*Abstract*- **There have been many advancements in miniaturized microscopes that produce in-vivo calcium extraction imaging in the study of neuroscience. Typically these small form factor cameras use small lenses but are restricted by the normal physical tradeoffs of resolution and field of view. In place of small lenses there has also been a lot of development for on-chip fluorescence microscopy using diffuser lenses that can allow for better tradeoffs. While diffuser lenses provide a low-cost and simple alternative to traditional lenses, they do require the solution of an ill-posed inverse problem that for in-vivo calcium extraction require real-time processing. We aim to provide in this paper a comprehensive comparison between the latency computation time using a gradient descent optimization minimizing a convex function for three processing devices, a Graphical Processing Unit (GPU), Central Processing Unit (CPU), and Field Programmable Gate Array (FPGA).**

Typically, the processing of calcium extracntion entails

FPGA/CPU/GPU/Other…..

Build a real-time deconvolution in as small a footprint as possible.

In this study we would like to compare the speed(latency) of

Our target is computational imaging of calcium imaging using a diffuser

Our goal is to compare the latency of a low-cost high, quality imaging system to be eventually used as the front end of an optical system for calcium trace extraction.

Why would we combine the diffuser calculations with the traditional calcium imaging which entail

1. Background

Talk about calcium imaging

At its’ core calcium imaging requires the processing of four major areas: background filtering, motion correction, neural enhancement and finally calcium trace extraction for processing.

While in this paper we focus on the front-end processing of reconstructing the image,

Talk about diffusers and work done here

As we are talking about diffusers we mention stuff about PSF

There are models that can be used to calibrate

Talk about models and variant and invariant

Diffusers provide a light field representation of 4-D information into a 2-D format, and in the process capture spatial and angle information. While in this paper diffusers provide our principal optical component, they have been successfully used in many other application (3D neural activity [3], compressive radar imaging [4], synthetic aperture [5], and visual odometry [6].). As a highly diffractive medium, the speckles (caustic patterns) from interference in our diffuser create a type of signature that can encode plenoptic information about our illuminated object and with proper assumptions allows us linearity in reconstruction that lends itself to well-established inverse problem definition and optimization techniques.

An important point to using diffusers and the randomness they provide is to see if this fits within the theoretical framework of compressive sensing, that has conditions on the inverse matrix. When trying to solve diffuser inverse problems with less than full rank, it is possible to recover the original vector if some conditions on a matrix are met. Such properties if satisfied such as Full Spark of matrix, and Null Space Property, both have full rank submatrices. A less computationally intensive way to calculate the sparsity that can be recovered is by computing the mutual coherence of the matrix, the combination of the measurement and representation matrix. Here is this paper we forego the formality of the mathematics and assume that our 3D objects are sparse in some domain.

Notes:

What is multi-row 1D FFT

If we use something other than the FFTW for CPUs or either the GPUFFTW for GPU or the CUFFT the CUDA based FFT, we should make sure that ???

Note: Here we will use the data flow for the CUFFT.

For both CPU and GPU we perform a column order 1-D FFT followed byua 1D-FFT row-order computation

Field Programmable Gate Arrays (FPGAs) are widely used in embedded products that require speed (MHz), latency(time) and allow for the flexibility of being able to re-program an algorithm even when the product is deployed in the field.

Talk about CPUs

Talk about GPUs

Talk about overall computational methods

( from paper in Fall)

1. Methods

While there are many ways to solv

Diagram

Description automatically generated

1. In theory a lens less system has all the information available at the sensor. However, the problem is ill-posed. Introducing a diffuser gives structure to our point spread function, or random matrix, allowing us to solve an otherwise intractable problem. If we enforce sparsity as a prior, and non-negativity (no negative pixels), minimize for the least squares we are trying to solve the equation below
2. Text

   Description automatically generated
3. Here in (1) we do not show the regularization term that helps enforce sparsity, instead we want to show the main thrust of how we approach minimization overall
4. Text

   Description automatically generated
5. Here, the significance of (4) is that we have reduced the problem of computing A which could be a very large matrix to the adjunct of M and C, a much more manageable problem. (C is introduced as a cropping matrix)
6. A screenshot of a computer

   Description automatically generated with low confidence
7. Here there are a few points to note. The first is that the M matrix in (5) has already been decoupled from A, and this is because if we included the cropping effect into A, the matrix would be ill-conditioned. Continuing with (5), we have just expressed the convolution of “Mv” as a product of Fourier and the inverse of that dot product but have expressed the equation as products of matrices.
8. Text

   Description automatically generated
9. Here we substituted (7) back into (4) to give us (8) and taken the adjunct of (8) to give us (9). At this point (8) and (9) we can implement into code to solve our iteration (10).
10. Text

    Description automatically generated

A picture containing text, gallery, different, clock

Description automatically generated

ADMM

Gradient descent

Algorithm ( from Paper in Fall)

1. System Implementatio

Sh Diagram

Description automatically generatedow figure

Describe steps of our block diagram and show

In our cropped system where we have reduced the size of an image thru don

We will use a fixed point distance of 16 bits, note this can be adjusted empirically as we begin to implement our system in Simulink.

With CPU

With GPU

( Reference paper is in file : IEEE under FPGa folder/ “An Efficient, Model-Based CPU-GPU Heterogeneous FFT Library”, by Ogata, et al. CUDA can help in some areas here.( See reference [11] from Ogata paper above)

For the Cores (CPU and GPU)we see ( from Ogata [5] and [8]))

Memory limitations

Programming limitations that relate to DirectX or OpenGL

With combined CPU and GPU

With FPGA show our diagram

( From Proposal). How we do what we set out to do. Some of this comes from proposal

1. Results

( Running in Simulink)

Figure that shows images between diffused image and FFT of it.( Maybe get this picture with Python)

Estimate with FPGA

In our system we will only compare the 2- dimensional FFTs since they account for the major processing modules in our FISTA deconvolution design.

Estimates with CPU

Estimates with GPU

Estimate with CPU and GPU in tandem.

Do the times meet our system requirements for processing 30 frames a second for a size image that has been downsampled from the original full CMOS array.

1. Discussion and Conclusion

Talk about calcium imaging to be done Other types of processors

A little about optogenetics

1. References