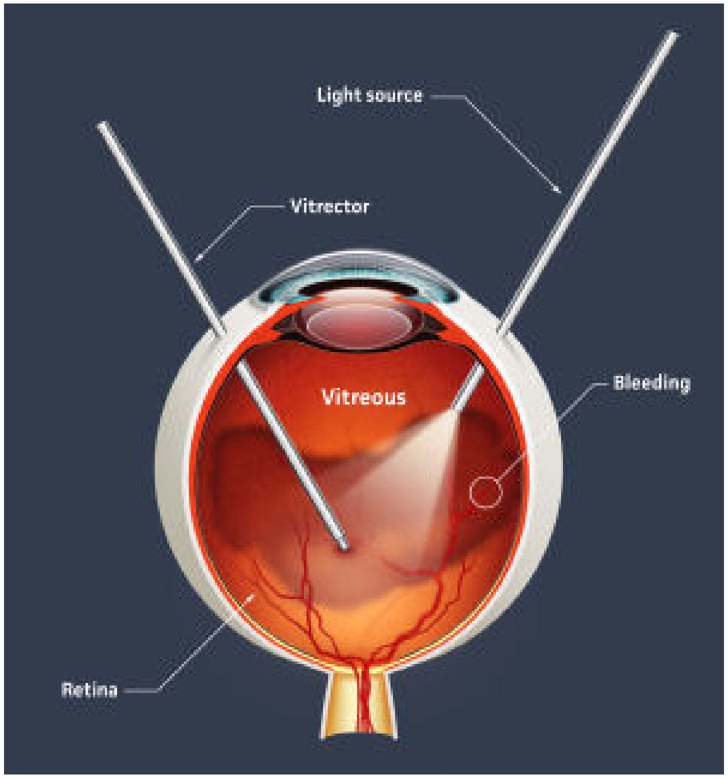
Visual Tracking of Surgical Tools in Retinal Surgery using Particle Filtering

Group 14
William Yang and David Li
Presenter: William Yang
Mentor: Dr. Rogerio Richa

Project Background

- Vitreoretinal surgery treats problems with retina, macula, and vitreous fluid
- Many complications due to fragility of retina, indirect visualization, and physiological tremor
- Long operating times and risks of surgical error



http://www.eyedoctorguide.com/eye_problems/vitreoretinal_surgery_retina.html

Project Description

- Goal: Develop a direct visual tracking method for retinal surgical tools using particle filtering and mutual information
- Advantages:
  - Particle filter is computationally efficient and robust
  - Mutual information performs better than SSD and NCC in many cases
- Opportunity to parallelize computation in both particle filter and mutual information

ERC | CISST

LABORATORY FOR Computational Sensing + Robotics THE JOHNS HOPKINS UNIVERSITY

Research Paper

Mutual Information Computation and Maximization Using GPU

Yuping Lin and Gérard Medioni

Computer Vision and Pattern Recognition Workshops (CVPR)

Anchorage, AK, pp. 1-6, June 2008

Summary

- Problem:
  - Multi-modal registration of images is difficult to solve
  - Maximization of mutual information works but is slow
- Key result: Validated that calculation of mutual information can be parallelized on a graphics processing unit (GPU)
- Significance:
  - Use of mutual information to register multi-modal images can run in a reasonable amount of time

Mutual Information

$$h(v) = -\sum_{v} p(v) \ln(p(v))$$

$$h(u,v) = -\sum_{u,v} p(u,v) * \ln(p(u,v))$$

$$MI(u,v) = h(u)+h(v)-h(u,v)$$

Mutual Information Approximation

- "Viola's Approach" provides a closed form solution of derivatives
- Estimate probability density:

$$p(v) = (1/N_A) * \sum_{v_j \in A} G_\varphi(v-vj)$$

- Entropy:

$$h(v) \approx \frac{-1}{N_B} \sum_{v_i \in B} \ln \frac{1}{N_A} \sum_{v_j \in A} G_\psi(v_i - v_j)$$

Mutual Information Approximation

$$MI(u,v) \approx \frac{-1}{N_B}\left\{ \sum_{u_i \in B} \ln \frac{1}{N_A} \sum_{u_j \in A} G_{\psi_u}(u_i - u_j) + \sum_{v_i \in B} \ln \frac{1}{N_A} \sum_{v_j \in A} G_{\psi_v}(v_i - v_j) - \sum_{w_i \in B} \ln \frac{1}{N_A} \sum_{w_j \in A} G_{\psi_{uv}}(w_i - w_j) \right\}$$

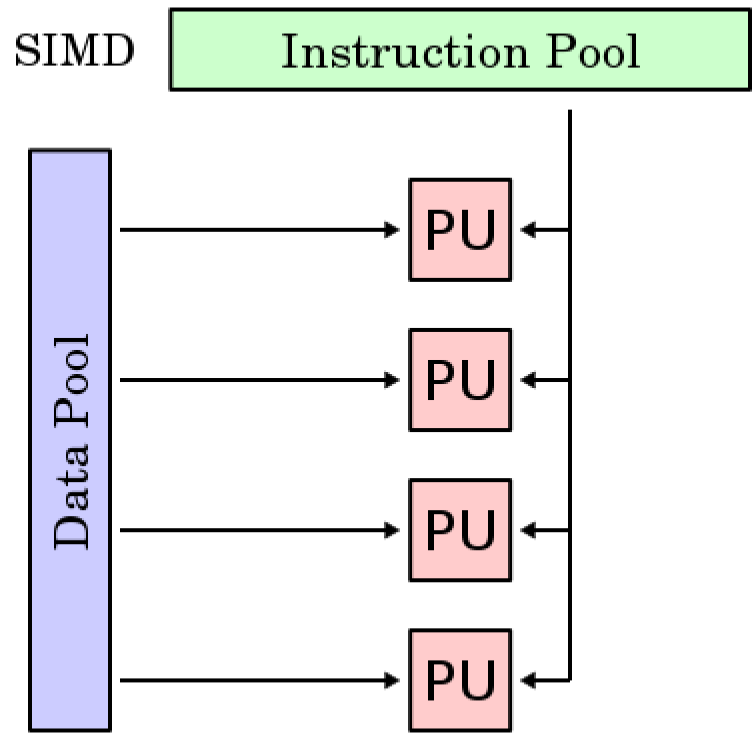
- Maximize this over a rigid transformation T using approximate derivative:

$$\frac{d}{dT} MI(I_u(x), I_v(T(x))) = \frac{1}{N_B} \sum_{x_i \in B} \sum_{x_j \in A} (v_i - v_j)^T W(x_i, x_j) \frac{d}{dT}(v_i - v_j)$$

Independent Calculations!

GPU/CUDA Architecture

- Single Instruction Multiple Data (SIMD) architecture

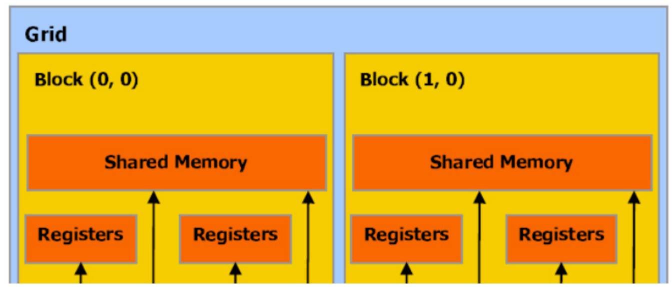


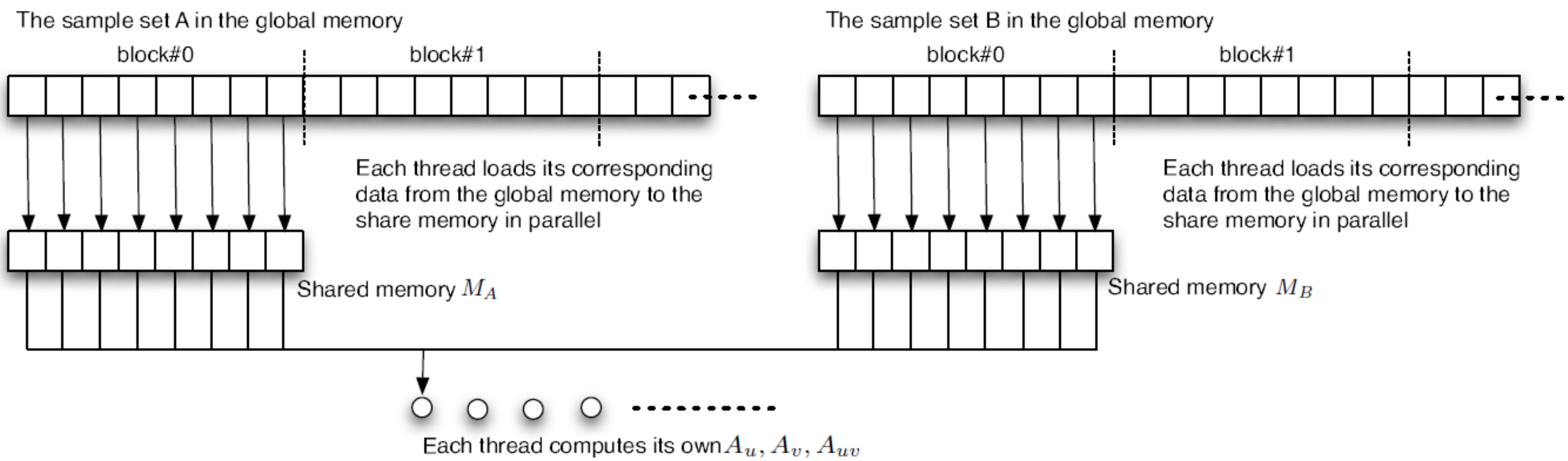
http://en.wikipedia.org/wiki/File:SIMD.svg

GPU/CUDA Architecture

Lin and Medioni, Mutual Information Computation and Maximization Using GPU

- Shared memory is key: on-chip and programmable

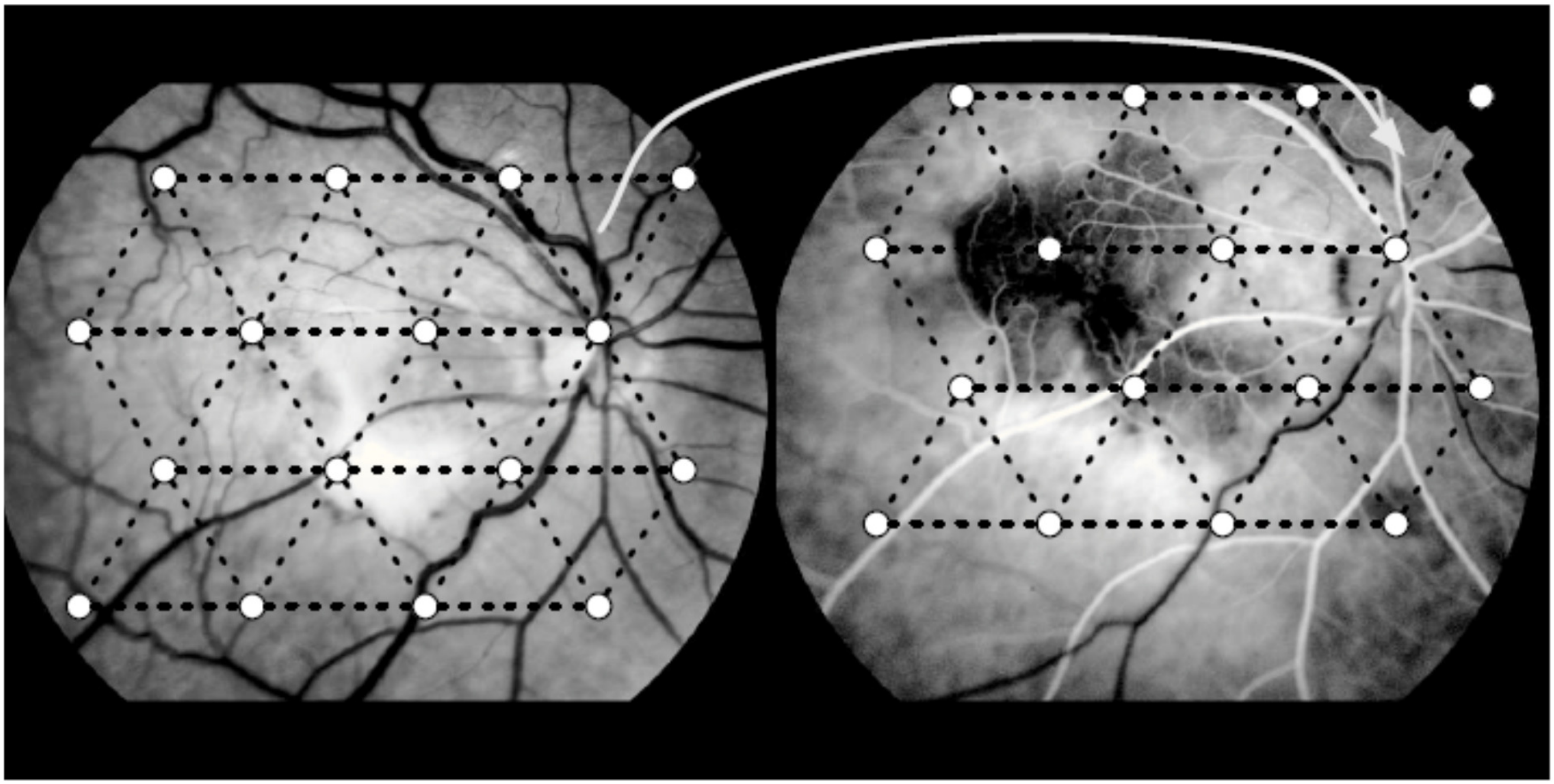




Lin and Medioni, Mutual Information Computation and Maximization Using GPU

Validation and Results

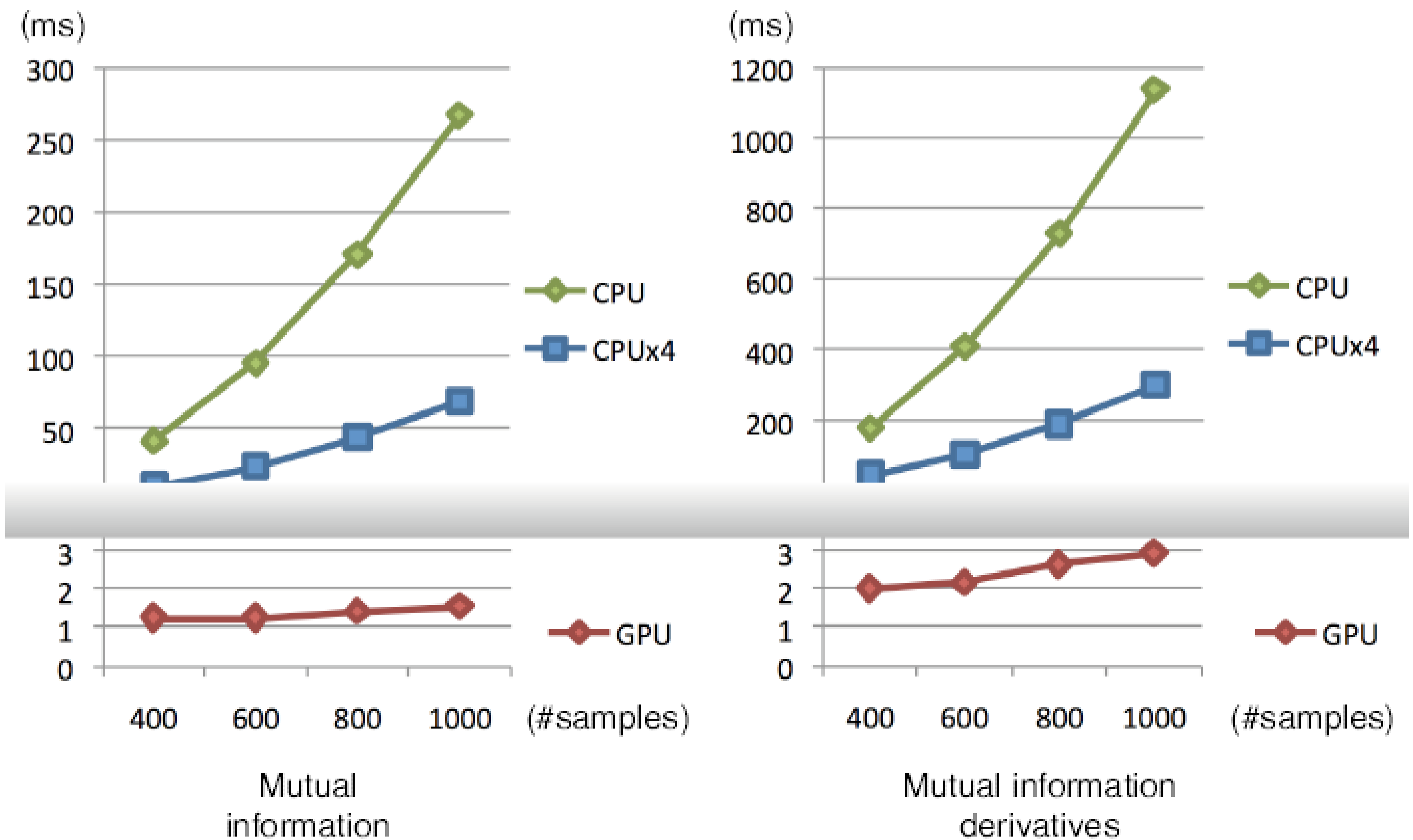
Test dataset



Lin and Medioni, Mutual Information Computation and Maximization Using GPU

Validation and Results

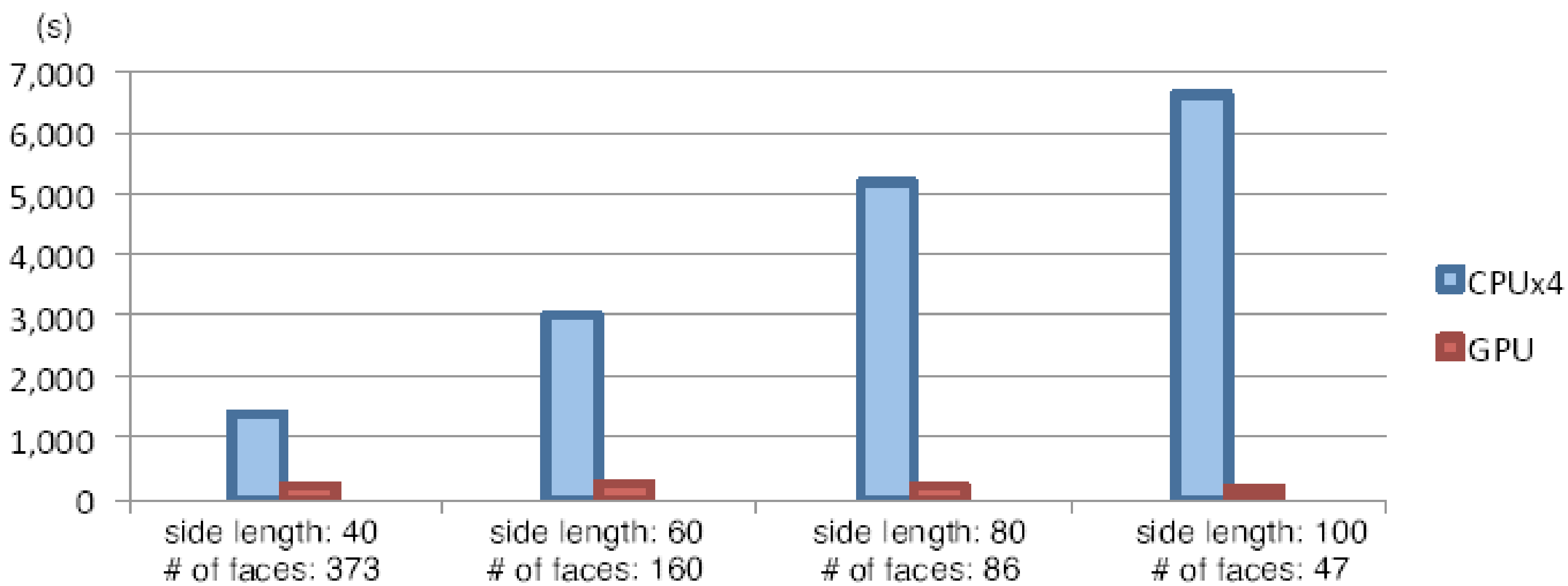
Time to compute MI derivatives



Lin and Medioni, Mutual Information Computation and Maximization Using GPU

Validation and Results

Time to perform registration (50 iterations)



Lin and Medioni, Mutual Information Computation and Maximization Using GPU

Conclusions and Future Work

- Approximation methods and CUDA algorithm enables use of MI for image registration within a reasonable amount of time as compared to a CPU-based approach
- Integration into applications
- Performance of other MI approximations

Comments: Strengths

- Novel approach to solve problem
- Mathematical details presented adequately
- Pseudocode (serial execution) algorithm
- Description of computational hardware included

Comments: Weaknesses

- More information on test datasets
- No info on inter-trial variance in results
- Why was Viola's algorithm chosen?
- CPU memory bandwidth not discussed
- Future work: bilinear interpolation

Relevance to Project

- MI on GPU: high throughput!, but…
- MI calculation method differs
- Gradient descent: local minima problem
- Repeating many calculations of MI with low sample count
- Serial fast MI or parallel MI?

Reading List

- NVIDIA CUDA C Programming Guide 4.1. 2011
- R. Shams and N. Barnes. Speeding up mutual information computation using NVIDIA CUDA hardware. *Digital Image Computing Techniques and Applications*, pages 555–560, 2007.

Questions?