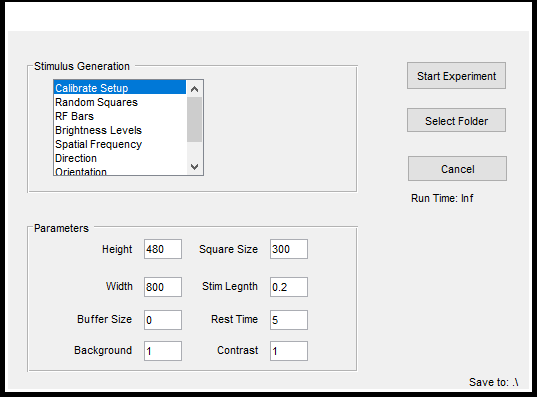
PIPELINE OVERVIEW

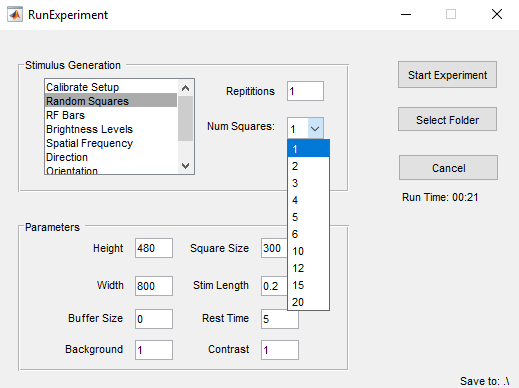
In 2016-2017, the following code was written for the purposes of showing visual stimuli to larval zebrafish (6 dpf) with nuclear-targeted GCaMP and analysing cellular responses by processing images collected by 2-photon microscopy. This document is a brief overview of the pipeline.



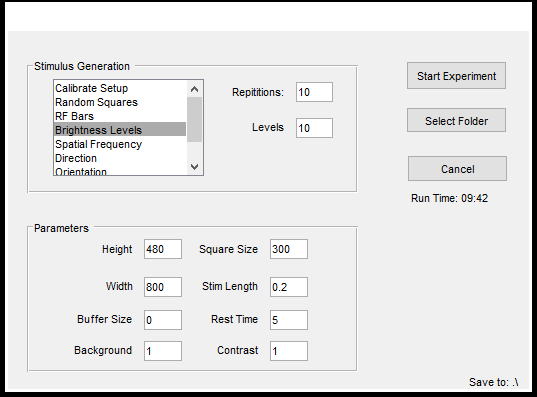
This is the GUI interface for the function *RunExperiment.m*.

## Stimulus Types Overview

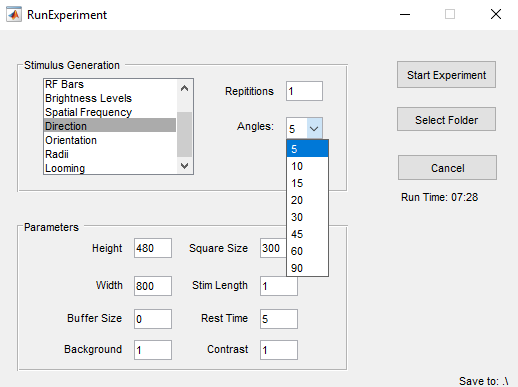
* Calibrate Setup: This simply displays the display area defined by the parameters *Square Size* (see parameter overview). This is useful for making sure your screen is aligned properly.



* Random Squares: This stimulus type presents squares in different spatial locations. Used for receptive field mapping. The field *Num Squares* defines the number of squares that are shown, i.e. if 6 is selected, the display field is divided into a 6 x 6 grid. For each stimulus presentation, one of the grids is activated. The list for *Num Squares* is therefore made up of the factors of the parameter *Square Size* (see parameter overview). i.e. 1, 2, 3, 4, 5, 6, 10 , 12, 15, 20 are all divisors of 300. The control stimulus is a blank screen.
* RF Bars: This stands for receptive field bars. It is similar to *Random Squares*, except instead of squares it displays bars that are horizontal and vertical in orientation. This maps the azimuthal and elevation separately.
* Brightness Levels: This displays different levels of brightness, the number of levels being defined by the parameter *Levels*. In this example, 10 levels will be displayed, that is images with intensity levels {0.1, 0.2, …, 1}. The control stimulus is a blank screen, i.e. an intensity level of 0.



* Spatial Frequency: This displays sinusoidal gratings with different frequencies. These frequencies, similar to *Num Squares* are all of the factors of the width of the screen. For example with a width of 800 pixels, there will be 16 frequencies; sinusoids with periods of {800,400,200,160,100,80,50,40,32,25,20,16,10,8,5,4}. Note that periods of 1 and 2 are not shown due to aliasing.[[1]](#footnote-1) The sinusoid then counterphases with a frequency of 5 HZ (hardcoded.) The control stimulus is a static presentation of the previous spatial frequency to maintain average light levels coming from the screen.[[2]](#footnote-2) For more information, see *Spatial Frequency Stimulus Information.txt.*
* Direction: This shows sinusoidal gratings moving in the defined directions. The parameter *Angles* denotes the interval between angles. E.g. if angles = 5, that means grating will be shown moving at 5 degrees, 10 degree, 15… etc. We tend to use angles = 30. This gives us 12 direction. For the cardinal directions, choose angles = 90. The control stimulus is a stationary grating or a random orientation.



* Orientation: Shows a bar intersecting the center of the display area at defined angles similar to *Direction*. Currently, the only way to change the width of the bar is by changing the code directly.
* Radii: This shows circles of different radii centered on the display area. Radii are linearly spaced from 0 to half the height of the screen, with the number of radii defined by the parameter *Levels* (similar to *Brightness*). The control stimulus is a blank screen (a radius of zero).
* Looming: Created by Michael Lynn, this presents a looming stimulus, with the speed of looing defined by *radius/velocity* and the starting size of the circle defined by *Min obj size.* The control is a blank screen.

### Parameter Overview

* Height/Width: This is the height and width of the screen on which the stimulus will be presented. In this example a 480x800px LED screen is being used. This is important for determining the size of the stimulus being presented, especially for stimuli like Spatial Frequency and Radii.
* Square Size: This is the size of the part of the screen where you want to display your stimuli. This is important for stimuli such as Random Squares and RF Bars. Currently, the display area can only be a square, and is centered in the middle of the screen. You can adjust the height of the area with the field Buffer Size.
* Buffer Size: This is the number of pixels from the bottom of the screen to the bottom of your desired display area. For example, if your screen has a height of 480px, a display area of 300x300px and you want the display area to be perfectly centered on the screen, you would need to set the Buffer Size to 90.
* Stim Length: This is the length of time each stimulus will be displayed for in seconds. For the brightness level example the stimulus will be presented for 200 ms. For the direction stimulus, it will be presented for 1 s.
* Rest Time: This is the length of time between stimuli, during which the “Blank” or “Control” stimulus will be presented. In each example, each stimulus will be followed by a 5 second rest.
* Background: This is the brightness of the background. 1 = white, 0 = black. In between are different shades of grey.
* Contrast: This defines the contrast between the background (see *Background*) and the stimuli. If contrast is 1, and the background is 1, the background will be white and the stimuli will be black. Alternatively if the background is 0, the background will be black and the stimuli will be white. We find black stimuli on a white background elicit the strongest responses. If the background is less than 0.5, the stimulus will be *Contrast* brighter than the background. Otherwise the stimulus will be *Contrast* darker than the background. E.g. if the background is 0.5 and the contrast is 0.5, the stimuli will be white on a grey background. If the background is 0.49 and the contrast is 0.49, the stimuli will be black on a grey background.

### Execution Overview

The parameter *Repetitions* defines the number of repetitions of each stimulus (I draw the distinction between different *stimuli*, e.g. circles of different radii, and different *stimulus types,* which are radii, random squares, spatial frequency, etc.)

On the computer connected to the microscope, open ThorSync and ThorImage. Make sure that ThorImage has trigger input enabled and is connected to ThorSync. You will also have connect ThorSync to ThorImage. There are boxes for this in each program respectively. Also ensure raw data capture is enabled. Under the cancel button on the stimulus generation GUI, there is the run time of the stimuli. Adjust the number of frames collected accordingly.

Once all the parameters are appropriately chosen and the stimuli are generated, a random permutation of the stimuli is generated.

Next a five volt, one second trigger is sent to ThorSync through the ao1 port of the [National Instruments device USB 6009](http://sine.ni.com/psp/app/doc/p/id/psp-115/lang/en), initiating image capture.

The permutation of the stimuli is then presented. By the end of the experiment each stimulus, including a control stimulus, will have been presented *n* times, where *n* is the number of repetitions.

After all the stimuli are presented, two text files are created*. StimulusData.txt* and *StimulusConfig.txt*. The former contains the post-trigger time in seconds of each stimulus as well as an identifying stimulus number. The latter contains the parameters used in stimulus presentation.

### Notes and Troubleshooting

Sometimes the NI device is not found, or the trigger can’t be sent. We’re still not sure why this happens sometime, but sometimes restarting MATLAB with the device already plugged in solves the problem. Otherwise, try restarting your computer.

If you click on the RunExperiment GUI while the stimuli are being presented, the stimuli may start being presented on the GUI. So don’t click on the GUI while the experiment is running.

If your selected folder doesn’t have write permission enabled, your stimulus data won’t save. The data will still be printed on the screen, so you can copy and paste it into a text file.

You can edit *RunExperiment.fig* with MATLAB’s GUIDE (GUI Development Environment,) which is helpful to change the default values for the stimulus parameters. You can also defined them in the code *RunExperiment.m.* Otherwise, update the stimulus parameters each time you launch RunExperiment*.*

You should save the stimulus files in a folder with the same name as the one in which you save the image data. This will allow the automatic merging of stimulus data and image data into a single folder later.

## Data Analysis

You should have the following files in a single folder:

* + *Episode001.h5*
  + *Experiment.xml*
  + *Image\_0001\_0001.raw*
  + *StimulusConfig.txt*
  + *StimulusTimes.txt*
  + *ThorRealTimeDataSettings.xml*

Run extractData(**Folder**), where **Folder** is a string containing the path to the folder in which all your data for the experiment is stored. This will generate two files:

“(**Folder**).mat”

“Analysed (**Folder**).mat”

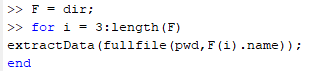
The first contains the raw pixel brightnesses for each identified ROI, the second contains the analysed data, i.e. computes and calculates responses to each stimulus as average as well as Z-Score as compared to the ROI’s response to the blank stimulus.

After running extractData, and you have all your experiments of the same stimulus type compiled in a folder, you can run the function *sampleData.m.* This will create a file *Sample.mat* in the folder which contains a sample for *n* cells from each fish, where *n* is the number of cells in the fish with the fewest number of cells. This ensures no one fish is weighted more than others in a pooled analysis.

### Notes and Troubleshooting

“(**Folder**).mat” contains the structures **header** and **ImageData**. Extracting the data from the large image file takes up the majority of the analysis time. If you wish to change some things in the analysis part of the function, you can also run extractData(**header**,**ImageData**) and only the analysis part of the function will run.

The way the hard drive is set up, there is a folder for each stimulus type as well as a folder for new data. I recommend putting all data into the new data folder and running this script:



*pwd* outputs your current folder. Navigate to the new data folder in MATLAB, run this script, and the function will run on every subfolder. Then you can move the folders, containing the analysed data, into the appropriate stimulus-type folder.

## Analysed Data File Overview

Description of the variables contained in the data files generated by functions extractData and sampleData.

### Analysed (**Folder**) .mat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| header. |  |  |  |  |
|  | FileName | | | The name you gave the experiment |
|  | RoiCount | | | Number of ROI |
|  | TimeLapse | | | Total experiment run time |
|  | FPS  Frames | | | Capture speed |
|  | Number of frames recorded |
|  | Slices | | | Number of slices recorded |
|  | ImageWidth | | | Image width (pixels) |
|  | ImageHeight | | | Image height (pixels) |
|  | fieldSize | | | Length of image field in microns |
|  | zScale | | | Distance between Z Slices in microns |
|  | zStart | | | Depth of first Z Slice |
|  | FlyBackFrames | | | Number of fly-back frames |
|  | RoiMask | | | 1 x (Slices) cell array. Each cell contains a p x 2 cell array where p is the number of ROIs in that slice. The cells in the first column contains the x coordinates and the second column contains the y coordinates for each ROI |
|  |  |  |  |  |
| AnalysedData. | |  |  |  |
|  | Times | | | Time of each frame for each ROI. N x T matrix |
|  | dFF0 | | | each ROI at each frame. N x T matrix |
|  | RoiCoords | | | Coordinates of each ROI. 3 x N matrix |
|  | Responses | | | RoiData(n).XCor data (see below) for each ROI, averaged over each repetition of the stimulus. N x (StimuliCount) matrix |
|  | ZScore | | | ZScore for each ROI and each Stimulus. N x (StimuliCount – 1) matrix. |
|  |  |  |  |  |
| RoiData(n). | | | |  |
|  | Brightness | | | Brightness profile of Roi n |
|  | Coordinates | | | Coordinates of Roi n |
|  | XCor | | | Stimulus Count by Repetition Count matrix containing the average for the 2 seconds following each stimulus presentation. |
|  |  | | |  |
| StimulusData. | |  |  |  |
|  | Raw | | | 3 x n matrix. First column is simply the count 1 through n. The second column is the time of that stimulus. The third column is a description of the stimulus, which is different for each stimulus type. |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  | Times | | | A copy of the second column of the raw data |
|  | Configuration. | | |  |
|  |  | StimuluiCount | | Number of different stimuli presented |
|  |  | Repetitions | | Number of repetitions |
|  |  | Type | | Stimulus type. Use stimType.m to decode |
|  |  | DisplayLength | | Length of stimulus |
|  |  | RestLength | | Pause between stimuli |
|  |  | PlusMinus | | Amplitude of up-down from grey background |
|  |  | Number | | Number of stimuli per repetition |
|  |  | Height | | Height of display window |
|  |  | Width | | Width of display window |
|  |  | BottomPad | | Verticle offset of display area |
|  |  | Area | | Size of display area |
|  |  | Background | | Shade of background between 0 and 1 |

### Sampled.mat

|  |  |
| --- | --- |
| RoiMin | Number of ROIs in fish with fewest ROIS |
| Responses | RoiMin x StimuliCount x FishCount matrix of data |
| SI | RoiMin x FishCount matrix of selectivity index |
| XCor | RoiMin x StimuliCount x Repitition x FishCount matrix of data |
| Zscore | RoiMin x StimuliCount-1 x FishCount matrix of Z-Score Data |

## Spatial Frequency Stimulus Information

Full Screen Width: 11.5 cm

Screen Pixel Width: 800 px

Visible Width: 6.5 cm

Visible Height: 4.0 cm

Distance from Screen: 2.3 cm

|  |  |  |
| --- | --- | --- |
|  | Cycles Width (Pixels) | Aprx. Spatial Frequency\* (Cycles/Degree) |
| Stimulus 1 | 800 | 0.0073 |
| Stimulus 2 | 400 | 0.0147 |
| Stimulus 3 | 200 | 0.0293 |
| Stimulus 4 | 160 | 0.0367 |
| Stimulus 5 | 100 | 0.0587 |
| Stimulus 6 | 80 | 0.0733 |
| Stimulus 7 | 50 | 0.1173 |
| Stimulus 8 | 40 | 0.1466 |
| Stimulus 9 | 32 | 0.1833 |
| Stimulus 10 | 25 | 0.2346 |
| Stimulus 11 | 20 | 0.2933 |
| Stimulus 12 | 16 | 0.3666 |
| Stimulus 13 | 10 | 0.5865 |
| Stimulus 14 | 8 | 0.7332 |
| Stimulus 15 | 5 | 1.173 |
| Stimulus 16 | 4 | 1.4663 |

\*Does not include distortion due to flat screen. Naively calculated as:

T = compFact(800)/(180/pi\*2\*atan(11.5/(2\*2.3)));

T = T(1:16);

compFact(n)

Input: Integer

Output: All factors of n

1. Analysis indicates we shouldn’t be showing a period of 4 either, but for now this frequency is still included [↑](#footnote-ref-1)
2. Perhaps this should be changed to a uniformly grey screen? Depends on what exactly you want to control for. [↑](#footnote-ref-2)