Donald Whyte
sc10dw

Accelerating Multi-Dimensional Search

Start of Iteration Supervisor Meeting

Iteration #3 (14/04/14 to 21/04/14)

Summary of Previous Iteration

Objective	Progress
Developed and optimised variants of Pyramid Tree	Complete
Implemented dynamic kd-tree implementation	Complete
Implemented Splay quadtree	Abandoned
Analysed performance of pyramid tree variants and baselines in detail	Complete
Wrote up and explained all variants and provided timings up to the end of the first iteration	Complete
Wrote up and explained all variants and provided timings up to the end of the second iteration	Not Complete

Structures Developed

Structure	Description
Pyramid Tree	As described by Berchold et. Al in 1998 paper, except using hash table instead of B+-tree
Pseudo-Pyramid Tree	Simplified version of Pyramid tree present in vtkPyramidTree class given. Uses a point's distance from the boundaries to compute a hash value
Hash Table	Uses boost's hashing function to give almost-unique hash to every point
KD-Tree	Simple kd-tree implementation where dimensions are cycled through at each level as cutting dimension. Single point at each node

Donald Whyte
sc10dw

Performance Summary

Structure	**Description**
Small Synthetic Data	Pseudo-Pyramid tree provides best performance
Large Synthetic Data	Hash table provides best performance, followed by pyramid tree variants and then kd-tree (twice as slow)
Astrophysics	Pyramid tree and Pseudo-pyramid tree perform the slowest, with as many as 200,000 points in one bucket! Hash table fastest, followed by kd-tree
Hurricane Isabel	*Not performed yet (currently converting data to desired format)*
3D Mesh	Hash table, pyramid tree, pseudo-pyramid tree, kd-tree
Program Trace (active-w4)	kd-tree performs best, followed by pyramid tree, hash table, then pseudo-pyramid tree (most likely due to Pyramid tree being more expensive to reset – more *mallocs*??)
Program Trace (centers-w4)	Pyramid tree and hash table roughly same performance, being 2.46 times faster than kd-tree. Pseudo-pyramid tree fastest however, being ~3.5 times faster (2 seconds compared to PT's 7 seconds)

Donald Whyte
sc10dw

New Items to Discuss

- Final report structure discussion
  - table of contents run-through
  - footnotes allowed in project?
- Current datasets for evaluation:
  - Uniformly Randomly Distributed Data (10,000 points, varying *d*)
  - Uniformly Randomly Distributed Data (16 dimensions, varying size)
  - Astrophysics Data (10 dimensions, randomly sampled from full set)
  - Hurricane Isabel (13 dimensions, randomly sampled from full set)
  - Program Trace (active-w4 and centers-w4)
  - which datasets to focus on?
  - too many tables/plots? How in-depth to go?
- Level of detail of report in general
  - is current detail too much?
- Recording decision not to use Splay quadtree and justifying it in report
  - was stated I would implement it in mid-project report
  - iteration #1/#2 or methodology?
- Reasons for Pyramid tree performing so poorly on astrophysics data
- Reasons for Pyramid tree and hash table performing poorly on active program trace
- Idea for a new structure to combat problematic astrophysics data:
  - Recursive Pyramid Tree
- What to do next – objectives for this week!
  - only four weeks of project left, including write-up
  - worth optimising kd-tree further?
    Worth implementing recursive pyramid tree?
- Possible to open-source code after project is released?