DEVELOPMENT & APPLICATION OF A CLOSED-LOOP CONTINUOUS OPTICAL NEURAL INTERFACE

Procedures for real-time image processing, neural signal extraction, and application to closed-loop control using wide-field Ca2+ fluorescence with awake behaving animals

Mark E Bucklin

Prospectus Committee:

Xue Han, PhD (Research Advisor, Chair)
Jerome Mertz, PhD
Jason Ritt, PhD
Ian Davison, PhD

CONTENTS

ABSTRACT

The latest generation of genetically encoded calcium sensors deliver a substantial boost in signal strength. This – combined with equally critical advances in the size, speed, and sensitivity of image sensors available in scientific cameras – enables high-throughput detection of neural activity in behaving animals using traditional wide-field fluorescence microscopy. However, the tremendous concomitant increase in data flow presents challenges to processing, analysis, and storage of captured video, and prompts a reexamination of traditional routines used to process data in neuroscience.

In this document I describe an open-source MATLAB toolbox for efficiently analyzing and visualizing large imaging data sets. The toolbox is capable of interactive or fully automated use. This software package provides a library of image pre-processing routines optimized for batch-processing of continuous functional fluorescence video, and additionally automates a fast unsupervised ROI detection and signal extraction routine. Further, I describe an extension of this toolbox that uses GPU programming to process streaming video, enabling the identification, segmentation and extraction of neural activity signals on-line.

The final component of this project is evaluation of this system in a closed-loop signal extraction and neural control setup. Using a wide-field Ca^{2+} fluorescence microscope and awake behaving mice running on adjacent spherical treadmills, I'll train a feature extractor to encode motor states from one mouse, and use the output to modulate motor control of the other mouse using optogenetics.

GLOSSARY

GECI – Geneticall Encoded Calcium Indicator

GCaMP – Fusion protein combining Green Fluorescent Protein with Calmodulin

sCMOS – Scientific Complementatry Metal Oxide

GPU – Graphics Processing Unit

SPMD – Single-Program Multiple Data

SIMD – Single-Instruction Multiple Data

PD – Parkinson's Disease

PROJECT SUMMARY

Using a wide-field fluorescence microscope with a scientific-CMOS camera, and the genetically encoded fluorescent calcium sensor GCaMP6f, we are able to record simultaneous activity in thousands of neurons in awake behaving mice, at a spatial and temporal resolution sufficient to capture rich subcellular dynamics. However, handling the mass of data generated by imaging this way puts a strain on standard data processing and management tools. In this document I propose procedures for addressing bottlenecks in currently available software.

Aim 1: Build a library of adaptable software that enables neuroscientists to acquire, process, analyze, and visualize large volumes of fluorescence imaging data from awake behaving animals.

Aim 2: Extend the software for continuous real-time processing on a GPU.

Aim 3: Detect motor states from extracted neural activity and apply to closed-loop neuromodulation.

BACKGROUND & SIGNIFICANCE OF PROPOSED RESEARCH

Optical techniques for observing neural activity have advanced recently owing to both an evolution of digital imaging technology, and the development of synthetic proteins that act as fluorescent indicators of neural activity. Image sensors, like those found in scientific-CMOS (sCMOS) cameras are larger, faster, and more sensitive than what was previously available in science-grade cameras. Meanwhile, the latest generation of Genetically Encoded Calcium Indicators (GECIs), collectively called GCaMP6, reports fluctuations in neural activation with extremely high fidelity. This combination of developments enables neuroscientists to open a wider channel to the brain than previously possible – using conventional epifluorescence microscopy techniques – enabling simultaneous recording from hundreds to thousands of neurons. Expanding the fraction of the observable neurons in an interconnected network may provide insight into mechanistic properties of neural disease, or may lead to a better understanding of neural coding. Additionally, feeding a large set of neural response information to a machine learning algorithm in a neuroprosthetic application may provide improved predictive performance, even if the exact mechanism of prediction remains difficult to discern. However, a few major challenges currently prevent realization of the potential benefits that these new technologies offer:

1. The increased size of raw data from a single imaging session can easily overwhelm the computational resources typically used to process similar but smaller sets of data.
2. The accumulation of raw data on disk over multiple imaging sessions quickly exceeds the data-storage capacity of most lab-scale servers, forcing researchers to halt data collection to process and delete, a nightmare scenario for some.
3. The experimental design and data analysis procedures that neuroscientists are familiar with applying for network activity data when there are 5 to 10 cells will produce highly biased spurious results, unless provided with many more stimulus-response repetitions, i.e. trials. The number of repeated trials sufficient for producing an accurate description of the neural response to any stimulus is on the order of 2^N, where N is the number of neurons being measured.

The objective of this project is to establish procedures that can address these challenges, then use these procedures to evaluate the effect that expanding available neural response input has on performance of a closed-loop encoder. This closed-loop encoder will attempt to predict changes in motor state of a mouse running on a ball, using sensors on the ball to train the encoder. It will then use the predicted motor state to modulate motor state in another mouse using opsins. This can be thought of as a model neuroprosthetic whos function is to overcome dysfunction caused by pathologically disconnected brain areas, such as exists in Parkinson's disease (PD). The goal will be to increase synchronization of mice beyond chance, such that they tend to run together and rest together.

Below I provide some background on the general procedure for offline video processing. I also discuss some of the issues with carrying out these procedures on a large dataset, and the variety of approaches that I and others have attempted for dealing with the issue. I then introduce the streaming approach (i.e. Aim 2), which is capable of processing video during acquisition and extracting signals directly, saving relevant signals only

and discarding or compressing the raw video. This approach relies on GPU programming, so I also provide some background on the application of graphics cards for computationally demanding tasks. Using a graphics card for programming in the MATLAB environment is also discussed.

Aim 1: Build a library of adaptable software that enables neuroscientists to acquire, process, analyze, and visualize large volumes of fluorescence imaging data from awake behaving animals.

Capturing wide-field fluorescence images at high spatial and temporal resolution enables us to measure functional dynamic changes in many cells within a large interconnected network. Extracting a measure for each cell in a way that preserves spatial and temporal continuity with uniform/unbiased sampling of the observed signal is achievable, but implementing a procedure to accomplish the task can be made difficult by a number of factors. One class of computer-vision procedure commonly applied to this task is image-segmentation (cell-segmentation in histology applications), a procedure that seeks to represent distinct objects in an image by association of each image pixel with one of any number of abstract objects, or with the background. A variety of algorithms exist for performing this operation efficiently on single images. Most methods can be extended to operate in a 3rd dimension, applied to stacks of image frames to enable tracking cells at multiple depths, or equivalently over time.

However, motion induced by physiologic changes and animal movement necessitates alignment of all frames in the sequence. Moreover, the massive fluctuations in signal intensity from individual and spatially overlapping cells can breed unstable solutions for alignment and radically complicate cell identification routines by disrupting temporal continuity. Implementing a reliable procedure for identifying and tracking the same cells in each frame throughout the sequence thus becomes non-trivial.

Procedures for Calcium Imaging

The general goal of processing image data from functional fluorescence imaging experiments is to restructure raw image data in a way that maps pixels in each image frame to distinct individual cells or subcellular components, called 'Regions-Of-Interest' (ROI). Pixel-intensity values from mapped pixels are typically then reduced by combination to single dimensional 'trace' time-series. These traces indicates the fluorescence intensity of an individual neuron over time, and the collection approximates the distinct activity of each and every neuron in the microscope's field of view. However, this task is made difficult by motion of the brain throughout the experiment, and also by the apparent overlap of cells in the image plane captured from the camera's 2-dimensional perspective. These issues can be partially mitigated with a few image pre-processing steps – alignment of images to correct for motion being the most critical. These options are described in the Methods & Approaches section below. Most software packages geared specifically toward functional imaging implement either of two basic classes of pixel->cell mapping algorithms. One approach is to use image-

segmentation routines for computer vision, which seeks to combine adjacent pixels into distinct spatially segregated regions representing objects in the image.

The other common approach is to perform an eigenvalue decomposition on the covariance matrix from a stack of image frames (also called spectral decomposition, or Principal Component Analysis, PCA), resulting in an assembly of basis vectors defining the weighting coefficients for each pixel. Multiplying the basis-vectors (i.e. "components") with all frames produces a one-dimensional trace for each component. The linear combination is similar to the weighted image-segmentation method in that it assigns fractional coefficients to pixels. However the procedure for computing the covariance matrix employed by PCA operates on as many pixels as are in the image, multiplying each with every other pixel – a problem with np^2 complexity, where p is the number of pixels in the image. I mention these issues inherent to PCA not because this project will attempt to address them, but because this project was initiated following tremendous difficulty attempting to use PCA-based cell sorting methods with large datasets.

Computer Software Environments for Image Processing

The widespread usage of MATLAB in neuroscience communities lends potential for greater usability and easier adaptation to software developed in this environment. While software development environments with a focus on "ease-of-use" have traditionally presumed crippling sacrifices to computational performance, this assumption is getting to be less accurate.

Standard programs include ImageJ, the built-in routines in MATLAB's Image Processing Toolbox, Mosaic from Inscopix, which is merely a compiled version of MATLAB routines which uses the MATLAB engine, Sci-Kits Image for Python, and a remarkable diversity of other applications. MATLAB is a commercial software development platform which is geared toward fast production and prototyping of data processing routines in a high-level programming language. It implements several core libraries (LINPACK, BLAS, etc.) that make multithreaded operations on matrix type data highly efficient. While MATLAB has traditionally been a considered the standard across neuroscience research labs, it was also well recognized that its performance was lacking for routines that aren't "vectorized", when compared to applications developed using lower-level languages like FORTRAN, C, and C++. Nevertheless, it remained in common use, and recent releases have added features that can drastically mitigate its performance issues, particularly through the development of a "Just-In-Time" compiler that automatically optimizes the deployment of computation accelerator resources for standard MATLAB functions. This feature enables code that performs repeated operations using for-loops or while-loops nearly as fast as equivalent code written in C. Additionally, code can be compiled into executable format using the Matlab Compiler toolbox, or used to generate equivalent C or C++ code using Matlab Coder.

Computational Resources for Processing Large Data Sets

Routines for extracting the activity in each cell from a collection of raw imaging data rely on an ability to simultaneous access many pixels separated over space and time (and consequently separated on disk). For long recording sessions, however, the size of the collection of stored image data grows dramatically. This

substantial increase in the size of data easily exceeds the capacity of system memory in the typical workstation computer available to researchers. Thus, performing the necessary processing routines using standard programs is often unfeasible.

Another popular approach to this challenge is the migration of processing routines to a cluster-based system. In this way image data can be distributed across many interconnected computer nodes capable of performing all locally restricted image processing procedures in parallel, then passing data to other nodes in the cluster for tasks that rely on comparisons made across time. Access to clusters capable of performing in this way has historically been restricted to those working in large universities or other large organization, and the diversity of cluster types is sizeable, with clusters often having very particular configuration requirements for implementing data processing jobs efficiently. These issues would pose some difficulty to the use and shared development of software libraries for image processing routines, although the growth of "cloud computing" services such as Amazon's EC2 and the Google Compute Engine, and also collaborative computing facilities like the Massachusetts Green High-Performance Computing Center (http://www.mghpcc.org) mitigate many of these issues. Additionally, efforts to produce a standardized interface for accessing and distributing data, and for managing computing resources across diverse computing environments have seen appreciable success. Apache's release of the open-source cluster computing framework, Hadoop, and a companion data-processing engine called Spark (http://spark.apache.org/), has encouraged a massive growth in collaborative development projects, a consequently increased the availability of robust shared libraries for data processing in a variety of applications. The Spark API can be accessed using the open-source programming Python, and also using other languages like Java, Scala, or R. One project specifically geared for image processing of neural imaging data is the Thunder library, a Spark package released by the Freeman lab and developed in collaboration with a number of other groups at Janelia farm and elsewhere.

Many applications will find the recent improvements in accessibility and standardization make cluster computing an attractive and worthwhile option for processing a very large set of reusable data. However, this strategy would impose harsh limitations for a neuroscientist with a project that is continuously generating new data, as the time required to transfer entire imaging data sets across the internet may be prohibitive. Unfortunately, storage on the cloud is not so unlimited that it can manage an accumulated collection of imaging data generated at anything near the rate that sCMOS cameras are capable of producing. This rate imbalance is a central motivating issue for Aim 2 this project, and is discussed in more detail below.

Aim 2: Extend the software for continuous real-time processing on a GPU.

The current generation of sCMOS cameras can capture full-frame resolution video at either 30 fps or 100 fps, depending on the data interface between camera and computer (USB3.0 or CameraLink). At 16-bits per pixel and 2048x2048 pixels, the maximum data rate for the USB3.0 camera is 240 MB/s. Imaging sessions typically last 30-minutes or less. However, pixels are typically binned down 2x2, and frame rate

often reduced; processing speed and storage constraints are the primary motivation for doing so. The effect of doubling resolution on processing time when using the graphics card is nearly negligible, however. By identifying ROIs online and extracting the traces of neural activity allows us to discard acquired images and instead store the traces only, or feed them into an encoder for online analysis.

Graphics Processing Units were traditionally developed for the consumer gaming market. They are optimized for the process which involves translating a continuous stream of information into a two-dimensional image format for transfer to a computer monitor. In the context of gaming, the stream of information received by a GPU describes the state of objects in a dynamic virtual environment, and is typically produced by a video game engine. These processors are highly optimized for this task. However, they are equally efficient at performing the same type of procedure in reverse – reducing a stream of images to structured streams of information about dynamic objects in the image – and thus are popular for video processing and computer vision applications.

Any GPU architecture will consist of a hierarchy of parallel processing elements. NVIDIA's CUDA architecture refers to the lowest level processing element as "CUDA Cores" and the highest level as "Symmetric Multiprocessors." Typically data is distributed across cores and multiprocessors by specifying a layout in C-code using different terminology, "threads" and "blocks." Blocks are then said to be organized in a "grid." Adapting traditional image processing or computer vision algorithms to run quickly on a GPU involves finding a way to distribute threads efficiently, ideally minimizimg communication between blocks.

MATLAB makes processing data using the GPU seemingly trivial by overloading a large number of built in functions. Performance varies, however, and often the fastest way to implement a routine is by writing a kernel-type subfunction – written as if it operates on single (scalar) elements only – that can be called on all pixels at once, or all pixel-subscripts, which the function can then use to retrieve the pixel value at the given subscript. The kernel-type function is compiled into a CUDA kernel the first time it's called, then repeated calls call the kernel directly, having minimal overhead. Calls go through the *arrayfun()* function.

Data transfers between system memory and graphics memory is often the major bottle-neck. Therefore, this operation is best performed only once. However, once data is on the GPU, many complex operations can be performed to extract information from the image, all while staying under the processing-time limit imposed by the frame-rate of the camera sending the images.

Aim 3: Detect motor states from extracted neural activity and apply to closed-loop neuromodulation.

The function of the brain is to translate/encode sensory input into neural output that actuates an effect that promotes survival of the organism or propagates to promote the survival of offspring (generation of a response). It does this by communicating input through interconnected neurons via converging and diverging connections which comprise the neural network. One way we study the brain is by testing and observing the properties of individual neurons and the response to changing conditions at the direct connections they form

with others. Another way is by observing a collection of neurons and to measure their response to variable conditions in their external environment, either by recording or stimulating variations in sensory input, or measuring an organisms physical/behavioral response.

One might presume that the expansion of information provided by being able to measure activity from a larger proportion of cells in a network would make it easier to analyze stimulus-response type experiments and gain insight about underlying functional mechanisms. Unfortunately, the correlation and information theoretic procedures traditionally used to make these associations suffer from a systematic bias that grows exponentially with the number responses considered for each stimulus (i.e. the number of cells included). The number of trials necessary for overcoming this bias gets exponentially large, though methods do exist for bias correction, such as through shuffling/resampling tests.

A systems neuroscience experiment will benefit from online feedback in one or both of two ways:

1. For an experiment that seeks to learn the neural response/pattern associated with a *specific stimulus*, it can inform the user whether the current number of trials – i.e. repeated presentations of the stimulus – will be sufficient for overcoming *limited sampling bias*. This could be done by testing pattern hypotheses online to subsets of the collected data and assessing their stability.
2. If the intention of the experiment is to study neural coding in general, for which it's sufficient to have an *arbitrary stimulus*, then online pattern recognition feedback can aid in maximizing the information in the response about that stimulus, either by directing modification of the stimulus, or directing modification of the field-of-view.

Online streamed processing, as specified by Aim 2, addresses the issues of processing and storing large data-sets directly. This would make the longer procedures necessary for sufficient learning from large networks possible. Additionally, I propose a strategy in the Aim 3 methods section by which incorporating this online processing stream into stimulus-response-type experiments could help correct *limited sampling bias*, enabling neural coding analysis in large populations of neurons (Ince 2009).

Overall, however, the third goal of this project will focus on the ability to use the expanded information made available by the first two project components to train an encoder that predicts intended motor states from one healthy mouse, and uses the predictions to direct neuromodulatory control of another mouse. This setup will simulate pathologic disconnection in a brain, and will test the ability to distinguish intention to start or stop running, and apply that in a way that performance is easily measureable.

METHODS & APPROACH

Aim 1: Build a library of adaptable software that enables neuroscientists to acquire, process, analyze, and visualize large volumes of fluorescence imaging data from awake behaving animals.

Image processing is performed offline using MATLAB software. The goal of this procedure is to reduce the raw image sequence to a collection of one-dimensional traces, where each trace indicates the fluorescence intensity of an individual neuron over time, and the collection approximates the distinct activity of each and every neuron in the microscope's field of view. We implement the process in 3 distinct stages as described below. The main novel contribution of this work is the efficient extension of segmented ROIs into the third dimension by clustering features of ROIs segmented separately in two dimensions. Online processing uses a similar approach, and the differences are explained in the next section.

Image Pre-processing: Contrast Enhancement and Motion Correction

Alignment of each frame in the image sequence with all other frames is essential to the methods we use in subsequent steps for identifying and tracking cells over time. Thus, the goal of the first stage is to correct for any misalignment caused by movement of the brain relative to the microscope and camera.

Many algorithms for estimating and correcting image displacement exist and are well described in the medical imaging literature. We elected to use phase-correlation to estimate the induced motion in each frame, as we found this method to be highly stable, moderately accurate, and (most importantly) fast, especially when implemented in the frequency domain and using a decent graphics card.

Phase-correlation estimates the mean translational displacement between two frames, one being the template or "fixed" frame, and the other being the uncorrected or "moving" frame. In the spatial domain this is accomplished by computing the normalized cross-correlation, which implies a 2-dimensional convolution of large matrices. The equivalent operation in the frequency domain is a simple scalar dot-product of the discrete Fourier transforms of each image normalized by the square of the template, followed by the inverse Fourier transform. The intermediate result is the cross-correlation (or phase-correlation) matrix, which should have a peak in its center for correctly aligned images, or a peak near the center, the offset of which indicates the mean offset between the two images. This peak can be found with subpixel precision by interpolation to give a more accurate alignment, although at some moderate expense in computation time.

For the template image we used a moving average of previously aligned frames when processing frames sequentially, or alternatively a fixed mean of randomly sampled and sequentially aligned images from the entire set when processing files in parallel. The simplest way to perform this operation is to use the built-in MATLAB function normxcorr2, which makes optimization decisions based on image size and available

hardware automatically. However, performance can be improved by tailoring the operation to your particular hardware and image size, i.e. using fft2 and ifft2 for large images and a good graphics card.

Region of Interest (ROI) identification & segmentation:

The ROI detection process used an adaptive threshold on the z-score of pixel intensity to reduce each frame to binary 1's and 0's (logical true or false). These binary frames were then processed using morphological operations to find and label connected components within each frame. For example, beginning with a z-score threshold of 1.5, all pixels that were more than 1.5 standard deviations above their mean were reduced to 1 (true), and all others reduced to 0 (false). Pixels reduced to 1 were often pixels overlying a cell that was significantly brighter during that frame due to activation of GCaMP. This initial threshold was adjusted up or down based on the number of non-zero pixels detected with each threshold. This was done to prevent spurious motion-induced shifts of the image frame from producing ROIs along high contrast borders. All morphological operations were performed using built-in MATLAB functions from the Image Processing Toolbox, which have fast parallel versions if the operation is run on a graphics card (e.g. *imclose*, *imopen*, etc.). Furthermore, the connected-component labeling and region formation operations were run using built-in MATLAB functions *bwconncomp*, and *regionprops*. Connected components were stored in a custom class and termed "single-frame ROIs," and these were then passed to the 3rd stage of processing, which merges them into a "multi-frame ROI" that represents the location and spatial distribution of each cell identified over the entire video.

Region of Interest (ROI) merging:

The standard structure of region properties output by the MATLAB function *regionprops* (Area, BoundingBox, Centroid, etc.) are mimicked in a custom class called *RegionOfInterest*, where each field of the structure becomes a property of the custom class. We add additional properties for storing state information and data associated with each ROI, along with a number of methods for comparing, merging, manipulating, and visualizing the single-frame and multi-frame ROIs. The single-frame to multi-frame ROI merging procedure is essentially a clustering process that merges single-frame ROIs together using such criteria as the proximity of their centroids, as well as proximity of their bounding-box (upper-left and lower-right corners). Performing this operation quickly was highly dependent on pre-grouping ROIs based on centroid location in overlapping blocks of the image frame, as well as grouping by size. This enabled the clustering to be performed in parallel (across CPU cores) followed by a second iteration of clustering to deal with redundancy in overlapping regions.

Visualization

Once ROIs are established, all video data is reloaded and passed to a method in the *RegionOfInterest* class that extracts the 1-dimensional trace for each ROI representing the fluorescence intensity in that region over time. The ROIs and their traces can then be interactively visualized using another method in the *RegionOfInterest* class.

The *RegionOfInterest* class defines methods for rapid spatial comparison operations which can typically be viewed as an adjacency matrix using built-in image viewing commands. Visualization of the segmented cell overlay and 1D traces can be manipulated by assigning colors, removing ROIs, hiding ROIs, and more.

Predicting Activation State & Assessing Network Activity

The continuous signal intensity signals can be reduced to binary indicators of activity for each frame. This enables simplified and fast calculation of information theory measures, such as activation probability, joint and conditional probabilities, response entropy, mutual information, etc. The conversion from continuous to binary uses several abstractions of the signal applied to a Gaussian Mixture Model (GMM). The abstractions are calculated from the following:

1. Linear least-squares fits to moving windows with variable size to find slope of the line surrounding each point.
2. Skewness and Kurtosis of finite windows surrounding each data point.
3. Temporal difference of fluorescence intensity.

The gaussian mixture model employs all measures to infer periods of reliable distinct activation of neurons.

Parallel Processing

Many built-in MATLAB functions are implemented using efficient multi-threaded procedures, and these are used to the extent that they can be. However, for procedures that must operate on data in irregular formats (i.e. any format other than N-dimensional arrays of primitive data types), one also has the option of performing explicitly defined parallel operations by distributing data across multiple parallel processes, each with their own memory space. Below I give examples of how implementing in a multi-threaded fashion can substantially boost performance, and also an example of a situation where multi-threaded operations aren't possible without explicit calls for parallel distribution.

Standard elementwise operators like *plus* (+) and *times* (.*), as well as comparison operators like *equals* (==) and *less-than* (<) will be performed efficiently using as many processing cores as available when applied to large n-dimensional arrays of the same size. However, when operand sizes differ a simple call to the built-in operation will not work. For example, if we wish to subtract the average from each pixel over time from all frames in the series we can accomplish this with a call to MATLAB's *bsxfun* function, which stands for Binary-Singleton-eXpansion-FUNction, as shown below:

```
>> Fmeansub = bsxfun( @minus, F, mean(F,3) );
```

This operation passes a function handle as the first argument (denoted by the '@' symbol) indicating the operation to perform. It then passes the entire [IxJxK] array of image data as the second argument, and it's temporal mean with size [IxJx1] is calculated once and passed as the third. The function efficiently expands the mean argument as needed for fast distribution across parallel threads.

Managing Continuity

Data such as baseline frames and frames for alignment must be passed between parallel processors to maintain continuity between data divided temporally between processors. However, the efficient application of this approach was limited by the system memory and number of cores available, and was meant to be applied in a cluster environment.

Building the set of functions for offline processing enabled application to data already gathered, and also served as a framework that informed the necessary procedures to be included in the online extension of this toolbox.

Aim 2: Extend the software for continuous real-time processing on a GPU.

The entire procedure for processing images and extracting cell signals can be performed in substantially less time than most commonly available tools using the approach described in Aim 1, particularly the methods for restricting the spatial extent of pixel-association operations, and distributing operations across parallel processing cores using a Single Program Multiple Data (SPMD) archetype. However, the total time still exceeds that of the acquisition session. Inefficiency arises from the overhead involved with distributing data and passing information between separate parallel processes. Graphics cards, however execute in what's called Single Instruction Multiple Data (SIMD) fashion, to distribute computation across the thousands of processing cores.

The processing components are implemented using the MATLAB System-Object framework, which allows for slightly faster performance through internal optimizations having to do with memory allocation. Most system objects, each representing one step in the serial processing and signal-extraction procedure, also have companion functions that implement the computation-heavy components of each algorithm using a pre-compiled CUDA kernel.

Benchmarking & General Performance

Built-in MATLAB functions that execute on the GPU can be profiled with benchmarking functions like *gputimeit()*, or with the *tic/toc* functions. When execution isn't fast enough, they need to be replaced with custom functions. The custom functions typically achieve the speed up necessary by enabling the operation to carried out on several frames at once. This reduces the over-head costs inposed for each function call by spreading it over several frames. This solution is not ideal, as it increases the latency of solutions, however does not preclude implementation in real-time system if the procedures are adapted to run on a real-time hybrid system-on-module like NVIDIA's Tegra X1, which should involve minimal effort once a standard set of successful procedures is realized. The current implementation tests the processing time of each stage of the process to ensure that the sum is less than the acquisition time for each frame dictated by the inverse of the frame-rate (30-50 milliseconds).

Buffered Operations

Combining frames for each operation can result in near linear speedup. For example, for the phase-correlation step required for motion correction, the FFT and IFFT are called on 16 image-frames at once, and the time take to accomplish is approximately the same as if the operation were called on 1 frame. This essentially leads to a 16x speedup, though the latency is also increased slightly. The best size to use is difficult to pre-determine, and typically must be measured for varying size 'chunks' using the benchmarking functions indicated above. The system objects manage the details necessary to allow buffered chunks of video to be passed to each stage without introducing artifacts at the temporal edges between chunks.

Image Pre-Processing & Motion Correction

Pre-processing is implemented as with the offline procedure, with a few changes. Images are aligned in chunks, and they are aligned sequentially to two templates. One template is the most recent stable frame from the preceding chunk. The other is a recursively temporal-low-pass filtered image that mitigates slow drifts. Aligning to the first template is usually more stable as the brightness of cells in the recent image will be more similar to those in the current chunk than will be the brightness of cells in the slow-moving average.

The displacement of each frame is found to sub-pixel precision, then used with a custom bicubic resampling kernel that replaces any pixels at the edges with images from the moving average.

Sequential Statistics

A number of statistics for each pixel are updated online and can be used for normalization and segmentation procedures later in the process. These include the minimum and maximum pixel intensity, and the first four central moments, which are easily converted to the mean, variance, skewness, and kurtosis. The formulas for making these calculations are given below, and are performed in a highly efficient manner as data are kept local to each processing core, and repeat computations are minimized [32].

```
n = n + 1;

% GET PIXEL SAMPLE
f = F(rowIdx,colIdx,k);

% PRECOMPUTE & CACHE SOME VALUES FOR SPEED
d = single(f) - m1;
dk = d/n;
dk2 = dk^2;
s = d*dk*(n-1);

% UPDATE CENTRAL MOMENTS
m1 = m1 + dk;
m4 = m4 + s*dk2*(n.^2-3*n+3) + 6*dk2*m2 - 4*dk*m3;
m3 = m3 + s*dk*(n-2) - 3*dk*m2;
m2 = m2 + s;

% UPDATE MIN & MAX
fmin = min(fmin, f);
fmax = max(fmax, f);
```

Furthermore, the value used to update each central moment at each point in time can be used as a measure of change in the distribution of each pixel caused by the current pixel intensity, as explained next.

Non-Stationarity & Differential Moments

Stationary refers to the property of a signal such that its statistics do not vary over time, i.e. its distribution is stable. Neural signals tend to specifically *not* have this property, in contrast to other measurable components such as those contributed by physiologic noise (heart-rate, respirations, etc.). Thus, by analyzing the evolution of statistical measures calculated for each pixel as frames are added in sequence gives a highly sensitive indicator of neural activity. This is done using a routine analogous to that for updating central moments given above, except the values returned are not only the updated moment, but also the updating component – essentially the partial derivative with respect to time. This is illustrated below, including the normalization functions which convert the partial-moment values to their variance, skewness, and kurtosis analogues:

```
% COMPUTE DIFFERENTIAL UPDATE TO CENTRAL MOMENTS
dm1 = dk;
m1 = m1 + dm1;
dm4 = s*dk2*(n^2-3*n+3) + 6*dk2*m2 - 4*dk*m3;
dm3 = s*dk*(n-2) - 3*dk*m2;
dm2 = s;
m2 = m2 + dm2;
% NORMALIZE BY VARIANCE & SAMPLE NUMBER -> CONVERSION TO dVar, dSkew, dKurt
dm2 = dm2/max(1,n-1);
dm3 = dm3*sqrt(max(1,n))/(m2^1.5);
dm4 = dm4*n/(m2^2);
```

These functions run on images representing the image intensity, and also on images taken from sequential differences indicating the temporal derivative of image intensity. The combination of outputs from these operations indicate both when image intensities are significantly high relative to past distribution, and also when intensities are changing significantly faster than learned from their past distribution.

Surface Classification: Peaks, Edges, Curvature

Edge-finding methods are employed for establishing boundaries between cells, and first and second-order gradients are used to compute local measures of curvature from an eigenvalue decomposition of the local Hessian matrix. I won't go into detail, as the utility of these procedure in the most recent implementation has been lost, but nevertheless, the operation is optimized and ready to be plugged back in when further development calls for better accuracy informing cell-segmentation, or when a faster or more accurate motion-correction algorithm is called for.

Online Cell Segmentation & Tracking

Cells are segmented by first running sequential statistics on the properties of identifiable regions on a pixel-wise basis. That is, as regions are identified in a method similar to that used offline in Aim 1, the region-properties are calculated (Centroid, Bounding-Box, etc.) and statistics for these properties are updated at each pixel covered by a proposed region. After sufficient evidence has gathered, Seeds are generated by finding the local peak of a seed-probability function that optimizes each pixel's proximity to a region centroid, and distance from any boundary. Regions are grown from these seed regions, and registered in a hierarchy that allows for co-labeling of cellular and sub-cellular components. Newly identified regions occur as new seeds, where as seeds overlapping with old regions are used to identify sub-regions, or to track regions over time.

Signal Extraction from Subcellular Compartments

I also have functions for the extraction of normalized Pointwise-Mutual-Information (nPMI), which can operate on a pixel-to-pixel basis or on a region-to-pixel basis. This operation accumulates mutually informative changes in all pixels in the maximal bounding-box (e.g. 64x64 pixels) surrounding each identified regions centroid. The weights given by this function can take on values between -1 and 1, and can be used to inform any reduction operations to follow. Additionally, spatial moments can indicate the subcellular distribution of activity across the identified region. In this context, the first spatial moment M_{00} indicates the mean signal intensity.

User Interface for Parameter Tuning

Some system-objects also incorporate a user interface to aid in parameter selection for tuning.

Aim 3: Detect motor states from extracted neural activity and apply to closed-loop neuromodulation.

Throughout an organism's life its brain will receive a continuous barrage of sensory input, and through increasingly complex abstractions is able to develop representations of that sensory input, and translate these into appropriate behavior. However, the ability of the brain to complete this task is sometimes disrupted in pathological conditions like Parkinson's disease, which is characterized by difficulty initiating changes in a motor state – i.e. starting or stopping an action like taking a step or reaching for some item in the environment – among other difficulties. Parkinson's disease can be treated using a rather basic neuromodulation therapy known as Deep Brain Stimulation (DBS). However the stimulation provided to basal-ganglia and lower-motor areas by DBS is static, and could benefit from information that indicates motor intentions as may be discernible from high motor areas like motor or pre-motor cortices.

Here the goal is to probe the capabilities of a neural interface being supplied with far more information than what others have demonstrated sufficient for encoding, classification, and predictive tasks. While the proposition may seem odd, learning the properties and performance such a well-informed encoder, and comparing to encoders that achieve decent performance with much less will likely provide worthwile insight.

This is especially significant owing to the difficulties conceiving of what type of pattern could emerge in such a large group of neurons and how this pattern may represent features of the animal's behavior.

I'll implement this encoder by training supervised machine-learning algorithms with measured movement of a mouse on a spherical treadmill. The mouse will have chronic imaging windows in motor cortex, and/or striatal areas, as we have already seen that patterns relating to the mouses movement on the treadmill emerge.

I will train the encoder to predict motor state by training with movement from the treadmill, and will use the predicted output to direct control of another mouse running on an adjacent treadmill. How the output will translate to control of the other mouse will require running a battery of tests that modulate the pattern and intensity of light directed to the controlled mouse's brain to establish limits of controllability. If the second step doesn't produce a suffiently reliable map of controllable states, I will evaluate performance by comparing the output of the motor-state prediction encoder to the measured output from the treadmill, once the training option has been shut off.

To perform neuromodulation with high spatiotemporal precision, optogenetic methods will be used to activate or silence specific neurons expressing rhodopsins in mice using certain colors of light. Initial trials will focus on using red-shifted proton pumps (e.g. JAWS) expressed in basal-ganglia, possibly using cre-driver lines for D1- and D2- Receptors.

PRELIMINARY RESULTS

Aim 1: Build a library of adaptable software that enables neuroscientists to acquire, process, analyze, and visualize large volumes of fluorescence imaging data from awake behaving animals.

A basic flow-chart showing the procedure for offline processing is show in **figure 1**:

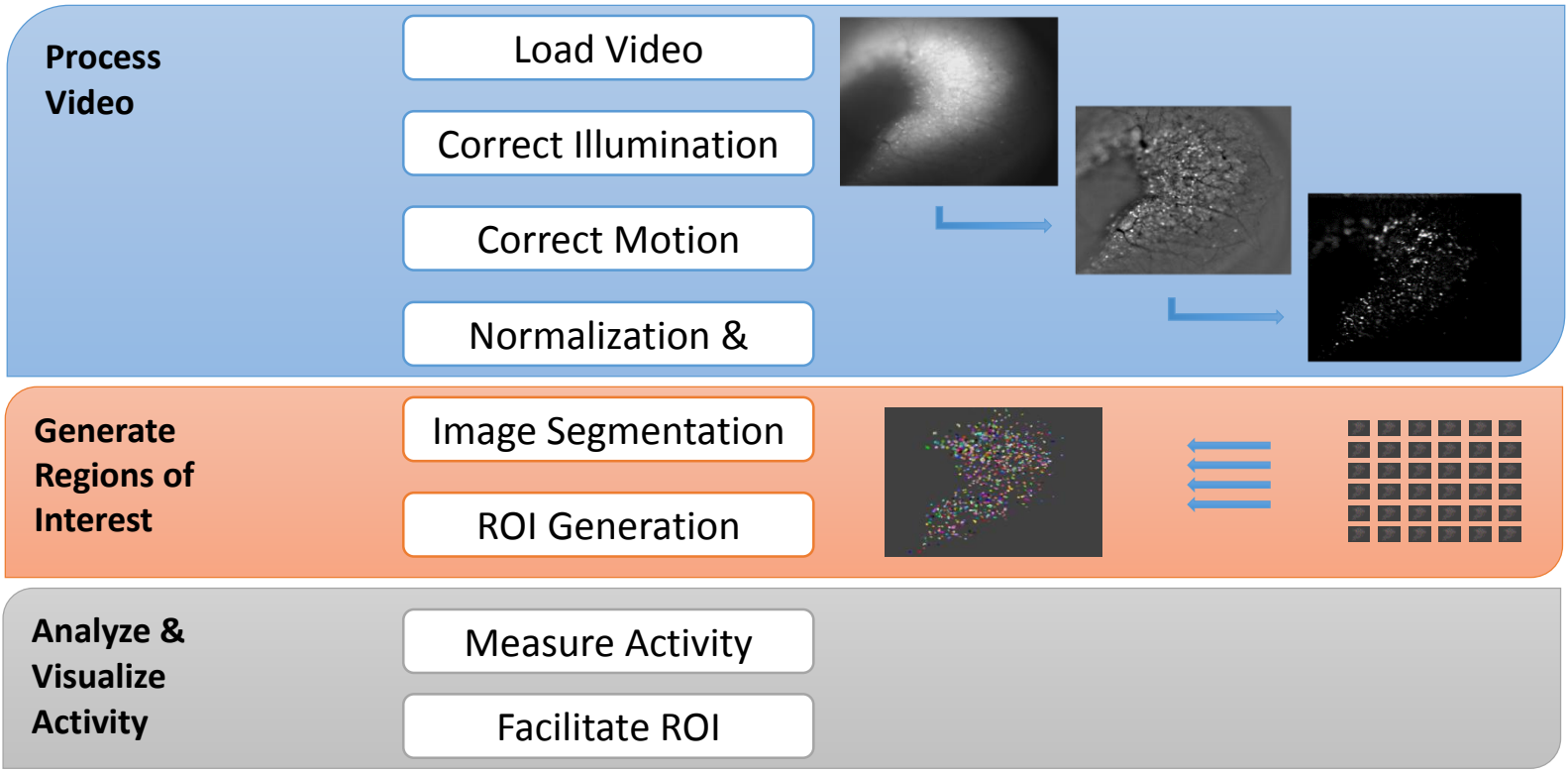
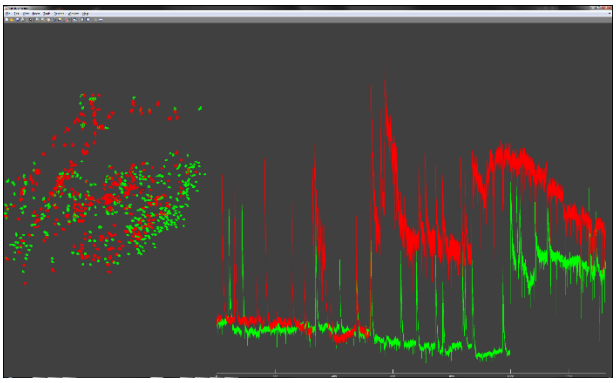


Figure 2 shows the output for interactive visualization of Regions-Of-Interest (ROIs):



Aim 2: Extend the software for continuous real-time processing on a GPU.

Shown below are screenshots from the real-time procedure as it runs, showing processing times for each aspect of the procedure, along with maps showing the non-stationarity measurements described above, where red indicates a normalized increase in the differential kurtosis of signal intensity, and blue indicates normalized increase in the skewness of the temporal derivative of signal intensity.

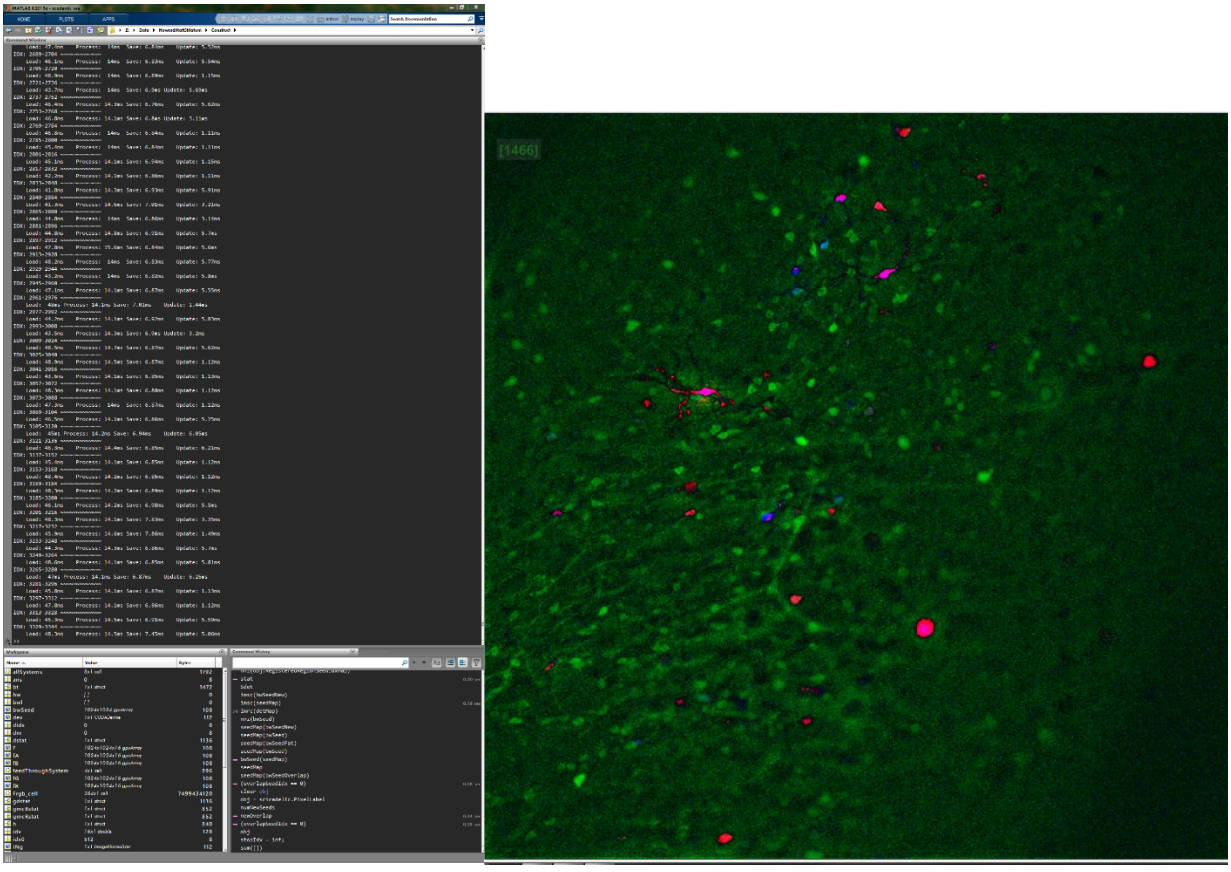


Figure 3 – Real-time processing and genereation of differential moments (measures of non-stationarity).

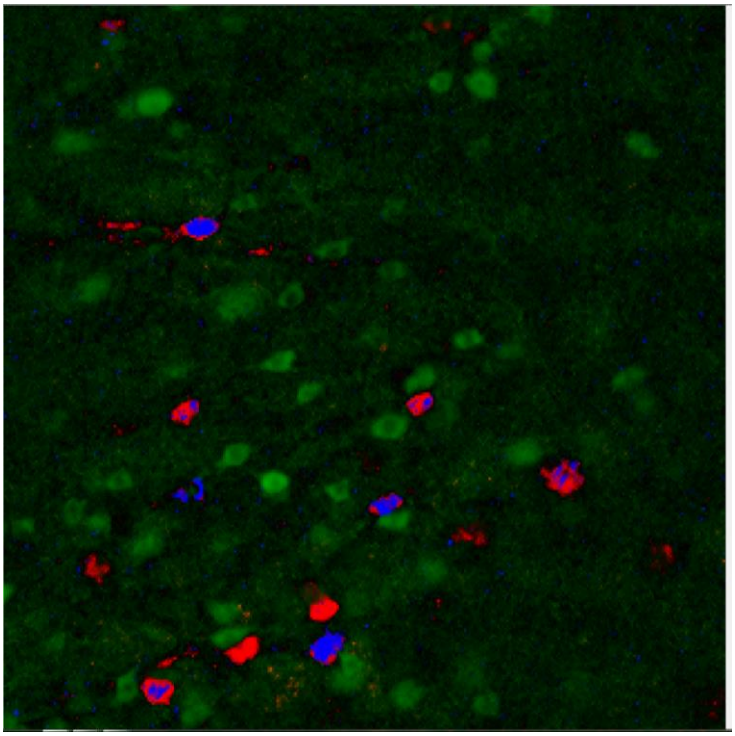


Figure 4 – activity can be reliably localized using sensitive measures of instantaneous change in distribution of the temporal derivative of each pixel (BLUE) or the signal intensity (RED). Typically changes in the signal intensity follow changes in the derivative, and blue signals are more indicative of cell activity, while the red signal intensity measure is indiciative of accumulated calcium from past activity.

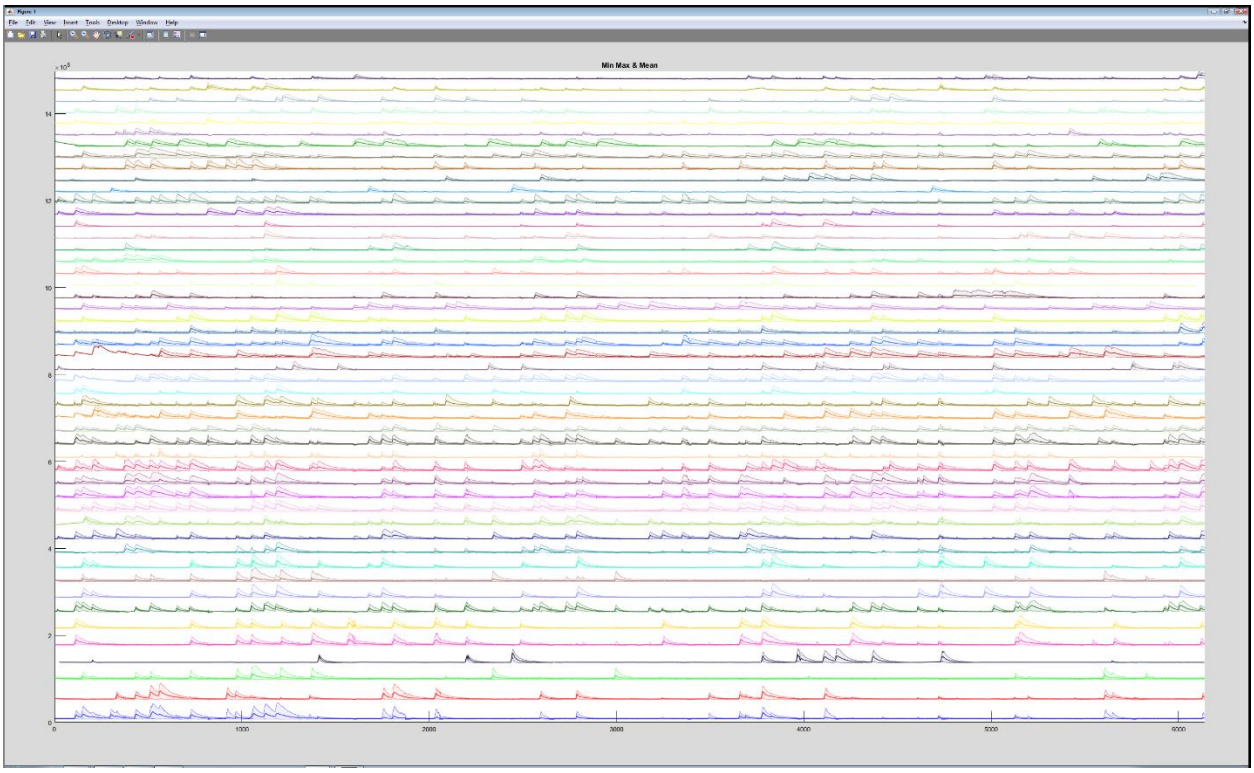


Figure 5 – Trace mean, maximum and minimum intensities can be taken extracted and displayed. However one must typically select a subset of ROIs randomly in order to see the details in any one trace.

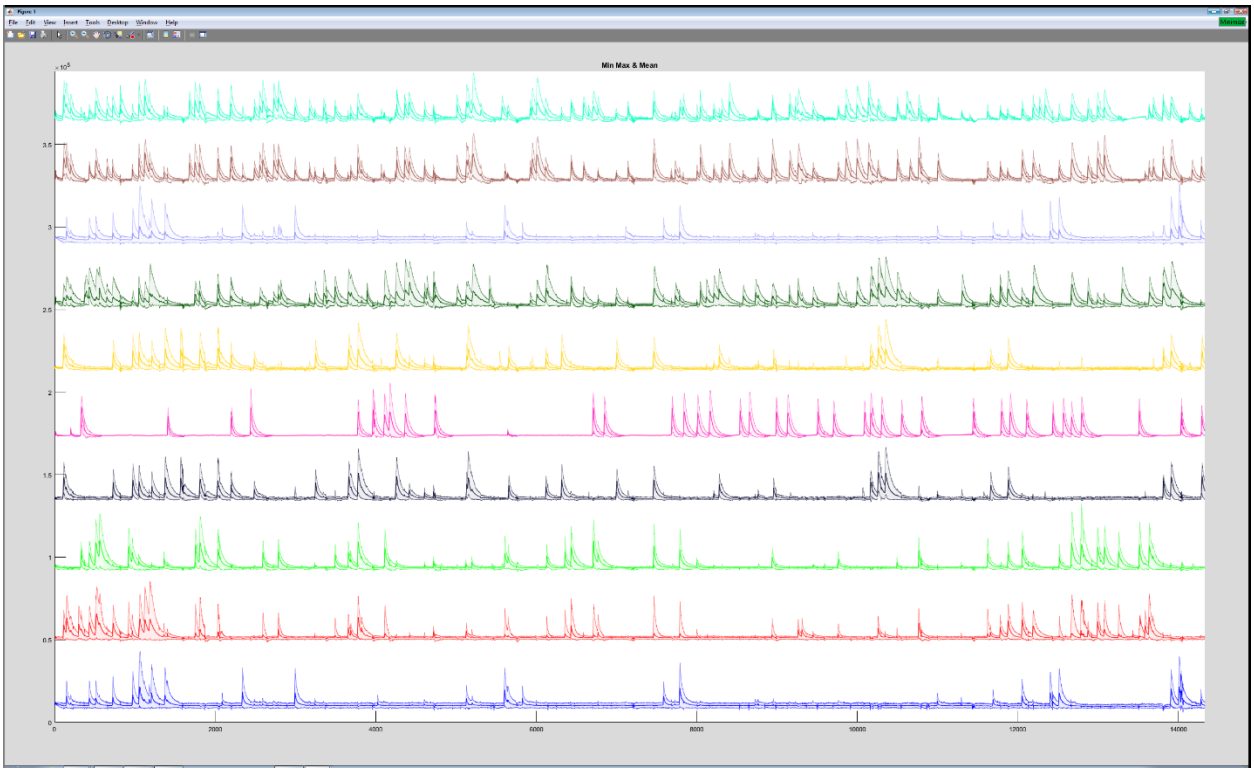


Figure 6 – A smaller subset of ROI traces.

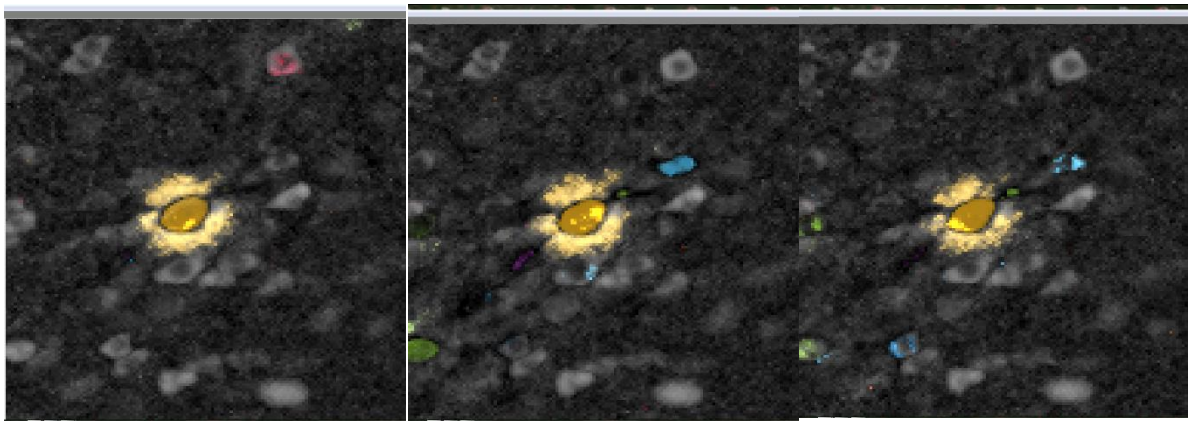


Figure 7 – Labeled overlays of activity zoomed in around a cholinergic neuron

Aim 3: Detect motor states from extracted neural activity and apply to closed-loop neuromodulation.

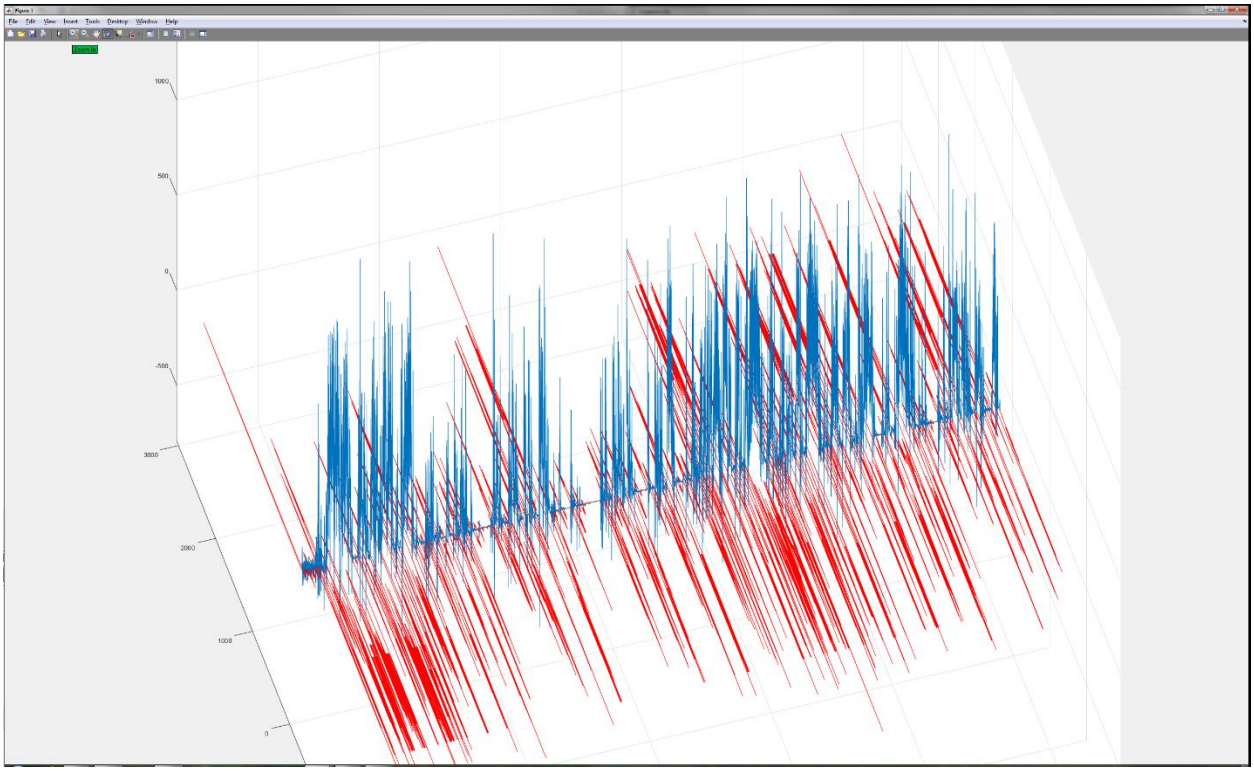


Figure 8 - Output from the spherical treadmill showing forward velocity on the y-axis (blue) and rotational velocity of the ball on the x axis (red) over time.

REFERENCES

1. Ahrens, Misha B, and Florian Engert. “Large-Scale Imaging in Small Brains.” *Current Opinion in Neurobiology* 32 (June 2015): 78–86. doi:10.1016/j.conb.2015.01.007.
2. Ahrens, Misha Benjamin, Kuo-Hua Huang, Sujatha Narayan, Brett D. Mensh, and Florian Engert. “Two-Photon Calcium Imaging during Fictive Navigation in Virtual Environments.” *Frontiers in Neural Circuits* 7 (2013): 104. doi:10.3389/fncir.2013.00104.
3. Akemann, Walther, Mari Sasaki, Hiroki Mutoh, Takeshi Imamura, Naoki Honkura, and Thomas Knöpfel. “Two-Photon Voltage Imaging Using a Genetically Encoded Voltage Indicator.” *Scientific Reports* 3 (July 19, 2013). doi:10.1038/srep02231.
4. Akerboom, Jasper, Tsai-Wen Chen, Trevor J. Wardill, Lin Tian, Jonathan S. Marvin, Sevinç Mutlu, Nicole Carreras Calderón, et al. “Optimization of a GCaMP Calcium Indicator for Neural Activity Imaging.” *The Journal of Neuroscience* 32, no. 40 (October 3, 2012): 13819–40. doi:10.1523/JNEUROSCI.2601-12.2012.
5. Amat, Fernando, William Lemon, Daniel P. Mossing, Katie McDole, Yinan Wan, Kristin Branson, Eugene W. Myers, and Philipp J. Keller. “Fast, Accurate Reconstruction of Cell Lineages from Large-Scale Fluorescence Microscopy Data.” *Nature Methods* 11, no. 9 (September 2014): 951–58. doi:10.1038/nmeth.3036.
6. ———. “Fast, Accurate Reconstruction of Cell Lineages from Large-Scale Fluorescence Microscopy Data.” *Nature Methods* 11, no. 9 (September 2014): 951–58. doi:10.1038/nmeth.3036.
7. “arXiv:nlin/0001042v1 [nlin.CD] 19 Jan 2000 - 0001042.pdf.” Accessed March 26, 2015. http://arxiv.org/pdf/nlin/0001042.pdf?origin=publication_detail.
8. Assini, Robert, Yevgeniy B. Sirotin, and Diego A. Laplagne. “Rapid Triggering of Vocalizations Following Social Interactions.” *Current Biology* 23, no. 22 (November 18, 2013): R996–97. doi:10.1016/j.cub.2013.10.007.
9. “A Threshold Selection Method from Gray-Level Histograms.” *IEEE Transactions on Systems, Man and Cybernetics* 9, no. 1 (January 1979): 62–66. doi:10.1109/TSMC.1979.4310076.
10. Bagdasaryan, Juliana, and Michel LE VAN QUYEN. “Experiencing Your Brain: Neurofeedback as a New Bridge between Neuroscience and Phenomenology.” *Frontiers in Human Neuroscience* 7 (2013): 680. doi:10.3389/fnhum.2013.00680.
11. Bale, Michael R., Robin A. A. Ince, Greta Santagata, and Rasmus S. Petersen. “Efficient Population Coding of Naturalistic Whisker Motion in the Ventro-Posterior Medial Thalamus Based on Precise Spike Timing.” *Frontiers in Neural Circuits*, 2015, 50. doi:10.3389/fncir.2015.00050.
12. Bayati, Mehdi, Alireza Valizadeh, Abdolhossein Abbassian, and Sen Cheng. “Self-Organization of Synchronous Activity Propagation in Neuronal Networks Driven by Local Excitation.” *Frontiers in Computational Neuroscience* 9 (2015): 69. doi:10.3389/fncom.2015.00069.
13. Bell, A. J., and T. J. Sejnowski. “An Information-Maximization Approach to Blind Separation and Blind Deconvolution.” *Neural Computation* 7, no. 6 (November 1995): 1129–59.

14. Bezanson, Jeff, Stefan Karpinski, Viral B. Shah, and Alan Edelman. “Julia: A Fast Dynamic Language for Technical Computing.” *arXiv:1209.5145 [cs]*, September 23, 2012. http://arxiv.org/abs/1209.5145.
15. Boba, Patrick, Dominik Bollmann, Daniel Schoepe, Nora Wester, Jan Wiesel, and Kay Hamacher. “Efficient Computation and Statistical Assessment of Transfer Entropy.” *Computational Physics* 3 (2015): 10. doi:10.3389/fphy.2015.00010.
16. ———. “Efficient Computation and Statistical Assessment of Transfer Entropy.” *Computational Physics* 3 (2015): 10. doi:10.3389/fphy.2015.00010.
17. “Brain Temperature and Hippocampal Function - Andersen - 2004 - Hippocampus - Wiley Online Library.” Accessed December 21, 2015. http://onlinelibrary.wiley.com/doi/10.1002/hipo.450050602/abstract;jsessionid=F6A7C4C6A6BDE7A6315EEB47B80F0B38.f01t01.
18. Broxton, Michael, Logan Grosenick, Samuel Yang, Noy Cohen, Aaron Andalman, Karl Deisseroth, and Marc Levoy. “Wave Optics Theory and 3-D Deconvolution for the Light Field Microscope.” *Optics Express* 21, no. 21 (October 21, 2013): 25418. doi:10.1364/OE.21.025418.
19. Carhart-Harris, Robin Lester, Robert Leech, Peter John Hellyer, Murray Shanahan, Amanda Feilding, Enzo Tagliazucchi, Dante R. Chialvo, and David Nutt. “The Entropic Brain: A Theory of Conscious States Informed by Neuroimaging Research with Psychedelic Drugs.” *Frontiers in Human Neuroscience* 8 (2014): 20. doi:10.3389/fnhum.2014.00020.
20. Carignan, Charles S, and Yukako Yagi. “Optical Endomicroscopy and the Road to Real-Time, in Vivo Pathology: Present and Future.” *Diagnostic Pathology* 7 (2012): 98. doi:10.1186/1746-1596-7-98.
21. Center for History and New Media. “Zotero Quick Start Guide,” n.d. http://zotero.org/support/quick_start_guide.
22. Chakrapani, Lakshmi N., Bilge E. S. Akgul, Suresh Cheemalavagu, Pinar Korkmaz, Krishna V. Palem, and Balasubramanian Seshasayee. “Ultra-Efficient (Embedded) SOC Architectures Based on Probabilistic CMOS (PCMOS) Technology.” In *Proceedings of the Conference on Design, Automation and Test in Europe: Proceedings*, 1110–15. DATE ’06. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2006. http://dl.acm.org/citation.cfm?id=1131481.1131790.
23. Chalfoun, Joe, Michael Majurski, Alden Dima, Christina Stuelten, Adele Peskin, and Mary Brady. “FogBank: A Single Cell Segmentation across Multiple Cell Lines and Image Modalities.” *BMC Bioinformatics* 15 (2014): 431. doi:10.1186/s12859-014-0431-x.
24. Chandra, S., and W.W. Hsu. “Lossless Medical Image Compression in a Block-Based Storage System.” In *Data Compression Conference (DCC), 2014*, 400–400, 2014. doi:10.1109/DCC.2014.70.
25. Chan, Lawrence Wing Chi, Bin Pang, Chi-Ren Shyu, Tao Chan, and Pek-Lan Khong. “Genetic Algorithm Supported by Graphical Processing Unit Improves the Exploration of Effective Connectivity in Functional Brain Imaging.” *Frontiers in Computational Neuroscience* 9 (2015): 50. doi:10.3389/fncom.2015.00050.
26. Chan, T. F., G. H. Golub, and R. J. LeVeque. “Updating Formulae and a Pairwise Algorithm for Computing Sample Variances.” In *COMPSTAT 1982 5th Symposium Held at Toulouse 1982*, edited by H. Caussinus, P. Ettinger, and R. Tomassone, 30–41.

Physica-Verlag HD, 1982. http://link.springer.com/chapter/10.1007/978-3-642-51461-6_3.
27. Chan, Tony F., Gene H. Golub, and Randall J. LeVeque. “Algorithms for Computing the Sample Variance: Analysis and Recommendations.” *The American Statistician* 37, no. 3 (1983): 242–47. doi:10.2307/2683386.
28. Chen, Tsai-Wen, Trevor J. Wardill, Yi Sun, Stefan R. Pulver, Sabine L. Renninger, Amy Baohan, Eric R. Schreiter, et al. “Ultrasensitive Fluorescent Proteins for Imaging Neuronal Activity.” *Nature* 499, no. 7458 (July 18, 2013): 295–300. doi:10.1038/nature12354.
29. ———. “Ultrasensitive Fluorescent Proteins for Imaging Neuronal Activity.” *Nature* 499, no. 7458 (July 18, 2013): 295–300. doi:10.1038/nature12354.
30. Coelho, Luis. “Mahotas: Open Source Software for Scriptable Computer Vision.” *Journal of Open Research Software* 1, no. 1 (July 29, 2013). doi:10.5334/jors.ac.
31. Cohen, Lior, Noa Koffman, Hanoch Meiri, Yosef Yarom, Ilan Lampl, and Adi Mizrahi. “Time-Lapse Electrical Recordings of Single Neurons from the Mouse Neocortex.” *Proceedings of the National Academy of Sciences* 110, no. 14 (April 2, 2013): 5665–70. doi:10.1073/pnas.1214434110.
32. “Computing Higher-Order Moments Online.” Accessed December 5, 2015. https://people.xiph.org/~tterribe/notes/homs.html.
33. Deisseroth, Karl, and Mark J. Schnitzer. “Engineering Approaches to Illuminating Brain Structure and Dynamics.” *Neuron* 80, no. 3 (October 30, 2013): 568–77. doi:10.1016/j.neuron.2013.10.032.
34. Diavatopoulos, Dean, James S. Doran, Andy Fodor, and David R. Peterson. “The Information Content of Implied Skewness and Kurtosis Changes Prior to Earnings Announcements for Stock and Option Returns,” December 1, 2008. doi:10.2139/ssrn.1309613.
35. Drew, Patrick J., Andy Y. Shih, Jonathan D. Driscoll, Per Magne Knutsen, Pablo Blinder, Dimitrios Davalos, Katerina Akassoglou, Philbert S. Tsai, and David Kleinfeld. “Chronic Optical Access through a Polished and Reinforced Thinned Skull.” *Nature Methods* 7, no. 12 (December 2010): 981–84. doi:10.1038/nmeth.1530.
36. Engert, Florian. “The Big Data Problem: Turning Maps into Knowledge.” *Neuron* 83, no. 6 (September 17, 2014): 1246–48. doi:10.1016/j.neuron.2014.09.008.
37. Fischer, Jörg, Tomislav Milekovic, Gerhard Schneider, and Carsten Mehring. “Low-Latency Multi-Threaded Processing of Neuronal Signals for Brain-Computer Interfaces.” *Frontiers in Neuroengineering* 7 (2014): 1. doi:10.3389/fneng.2014.00001.
38. Ganguly, Karunesh, and Jose M. Carmena. “Emergence of a Stable Cortical Map for Neuroprosthetic Control.” *PLoS Biol* 7, no. 7 (July 21, 2009): e1000153. doi:10.1371/journal.pbio.1000153.
39. Garcia, Samuel, Domenico Guarino, Florent Jaillet, Todd R. Jennings, Robert Pröpper, Philipp L. Rautenberg, Chris Rodgers, et al. “Neo: An Object Model for Handling Electrophysiology Data in Multiple Formats.” *Frontiers in Neuroinformatics* 8 (2014): 10. doi:10.3389/fninf.2014.00010.
40. Glickfeld, Lindsey L., Mark L. Andermann, Vincent Bonin, and R. Clay Reid. “Cortico-Cortical Projections in Mouse Visual Cortex Are Functionally Target Specific.” *Nature Neuroscience* 16, no. 2 (February 2013): 219–26. doi:10.1038/nn.3300.

41. Gray Roncal, William R., Dean Mark Kleissas, Joshua T. Vogelstein, Priya Manavalan, Kunal Lillaney, Michael Pekala, Randal Burns, et al. “An Automated Images-to-Graphs Framework for High Resolution Connectomics.” *Frontiers in Neuroinformatics*, 2015, 20. doi:10.3389/fninf.2015.00020.
42. Grewe, Jan, Thomas Wachtler, and Jan Benda. “A Bottom-up Approach to Data Annotation in Neurophysiology.” *Frontiers in Neuroinformatics* 5 (2011): 16. doi:10.3389/fninf.2011.00016.
43. Grynkiewicz, G., M. Poenie, and R. Y. Tsien. “A New Generation of Ca2+ Indicators with Greatly Improved Fluorescence Properties.” *Journal of Biological Chemistry* 260, no. 6 (March 25, 1985): 3440–50.
44. Guizar-Sicairos, Manuel, Samuel T. Thurman, and James R. Fienup. “Efficient Subpixel Image Registration Algorithms.” *Optics Letters* 33, no. 2 (January 15, 2008): 156. doi:10.1364/OL.33.000156.
45. ———. “Efficient Subpixel Image Registration Algorithms.” *Optics Letters* 33, no. 2 (January 15, 2008): 156. doi:10.1364/OL.33.000156.
46. Hampson, Robert E., Greg A. Gerhardt, Vasilis Marmarelis, Dong Song, Ioan Opris, Lucas Santos, Theodore W. Berger, and Sam A. Deadwyler. “Facilitation and Restoration of Cognitive Function in Primate Prefrontal Cortex by a Neuroprosthesis That Utilizes Minicolumn-Specific Neural Firing.” *Journal of Neural Engineering* 9, no. 5 (October 1, 2012): 056012. doi:10.1088/1741-2560/9/5/056012.
47. Hanke, Michael, and Yaroslav O. Halchenko. “Neuroscience Runs on GNU/Linux.” *Frontiers in Neuroinformatics*, 2011, 8. doi:10.3389/fninf.2011.00008.
48. Hanke, Michael, Yaroslav O. Halchenko, James V. Haxby, and Stefan Pollmann. “Statistical Learning Analysis in Neuroscience: Aiming for Transparency.” *Frontiers in Neuroscience* 4 (2010): 10. doi:10.3389/neuro.01.007.2010.
49. Hodneland, Erlend, Tanja Kögel, Dominik Michael Frei, Hans-Hermann Gerdes, and Arvid Lundervold. “CellSegm - a MATLAB Toolbox for High-Throughput 3D Cell Segmentation.” *Source Code for Biology and Medicine* 8 (2013): 16. doi:10.1186/1751-0473-8-16.
50. Holtmaat, Anthony, Tobias Bonhoeffer, David K. Chow, Jyoti Chuckowree, Vincenzo De Paola, Sonja B. Hofer, Mark Hübener, et al. “Long-Term, High-Resolution Imaging in the Mouse Neocortex through a Chronic Cranial Window.” *Nature Protocols* 4, no. 8 (July 2009): 1128–44. doi:10.1038/nprot.2009.89.
51. “Ilastik/pgmlink.” *GitHub*. Accessed January 24, 2016. https://github.com/ilastik/pgmlink.
52. Ince, Robin A. A., Alberto Mazzoni, Andreas Bartels, Nikos K. Logothetis, and Stefano Panzeri. “A Novel Test to Determine the Significance of Neural Selectivity to Single and Multiple Potentially Correlated Stimulus Features.” *Journal of Neuroscience Methods*, Special Issue on Computational Neuroscience, 210, no. 1 (September 15, 2012): 49–65. doi:10.1016/j.jneumeth.2011.11.013.
53. Ince, Robin A. A., Alberto Mazzoni, Rasmus S. Petersen, and Stefano Panzeri. “Open Source Tools for the Information Theoretic Analysis of Neural Data.” *Frontiers in Neuroscience* 4 (2010): 10. doi:10.3389/neuro.01.011.2010.
54. ———. “Open Source Tools for the Information Theoretic Analysis of Neural Data.” *Frontiers in Neuroscience* 4 (2010): 10. doi:10.3389/neuro.01.011.2010.

55. Ince, Robin A. A., Stefano Panzeri, and Christoph Kayser. “Neural Codes Formed by Small and Temporally Precise Populations in Auditory Cortex.” *The Journal of Neuroscience* 33, no. 46 (November 13, 2013): 18277–87. doi:10.1523/JNEUROSCI.2631-13.2013.
56. Ince, Robin A. A., Stefano Panzeri, and Simon R. Schultz. “Summary of Information Theoretic Quantities.” *arXiv:1501.01854 [q-Bio]*, 2014, 1–6. doi:10.1007/978-1-4614-7320-6_306-1.
57. Ince, Robin A. A., Rasmus S. Petersen, Daniel C. Swan, Stefano Panzeri, Robin A. A. Ince, Rasmus S. Petersen, Daniel C. Swan, and Stefano Panzeri. “Python for Information Theoretic Analysis of Neural Data.” *Frontiers in Neuroinformatics* 3 (2009): 4. doi:10.3389/neuro.11.004.2009.
58. Ince, Robin A. A., Simon R. Schultz, and Stefano Panzeri. “Estimating Information-Theoretic Quantities.” *arXiv:1501.01863 [q-Bio]*, 2014, 1–13. doi:10.1007/978-1-4614-7320-6_140-1.
59. Ince, Robin A. A., Nicola J. van Rijsbergen, Gregor Thut, Guillaume A. Rousselet, Joachim Gross, Stefano Panzeri, and Philippe G. Schyns. “Tracing the Flow of Perceptual Features in an Algorithmic Brain Network.” *Scientific Reports* 5 (December 4, 2015): 17681. doi:10.1038/srep17681.
60. Ito, Shinya, Michael E. Hansen, Randy Heiland, Andrew Lumsdaine, Alan M. Litke, and John M. Beggs. “Extending Transfer Entropy Improves Identification of Effective Connectivity in a Spiking Cortical Network Model.” Edited by Michal Zochowski. *PLoS ONE* 6, no. 11 (November 15, 2011): e27431. doi:10.1371/journal.pone.0027431.
61. Jabbour, Joey M., Meagan A. Saldua, Joel N. Bixler, and Kristen C. Maitland. “Confocal Endomicroscopy: Instrumentation and Medical Applications.” *Annals of Biomedical Engineering* 40, no. 2 (February 1, 2012): 378–97. doi:10.1007/s10439-011-0426-y.
62. Kaifosh, Patrick, Jeffrey D. Zaremba, Nathan B. Danielson, and Attila Losonczy. “SIMA: Python Software for Analysis of Dynamic Fluorescence Imaging Data.” *Frontiers in Neuroinformatics* 8 (2014): 80. doi:10.3389/fninf.2014.00080.
63. Keller, Philipp J, Misha B Ahrens, and Jeremy Freeman. “Light-Sheet Imaging for Systems Neuroscience.” *Nature Methods* 12, no. 1 (December 30, 2014): 27–29. doi:10.1038/nmeth.3214.
64. Kemere, Caleb, Gopal Santhanam, Byron M. Yu, Afsheen Afshar, Stephen I. Ryu, Teresa H. Meng, and Krishna V. Shenoy. “Detecting Neural-State Transitions Using Hidden Markov Models for Motor Cortical Prostheses.” *Journal of Neurophysiology* 100, no. 4 (October 1, 2008): 2441–52. doi:10.1152/jn.00924.2007.
65. Kerr, Jason N. D., David Greenberg, and Fritjof Helmchen. “Imaging Input and Output of Neocortical Networks in Vivo.” *Proceedings of the National Academy of Sciences of the United States of America* 102, no. 39 (September 27, 2005): 14063–68. doi:10.1073/pnas.0506029102.
66. Koucky, Michael H., and Mark C. Pierce. “Axial Response of High-Resolution Microendoscopy in Scattering Media.” *Biomedical Optics Express* 4, no. 10 (September 25, 2013): 2247–56. doi:10.1364/BOE.4.002247.
67. Krawinkel, Lutz A., Andreas K. Engel, and Friedhelm C. Hummel. “Modulating Pathological Oscillations by Rhythmic Non-Invasive Brain Stimulation—a Therapeutic Concept?” *Frontiers in Systems Neuroscience*, 2015, 33. doi:10.3389/fnsys.2015.00033.

68. Langer, Dominik, Marcel van ’t Hoff, Andreas J. Keller, Chetan Nagaraja, Oliver A. Pfäffli, Maurice Göldi, Hansjörg Kasper, and Fritjof Helmchen. “HelioScan: A Software Framework for Controlling in Vivo Microscopy Setups with High Hardware Flexibility, Functional Diversity and Extendibility.” *Journal of Neuroscience Methods* 215, no. 1 (April 30, 2013): 38–52. doi:10.1016/j.jneumeth.2013.02.006.
69. Lasso, Andras, Shachar Avni, and Gabor Fichtinger. “Targeting Error Simulator for Image-Guided Prostate Needle Placement.” *Conference Proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference* 2010 (2010): 5424–27. doi:10.1109/IEMBS.2010.5626494.
70. Lillis, Kyle P., Alfred Eng, John A. White, and Jerome Mertz. “Two-Photon Imaging of Spatially Extended Neuronal Network Dynamics with High Temporal Resolution.” *Journal of Neuroscience Methods* 172, no. 2 (July 30, 2008): 178–84. doi:10.1016/j.jneumeth.2008.04.024.
71. ———. “Two-Photon Imaging of Spatially Extended Neuronal Network Dynamics with High Temporal Resolution.” *Journal of Neuroscience Methods* 172, no. 2 (July 30, 2008): 178–84. doi:10.1016/j.jneumeth.2008.04.024.
72. Li, Nuo, Tsai-Wen Chen, Zengcai V. Guo, Charles R. Gerfen, and Karel Svoboda. “A Motor Cortex Circuit for Motor Planning and Movement.” *Nature* 519, no. 7541 (March 5, 2015): 51–56. doi:10.1038/nature14178.
73. Liston, Adrian, Anselm Enders, and Owen M. Siggs. “Unravelling the Association of Partial T-Cell Immunodeficiency and Immune Dysregulation.” *Nature Reviews Immunology* 8, no. 7 (July 2008): 545–58. doi:10.1038/nri2336.
74. Looger, Loren L, and Oliver Griesbeck. “Genetically Encoded Neural Activity Indicators.” *Current Opinion in Neurobiology*, Neurotechnology, 22, no. 1 (February 2012): 18–23. doi:10.1016/j.conb.2011.10.024.
75. ———. “Genetically Encoded Neural Activity Indicators.” *Current Opinion in Neurobiology*, Neurotechnology, 22, no. 1 (February 2012): 18–23. doi:10.1016/j.conb.2011.10.024.
76. Ludwig, George D. “The Velocity of Sound through Tissues and the Acoustic Impedance of Tissues.” *The Journal of the Acoustical Society of America* 22, no. 6 (November 1, 1950): 862–66. doi:10.1121/1.1906706.
77. Lütcke, Henry, David J. Margolis, and Fritjof Helmchen. “Steady or Changing? Long-Term Monitoring of Neuronal Population Activity.” *Trends in Neurosciences* 36, no. 7 (July 2013): 375–84. doi:10.1016/j.tins.2013.03.008.
78. Luyten, Laura, Natalie Schroyens, Dirk Hermans, and Tom Beckers. “Parameter Optimization for Automated Behavior Assessment: Plug-and-Play or Trial-and-Error?” *Frontiers in Behavioral Neuroscience* 8 (2014): 28. doi:10.3389/fnbeh.2014.00028.
79. Machado, Juliano, and Alexandre Balbinot. “Executed Movement Using EEG Signals through a Naive Bayes Classifier.” *Micromachines* 5, no. 4 (November 13, 2014): 1082–1105. doi:10.3390/mi5041082.
80. Malagarriga, Daniel, Mariano Alberto García-Vellisca, Alessandro E. P. Villa, Javier Martín Buldú, Jordi García-Ojalvo, and Antonio Javier Pons. “Synchronization-Based Computation through Networks of Coupled Oscillators.” *Frontiers in Computational Neuroscience*, 2015, 97. doi:10.3389/fncom.2015.00097.

81. McAlinden, Niall, Erdan Gu, Martin D. Dawson, Shuzo Sakata, and Keith Mathieson. “Optogenetic Activation of Neocortical Neurons in Vivo with a Sapphire-Based Micro-Scale LED Probe.” *Frontiers in Neural Circuits*, 2015, 25. doi:10.3389/fncir.2015.00025.
82. McDonnell, Mark D., and Lawrence M. Ward. “The Benefits of Noise in Neural Systems: Bridging Theory and Experiment.” *Nature Reviews Neuroscience* 12, no. 7 (July 2011): 415–26. doi:10.1038/nrn3061.
83. Modat, Marc, Gerard R. Ridgway, Pankaj Daga, M. J. Cardoso, David J. Hawkes, John Ashburner, and Sébastien Ourselin. “Log-Euclidean Free-Form Deformation,” 7962:79621Q – 79621Q – 6, 2011. doi:10.1117/12.878189.
84. Modi, M. N., A. K. Dhawale, and U. S. Bhalla. “CA1 Cell Activity Sequences Emerge after Reorganization of Network Correlation Structure during Associative Learning.” *eLife* 3, no. 0 (March 25, 2014): e01982–e01982. doi:10.7554/eLife.01982.
85. Monteforte, Michael, and Fred Wolf. “Dynamical Entropy Production in Spiking Neuron Networks in the Balanced State.” *Physical Review Letters* 105, no. 26 (December 30, 2010): 268104. doi:10.1103/PhysRevLett.105.268104.
86. Muir, Dylan Richard, and Björn Kampa. “FocusStack and StimServer: A New Open Source MATLAB Toolchain for Visual Stimulation and Analysis of Two-Photon Calcium Neuronal Imaging Data.” *Frontiers in Neuroinformatics* 8 (2015): 85. doi:10.3389/fninf.2014.00085.
87. ———. “FocusStack and StimServer: A New Open Source MATLAB Toolchain for Visual Stimulation and Analysis of Two-Photon Calcium Neuronal Imaging Data.” *Frontiers in Neuroinformatics* 8 (2015): 85. doi:10.3389/fninf.2014.00085.
88. Mukamel, Eran A., Axel Nimmerjahn, and Mark J. Schnitzer. “Automated Analysis of Cellular Signals from Large-Scale Calcium Imaging Data.” *Neuron* 63, no. 6 (September 2009): 747–60. doi:10.1016/j.neuron.2009.08.009.
89. ———. “Automated Analysis of Cellular Signals from Large-Scale Calcium Imaging Data.” *Neuron* 63, no. 6 (September 24, 2009): 747–60. doi:10.1016/j.neuron.2009.08.009.
90. ———. “Automated Analysis of Cellular Signals from Large-Scale Calcium Imaging Data.” *Neuron* 63, no. 6 (September 24, 2009): 747–60. doi:10.1016/j.neuron.2009.08.009.
91. Nan, Hao, Kevin C. Boyle, Nikhil Apte, Miaad S. Aliroteh, Anshuman Bhuyan, Amin Nikoozadeh, Butrus T. Khuri-Yakub, and Amin Arbabian. “Non-Contact Thermoacoustic Detection of Embedded Targets Using Airborne-Capacitive Micromachined Ultrasonic Transducers.” *Applied Physics Letters* 106, no. 8 (February 23, 2015): 084101. doi:10.1063/1.4909508.
92. “New Tab.” Accessed December 5, 2015. about:newtab.
93. O’Connor, Daniel H., Daniel Huber, and Karel Svoboda. “Reverse Engineering the Mouse Brain.” *Nature* 461, no. 7266 (October 15, 2009): 923–29. doi:10.1038/nature08539.
94. Oñativia, Jon, Simon R. Schultz, and Pier Luigi Dragotti. “A Finite Rate of Innovation Algorithm for Fast and Accurate Spike Detection from Two-Photon Calcium Imaging.” *Journal of Neural Engineering* 10, no. 4 (August 1, 2013): 046017. doi:10.1088/1741-2560/10/4/046017.
95. “Online Skewness Kurtosis.” Accessed December 5, 2015. http://www.johndcook.com/blog/skewness_kurtosis/.

96. Packer, Adam M., Lloyd E. Russell, Henry W. P. Dalgleish, and Michael Häusser. “Simultaneous All-Optical Manipulation and Recording of Neural Circuit Activity with Cellular Resolution in Vivo.” *Nature Methods* 12, no. 2 (February 2015): 140–46. doi:10.1038/nmeth.3217.
97. ———. “Simultaneous All-Optical Manipulation and Recording of Neural Circuit Activity with Cellular Resolution in Vivo.” *Nature Methods* 12, no. 2 (February 2015): 140–46. doi:10.1038/nmeth.3217.
98. Palmer, Tim. “Modelling: Build Imprecise Supercomputers.” *Nature* 526, no. 7571 (September 29, 2015): 32–33. doi:10.1038/526032a.
99. Palmer, Tim N., and Michael O’Shea. “Solving Difficult Problems Creatively: A Role for Energy Optimised Deterministic/stochastic Hybrid Computing.” *Frontiers in Computational Neuroscience*, 2015, 124. doi:10.3389/fncom.2015.00124.
100. Panzeri, Stefano, Riccardo Senatore, Marcelo A. Montemurro, and Rasmus S. Petersen. “Correcting for the Sampling Bias Problem in Spike Train Information Measures.” *Journal of Neurophysiology* 98, no. 3 (September 1, 2007): 1064–72. doi:10.1152/jn.00559.2007.
101. Patel, Tapan P., Karen Man, Bonnie L. Firestein, and David F. Meaney. “Automated Quantification of Neuronal Networks and Single-Cell Calcium Dynamics Using Calcium Imaging.” *Journal of Neuroscience Methods* 243 (March 30, 2015): 26–38. doi:10.1016/j.jneumeth.2015.01.020.
102. Pierce, Mark, Dihua Yu, and Rebecca Richards-Kortum. “High-Resolution Fiber-Optic Microendoscopy for in Situ Cellular Imaging.” *Journal of Visualized Experiments*, no. 47 (January 11, 2011). doi:10.3791/2306.
103. Pnevmatikakis, Eftychios A., Yuanjun Gao, Daniel Soudry, David Pfau, Clay Lacefield, Kira Poskanzer, Randy Bruno, Rafael Yuste, and Liam Paninski. “A Structured Matrix Factorization Framework for Large Scale Calcium Imaging Data Analysis.” *arXiv:1409.2903 [q-Bio, Stat]*, September 9, 2014. http://arxiv.org/abs/1409.2903.
104. Poline, Jean-Baptiste, Janis L. Breeze, Satrajit S. Ghosh, Krzysztof Gorgolewski, Yaroslav O. Halchenko, Michael Hanke, Karl G. Helmer, et al. “Data Sharing in Neuroimaging Research.” *Frontiers in Neuroinformatics* 6 (2012): 9. doi:10.3389/fninf.2012.00009.
105. Reimann, Michael W., James G. King, Eilif B. Muller, Srikanth Ramaswamy, and Henry Markram. “An Algorithm to Predict the Connectome of Neural Microcircuits.” *Frontiers in Computational Neuroscience*, 2015, 120. doi:10.3389/fncom.2015.00120.
106. Rolston, John D., Robert E. Gross, Steve M. Potter, John D. Rolston, Robert E. Gross, and Steve M. Potter. “A Low-Cost Multielectrode System for Data Acquisition Enabling Real-Time Closed-Loop Processing with Rapid Recovery from Stimulation Artifacts.” *Frontiers in Neuroengineering* 2 (2009): 12. doi:10.3389/neuro.16.012.2009.
107. Ross, James D., D. Kacy Cullen, James Patrick Harris, Michelle C. Ph D. LaPlaca, and Stephen P. DeWeerth. “A Three-Dimensional Image Processing Program for Accurate, Rapid, and Semi-Automated Segmentation of Neuronal Somata with Dense Neurite Outgrowth.” *Frontiers in Neuroanatomy* 9 (2015): 87. doi:10.3389/fnana.2015.00087.

108. ———. “A Three-Dimensional Image Processing Program for Accurate, Rapid, and Semi-Automated Segmentation of Neuronal Somata with Dense Neurite Outgrowth.” *Frontiers in Neuroanatomy* 9 (2015): 87. doi:10.3389/fnana.2015.00087.
109. Ros, Tomas, Bernard J. Baars, Ruth A. Lanius, and Patrik Vuilleumier. “Tuning Pathological Brain Oscillations with Neurofeedback: A Systems Neuroscience Framework.” *Frontiers in Human Neuroscience* 8 (2014): 1008. doi:10.3389/fnhum.2014.01008.
110. Rupprecht, P. T. R. “Beyond Correlation Analysis: Transfer Entropy.” *A Blog about Neurophysiology*. Accessed March 26, 2015. https://ptrrupprecht.wordpress.com/2014/03/19/beyond-correlation-analysis-transfer-entropy/.
111. Rutishauser, Ueli, Andreas Kotowicz, and Gilles Laurent. “A Method for Closed-Loop Presentation of Sensory Stimuli Conditional on the Internal Brain-State of Awake Animals.” *Journal of Neuroscience Methods* 215, no. 1 (April 30, 2013): 139–55. doi:10.1016/j.jneumeth.2013.02.020.
112. Sacan, Ahmet, Hakan Ferhatosmanoglu, and Huseyin Coskun. “CellTrack: An Open-Source Software for Cell Tracking and Motility Analysis.” *Bioinformatics* 24, no. 14 (July 15, 2008): 1647–49. doi:10.1093/bioinformatics/btn247.
113. ———. “CellTrack: An Open-Source Software for Cell Tracking and Motility Analysis.” *Bioinformatics* 24, no. 14 (July 15, 2008): 1647–49. doi:10.1093/bioinformatics/btn247.
114. Schultz, Simon R., Robin A. A. Ince, and Stefano Panzeri. “Applications of Information Theory to Analysis of Neural Data.” *arXiv:1501.01860 [q-Bio]*, 2014, 1–6. doi:10.1007/978-1-4614-7320-6_280-1.
115. Schwarzkopf, D. Samuel. “We Should Have Seen This Coming.” *Frontiers in Human Neuroscience* 8 (2014): 332. doi:10.3389/fnhum.2014.00332.
116. Science, Armed with. “Remote Control of Brain Activity Using Ultrasound.” *Armed with Science*. Accessed December 21, 2015. http://science.dodlive.mil/2010/09/01/remote-control-of-brain-activity-using-ultrasound/.
117. Senkov, Oleg, Andrey Mironov, and Alexander Dityatev. “A Novel Versatile Hybrid Infusion-Multielectrode Recording (HIME) System for Acute Drug Delivery and Multisite Acquisition of Neuronal Activity in Freely Moving Mice.” *Neural Technology*, 2015, 425. doi:10.3389/fnins.2015.00425.
118. Shi, J., and J. Malik. “Normalized Cuts and Image Segmentation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, no. 8 (August 2000): 888–905. doi:10.1109/34.868688.
119. Sieu, Lim-Anna, Antoine Bergel, Elodie Tiran, Thomas Deffieux, Mathieu Pernot, Jean-Luc Gennisson, Mickaël Tanter, and Ivan Cohen. “EEG and Functional Ultrasound Imaging in Mobile Rats.” *Nature Methods* advance online publication (August 3, 2015). doi:10.1038/nmeth.3506.
120. Singh, Abhinav, and Nicholas A. Lesica. “Incremental Mutual Information: A New Method for Characterizing the Strength and Dynamics of Connections in Neuronal Circuits.” *PLoS Comput Biol* 6, no. 12 (December 9, 2010): e1001035. doi:10.1371/journal.pcbi.1001035.
121. “Software | Scanbox.” Accessed August 5, 2014. http://scanbox.wordpress.com/category/software/.

122. Soriano, Miguel C., Daniel Brunner, Miguel Escalona-Morán, Claudio R. Mirasso, and Ingo Fischer. “Minimal Approach to Neuro-Inspired Information Processing.” *Frontiers in Computational Neuroscience* 9 (2015): 68. doi:10.3389/fncom.2015.00068.
123. Stegmaier, Johannes, Fernando Amat, William C. Lemon, Katie McDole, Yinan Wan, George Teodoro, Ralf Mikut, and Philipp J. Keller. “Real-Time Three-Dimensional Cell Segmentation in Large-Scale Microscopy Data of Developing Embryos.” *Developmental Cell* 36, no. 2 (January 25, 2016): 225–40. doi:10.1016/j.devcel.2015.12.028.
124. Stein, Erich W., Konstantin Maslov, and Lihong V. Wang. “Noninvasive, in Vivo Imaging of the Mouse Brain Using Photoacoustic Microscopy.” *Journal of Applied Physics* 105, no. 10 (May 15, 2009). doi:10.1063/1.3116134.
125. Stroh, Albrecht, Helmuth Adelsberger, Alexander Groh, Charlotta Rühlmann, Sebastian Fischer, Anja Schierloh, Karl Deisseroth, and Arthur Konnerth. “Making Waves: Initiation and Propagation of Corticothalamic Ca2+ Waves In Vivo.” *Neuron* 77, no. 6 (March 20, 2013): 1136–50. doi:10.1016/j.neuron.2013.01.031.
126. Sullivan, Megan R., Axel Nimmerjahn, Dmitry V. Sarkisov, Fritjof Helmchen, and Samuel S.-H. Wang. “In Vivo Calcium Imaging of Circuit Activity in Cerebellar Cortex.” *Journal of Neurophysiology* 94, no. 2 (August 1, 2005): 1636–44. doi:10.1152/jn.01013.2004.
127. Szabo, Vivien, Cathie Ventalon, Vincent De Sars, Jonathan Bradley, and Valentina Emiliani. “Spatially Selective Holographic Photoactivation and Functional Fluorescence Imaging in Freely Behaving Mice with a Fiberscope.” *Neuron* 84, no. 6 (December 17, 2014): 1157–69. doi:10.1016/j.neuron.2014.11.005.
128. Tokuda, Junichi, Gregory S. Fischer, Xenophon Papademetris, Ziv Yaniv, Luis Ibanez, Patrick Cheng, Haiying Liu, et al. “OpenIGTLink: An Open Network Protocol for Image-Guided Therapy Environment.” *The International Journal of Medical Robotics + Computer Assisted Surgery : MRCAS* 5, no. 4 (December 2009): 423–34. doi:10.1002/rcs.274.
129. Tye, Kay M., and Karl Deisseroth. “Optogenetic Investigation of Neural Circuits Underlying Brain Disease in Animal Models.” *Nature Reviews Neuroscience* 13, no. 4 (April 2012): 251–66. doi:10.1038/nrn3171.
130. “Updating Formulae and a Pairwise Algorithm for Computing Sample Variances - IP.com.” Accessed December 5, 2015. http://priorart.ip.com/IPCOM/000150677.
131. Victor, Jonathan D. “Approaches to Information-Theoretic Analysis of Neural Activity.” *Biological Theory* 1, no. 3 (2006): 302–16.
132. Vogelstein, Joshua T., Adam M. Packer, Timothy A. Machado, Tanya Sippy, Baktash Babadi, Rafael Yuste, and Liam Paninski. “Fast Nonnegative Deconvolution for Spike Train Inference From Population Calcium Imaging.” *Journal of Neurophysiology* 104, no. 6 (December 1, 2010): 3691–3704. doi:10.1152/jn.01073.2009.
133. Vogt, Nina. “Neuroscience: Injectable Meshes for Neural Recordings.” *Nature Methods* 12, no. 8 (August 2015): 702–3. doi:10.1038/nmeth.3511.
134. Wählby, C., I.-M. Sintorn, F. Erlandsson, G. Borgefors, and E. Bengtsson. “Combining Intensity, Edge and Shape Information for 2D and 3D Segmentation of Cell Nuclei in Tissue Sections.” *Journal of Microscopy* 215, no. 1 (July 1, 2004): 67–76. doi:10.1111/j.0022-2720.2004.01338.x.

135. ———. “Combining Intensity, Edge and Shape Information for 2D and 3D Segmentation of Cell Nuclei in Tissue Sections.” *Journal of Microscopy* 215, no. 1 (July 1, 2004): 67–76. doi:10.1111/j.0022-2720.2004.01338.x.
136. Warden, Melissa R., Jessica A. Cardin, and Karl Deisseroth. “Optical Neural Interfaces.” *Annual Review of Biomedical Engineering* 16, no. 1 (2014): 103–29. doi:10.1146/annurev-bioeng-071813-104733.
137. Wickham, Hadley. “A Layered Grammar of Graphics.” *Journal of Computational and Graphical Statistics* 19, no. 1 (January 2010): 3–28. doi:10.1198/jcgs.2009.07098.
138. Wilson, J. Adam, Justin C. Williams, J. Adam Wilson, and Justin C. Williams. “Massively Parallel Signal Processing Using the Graphics Processing Unit for Real-Time Brain–computer Interface Feature Extraction.” *Frontiers in Neuroengineering* 2 (2009): 11. doi:10.3389/neuro.16.011.2009.
139. ———. “Massively Parallel Signal Processing Using the Graphics Processing Unit for Real-Time Brain–computer Interface Feature Extraction.” *Frontiers in Neuroengineering* 2 (2009): 11. doi:10.3389/neuro.16.011.2009.
140. Wood, Dylan, Margaret King, Drew Landis, William Courtney, Runtang Wang, Ross Kelly, Jessica A. Turner, and Vince D. Calhoun. “Harnessing Modern Web Application Technology to Create Intuitive and Efficient Data Visualization and Sharing Tools.” *Frontiers in Neuroinformatics* 8 (2014): 71. doi:10.3389/fninf.2014.00071.
141. Wu, Fan, Eran Stark, Pei-Cheng Ku, Kensall D. Wise, György Buzsáki, and Euisik Yoon. “Monolithically Integrated μLEDs on Silicon Neural Probes for High-Resolution Optogenetic Studies in Behaving Animals.” *Neuron* 88, no. 6 (December 16, 2015): 1136–48. doi:10.1016/j.neuron.2015.10.032.
142. W&#xe4, Carolina Hlby, Joakim Lindblad, Mikael Vondrus, Ewert Bengtsson, Bj&#xf6, and Lennart Rkesten. “Algorithms for Cytoplasm Segmentation of Fluorescence Labelled Cells.” *Analytical Cellular Pathology* 24, no. 2–3 (2002): 101–11. doi:10.1155/2002/821782.
143. Xu, Ning-long, Mark T. Harnett, Stephen R. Williams, Daniel Huber, Daniel H. O’Connor, Karel Svoboda, and Jeffrey C. Magee. “Nonlinear Dendritic Integration of Sensory and Motor Input during an Active Sensing Task.” *Nature* 492, no. 7428 (December 13, 2012): 247–51. doi:10.1038/nature11601.
144. Yaroslavsky, L. P. “Compression, Restoration, Resampling, ‘compressive Sensing’: Fast Transforms in Digital Imaging.” *Journal of Optics* 17, no. 7 (July 1, 2015): 073001. doi:10.1088/2040-8978/17/7/073001.
145. Zhang, H. X., D. Massoubre, J. McKendry, Z. Gong, B. Guilhabert, C. Griffin, E. Gu, P. E. Jessop, J. M. Girkin, and M. D. Dawson. “Individually-Addressable Flip-Chip AlInGaN Micropixelated Light Emitting Diode Arrays with High Continuous and Nanosecond Output Power.” *Optics Express* 16, no. 13 (June 23, 2008): 9918. doi:10.1364/OE.16.009918.
146. Accessed July 15, 2015. about:blank.