Hypergraph Partitioning on GPUs and Image Classification

Trinity College
HARTFORD • CONNECTICUT

Hyunsu Cho '15
Faculty Advisors: Peter A. Yoon, Department of Computer Science
Lin Cheng, Department of Engineering

Summary

- Goal: classify pictures into two categories
- **Hypergraph partitioning** is a good algorithm for image classification
- Hypergraphs capture similarities among neighboring pictures
- Use GPUs to accelerate hypergraph partitioning
- GPUs have lots of small workers and thus require a different programming model
  (many-core architecture)

Image Classification

- Want to **classify** a set of pictures into two groups

- Similar pictures should belong to same category
- **Hypergraph**: compare multiple pictures at a time
  group nearest neighbors into **hyper-edges**

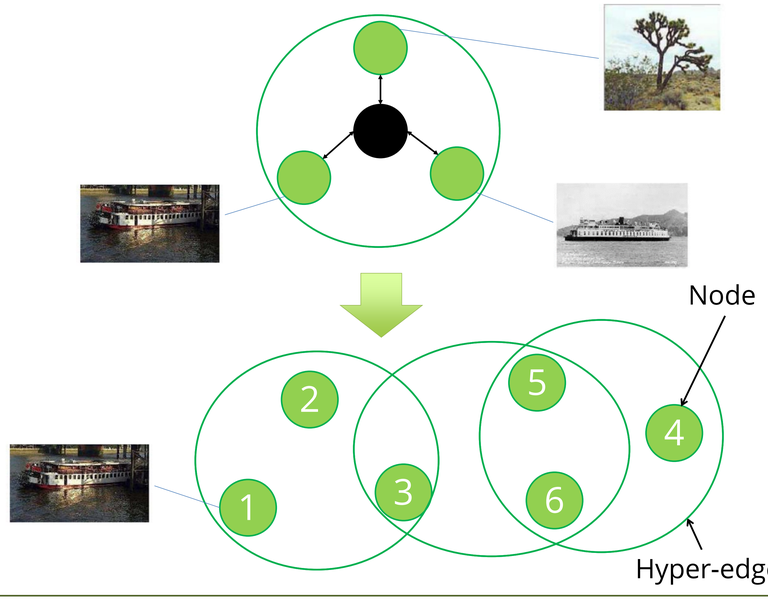
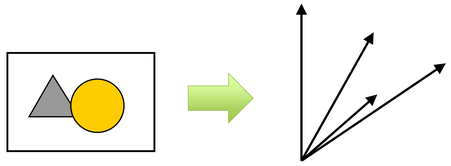
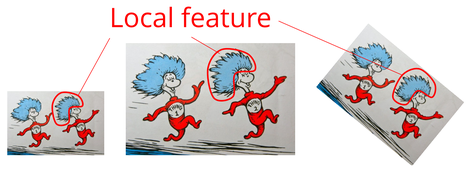


Image Classification Pipeline

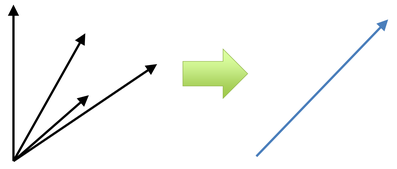
1. Read pictures
2. Extract **local features**



Local features: independent of rotation, resizing, relocation



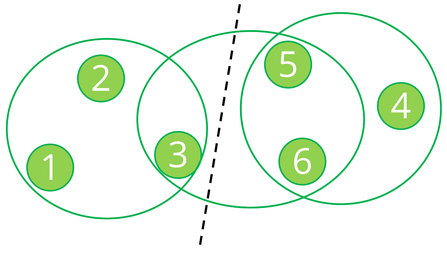
3. Pool local features into summary vectors



4. Measure pairwise distances
5. Form hyper-edges by grouping nearest neighbors
6. **Partition the hypergraph (most expensive)**

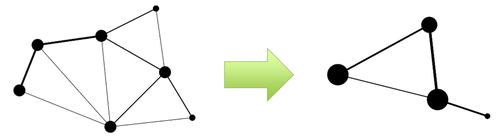
Hypergraph partitioning

- Goal: **partition hypergraph to minimize edge cut**
  Some hyper-edges are stronger than others
- **Does not require prior training**



Multi-level Paradigm

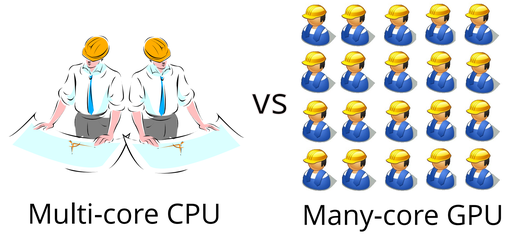
- High computational cost to partition hypergraphs
- **Coarsening**: reduce # of nodes by fusing them
- Solve the reduced problem, and then
  estimate solution for the original problem
- Reduce recursively until # nodes become manageable



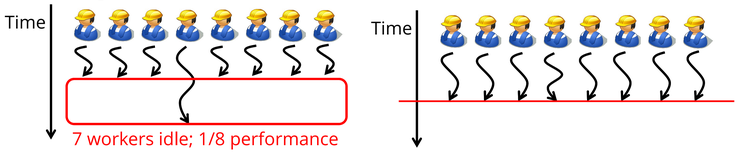
- Criteria for coarsening: combine nodes with similar list of hyperedges *"judge a person by his friends"*

Graphical Processing Units (GPUs)

- **Massively parallel**: large number of small workers
- Powerful and affordable
- Must break down tasks into small, independent pieces

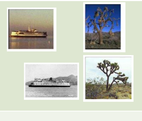


- Issue: 32 workers in each **warp** walk in lockstep
  if one worker has lots of work, others have to wait
  Response: collaborative planning to even out work assignments



- Result: **15-20x** speedup vs. single-core implementation

Classification Results

	Support Vector Machine (needs training)	Hypergraph partitioning (does not need training)
	98.03 %	99.24 %
	91 %	80 %