CUDA (Compute Unified Device Architecture) and OpenCL (Open Compute Language): Programming GPUs

Dr. Bharathwaj "Bharath" Muthuswamy

Visiting Assistant Professor in Computer Science
The College of New Jersey
bharath.berkeley@gmail.com

February 20th 2018

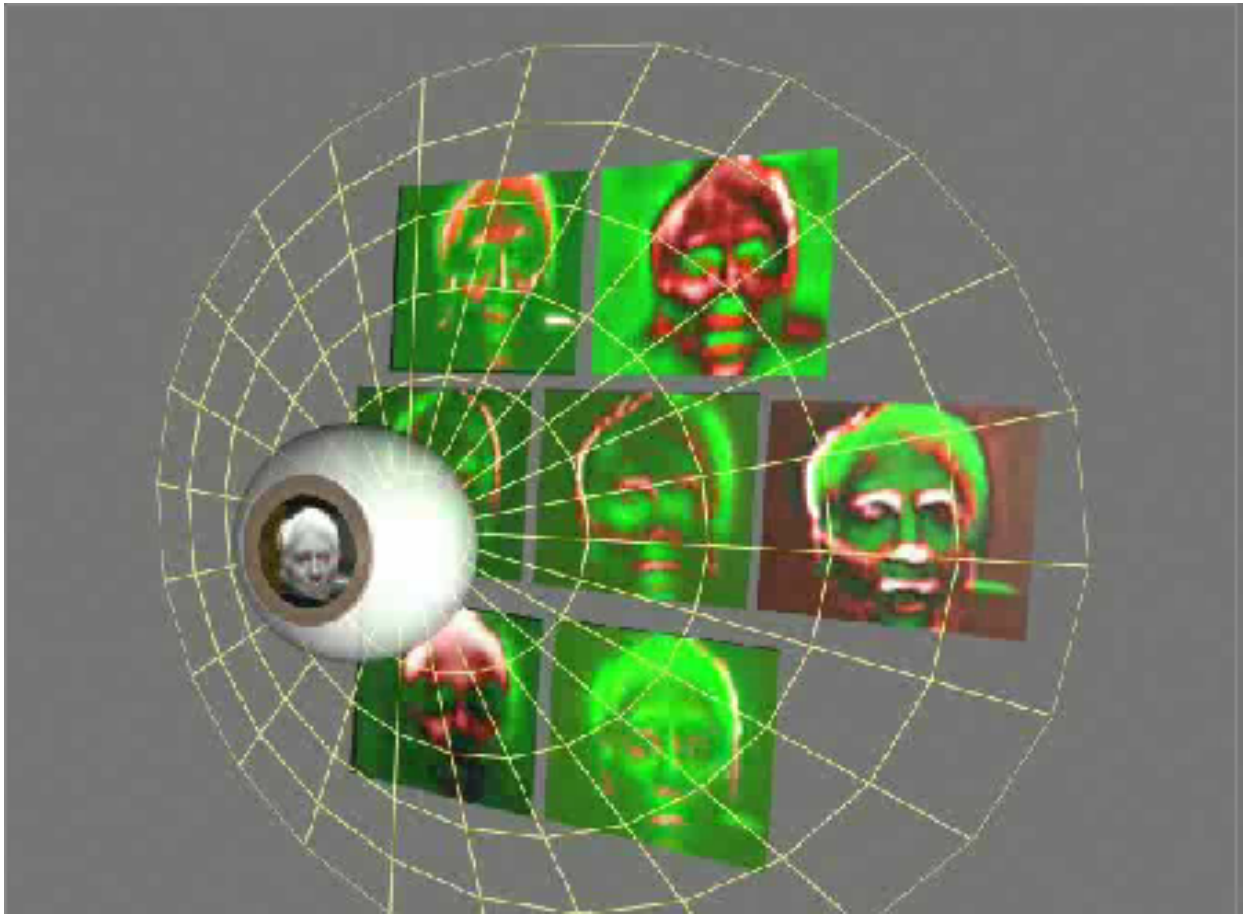
About me...

- BS (2002), MS (2005), PhD (2009) in EECS from the University of California, Berkeley
  (advisors: Dr. Leon O. Chua, Dr. Pravin P. Varaiya)
  - For my MS, I worked on biomimetic bipedal robotics using Central Pattern Generators (I did not work on this after 2006)
  - For my PhD, my primary contribution was designing, implementing and rigorously proving the existence of chaos in the Muthuswamy-Chua system (circuit): an inductor-capacitor-memristor circuit in series (parallel)
- Areas of interest:
  - Computer vision and Quantum Computing
  - Nonlinear Dynamics (Circuits). Specifically: chaotic circuits and memristors
  - Embedded (FPGA) Systems and Education

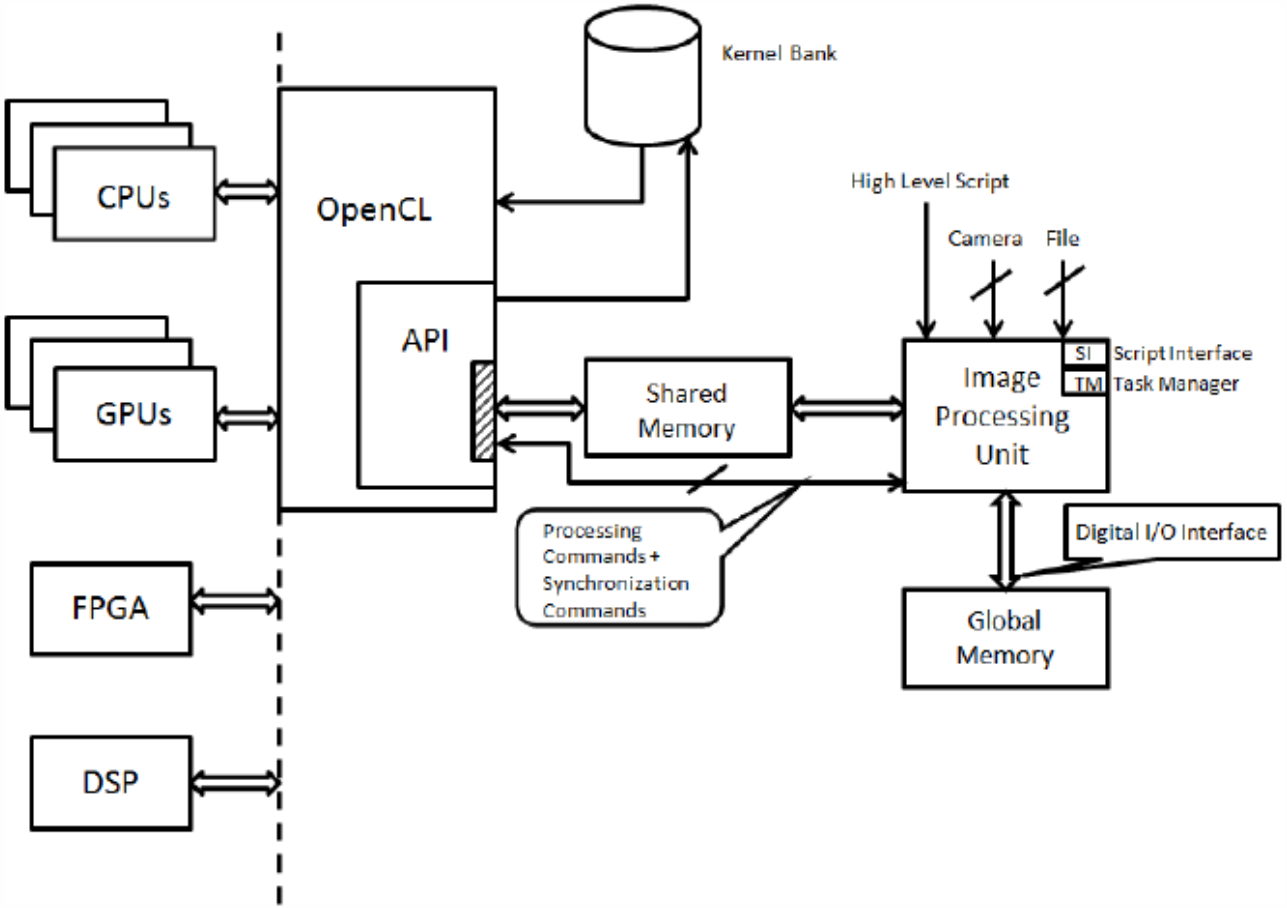
Project goal, Presentation goal and Organization

- Goal of my research: Implement a model for the mammalian visual pathway on a heterogeneous platform using Discrete Time Cellular Nonlinear Networks (DTCNN).
  - Application: Robust recognition of hand-drawn electronic circuit diagrams
- Goal of this talk: Discuss GPU programming strategies (CUDA and OpenCL)
- Organization:
  - Conceptual Overview of the Project
  - Understanding the graphics pipeline
  - Part I: CUDA - Hello, world; SAXPY
  - Part II: OpenCL - Hello, world
  - Current work: OpenCL specification of DTCNN
  - Conclusion and Q/A

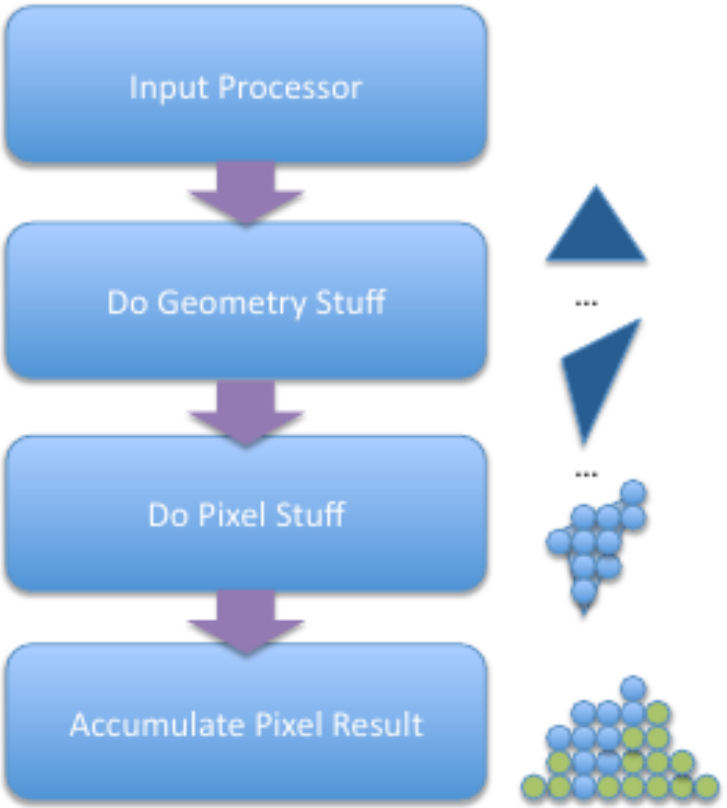
Mammalian Retina



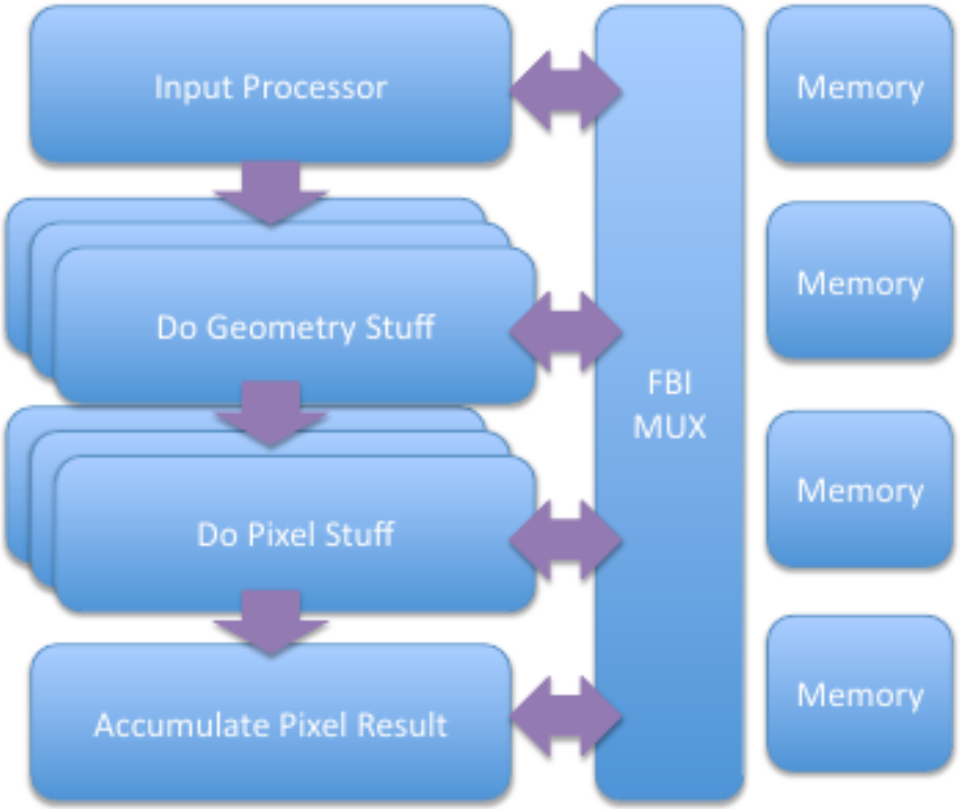
Block Diagram of the Project



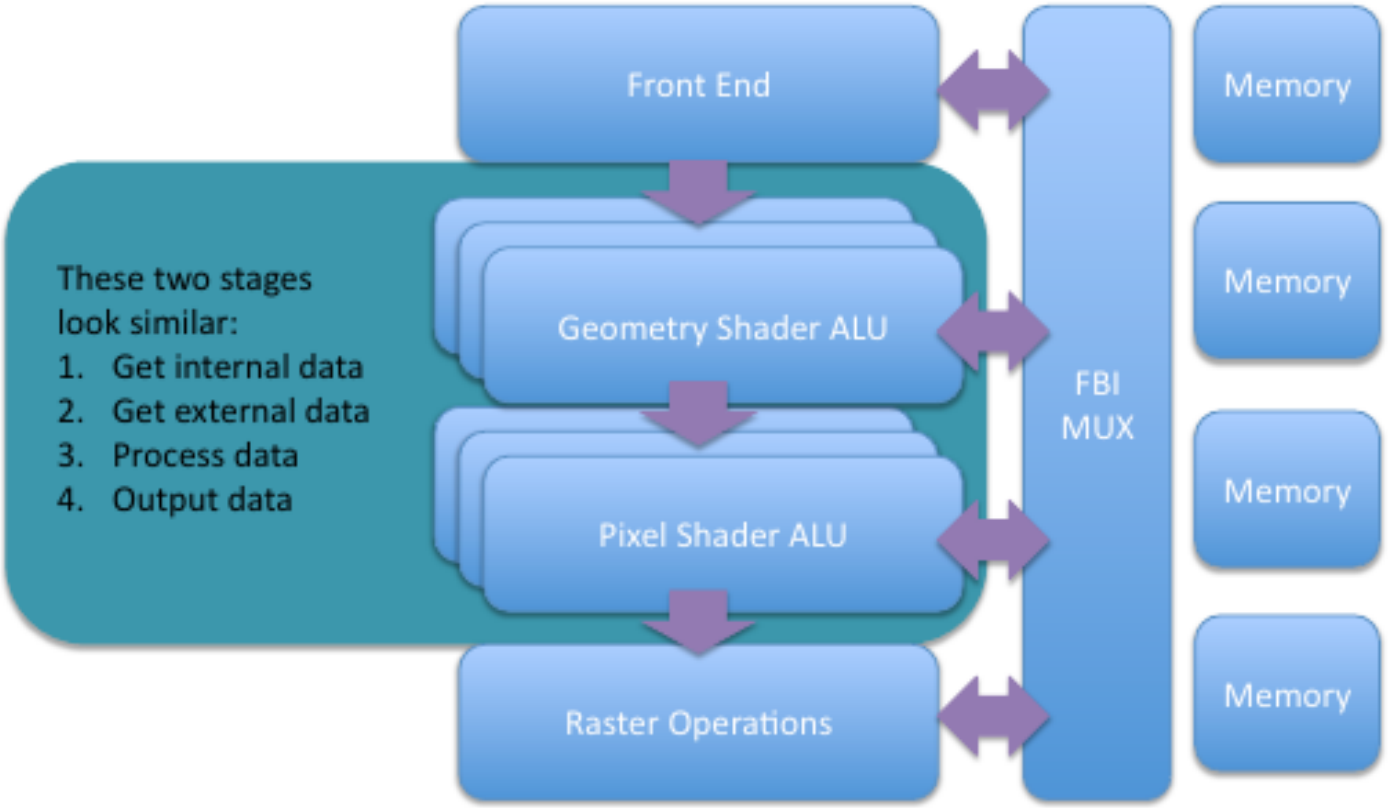
Simple Raster Graphics Pipeline



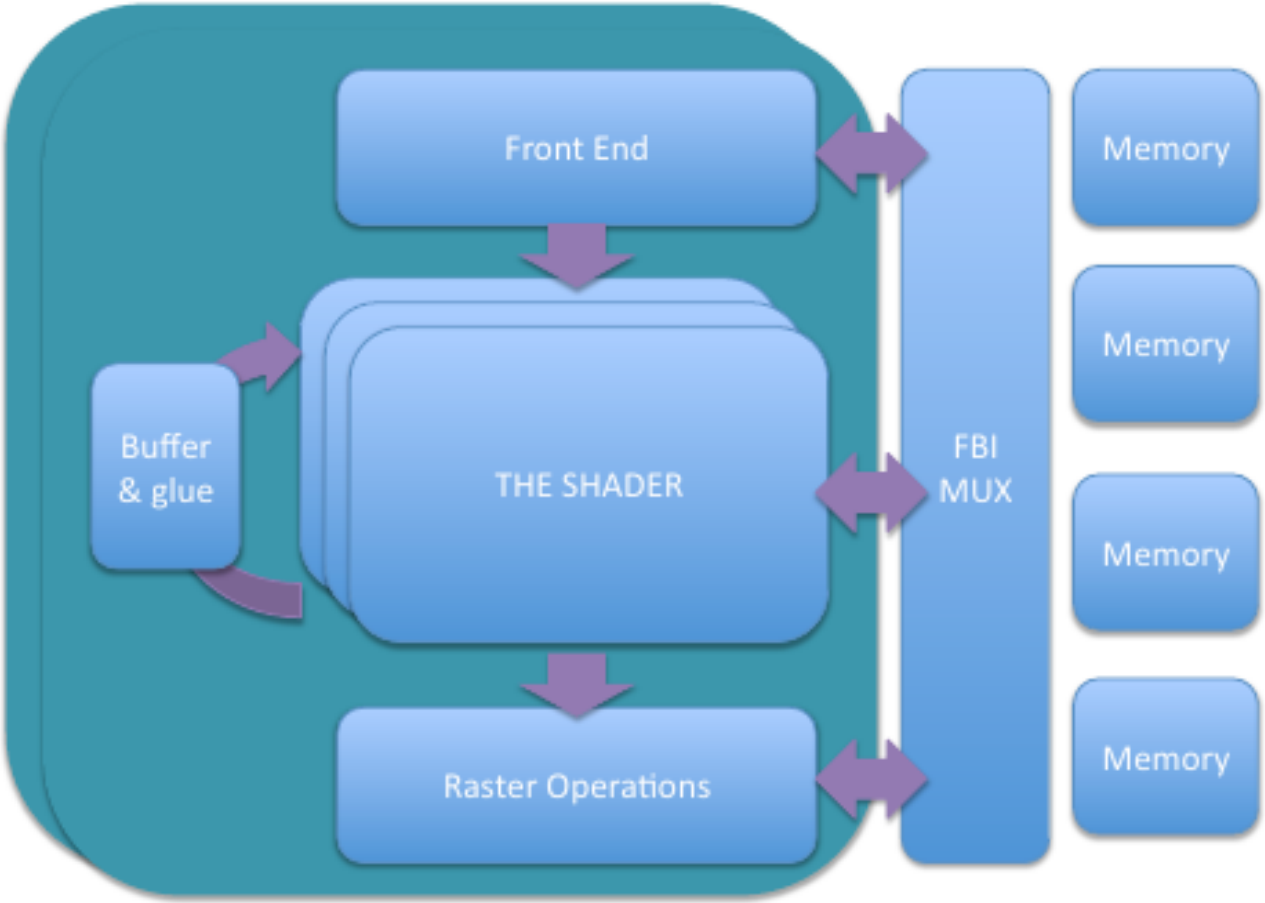
Enhanced Graphics Pipeline



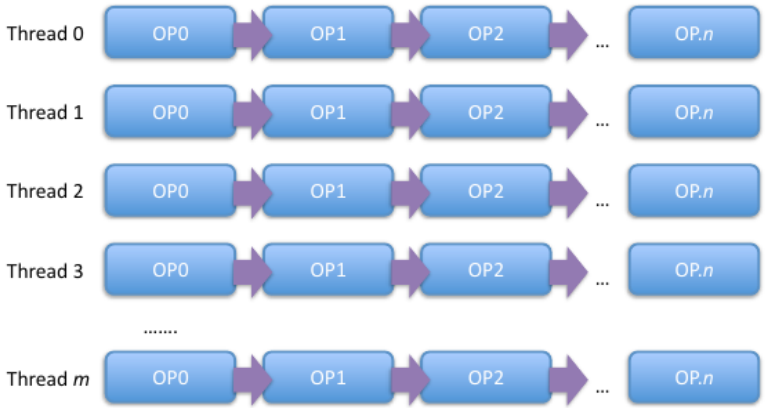
General Purpose Pipeline



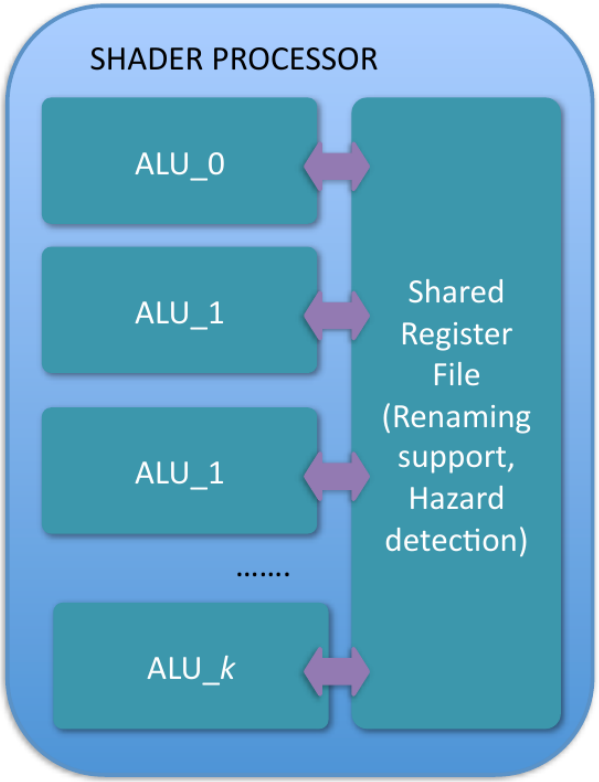
Shader Pipeline

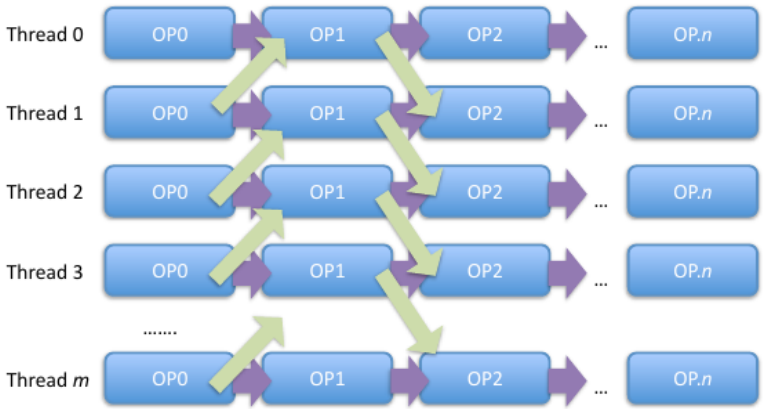


SIMD Pipeline



SIMT Pipeline





CUDA: High-level overview

- Traditional serial version of the program...
- ...that sets up subroutines or **kernels** that are to be executed in parallel

CUDA Examples

- http://www.harpgroup.org/muthuswamy/computerVision/CUDA/helloWorld
- http://www.harpgroup.org/muthuswamy/computerVision/CUDA/saxpy

OpenCL: High-level overview

- Think of OpenCL as a "personal robot army":
  - Each robot or **OpenCL work unit** works on specific data **independent** of other robots
  - Each robot has a unique id, robots can be grouped
- Limitation: data-parallelism (see example of DTCNN later)

OpenCL: Steps

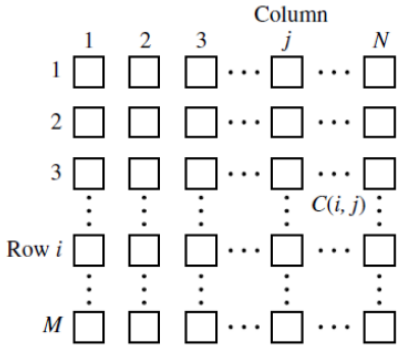
- Get list of available platforms
- Use platforms to select appropriate device(s)
- Create an OpenCL context
- Create command queue
- Create memory objects
- Create kernel
  - Load kernel specified via OpenCL C-extension(s)
  - Compile kernel to obtain kernel object
  - Set any kernel parameters
- Execute the kernel
- Read result from previously created memory object
- Free memory

OpenCL: Example

- http://www.harpgroup.org/muthuswamy/computerVision/OpenCL/helloWorld

(DT)CNN: Concept(s)

- $M \times N$ CNN architecture:



- State equation and output equation (nonlinear):

$$\dot{x}_{ij} = -x_{ij} + \sum_{C(k,l)\in S_r(i,j)} A(i,j;k,l)y_{kl} + \sum_{C(k,l)\in S_r(i,j)} B(i,j;k,l)u_{kl} + z_{ij}$$

$$y_{ij} = \frac{1}{2}\left(|x_{ij}+1| - |x_{ij}-1|\right)$$

- DTCNN:

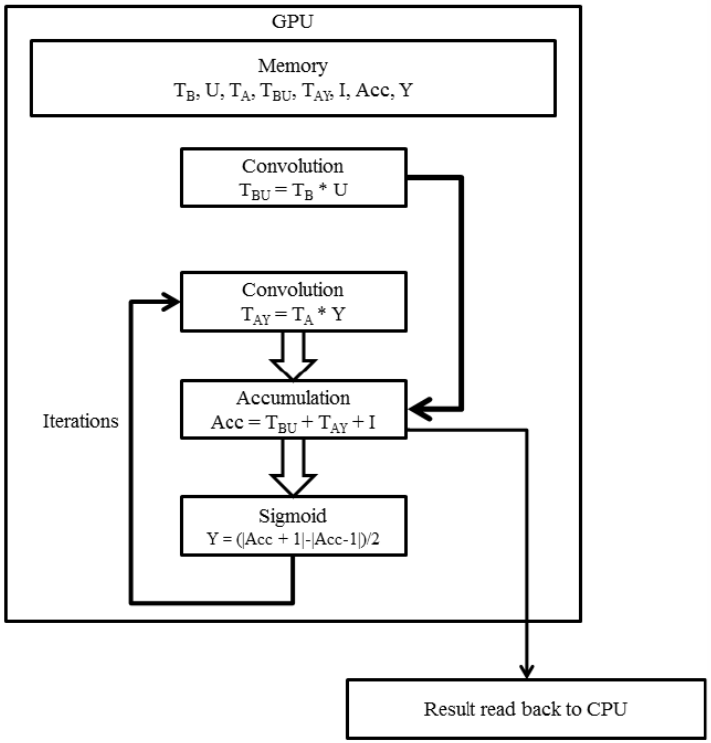
$$X_{i,j}(t+1) \approx \sum_{c(k,l)\in N\, r(i,j)} A\,(i,j;k,l) f\left(X_{k,l}(t)\right) + \sum_{c(k,l)\in N\, r(i,j)} B\,(i,j;k,l)\; U_{k,l} + I$$

OpenCL flow

- DTCNN state equation can be written as:

$$x_{ij}(t+1) = T_A * y_{kl}(n) + T_B * u_{kl} + I$$

- Proposed OpenCL implementation:



Conclusion and Q/A

- Current work: Implementing various retinal "filters" via DTCNN using OpenCL
- Future work: post-processing of images obtained from hand-drawn electronic circuit diagrams:
  - Deep learning (?): Learn data representations, but use images obtained from retinal "filters". Note: it is known the visual cortex processes in **parallel** the images obtained from the retina
- *Potential* funding: https://www.grants.gov/web/grants/view-opportunity.html?oppId=300721
- Reference: Detailed howto and code is online: http://www.harpgroup.org/muthuswamy/computerVision