Time to upgrade: A new OpenSPIM guide to build and operate advanced OpenSPIM configurations

Johannes Girstmair, HongKee Moon, Charlène Brillard, Robert Haase & Pavel Tomancak

Abstract

The initial OpenSPIM platform allowed hundreds of laboratories around the world to generate and process light-sheet data in a cost-effective way due to open-source hardware and software. While setting up a basic OpenSPIM configuration can be achieved expeditiously, correctly operating OpenSPIMs, in particular more complex configurations, e.g., using multiple illumination sides and cameras, can be challenging for routine standard OpenSPIM users.

For this reason, we provide here detailed instructions on how to operate an OpenSPIM with dual-sided illumination and two detection axes (X-OpenSPIM) by implementing an ArduinoUNO microcontroller. Additionally, we introduce µOpenSPIM, a new software plugin for OpenSPIM, with the aim to facilitate multiview time lapse recordings and to give greater flexibility for user specific hardware choice.

Our new software works on any OpenSPIM configuration, comes with drift correction functionality and gives users more options in the way time-lapse movies are initially set up. We also provide an easier, stepwise guide on how to align the laser beam.

With this we empower current OpenSPIM users in various ways and help newcomers striving for using more advanced OpenSPIM systems.

Introduction

Multiview light-sheet imaging has rapidly become an important, often mandatory, technique to study developmental dynamics of living biological samples (Power & Huisken, 2017). Due to its meticulous instructions and various solutions regarding assembly, operation and image processing, the open-source platform OpenSPIM (https://openspim.org/) makes it possible to custom-build affordable and robust light-sheet microscopes capable of acquiring high resolution multiview optical sections of fluorescent samples (Marx, 2016; Pitrone et al., 2013). However, to set up a light-sheet microscope and acquiring multiview datasets remains a complex task (Paysan, 2015). In particular for newcomers every step, from purchasing parts over assembly to operation and finally processing light-sheet data, can pose unexpected challenges, such as incompatible hardware pieces, driver issues, software crashes etc., which quickly accumulate when modifications from the basic OpenSPIM design are made (Girstmair et al., 2016).

The original OpenSPIM design is well-known for its extensibility and options for customization, a key feature as many laboratories seek to adapt it to their unique imaging applications. When the OpenSPIM platform was launched back in 2013, SPIM was already a well-established imaging technique and led to several types of four-lens light-sheet microscopes equipped with more than one camera (Krzic, Gunther, Saunders, Streichan, & Hufnagel, 2012; Schmid et al., 2013; Tomer, Khairy, Amat, & Keller, 2012). Therefore, it was already anticipated that its original SPIM design, which comprises only one illumination lens and only one detection lens, also known as the L-configuration, would be extended into more complex configurations featuring geometries with three or four lenses (Pitrone et al., 2013). A major advantage of such geometries are that illumination and acquisition can happen from different views simultaneously. This is advantageous over setups where the sample is rotated and subsequently imaged from multiple angles, because it reduces phototoxicity or bleaching of fluorophores and reduces acquisition time by the factor of additional cameras used.

An OpenSPIM assembled with a four-lens geometry is referred to as the X-OpenSPIM configuration and is for many users of the OpenSPIM community the obvious next step. An X-OpenSPIM is equipped with two detection objectives and two illumination objectives, which demands aligning two illumination axes, two opposing cameras as well as co-aligning the focal plane of both detection objectives. Additionally, the imaging software has to be configured accordingly to synchronize all devices with each other. This can be improved by using an Arduino board, a single piece of affordable hardware (https://www.arduino.cc/), which can control several hardware devices used in light-sheet systems such as laser shutters, stepper motors, filter wheels and galvanometric mirrors (Gualda et al., 2013).

In summary, building an X-OpenSPIM creates additional complexity and may come with the danger that a newly assembled OpenSPIM system, that significantly deviates from the original design, may become abandoned due to challenges in operation and maintenance that are already present in the early test stages (hardware- or software-wise), a time where the ground still needs to be prepared to generate satisfying acquisition results or any acquisition results at all.

It is the aim of this publication to guide OpenSPIM users to build more advanced OpenSPIM configurations such as the X-OpenSPIM and to facilitate image acquisition, in both, basic and advanced OpenSPIM systems, and to help operating these systems by providing a completely overhauled μManager-based (Stuurman, Amdodaj, & Vale, 2007) OpenSPIM software, which we call μOpenSPIM.

Implementation

Overview of the assembled X-OpensSPIM

We assembled and tested an X-OpenSPIM based on the original OpenSPIM design with improved acquisition speed, a higher magnification, simultaneous multi-view acquisition and more evenly distributed sample illumination. It features two laser lines (VersaLase), two illumination and two detection axes, for which new objective lenses (Nikon, 10 x objectives for illumination and 40 x objectives for detection), and appropriate tube lenses and adapters had to be acquired. The latter also required minor modifications of the infinity space tube. The two cameras (Andor) were used and aligned to each other by installing a corner mirror mount holding an elliptical mirror into one of the detection axes as demonstrated e.g. with the eduSPIM (Jahr, Schmid, Weber, & Huisken, 2016). In order to co-align the focal plane of both detection objectives, we modified one objective holder ring to allow minor manual adjustments along the z-axis within the acquisition chamber.

One of the two cameras was used to trigger the available laser lines as well as the second camera by interposing the ArduinoUNO microcontroller.

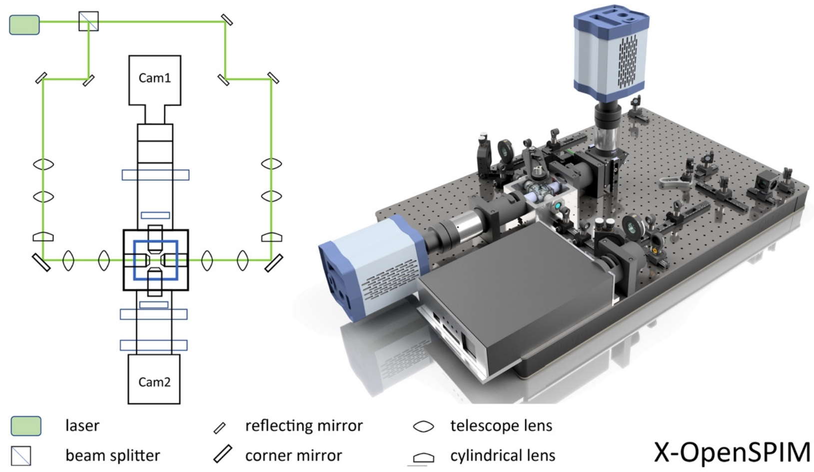


Figure 1- Schematic drawing (left) and solid works rendering (right) of the assembled X-OpenSPIM configuration.

Overview of the new OpenSPIM plugin

OpenSPIM systems are typically run by µManager, a ready-to-use open-source image acquisition software that is free of charge and compatible with a wide range of hardware devices, which gives OpenSPIM users a fast entry point and high flexibility.

With µManager's "Multi-dimensional Acquisition tool" any OpenSPIM is functional. However, upon the initial release of OpenSPIM a specific plugin has been developed to provide a place, which would make it easier for OpenSPIM users to set up complex time-lapse recordings with multiple angles (multiview imaging) and featured a function which compensates for samples drifting out of the field of view (Pitrone et al. 2014).

The plugin itself relies on µManager version 1.4 and came with its own specific GUI (see https://openspim.org/images/Stagecontrols.png) and fit the original OpenSPIM hardware with only a single detection axis and one camera. It supported only a single laser line (OBIS 488 nm LS 100 mW, Coherent, Inc., Santa Clara, CA, USA). Because of these limitations, the plugin has become obsolete for users with more complex OpenSPIM systems. Additionally, µManager has gone through significant changes and improvements and is currently available for download as µManager 2.0.0 version.

Here we provide a detailed description how an X-OpenSPIM and other OpenSPIM configurations can be operated using our home-made plugin, µOpenSPIM, which runs on the newly available µManager 2.0.0.

Summary of new plugin features

- A complete overhaul of the GUI has been made including simple graphic visualizations and an improved control over Picard's 4D-stage.
- A user-friendly way of setting up multiview time lapse recordings with several positions and the option to acquire periodic and sporadic intervals with optional breaks during time-lapse recordings.
- A quick save function for user specified acquisitions settings to save time in case an imaging session is interrupted or a similar session will take place at another time.

Commented [RH1]: Is it possible to automatically save the acquisition settings together with the acquired image data? That would allow users to replicate experiements

Commented [JG2R1]: This is a great idea and we will do this.

- ArduinoUNO support for efficient control of several connected hardware devices so that OpenSPIM users can benefit from improved acquisition speed and hardware synchronization.
- An option for on-the-fly maximum intensity projections
- The possibility to take advantage of GPU-accelerated image processing using the CLIJ library.
- a revised drift-correction function with new options that can help with keeping a drifting sample within the field of view during long term image acquisition.

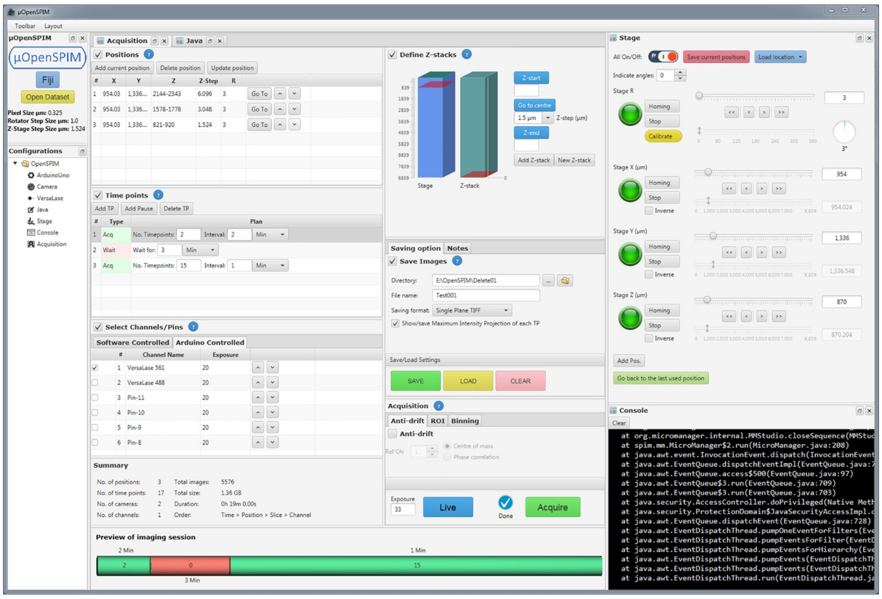


Figure 2 – A screenshot made during before an imaging session using µOpenSPIM operating a fully assembled X-OpenSPIM.

Upgrading the original OpenSPIM design into an X-OpenSPIM configuration

As mentioned above going from the original L- configuration (Pitrone et al. 2014), to an X-OpenSPIM requires doubling all optical elements for the illumination and detection axis and the purchase and manufacturing of additional parts that are described below and summarized in Table 1. Furthermore, all additional parts needed for the upgrade can be accessed under the following link:
https://openspim.org/Table_of_parts_X-OpenSPIM.
A complete overview of the fully assembled X-OpenSPIM is shown in Figure 8.

New illumination/detection objectives and tube lenses

Two illumination objectives and two detection objectives are needed, which should be suitable for the four-lens geometry by providing long enough working distances. This benefits freedom for sample positioning inside the sample chamber. Be aware that new objectives may require new objective holder rings and appropriate tube lenses. E.g., using new Nikon objectives (CFI

Apochromat NIR 40X W) requires a tube lens with f = 200 mm, as well as new tube lens adapters (SM2A20, Thorlabs) and an additional self-made linker part to connect the tube lens adapters (SM2A20, Thorlabs) with the used Olympus U-TV1X-2 Camera Adapters. We provide details of our objective holder rings and linker parts, which can be found and downloaded on the openspim.org website (https://openspim.org/Table_of_parts_X-OpenSPIM).

Acquisition chamber modifications to control the temperature

A new acrylic sample chamber and a new metal chamber holder have to be self-made e.g., by a mechanical workshop, to fit four water dipping objectives. Drafts of such a chamber created with Inventor have been uploaded to the original OpenSPIM website and can be accessed under the following link https://openspim.org/Table_of_parts_X-OpenSPIM.
We would like to point out that similar X-OpenSPIM chambers for Olympus and Nikon objectives can also be purchased from Pieter Fourie (http://www.pfde.co.uk/).

Upgrading the Picard Industries USB-4D-STAGE

New objectives can lead to slightly larger acquisition chamber dimensions. In such a case the original sample arm holder (Picard Industries) has to be replaced with a new custom-made sample arm holder. In our case the new custom-made acquisition chamber required a 2 cm longer sample arm holder, which we additionally equipped with a novel syringe holder, which can be glued above the pulley and allows to quickly immobilize the syringe using a single screw (Figure 3).
Finally, we recommend to replace the standard O-ring belt drive of the rotational stage axis arm with a new tooth-belt drive to achieve a more precise sample rotation (Picard Industries, http://picardindustries.com/).

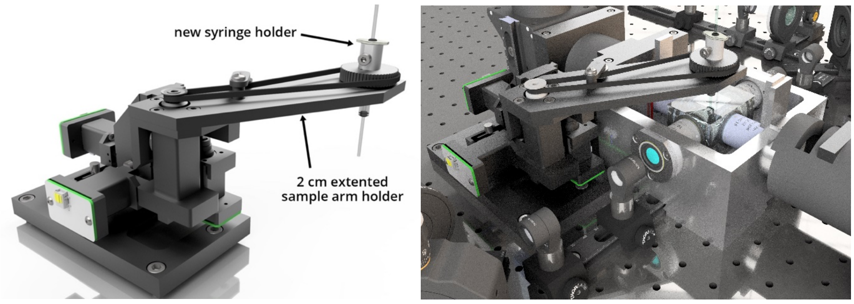


Figure 3 – Modified 4D-USB stage from Picard Industries showing the replaced self-made sample arm holder, which has been extended 2 cm on length and a new syringe holder. Additionally, the stage has been modified with Picard's upgrade kit, which provides with a new, more precise tooth-belt drive.

Corner mirror installation for one of the two detection axes

To correctly align both fields of view from the two Andor sCMOS cameras, a dielectric elliptical mirror (BBE2-E02, Thorlabs) has to be installed into one detection axis. Using a 2'' corner mirror mount (KCB2EC/M, Thorlabs) allows to flip one half of the detection axes, comprising the tube lens and camera adapters by 90 degrees. The weight of the camera has to be sustained e.g., by using cut off ends of optical rails (RLA300/M, Thorlabs) as shown in Figure

4. With this, the field of view of both cameras can be matched mechanically by using the fine adjustments of the corner mirror mount knops.

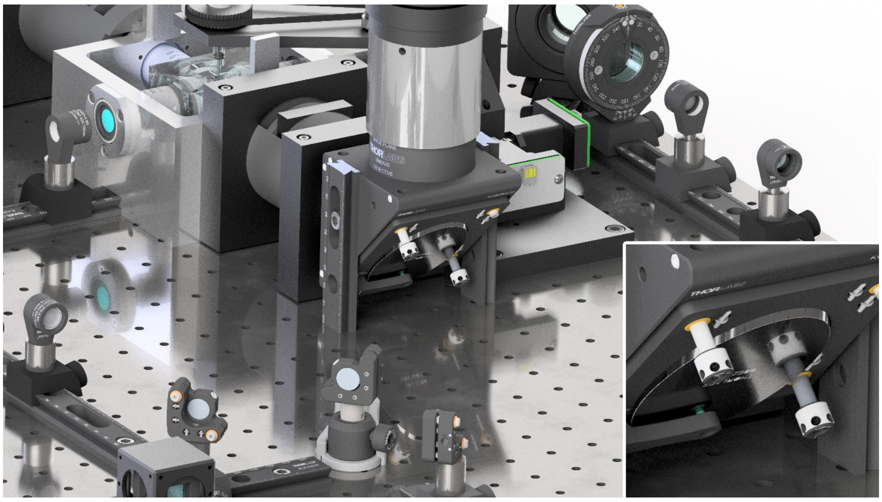


Figure 4 - Corner mirror featured in one of the two detection axes to mechanically align the two camera views by using the two mirror mount knobs (see also inset).

Modifications of one objective holder ring for co-focusing both detection objectives

In order to co-align the focal plane of both detection objectives to the light-sheet, one of the two objective holder rings has to be equipped with a small handle (Figure 5). This provides enough freedom to make minor adjustments along the detection axis in one of the 40x detection objectives and thereby aligning its focal plane with the second detection objective. The moveable detection objective is therefore held in place merely by the O-ring, whereas the other detection objective, to which the light-sheet has to be initially aligned, remains screwed to the acquisition chamber holder.

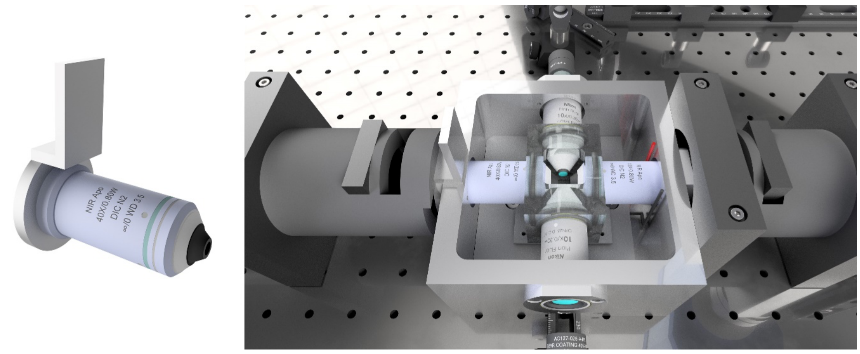


Figure 5 – Modified 40x detection objective holder ring with handle and mounted objective (left) and its place within a fully assembled X-OpenSPIM acquisition chamber (right.

Other deviations from the original OpenSPIM design
It might be worth to abrade the emission filter holder top by about 1 mm to better fit the filter holder into the infinity space slits.
We also recommend to replace the lenses of the first telescopic system, which in the original OpenSPIM design comprises two lenses of 25 mm and 50 mm focal length, by two lenses of 19 mm and 75 mm focal lengths respectively, resulting in a slightly thinner light-sheet. Another minor change concerns the orientation of the lenses. The first telescopic lens (with 19 mm of focal length) should face the arriving beam, while the second telescopic lens (with 75 mm of focal length) faces the direction of the beam as shown in Figure 6.

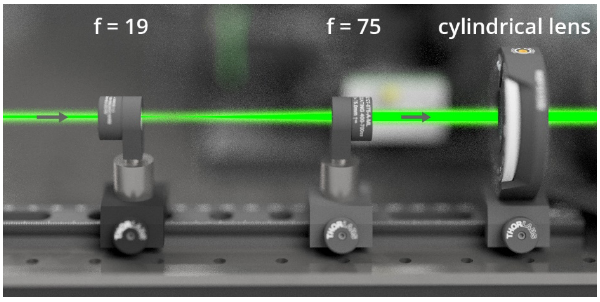


Figure 6 – Rendering of the two telescopic lenses with 19 mm and 75 mm focal length. Arrows depict the direction of the laser beam.

We replaced all Ø1" Mirrors by Ø1/2" Broadband Dielectric Mirrors (BB05-E02, Thorlabs) mounted on a Kinematic Mount (KM05/M, Thorlabs) respectively (see Figure 7). However, this is not an essential modification and there is no reason why other mirrors, including Ø1" mirrors should not be installed. What we do recommend is using a Ø1" Mirror (BB1-E02) mounted on a Gimbal Mirror Mount (GM100/M, Thorlabs) instead of the original corner mirror because it gives good control for the laser alignment step on the sample. Additionally, we placed one of the reflecting mirrors into a Pedestal Post Holder (PH20E/M, Thorlabs), which was clamped to the breadboard by a Clamping Fork, (CF125, Thorlabs) and mounted on an Optical Post (TR20V/M, Thorlabs) (see Figure 7). This gives higher flexibility to adjust the reflecting mirrors while rough aligning the beam along the rails. To split the laser beam for each illumination axis a beam splitter has to be installed (BS004, Thorlabs) and therefore mounted into a cube adapter (BS127CAM), which has to be placed into a cage cube (CCM1-4ER/M, Thorlabs).

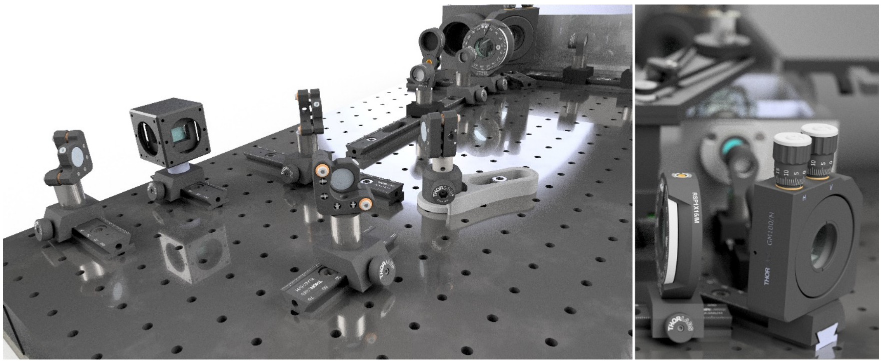


Figure 7 –Detail of the X-OpenSPIM showing the beam splitter with four Ø1/2" Broadband Dielectric Mirrors (left) and the Gimbal Mirror Mount of the corner mirror next to the cylindrical lens (right).

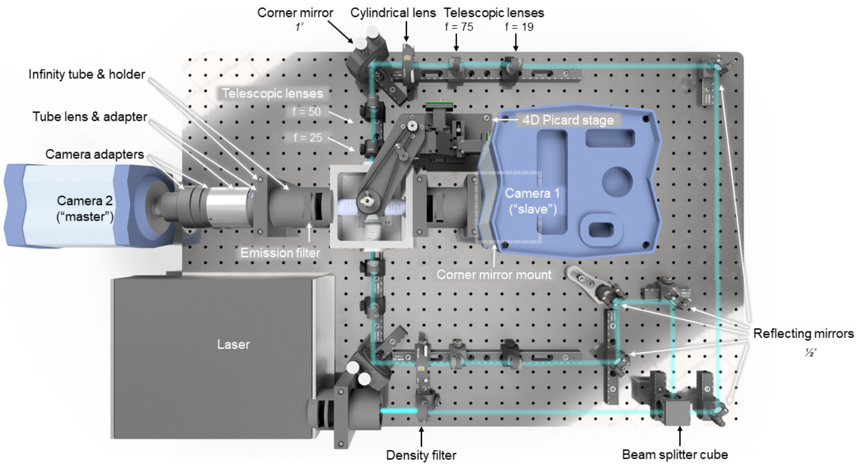


Figure 8 – Rendering of the X-OpenSPIM (top view). The light path of the laser beam is shown in cyan.

Configuring multiple cameras (Andor sCMOS) in µManager

The described X-OpenSPIM has two detection axes with two detection objectives and two Andor sCMOS Neo-5.5 cameras of which the exposure digital output of the designated ("master") camera is wired to the second ("slave") camera as shown in *Figure 24*. In case Andor sCMOS cameras are used, the Andor SDK3 has to be downloaded from Andor and installed into the working µManager directory (see also the µManager website: https://micro-manager.org/wiki/Andor_SDK3).

In case one is not yet familiar with µManager's Hardware Configuration Wizard, its Device Property Browser and how to create Configuration "Groups" and "Presets", we recommend reading through µManager's Configuration Guide:
https://micro-manager.org/wiki/Micro-Manager_Configuration_Guide.
It is also recommended to read through µManager's Multi Camera configuration guide, which covers the same steps in greater detail:
https://micro-manager.org/wiki/Utilities#Multi-Camera.

We depicted all necessary steps on how to configure the two sister cameras in µManager (see supplementary files, Guide 1). Furthermore, the guide can be followed and accessed on our website: https://openspim.org/µOpenSPIM_µManager-configuration, where additional video guides are available.

Setting up an Arduino microcontroller in an X-OpenSPIM

Why using Arduino in an OpenSPIM?

Arduino is an open-source project (https://www.arduino.cc/), and a low-cost solution to build and control all kinds of devices. It uses an Arduino Integrated Development Environment (IDE) to create small programs. Such programs are called sketches and can be uploaded to the Arduino board. In this way it is also possible to gain control over hardware components of microscopes including OpenSPIM systems.
Benefits that typically come along with such hardware-controlled microscopes are improved hardware synchronization and thus faster image acquisition.
Synchronisation between different hardware components in a microscope is often achieved using a TTL trigger-pulse, which may initially be sent by a camera via a BNC connector and to a microcontroller module, which in our case is an Arduino UNO board. In most sCMOS cameras such a trigger-pulse is a common feature and an Arduino board has several output pins, to which other hardware components can be connected (see *Figure 24*).
In practice, up to six devices can be wired to the pins of the Arduino UNO board (Pin-8 to Pin 13) from where they receive the external trigger signal.

Any Arduino board has to be initially programmed to be recognised by µManager. The most widespread Arduino hardware piece is the ArduinoUNO board. It benefits from many available descriptions and known functionality within µManager. Detailed installation guidelines for the Arduino UNO are available at the µManager website:
https://micro-manager.org/wiki/Arduino

To facilitate setting up an Arduino board in µManager, we created a detailed step-by-step guide (see supplementary files, Guide 2), which can also be followed and accessed on our website: https://openspim.org/µOpenSPIM_µManager-configuration, where additional video guides are available.

A summary of the configurations settings and Groups, which we created in µManager together with the ArduinoUNO circuit, which depicts how we wired the various hardware devices together is shown in Figure 9.

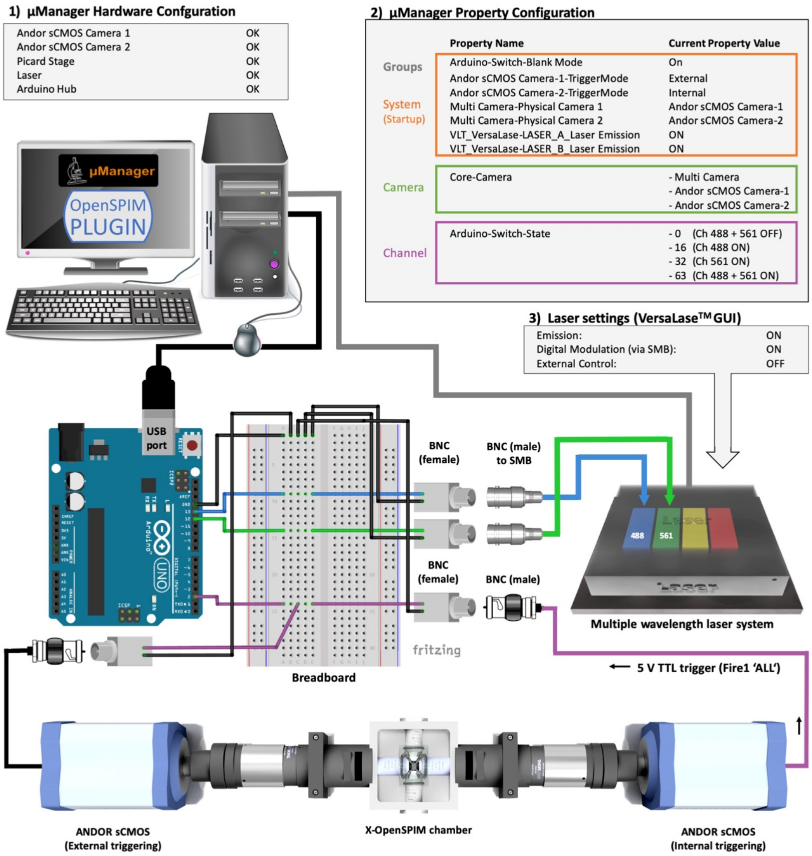


Figure 9 - Arduino circuit using a breadboard to control multi-channel laser triggering while simultaneously imaging with two sCMOS cameras. The Arduino UNO board is located on the top left and connects the Ground (GND) pin, output pins 13 and 12 as well as the input (digital) trigger pin 2 with an Arduino breadboard. The breadboard connects accordingly to the twin cameras via BNC connectors and to the two lasers (488, 561) via a BNC connector and a BNC-SMB adapter. The twin cameras and the OpenSPIM acquisition chamber are shown schematically below. Camera 2 (on the right) is set as 'master' and sends the TTL trigger output signal to the Arduino board as well as to Camera 1 ('slave').

Laser alignment of an X-OpenSPIM system

It takes time to get familiar with correctly aligning an OpenSPIM system and there is always a possibility to improve. With two illumination sides and two detection objectives this task becomes particularly challenging. Therefore, a simple, reproducible way is needed.

We provide a complete alignment guide from aligning the beam along the rails over tuning it within the field of view to the final alignment adjustments on the sample. It is accessible in our supplementary files (Guide 3).

Implementation of µOpenSPIM

We developed our own GUI for imaging called µOpenSPIM and comes with several features that we present here for the first time. It is a µManager Version 2.0 gamma plugin written in Java and in principle works with all three OpenSPIM configurations. Therefore, an X-OpenSPIM with two cameras and an ArduinoUNO microcontroller as described above is not a requisite.

Requirements for µOpenSPIM:

All hardware components of an OpenSPIM system (Laser, Camera, Stage, etc.) have to be pre-configured with µManager using Version 2.0 gamma on a Windows 10 computer. For this µManager's hardware configuration wizard has to be used to create a working configuration (.cfg) file containing all hardware devices. This configuration file will be read by µManager upon the startup of µOpenSPIM.

In case one is not yet familiar with µManager's Hardware Configuration Wizard, we advise reading through µManager's Configuration Guide first:
https://micro-manager.org/wiki/Micro-Manager_Configuration_Guide.

We provide a video depicting how a working configuration file was created from scratch for the described X-OpenSPIM using µManager's Hardware Configuration Wizard. The video can be accessed on our website:
https://micro-manager.org/wiki/Micro-Manager_Configuration_Guide.

Installation and start-up of µOpenSPIM:

Visit https://openspim.org/µOpenSPIM for installation files and detailed installation guides.

Graphical user interface of µOpenSPIM:

Our µOpenSPIM consists of several configurable tabs (see Figure 10), of which five will be introduced in more detail: a) "ArduinoUNO", b) "Camera", c) "Lasers (VersaLase)", d) "Stage" and e) "Acquisition".

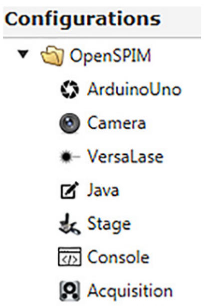
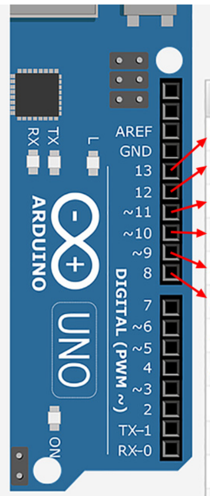


Figure 10 - Configurable taps found in µOpenSPIM

a) ArduinoUNO

With an ArduinoUNO microcontroller correctly configured as described in our supplementary files (Guide 2), the output pins of individual hardware components can be named within the "ArduinoUNO" tab of µOpenSPIM. In there, the device name is entered right next to the corresponding output pin of the Arduino board and corresponds to the triggered device. Once

specified, these names will show up within the “Channels/Pins” area of the “Acquisition” panel when the “Arduino Controlled” option is selected (see Figure 15).
In case of our X-OpenSPIM system we have wired the Arduino output Pin-13 to the digital beam modulation input connector of a 488 laser and in the same way Pin-12 to a 561 laser (depicted in Figure 9). Accordingly, we renamed Pin-12 and Pin-13 into ‘Laser 488’ and ‘Laser 561’ respectively (see Figure 11). They can now be considered as Channels. The description of the hardware component is entirely up to the user.
Any hardware component which is indented to be under the control of the ArduinoUNO board, should be named here prior to acquiring images.



Pin	Switch-State	Hardware component
13	32	VersaLase 561
12	16	VersaLase 488
11	8	Pin-11
10	4	Pin-10
9	2	Pin-9
8	1	BF
ALL_OFF	0	Pin-ALL_OFF
ALL_ON	63	Pin-ALL_ON

Figure 11 - Arduino settings that can be specified in the Configuration panel of µOpenSPIM. Output pin 13 has been physically connected to a laser firing a 561 nm beam and corresponds to the Arduino Switch-State 32. Output pin 12 has the Switch-State 16 and is connected to a 488 nm laser. Switch-State 0 turns all output pins (13-8) off while Switch-State 63 turns all of them on.

b) Camera

The camera configuration can be used to add a grid line to a camera’s field of view. This can be useful for example when the height of two laser beams in an OpenSPIM system with dual-sided illumination need to be centred. Additionally, in OpenSPIM systems with two detection axes images of each camera view can be mirrored, rotated and moved in x and y to test what changes to the camera might be needed in order to align the two field of views with each other. These changes will have no effect for the acquired images. Instead, it would be possible to use µManager’s “Image Flipper function to flip the camera 180 degrees or mirror one view. It is best to set up the two cameras in an already attuned fashion to avoid unnecessary image flipping.

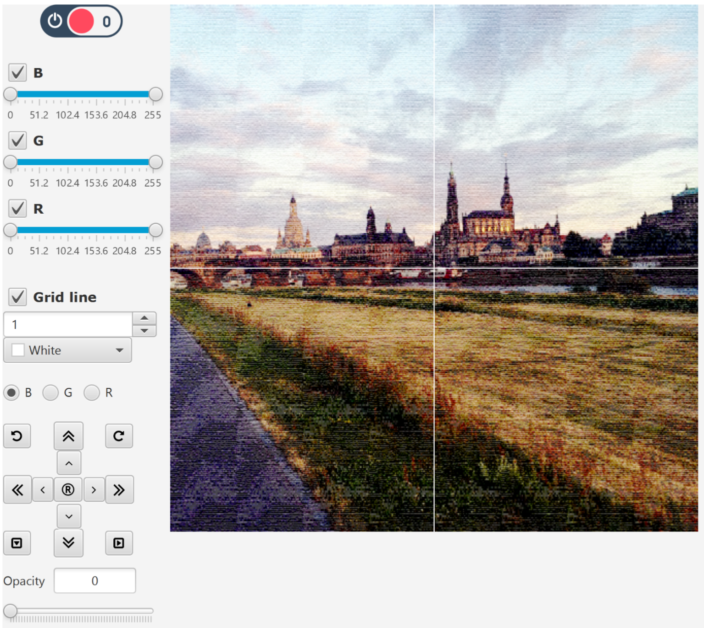


Figure 12 - The Camera configuration panel

c) Laser (VersaLase)

Here the laser power can be changed for various laser lines e.g., Coherent OBIS Laser, Vortran Stradus VersaLase and Cobolt Lasers (Hübner Photonics).

Note that many settings made in the GUI of µOpenSPIM (e.g., laser power, exposure time, etc.) can still be made with µManager's GUI. We recommend not to change the same settings in both interfaces but rather stick to one interface.

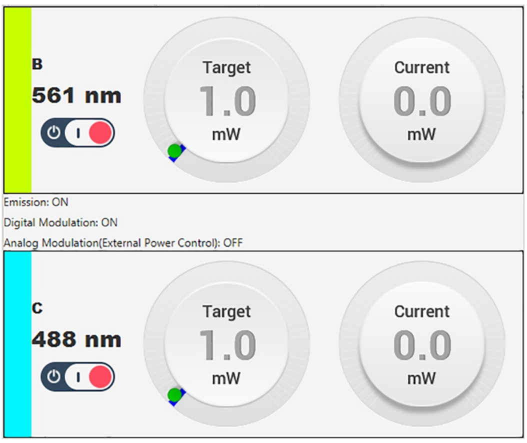


Figure 13 - VersaLase Power Settings for the 488 and 561 laser line.

d) Stage control

In case the Picard stage is connected, all stepper motors (X, Y, Z, R) will work straight away in µOpenSPIM's "Stage" tab. An example of the R stage control is shown in Figure 14.
Initially, it is important to specify a home position by setting all stepper motor locations to 0 using Picard's USB 4D Software (http://picardindustries.com/other/software-downloads/) and then finding a suitable default position from which the sample can be moved safely into the focal plane of the detection objectives. It is a good habit going back to the home position every time before a new imaging session is started.
Additionally, we recommend testing whether the rotational resolution from 0 to 360 degrees corresponds to a full revolution of the sample holder. This can vary in case the standard O-ring belt has been upgraded to Picard's tooth-belt drive or modified in any other way. Deviations from a full revolution can be adjusted by setting the current position of the rotational stage to "0" and then pressing the "Calibrate" Button as indicated in Figure 14. Afterwards, press the "Homing" button and use the arrows or double arrows until a full revolution of the sample holder wheel or sample syringe has been reached. No make a note of the new step size number and click "OK".
To keep the correct rotational step size value every time the Plugin is started, we recommend to modify the current configuration file using µManager's "Hardware Configuration Wizard". In there the Picard Twister can be selected and its step size value changed to the new, correct value and saved.

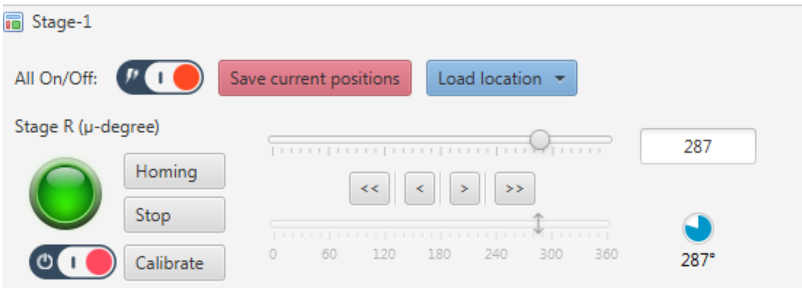


Figure 14 - Rotational control panel for Picard's 4D-USB stage

e) Acquisition

In the "Acquisition" panel (Figure 15) users can acquire single images, stacks and set up long-term image acquisition sessions. We recommend visiting our website for more detailed descriptions and features found within this panel: https://openspim.org/µOpenSPIM
It comprises (A) the "Positions" panel to save multiple sample positions or angles, (B) the "Define Z-stacks" panel where users define the beginning end the end of a new stack, (C) the "Time points"-panel where the amount of time-points and desired intervals/breaks can be added and adjusted, (D) the "Select Channels/Pins" panel where users select different channel settings, which can, but don't have to be under the control of the ArduinoUNO microcontroller, (E) the "Summary" panel, which provides an overview of the imaging session (e.g., images per stack, total number of images, total data size etc.), (F) the "Save Images" panel where

different saving formats can be specified as well as an option enabled to create Maximum Intensity Projections while imaging, (G) the "Save/Load Settings panel, where user specified acquisitions settings can be quick saved and quick loaded, (H) the "Acquisition" panel, where users can start image acquisition and which provides the possibility to specify a region of interest (ROI) and activate one of the two drift-correction options.
An additional "Preview of imaging session" panel (I), gives a schematic visualization of the current time-lapse session, which has been set up by the user and also indicates overall acquisition progress during imaging.

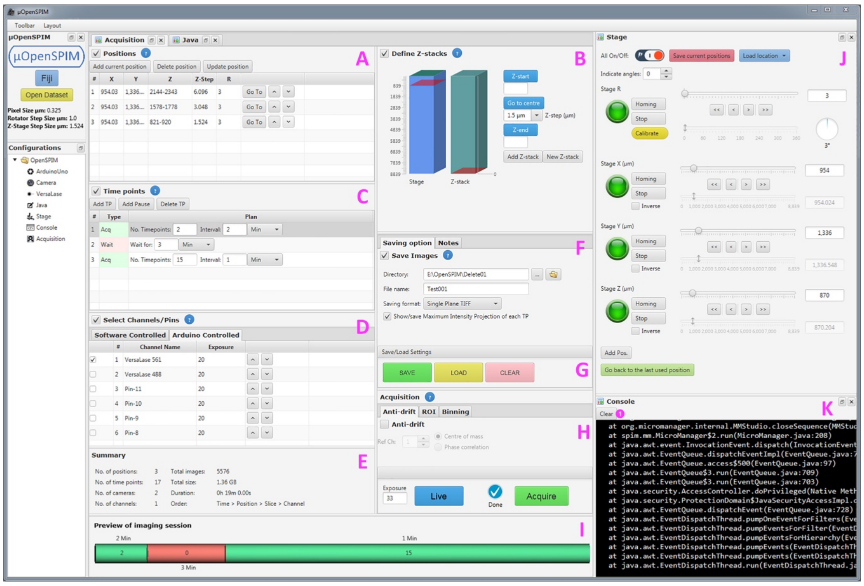


Figure 15 – The acquisition panel allows users to add positions/angles, add time-points, define the z-stack, select created channels and their exposure times and define the saving folder. It can also be used to enable drift-correction function and to save Maximum Intensity Projections of acquired stacks on the fly.

The "Acquisition" panel appearance can be customized and saved as is shown in detail on the OpenSPIM website (https://openspim.orgµOpenSPIM_GUI). As can be seen in Figure 15, it is possible to add e.g., the "Stage" controls (J) for Picard's USB 4D-Stage to the "Acquisition" panel.

- **Defining imaging stacks**

With a working 4D-Stage a 3-dimensional stack can be defined using the blue "Z-start" and" Z-end" buttons (Figure 16). Additionally, the Z-step size is selectable. Its value is predefined according to the positional stage resolution specified in the hardware configuration. E.g., the minimum step size of a Picard 4D-Stage is 1.524 µm.
After "Z-Start", "Z-End" and "Z-step (µm)" size has been defined a new position can be added to the "Positions/Angles" window by pressing the "Add Z-stack." button. This also takes the current rotational position into account. In this way views or positions can be added to the list.

Within the “Position”-panel unwanted position settings can be edited and whole positions either deleted or dragged into different orders using the arrow buttons.

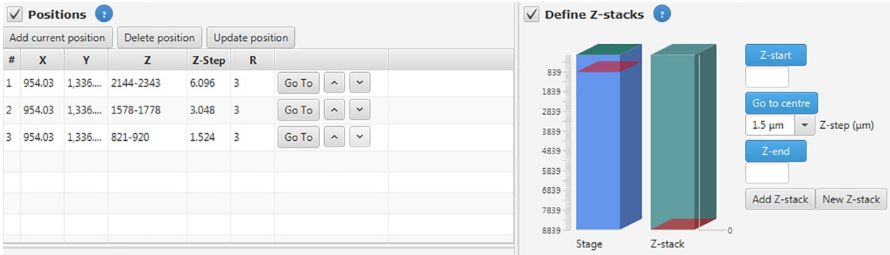


Figure 16 – Area of the acquisition panel where the z-stack is defined and can be added to the Positions/Angles list.

- **Setting up time points (using periodic or sporadic intervals)**

On the top left corner users can set up the desired number of time points with either periodic or sporadic intervals. It is also possible to add breaks to skip e.g., unwanted time-points during acquisition. This can be useful when the precise timing of developmental events of interest is known or to reduce vast amounts of SPIM data after key events have already taken place. An example of how an acquisition session can be set up is shown in Figure 17.

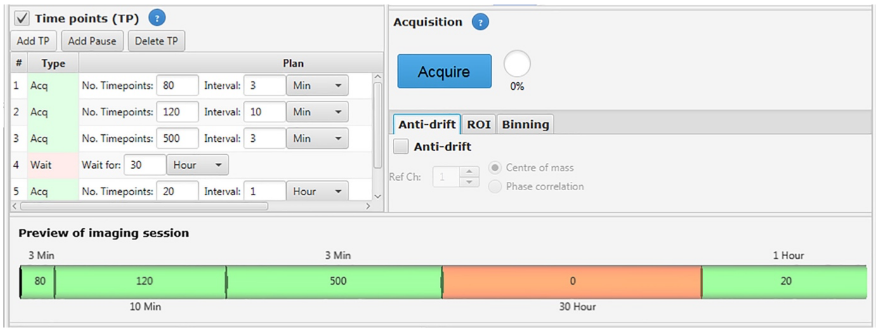


Figure 17 – An example of a time lapse recording using various time-points of different intervals and breaks.

- **Adding Channels**

The Channels area is used to set up multiple laser lines or other devices wired to the ArduinoUNO board. But images can also be acquired without Arduino by using the “Software Controlled” “Channels/Pins” option.

In case an ArduinoUNO board controls one or several devices of the OpenSPIM, one or multiple devices can be selected using the check boxes (red arrow on the left). The default name of the Arduino pins (Pin-9 to Pin-13) can be changed under the “ArduinoUNO” section of the Configuration Panel as described in the Arduino configuration section of the supplementary files (Guide 2). In case the OpenSPIM has no Arduino board equipped laser channels can be added under the “Software Controlled” tab by clicking the “Add a channel” button (red arrow on the right).

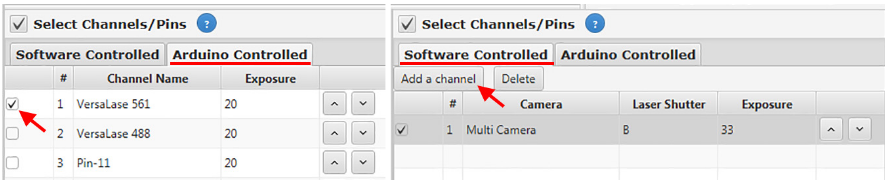


Figure 18 – The "Select Channels/Pins" panel, where channels can be added/selected and acquisition be switched between Software-controlled and Arduino-Controlled.

- **Saving/Acquisition**

Images can be saved in different formats: 1) as single plane tiff files, 2) as ometiff image stacks and 3) as a relatively new N5 format (see https://github.com/saalfeldlab/n5), of which the latter format can be directly opened with Fiji's BigDataViever (Pietzsch, Saalfeld, Preibisch, & Tomancak, 2015). It is also possible and recommended for time-lapse recordings to save Maximum Intensity Projections of each TP/stack on the fly to get a better idea of how well the recording advances.
Other options are: enabling drift-corrections, specifying a region of interest (ROI) and specifying the desired camera binning values.
When everything is set and ready to go, user have the possibility to save all of their previously specified settings using the "Save/Load Settings panel" before starting image acquisition by clicking the "Acquire" button.

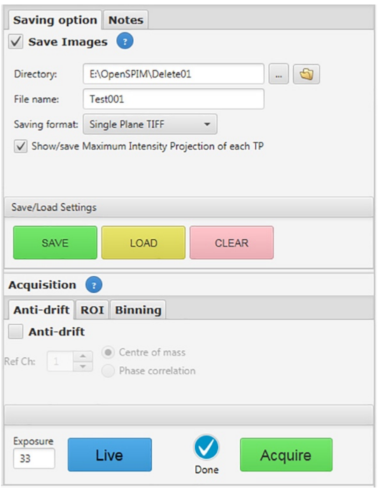


Figure 19 – A screenshot of µOpenSPIM showing various saving options and where to start the acquisition.

Demonstration of the described X-OpenSPIM system

We tested our four-lens X-OpenSPIM, on embryos from the flour beetle *Tribolium castaneum.* Because our OpenSPIM is equipped with 40x detection objectives, these embryos are too

large to fit into a single field of view. To acquire the entirety of the embryo we set up two positions, position 1 (top) and position 2 (bottom), and imaged 3d-volumes with a z-step size of 3 μm using both Andor cameras simultaneously.
Figure 20 and Movie 1 show the development of the beetle as maximum intensity projections (MIP). Both views (Camera 1 and Camera 2) were captured simultaneously every 6 minutes for more than 30 hours. The dataset was neither fused nor processed in any way except for background subtraction and bleach correction using Fiji and demonstrated that the X-OpenSPIM is capable of producing 4d-volumes of high quality over long acquisition periods. It becomes also clear how much information is gained from the second view, which is simultaneously captured without the need of rotating the sample.

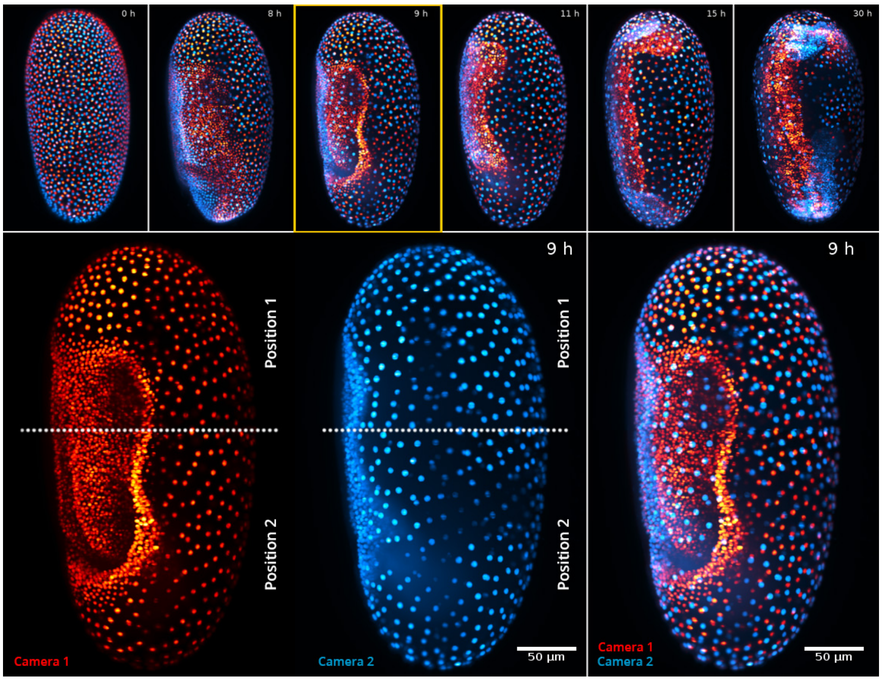


Figure 20 - Montage of a multiview time-lapse recording showing several developmental stages of the flour beetle *Tribolium castaneum*. Fluorescent signal comes from a ubiquitously expressed, nuclear-localized GFP reporter (EFA-nGFP transgenic line). In a single time-point selected at 9 h of continuous live-imaging the two views are also depicted separately in red (Camera 1) and blue (Camera 2). The dotted line indicates where the two positions slightly overlap. *Scalebar = 50 μm*

Conclusion

OpenSPIM is a widely used open-hardware platform to build light-sheet microscopes. But to get to more advanced OpenSPIM systems, both hardware and software modifications are necessary. To simplify and accelerate this process we provided detailed descriptions of how to get from a basic OpenSPIM L-configuration to a more advanced multi-laser controlled and hardware synchronized X-configuration system, which is capable of acquiring data from two views simultaneously.

We depict a) necessary hardware modifications e.g., to co-align the focal plane of the two detection objectives, to co-align both cameras and other minor modifications, b) provide the appropriate software configuration changes that have to be made within µManager c) present a new plugin, which we developed for OpenSPIM users with the aim to operate basic as well as more advanced systems such as the one described here and d) created three guides that depict step-by-step how to align an X-OpenSPIM and how to configure µManager and implement new hardware, which is recommended for more advanced OpenSPIM systems.
The user-friendly GUI of µOpenSPIM makes it easy to set up complex multiview time lapse recordings and supports an ArduinoUNO microcontroller which is a worthwhile enhancement for OpenSPIM systems in general.
OpenSPIM users and the OpenpSPIM community will benefit from several features our new plugin provides. We encourage them to build more advanced OpenSPIM configurations in case their biological applications will require or benefit from it. All our modifications are open access and explained in detail on the OpenSPIM website and we welcome contributions by the community.

Sample preparation
Tribolium castaneum embryos used for live-imaging came from the EFA-nGFP transgenic line (Sarrazin, Peel, & Averof, 2012), where a nuclear-localized GFP reporter is ubiquitously expressed in the animals.
Live embryos were briefly checked for fluorescent signal, then incubated in 1 % low melting agarose and immediately sucked into glass capillaries. The capillary was mounted into the X-OpenSPIM chamber using a 1 ml BD Plastikpak syringe (REF 300013).

1% of low melting agarose containing beads (1:4000 XC Estapor Fluorescent Beads) was used for visualizing the laser beam and the alignment of the light-sheet.

Abbreviations
GUI: graphical user interface; µManager: Micro-Manager

Acknowledgements
We would like to thank Daniela Vorkel for providing fresh *T. castaneum* embryos, Nicola Maghelli for his help with the tube lens modifications and answering several questions during the process of building the X-OpenSPIM. Furthermore, we thank the Light Microscopy Facility (LMF) of the MPI-CBG for providing a suitable workstation computer and guidance, in particular Sebastian Bundschuh for his support with measuring laser intensity values.
We are very grateful to the Mechanic Workshop of the MPI-CBG for their excellent work of all OpenSPIM self-made parts and continuous support.

Authors' contributions
G.J. and T.P. designed the project and wrote the manuscript. G.J. built and tested the X-OpenSPIM, acquired & processed imaging data, troubleshoot µOpenSPIM and prepared the figures for the manuscript. M.H. created, programmed and tested µOpenSPIM. B. C. modified the acquisition chamber and sample arm holder of Picard's 4D-USB stage and provided renderings of the X-OpenSPIM. H.R. modified µManager to make it suitable for GPU-accelerated image processing (CLIJ) and read and improved the manuscript.
…

Funding
…

Availability of data and materials
MicroManager-gamma nightly builds (date) is available at: https://micro-manager.org.
µOpenSPIM is available at: https://openspim.org/µOpenSPIM/

References

Girstmair, J., Zakrzewski, A., Lapraz, F., Handberg-thorsager, M., Tomancak, P., Pitrone, P. G., … Telford, M. J. (2016). Light-sheet microscopy for everyone? Experience of building an OpenSPIM to study flatworm development. *BMC Developmental Biology*, *16*(1), 1–16. https://doi.org/10.1186/s12861-016-0122-0

Gualda, E. J., Vale, T., Almada, P., Feijó, J. A., Martins, G. G., & Moreno, N. (2013). OpenSpinMicroscopy: An open-source integrated microscopy platform. *Nature Methods*, *10*(7), 599–600. https://doi.org/10.1038/nmeth.2508

Jahr, W., Schmid, B., Weber, M., & Huisken, J. (2016). eduSPIM: Light sheet microscopy in the museum. *PLoS ONE*, *11*(8), 1–12. https://doi.org/10.1371/journal.pone.0161402

Krzic, U., Gunther, S., Saunders, T. E., Streichan, S. J., & Hufnagel, L. (2012). Multiview light-sheet microscope for rapid in toto imaging. *Nature Methods*, *9*(7), 730–733. https://doi.org/10.1038/nmeth.2064

Marx, V. (2016). Microscopy: OpenSPIM 2.0. *Nature Methods*, *13*(12), 979–982. https://doi.org/10.1038/nmeth.4070

Paysan, J. (2015). OpenSPIM: Ein Hightech-Commons für Forschung und Lehre. In S. Helfrich, D. Bollier, & Heinrich-Böll-Stiftung (Eds.), *Die Welt der Commons* (pp. 165–170). https://doi.org/10.14361/9783839432457-027

Pietzsch, T., Saalfeld, S., Preibisch, S., & Tomancak, P. (2015). BigDataViewer: visualization and processing for large image data sets. *Nature Methods*, *12*(6), 481–483. https://doi.org/10.1038/nmeth.3392

Pitrone, P. G., Schindelin, J., Stuyvenberg, L., Preibisch, S., Weber, M., Eliceiri, K. W., … Tomancak, P. (2013). OpenSPIM - an open access platform for light sheet microscopy. *Nature Methods*, *10*(7), 598–599.

Power, R. M., & Huisken, J. (2017). A guide to light-sheet fluorescence microscopy for multiscale imaging. *Nature Methods*, *14*(4), 360–373. https://doi.org/10.1038/nmeth.4224

Sarrazin, A. F., Peel, A. D., & Averof, M. (2012). A Segmentation Clock with Two-Segment Periodicity in Insects. *Science*, *336*(6079), 338 LP – 341. https://doi.org/10.1126/science.1218256

Schmid, B., Shah, G., Scherf, N., Weber, M., Thierbach, K., Campos, C. P., … Huisken, J. (2013). High-speed panoramic light-sheet microscopy reveals global endodermal cell dynamics. *Nature Communications*, *4*, 1–10. https://doi.org/10.1038/ncomms3207

Stuurman, N., Amdodaj, N., & Vale, R. (2007). µManager: Open Source Software for Light Microscope Imaging. *Microscopy Today*, *15*(3), 42–43. https://doi.org/10.1017/s1551929500055541

Tomer, R., Khairy, K., Amat, F., & Keller, P. J. (2012). Quantitative high-speed imaging of entire developing embryos with simultaneous multiview light-sheet microscopy. *Nature Methods*, *9*(7), 755–763. https://doi.org/10.1038/nmeth.2062

Supplementary files

Guide 1: Configuring multiple cameras (Andor sCMOS) in µManager

The following steps depict how the two sister cameras of our X-OpenSPIM system are correctly configured. The configuration is facilitated by the fact that both cameras are of the same type.

Step 1: Because both Andor cameras use the same device adapter, they can be added and named through the "Hardware Configuration Wizard" by selecting successively the "AndorSDK3" option within the folder of the same name from the list of available devices. Thereby the camera, which is initially added, should be considered the "slave" camera, whereas the second camera becomes the "master". In case the two cameras are not of the same type, they must at least have the same width, height and pixel type properties.

Step 2: Use the "Hardware Configuration Wizard" to add the "Multi Camera" option, which is also located in the device list and can be found within the "Utilities" folder. Complete and save the "Hardware Configuration Wizard" comprising all other hardware components including lasers, 4D-stage and the ArduinoUNO board.

Step 3: Create a new "Group" within µManager's configuration settings named "System". This step is necessary to specify the physical camera properties. To do so, one has to go into the Group Editor by pressing the Group "Edit" button and tick the Multi Camera-Physical Camera 1 and Multi Camera-Physical Camera 2 from the Property Name list. Furthermore, the "Core-Camera", "Binning" and "TriggerMode" has to be added for both cameras and confirmed with OK.

Step 4: Now is the time to edit all preset values of our newly created "System" group. Press the "Edit" button of the preset panel. In there one can specify the Andor sCMOS Camera-1 as Multi Camera-Physical Camera 1 and Andor sCMOS Camera-2 as Multi Camera-Physical Camera-2. Additionally, the "TriggerMode" of Camera-1 (the 'Slave') has to be set to "External" and of Camera-2 (the 'Master') to "Internal (Recommended for fast acquisition)". Subsequently we recommend to set both Cameras to 2x2 binning. All above mentioned preset values are shown in Figure 21.

Step 5: Furthermore, we advise to follow µManager's "Multi-Camera" recommendations and set the preset name to "Startup" as this will automatically set all Properties within the "System"-Group to any given preset value during µManager's startup.
At this point the configuration settings for the two cameras are completed and can be saved.

Step 6: Finally, the "Multi Camera" of µManager's Core Camera Utility has to be selected in the Device Property Browser before a single multi-channel image (1 channel per camera) can be acquired.
To shorten this step, we added another group by selecting the Utility "Core Camera". This allowed us to quickly switch between Camera 1, Camera 2 and Multi Camera within the "Configuration Settings".

For multi camera imaging we recommend using a decent acquisition computer. E.g., we use a HPZ820 workstation with multiple processors and, notably, found that enabling the non-uniform memory access (NUMA) in the BIOS greatly improves image acquisition stability.

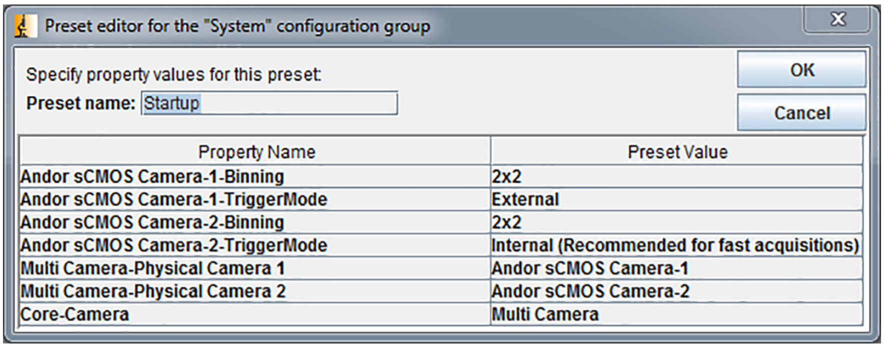


Figure 21 – Multi camera “Startup” preset values of the newly created “System” configuration group in µManager.

Guide 2: Setting up an Arduino microcontroller in an X-OpenSPIM
The following steps depict how an ArduinoUNO board can be configured to enable hardware-controlled triggering of two laser lines (in our case a 488 and 561 laser; see also *Figure 24*).
If one is not yet familiar with µManager’s Hardware Configuration Wizard, its Device Property Browser and how to create Configuration “Groups” and “Presets”, we recommend first reading through µManager’s Configuration Guide:
https://micro-manager.org/wiki/Micro-Manager_Configuration_Guide

Step1: Download and install the open-source Arduino UNO software (IDE:https://www.arduino.cc/en/Main/Software

Step2: USB-connect the Arduino board to your acquisition computer and upload the Arduino UNO firmware source code via a blank sketch window to the board. The firmware source code can be found e.g., on the µManager website under the following link:
https://valelab4.ucsf.edu/svn/micromanager2/trunk/DeviceAdapters/Arduino/AOTFcontroller/AOTFcontroller.ino

Step3: Run µManager’s “Hardware Configuration Wizard” (Toolbar > Devices) and choose to modify your current µManager or create a new configuration file. Then select the “Arduino” folder from the Available Devices list and add the “Arduino-Hub” from within.
Provide the correct “Port value”, which can be looked up in the Device Manager’s Ports list of Windows, where the connected Arduino UNO is listed.
In the “Pre-Initialization Properties” set the “BaudRate” to ‘57600’ and the “Verbrose” to ‘0’.
In the “Peripheral Devices” setup step, select the “Arduino-Switch” and “Arduino-Shutter” and press “OK”.
Make sure that the “Arduino-Hub”, “Arduino-Switch” and “Arduino-Shutter” are listed in µManager’s “Installed Devices”. During the subsequent steps of the “Hardware Configuration Wizard” select the “Arduino Shutter” as the “Default Shutter”.

Step4: After the "Hardware Configuration Wizard" is completed, a new group should be created within µManager's "Configuration Settings", called "System". In case this group already exists, due to the previous multi camera steps, simply select the group and press "Edit". Within the Group Editor select the following features from the Property Name list and add them:

- "Arduino-Switch-Blanking Mode"
- "VLT_VersaLase-LASER_{A-D}_LaserEmission"
- "VLT_VersaLase-LASER_{A-D}_DigitalModulation"
  (in case a VersaLase is used)

Confirm by pressing "OK", and select the newly created "System"-Group. Then specify the "Current Property Value" for all previously selected devices by pressing the Preset-"Edit" button. Set the "Arduino-Switch-Blanking Mode", the "VLT_VersaLase-LASER_{A-D} LaserEmission" as well as the "VLT_VersaLase-LASER_{A-D}_DigitalModulation" to "ON" and press "OK".
As already mentioned previously in the Multi-Camera section, we recommend changing the preset Name to 'Startup'. This will automatically set all properties within the "System"-Group to the given preset values whenever µManager is started.
In case the "System" Group with its "Startup" presets is not created, the "Property Values" have to be set correctly in the "Device Property Browser" every time µManager is started.
As shown in Figure 22, one can combine the Multi Camera property settings and presets for Binning and the camera triggering settings. We advise to create an additional group for each laser line (DigitalPeakPowerSetting), which can be used to control the digital peak power of both laser lines.

Step5 (Optional): For imaging with µManager's "Multi-Dimensional Acquisition" (MDA) we recommend creating another "Group" called e.g., "Channels" where the "Arduino-Switch-State" can be selected to toggle the digital output pattern across Pin-8 to Pin-13.
It is useful to get familiarized with the digital output pattern to better understand how pins 8-13 are switched using single number values from 0-63 as described at the Arduino µManager website (https://micro-manager.org/wiki/Arduino). In our example we will toggle between pin 13 and pin 12 with the Arduino-Switch-State values 16 and 32 respectively. *Figure 24* depicts how Pin-13 and Pin-12 are wired to control the two laser shutters, 488 and 561 respectively.

Step6: Make sure the digital outputs of Pin-13 to Pin-8 of the Arduino board are correctly triggered by the digital exposure signal of the sCMOS "master" camera, which has to be wired to the Pin-2 digital input on the Arduino UNO board as shown in *Figure 24*.
The trigger mechanism of a sCMOS camera is typically based on the digital exposure output of the "master" camera but there are several options how the synchronization of the camera with an Arduino board can be achieved. E.g., there are different FIRE output cables (such as Fire 1, Fire ALL, Fire n, et cetera) for some sCMOS Andor cameras), which can give distinctive trigger signals and depend among others things on the activation of rows on the camera's sensor chip in case the camera uses a rolling shutter instead of a global shutter. We used the output cable labelled "FIRE". In case an Andor camera is used, more information can be found under the following link.:
https://andor.oxinst.com/learning/view/article/synchronizing-to-andor-scmos-cameras

After μManager's Arduino Properties have been configured as described above and the sCMOS camera is wired to the Arduino UNO board and set into a state of exposure, e.g., by going "Live", then the corresponding digital output pins (Pin-8 to Pin-13) will provide an approximate 5V value when measured. Make sure to test the 5V digital outputs prior to connecting them to any device. This can be done e.g., by using a voltmeter, oscilloscope or simply by making LED lights go on.

Step7: The laser itself has to be set into the correct state in order to receive the digital trigger signal. The VersaLase laser system offers several trigger modes (analogue modulation via BNC or RS232, external triggering via BNC and digital modulation via SMB connectivity). One can use the available SMB connectivity, meaning that within μManager's Device Property Browser the VersaLase settings have to be set to emission "ON" and digital modulation "ON". External control has to be set to "OFF", which can be done and checked within the VersaLase GUI software provided by the company.

A full circuit of the described X-OpenSPIM using an ArduinoUNO microcontroller is shown in *Figure 24* and also summarizes all according settings specified in μManager and within the VersaLase GUI software provided by the company.

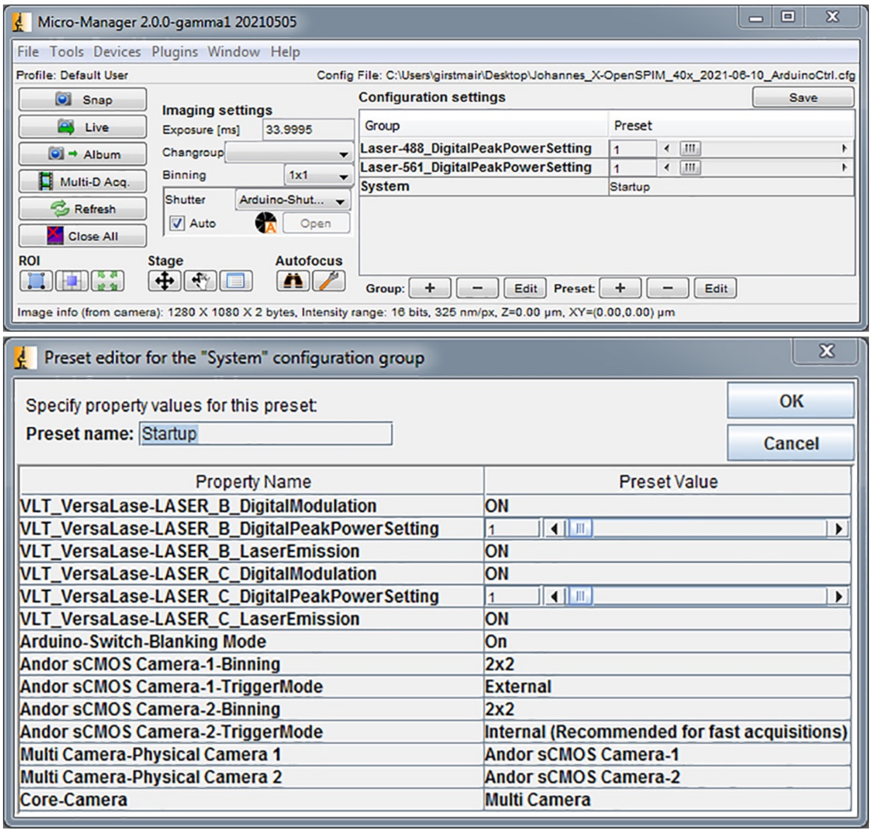


Figure 22 – Setting up the property values for an Arduino controlled OpenSPIM with a multiple laser line module (VersaLase). A new Group named “System” was created (top) and its preset values (named Startup) specified as shown in the list below.

Switching between Arduino and Software controlled laser triggering

Once an Arduino board controls lasers and/or other hardware devices, there might be little reason to go back to a Software controlled mode. In our case this would mean that the laser shutters would be controlled again by µManager instead of the Arduino shutter. But in can be done by creating an extended “System” Group which allows to change property values for the multiple laser module (VersaLase). The VersaLase would then become the “Core-Shutter” instead of the “Arduino-shutter” and it would require switching off the VersaLase “Digital modulation” feature. This requires setting the “LaserEmission” of the laser to “ON” because only then its digital modulation feature can be changed. Without digital modulation the VersaLase power is no longer set by a digital peak power (“DigitalPeakPowerSetting) but by the “Power-Setting” property instead.

Furthermore, in order to enable selecting one of the two laser lines during Software controlled acquisition the VersaLase Shutter was added for each laser as a new group to the configuration settings respectively and in the same way the “PowerSettings” to specify the laser power. Similarly, the DigitalPeakPowerSettings were added, which can be used during Arduino controlled acquisition.

In order to switch between a Software controlled acquisition and an Arduino controlled acquisition (Figure 23, red arrow) the configuration settings and the two “System” group presets (named “Startup” and “SoftwareControlled”) were set in µManager as shown in Figure 23.

Because our VersaLase requires to have laser emission “ON” in order to toggle DigitalModulation ON/OFF, both lasers will fire continuously after switching from Arduino controlled acquisition (Group: System; Preset: Startup) to Software controlled acquisition (Group: System; Preset: SoftwareControlled).

To switch in µOpenSPIM between Arduino controlled and Software controlled requires to described Group/Preset changes within µManager’s configurations settings.

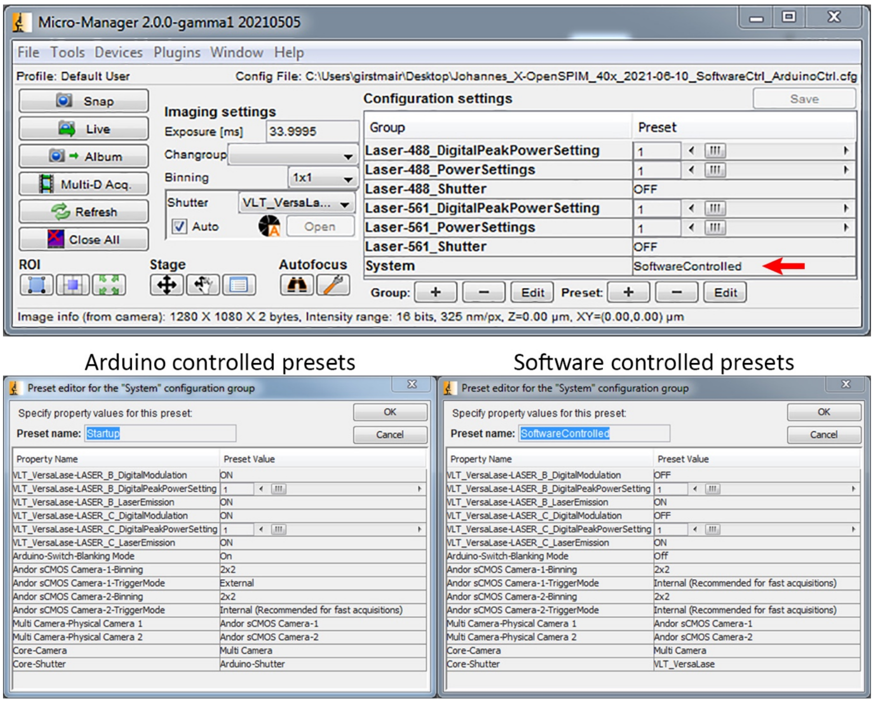


Figure 23 - µManager preset property values for Arduino controlled and Software controlled acquisition using the VersaLase multiple laser module with two laser lines (Laser_B and Laser_C). The red arrow marks where the two group presets can be switched.

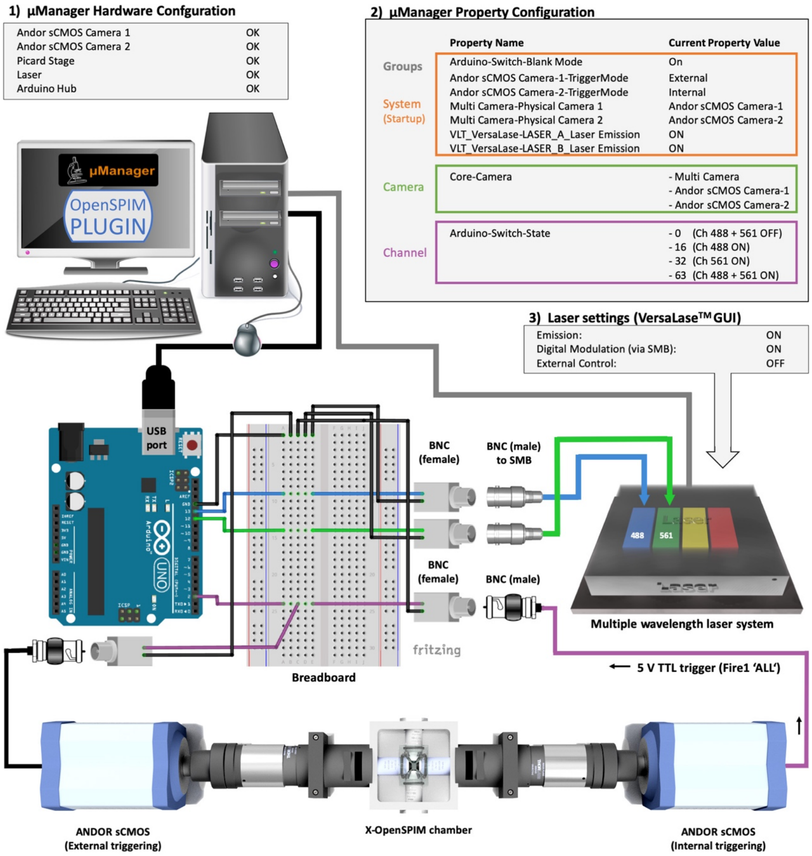


Figure 24 - Arduino circuit using a breadboard to control multi-channel laser triggering while simultaneously imaging with two sCMOS cameras. The Arduino UNO board is located on the top left and connects the Ground (GND) pin, output pins 13 and 12 as well as the input (digital) trigger pin 2 with an Arduino breadboard. The breadboard connects accordingly to the twin cameras via BNC connectors and to the two lasers (488, 561) via a BNC connector and a BNC-SMB adapter. The twin cameras and the OpenSPIM acquisition chamber are shown schematically below. Camera 2 (on the right) is set as 'master' and sends the TTL trigger output signal to the Arduino board as well as to Camera 1 ('slave').

Guide 3: OpenSPIM alignment guide

A) Aligning the laser beam along the rails

Make sure appropriate laser safety measures and training has been carried out and that µManager's "Hardware configuration wizard" has been completed for at least one camera, one laser and the USB-4D stage.

A syringe to mount a glass capillary into the acquisition chamber filled with agarose containing fluorescent beads is needed.

For the alignment along the rails, which carry the optical components of the first and second telescope system, we use the two mirrors mounted on two kinematic mounts and one corner mirror mounted on a gimbal mirror mount (see Figure 8). Note that the corner mirror is a modification of the original OpenSPIM design and is placed on its own short rail piece that was cut off from a 300 mm optical rail (RLA300/M, Thorlabs).
All mirrors are placed at approximately 45 degrees. After the alignment the emitted laser beam will hit roughly the centre of all mirrors and ends at the centre of the illumination objective. To achieve this, two separate reference points are needed at the correct height. In a typical OpenSPIM setup the beam would be elevated 50 mm off the surface of the breadboard. For the reference points one can use e.g., a nearly closed iris apertures (SM1D12D, Thorlabs), alignment disks (DG05-1500-H1-MD, Thorlabs) or alignment plates (LMR1AP) for lens mounts (LMR1/M). We used the latter placed on 1" rail carriers (RC1, Thorlabs) to create both reference points at the correct height.

Step 1: Remove any optical parts from the illumination axis except for the kinematic mirrors and the corner mirror.

Step 2: Add the Iris at the beginning of the first optical rail and the alignment disc at its end. Adjust the kinematic mounts (KM05/M, Thorlabs) of the reflecting mirrors. Start with the first kinematic mirror and aim for the central opening of the nearly closed iris (reference point 1). Then adjust the second kinematic mirror but this time try to hit the central hole of the alignment disc. The laser beam will not hit the alignment disc right away. Instead, it diminishes before reaching the central hole of the alignment disc. Whenever this is the case, go back to the first kinematic mirror and readjust for the Iris aperture hole. Play this back and forward until the strikes both reference points and the first optical rail is correctly aligned.

Step 3: Proceed to the second optical rail between the corner mirror and the illumination lens of the chamber. Again, place the two reference points at the beginning and the end of the rail and align the beam by adjusting the corner mirror with the horizontal and vertical adjuster knobs of the Gimbal mounts. Initially the corner mirror should be brought into its default position using the Gimbal mount knobs and the first rough alignment has to be done by sliding the corner mirror forward or backward along the where it is situated before the mirror can be fixed onto the rail.

Step 4: After adding all lenses of the first and second telescope system (without the cylindrical lens) it may be a good idea to verify if the telescopic lenses did not severely alter the alignment of the laser using two reference points. This can be avoided e.g., by ensuring that the telescopic lenses are tightly assembled and that the lens mounts face the illumination axis without any tilt.

Step 5: Repeat step 1-4 to align a possible second illumination axis.

B) Finding the laser beam in the field of view

Now that the alignment along the rails carrying the optical components is completed, the emitted laser beam can be visualized in agarose after adding the two telescope systems and adjusted by slightly changing the positions of the spherical lens assemblies along the rails.

Step 1: Prepare 1% liquid agarose containing fluorescent beads (e.g. 1:2000), which fit the laser excitation and emission filters of your OpenSPIM system. Sonicate the beads in a 1:100 stock solution for 3 min before mixing them with liquid agarose or vigorously vortex it for several minutes to avoid clustering.

Step 2: Vortex the Agarose containing the beads for 3 min and soak it into a glass capillary e.g. by using a plunger. For detailed descriptions for mounting and preparing samples visit the OpenSPIM website (http://openspim.org).

Step 3: Fill the acquisition chamber with water, mount and find the lower edge of the glass capillary using any available bright light source (even the torch of your phone can be used). Then push out the agarose by gently pressing the plunger until the column of agarose covers the entire field of view. Make sure the glass capillary completely out of view. Now focus on the left or right edge of the Agarose and centre the column of agarose in the field of view.

Step 4: Remove emission filters and cylindrical lenses of the illumination axes that should be aligned and slightly adjust the distance between the 25 and 50 mm telescopic lenses and their distance to the illumination objective. The laser beam should become visible, first as an indistinct broad fuzzy beam crossing the field of view horizontally from left to right.
Further adjust the telescope lenses to increase the sharpness of the beam until it looks similar to the one depicted in *Figure 25*, A.

Step 5: Adjust the horizontal gimbal mount knob of the corner mirror to bring the illumination beam in focus with the working distance of the detection objective up to the point where it can be seen as a very thin stripe instead of a coarse beam (*Figure 25*, B). At this point it often becomes obvious that the focal point of the beam is not centred in the field of view. This shift is shown in *Figure 25*, B. Again, carefully adjust one of the telescopic lenses to correct for this shift.

Step 7: Repeat step 4 to 6 on the second illumination axis until the left and right illumination beams are aligned and resemble the aligned beams depicted in *Figure 25*, C and D.

Step 8: Adjust the vertical gimbal mount adjuster knob to centre the beam. In this way both illumination paths are aligned and centred in the field of view until they overlap each other (*Figure 25*, E).

Step 9: Check if any rotational misalignments between the field of view of both sister cameras are visible. An example of this can be seen in in *Figure 25*, G. If this is the case one camera has to be rotated until the mismatch is corrected.

C) Fine alignment of the light-sheet using beads:

Step 1: By putting the cylindrical lenses and the emission filter back, the alignment of both excitation light-sheets can now be tested on fluorescent beads, which ideally homogeneously cover the field of view and/or on a specimen embedded in agarose (*Figure 25*, I). To achieve best imaging results the horizontal gimbal mount knob has to be slightly adjusted while playing with the rotation mount of the cylindrical lens.

Different light intensities of the beads between the left and right illumination axes, may indicate slight beam alignment differences. This issue can be corrected quickly by looking at the beads. In the illumination axis, where a lower intensity of beads can be observed, the height of the beam can carefully be adjusted by using the two reflecting mirrors positioned prior to the rail carrying the first telescopic system. By making slight adjustments with a 5/64" hex key adjuster, intensities should increase and decrease. Aim to maximise light intensity on both illumination axes.

Step 2: An overlap of the field of view of the two sister cameras can be achieved mechanically by looking at beads and using the fine adjustments of the corner mirror mount knops of one of the detection axes.
First make sure the same beads are visible in the two sister cameras and that the beads are co-focused by moving one of the detection objectives in z. Then use the corner mirror mount knops of the 2-inch corner mirror mount (KCB2EC/M, Thorlabs) to co-align the two field of views until all beads overlap. If a significant stronger mismatch of beads is visible at the outer corners of the field of view (*Figure 25*, H) and the impression of a spiralling feeling occurs while going through the beads in z, then one of the two cameras has to be rotated before the overall match of the beads can be improved (*Figure 25*, I).

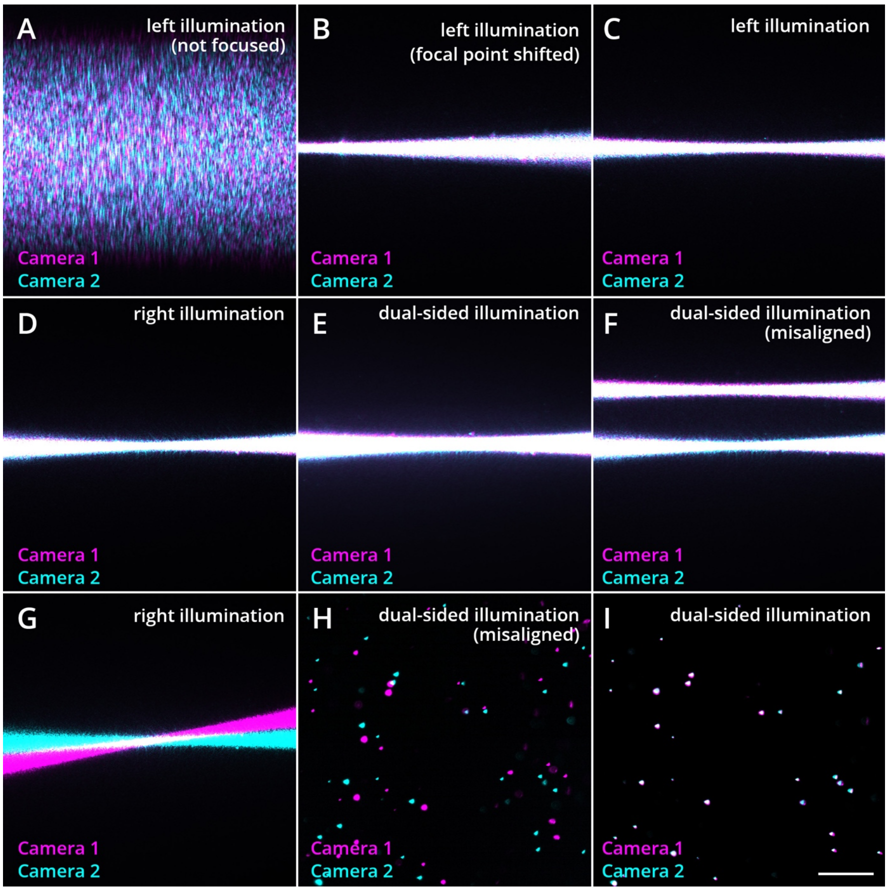


Figure 25 – Aligning the X-OpenSPIM by looking at the two laser beams (left and right illumination) and by looking at beads embedded in Agarose. In all images (A-I) the two detection objectives are already co-focused. (A) Broad beam of the left illumination before adjusting the horizontal gimbal mount knob *of the corner mirror. (B) aligned beam with a slight shift to the left of its focal point. (C-E) The left and right illumination beams are correctly aligned. (F) One of the illumination beams needs to be adjusted with* the vertical gimbal mount adjuster knob to centre *it in the field of view. (G) Camera 1 needs to be rotated to match the field of view of Camera 2. (H) Misaligned beads due to rotational mismatch between the two sister cameras. (I) An aligned light sheet shows beads evenly distributed across the field of view of both cameras. Scalebars are 50 µm.*

Purchased parts	**company**	**Product number**	**amoun t**
Arduino UNO Rev3 SMD	Arduino	A000073	1x
CFI Plan Fluor 10x W/ 0.30/ 3,50 (10x water-dipping objective, N.A. 0.10)	Nikon GmbH	CFI Plan Fluor 10X W	2x
CFI Apochromat 40x W NIR/ 0.80/ 3,50 (40x water-dipping objective, N.A 0.80)	Nikon GmbH	CFI Apochromat NIR 40X W	2x
Detection axis holder, base (https://www.pfde.co.uk/)	PFDE	9	2x
Detection axis holder, top (https://www.pfde.co.uk/)	PFDE	10	2x
Infinity space tube (https://www.pfde.co.uk/)	PFDE	11	2x
USB-4D-STAGE R-Axis Upgrade kit	Picard Industries	N/A	1x
Peltier cooler module 10.5W, 8.5A, 2.1V; ET-017-14-11	RS Components	490-1244	1x
Achromatic Doublet, f=19 mm, Ø1/2", SM05-Threaded Mount, ARC: 400-700 nm	Thorlabs	AC127-019-A-ML	2x
Achromatic Doublet, f=75 mm, Ø1/2", SM05-Threaded Mount, ARC: 400-700 nm	Thorlabs	AC127-075-A-ML	2x
Adapter for Tube Lens	Thorlabs	SM2A20	2x
Alignment Plate for Ø1" Fixed Lens Mounts	Thorlabs	LMR1AP	2x
Beamsplitter Cube 50:50	Thorlabs	BS004	1x
Beamsplitter Cube Adapter for Compact 30 mm Cage Cube	Thorlabs	BS127CAM	1x
Broadband Dielectric Elliptical Mirror	Thorlabs	BBE2-E02	1x
Broadband Dielectric Mirror, Ø1", 400 - 750 nm	Thorlabs	BB1-E02	2x
Broadband Dielectric Mirror, Ø1/2", 400-750 nm	Thorlabs	BB05-E02	5x
Clamping Fork, 1.24" Counterbored Slot, Universal	Thorlabs	CF125	2x
Cylindrical Achromat Lens	Thorlabs	ACY254-050-A	1x
Dovetail Optical Rail, 300 mm, Metric	Thorlabs	RLA300/M	2x
Gimbal Mirror Mount, Ø25.4 mm	Thorlabs	GM100/M	2x
ITL200 Tube lens for Nikon	Thorlabs	ITL200	2x
Kinematic Mount for Ø12.7 mm Optics, Metric	Thorlabs	KM05/M	5x
Lens Mount for Ø1/2" Optics, One Retaining Ring Included, M4 Tap	Thorlabs	LMR05/M	4x
Lens Mount with Retaining Ring for Ø1" Optics, M4 Tap	Thorlabs	LMR1/M	2x
Mirror 30 mm Cage Cube, M4 Tap	Thorlabs	CCM1-4ER/M	1x
Pedestal Post Holder	Thorlabs	PH20E/M	2x
Rotation Mount for Ø1" Optics	Thorlabs	RSP1X15/M	1x
Right-Angle Kinematic Elliptical Mirror Mount	Thorlabs	KCB2EC/M	1x
Vacuum-Compatible Optical Post, Ø12.7 mm, M4 Setscrew, M6 Tap, L = 20 mm	Thorlabs	TR20V/M	2x

Table 1 - Additional parts purchased to upgrade from the original OpenSPIM design into an X- OpenSPIM configuration

Self made parts	File link (OpenSPIM.org)	amount
4D-Stage sample arm holder	https://openspim.org/Table_of_parts	1x
4D-Stage syringe holder	https://openspim.org/Table_of_parts	1x
Acrylic sample chamber	https://openspim.org/Table_of_parts	1x
Metal holder for acrylic sample chamber	https://openspim.org/Table_of_parts	1x
Objective holder ring	https://openspim.org/Table_of_parts	3x
Objective holder ring with handle	https://openspim.org/Table_of_parts	1x
RC1 Ø1/2" lens stilt	https://openspim.org/Table_of_parts	4x

Table 2 - List of additional OpenSPIM parts that were self-made in the workshop

Arduino UNO firmware source code used for µManager:
Link (https://micro-manager.org/wiki/Arduino):
https://valelab4.ucsf.edu/svn/micromanager2/trunk/DeviceAdapters/Arduino/AOTFcontroller/AOTFcontroller.ino