Assignment 2 - CS618A

Sumedh Masulkar
sumedh@,11736

January 24, 2015

1 System Configuration

- Processor: Intel(R) Core(TM) i3-2310M CPU @ 2.10GHz
- Memory: 3 GB

2 Results

Following table shows time statistics for Region Quad Tree and Region K-d Tree after insertion of 10^6 nodes in each tree and followed by 5000 queries. All values are in milliseconds.

- Region Quad Tree

	Minimum	Maximum	Mean	Standard Deviation
Insertion	0.001	23.944	0.048866	0.889071
Point Query	0.001	0.009	0.002821	0.000980285
Range Query	0.005	435.037	218.386	125.267
kNN Query	0.161	41.149	17.506	9.1818
Window Query	0.011	278.619	36.0038	38.4292

- Region K-d Tree

	Minimum	Maximum	Mean	Standard Deviation
Insertion	0.002	0.033	0.005842	0.00251019
Point Query	0.002	0.013	0.00434	0.00143192
Range Query	0.005	514.071	250.154	144.087
kNN Query	0.174	58.104	22.7776	12.0539
Window Query	0.013	273.633	39.2607	41.7373

3 Observations

- For Quad Tree, we observe, Range Query > Window Query > kNN Query > Point Query > Insertion.

- For K-d Tree, we observe, Range Query > Window Query > kNN Query > Insertion > Point Query.
- Insertion and point query take almost same time. This should be because they involve similar traversal of trees from root to leaf.
- Range Query is costly amongst all as it involves finding intersection of circle with a region(rectangle), similarly window query takes a lot of time finding intersection of regions with given rectangle in addition to traversing the nodes.
- kNN Query is expensive too, as it involves implementation of a priority queue, the managing of queue consumes a lot of computaion time, but it is lesser than range queries and window queries, may be since it is bounded by k.
- For given queries, Quad Tree and K-d Tree performs almost the same. But still Quad Tree performs slightly better for range, kNN, window queries. This may be as branching factor is more in Quad Tree thus resulting in lesser height.