Architecture Aware Design for a Parallel Object Recognition System

Bor-Yiing Su, Bryan Catanzaro, Tasneem Brutch, Kurt Keutzer

This research is supported in part by Microsoft (Award #024263) and Intel (Award #024894) funding and by matching funding by U.C. Discovery (Award #Digi07-15227)

Object Recognition

- Trained Categories
- Image Queries
- Outputs

System Performance

- Detection Quality
- Speedup by Parallel Implementation on Nvidia Tesla C1060

Parallel Graph Traversal on Images

- Parallelization strategies
  - Inner product based algorithm
  - Outer product based algorithm

Parallel Pair-wise Distance

- Widely used to measure the difference between features

Weight Learning

- Contour Feature Extraction
  - Use a 128-dimension histogram to represent contour feature
  - Select contour strength of 8 orientations on a 4x4 grid

Training

- Parallel BFS Graph Traversal on Images Using Structured Grid
- Graph representation of an image
  - Each pixel is represented by a node
  - Neighborhood relationship between pixels represented by edges
- BFS graph traversal algorithm is widely used in region and boundary analysis

- Parallelization strategies
  - Transform the BFS traversal problem into structured grids computation
  - Parallelize the task queue in the BFS traversal algorithm
  - Cache choices
    - No cache at all
    - Use texture memory to cache both vector sets
    - Use shared memory to cache vector elements

Conclusion

- The performance of parallelizing a computation will be influenced by
  - Parallelization strategy
  - Underlying hardware architecture
  - Input data properties
  - We need to understand the trade-offs between different parallelization implementations to optimize the computation
  - Ideally, we should dynamically adjust the parallelization strategy according to the input data properties at runtime

Experimental Results

- If the # of vector pairs is small, apply the inner product algorithm
- If the # of vector pairs is large, apply the outer product algorithm
- Always use shared memory to cache vector elements