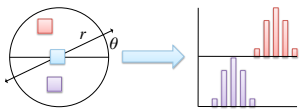
Efficient, High Quality Image Contour Detection

Bryan Catanzaro, University of California, Berkeley

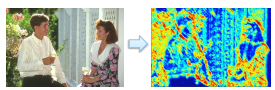
Overview

- Highly parallel processors, such as GPUs, bring new capabilities to computationally intensive problems
- We reexamine important computations, finding efficient parallel algorithmic approaches
- Performance gains we realize enable new applications for these algorithms

Localcues

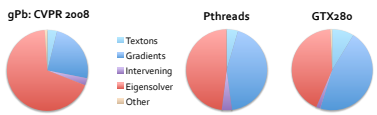


- Local image analysis
- Finds oriented edges at a given scale by forming histograms in half discs, then comparing histogram distance
- Histograms constructed by scan (better parallelism)



Results

Computation	Original Serial	Our algorithms, C + Pthreads (8 threads)	CUDA (GTX280)
Textons (K-means)	8.6	1.35	0.159
Gradients	53.8	12.9	0.84
Intervening Contour	6.3	1.21	0.03
Eigensolver	151.0	14.3	0.78
Overall	222 seconds	29.8 seconds	1.8 seconds



Parallel Scalability

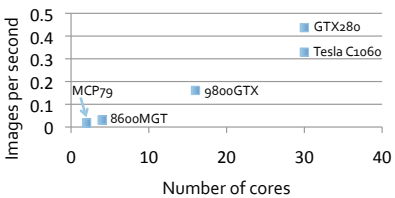
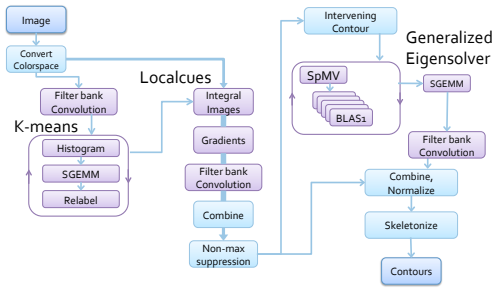


Image Contour Detection

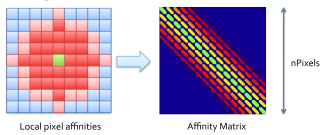
Image | Image Contours | Image Segments

- Segmentation is fundamental to analysis
- Break image into pieces, analyze separately
- Image Recognition, Retrieval, and Synthesis problems all make use of image segments

A Library of Components

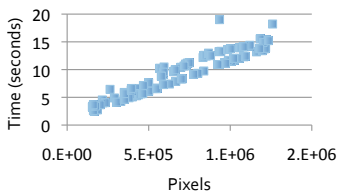


From Image to Sparse Matrix

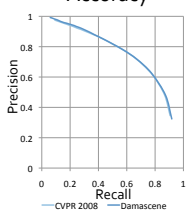


- Sparse matrix encodes local affinities into a globally coupled system for good segmentation
- Done by Intervening Contour computation
- Fast Sparse Matrix Vector Multiply:
- 40 SPGFLOPS on GTX280
- 12 SPGFLOPS on Nehalem

Image Size Scalability (Tesla C1060)



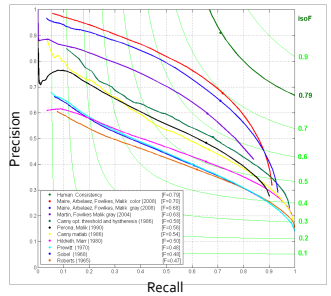
Accuracy



45 Years of Progress in Contour Detection

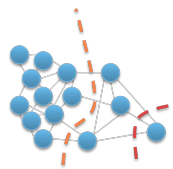
- Contour detection is a hard problem
- We're trying to find contours humans would find
- We benchmark against a database of human labeled images
- The gPb algorithm is currently closest to human performance, as shown on right
- But it is too computationally intensive for widespread use – until now

4 minutes to 2 seconds



Generalized Eigensolver

- Graph Partitioning is a fundamental operation in image segmentation
- Standard approach to graph partitioning: Min-cut
  - Poor image segmentation due to isolated pixels in a segment



Normalized cut | Min-cut

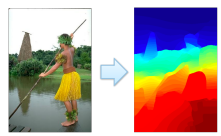
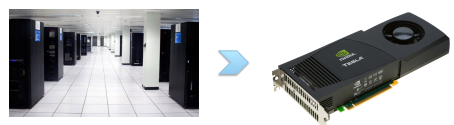


Image | Normalized Cuts Eigenvector

- Normalized cuts gives higher quality results, especially for image segmentation
- Normalized cuts graph partitioning done through sparse eigensolver
- We use the Lanczos algorithm with the Cullum-Willoughby test

Conclusions

- Rethinking algorithms to express more parallelism, along with implementation on highly parallel platforms such as GPUs, can lead to transformational performance
- This is especially important for large scale image processing, such as in large scale image retrieval systems
  - Instead of needing a huge cluster to process a database, we can use a few GPUs



- We also can bring new classes of algorithms to solve hard problems – for example, things that are traditionally reserved for image recognition can be applied to search