April 12 – fixed bug in binning – DataArray.cpp lines 177 and 230

// int raw\_diode = bin\_diode \* digital\_binning; this version is incorrect

int raw\_diode = (bin\_diode / row\_length)\*digital\_binning\*m\_raw\_width + (bin\_diode%row\_length)\*digital\_binning;

-7 removed from NumPts in UserInterfaceMethods.ccp line 168 and TraceWindow.ccp line 604

Removed clumsy effort to correct odd binning artifact at start of traces.

ArrayWindow yscale ArrayWindow.ccp lines 49, 50 UI.ccp lines 2235,2245, UI.fl line 324, UIMethods line 36

April 18 – sending configuration to camera.

Camera.cpp lines 160-167. Previous version did not use @PRG. Not clear why @RCL was used.

Tested several combinations of @ commands in Camera::program. ***See later entries ~May 15***

April 23 – successful incorporation of initcam in DapController line 223 and MainController line 90.

sprintf(command1, "c:\\EDT\\pdv\\initcam -u pdv0 -f c:\\EDT\\pdv\\camera\_config\\Red-Shirt-80x80.cfg");

system(command1);

April 28 used line 113 of MainControllerAcgui to check that rli readings are good. The 100-150 time points are used for dark frame subtraction.

Removed rligain from userinterface and code

April 29 ~~replaced RLI division with dark subtraction followed by division. rliDividing defined in data.cpp lines 537-552. Called in DataArray.cpp line 932. rliLow had to be made available in data.cpp. (May 3) found problem and modified DataArray::calRli() on lines 468-485. rliLowDouble created for dark frame~~. Unnecessary because baseline correction takes care of the subtraction.

Removed skip trials (unused and inactive) from user interface

Modifying gain control for selection 0-3 according to camera specs

May 1

Incorporated gain control and tested in Camera.cpp

May 6, 2020

~~SignalProcessor.cpp line 381. Skip first 3 points in calculation of 10 point average to subtract from trace in baseline correction with “start point” option.~~

Corrected baseline problem by skipping first data point. Start indices in for statements changed from i=0 to i=1 on lines 83 and 144 of polynomialFitting.cpp.

May 24, 2020

Revisited baseline problem and initial points to fine tune the initial point problem.

Changed PolynomialFitting.cpp to skip first 5 points – keep gap between LED and acquisition below 40 msec.

May 9, 2020

To change size of array window see lines 867 and 868 if ArrayWindow.ccp

May 13, 2020

Repurposed SaveSelected from saving the selected diodes to saving the diodes in each selected ROI. MainControllerArrayWindow.cpp lines 197-229. LoadSelected follows.

May 15, 2020

Unable to fix glitch in first 5 frames. Tried Sleep and moving serial commands but nothing worked.

Determined that when light goes on before acquisition the camera is driven up and it takes a few frames to clear. This can be minimized by shorter times between shutter-on and acquisition-start. But too short leads to LED settling during acquisition. Optimum is with 20 msec from LED on to acquisition start. Chun’s emails were helpful.

May 18

NeuroPlex starts acquisition before shutter opens to include enough good dark frames. It cuts off some frames at the beginning and keep the rest before shutter opening for dark frames (which will be averaged and saved at the end of the data. The data only keeps the part after shutter opening (also excluding the shutter wait period). So there is a lot of chopping off.

Turbo-SM has better control of timing, and if there is NI\_plugin, we cut off about 2 frames. The dark frames are acquired separately.

May 16

The order doesn’t matter much. In fact, the default SEQ is 1, and it’s a safety measure in case SEQ 0 was by accident saved.

I didn’t realize you are already using @TXC 1. Sorry. I thought you are dealing with just camera before moving to other devices like the A/D card. @TXC 1 is making the camera stop and wait for clock trigger pulses (or 0V). If there is light and camera is waiting, the photon will accumulate and the camera can be saturated and it takes a few frame to empty the charge. You will either make sure the camera is in the dark (but there will still be dark current accumulation in CMOS) during the wait, or simply cut off the first few frames.

 Email of May 14

For NeuroCCD or any camera single-edt/pdv-channel   
  
                                    pdv\_flush\_fifo(pdv\_pt);  
                                    pdv\_multibuf(pdv\_pt, 4);       //has to be 4 for 64bit. (You are missing this line)  
  
                                    pdv\_start\_images(pdv\_pt, loops);  
                                    for (int j = 0; j < loops; j++) {  
                                                image\_ptr = pdv\_wait\_image\_raw(pdv\_pt);  
                                                memcpy(data\_ptr, image\_ptr, frame\_length);  
                                                data\_ptr[0] += image\_length;  
                                    }  
The loops can just be the number of frames for a start.  
  
To reduce interrupts and ensure not to lose any frames (especially at high frame rate), we make super frames, meaning each frame will have height of orig\_height\*factor, then the loop = num\_frames/factor.num\_frames/factor.  
  
For 32bit, we used to read in another config that sets the height 10x (80x800 from 80x80). For 64bit, we do  
  
pdv\_setsize(pdv\_pt, config\_num\_col, config\_num\_row\*factor);

Email of April 21

Camera set incorporated some of Chun’s suggestions

1. Load the configuration file

sprintf(command, “c:\\EDT\\pdv\\initcam -u pdv0 -f %s", edt\_cfg\_name);  
system(command);

// this is a dos command  
2. Open the pdv device pointer and send serial command

prog = 8\*curCamGain + curConfig;  
sprintf(command,"@RCL %d",prog);  
SM\_serial\_command(command);

May 21, 2020

In the ROI selection process crashes occurred often. The crashes appear to be associated with selectedDiodesAverage commands. When currentRegionIndex<0 the vectors crash.

Preventing negative values with

if (currentRegionIndex < 0) currentRegionIndex = 0;  
on lines 697 and 743 of ArrayWindow.cpp appears to have eliminated the problem.

ArrayWindow::clearSelected (line 796) was also modified to use selectedDiodesAverage[i].clear();

May 21, 2020

Changed selection of diodes for ROI. To remove a selected diode one originally had to hold shift down. Modified diode.cpp lines 222 and 231 to make clicking without shift work. This will allow use of shift for other more elaborate selections of diodes in the future.

Select diode – Line 249 shiftdown = altdown = 0

Deselect diode – line 256 shiftdown = altdown = 0

Add diode to ROI – line 240 altdown = 1; shiftdown = 0

Remove diode – line 228 altdown = 1; shiftdown = 0

May 24, 2020

Use revealed that when trials were averaged the scale for display was much larger than for individual trials. Line 757 of DataArray.cpp was originally

double scale=1.0/(Gain\*3.2768\*numTrials);

This value for scale was passed to DataArray::get\_binned\_diode\_trace on line 172 to calculate traces. Removing numTrials from the expression for scale solved the display problem. It was checked by acquiring data with the LED pulsed on and dividing by RLI. The value was 1.04 on the individual trials and the average of 4 trials. numTrials is probably divided in void DataArray::get\_binned\_diode\_trace, which performs an average of trials.

May 25, 2020

Load previously saved ROIs now works. MainControllerArrayWindow::loadSelected line 234.

It uses ArrayWindow::setContinuous and ArrayWindow::addToSelectedList. ArrayWindow.cpp was not changed. Cout from loadSelected presents ROIs.

May 27, 2020

Fixed save left and right window feature by adding y0 += 18; to line 77 of WindowExporter.cpp. Very clumsy but effective.

May 29, 2020

Changed pow commands in MainControllerArrayWindow.cpp lines 72 and 75 to increase range in yscale of array window.

June 15, 2020

Implemented display value for ROIs. Changes in TraceWindow::DrawValue.

Created DataArray::aveROIData and DataArray::getROIAve

Key advice from Payne on use of memcpy to set values of array\_data\_ROI. See Payne on ROI\_AVE.txt

Incorporated ROI values into time course. Changes in TraceWindow::drawTimeCourse

Key change was in declaration Data \*array\_data\_ROI[50]. This allowed transfer of data\_array\_ROI individually to other objects.

Modified TraceWindow::saveTimeCourse to save ROI time course data along with diodes.

June 22, 2020

Modified ArrawWindow.cpp line 260-261 and lines 819-829 so control-right click clears only the current ROI. Must reset colors of diodes in that ROI to -1 and use .clear to empty the vector. Not sure why colors had to be reset separately because clearing all ROIs takes care of that.

Added save as jpeg button between save left window and save right window. Created MainController::exportAw\_Jpeg and found window dimensions by trial and error.

June 30, 2020

Added capability to read Neuroplex files. Changes to UI. Added MainController::openDataFileNP() which calles FileController::openNPFile. openNPFile sets NPFLAG to TRUE. FileController::loadNPRecControl loads header, then data, then dark frame. RLI calculation in NP files is based on first five frames recorded in light.

July 4, 2020

Solved problem of first ~5 frames being garbage due to camera saturation when light is on without acquisition. Added following to DapController.ccp line 171

for (int ii=0; ii<7; ii++) image = cam.wait\_image();

Throwing away first 7 frames made initial offset disappear. 6 was enough but did 7 just to give a little margin. This caused time out problem for last 6 frames. This was solved by adding 7 to numPts on lines 327 and 399. Looked for 3.5 msec delay in stimulation starts but did not notice any. Not sure why???