

三维点云算法

Date: 2020.07.11

Author: MagicTZ

Homework: 2

Description:

The implementation of tree algorithm regarding:

1. Construction of octree-tree and kd-tree
2. Search algorithms: KNN, radius search and brute search
3. The time comparison between different search algorithms in different tree structures using API or function written by myself

1 Results

(1) 从 124668 个数据点中对前 10000 个点进行搜索

```
搜索数据集中的 10000 个点 :
-----Octree -----
The sum of the points: 124668
Octree: build 6510.755, knn 21331.191, radius 65443.309, brute 98443.819
-----Kdtree -----
The sum of the points: 124668
Kdtree: build 195.246, knn_scipy 9297.525, knn 28460.773, radius_scipy 103821.255, radius 46487.501, brute 98011.808
```

	Octree [ms]	Kdtree [ms]
Build	6510.755	195.246
Knn	21331.191	28460.773
Radius	65443.309	46487.501
Knn_scipy	-	9297.525
Radius_scipy	-	103821.255
Brute	98443.819	98011.808

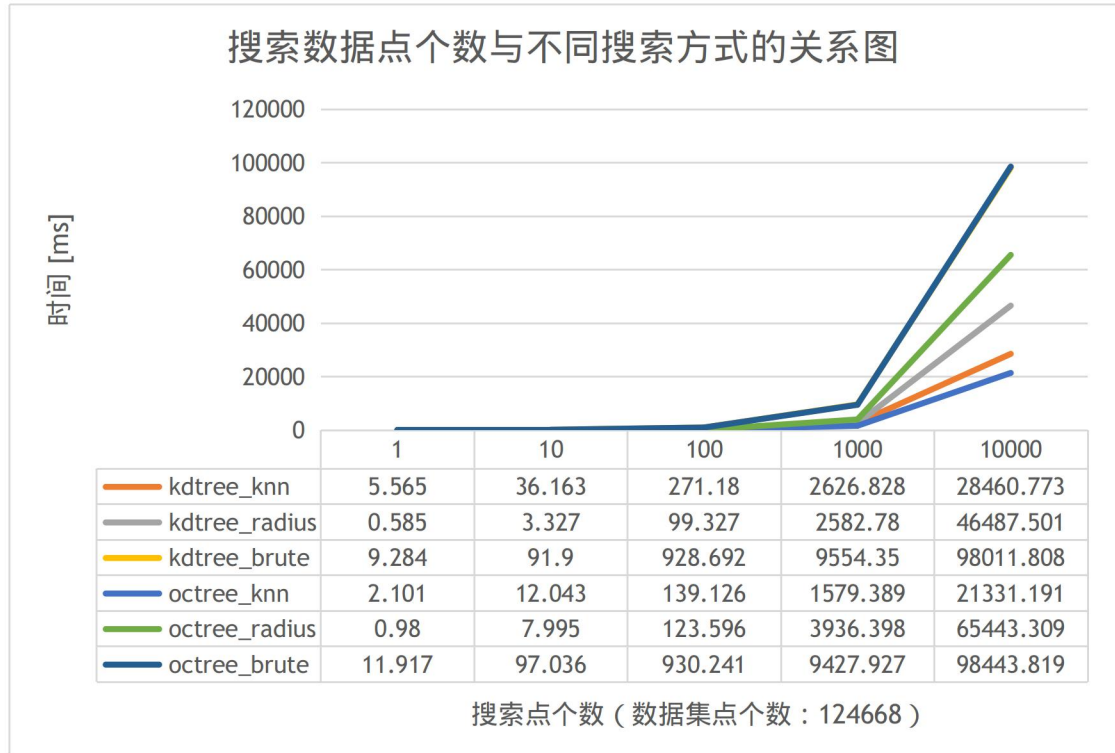
表格 1 不同建树方式和搜索策略的时间开销表格（索引点个数：10000；数据集点个数：124668）

(2) 对原有数据进行随机采样，得到 3 万个数据点后，对这 3 万个数据点进行搜索

```
-----Octree -----
The sum of the points: 30000
Octree: build 1370.373, knn 56806.823, radius 141886.272, brute 80459.834
-----Kdtree -----
The sum of the points: 30000
Kdtree: build 46.722, knn_scipy 23530.151, knn 93516.768, radius_scipy 235239.711, radius 97622.983, brute 81872.517
```

	Octree [ms]	Kdtree [ms]
Build	1370.373	46.722
Knn	56806.823	93516.768
Radius	141886.272	97622.983
Knn_scipy	-	23530.151
Radius_scipy	-	235239.711
Brute	80459.834	81872.517

表格 2 不同建树方式和搜索策略的时间开销表格（索引点个数：30000；数据集点个数：30000）



2 Result Analysis

从上面图表可以发现,随着索引数据量的增加,暴力搜索的方式明显不能很高效地处理数据,又结果可知,从搜索方式进行比较,使用 knn 方式对进行搜索的时间花销要明显比使用 radius 搜索的时间花销要小;从数据结构进行比较,octree 相较于 kdtree 而言,虽然前者在构建的过程中,需要比较大的时间花销,但是对数据进行搜索的时候,明显能够节省更多的时间,尤其体现在数据量非常大的时候。

总而言之,数据量较小,使用 kdtree,搜索方式 knn 和 radius 都可以;数据量较大,使用 octree 建树,并使用 knn 进行数据搜索。