09.14.16

Tested WhiteNoise\_1D\_fullfieldflash.m on the photodiode. Frames are added or dropped occasionally. It may be a Psychtoolbox problem (Screen(‘Flip’)) or a LightCrafter problem (can’t handle fast stimuli stably) To test a simple stimulus, I made WhiteNoise\_1D\_fullfieldflash\_testflip.m which just flips at the contrast update rate (duration) and alternates between black and white contrast on each flip. Even for this stimulus, I noticed that frames were dropped or added every 5-10 seconds.

Is this a serious problem? Maybe not. The original DLP most likely dropped/added frames as well. As long as we have the photodiode readout, we can address this issue in the analysis.

09.09.16

Testing to do

1. Start of recording:

* Michelle found that upon hitting “start record” imaging started before the user started the stimulus. Check to see if this is an issue because use initiates “start record” before playStimMain starts running. If so, make the necessary changes.
  + Not an issue as of 09/10/16

1. Make some “edge case” .txt files for the white noise and impulse stimuli
2. Why do the stimuli run indefinitely?

Why does whitenoise stimulus run forever instead of for stimDuration in testStimDisplay.m?

* 1. Because stimDUration is only used for array allocation so far, not for stopping the while loop – need to edit

1. Issues with Github syncing? All is up to date so far
2. L2 GCamp6f 2sFFF -test both fly mount orientations

08.31.16

To Do

1. Analyze the timing data collected from Hyd1. Imaging at 80Hz (“minimize”). FOV 200x20. Frames = 3000. Used green LED at current = 0.
   1. Start at frame-resolution: average all the pixel intensities in one frame and plot over time
   2. Line scan resolution: use Y axis as the “time” axis, average over all the pixels in each line and plot the intensity over each line. Not that between frames, it takes about 4ms to start the new frame. The result should look very similar to the photodiode output except shifted at a constant due to some delay in sending the frame trigger
2. Test the programmed stimulus vs. photodiode output for WhiteNoise\_1D and ShortFlashOntoGray,
3. Code fixes
   1. Add an option to choose whether the random number generator produces the same sequence or different sequence – user chooses 0 or 1 corresponding to rng(‘default’) or rng(‘shuffle’);
   2. Saving data
      1. save data even if the user stops via keyboard press
      2. get the right directory before asking the user to name the folder to save the data to. Also make sure to cd to the right directory at the start of playStimMain.
      3. How to save rawStim() and rndSeed?
   3. How to stop the stimulus? There be an option to do both – automatically stop vs. user stop
   4. Integrate Helen’s 3D perspective code
   5. Add Marion’s image alignment code
4. Analysis code!

**Testing new stimuli**

* Compare the values between the input (e.g. sequence of random contrast values) and the photodiode output - what the computer thinks it presented and what the LC actually presents may be different
* Test how robust the presentation is at very fast update rates (e.g. 10ms, 25ms)
* Test how stable the stimulus is after running for very long time (e.g. 20 min)

**Bug in stimuli**

My ShortFlashOntoGray stimulus doesn’t seem to be able to present gray on the lightcrafter. Helen found two issues:

1) the monitor was not displaying color contrasts correctly because I set the monitor and LC resolution out of order. When we present the color gradient, the contrasts are out of order.

2) When I set the contrast values of the stimulus, I assumed that the contrast spectrum was between 0 and 1 with 0.5 being some shade of grey. However, this is only true if the bit-depth is 8, which makes the spectrum span from 0 to 255. For the LC, we are using 6-bits to encode contrast, in which the maximum contrast (white) = 63/255 (spectrum is from 0/255 to 63/255). So to convert 8-bit to 6-bit contrast values, we must multiply the contrast value by 63/255. I’ve added this detail to the WhiteNoise\_1D and ShortFlashOntoGray stimuli.

08.24.16

To do

1. Warping - search stimulus

2. Set up stimulus computer - install software

3. Send triggers to and receive from microscope (Thurs morning)

4. Modify fullfieldflash to account for screen and photodiode - see Baccus lab code to start

5. gamma correction, rescale for 6-bit

6. GUI for user to select txt files from a drop-down list

08.23.16

**Question for Helen:** Why does the photodiode voltage look different between one scenario where PsychtoolboxDefaultSetup is only called once vs. scenario where it is called every time main script runs? (see saved plots)

**Bug**: Tried running playStimMain. Multiple times in a row. If I run “clear all” before each trial, the NIDAQ is able to acquire data from the photodiode each time. However, if instead of “clear all” I run just “clear” or “clearvars”, the NIDAQ doesn’t save anything in pdData or pdTime after I run the program again. Why is this the case?

I suspect that the NIDAQ with clear or clearvars, something is not being refreshed or reset, preventing the NIDAQ from saving more data.

Seems like clear affects only the main script but doesn’t affect the workspace of functions, whereas clear all wipes everything from both main script and functions. So I put “clear all” in the function initNiDAQ(). Doing this refreshes the memory every time the function is called, so there is no interference between sessions. It works – photodiode data is saved.

So for now, we are going to reinitialize the NIDAQ each time we run the main script playStimMain.m

**Code architecture update:**

Made a subclass version of FullFieldFlash called FullFieldFlashSubclass under the superclass Stimulus. Didn’t have to change anything in the main script playStimMain.m – all the function calls (getStim(), s.setParams() et al. are the same! It works!

08.20.16

Pause(0) is super important – without it, the NIDAQ won’t save the photodiode data

08.19.16

To Do Next

1. Test playStimMain.m on the lightcrafter
2. Make the Stimulus superclass

Things to think about

* How are we going to put the NIDAQ and Psychtoolbox initialization outside of playStimMain.m?
* Need Helen’s input on the current organization of playStimMain.m

08.15.16

To do

* Create a Stimulus object with generic structure – param – and file reading function and stimulus display function
  + Function that outputs information about the particular stimulus played (e.g. seed for rand()), out struct?
* How should we set stimulus duration?
  + Probably make it a parameter in the txt file
* GUI to select .txt file from Drop-down menu (see maybe uiget())
* How can we speed up the NIDAQ initialization and PsychToolbox?
  + Separate function for initializing NiDAQ- run them once? And a more separate acquisition function
  + Psychtoolbox initialization outside of main function
  + 08/24 update: initializing NIDAQ and Psychtoolbox once outside of the main script doesn’t seem to work (photodiode output is not saving, output of photo diode looks weird when Psychtoolbox is initialized outside of main script). So we are reinitializing every time the main script runs.
* GitHub everything
* Split output of lightcrafter to send different images to fiber optic cable and photodiode
  + Note: lightcrafter dimensions are 1140 (short) x 912 (long), so a square stimulus is actually 2:1 ratio for vertical:horizontal
* Trigger inputs/outputs – how do we set them up?
* Creating actual stimulus functions
* Generalizable function to correct for different distances between edge vs. center of screen from the fly

Tests

* Intensity measurements of lightcrafter output
* Gamma-correcting the projector: scale from dark to light is nonlinear, so make it linear
  + Account for 6-bits
* Wavelength measurements