GPU Accelerated Vessel Segmentation Using Laplacian Eigenmaps

Lin Cheng, Hyunsu Cho and Peter A. Yoon
Trinity College
Problem

Image segmentation
Partition pictures of vessels into segments
Laplacian eigenmap [1]

Local info embedded in **high dimensional space**

**Project** local info onto low-dimensional plane

**Optimize** the projection to preserve essential characteristics

**Cluster** the projected data points into segments

[1] Tziakos, Laskaris, and Fotopoulos
Segmentation process

1. Build graph of local info
2. Apply Laplace operator
3. Solve optimization problem
Build graph of local info

Store the resulting graph in a **weight matrix**

**Edges** reflect variations among different regions (global variation)
Apply Laplace operator

Form **Laplacian matrix** $L = I - D^{1/2}WD^{1/2}$ encoding the Laplace operator.

The operator formulates an **optimization problem**: Projections of well-connected nodes should also be tightly clustered.
Solve optimization problem

Solutions to eigenvalue problem $Ly = \lambda y$ are optimal solutions
If solutions are good, we can detect clusters
Characteristics of GPUs

Massively parallel – lots of small cores (workers)
Good for high-throughput, compute-bound tasks
Separate memory space from main memory
Strategy: Reduce memory footprint

On-GPU memory is limited
Reduce memory usage and we can pack in more work into GPU
Strategy: Reduce memory footprint

Weight matrix generation:
- Do not store intermediate results
- More entries can be calculated in parallel; **10x faster**
Worker allocation
Strategy: use Lanczos method

We need only a few smallest eigenvalues of $L$

Lanczos method iteratively solve for the eigenvalues needed
Takes 1/28 time of conventional method
Performance of Lanczos on GPU, by parts

- `malloc`
- `eigenvalue`
- `laplacian pairweight`
Performance: vs. multicore CPUs

Scaling by number of cores, pairweight

- **CPU**: two Intel® Xeon® E5-2620
- **GPU**: one Nvidia Tesla® K20c
Acknowledgement

Trinity College, Student Research Program
Nvidia Corporation, CUDA Teaching Center Program