Development & Application of a Closed-Loop Continuous Optical Neural Interface

Mark Bucklin

April 13, 2017

Abstract

The latest generation of genetically encoded sensors to emerge from molecular engineering labs are highly sensitive. These - combined with equally critical advances in the performance of affordable image sensor – have been put to use in labs conducting research neuroscience research to enable high-throughput detection of neural activity in behaving animals using both multi-photon and traditional wide-field fluorescence microscopy. Unfortunately, expanded sensing capability can generate a flow of data in proportions that challenge the standard procedures used to process, analyze, and store captured video. The torrent can easily overwhelm and debilitate, even when applying the latest and greatest from our ever-expanding arsenal of cluster computing resources. Sensing capabilities available to scientists, physicians and engineers will continue to grow exponentially, while traditional raw data storage and batch-processing routines will impose the same limits on throughput utilization.

The work presented here demonstrates the ease with which a dependable and affordable wide-field fluourescence imaging system can be assembled, and integrated with behavior control and monitoring system such as found in a typical neuroscience laboratory. Application of standard image processing and computer vision routines demonstrate the remarkable value of such a system, but also highlights the woeful inability of standard batch processing routines to manage the volume of data available. After describing a slew of marginally successful naive attempts to pre-shrink long streams of raw video data to more manageable proportions, a more likely plan is presented. Here you will find the strategic ingredients to consider if your intent is to transform an abundant flow of raw data into proportionally informative knowledge. Certainly, aggressive deployment of streamed computation on graphics processing hardware will be vital component, but not solely sufficient. A likely solution will also recognize opportunities afforded by implementing performance-tuned data structures, modular and dynamically reconfigurable data processing elements, and graph oriented stream semantics coordinating data-flow.

Forward

I have structured this document to roughly coincide with a chronological account of 6 years spent in a neuro-oriented biomedical engineering lab. My role in the lab was centered around exploratory device design and development, mostly targeting application in neuroscience research, with intended users being neuroscientist colleagues. One of the lab's most remarkable assets is the breadth and diversity of its constituents in terms of their skills and experience, both within and between the engineering/development and the science/medical sides of the lab. All efforts stood to benefit from the close proximity to skilled colleagues, most notably for the complementary guide and provide roles that assisted the development process of new devices and the experiments they were intended for.

My initial experience in optoelectronic device development was as an undergrad at Columbia University where I was advised by Elizabeth Hillman, and developed a device that combined thermography and near-infrared spectroscopy in a portable and inexpensive device intended to provide early detection of adverse neoplastic changes through at-home daily monitoring, particularly targeting use by patients with high-risk for breast cancer.

I then went to the Das Lab where I developed macroscopic imaging systems used for intrinsic imaging in the visual cortex of awake primates.

As a MD/PhD student, I attempt to maintain a potential to adapt the end-products of each development for clinical applicability.

The story presented here is rather unusual in that success precedes failure. The volume of tangible presentable results is greatest toward the beginning stages of the work described here. This unusual inversion is what make this story worth hearing, however. Thank you for taking the time to read this. I hope that at least the technical information provided herein, if not the procedural insight, is valuable in your current or future endeavors.

Microscopy

This section describes the background in microscopy in the neurosciences, and also how it relates to imaging in healthcare and electrophysiology in neuroscience. It will also describe the basic elements necessary for the construction of a microscope in a laboratory where calcium imaging in an animal is available. It will also refer to later sections which cover the design and construction of mechanical elements for animal handling and optical access (i.e. the headplate and a chronic optical window).

Filters

Lenses

Mechanics

Microscopy and Functional Imaging Two core innovations in available

- technology 1. Synthetic bio (i.e. GCaMP) 2. Cameras
- scientific CMOS

Cameras and Sensors

This section details the evolution of cameras sensors and other sensors that provide bio-relevant data. Emphasis is on

Telemetry & Control

SCADA on the Cheap

Development boards

Data Scaling

This section describes the reality of how data scales as more and more sensors are added. Typically we humans think linearly, but the ramifications of increasing something like sensor size is often exponential. Furthermore, many operations we perform have costs that scale exponentially, and in my humble opinion are not even worth attempting if any such procedure must applied continuously on a data=stream.

Image Processing

This section borrows from AIM-1 and AIM-2 of the prospectus.

Computing Power and Connectivity

- Remote Clusters (AWS)
- Graphics Processing Units (NVIDIA GTX)
- Embedded Units (NVIDIA Tegra X2) 2. Well developed libraries
- ImageJ (so so)
- OpenCV (uses OpenCL)
- GStreamer (much better)
- OpenGL
- Shader

Image Processing

- Motion Correction
- Image Enhancement

Motion Correction Two approaches to find displacement

Spatially Homogeneous phase correlation

- aka normalized cross correlation - Feature Matching
- Detect features (i.e. corners) ### Triangulate best

Image Enhancement

1. Contrast - Linear Scaling - Lookup Tables
2. Spatial and Temporal Filtering
3. Feature images - Gradients

Feature Extraction

1. Feature images (temporally independent)

- Gradients - Surface Curvature 2. Long Term Memory - Statistics - changes (single pixel)
- Mutual information changes (inter-pixel)

Acceleration and Optimization Procedures for Online Video Processing

Incremental Update of Statistics

central moments

```
function [m1,m2,m3,m4,fmin,fmax] = updateStatistics(x,m1,m2,m3,m4))
      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);
end
```

Extract Features

```
function [dm1,dm2,dm3,dm4] = getStatisticUpdate(x,m1,m2,m3,m4)
      % 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);
end
```

Simple Processing on GPU

```
[dm1,dm2,dm3,dm4] = arrayfun(@getStatisticUpdate(x,m1,m2,m3,m4)
[dm1,dm2,dm3,dm4] = arrayfun(@getStatisticUpdate(rowidx,colidx)
```

Alternative Libraries

- NVIDIA Performance Primitives
- OpenCV
- VLFeat
- OpenGL
- OpenCL
- OpenVX
- CLosedDoesNotExist (. . . ?)
- Shader Languages
  - GLSL
  - HLSL
  - WebGL
  - Halide
  -
- FFmpeg
- GStreamer

Choice of Interface

Procedural Framework: Pipes, Streams, & Graphs

Concurrency: Parallel = Performance?

Not always, no. While concurrent processing of independent tasks or sequentially arriving data elements will almost always increase performance, this is not always the case. At a lower instruction-level than we typically program, synchronous operations can often be optimized in ways that asynchronous operations cannot, typically through strategic register allocation, or by taking cache-hit performance). "Globally Asynchronous Locally Synchronous"

Scheduling

Adaptive

Choice of Operations

- What is the goal?
- Is it effective?
  - Is the computation cost worth the result?
- Are there side-effects or artifacts?
  - Can they be reliably controlled or accounted for?

Motion Correction

In our application, the goal of a motion correction operation is to artificially suppress translation of the brain tissue parallel to the image plane. *Phase-Correlation* (also referred to as *normalized cross-correlation*) has consistent performance across a range of image sources with varying spatial noise characteristics. However, a large non-uniform change from reference frames - such as occurs when cells with low baseline fluourescence are first activated - can cause drastic errors that must be recognized and corrected by a supervisory procedure. This can induce an undesirable, unpredicatable, and specifically inopportune latency Unfortunately in all the whole pipeline.

Unfortunately, "Globally Asynchronous Locally Synchronous"

as it's In some experimental setups,

The phase correlation method of #### Motion Estimation - cost: 2-10 ms/frame - Frequently unstable (depending on video content)

Motion Compensation & Interpolation

- cost: 400-800 us/frame
- Requires infill with nearby or prior pixel values if frame size is to be maintained

###Survey of Alternative Strategies

Implementation

EfficientCode

- Scalable - Reusable - Make it MODULAR
-

Operation

Video Processing

This section borrows from AIM-1 and AIM-2 from the prospectus.

Effective and Efficient Code

Biomimicry in Visual Processing

This section describes how image and video processing in the computer relate to visual processing in the mammalian brain. The overall goal is to emphasize the advantage and importance of biomimetic development.

Distributed Dataflow and Streaming

Choice of Implementation

Language: Is MATLAB the best tool for this job?

- Standard language in many engineering programs
- Proprietary
- Performance
- Compatibility
- Need for a "SandBox"
- Lacks modularity
-

Alternatives Languages

- Python
- C/C++
- Java/Scala
- Javascript/Node
- GO, Haskell

Databases

Big Data

- not exactly...
- disparate simple queries across

Map-Reduce

- Dataflow Processing
- Actors model
- Petri Nets
- Graph Processing - i.e. Tensorflow

Software for Procedure Development

Development Environment

Information and Informativity

This section presents a background in information theory in the context of behavior sensors and video streams. I further elaborate on what types of information neuroscientists and healthcare providers would hope to extract from these data, focusing on a larger picture of connecting the process of extraction directly the ultimate application.

Whereas previous attempts to process data focus largely on putting it in an intermediate state that is workable with current limitations on computation, visualization, and developer interaction with the data-streams, we now have an opportunity to skip intermediate states and develop with applications in mind.

Transition into compression by showing that previous attempts were essentially compression algorithms developed specifically for the data and preconceived notions about results, but that we can also apply very well developed general compression algorithms, with slight modifications to extract features such as localized bitrate that quantify (in an unbiased way) information content in an selectable region of a video-frame as it changes over time.

Compression: as a Tool, a Goal, as an Explanation

This section describes general compression algorithms as well as compression algorithms specifically tailored to application in video, accomplished by searching for both temporal and spatial redunancy.

Acknowledgements

The support and patience I have received from my committee has gone far beyond what should be expected of anyone. I can't thank you enough. * Xue Han, Ph.D. * Jerome Mertz, Ph.D. * Ian Davis, Ph.D. * Tom Bifano, Ph.D. * David Boas, Ph.D.

Appendix