多核程序设计与实践期末项目展示

高维空间中的最近邻

施天予　王郅成　王筝

中山大学计算机学院（软件学院）

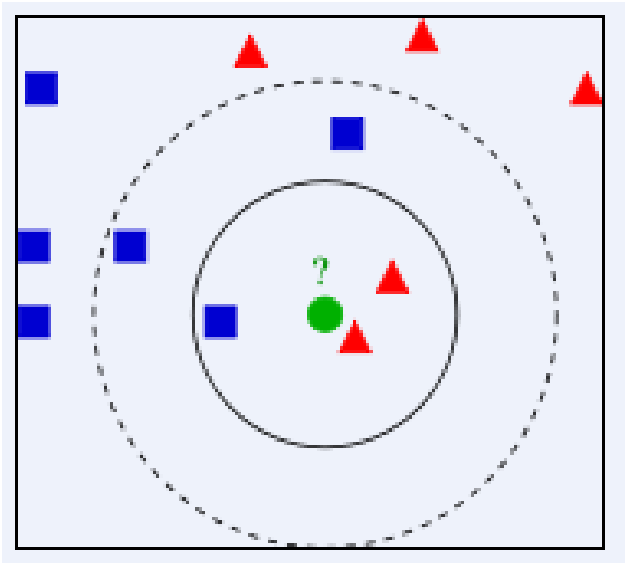
2022 年 6 月

中山大學
SUN YAT-SEN UNIVERSITY

问题描述

最近邻搜索 (NNS)

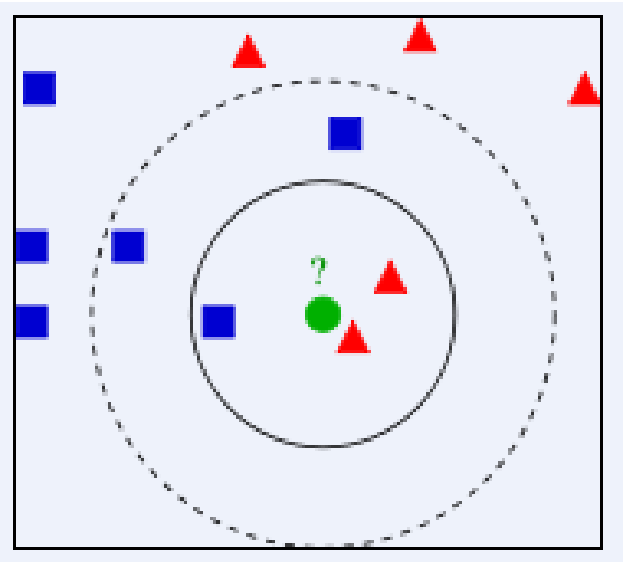
又称最近点搜索，是一个在尺度空间中寻找最近点的优化问题.



Version 0

CPU 串行版本

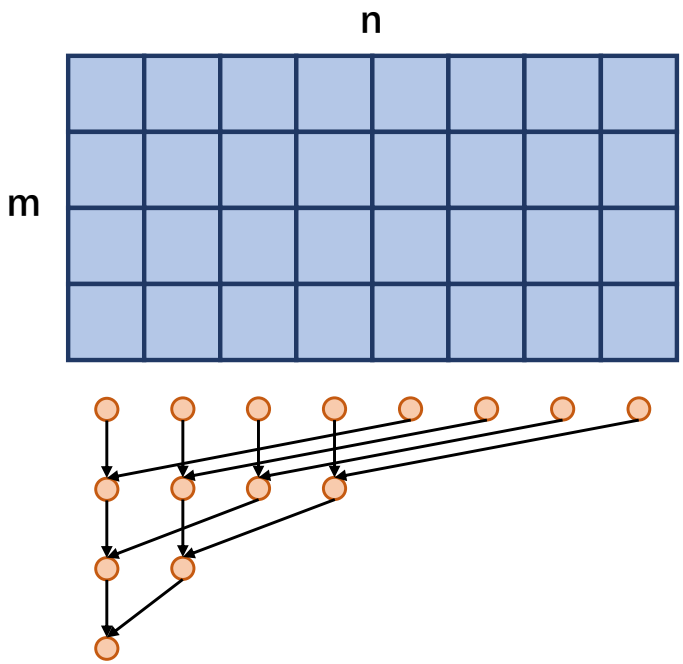
同时可以看作是验证 CUDA 计算正确性的函数.



Version 1

GPU 计算距离矩阵 + 共享内存树形归约

1. 核函数计算所有查询点到目标点的大小为 $m \times n$ 的距离矩阵.
2. 对距离矩阵的 m 行分别找出最小值点, 并保存对应下标.

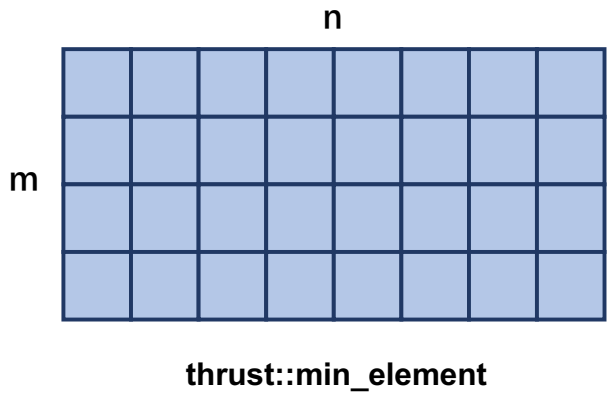


Version 2

GPU 使用 Thrust 库函数

Thrust 类似 STL.

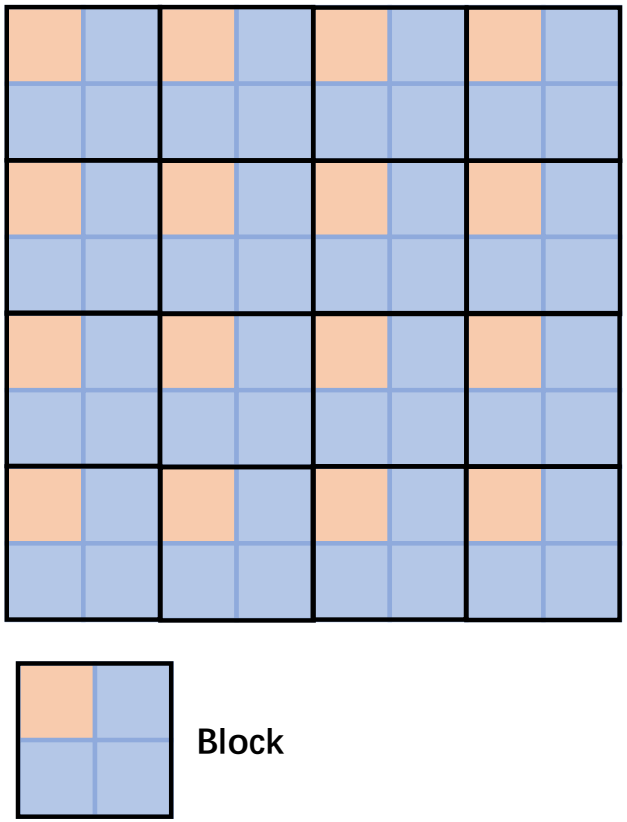
- thrust::device_vector 与全局内存数组开销一致
- thrust::min_element



Version 3

GPU 计算距离并同时归约

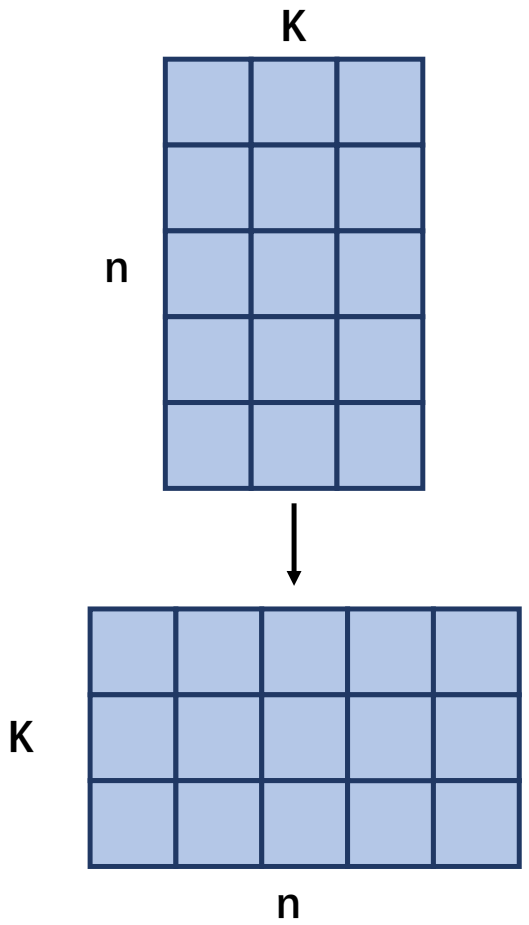
在每个线程计算距离的同时先进行一次线程内归约，再用共享内存树形归约.



Version 4

GPU AoS2SoA

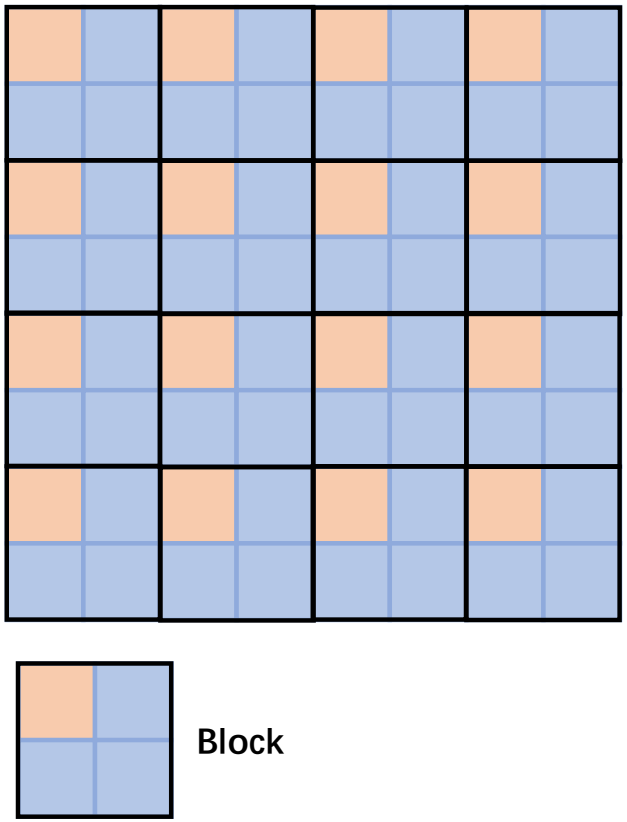
将点的存储方式从
Array of Structures (AoS)
改为
Structure of Arrays (SoA).



Version 5

GPU 使用纹理内存存储目标点集

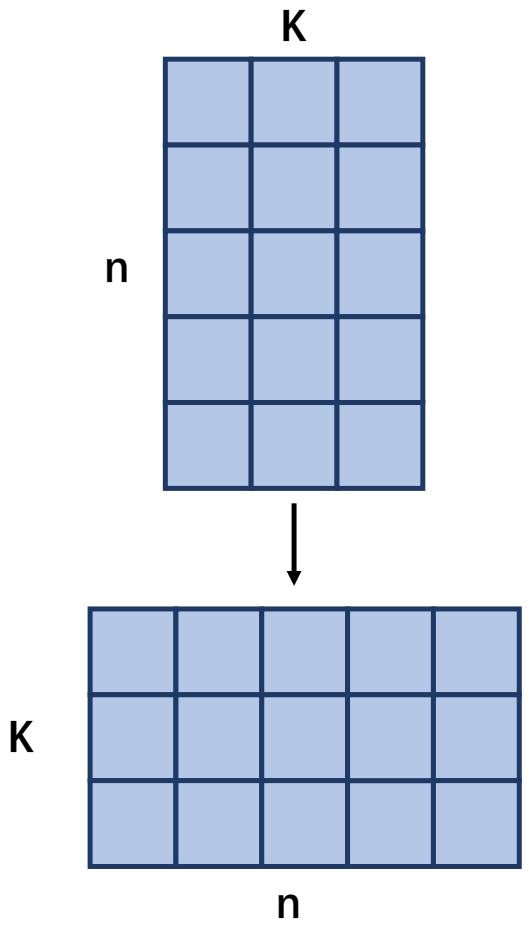
- 2D 纹理内存
- 空间局部性，避免额外转置



Version 6

GPU 使用常量内存存储查询点集

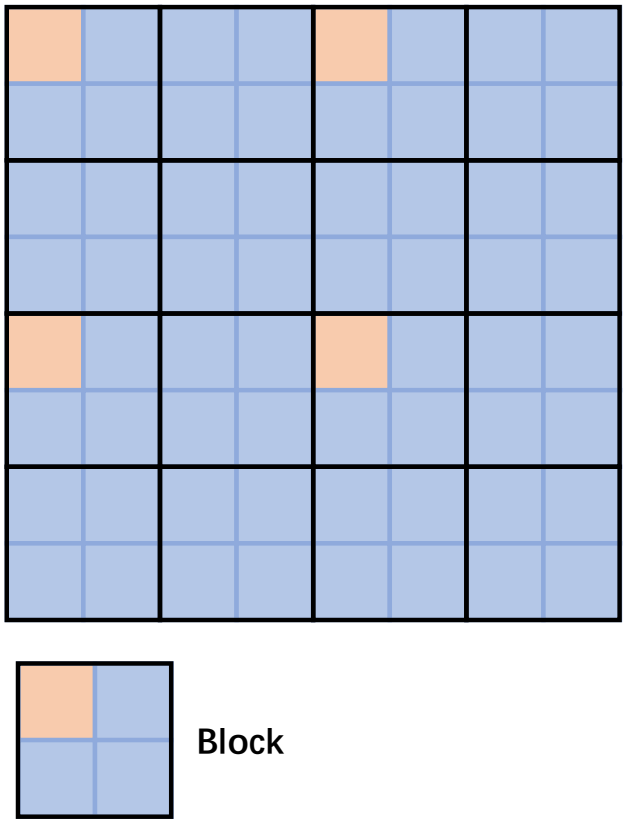
- 每个 block 对应一个查询点，适合常量内存基于 block id 广播访问
- 目标点集仍采用 SoA 形式存储



Version 7

GPU 使用多个 Block 计算单个查询点

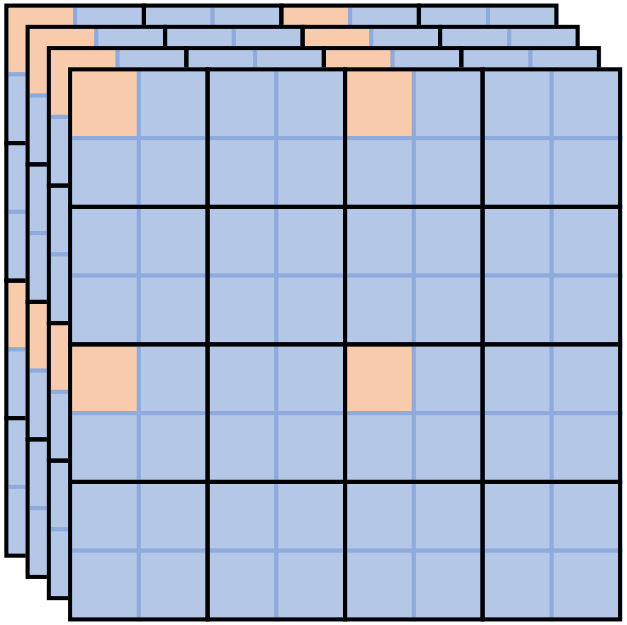
- 查询点较少时启动 block 也少，浪费资源
- 在 Version 4 的基础上，对每个查询点用多个 block 进行查询
- cudaOccupancyMaxActiveBlocksPerMultiprocessor



Version 8

GPU 多卡归约目标点集

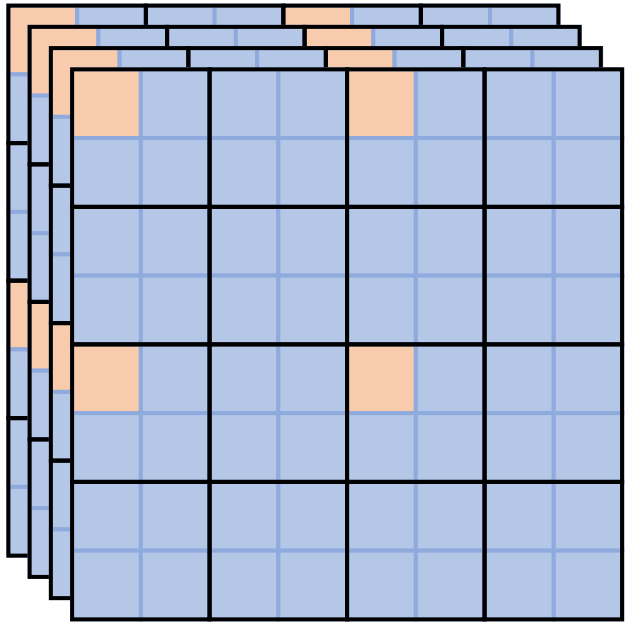
在 Version 7 的基础上，将目标点集均分到多张显卡上.



Version 9

GPU 完全循环展开

在 Version 8 的基础上，将共享内存树形归约中的 for 循环完全展开.



Version 10

KD-tree 查找

用 KD-tree 的树形策略替代线性搜索.

- 建立 KD-tree
- 查询 KD-tree
- CUDA 实现 KD-tree

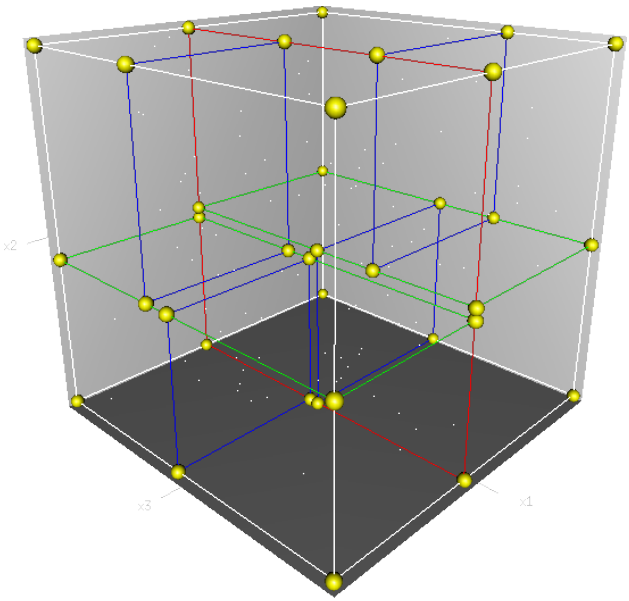


图 1: 3D KD-tree 示意图

Version 11

Octree 查找

用 Octree 的树形策略替代线性搜索.

- 计算树根中心点位置和相应的半径.
- 建立 Octree
- 查询 Octree
- CUDA 实现 Octree

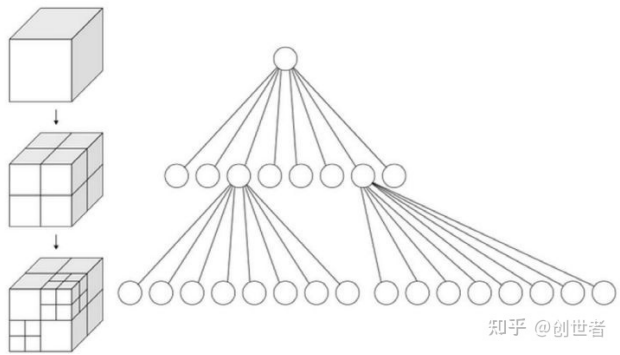


图 2: oc-tree 示意图

测试数据

参数介绍

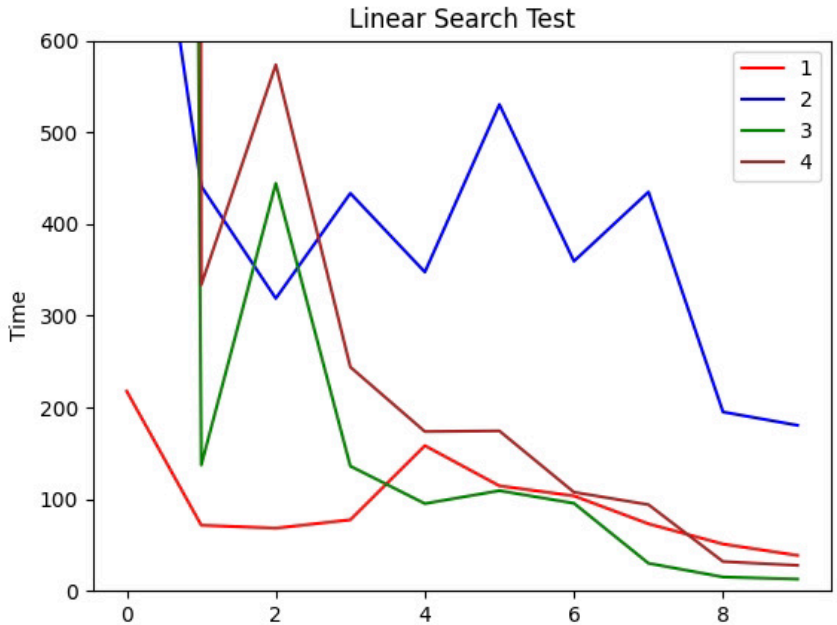
- **K**: 空间维度
- **m**: 查询点数量
- **n**: 目标点数量

id	k	m	n
1	3	1	16777216
2	16	1	16777216
3	3	1024	1048576
4	16	1024	1048576
5	3	1024	65536
6	16	1024	65536

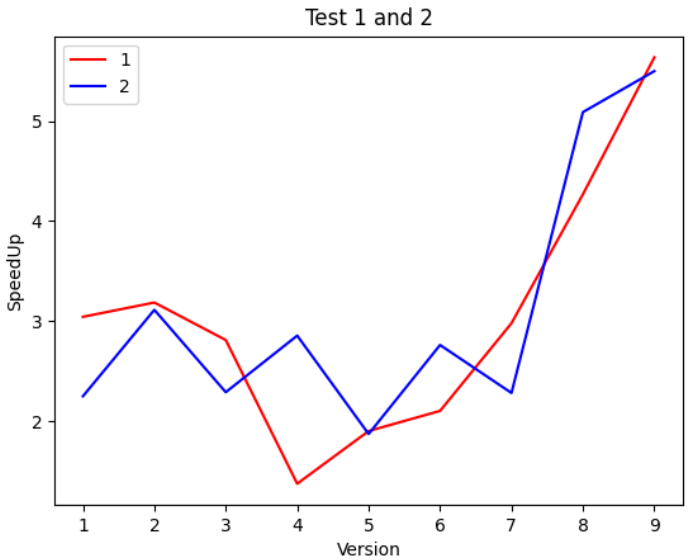
实验结果

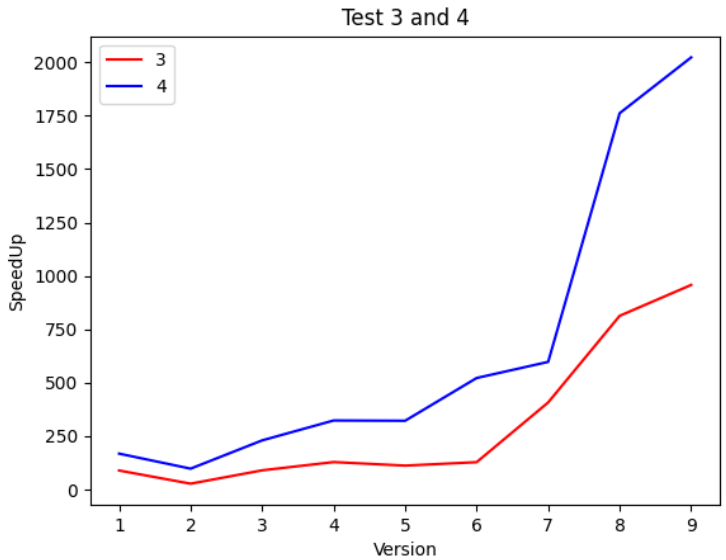
线性查找（单位：ms）

id	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	217.659	71.519	68.312	77.431	158.297	114.476	103.508	73.057	50.997	38.612
2	992.546	441.212	318.814	433.462	347.600	530.248	359.411	334.903	195.009	180.473
3	12248.817	137.139	444.331	135.974	95.110	109.187	95.472	30.037	15.060	12.787
4	56173.148	333.739	533.652	244.009	173.665	174.367	107.556	93.974	31.884	27.767



加速比


Test 1 and 2
1
2
SpeedUp
Version
1
2
3
4
5
6
7
8
9
2
3
4
5


Test 3 and 4
3
4
SpeedUp
Version
0
250
500
750
1000
1250
1500
1750
2000
1
2
3
4
5
6
7
8
9

综合对比（单位：ms）

Time / Version	Test5(k=3)		Test6(k=16)	
	Search	Total	Search	Total
V0(Linear CPU)	174.035	174.035	731.452	731.452
V7(Linear GPU)	1.117	1.117	3.442	3.442
V10(KD-tree CPU)	1.239	19.351	2612.25	2658.92
V10(KD-tree GPU)	0.443	18.373	24.877	60.398
V11(Octree CPU)	20.439	140.368	-	-
V11(Octree GPU)	0.978	121.373	-	-

Thanks!