zOPT: an open source Optical Projection Tomography system and methods for rapid 3D zebrafish imaging
HANQING ZHANG, LAURA WALDMANN, REMY MANUEL, TATJANA HAITINA, AMIN ALLALOU,
Uppsala University, Sweden.

zOPT user manual

This manual contains two sections to help you quickly get started with building up the setup and conducting experiment using our zOPT.

1. Experimental setup and GUI

To build an zOPT system, you need to purchase the components listed in Table.1. The list contains light source, lenses and cameras that we recommend. A 3D printer will be needed due to parts of the component are 3D printed. The .stl files for our 3D printable components can be found in https://github.com/Hq-Z/zOPT. We used the original version of Ultimaker and PLA (black, white) to print our components. After all the components are ready, we can build system as presented in Figure.1. In this figure, the optical components and sample stage is built on top of an aluminum breadboard with holes for fixing positions, while the white breadboard is used to connect wires for Arduino and drivers to control the stepper motor. The arrangement of the 3D components is presented in Figure. 2. The component 1 in Figure.2 is glued to a glass tapered capillary as shown in Figure.1. Other components can be fixed together either by glues or by nails. For connection amongst the Arduino, the driver, the stepper motor and the computer, please refer to the user manual from the manufacturer. After all electrical components are connected correctly, we can upload the codes to the Arduino for controlling the motor and a MATLAB GUI to complete the OPT recording.

	Item	Details	Cost €
1	Broadband LED	LED clamp spotlight, warm white LED, IKEA	13
2	Ground glass diffuser	AR-Coated, N-BK7 Ground Glass Diffusers, Thorlabs	12
3	Quartz cuvette	#CG-F-20, The Science Outlet, Keota, OK	15
4	Translation stage (optional)	Unknown manufacturer, Equivalent to RB13M, 3 Axis RollerBlock™ Long-Travel Stages,Thorlabs	(1388.91)
5	Stepper motor	Jugetek, 39BYG55022-13K	12
6	Motor drive carrier	AMIS-30543, Pololu	14.3
7	Glass capillary (optional)	40564010, Hecht Assistant; inner diameter 1.15 mm, outer disameter 1.55	-
	Glass capillary (optional)	9530-2, PYREX, inner diamter 0.8-1.1 mm (2000 Pieces)	(180)
	Glass capillary (optional)	B150-86-15, SUTTER INSTRUMENT,inner diameter 0.86 mm, outer diameter 1.5 mm (250 Pieces)	(65.5)
8	Stage framework and sample holder	3D printable components	5
9	Arduino board	ARDUINO UNO REV3	28.4
10	Aluminum breadboard (optional)	Unknown manufacturer, Equivalent to MB2025/M, Thorlabs	(111.15)
11	Telecentric lens system	3X, 110 mm CompactTL, Edmund Optics, depth of field 0.36 mm	625
12	CCD camera	FLIR Blackfly S BFSU3-23S3C, pixel size of 3.45 × 3.45 µm	565

Table. S1 Component list of our customized OPT.

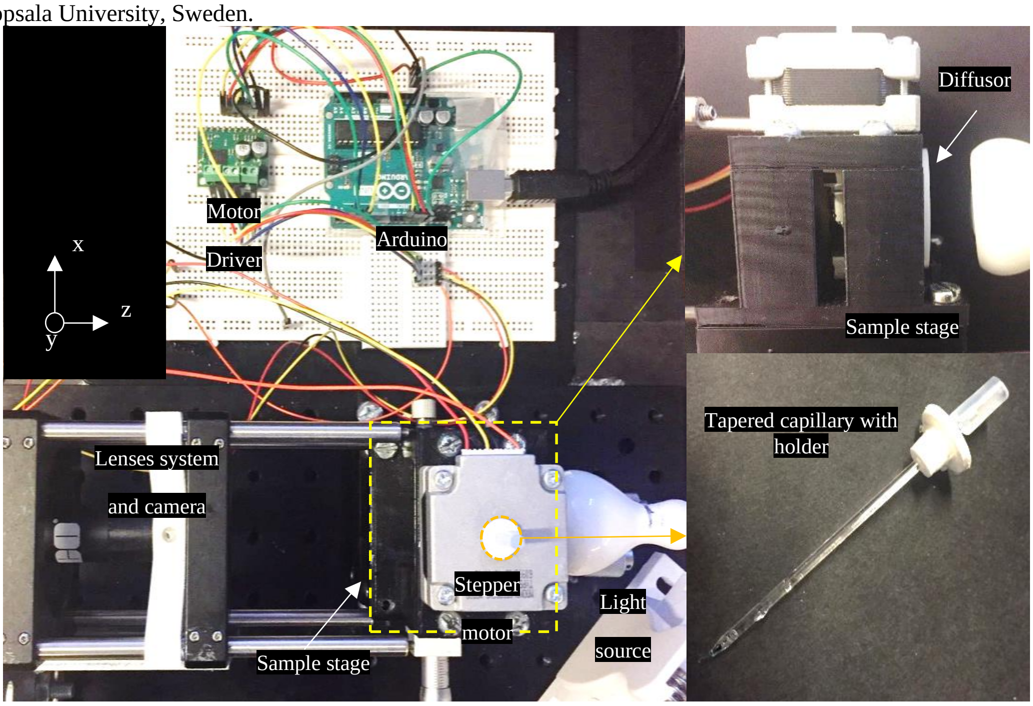

Figure.1 Experimental setup of the brightfield OPT system. The dashed square indicates the area of sample stage from the top view and the side view of the stage can be found on the top right. The dashed circle shows the position of the capillary that goes through the stepper motor and the entire structure of the tapered capillary, holder for stepper motor and inlet-outlet port is shown on the bottom right.

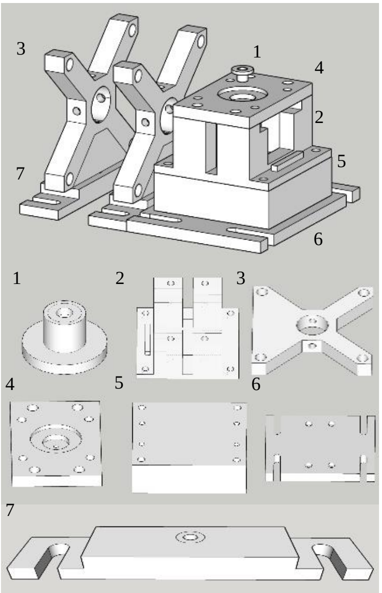

Figure. 2. Schematics of 3D printable components for the OPT system. The image above shows when printable components are placed together. Note that the position of each component can be fixed by the user using screws, glue or tape.

2. MATLAB GUI

To operate on the system, one need to install the Arduino IDE and MATLAB on your computer. We provide a MATLAB based GUI as shown in Figure. 3. The GUI allows us to control the stepper motor and to rotate the sample in real-time and record OPT data.

In our implementation, all connections to the computer is based on the USB port and we used drivers that are designed for the FLIR camera (USB 3.0) and used serial communication with the Arduino. We set the frame rate to 53 frames per second by default using the driver for FLIR Blackfly Camera in MATLAB. The specific camera settings, such as exposure time and color balance, can be changed manually using the FLIR SpinView software. Other cameras can also be used with the GUI. It searches for existing camera options if the FLIR camera cannot be found.

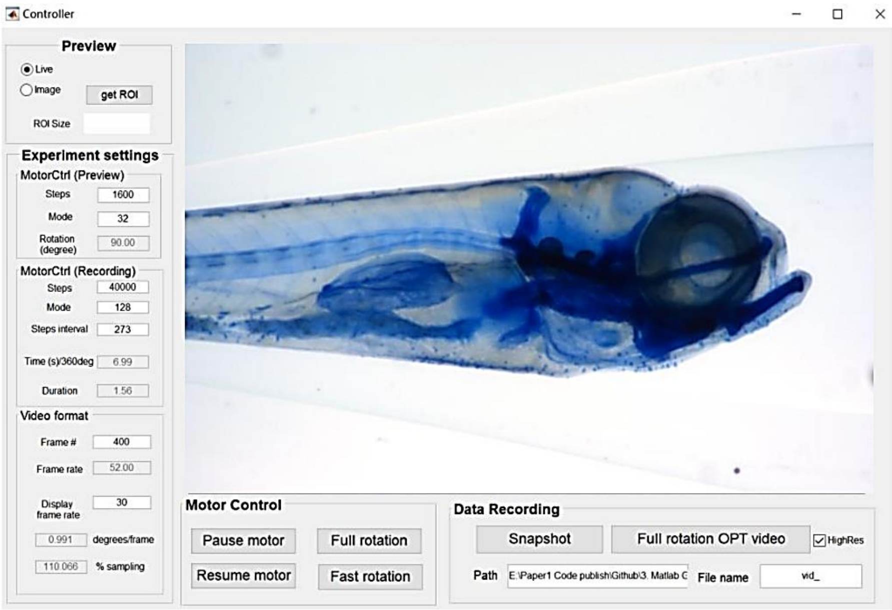

Figure. 3 GUI for OPT system based on MATLAB.

To record OPT data using the GUI, we recommend the following steps:

1. **Start the software** by running GUIOPT.m in MATLAB. Make sure the computer-camera connection as well as the computer-Arduino connection are connected. If the camera connection and detected properly image captured by the camera will appear in the GUI after selecting Live mode under the **Preview** panel. If the motor is connected, rotating the scroll wheel of the mouse and pointing on the image in the GUI will rotate the capillary accordingly.
2. **Check the motor rotation and your sample.** In **Motor Control** panel, clicking "Full rotation" provides a possibility to check the sample rotation for one revolution. The button "Fast rotation" spins the capillary at high speed. One can also control the capillary rotation by using the mouse scroll wheel as described above. In this case, the number of motor steps and mode for microstepping can be set in MotorCtrl (Preview) under **Experiment Settings**. By default the capillary rotates 90 degrees if the mouse wheel is moved one step. Note that the settings here are not controlling the sample rotation during data recording.
3. **Set parameters for OPT data.** The MotorCtrl (Recording) under **Experiment Settings** determines how the sample will rotate in OPT data. One can put in the number of steps, mode for microstepping and time interval in-between steps to achieve desired rotation time and duration. The duration is a relative measure and it is set to 1 for rotation of 360 degrees. The Video format under **Experiment Settings** sets the camera related parameters for OPT data, including the number of frames and video frame rate. In our setup, the frame rate for capturing new frames from the camera is fixed to 52 frames per second. The GUI automatically calculates the degree accuracy as well as the degrees in percentage (360 degrees as 100%) based on the motor and camera parameters.
4. **Record and save file.** Click the "Full rotation OPT video" under **Data recording** panel allows the system to obtain OPT data and save it to a local path by default and the file name can be specified by the user. Note that the preview will be closed during the recording of OPT data and recover after the saving is finished. By ticking the HighRes option, OPT video data will be saved uncompressed. Moreover, the button "Snapshot" saves single preview image. If a user wants to save OPT data within a ROI, click the button "get ROI" under the **Preview** panel. Drag the region in the image in the pop-up window and double-click to confirm the selection. The preview window will display the selected ROI length in horizontal and vertical direction in pixels. The default values for GUI can be changed in file initConfigurations.m.

Additionally, "Pause Motor" and "Resume Motor" enables and disables the motor. This function is optional depending if the stepper motor gets heated over time. The parameters for controlling the stepper motor are related to the settings in the codes for controlling Arduino. Make sure the Arduino codes are successfully uploaded to the Arduino board before running the MATLAB.

Automated reconstruction

We offer three demo programs in MATLAB for OPT reconstruction. The file Demo1_2D_SyntheticData.m runs automatically by creating synthetic 2D objects, transformed them to tomography data for the algorithm to apply reconstruction algorithms. The same reconstruction algorithm is used in the file Demo2_2D_RealOPTData.m. However, in this case, the input data is selected by the user. We recommend testing the algorithm with our zebrafish video data from OPT where the horizontal axis is the rotation axis. A column number is pre-determined in this case and the data sampled from the selected column is used for reconstruction. The full 3D reconstruction of OPT data is conducted using the Demo3_3D_RealOPTData.m file. The user can load a test video, then specify the region-of-interest and the algorithm will apply automatic background detection, finding 360 degrees rotation and apply our gCOR and iRRpw algorithms for center-of-rotation correction and FBP is applied for reconstruction. Note that our algorithm does not require any pre-knowledge of the sample and the environment, such as an image with only the background or feature points for generating reconstruction results. The user can set some parameters in InitOPT.m, for example, set the region-of-interest in the video or set the maximum iteration in the iRRpw algorithm to balance between performance and accuracy.

System Requirement for OPT Reconstruction using MATLAB

For running the GUI and script for center-of-rotation (COR) correction and reconstructing 3D OPT data, we tested our GUI using MATLAB 2018b and operate on Windows 10 system a 3.2 GHz Intel Core i7-8700 processor with 32 GB RAM, 256 GB SSD hard disk. The reconstruction of 855×855×950 voxels data, single-channel and with 3 iterations with our iRRpw algorithm is tested using MATLAB 2018b. It takes approximately 6 min to complete the iterations and we are using NVIDIA GeForce GTX 1060 for GPU acceleration.

Moreover, the GUI for save videos requires the time for extracting frames from the camera buffer to the hard disk to be reasonably short (at a maximum of 60s). The saving can fail if the user is trying to save the data to a slow hard disk or using a portable hard drive with slow USB connection. The reconstruction is dependent on the video frame size, so make sure to select proper size of ROI to match your computing resources.