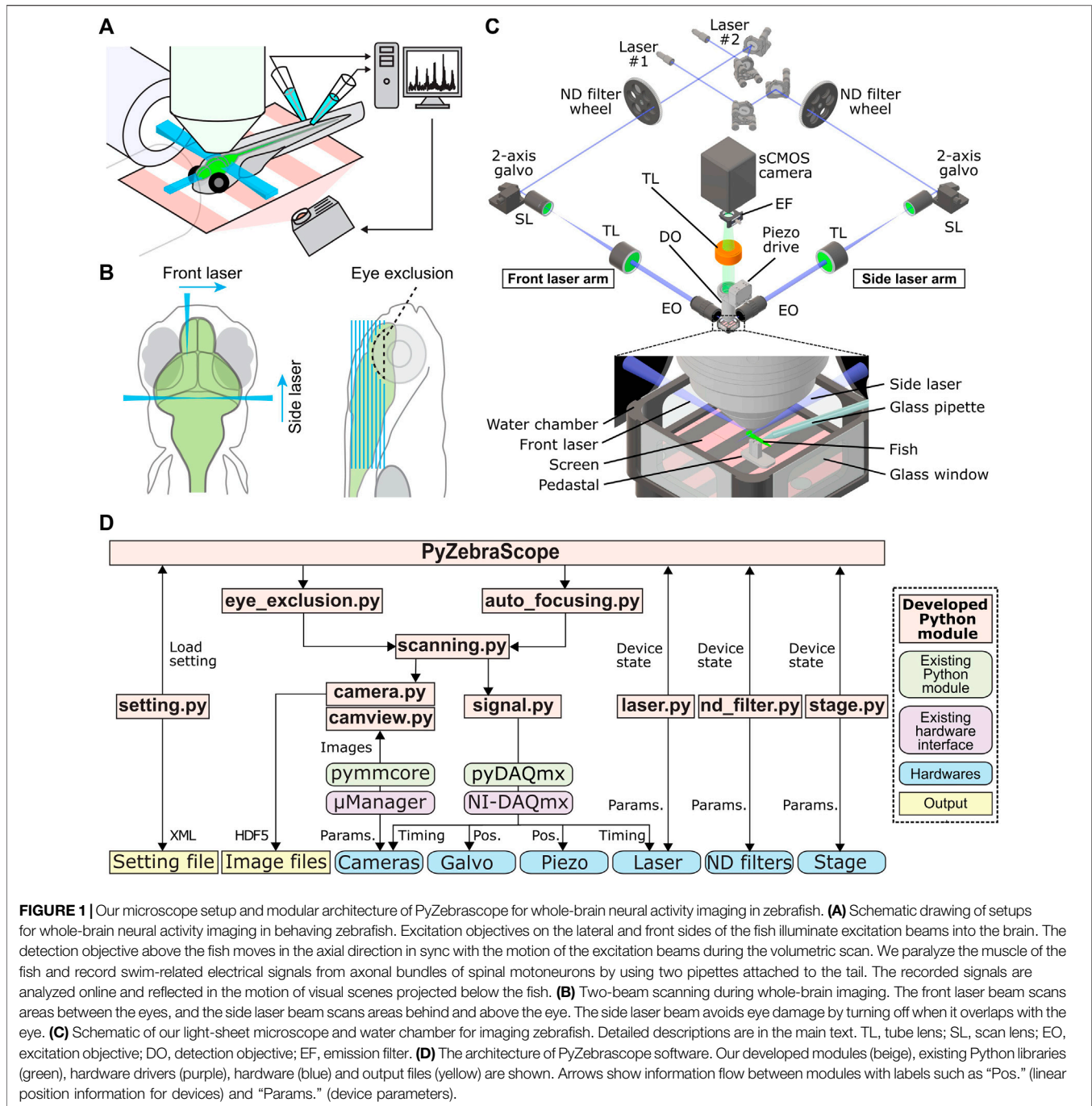
Recently, software interface for light-sheet microscopy in zebrafish, μ SPIM (Saska et al., 2021), was developed based on the open-source microscope control platform μ Manager (Edelstein et al., 2014) in Java/C programming language. Although this study only demonstrated partial volumetric imaging of the zebrafish brain based on single excitation beam scanning, it opened the possibility of developing an advanced open-source platform for light-sheet microscopy at a larger scale. Such software will reduce the laboriousness of imaging experiments by providing control of numerous device parameters on a simple and intuitive interface. Its implementation in a standard programming language for researchers will facilitate the addition of innovative functionalities toward their research goals by importing modules from a broad developer ecosystem for hardware control, signal processing, computer vision, and machine learning.

Toward this goal, we developed PyZebrascope, an open-source Python platform designed for whole-brain imaging in zebrafish using light-sheet microscopy. PyZebrascope supports essential features for whole-brain imaging, such as two orthogonal excitation beams and eye damage prevention. Its user interfaces allow the users to adjust parameters for lasers, filters, beam scanning, camera, sample positions, and eye damage prevention intuitively. Its multi-threaded camera module can handle stable high data throughput from multiple cameras. It is also possible to add advanced algorithms for microscope control and image processing developed in the Python community (Van Der Walt et al., 2014; Harris et al., 2020). As a proof of concept, we implemented a novel GPU-based automatic algorithm for maximizing the image resolution in the brain by precisely aligning the excitation beams to the image focal plane, which is usually a time-consuming and indecisive process for the experimenter. Lastly, we demonstrated that PyZebrascope enables whole-brain imaging in zebrafish behaving in a virtual reality environment. Thus, PyZebrascope is a versatile platform for disseminating and advancing technology for large-scale neural activity imaging in zebrafish and will accelerate our understanding of whole-brain neural dynamics during animal behaviors.

METHODS

Software and Hardware

PyZebrascope: All the modules of PyZebrascope were written in Python programming language on Spyder IDE (<https://www.spyder-ide.org/>). We used a PC for microscope control that



has Windows 10 operation system, two processors (Intel Xeon Gold 6,244), 384 GB of DDR4 memory, a GPU (nVidia GeForce GTX1660), and a 15 TB solid-state drive (Micron 9300 Pro, >3 GB/s writing/reading speed). We followed manufacturers' manuals for connecting devices, including the camera, data acquisition board and sample stages.

We benchmarked the resource usage and disk writing speed of PyZebraScope (**Figure 3**) by using psutil package (<https://github.com/giampaolo/psutil>) in Python. We identified the process id of PyZebraScope by using

Windows task manager and monitored its CPU and memory usage at an interval of 5 s in a separate Python instance while PyZebraScope records camera images for 30 min. Disk writing speed was also monitored using psutil by tracking the increase of disk usage in the above solid-state drive at the same time interval.

We calculated the image resolution measure in **Figure 4B** as half the wavelength of the identified maximum frequency after Fourier and polar transformations based with the below formula

$$r_{\text{image}} = \frac{[\text{image dimension}] * [\text{pixel dimension}]}{2 * F}$$

whereas [pixel dimension] is 0.405 μm per camera pixel in our setup and F is in the identified maximum frequency component (pixel distance after the polar transformation).

Light-sheet microscope: We designed a custom light-sheet microscope that has a virtual reality setup for behavioral recordings, two optical paths for excitation beams, two cameras for fluorescence detection, and additional space for future implementation of the third optical path for excitation beam and neural perturbations (**Supplementary Figure S1**). We designed this microscope using Inventor Professionals software (Autodesk), and our CAD model is available upon request. We listed commercially available parts and custom-designed parts for our light-sheet microscope in **Supplementary Table S1** and their overall costs in **Supplementary Table S2**.

Zebrafish Experiments

We used a 6-day old transgenic zebrafish that pan-neuronally express nuclear-localized, genetically-encoded calcium indicators (Dana et al., 2019) and co-express RFP in tph2^+ neurons ($\text{Tg}(\text{HuC:H2B-GCaMP7f})^{\text{j96}}$ and $\text{Tg}(\text{tph2:epNTR-TagRFP})^{\text{j41}}$, courtesy of Dr. Misha Ahrens) for the imaging experiment. The zebrafish was immobilized and mounted to an imaging chamber as described previously (Kawashima et al., 2016). Briefly, the fish larvae were immobilized by bath application of α -Bungarotoxin (B1601, Fisher Scientific, 1 mg/ml) dissolved in external solution (in mM: 134 NaCl, 2.9 KCl, 2.1 CaCl₂, 1.2 MgCl₂, 10 HEPES, 10 glucose; pH 7.8; 290 mOsm) for 25–30 s and embedded in agarose on a custom-made pedestal inside a glass-walled chamber with a diffusive screen underneath the fish (**Figure 1C**). Agarose around the head was removed with a microsurgical knife (#10318-14, Fine Science Tools) to minimize the scattering of the excitation laser. Laser power from the side beam path was, on average, approximately 21 μW . The distance between the fish and the display was about 4 mm.

We performed a fictive recording of fish's swim patterns during a motor learning task (Kawashima et al., 2016). Electric signals from motor neuron axons in the tail were recorded using borosilicate pipettes (TW150-3, World Precision Instruments) pulled by a horizontal puller (P-1000, Sutter) and fire-polished by a microforge (MF-900, Narishige). The pipettes were filled with fish-rearing water and connected to the tail using minimal negative pressure. Swim signals were recorded using an amplifier (RHD2132 amplifier connected to RHD-2000 interface board, Intan Technologies). We used custom-written Python software (available upon request) for executing the same algorithms for closed-loop virtual reality experiments as the software used in previous studies (Kawashima et al., 2016; Abdelfattah et al., 2019). It samples signals from the above amplifier at 6 kilohertz, automatically detects swim events, and moves the visual stimulus projected below the fish in a closed-loop with a delay of 35 milliseconds. The motor learning task consisted of 12 repetitions of a 157-s session, where we presented different types of tasks (7-s training, 15-s training, and 30-s training) in a pseudo-random manner.

Data analysis: We processed acquired imaging data on a Linux server in High Performance Computing (HPC) division in the Weizmann Institute of Science. This server has two Xeon processors (Xeon Gold 6,248, Intel), 384 GB RAM, 13-TB SSD array, and a GPU computing board (Tesla V100, nVidia). We performed data processing using custom Python scripts that execute the same algorithms as those established in our previous work (Kawashima et al., 2016). All the analyses of imaging data were performed on a remote JupyterLab environment (<https://jupyterlab.readthedocs.io/>).

Briefly, we first registered time-series images from the same Z-planes by using phase correlation algorithms on the above GPU. We then examined residual drifts in the lateral and axial directions and discarded data with excessive drifts ($>5 \mu\text{m}$) in either direction. We then identified individual neurons that express nuclear-localized GCaMP based on the average image by using an algorithm for detecting circular shapes in images. We then extracted fluorescent time series from the central part of identified neurons (49 pixels). We identified 79,176 neurons across the brain in the experiment described in **Figure 5**. We calculated the baseline fluorescence trace for each extracted fluorescence trace by taking the rolling percentile of the bottom 30% with a window size of 2 min and then divided the original fluorescent time series by this baseline trace to obtain $\Delta\text{F}/\text{F}$ time series for each neuron.

For the analyses shown in **Figures 5B; Supplementary Figure S5B**, we focused on neurons whose task-dependent activity modulation was consistent across multiple sessions. For extracting such neurons, we created a matrix of the average $\Delta\text{F}/\text{F}$ in each task period (12 periods) for each session and performed one-way ANOVA across trials for individual neurons. In this way, we identified 36,818 neurons with significant p -values ($p < 0.001$) for consistent task-dependent activity modulation across sessions. We then extracted neurons for each brain area by their spatial locations. We classified neurons into three groups in each area by applying a k -means clustering method ($n = 3$) to the trial average activities of neurons. We then picked a neuron that shows the largest response amplitude from each identified cluster in each brain area and plotted their time series in **Figure 5B**.

For the analysis shown in **Figure 5C**, we used the same set of 36,818 neurons identified in the above statistical test for response reliability. We normalized the $\Delta\text{F}/\text{F}$ trace of each neuron by its 99-percentile value and clipped values more than 1. We then applied non-negative matrix factorization to the activity of neurons ($n = 4$ components). We extracted neurons whose weight is more than 0.2 for each component and color-plotted their locations in the top projection and side projection images of the brain.

For the analysis shown in **Supplementary Figure S5B**, we used a volumetric stack of red fluorescence in the same fish acquired after the whole-brain imaging experiments described in **Figure 5**. We extracted 55 RFP-positive neurons among the above 36,818 neurons by using a hand-drawn mask over the area of the dorsal raphe nucleus. We identified the RFP-positive neurons by detecting the red fluorescence by a fixed threshold for mean pixel values (120). We calculated an average $\Delta\text{F}/\text{F}$ trace for different

tasks (7-s training, 15-s training, and 30-s training) across sessions for each neuron and further averaged it across them.

RESULTS

Microscope Design and the Architecture of PyZebrascope

We developed PyZebrascope for our custom-designed light-sheet microscope (Figures 1C; **Supplementary Figure S1**, **Supplementary Table S1** for parts lists), for which many components are common to those in the previous studies (Panier et al., 2013; Vladimirov et al., 2014; Mancienne, 2021; Markov et al., 2021). Our system is equipped with two lasers, providing light sources for two excitation arms that are required to scan fish brains from the lateral and front sides (**Figure 1B**). We control the lasers' on/off by digital input into the lasers. The brightnesses of laser outputs are controlled by 1) setting the power level of the laser using serial communication and 2) diminishing the laser output with different levels of neutral density (ND) filters. These two levels of brightness adjustment allow us to maintain laser output levels in a range causes less power fluctuation. Typically, the excitation beam for the front scanning (**Figure 1B**) requires less output power due to its narrow scanning range. Each brightness-adjusted beam is then scanned by sets of 2-axis galvanometers for volumetric scanning. The beams expand through a pair of a telecentric scan lens (SL) and a tube lens (TL) and then focus onto the sample through an excitation objective (EO). Fluorescence from the fish brain is collected by a detection objective (DO) placed above the brain, and this detection objective moves along the axial direction in sync with the excitation beam by a piezoelectric drive during volumetric scanning. The fluorescent image is focused onto scientific CMOS (sCMOS) camera through a tube lens (TL) and an emission filter (EF, **Figure 1C**). We can also add another sCMOS camera in the detection path for multicolor imaging, as shown in our CAD model (**Supplementary Figure S1**, CAD model files available on request).

PyZebrascope controls the above-mentioned multiple devices through its modular architecture and organized user interface (**Figures 1D**, **2**). It does not require compiling and is launchable from any Python development environment that allows PyQT applications. It controls devices through serial communications (laser setting, filter wheels, sample stage), analog output (galvanometer, camera timing), digital output (laser on/off), and camera interface cables (**Supplementary Figure S1**). The camera unit (camera.py) controls camera settings through a Python binding (pymmcore, <https://github.com/micro-manager/pymmcore>) for low-level device interface of μ Manager (Edelstein et al., 2014). It also acquires images from the camera and saves them in HDF files. The analog/digital output unit (signal.py) generates waveforms and outputs them through a Python binding (ni-daqmx, <https://nidaqmx-python.readthedocs.io/>) for data acquisition interface devices. Its outputs can be modified to drive alternative components of light-sheet microscopes that operate according to voltage inputs, such as electrically tunable lens (Favre-Bulle

et al., 2018) and scanners based on microelectromechanical systems (MEMS) (Migault et al., 2018). Other modules for controlling devices via serial communication (laser.py, nd_filter.py, piezo.py) are written in a generalizable manner so that external users can adapt them to their preferred devices.

The configuration unit (setting.py) allows the users to save all the microscope configuration parameters in an XML file. Savable parameters include the state of all connected devices, the position of the 3-axis stage, the choice of lasers and their output intensity, type of neutral density filters, the region of interest on the camera's field of view, and parameters for volumetric scanning. This file is also automatically saved for every experiment with time logs. The experimenter can load this setting file later to connect necessary devices and configure all the device parameters automatically. Although the user still needs to fine-tune sample positions and microscope scanning parameters for each experiment, this automatic loading minimizes the users' efforts in conducting experiments consistently across sessions.

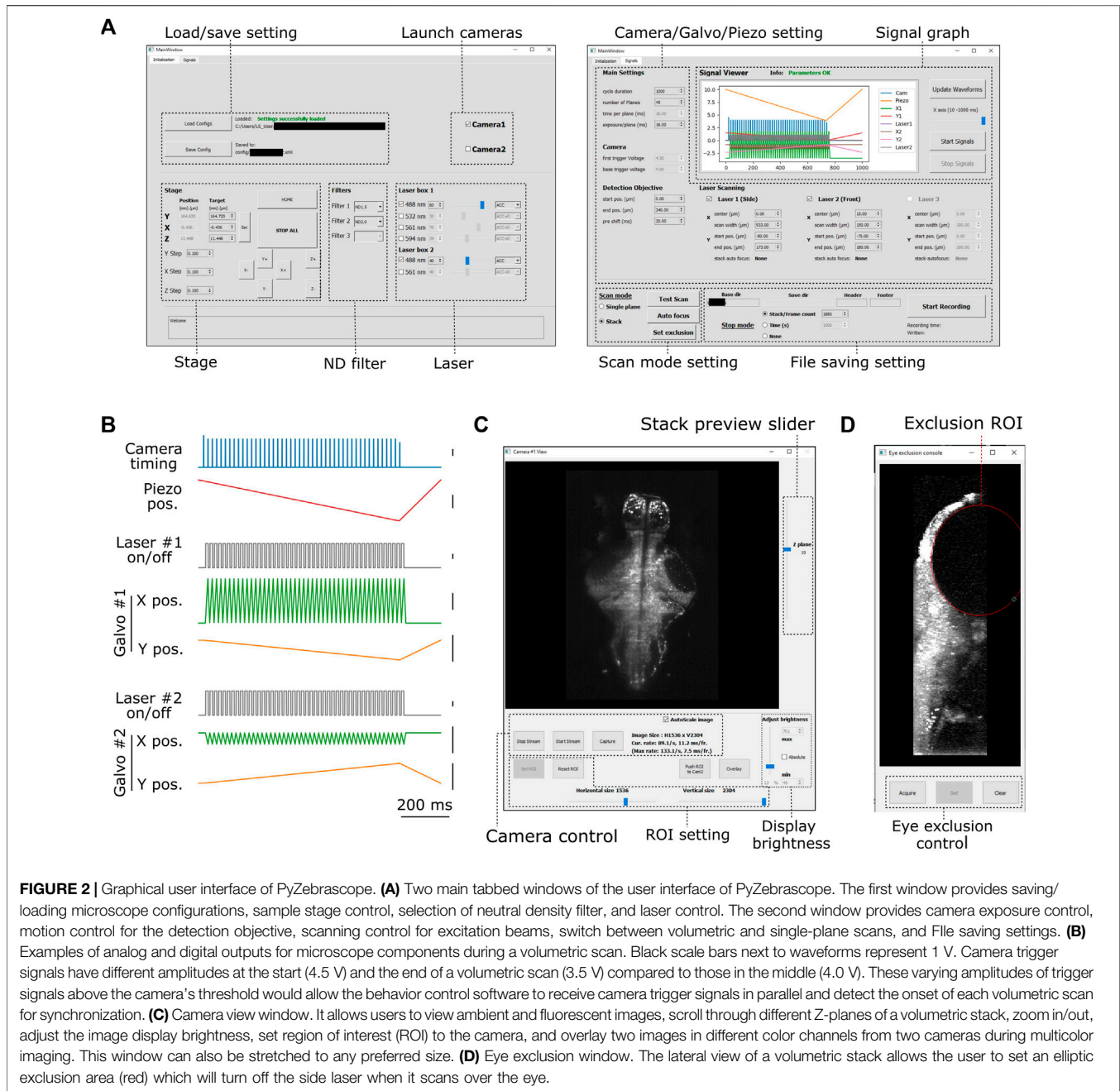
Sample Preparation and Stage Control

The fish for imaging resides Sample Stage Control a water chamber with glass windows (**Figure 1C**). Within the water chamber, the fish is embedded in low-melting agarose on a small pedestal, and this pedestal allows us to expose the fish head after removing surrounding agarose and at the same time record electrical signals from the tail for fictive swim recording (Ahrens et al., 2012; Vladimirov et al., 2014; Kawashima et al., 2016). The user mounts the fish into the water chamber outside the microscope at the beginning of each experiment. Then the chamber is placed on a motorized 3-axis stage, which brings the fish to the focus of the detection and excitation objectives (**Supplementary Figure S1**).

The PyZebrascope sample stage unit (stage.py) moves the motorized stage holding the water chamber according to the user's commands from the software interface. It also allows users to automatically move the stage between stereotypical positions, such as a home position for replacing water chambers and an imaging position for performing experiments. During imaging experiments, the water chamber, the detection objective, and the excitation objectives need to be placed in close physical proximity. Therefore, the stage unit moves the stage in a constrained manner to avoid the risk of accidental collisions between these components.

User Interface and Waveform Outputs of Pyzebrascope

We designed graphical user interfaces based on QT Designer (**Figure 2**, <https://doc.qt.io/qt-5/qt designer-manual.html>). All of the above device parameters are organized in two tabs in the main window (**Figure 2A**). Waveforms of analog and digital outputs (**Figure 2B**) can be viewed in the user interface window. We implemented a waveform generator module (signal.py) that synchronizes the beam scanning to the full opening of the rolling shutter in the sCMOS camera. It also moves the piezoelectric drives and the beam scanning in the axial



direction in a continuous manner (**Figure 2B**) rather than in a step-wise manner because the piezoelectric drive cannot move the weight of the detection objective with sub-millisecond accuracy (**Supplementary Figure S3**). Its continuous motion without frequent accelerating/decelerating events provides a better match of axial positions between the detection objective and beam scanning. Also, the waveform for moving piezoelectric drive can be shifted by a user-defined time (10–20 milliseconds) compared to those for scanning galvanometers to compensate for the movement delay between the movement of the piezoelectric drive and its analog input

(**Supplementary Figure S3**). This module uses Numpy (Harris et al., 2020), a common library for array programming for researchers, to generate waveforms and can be flexibly edited in the code (signal.py) to enable different types of scanning patterns such as bidirectional scanning for faster volumetric imaging (Saska et al., 2021).

The camera view window (**Figure 2C**, camview.py) will pop up separately when the camera is turned on. It allows the users to view the ambient and fluorescent images of the sample and adjust the display brightness and zoom. It also enables the inspection of the quality of the volumetric stack by using a slider for changing

the displayed Z planes. PyZebrascope supports two cameras in the detection path for multicolor imaging, and this camera view window can display images from two cameras separately or overlay them in RGB color channels.

PyZebrascope also has a dedicated module and interface for eye damage prevention (`eye_exclusion.py`) (**Figure 2D**). This feature is necessary for the side laser whose scanning pattern overlaps the eye during whole-brain imaging (**Figure 1B**) and not for the front laser that only needs to scan a narrow width between the eyes. In this interface, the experimenter looks at the side projection of a volumetric stack and draws an elliptic region of interest (ROI) to set where the lateral laser should be turned off (**Figure 1B**). This module automatically calculates the timing for turning off the laser across multiple Z-planes based on the set ROI. Thus, this interface prevents laser illumination into the eye while performing experiments, enabling behavioral tasks that depends on visual features presented to the fish.

Acquisition, Buffering and File Saving of Imaging Data

One of the challenges in neural activity imaging using light-sheet microscopy is its high image data throughput. Modern scientific CMOS cameras for light-sheet microscopy can acquire several megapixels per frame at the speed of over a hundred frames per second, which amounts to a data throughput of several hundred megabytes per second (MB/s). This high load poses significant challenges in ensuring continuous data flow from camera acquisition to file writing while avoiding writing delays and memory overflows.

PyZebrascope handles such flow of imaging data using three threads and three buffers for each camera (**Figure 3A**). The reading thread continuously transfers acquired images in the circular buffer of the camera to a first-in-first-out (FIFO) buffer in the system memory for file writing. This system buffer based on the Python queue module is advantageous compared to the camera's circular buffer in that it allows access from multiple threads, does not require buffer preallocation, and does not allow overwriting of previously queued data. The writing thread continuously takes the image data from this FIFO system buffer and saves them into Hierarchical Data Format 5 (HDF5) files in uncompressed form. We chose this file format because writing in HDF5 format using `h5py` library (<https://www.h5py.org/>) yielded higher data throughput than writing in other formats, such Tagged Image File Format (TIFF), at the time of our prototyping. Each HDF5 file stores a set of images for each volumetric scan or a set of 100 images for single-plane imaging. In parallel with these file writing processes, the reading thread intermittently copies image data to another FIFO buffer on system memory (**Figure 3A**). This buffer provides image data to the camera view thread for displaying acquired images on the software interface (**Figure 2C**) during imaging experiments.

This structure for handling image data, when combined with a high-speed solid-state drive (>3 GB/s writing speed, see Methods for our computer specifications), achieved a throughput of over 800 MB/s during the continuous acquisition of full-frame (2304

by 2304 pixels), 16-bit images at the speed of 80 frames per second (**Figure 3B**). PyZebrascope's CPU and memory usage remained stable during a 30-min recording, and its disk writing speed matched the data rate from the camera throughout the recording. We also achieved stable data throughput for multi-camera imaging (1,600 by 2304 pixels, 16-bit, 50 frames per second) and high-speed imaging (668 by 1,024 pixels, 16-bit, 300 frames per second). We expect it will be possible to scale up image dimensions or frame rates further by, for example, implementing a camera acquisition mode with a lower bit depth or by optimizing file saving algorithms.

Synchronization With Behavioral Recording

We synchronize behavioral control software to the image acquisition of PyZebrascope by diverging the camera trigger cable to an analog input channel for the behavioral software (**Supplementary Figure S4A**). The waveform generator (**Figure 2B**) of PyZebrascope assigns varying voltage amplitudes to camera trigger pulses above the triggering threshold (3.3 V). This varying pulse amplitude allows the behavioral software and post-processing algorithms to detect camera timings for the first axial plane in a volumetric scan (4.5 V), those for the second to the last axial planes (4.0 V), and those that terminate the acquisition of the last axial plane (3.5 V) (**Supplementary Figure S4B**). For example, behavioral software can switch the behavioral task by counting the number of volumetric stacks, rather than the number of all the pulses, by detecting 4.5 V pulses at the start of the acquisition based on voltage thresholding.

The waveform generator module (`signal.py`) allows the addition of digital outputs necessary for synchronizing image acquisition to other types of behavioral software. It is possible to add separate digital outputs for the start of the volumetric scan or individual image acquisitions as described above by adding binary arrays to the digital output channels.

Automatic Alignment of Excitation Beams to Image Focal Planes

The Python software ecosystem offers a variety of highly optimized libraries for image processing and machine learning. Such distinct advantages of using Python, as well as the modular architecture of PyZebrascope, enable the implementation of advanced algorithms for microscope control and online image analyses. As a proof of concept to demonstrate this advantage, we developed an autofocus module (`auto_focusing.py`) that adjusts the axial position of the excitation beam in the sample to the best focus of the detection objective (**Figure 4A**). This procedure is usually a time-consuming manual step during the preparation of whole-brain volumetric imaging. The experimenter needs to align the position of the excitation beam from the superficial part to the deep part of the brain. This alignment is necessary for both the front and side excitation beams. In addition, images at the bottom part of the brain are typically blurry due to the diffraction of fluorescence through the sample, which makes the alignment indecisive for the experimenter.

We implemented a novel automatic algorithm to assist such a time-consuming alignment process by using an image resolution

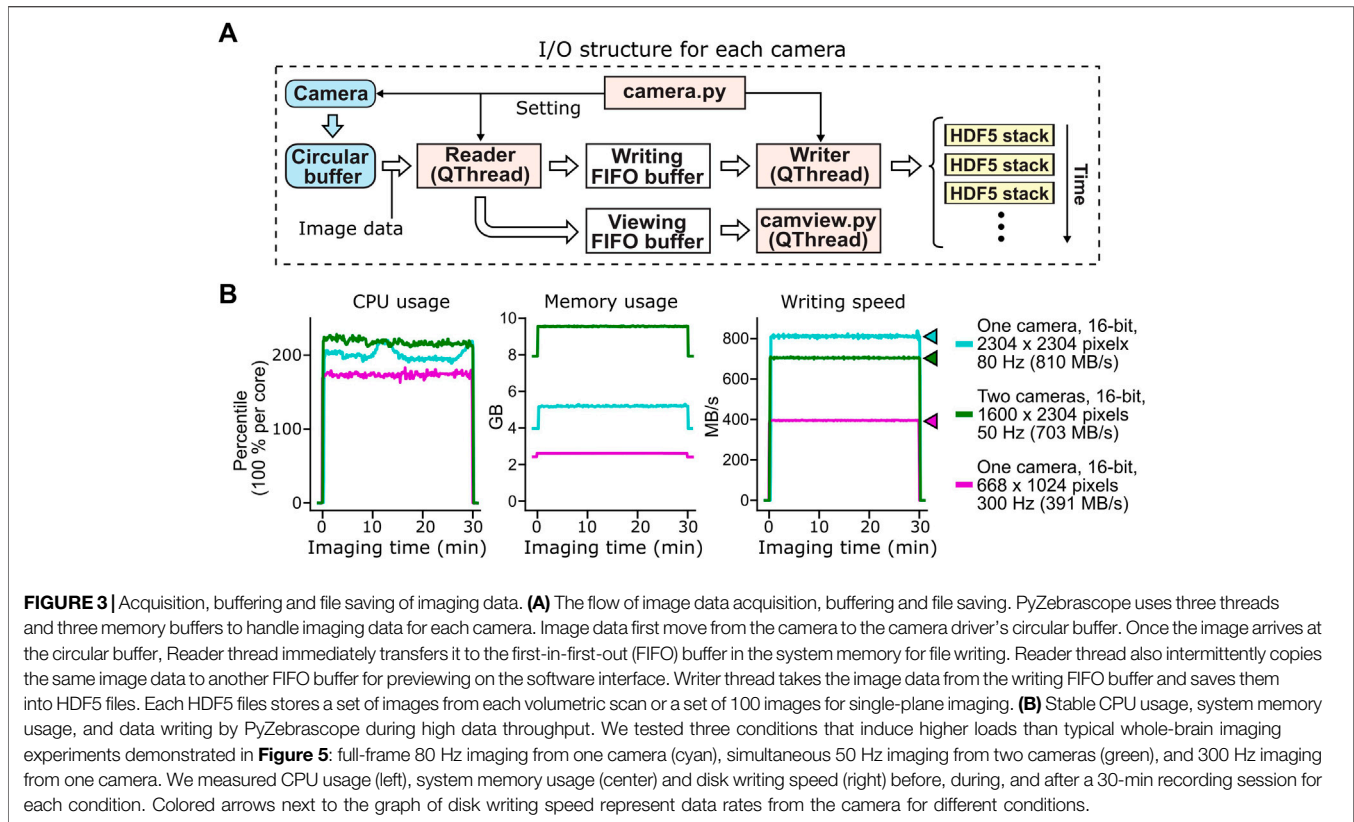


FIGURE 3 | Acquisition, buffering and file saving of imaging data. **(A)** The flow of image data acquisition, buffering and file saving. PyZebrascope uses three threads and three memory buffers to handle imaging data for each camera. Image data first move from the camera to the camera driver's circular buffer. Once the image arrives at the circular buffer, Reader thread immediately transfers it to the first-in-first-out (FIFO) buffer in the system memory for file writing. Reader thread also intermittently copies the same image data to another FIFO buffer for previewing on the software interface. Writer thread takes the image data from the writing FIFO buffer and saves them into HDF5 files. Each HDF5 files stores a set of images from each volumetric scan or a set of 100 images for single-plane imaging. **(B)** Stable CPU usage, system memory usage, and data writing by PyZebrascope during high data throughput. We tested three conditions that induce higher loads than typical whole-brain imaging experiments demonstrated in **Figure 5**: full-frame 80 Hz imaging from one camera (cyan), simultaneous 50 Hz imaging from two cameras (green), and 300 Hz imaging from one camera. We measured CPU usage (left), system memory usage (center) and disk writing speed (right) before, during, and after a 30-min recording session for each condition. Colored arrows next to the graph of disk writing speed represent data rates from the camera for different conditions.

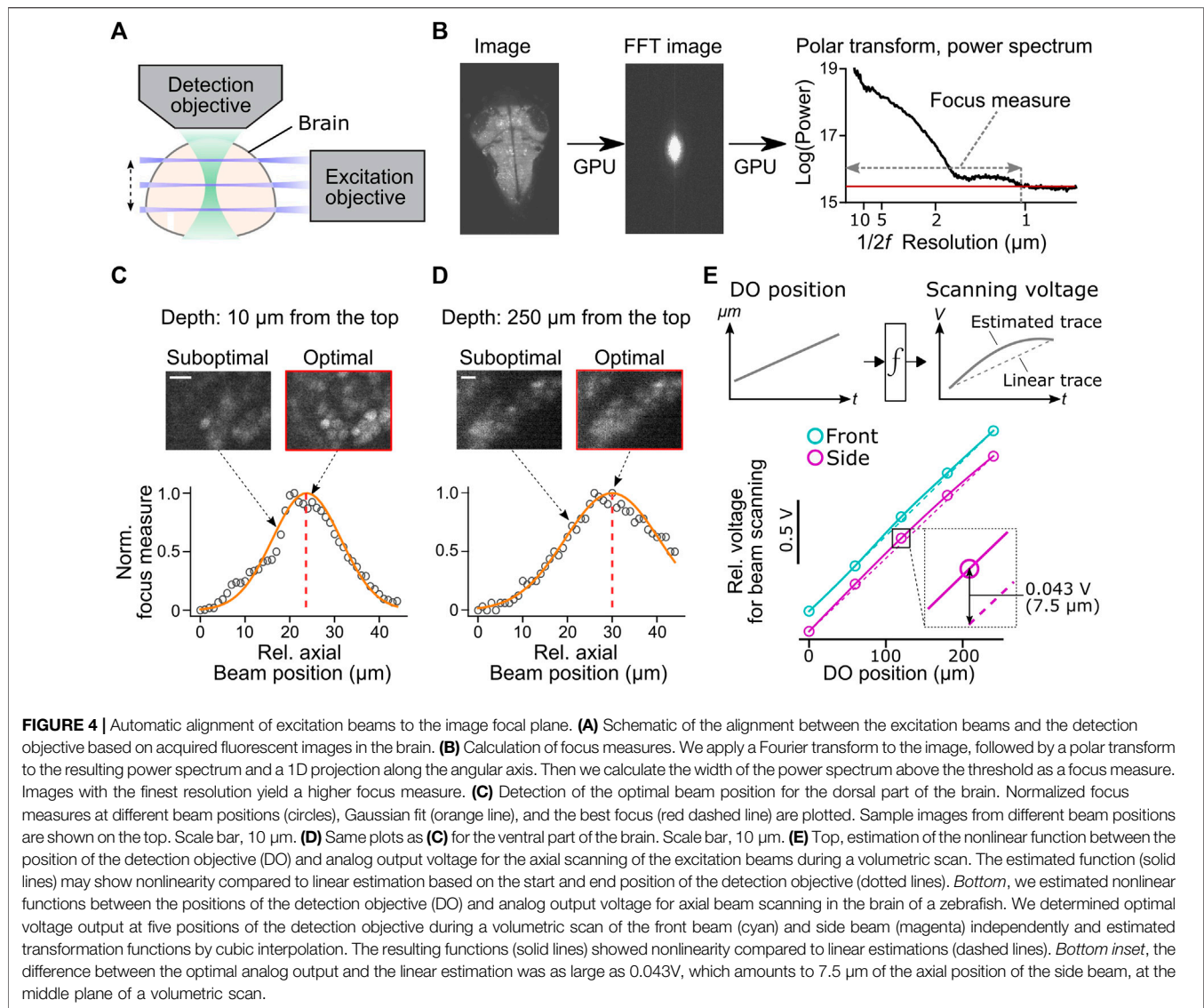
measure based on Fourier transformation (Mizutani et al., 2016) (**Figure 4B**). Optimal beam alignment yields a higher resolution of fluorescent images, which results in higher powers in the high-frequency domain in images. This algorithm first Fourier-transforms the fluorescent image to obtain its 2D power spectrum. A polar transform is applied to the 2D power spectrum and is projected onto 1D by averaging along the angular dimension. We then use the logarithm of the obtained 1D projected array to quantify image resolution. The threshold for quantifying image resolution is defined as the sum of the mean and three times the standard deviation of the baseline part of the power spectrum. We then count the number of points above the set threshold and use this number as an image resolution measure (**Figure 4B**). We implemented this algorithm using a library for GPU computing (CuPy, <https://cupy.dev/>) to minimize computation time for Fourier transform and polar transform.

Our auto-focusing module searches for the best focus by acquiring images at different axial beam positions (41 planes at an interval of 1 μm) for a given position of the detection objective (DO). It calculates the above resolution measure for each acquired plane and normalizes resolution measures to between 0 (poor resolution) and 1 (best resolution) across different planes. The peak position is detected by fitting a Gaussian distribution function to the normalized resolution measures. The center of the estimated distribution was designated as the best focal plane. These sampling and computing processes take less than 2 s in total and accurately detect the best focus for the dorsal (**Figure 4C**) and

ventral (**Figure 4D**) part of the fish brain. We further applied this technique to estimate the nonlinear function between the position of the detection objective (DO) and voltage commands for axial positioning of the excitation beams during a volumetric scan (**Figure 4E**). Such nonlinearity occurs because the angle of the scanning galvanometer is not necessarily linear to the axial position of the excitation beam. Our auto-focusing module automatically finds the best axial positions of the excitation beams, independently for the side and front beams, for five different DO positions along the Z-axis of a volumetric scan. It then estimates the optimal transformation function between the DO movements and scanning voltage for the excitation beams. Transformation functions obtained in a real zebrafish brain showed nonlinearity (**Figure 4E**), and the optimal beam position at the middle of a volumetric scan differed from the linear estimation by 7.5 μm , a large enough distance to affect image resolution significantly (**Figures 4C,D**). This estimated nonlinear function is passed to the waveform generator module (*signal.py*) as *interp1d* function of Scipy package (Virtanen, 2020) to transform the analog output waveform. Thus, this algorithm allows us to obtain accurate alignment between the objective and the excitation beam beyond manual, linear adjustments based on the start and end position of the volumetric acquisition.

Whole-Brain Imaging in Behaving Zebrafish

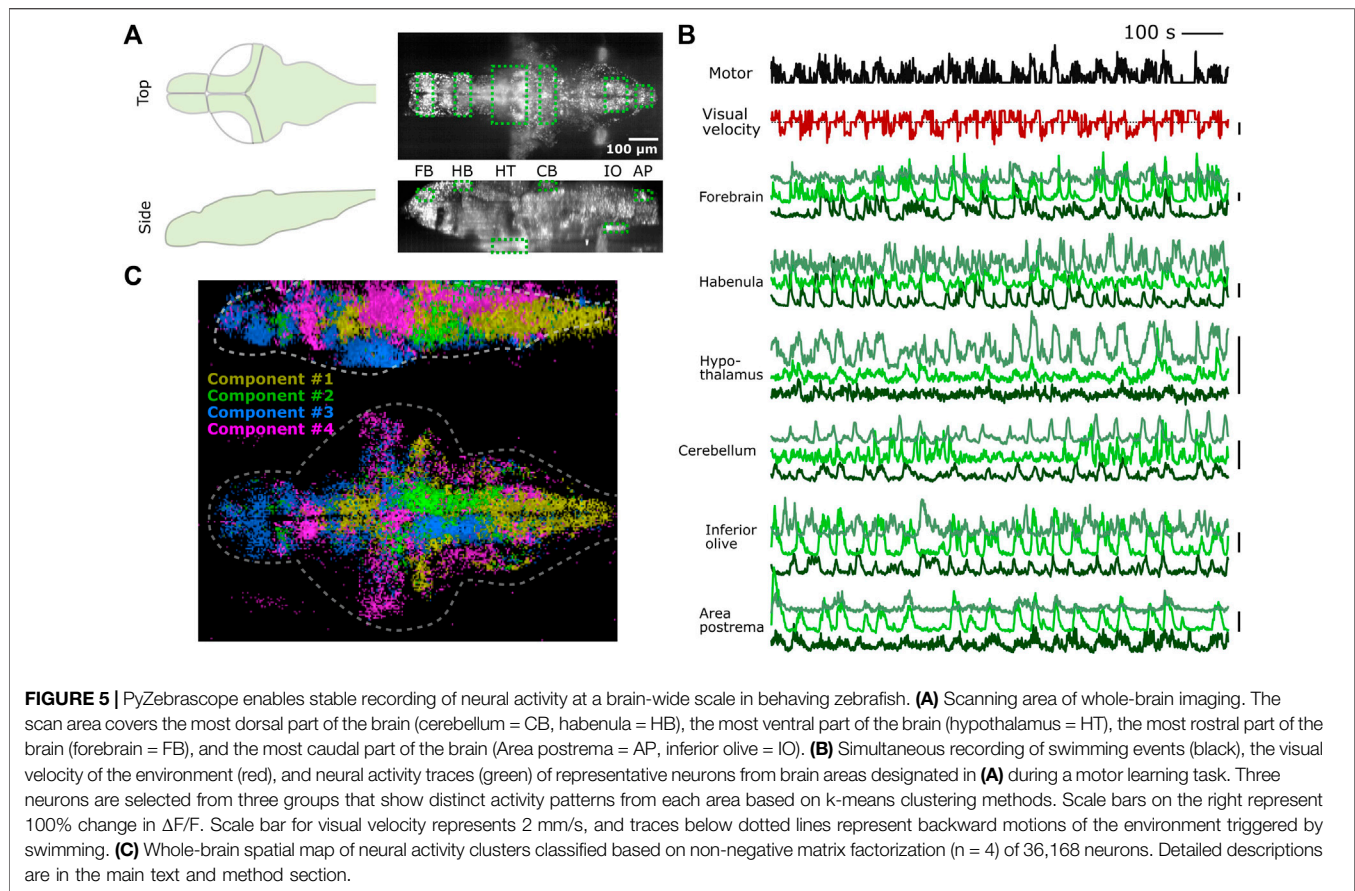
We tested whether PyZebrascope can perform whole-brain imaging in zebrafish while recording its behavior during a



behavioral task (Figure 5A). We used a 6-day old transgenic zebrafish that pan-neuronally express nuclear-localized, genetically-encoded calcium indicators (Dana et al., 2019) (HuC:H2B-GCaMP7f) and co-express red fluorescent proteins in *tph2+* serotonergic neurons (Supplementary Figure S5A). The tested fish was able to react to the visual stimuli presented from below (Figure 5B) and performed the motor learning task described in our previous work (Kawashima et al., 2016). We were able to acquire continuous volumetric scans (45 planes at 1 Hz) across the entire brain (Figure 5A), from the extremities of the dorsal to the ventral regions (cerebellum and hypothalamus, respectively) and from extremities of the rostral to the caudal regions (forebrain and area postrema, respectively).

We analyzed the acquired data by using image registration and segmentation algorithms developed in our previous work (Kawashima et al., 2016). We identified ~80,000 neurons

across the brain, and their neural activity patterns remained stable for the entire duration of the experiment from the above areas of the brain (Figure 5B). These results demonstrate PyZebrascope's robustness for continuously acquiring large volumetric data over time. We tested the reliability of the data acquired by PyZebrascope by examining whether it is possible to identify behavior-related neural ensembles identified in other studies. We applied non-negative matrix factorization (4 set components) to the neural activity of all neurons across the entire brain and mapped the spatial locations of neurons that have significant weights for the identified components (Figure 5C). We were able to identify a swim-related network in the midbrain and the hindbrain (Orger et al., 2008; Vladimirov et al., 2014) (Component 1), a network in the hindbrain that biases the fish's swimming to the left side and to the right side (Ahrens et al., 2013; Dunn et al., 2016)



(Component 2, 3), and neurons in the optic tectum, the dorsal raphe and the thalamus that respond to visual feedback during motor learning task (Kawashima et al., 2016) (Component 4).

Additionally, we analyzed the activity patterns of *tph2+* serotonergic neurons in the dorsal raphe nucleus that mediate motor learning effect in this behavioral task (Kawashima et al., 2016). We identified 55 RFP+ neurons in the DRN whose task-dependent activity modulation was consistent across multiple sessions (see Methods). Their activity patterns show slow integration during a training period when the fish learns weak swim patterns. Their activity then slowly decay during the delay period of no swimming (Supplementary Figure S5B). These patterns are consistent with the finding of our previous work (Kawashima et al., 2016). These results demonstrate that our open-source PyZebrascope allows us to perform whole-brain neural activity imaging in behaving zebrafish in a quality comparable with the pioneering studies that relied on commercially developed software.

DISCUSSION

Here we described the development of the open-source Python software, PyZebrascope, that controls a light-sheet microscope designed for neural activity imaging experiments in zebrafish. Its intuitive graphical user interfaces and ease of managing complex

device parameters allow the users to minimize efforts in setting up consistent whole-brain imaging experiments across experiments. Its multi-thread structure for handling imaging data achieved a high throughput of over 800 MB/s of camera acquisition and file writing while maintaining stable usage of CPU and memory resources. The choice of Python as a programming language allowed us to implement an advanced microscope control algorithm, such as automatic focusing, which further reduces the tedious efforts for ensuring the quality of multi-beam volumetric scans. These features allowed us to perform whole-brain imaging in behaving zebrafish at a significantly larger scale than the first demonstration of open-source software for light-sheet microscopy in zebrafish (Saska et al., 2021). Its data quality reached at least a comparable level to those demonstrated by commercially developed software (Vladimirov et al., 2014; Kawashima et al., 2016).

PyZebrascope and its components are written in a generalizable manner, enabling research teams that use different types of devices to adapt the code to their configurations. It is available from Github (https://github.com/KawashimaLab/PyZebraScope_public). It only requires pre-installation of Anaconda Python package, μ Manager package (Edelstein et al., 2014) with a matching version of its Python interface (pymmcore), a Python library for controlling data acquisition board (ni-daqmx), a Python library for GPU computing (CuPy), and a few other Python packages, all free of cost. Therefore, we expect that PyZebrascope will help disseminate

the whole-brain imaging technique throughout the zebrafish neuroscience community.

The architecture of PyZebrascope further enables the implementation of advanced microscope control and image processing algorithms. For example, a common issue during whole-brain imaging in zebrafish is the sample's translational and rotational drift resulting from gravity force, tail motions of unparalyzed fish (Markov et al., 2021), or the pressure of pipette attachment during fictive recording in paralyzed fish (Ahrens et al., 2012; Vladimirov et al., 2014; Kawashima et al., 2016). A small amount of drift, especially in the axial direction along which the volumetric scan is under-sampled, can result in the loss of neurons during imaging because the neuronal diameter in the zebrafish brain is usually less than 5 μm . Our previous work detected such drifts during time-consuming post processing (Kawashima et al., 2016). Given that sample drift usually occurs at a slow rate, it will be possible to monitor it in real-time during the experiment by occasionally sampling the scanned volume. It will also be possible to further compensate for translational drifts by moving the 3-axis stage holding the sample. The effectiveness of such online drift correction is demonstrated in a multi-beam light-sheet microscope (Royer et al., 2015). Therefore, implementing such algorithms likely to increase the throughput and the success rate of whole-brain imaging experiments in zebrafish.

It will also be possible to implement real-time image processing features to identify neurons of specific activity patterns for subsequent neural perturbation experiments at a single-cell resolution (dal Maschio et al., 2017; Vladimirov et al., 2018). Real-time registration and segmentation of individual neural activity across the brain will require large computing resources and may not be feasible on the microscope control computer. Nonetheless, it will be possible to calculate, for example, trial-averaged neural activation maps based on a simple recurrent formula if the behavioral events or sensory stimulus occur at regular volumetric scan intervals. Once we identify neurons that show specific activity patterns, we can further laser-ablate or photostimulate these populations using an open-source Python resource for holographic two-photon stimulation (dal Maschio et al., 2017; Pozzi and Mapelli, 2021). Such experiments will allow us to investigate the functional roles of neurons that show specific activity patterns beyond what is obtainable by genetic labeling of neurons.

Lastly, PyZebrascope will enable further development of voltage imaging techniques in zebrafish. Voltage imaging requires high image resolution, high photon collection efficiency, and high imaging speed at the rate of at least 300 frames per second (Cohen and Venkatachalam, 2014; Abdelfattah et al., 2019; Böhm et al., 2022). Light-sheet microscopy has the potential of realizing these conditions at a brain-wide scale in zebrafish. For example, it will be possible to implement a custom module that performs single-plane high-speed imaging in a sequential manner across different depth while the zebrafish perform a simple, stereotypical sensorimotor task as demonstrated for calcium imaging in zebrafish (Ahrens et al., 2012). Implementing such a capability will advance our understanding of how whole-brain neural dynamics control animal behaviors at a millisecond timescale.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

ETHICS STATEMENT

The animal study was reviewed and approved by Institutional Animal Care and Use Committee (IACUC) and the Institutional Biosafety Committee (IBC) of the Weizmann Institute of Science.

AUTHOR CONTRIBUTIONS

TK and RB conceived the project. RB established the device control and first versions of PyZebrascope. TK designed the custom light-sheet microscope, organized the later versions of PyZebrascope and conducted data analyses. RH provided initial user feedback for refining PyZebrascope and collected whole-brain imaging data in behaving zebrafish. MK, and KH developed algorithms for automatic image focusing and contributed a figure. TK wrote the manuscript with inputs from all authors.

FUNDING

MNK is supported by iNAMES MDC-Weizmann Research School, "Imaging from NAno to MESo." This research is supported by Azrieli faculty fellowship (TK), Dan Lebas Ruth Sonnewend (TK), Birnbach Family foundation (TK), Internal grant from the Center for New Scientists at Weizmann Institute of Science (TK), Helmholtz Imaging (KH), and an Oak Ridge National Laboratory, Laboratory Directed Research and Development Strategic Hire grant (KH).

ACKNOWLEDGMENTS

We thank Dorel Malamud and other members of Kawashima laboratories for experimental help, members of Veterinary Resource at Weizmann Institute of Science for animal care, Meir Alon, Haim Sade, Alex Jahanfard and members of Physics Core Facility at Weizmann Institute of Science for machining and assembling custom microscope parts, and Uri Rosen for the maintenance of our computing server in the High Performance Computing unit. We thank Misha Ahrens, Igor Negrashov, Davis Bennet, Raghav Chhetri, Nikita Vladimirov and Steven Sawtelle (HHMI Janelia) for their advice on our initial design of the microscope components and PyZebrascope, and Niels Cautauts (Data Minded) for his advice on GPU-based codes. We thank Inbal Shainer for her critical reading of the manuscript.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fcell.2022.875044/full#supplementary-material>

REFERENCES

- Abdelfattah, A. S., Kawashima, T., Singh, A., Novak, O., Liu, H., Shuai, Y., et al. (2019). Bright and Photostable Chemigenetic Indicators for Extended *In Vivo* Voltage Imaging. *Science* 365, 699–704. doi:10.1126/science.aav6416
- Ahrens, M. B., and Engert, F. (2015). Large-Scale Imaging in Small Brains. *Curr. Opin. Neurobiol.* 32, 78–86. doi:10.1016/j.conb.2015.01.007
- Ahrens, M. B., Li, J. M., Orger, M. B., Robson, D. N., Schier, A. F., Engert, F., et al. (2012). Brain-Wide Neuronal Dynamics During Motor Adaptation in Zebrafish. *Nature* 485, 471–477. doi:10.1038/nature11057
- Ahrens, M. B., Orger, M. B., Robson, D. N., Li, J. M., and Keller, P. J. (2013). Whole-Brain Functional Imaging at Cellular Resolution Using Light-Sheet Microscopy. *Nat. Methods* 10, 413–420. doi:10.1038/nmeth.2434
- Böhmer, U. L., Kimura, Y., Kawashima, T., Ahrens, M. B., Higashijima, S. I., and Engert, F. (2022). Voltage Imaging Identifies Spinal Circuits That Modulate Locomotor Adaptation in Zebrafish. *Neuron* 110, 1211–1222. doi:10.1016/j.neuron.2022.01.001
- Bruzzone, M. (2021). Whole Brain Functional Recordings at Cellular Resolution in Zebrafish Larvae with 3D Scanning Multiphoton Microscopy. *Sci. Rep.* 11 (11), 1–11. doi:10.1038/s41598-021-90335-y
- Burgstaller, J., Hindinger, E., Donovan, J., Dal Maschio, M., Kist, A. M., Gesierich, B., et al. (2019). Light-Sheet Imaging and Graph Analysis of Antidepressant Action in the Larval Zebrafish Brain Network. *bioRxiv*, 618843. doi:10.1101/618843
- Chen, T.-W., Wardill, T. J., Sun, Y., Pulver, S. R., Renninger, S. L., Baohan, A., et al. (2013). Ultrasensitive Fluorescent Proteins for Imaging Neuronal Activity. *Nature* 499, 295–300. doi:10.1038/nature12354
- Chen, X., Mu, Y., Hu, Y., Kuan, A. T., Nikitchenko, M., Randlett, O., et al. (2018). Brain-wide Organization of Neuronal Activity and Convergent Sensorimotor Transformations in Larval Zebrafish. *Neuron* 100, 876e5–890. doi:10.1016/j.neuron.2018.09.042
- Cohen, A. E., and Venkatachalam, V. (2014). Bringing Bioelectricity to Light. *Annu. Rev. Biophys.* 43, 211–232. doi:10.1146/annurev-biophys-051013-022717
- Cong, L., Wang, Z., Chai, Y., Hang, W., Shang, C., Yang, W., et al. (2017). Rapid Whole Brain Imaging of Neural Activity in Freely Behaving Larval Zebrafish (*Danio rerio*). *Elife* 6, e28158. doi:10.7554/eLife.28158
- dal Maschio, M., Donovan, J. C., Helmbrecht, T. O., and Baier, H. (2017). Linking Neurons to Network Function and Behavior by Two-Photon Holographic Optogenetics and Volumetric Imaging. *Neuron* 94, 774e5–789. doi:10.1016/j.neuron.2017.04.034
- Dana, H., Sun, Y., Mohar, B., Hulse, B. K., Kerlin, A. M., Hasseman, J. P., et al. (2019). High-Performance Calcium Sensors for Imaging Activity in Neuronal Populations and Microcompartments. *Nat. Methods* 16, 649–657. doi:10.1038/s41592-019-0435-6
- Demas, J., Manley, J., Tejera, F., Barber, K., Kim, H., Traub, F. M., et al. (2021). High-speed, Cortex-wide Volumetric Recording of Neuroactivity at Cellular Resolution Using Light Beads Microscopy. *Nat. Methods* 18, 1103–1111. doi:10.1038/s41592-021-01239-8
- Dunn, T. W., Gebhardt, C., Naumann, E. A., Riegler, C., Ahrens, M. B., Engert, F., et al. (2016). Neural Circuits Underlying Visually Evoked Escapes in Larval Zebrafish. *Neuron* 89, 613–628. doi:10.1016/j.neuron.2015.12.021
- Edelstein, A. D., Tsuchida, M. A., Amodaj, N., Pinkard, H., Vale, R. D., and Stuurman, N. (2014). Advanced Methods of Microscope Control Using μ Manager Software. *J. Biol. Methods* 1, 10. doi:10.14440/jbm.2014.36
- Favre-Bulle, I. A., Vanwallegem, G., Taylor, M. A., Rubinsztein-Dunlop, H., and Scott, E. K. (2018). Cellular-Resolution Imaging of Vestibular Processing Across the Larval Zebrafish Brain. *Curr. Biol.* 28, 3711–3722. doi:10.1016/j.cub.2018.09.060
- Freeman, J., Vladimirov, N., Kawashima, T., Mu, Y., Sofroniew, N. J., Bennett, D. V., et al. (2014). Mapping Brain Activity at Scale with Cluster Computing. *Nat. Methods* 11, 941–950. doi:10.1038/nmeth.3041
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array Programming with NumPy. *Nature* 585, 357–362. doi:10.1038/s41586-020-2649-2
- Inoue, M., Takeuchi, A., Manita, S., Horigane, S.-i., Sakamoto, M., Kawakami, R., et al. (2019). Rational Engineering of XCaMPs, a Multicolor GECI Suite for *In Vivo* Imaging of Complex Brain Circuit Dynamics. *Cell* 177, 1346e24–1360. doi:10.1016/j.cell.2019.04.007
- Kawashima, T., Zwart, M. F., Yang, C.-T., Mensh, B. D., and Ahrens, M. B. (2016). The Serotonergic System Tracks the Outcomes of Actions to Mediate Short-Term Motor Learning. *Cell* 167, 933–946. doi:10.1016/j.cell.2016.09.055
- Keller, P. J., Schmidt, A. D., Wittbrodt, J., and Stelzer, E. H. K. (2008). Reconstruction of Zebrafish Early Embryonic Development by Scanned Light Sheet Microscopy. *Science* 322, 1065–1069. doi:10.1126/science.1162493
- Mancienne, T. (2021). Contributions of Luminance and Motion to Visual Escape and Habituation in Larval Zebrafish. *Front. Neural Circuits* 15, 115. doi:10.3389/fncir.2021.748535
- Markov, D. A., Petrucco, L., Kist, A. M., and Portugues, R. (2021). A Cerebellar Internal Model Calibrates a Feedback Controller Involved in Sensorimotor Control. *Nat. Commun.* 12, 1–21. doi:10.1038/s41467-021-26988-0
- Marques, J. C., Li, M., Schaak, D., Robson, D. N., and Li, J. M. (2020). Internal State Dynamics Shape Brainwide Activity and Foraging Behaviour. *Nature* 577, 239–243. doi:10.1038/s41586-019-1858-z
- Migault, G., van der Plas, T. L., Trentesaux, H., Panier, T., Candelier, R., Proville, R., et al. (2018). Whole-Brain Calcium Imaging During Physiological Vestibular Stimulation in Larval Zebrafish. *Curr. Biol.* 28, 3723–3735. doi:10.1016/j.cub.2018.10.017
- Mizutani, R., Saiga, R., Takekoshi, S., Inomoto, C., Nakamura, N., Itokawa, M., et al. (2016). A Method for Estimating Spatial Resolution of Real Image in the Fourier Domain. *J. Microsc.* 261, 57–66. doi:10.1111/jmi.12315
- Mu, Y., Bennett, D. V., Rubinov, M., Narayan, S., Yang, C.-T., Tanimoto, M., et al. (2019). Glia Accumulate Evidence that Actions Are Futile and Suppress Unsuccessful Behavior. *Cell* 178, 27e19–43. doi:10.1016/j.cell.2019.05.050
- Naumann, E. A., Fitzgerald, J. E., Dunn, T. W., Rihel, J., Sompolinsky, H., and Engert, F. (2016). From Whole-Brain Data to Functional Circuit Models: The Zebrafish Optomotor Response. *Cell* 167, 947–960. doi:10.1016/j.cell.2016.10.019
- Orger, M. B., Kampff, A. R., Severi, K. E., Bollmann, J. H., and Engert, F. (2008). Control of Visually Guided Behavior by Distinct Populations of Spinal Projection Neurons. *Nat. Neurosci.* 11, 327–333. doi:10.1038/nn2048
- Panier, T., Romano, S. A., Olive, R., Pietri, T., Sumbre, G., Candelier, R., et al. (2013). Fast Functional Imaging of Multiple Brain Regions in Intact Zebrafish Larvae Using Selective Plane Illumination Microscopy. *Front. Neural Circuits* 7, 65. doi:10.3389/fncir.2013.00065
- Pologruto, T. A., Sabatini, B. L., and Svoboda, K. (2003). ScanImage: Flexible Software for Operating Laser Scanning Microscopes. *Biomed. Eng. Online* 2, 13–19. doi:10.1186/1475-925X-2-13
- Ponce-Alvarez, A., Jouary, A., Privat, M., Deco, G., and Sumbre, G. (2018). Whole-Brain Neuronal Activity Displays Crackling Noise Dynamics. *Neuron* 100, 1446–e6. doi:10.1016/j.neuron.2018.10.045
- Pozzi, P., and Mapelli, J. (2021). Real Time Generation of Three Dimensional Patterns for Multiphoton Stimulation. *Front. Cell. Neurosci.* 15, 34. doi:10.3389/fncel.2021.609505
- Royer, L. A., Weigert, M., Günther, U., Maghelli, N., Jug, F., Sbalzarini, I. F., et al. (2015). ClearVolume: Open-Source Live 3D Visualization for Light-Sheet Microscopy. *Nat. Methods* 12, 480–481. doi:10.1038/nmeth.3372
- Saska, D., Pichler, P., Qian, C., Buckley, C. L., and LagnadoToolset, L. (2021). μ SPIM Toolset: A Software Platform for Selective Plane Illumination Microscopy. *J. Neurosci. Methods* 347, 108952. doi:10.1016/j.jneumeth.2020.108952
- Sofroniew, N. J., Flickinger, D., King, J., and Svoboda, K. (2016). A Large Field of View Two-Photon Mesoscope with Subcellular Resolution for *In Vivo* Imaging. *Elife* 5, e14472. doi:10.7554/eLife.14472
- Van Der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., et al. (2014). Scikit-image: Image Processing in python. *PeerJ* 2, e453. doi:10.7717/peerj.453
- Virtanen, P. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* 17, 261–272.
- Vladimirov, N., Mu, Y., Kawashima, T., Bennett, D. V., Yang, C.-T., Looger, L. L., et al. (2014). Light-sheet Functional Imaging in Fictively Behaving Zebrafish. *Nat. Methods* 11, 883–884. doi:10.1038/nmeth.3040
- Vladimirov, N., Wang, C., Höckendorf, B., Pujala, A., Tanimoto, M., Mu, Y., et al. (2018). Brain-wide Circuit Interrogation at the Cellular Level Guided by Online Analysis of Neuronal Function. *Nat. Methods* 15, 1117–1125. doi:10.1038/s41592-018-0221-x

- Voigt, F. F., Vladimirov, N., Schulze, C., Campbell, R., and Helmchen, F. (2022). MesoSPIM Control: An Open-Source Acquisition Software for Light-Sheet Microscopy Written in Python and Qt. *Zenodo*. doi:10.5281/ZENODO.6109315
- Voleti, V., Patel, K. B., Li, W., Perez Campos, C., Bharadwaj, S., Yu, H., et al. (2019). Real-Time Volumetric Microscopy of *In Vivo* Dynamics and Large-Scale Samples with SCAPE 2.0. *Nat. Methods* 16, 1054–1062. doi:10.1038/s41592-019-0579-4
- Yang, C. (2018). All-Optical Imaging and Manipulation of Whole-Brain Neuronal Activities in Behaving Larval Zebrafish. *Biomed. Opt. Express* 9 (Issue 12), 6154–6169. doi:10.1364/BOE.9.006154

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Barbara, Nagathihalli Kantharaju, Haruvi, Harrington and Kawashima. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.