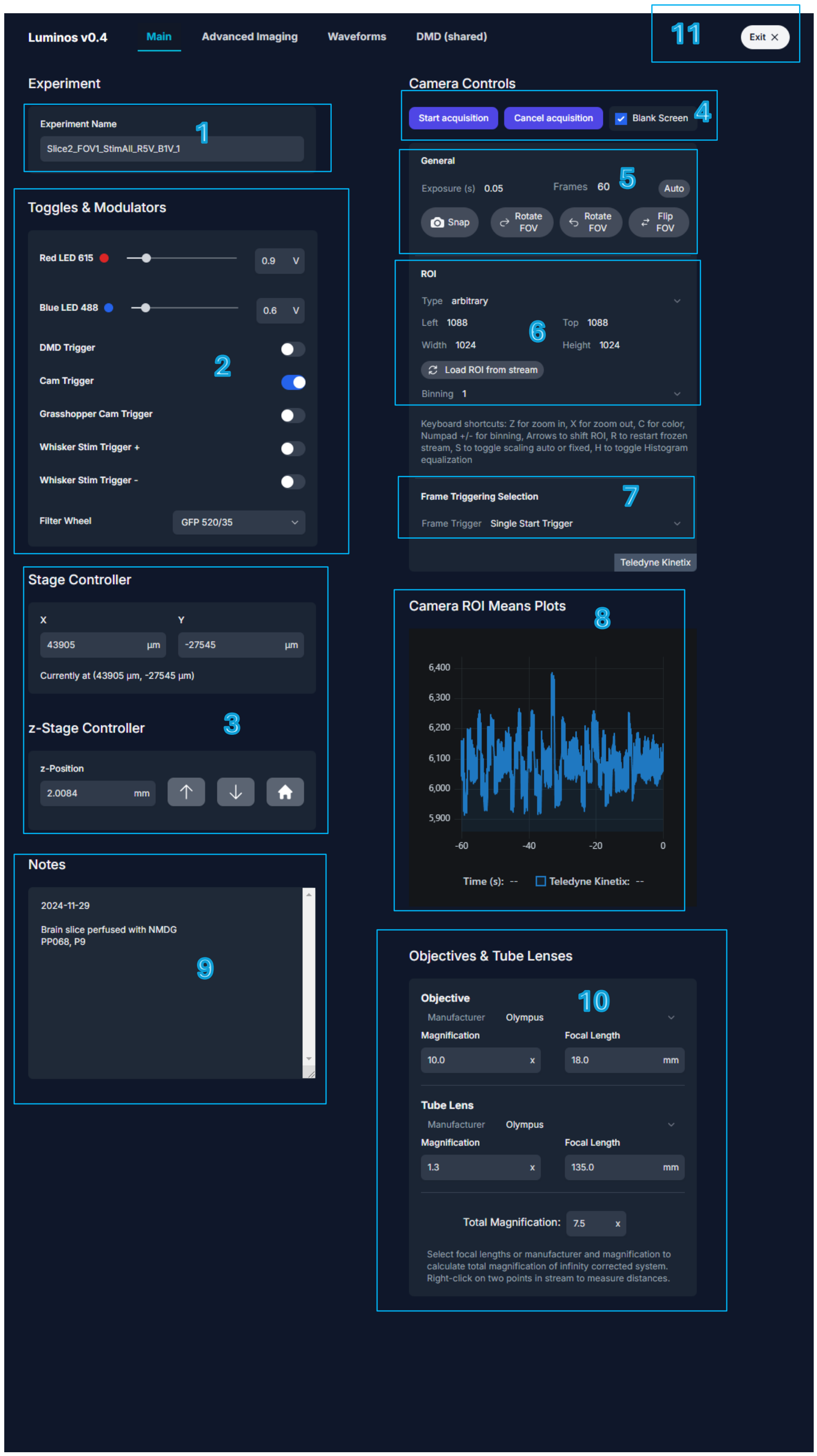
Luminos User Manual v0.4

Table of Contents:

Daniel G. Itkis, 11/2024

Cohen Lab, Harvard University

Main Tab: Single Camera example



1. **Set directory name** to save to. Data will be saved in inside a directory with timestamp and experiment name as a single string, inside of a date directory in the user directory in the dataDirectory (defined in the app initializer .json file):
   \dataDirectory\user_name\date\timestampExperimentname

2. **Manually set NI_DAQ_Modulators and NI_DAQ_Shutters** that are given a name in the initializer .json file. Set analog values using sliders and turn digital outputs on and off using toggles. Set filter wheel positions.

3. **Stage control:** Explicitly set stage positions in the input field. Depending on the connected stages, these can be z-Stages, xy-stages, or xyz-stages. For z-Stage, type in value or jog using up/down buttons. Movement speed switches from fine step to coarse step if button is held down for >2s.

4. **Acquisition buttons:**
   Start Acquisition with parameters set in Main tab and the waveforms set in the Waveform tab. This calls `Waveform_Camera_Sync_Acquisition.m`, which could be modified according to individual needs. Camera stream freezes for the duration of the recording. Once the DAQ has finished running its waveforms, acquisition is finished, cameras are restarted (whether there are outstanding requested frames or not) and "Done with experiment." is printed to the MATLAB terminal.
   Cancel Acquisition currently in progress. Use to abort and reset DAQ and restart cameras if incorrect values were used etc. This can be used for both regular acquisition and to abort running waveforms from the **Run Waveforms** button in waveforms tab.
   Blank Screen toggle whether all monitors should be set to black during acquisition to minimize stray light from monitors. The screen will automatically un-blank itself after the acquisition is finished, or manually by clicking any key during acquisition.

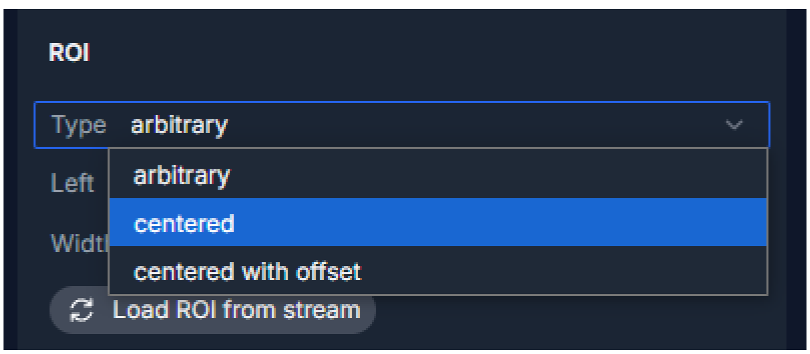
5. **Set exposure time and frame number, take snap, stream options.** Minimum exposure time in s should be selected based on size of selected ROI and camera's line readout rate. Selecting a shorter exposure time will lead to many dropped frames and buggy behavior. Number of requested **Frames** can be input manually or using **Auto** button.
   The **Auto** button will calculate maximum number of frames for the selected **Duration** in the Waveforms tab (Duration/Exposure time). When using **Manual triggering** from Waveform tab, **Auto** will automatically count the number of rising edges to be sent to camera trigger based on the selected waveforms.

If by the end of an acquisition the camera has not collected the requested number of frames either due to dropped frames or due to improperly set triggers or parameters, the camera acquisition is aborted and no further frames are collected (bin file will contain all collected files up to that point). In case the camera does not resume streaming real time data after acquisition, it can be refreshed by clicking inside the camera stream and pressing **'r'**.

The **Snap** button will take a snap and save it to the user's directory/date in the Snaps folder as a .tiff file, alongside a .mat file containing metadata about the snap (binning, ROI, DMD tform during snap, etc.).

Rotate FOV buttons will rotate the camera stream by 90 degrees each time they're clicked, while **Flip FOV** flips stream horizontally. These only act as a convenience for viewing and only affect the displayed real time stream and have no effect on data collection or snaps.

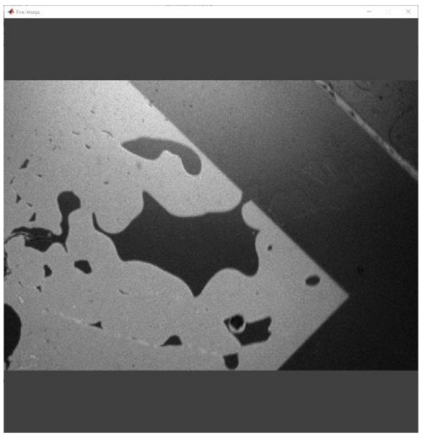
6. **Set ROI.** Select between arbitrary ROI, centered for ROI in the center of the camera (as some camera types always start reading out from the sensor midline), or centered with offset (to conveniently adjust width and height of FOV without modifying top-left corner).



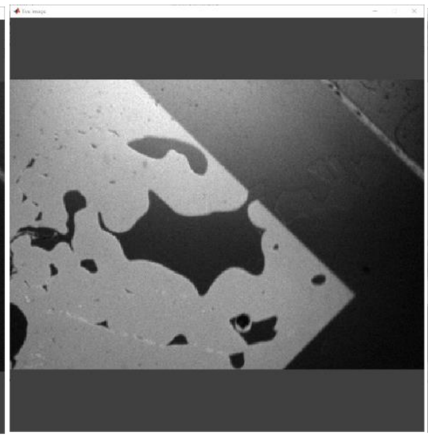
Pixel values are automatically set to multiples of 4 due hardware limitations. ROI can also be changed inside of Camera stream window. Clicking two corner points will select ROI and **z** zooms into the selected ROI. **x** zooms out to the full sensor size and the arrow buttons shift the ROI in small steps in the requested direction. Doing this does not update the ROI in the Main tab. Click **Load ROI from stream** to update and get current ROI.

Binning sums the 1 pixel, 2x2 pixel or 4x4 pixel at a time on the hardware level. This does not require any changes to ROI as ROI is always defined with respect to physical sensor pixel. Binning can also be adjusted inside the camera stream window by pressing the buttons **+** and **-**. The binning parameter is saved as camera.bin in the device data for each acquisition and in snap.bin for each snap .mat file. Nrow and

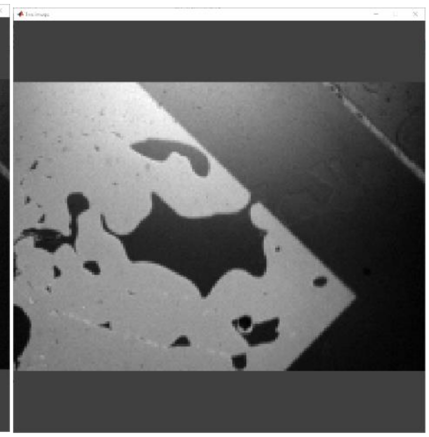
Ncol need to be adjusted by dividing by this parameter when loading using `readBinMov.m`.



Binning: 1



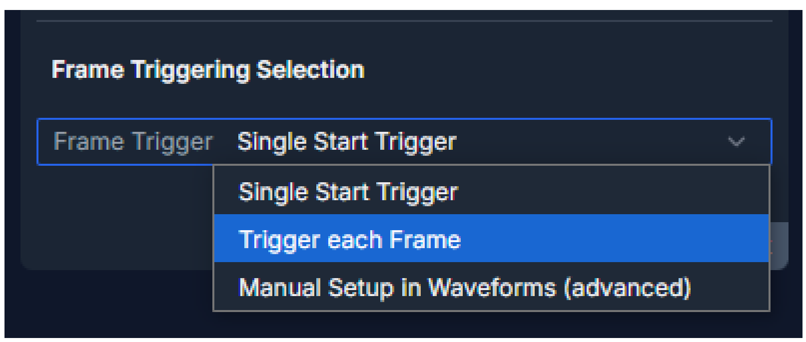
Binning: 2



Binning: 4

7. **Frame trigger selection**

   Cameras can typically be triggered in one of two ways: Using a single trigger to start acquisition (**Single Start Trigger**), after which each frame is timed based on the camera's internal clock, or using waiting for a trigger to start the next frame (other two options). Pulses are always sent to the camera trigger port set in the rig initializer .json.

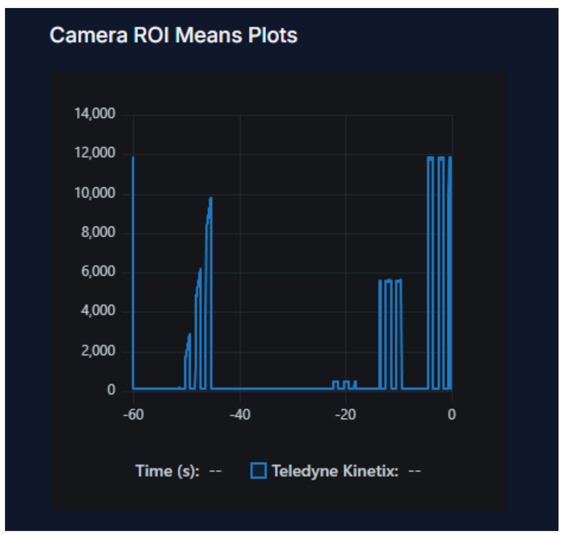


   When triggering each frame with a pulse, two options are available: **Trigger each Frame** requires an additional input **DAQ Trigger Period (ms)** and will set the DAQ to output a train of pulses at the set rate until DAQ acquisition is completed.

   Manual Setup in Waveforms is the most flexible option. When using this mode, camera triggers are not set automatically. Instead, use the Waveforms tab to manually set up an arbitrary waveform for the camera trigger, e.g. for delayed camera acquisition, bursts of frames at specific intervals, etc. This mode is also compatible with external triggering, i.e. if a separate device is used to trigger the camera. No waveform needs to be set in this case, but a warning will be displayed warning that no waveform is set for the camera.

The availability of these modes depends on on what type of port is used to trigger the camera. Using a regular DO port (e.g. Dev1/port0/line7) allows for use of all modes. Using a CTR output (check portout of specific DAQ in use. For USB-6343 these are PFI12-15) directly downsamples camera clock input (clock needs to be connected to corresponding CTR source in this case, for USB-6343 these are PFI8, PFI3, PFI0, PFI6). This mode is useful for further improving frame synchronization when using multiple cameras and reduces jitter, but does not support setup of manual waveforms. Using a non-CTR PFI is not recommended as it has no advantages and only supports use of **Single Start Trigger**.

8. **ROI plotter**. Plots mean pixel value of selected ROI over the last 60s. The relevant ROI here is what is selected inside the camera stream by clicking two corner points, not the global ROI selected in the Main tab. One plot per camera.



9. **Notes.** These are session notes shared across a whole day, which will be saved alongside acquisition data and snaps in ‘session_notes.txt’.
10. **Objectives & Tube Lenses** is a convenient calculator tool to convert between magnification and focal length based on manufacturer standard reference tube lens focal length. These values are also used to convert pixel distances to absolute length when measuring distances in the camera stream or when drawing circles of specified radius in µm in DMD tab.

11. **Exit button.** Use this button to end Luminos session and close camera streams properly. Gives option to copy data to user folder set in user .json file.

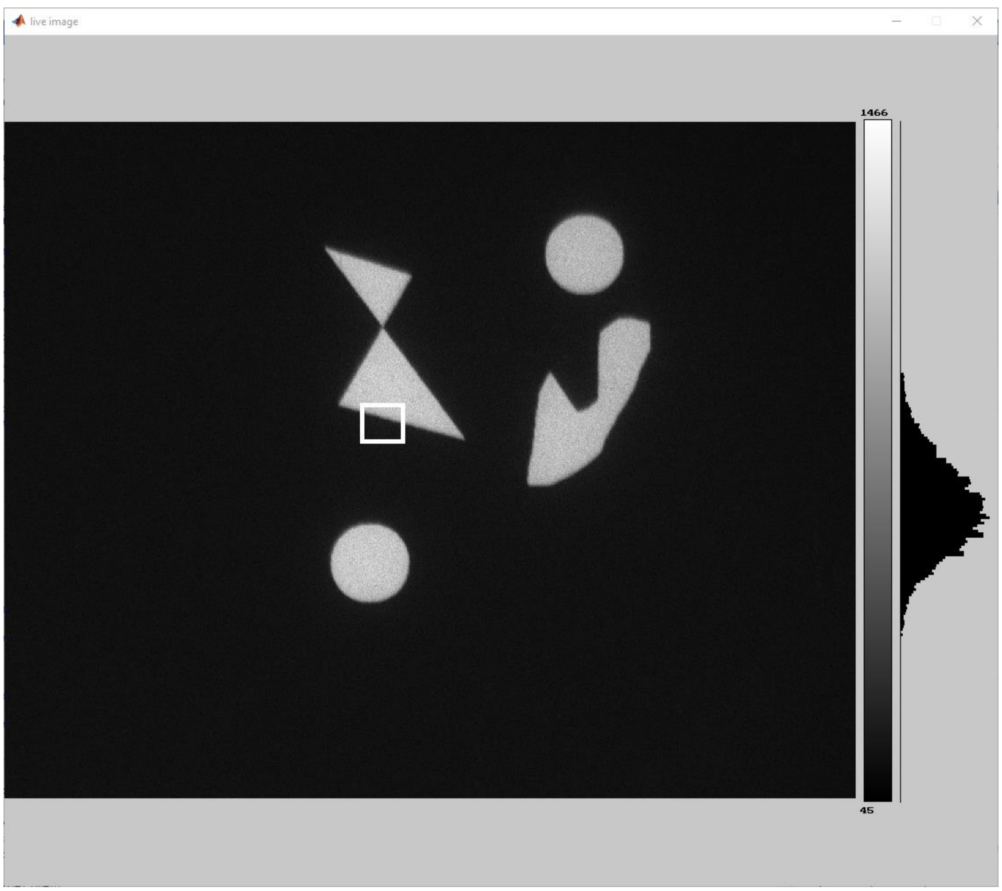
Camera Stream:

The camera stream shows current camera data when not acquiring data and contains several useful functions. Various Keyboard shortcuts are displayed in the main tab:

Keyboard shortcuts: Z for zoom in, X for zoom out, C for color, Numpad +/- for binning, Arrows to shift ROI, R to restart frozen stream, S to toggle scaling auto or fixed, H to toggle Histogram equalization

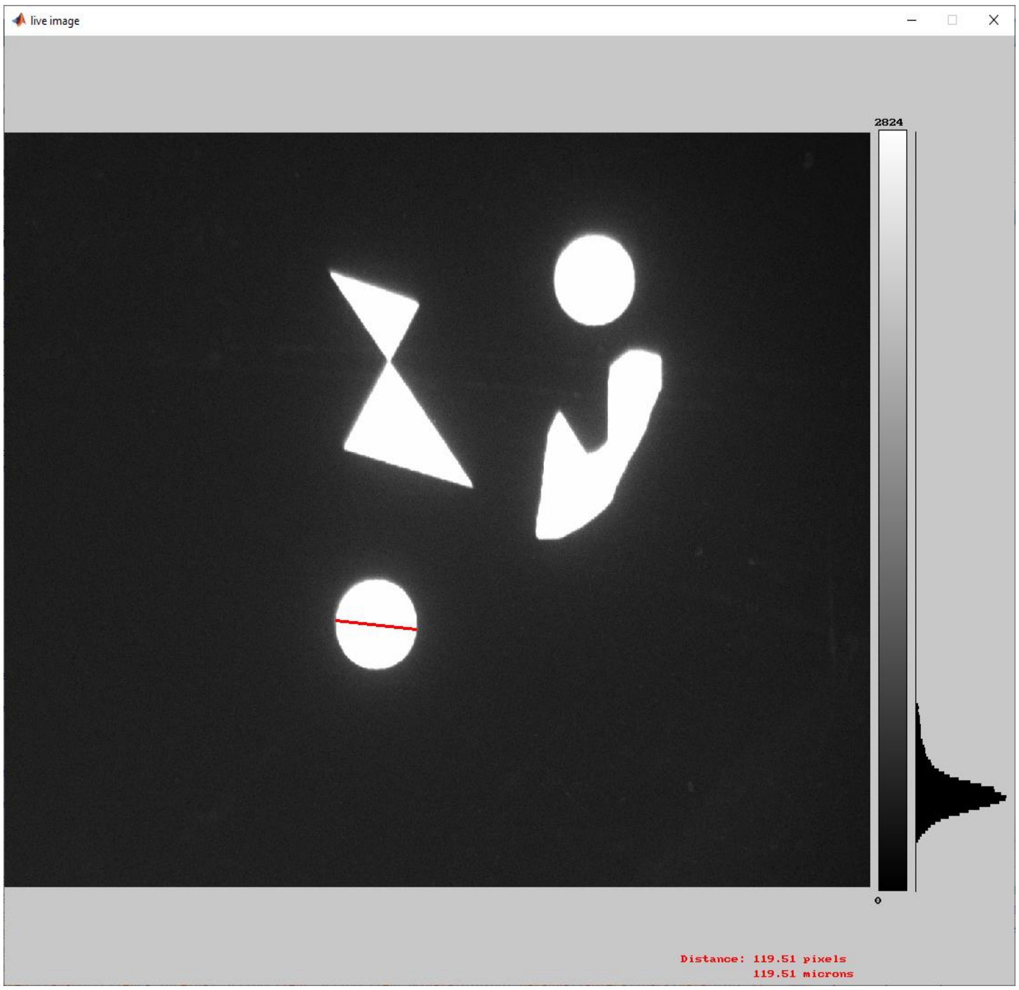
1. Select ROI

Left click two corner points to select an ROI. The mean pixel values of this ROI will be displayed in the Canera ROI Means Plots in the main tab, or zoom into ROI by pressing **z**.



2. Measure distances

Right click two points to calculate the distance between them. This can be used as a convenient scale bar, although it is not saved when taking a snap (use snipping tool or other utilities to get image). Distances in microns are calculated based on net magnification as set in Objectives & Tube Lenses.



3. Values ranges: Automatic range vs. fixed range

By default, the stream display shows data scaled to grayscale between the min and max pixel values in each frame. The **s** button toggles to fixed range. min and max of the fixed range can be adjusted as needed using the **mouse wheel** near the min and max values at the top and bottom of the color bar.

4. Colormaps

Switch between grayscale, grayscale inverted, hot and jet colormaps using the **c** button.

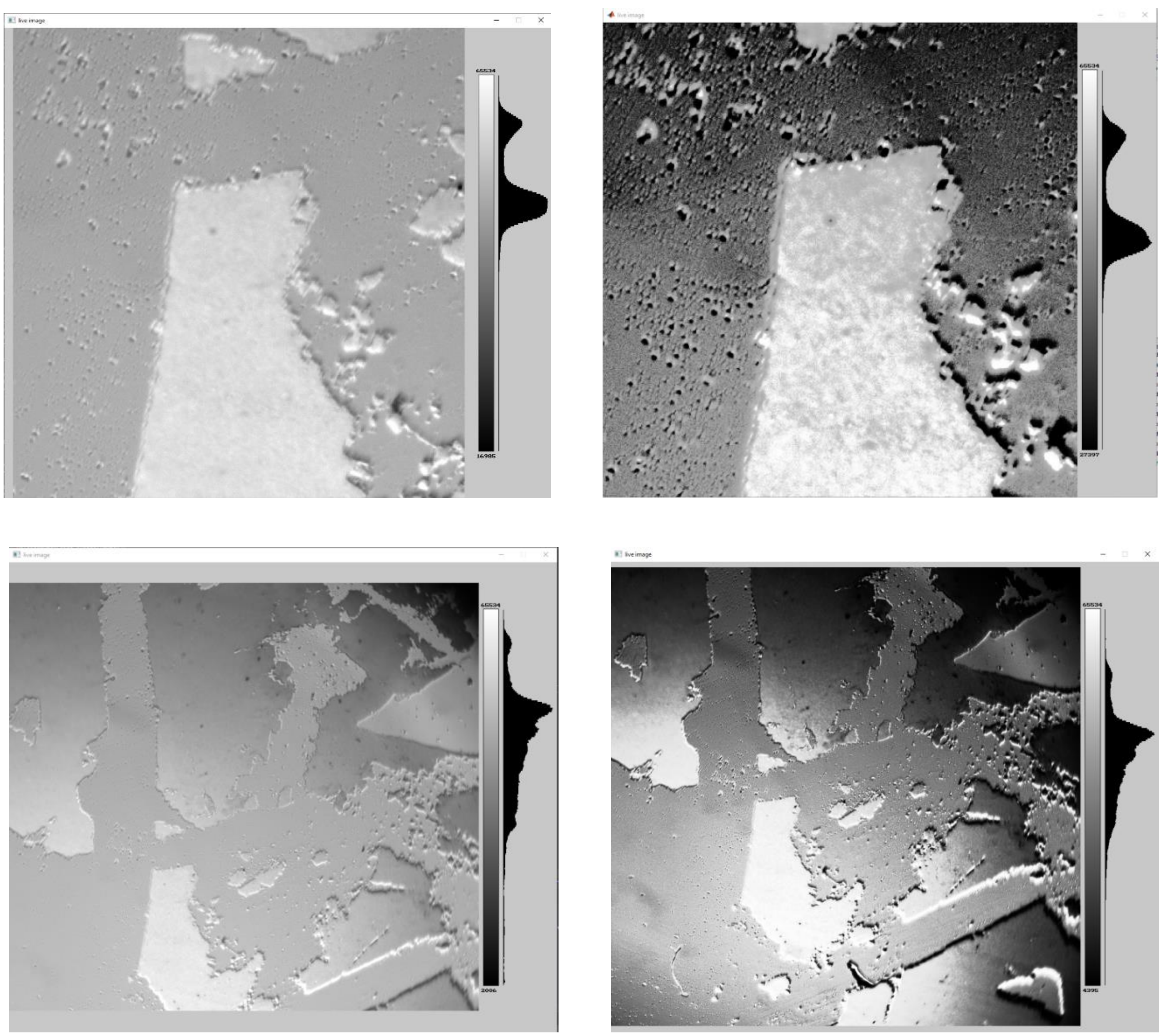
5. Histogram

A pixel brightness histogram is displayed next to the color bar. Use the **h** button to switch from a linear color map to Histogram equalization. This uses the histogram to calculate a non-linear color map that maximizes contrast, see e.g.
https://concepts.dhruvbadaya.in/histogram-equalization

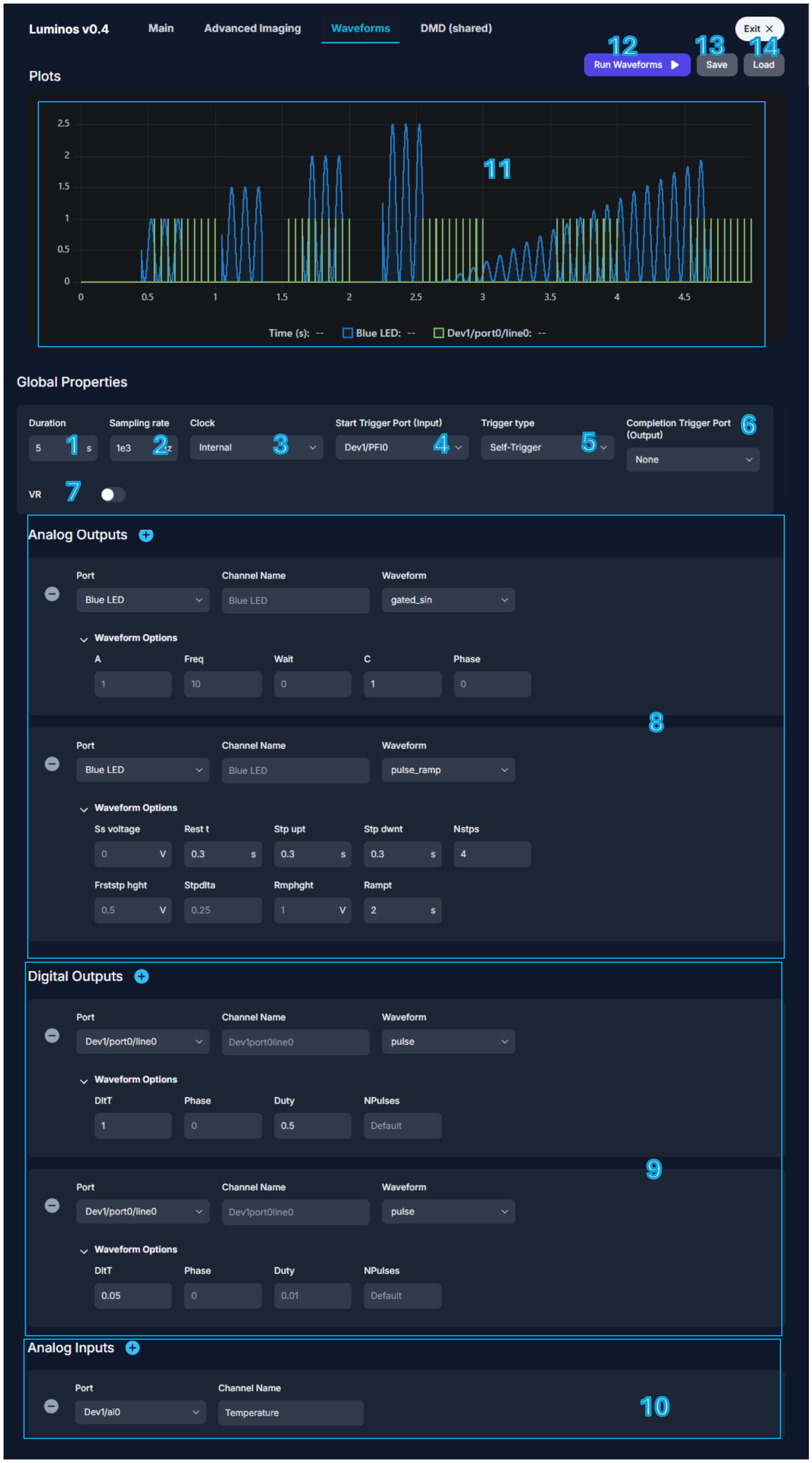
This does not affect collected data or snaps.

Examples:

Histogram equalization off | Histogram equalization on



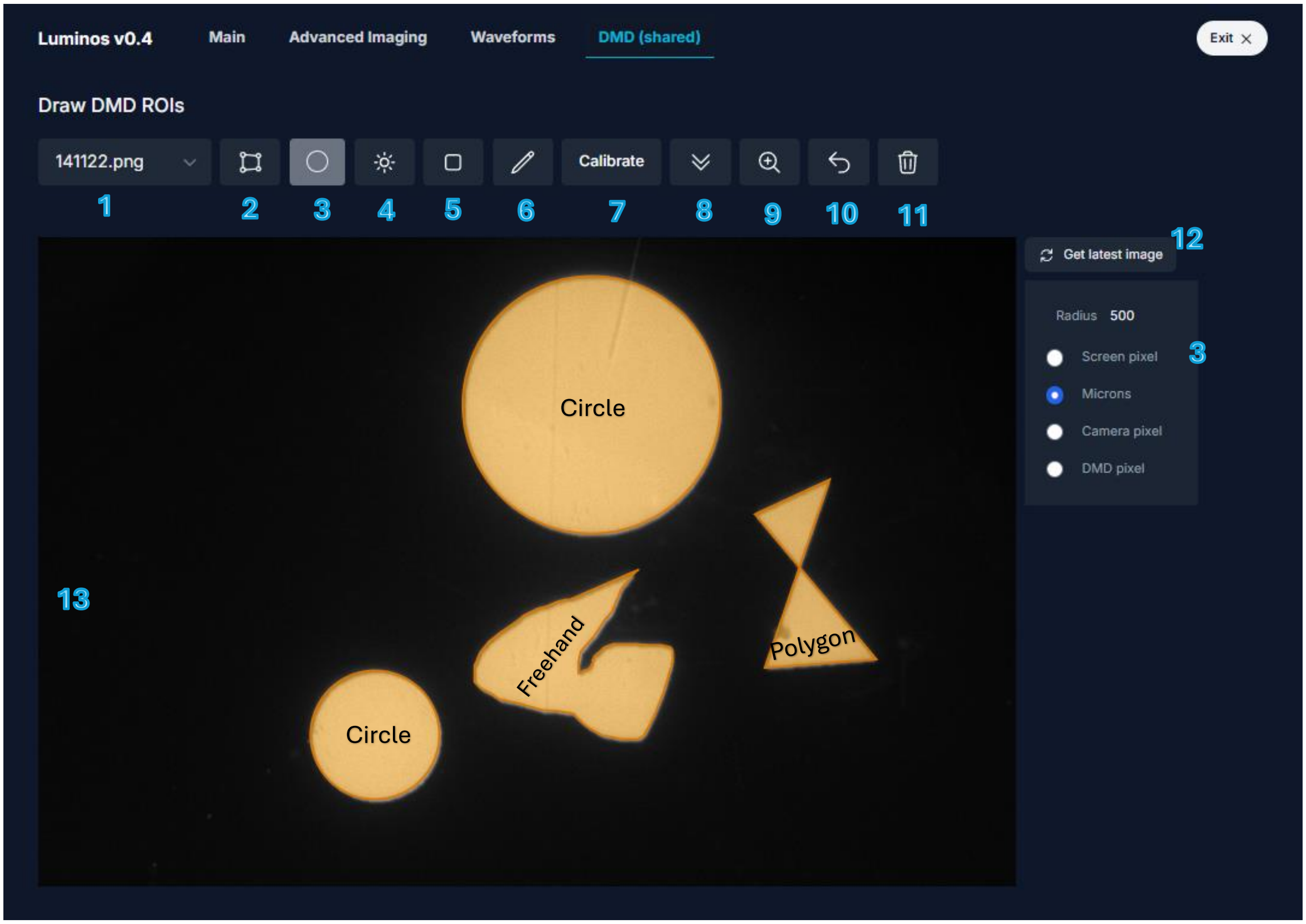
Waveforms Tab:



1. **Duration**. Set the total duration of the DAQ waveform in seconds.
2. **Sampling rate.** DAQ temporal sampling rate/resolutions for waveforms.
3. **Clock source.** Select the clock source for timing the waveform. Options include internal (DAQ) or external clock sources. Typically use camera clock or Internal DAQ clock here. If camera clock is selected, sampling rate is automatically set to hsyn rate of camera as set in the rig initializer.
4. **Start Trigger Port.** When waveforms are set up, they will wait for a pulse to this port to start. If trigger type is **Self-Trigger**, a trigger pulse is automatically sent to this port at the start of the acquisition, starting all waveforms. If it is **External**, the waveforms will not start until an external trigger pulse is received to this port.
5. **Trigger type** sets the trigger source for starting the waveform execution. "Self-Trigger" will start upon first rising edge from clock, "External" will wait for rising edge or TTL input from channel selected in 4.
6. **Completion Trigger Port** sends an optional pulse output at the end of the acquisition for communication with external devices (e.g. automatic sample changer).
7. **Use VR** for acquisition with VR server running in the background, e.g. for behavioral navigation experiments.
8. **Set up Analog Outputs**. Configure the AO channels. Press **+** button to add waveforms and **-** to remove them. Select waveform from presets or make your own file and add it to luminos\src\Devices\DAQ\Waveform_Functions with prefix awfm. Selecting two waveforms for the same port to multiply them together in order to build complex experiments from simpler primitives, see e.g. the sine-modulated pulses/ramp.
9. **Set up Digital Outputs**. Configure the DO channels Press **+** button to add waveforms and **-** to remove them. Select waveform from presets or make your own file and add it to luminos\src\Devices\DAQ\Waveform_Functions with prefix dwfm. Selecting two waveforms for the same port to multiply them together in order to build complex experiments from simpler primitives.
10. **Set up Analog Inputs.** Select AI channels to record input data from. Typically need to reject common mode signal by connecting both wires to ground across a high Ω resistor (i.e. differential measurement). Saved to buffered_tasks in Device_Data with resolution according to **Sampling rate**.
11. **Waveform plotter.** Visualize the generated waveform. The plotter displays the waveform in real-time, allowing adjustments to be made before running the waveform. When the total number of points per plot is > 1e6, auto-plotting is suspended for smoother performance.

12. **Run Waveform.** Execute the configured waveform without camera acquisition. Starts the waveform generation based on the set parameters and trigger conditions, without running the camera acquisition. This runs `Waveform_Standalone_Acquisition_JS.m`. AO/DO waveforms and analog input data are saved to a .mat file, but no .bin file is produced. Can be aborted from the main tab.
13. **Save Waveforms.** Save the current waveform settings to a file in luminous\src\User_Interface\relay\data. This allows for easy reuse and sharing of waveform configurations.
14. **Load Waveforms.** Load previously saved waveform settings. Waveforms are shared across all users on the rig and are stored in luminous\src\User_Interface\relay\data. When the list gets too cluttered the old waveforms can be deleted manually.

DMD Tab:



1. **Select reference image.** New images can be added by clicking 'Get latest image'. Image size, ROI or binning do not matter for drawing or calibration.

2. **Polygon mode.** Draw polygon ROIs by clicking vertices (like clicky).

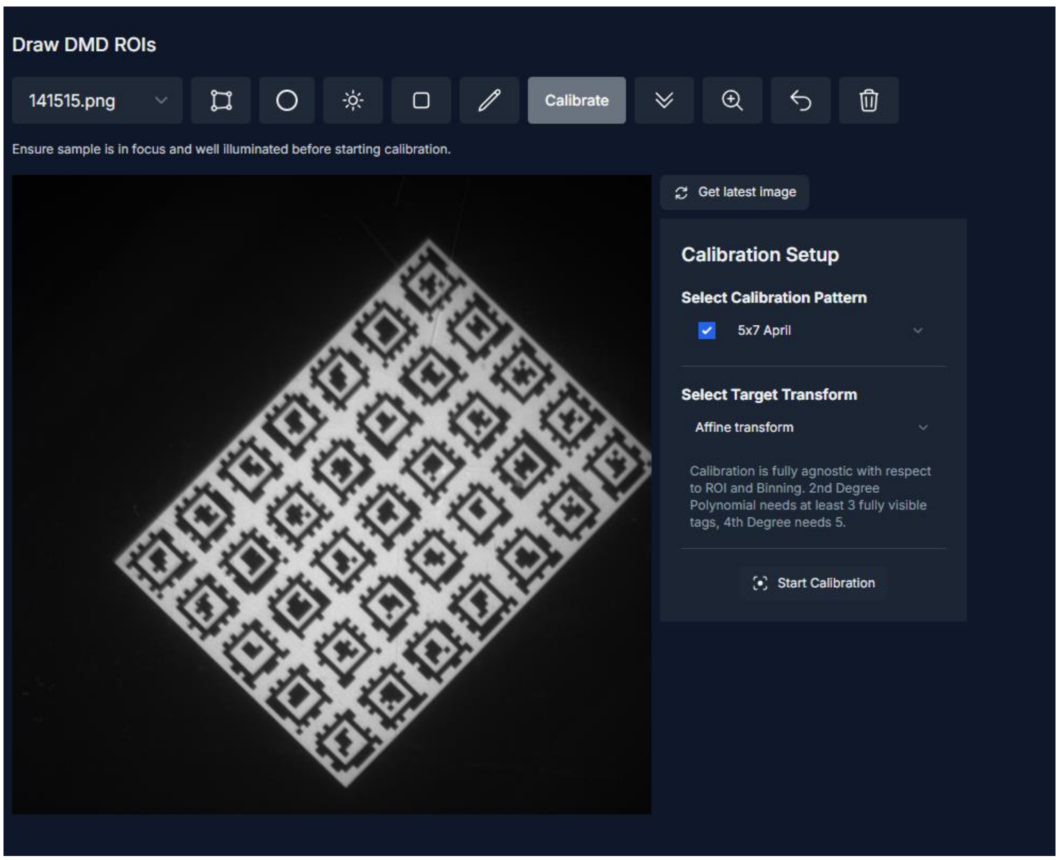
3. **Circle mode.** Click once to pick circle center, click second time to select radius. Radius units can be switched between screen pixel (size of circle on the GUI screen), camera pixel (in units of pixel on camera sensor), DMD pixel (based on calibration t-form determinant) or absolute size in microns (based on Objectives & Tube Lenses net magnification and camera pixel size).
   Tip: Double-clicking places a circle of the same size as the previous selection.

4. **All on.** Turn all DMD pixels on, transmit all light.

5. **Current FOV on.** Transmit all light for the entire current reference image.

6. **Freehand mode.** Draw ROI by drawing with mouse like MS Paint.

7. **Calibrate registration between camera and DMD.** Opens registration setup window. Registration is automatically performed by projecting an array of AprilTags or Manual. The numbers that are encoded in the tags are read out automatically and a tform between the camera and DMD is calculated and saved as DMD.tform. The calibration is **fully ROI agnostic**, i.e. the ROI during calibration and when picking the reference image do not matter. See next page for tips on getting a successful and accurate registration. See DMD calibration tips below.



8. **Export shapes to MATLAB.** Export all currently drawn shapes separately to MATLAB to dmd.shapes and corresponding DMD patterns to dmd.all_patterns. From MATLAB, combine and reorder patterns based on need and write to DMD by setting: `dmd.pattern_stack = patterns; dmd.Write_stack();`

9. **Zoom in to reference image.**

10. **Remove last drawn shape.**

11. **Clear all shapes.**

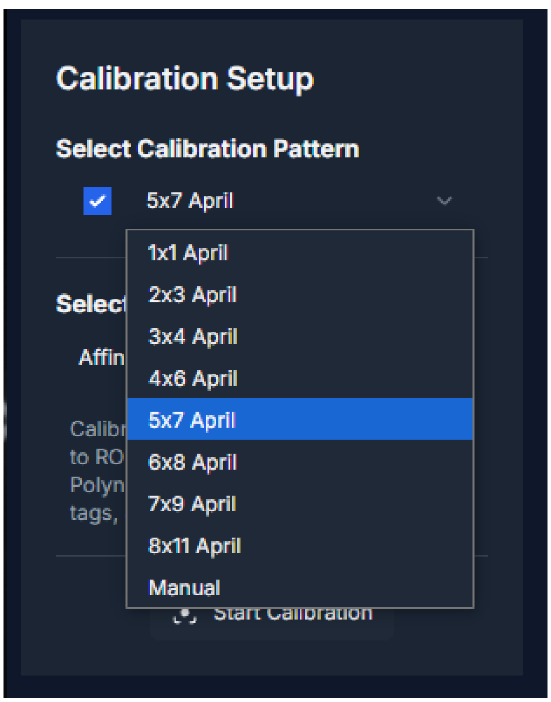
12. **Take snapshot**. Same as Snap in Main tab.

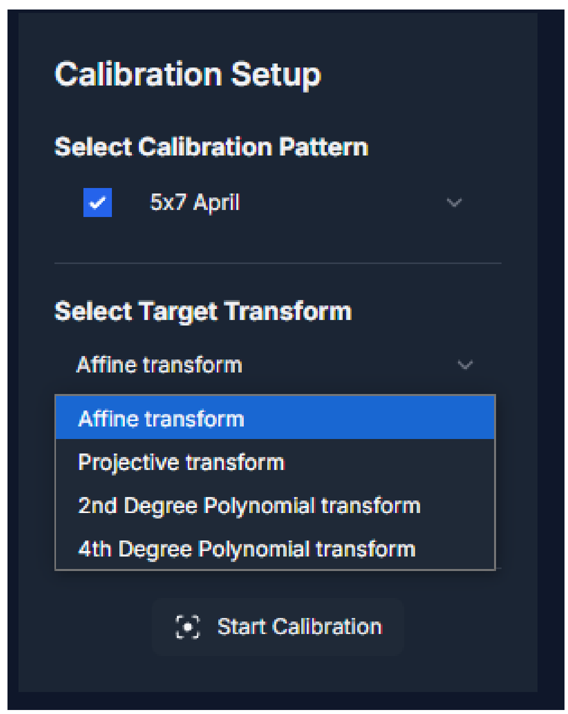
13. **Current drawing surface**, showing drawn shapes.

DMD Calibration Tips:

DMD calibration is automated and supports several calibration patterns and transform types.

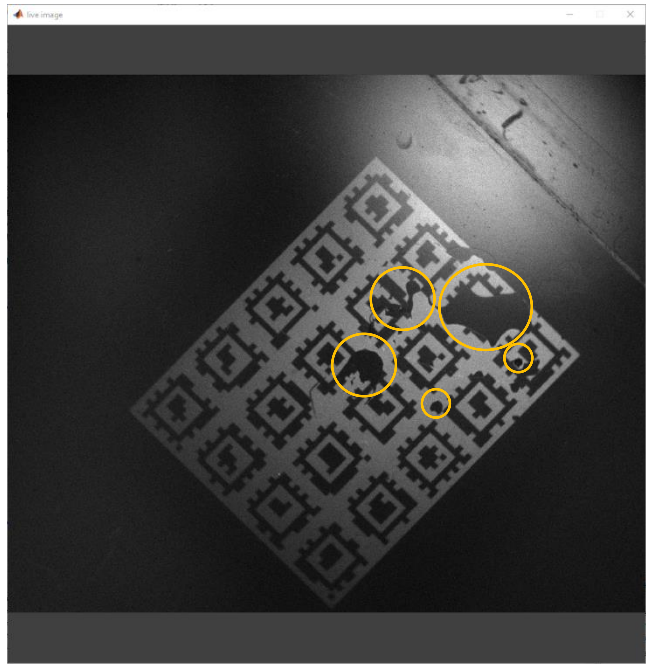
1. Select desired calibration pattern. Different numbers/sizes of April tags are available, e.g. 5x7 April will use a grid of 5x7 April tags for calibration. The more April tags are used the smaller each individual tag is. Turn on illumination (laser/LED) and pick a pattern that has at least one fully visible tag in the illuminated area. Preview patterns using checkbox.





Get the tags in focus on a homogeneous fluorescent slide.

Zoom out to see as many tags as possible.



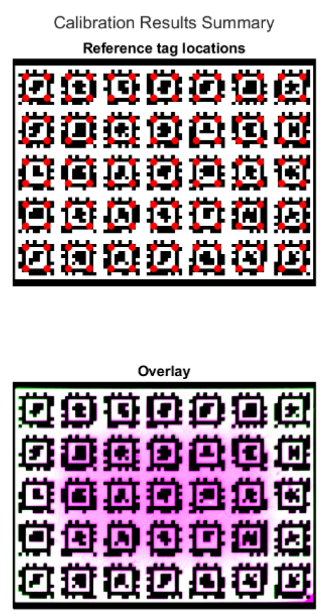
This is a decent registration sample. Inhomogeneities in the sample (marked with circles) prevent readout of some tags. Avoid areas like this.

The successfully read tags are shown below in “Reference tag locations” as red corner points, along with the overlay of DMD pattern and camera snap based on the calculated registration.

2. Pick transformation type. Affine and projective transformations usually perform well and only require a single visible tag. Polynomial transformations can help account for distortions and aberrations but require multiple tags. I recommend using the default Affine transformation in general.

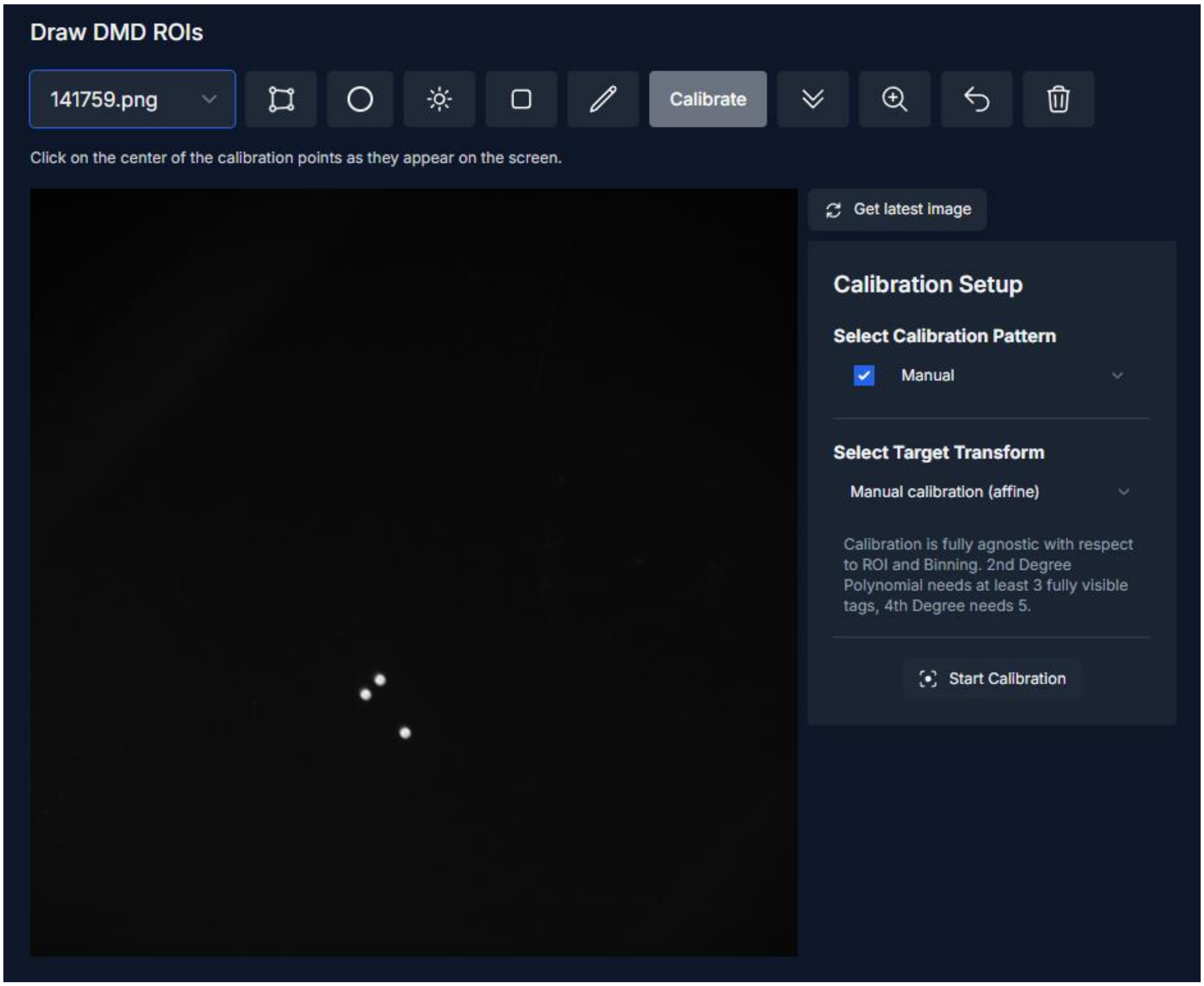
3. Click **Start Calibration**. Calibration is performed automatically, and a MATLAB window opens upon successful completion. If failed, retry with a better sample or different pattern.

Successful completion will open a MATLAB figure with calibration results. The first plot marks all successfully detected tags, the second plot is an overlay of the displayed DMD pattern and transformed camera FOV. The two patterns should match well if calibration was successful.

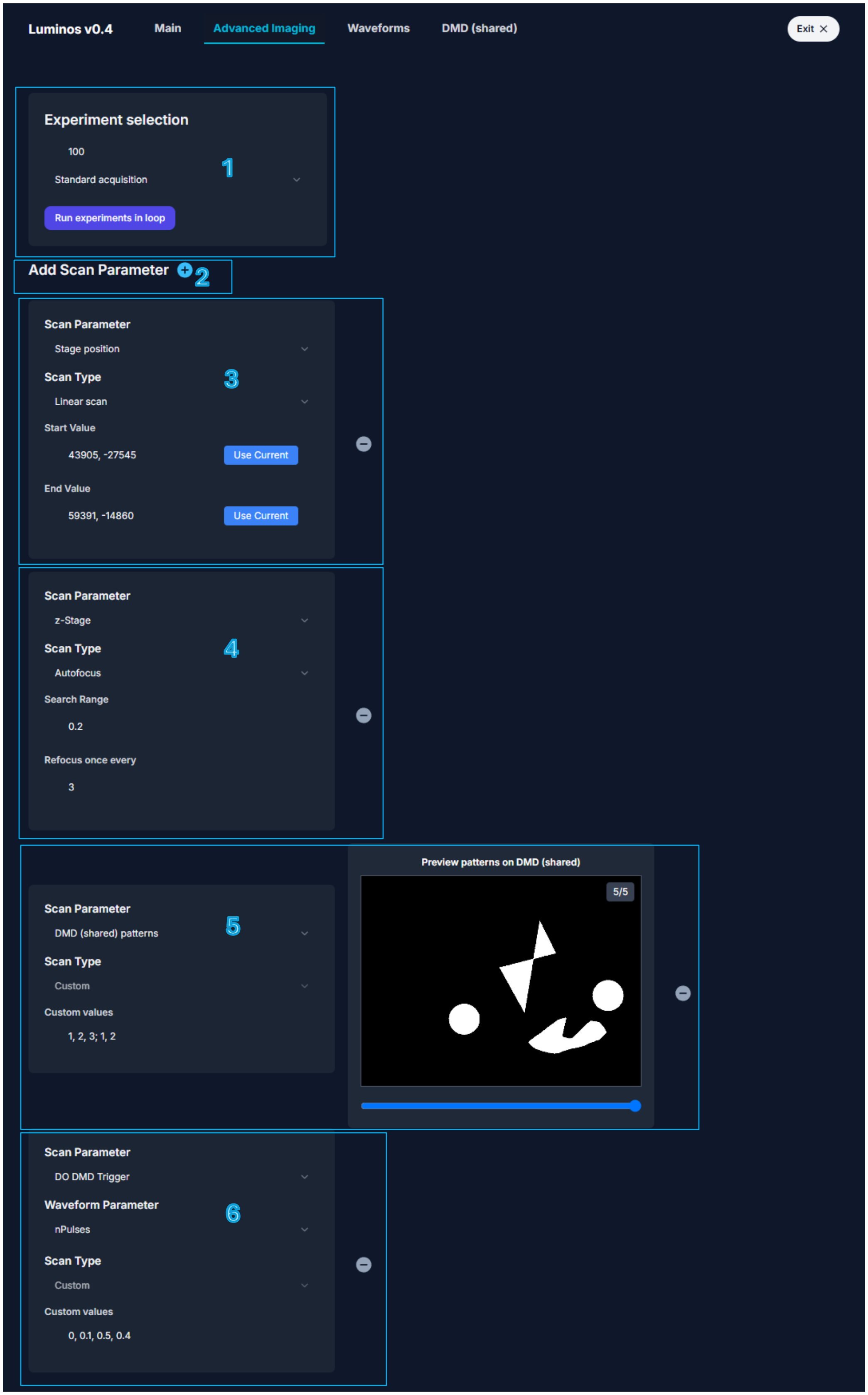


4. Manual Calibration

Besides the automatic option, manual calibration can be performed by using the **Manual** pattern option. This will sequentially project a series of points (as set in for the DMD in the rig initializer .json). Click Start Calibration and left click in the center of the points as they appear. This calibration mode is generally inferior to automatic calibration, but can be used if no good calibration sample is available or automatic consistently fails.

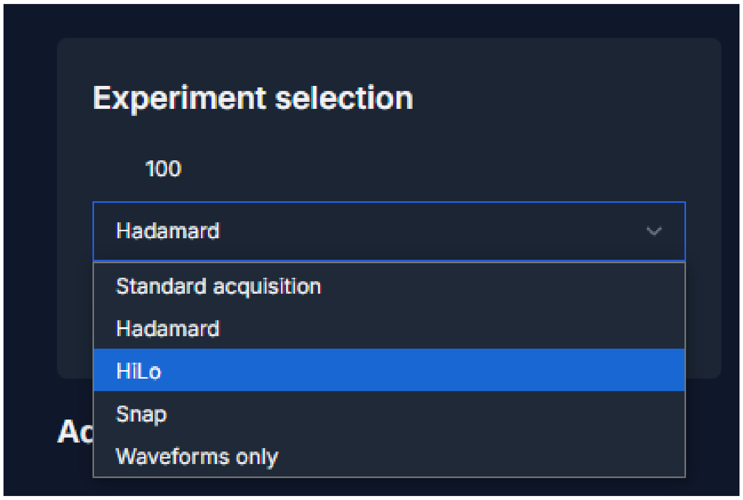


Advanced Imaging Tab:



The Advanced Imaging Tab is designed for convenient setup of various meta experiments.

1. **Experiment selection**. Set number of experiment repeats and select out of one of the supported experiment types. Click **Run experiments in loop** to start series of experiments. In general, data is saved in a series of folders named according to the Experiment name field in the main tab + burst1, burst2, burst 3, etc. Supported experiments include:



Standard acquisition: Same as running **Start Acquisition** in main tab. Set up camera parameters in main tab and waveforms in waveform tab, then run repeat acquisition.

Hadamard: This option is only available if a DMD is connected. Display a series of digital confocal speckle pattern and reconstruct afterwards to reject scattered light. See detailed description further below.

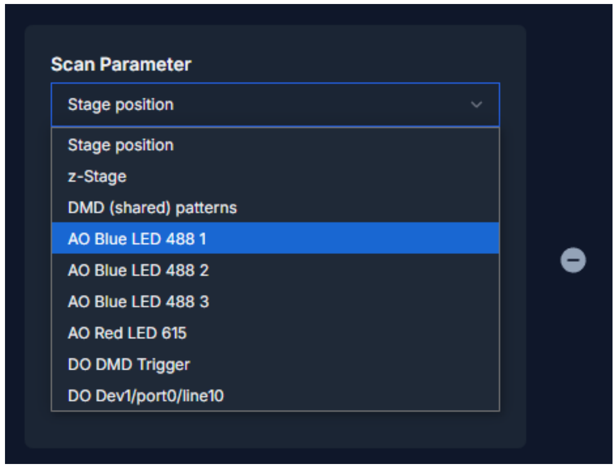
HiLo: This option is only available if a DMD is connected. Display a single digital confocal speckle pattern and a single all on image and reconstruct afterwards to reject scattered light. Similar to Hadamard, but faster and less accurate as fewer images are used. See detailed description further below.

Snap: Take a series of snapshots.

Waveforms only: Same as running **Run Waveforms** in waveform tab. Set up waveforms in waveform tab, then run repeatedly.

If multiple cameras or DMDs are available, the Hadamard, HiLo and Snap options display additional drop down menus to pick which to use.

2. **Add Scan Parameter**. The Advanced imaging tab supports changing or scanning an arbitrary number of various parameters between experiment repeats. Click the **+** button to add a new scan parameter, similar to adding waveforms, and remove using **-**. Scannable parameter options include:



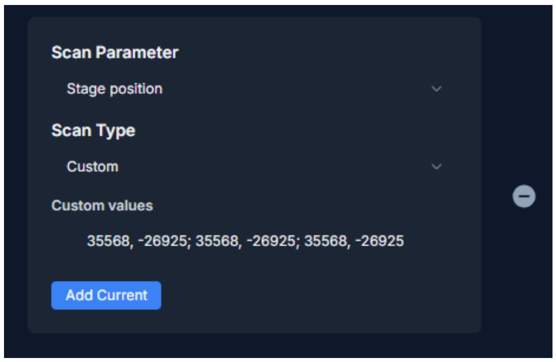
- xy/xyz-Stage positions and z-Stage if connected
- DMD patterns if connected
- Analog (AO) and Digital (DO) waveform parameters if waveforms are set up

The following are more detailed examples of scannable parameters. In this example, a xy-Stage is connected in addition to a separate z-Stage, which can be controlled separately.

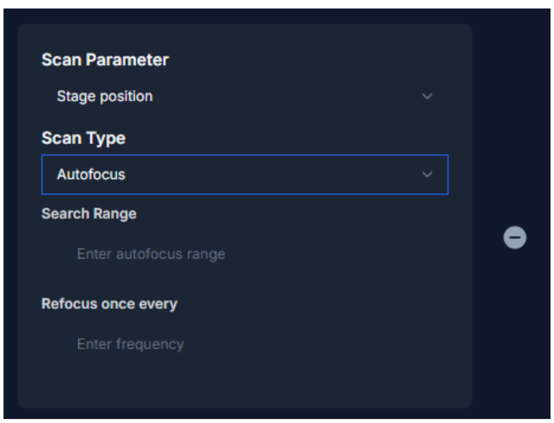
3. **Stage Position scan:** If a xy-Stage or xyz-Stage is connected, set position between runs. **Scan types** for stages include:

   Linear scan, which interpolates positions between start and end using linspace, using the what is entered for **Experiment number** in **Experiment selection** of steps (here 100 steps). Use current loads and inputs current stage position for start or end. Useful for z-scans or sequentially imaging a large FOV or multiwell-plate.

   Custom, which visits entered values (in a loop if Experiment number is larger than number of entered positions). Use Add Current to automatically add current stage position to list in the right format:



   Autofocus, only available for xyz-Stages and z-Stages, refocuses between acquisitions by finding position within selected **Search Range** (in µm for xyz-Stages, mm for z-Stages) which maximizes brightness of 99.99th percentile pixel brightness. 100µm of **Search Range** will search from 100µm below to 100µm above current position, using 2 quick scans (one coarse, one fine). This usually takes ~6s, depending on the search range. This is particularly useful for cell culture and other thin samples. **Refocus once every** sets how often to refocus, e.g. 3 will refocus once every 3 acquisitions. This is compatible with all experiment types.

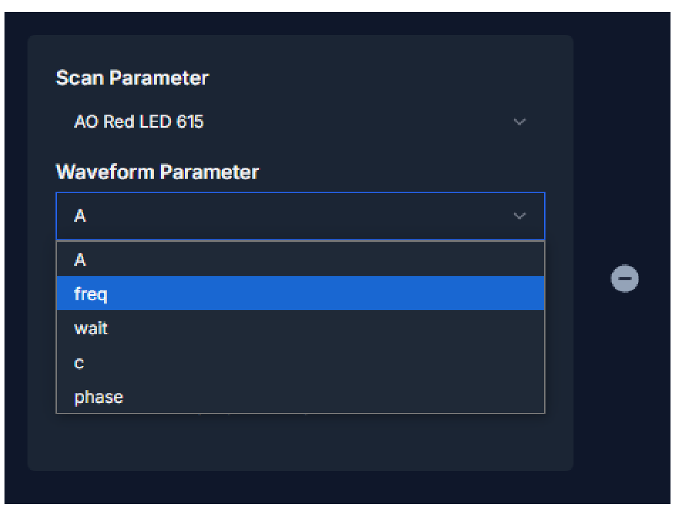


Autofocus requires light to be on, but modulators and shutters are turned off/closed between acquisitions. To use, set up a waveform with **constant** values for each shutter and light source/modulator that needs to be turned on (duration and trigger settings do not matter), and save as “**Autofocus**”. The function will look for the most recent file containing “**Autofocus**” in its title and will open shutters/set modulators as in this file for the duration of the autofocus process.

4. **z-Stage Position scan:** Analogous to stage positions. Note that units are in mm.

5. **DMD patterns:** If DMD is connected, send arbitrary patterns from library in dmd.all_patterns. **Scan type** is only **Custom**. In the example above, the first experiment will send patterns 1, 2 and 3 to DMD for the first experiment, then 1 and 2 for the second experiment, and then alternate between these. Note that switching between patterns requires DAQ trigger pulses to the DMD trigger port. Use the preview window and slider to preview the library in dmd.all_patterns and select which patterns to add. Patterns are automatically added when using **Export** function in DMD tab or **Generate Patterns** in Hadamard setup but can also be added manually.

6. **Analog and Digital waveform parameters:** Once waveforms have been set in waveform tab, select waveform parameters to scan through. **Scan type** is only **Custom**. Select a waveform parameter to scan from dropdown for given waveform, then enter values. Using this overwrites the value set in the waveform tab and cycles through the custom values instead.



Hadamard Imaging:

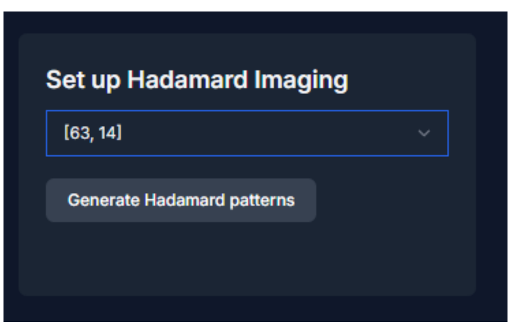
Hadamard imaging is a digital confocal technique which requires a DMD in the beam path and can be set up from the Advanced Imaging tab which calls the MATLAB function `Waveform_Camera_Sync_Hadamard.m`. Hadamard imaging computationally rejects scattered light by projecting a series (12-64 patterns) of speckle-like patterns and only using data from camera pixels that were illuminated directly during a given patter. The main application of Hadamard imaging is the acquisition of high-quality structural images.

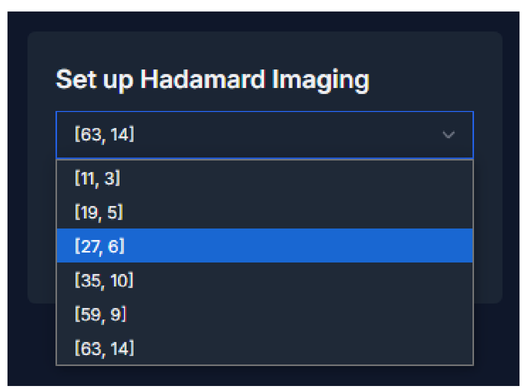
For a deeper discussion see:
https://github.com/adamcohenlab/Compressed-Hadamard-Code
and
Vicente J Parot *et al* 2019 *J. Phys. D: Appl. Phys.* **52** 144001
https://doi.org/10.1088/1361-6463/aafe88

The process consists of four parts:

1. Set up Hadamard patterns:

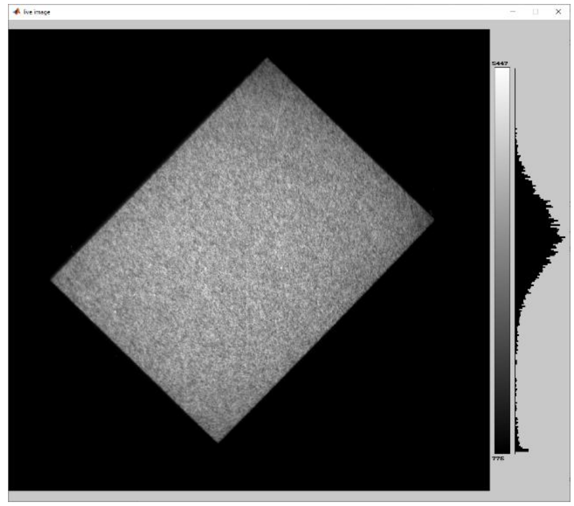
In the Advanced Imaging tab, select the **Hadamard** option. This opens the **Set up Hadamard Imaging** window, which contains dropdown menus for pattern selection and additional menus for selecting which camera and DMD to use if there are multiple options. The number of patterns is the first number in the pattern selection dropdown menu + 1, e.g. using [11, 3] will generate 12 Hadamard patterns, [63, 14] will generate 64, and send them to the selected DMD (to dmd.all_patterns, dmd.Pattern_stack) and will Write_Stack() to the DMD. The first pattern is displayed once the DMD receives a pulse.





Pattern in DMD preview:

Pattern in Camera Stream:



2. Acquire Hadamard reference images:

In order to reject light from pixels that are not directly illuminated it is necessary to get a map of which pixels on the camera are illuminated for any given Hadamard DMD pattern. The preferred way to do this is to acquire a stack of Hadamard images using a homogeneous fluorescent sample, as is used for the DMD calibration. Once the patterns have been generated, make sure the sample is in focus, set the camera exposure time and ROI, and set up a constant waveform so the sample is illuminated but not overexposed. The **Duration** should be set so the camera has enough time to acquire all Hadamard images, e.g. if using 64 patterns with 0.05s **Exposure time**, use >3.2s **Duration** in the waveforms tab. Use **Single Start Trigger** or **Trigger each Frame** for the camera trigger. `Waveform_Camera_Sync_Hadamard.m` automatically sets up a waveform for the DMD to change patterns at the start of each new camera frame. Adding a DMD Trigger waveform in the waveform tab will override this waveform. Clicking Run experiments in loop (even with just **Number of experiments** = 1) calls `Waveform_Camera_Sync_Hadamard.m` and acquires one frame per Hadamard pattern.

3. Acquire Hadamard images of sample:

Repeat the setup for step 2 with the sample. Light intensity and camera exposure time may differ between the reference and the sample, but the camera ROI must be the same for successful reconstruction.

4. Reconstruct Hadamard images:

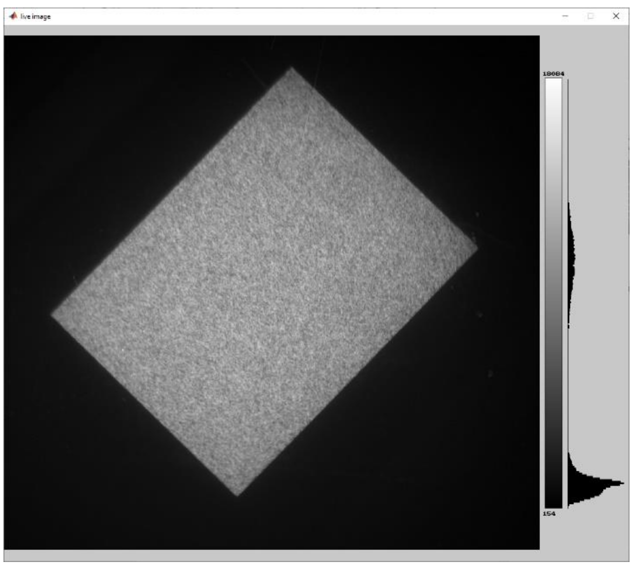
Use the `hadamard_reconstruct.m` function to reconstruct the sample Hadamard image from the reference and sample image stacks. The output of this file is the reconstructed Hadamard image of the sample with scattered light rejected.

HiLo Imaging:

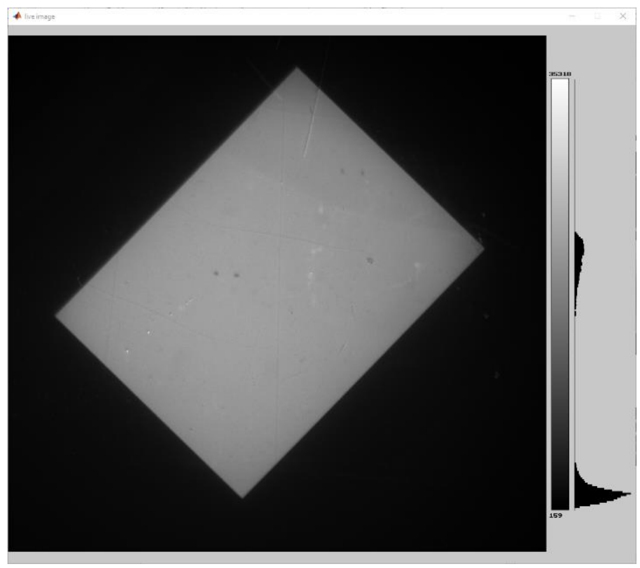
HiLo Imaging serves a similar purpose to Hadamard but uses only two DMD patterns: An all-on image and an image with a speckle-pattern generated by `round(rand(dmd_size))`. After reconstruction the background fluorescence and scattered light can be filtered out for a high-quality structural image.

For a deeper discussion see:
Daryl Lim *et al* 2008 *Opt. Lett.* **33** 1819-1821
https://opg.optica.org/ol/abstract.cfm?uri=ol-33-16-1819

Speckle Image in Camera Stream:



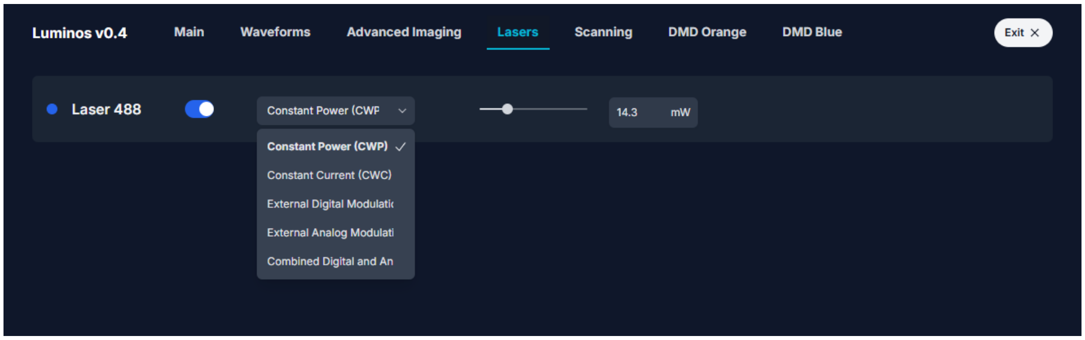
All-on Image in Camera Stream:



To set up HiLo acquisition, illuminate the sample and set camera exposure time and ROI. In the Advanced Imaging tab, select **HiLo** as the **Experiment type**. If multiple DMDs or cameras are available, a window with dropdown menus allows for the selection of the right devices. Click **Run experiments in loop** to acquire HiLo data. No waveforms are run during HiLo acquisition. A stack of images, with two images per repetition, is collected and saved together in a .bin file.
To reconstruct background-free HiLo images from the image stack use `hadamard_HiLo.m`

Lasers Tab:

Straightforward control over lasers that support direct external control. Toggle laser on/off and adjust power. Various control modes are available depending on needs.

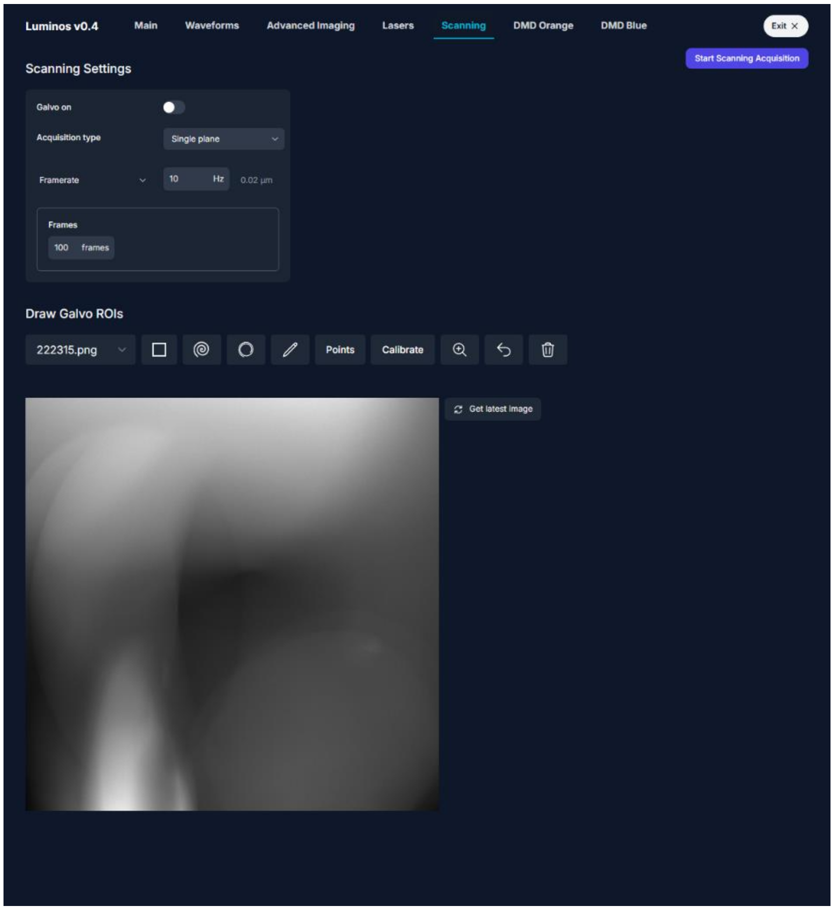


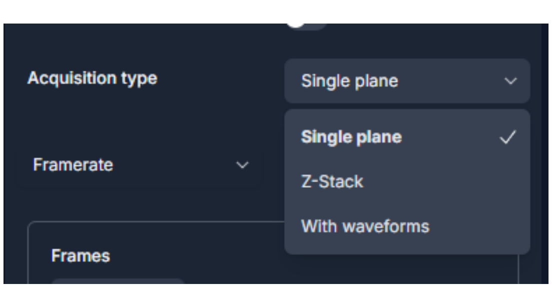
SLM Tab:

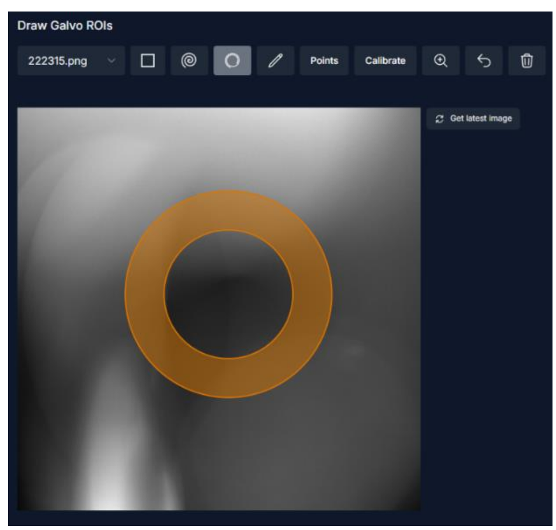
Under construction. Will add more information after remodeling.

Scanning Tab:

Under construction. Will add more information after remodeling.







Setting up camera triggers, clocks and vsync counters:

Cameras can typically be triggered in one of two ways: Using a single trigger to start acquisition (**Single Start Trigger**), after which each frame is timed based on the camera's internal clock, or using waiting for a trigger to start the next frame (other two options). Pulses are always sent to the camera trigger port set in the rig initializer .json under "trigger" (use of "daqTrigCounter" is deprecated).
Camera trigger ports should be connected to the camera trigger input. When triggering each frame with a pulse, two options are available: **Trigger each Frame** requires an additional input **DAQ Trigger Period (ms)** and will set the DAQ to output a train of pulses at the set rate until DAQ acquisition is completed.

Manual Setup in Waveforms is the most flexible option. When using this mode, camera triggers are not set automatically. Instead, use the Waveforms tab to manually set up an arbitrary waveform for the camera trigger, e.g. for delayed camera acquisition, bursts of frames at specific intervals, etc. This mode is also compatible with external triggering, i.e. if a separate device is used to trigger the camera. No waveform needs to be set in this case, but a warning will be displayed warning that no waveform is set for the camera.
The availability of these modes depends on on what type of port is used to trigger the camera. Using a regular DO port (e.g. Dev1/port0/line7) allows for use of all modes. Using a CTR output (check portout of specific DAQ in use. For USB-6343 these are PFI12-15) directly downsamples camera clock input (clock needs to be connected to corresponding CTR source in this case, for USB-6343 these are PFI8, PFI3, PFI0, PFI6). This mode is useful for further improving frame synchronization when using multiple cameras and reduces jitter, but does not support setup of manual waveforms. Using a non-CTR PFI is not recommended as it has no advantages and only supports use of **Single Start Trigger**.
Waveforms in the DAQ update every time they receive a pulse from the **Clock** source that is set in the waveform tab. The are two common ways to run acquisitions:

1. Use the **Internal** DAQ clock base. This clock can be set to arbitrary sampling rates by setting **Sampling rate** in the waveform tab. Typically, voltage imaging experiments are run at 100kHz, with about 100 clock ticks per frame at 1kHz frame rate, resulting in at most 1% jitter from misaligned camera and DAQ clocks. When using a slower frame rate, e.g. 5Hz for Calcium imaging, lower clock rates (typically 1kHz) can be used without issues.
2. Use the **Camera clock** base. Cameras typically have an **hsync/clock** output, which outputs a pulse every time a new line is read out (100kHz for Hamamatsu Orca Flash, 200kHz for Orca Fusion). The clock rate is set under "**hsync_rate**" in the rig initializer, along with the PFI port (generally use a CTR source here) that receives the camera clock under "**clock**". Setting the clock source to the Camera clock in the waveform tab will then use the camera

clock pulses to tick the waveforms forward, thereby synchronizing the DAQ to the camera clock and reducing jitter. If a "**hsync_rate**" is set in the rig initializer .json and a valid camera clock (matching "**clock**" in the initializer .json) is selected, the **Sampling rate** is automatically set to "**hsync_rate**" when running **Start Acquisition** from the main tab or **Standard acquisition** or **Hadamard** from the Advanced imaging tab, ignoring GUI inputs. **Run Waveforms** from the waveform tab or **Waveforms only** in the Advanced imaging tab respects the clock rate set in the GUI, even when using a camera clock base.

In principle, using an external clock base is possible, if ever necessary (potentially useful for analog input data acquisition at irregular intervals, as data is collected once every clock tick?). Connect clock source to a PFI port and select the appropriate **Clock** and **Sampling rate** (if the goal is to do analog input data acquisition, then the only thing that matters here is the total number of samples **Duration*Sampling rate**).

vsync frame counter: For cameras that have a **vsync** output, which sends a pulse every time a frame is acquired, the output can be connected to a CTR source. The counter increments by 1 every time a pulse is received and can be used to align camera frames and other waveforms more precisely and check for dropped frames. The **vsync** port can be defined in the rig initializer .json under "vsync" and is automatically read out when running an acquisition, and later checked from the .mat metadata under **dq_session.Counter_Inputs**.

Running Luminos with multiple cameras:

In Luminos camera objects have a property **camera.slave**, which is **false** by default if not explicitly set to **true** in the rig initializer .json. Trigger waveforms are only built for **non-slave** cameras. When using a single camera, camera**.slave** should always be **false** so trigger waveforms are generated. When using multiple cameras, different setups are possible depending on the experimental needs.

Master-Slave: If very precise synchronization between cameras is crucial, e.g. when performing voltage imaging with two cameras in different focal planes or ROIs simultaneously, one camera should be master (**camera.slave = false**) and one should be slave (**camera.slave = true**). Both cameras should be connected to the same trigger output port (in this case a CTR out is best), the master camera clock should be used as the acquisition clock base, and both cameras should be set to “**Trigger each Frame**”. As cameras are triggered from the same source off the master camera clock, optimal synchronization between cameras is achieved. In this configuration, the slave camera can only be run at the same frame rate as the master. Setting the slave camera to “Single Start Trigger” allows for different frame rates, however doing this comes decreases the synchronization between cameras and defeats the main advantage of the configuration. Both slave and master cameras can have separate **vsync** counters.

Master-Master: If very precise synchronization between cameras is not a priority, e.g. when imaging voltage on one camera and Calcium on the other, using both cameras as master is the preferred configuration. In this case each camera has its own separate trigger port, trigger waveform, frame rate, etc. Trigger sources can be mixed-and-matched, e.g. one camera can run on a non-PFI port with Manual Setup in Waveforms, while the other runs off a CTR out. Since only one clock base source can be set per experiment, it is better to use the **Clock** of the camera running at the higher frame rate for less jitter.

Loading binary video data from .bin files:

Use `[mov, avgImg, Device_Data] = Extract_Mov(path)` or `data = readBinMov(path,nrow,ncol)`.

Extracting waveforms and other data from .mat files:

In order to correctly read waveforms on Device Data, the class definitions for buffered tasks need to be in the MATLAB path. Add the following files to your path:

- luminos\src\Devices\DAQ\NI_DAQ\DQ_channel.m
- All contents of luminos\src\Devices\DAQ\NI_DAQ\Buffered_Tasks

The data can be found in Device_Data{*x*}.buffered_tasks(*y*).channels(*z*).data. *x* will depend on the specific rig, e.g. *x* = 3 on Firefly. *y* = *1* through *3* for analog outputs, analog inputs and digital outputs respectively, *z* is the channel index. Check Device_Data{*x*}.buffered_tasks(*y*).channels(*z*).name for to pick the channel.