Exploring the Design Space of a Parallel Object Recognition System
Bor-Yiing Su, Tasneem Brutch, Kurt Keutzer

Statements
1. Exploring the design space is necessary to achieve high performance on a hardware platform of choice
2. Take advantage of domain knowledge is necessary to understand trade-offs among different parallelization methods and achieve peak performance

Targeting Object Recognition System

Overall Performance

Exploring the Algorithm Layer
- Traditional BFS-graph traversal algorithm
  - Propagate information to nearby neighbors
- Structured grid algorithm
  - Gather information from nearby neighbors

Exploring the Parallelization Strategy Layer
- Graph Partition
- Parallel Task Queue

Future Work
- Develop Frameworks for Object Recognition Key Computations
  - Automate the design space exploration procedure
  - Integration with the Par Lab stack
  - Use SEJITS to explore the parallelization strategy layer
  - Use autotuners to explore the platform layer

BFS Graph Traversal Kernel
- Graph representation of an image
  - Each pixel is represented by a node
  - Neighborhood relationship between pixels represented by edges
  - BFS Graph traversal on an image
    - Propagate information from some pixels to other pixels

Histogram Kernel
- Use a 128-bin histogram to represent the contour feature of a region
  - The location information is discretized into a 4 x 4 grid
  - The orientation information is discretized into 8 orientations

Pair-wise χ² Distance Kernel
- Compute χ² Distance between each pair of regions
  - Similar region pairs have shorter distances
  - Different region pairs have longer distances

Future Work
- Develop Frameworks for Object Recognition Key Computations
  - Automate the design space exploration procedure
  - Integration with the Par Lab stack
  - Use SEJITS to explore the parallelization strategy layer
  - Use autotuners to explore the platform layer

Future Work
- Develop Frameworks for Object Recognition Key Computations
  - Automate the design space exploration procedure
  - Integration with the Par Lab stack
  - Use SEJITS to explore the parallelization strategy layer
  - Use autotuners to explore the platform layer