# 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.

~~of guiding and providing new tools. offering broad coverage of problem solving capabilities, expanded availability of tools, and rapid feedback when) stimulating continuous awareness of the fast-paced changes~~

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.

# Body

* [Forward](#forward)
* [Body](#body) - [Background](#background) - [Image Processing](#image-processing) - [Effective and Efficient Code](#effective-and-efficient-code) - [Microscopy and Functional Imaging Two core innovations in available](#microscopy-and-functional-imaging-two-core-innovations-in-available) - [Image Processing 1. Computing Power and Connectivity](#image-processing-1-computing-power-and-connectivity) - [Big Data](#big-data) - [Map-Reduce](#map-reduce) - [Image Processing](#image-processing) - [Motion Correction Two approaches to find displacement](#motion-correction-two-approaches-to-find-displacement) - [Spatially Homogeneous phase correlation](#spatially-homogeneous-phase-correlation) - [Triangulate best](#triangulate-best) - [Image Enhancement](#image-enhancement) - [Feature Extraction](#feature-extraction) - [EfficientCode](#efficientcode) - [Tutorial](#tutorial) - [Incremental Update of Statistics](#incremental-update-of-statistics) - [central moments](#central-moments) - [Extract Features](#extract-features) - [DataFLow Architectures](#dataflow-architectures) - [“Globally Asynchronous Locally Synchronous”](#globally-asynchronous-locally-synchronous)

## Background

### Image Processing

### Effective and Efficient Code

* Background - Microscopy and Functional Imaging - Image Processing

## Microscopy and Functional Imaging Two core innovations in available

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

## Image Processing 1. 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

## Big Data

* not exactly...
* disparate simple queries across

## Map-Reduce

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

## 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)

## EfficientCode

* Scalable - Reusable - Make it MODULAR

## Tutorial

### 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)

## DataFLow Architectures

### “Globally Asynchronous Locally Synchronous”