深蓝学院
shenlanxueyuan.com

自动驾驶中的SLAM技术

第四章作业思路提示

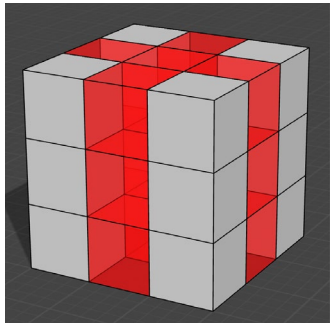
主讲人 陈梓杰

第一题

深蓝学院 shenlanxueyuan.com

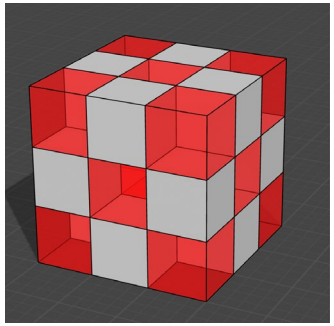
1. 在三维体素中定义NEARBY14, 实现14格最近邻查找

NEARBY14有很多中实现方式, 任意找14个邻近体素都叫NEARBY14:



NEARBY14_1:

```
nearby_grids_ = {KeyType(0, 0, 0),  KeyType(-1, 0, 0), KeyType(1, 0, 0),   KeyType(0, 1, 0),
                 KeyType(0, -1, 0), KeyType(0, 0, -1), KeyType(0, 0, 1),   KeyType(1, 0, 1),
                 KeyType(1, 0, -1), KeyType(-1, 0, 1), KeyType(-1, 0, -1), KeyType(0, 1, 1),
                 KeyType(0, 1, -1), KeyType(0, -1, 1), KeyType(0, -1, -1)};
```



NEARBY14_2:

```
nearby_grids_ = {KeyType(0, 0, 0),   KeyType(-1, 0, 0),  KeyType(1, 0, 0), KeyType(0, 1, 0),
                 KeyType(0, -1, 0),  KeyType(0, 0, -1),  KeyType(0, 0, 1), KeyType(-1, -1, 1),
                 KeyType(1, -1, 1),  KeyType(-1, 1, 1),  KeyType(1, 1, 1), KeyType(-1, -1, -1),
                 KeyType(1, -1, -1), KeyType(-1, 1, -1), KeyType(1, 1, -1)};
```

第一题

深蓝学院
shenlanxueyuan.com

注意修改代码中的条件判断, 不然调用不了NEARBY_14

```
} else if (dim == 3 && (nearby_type_ != NearbyType::NEARBY6 && nearby_type_ != NearbyType::CENTER &&
                        nearby_type_ != NearbyType::NEARBY14)) {      You, 17 seconds ago • Uncommitte
    LOG(INFO) << "3D grid does not support nearby4/8, using nearby6 instead.";
    nearby_type_ = NearbyType::NEARBY6;
}
```

第一题

深蓝学院 shenlanxueyuan.com

结果分析

```
172] ==================
:32] 方法 Grid6 3D 多线程 平均调用时间/次数: 1.23227/10 毫秒.
66] truth: 18779, esti: 18779
93] precision: 0.911339, recall: 0.415997, fp: 760, fn: 10967
178] ==================
:32] 方法 Grid14 3D 单线程 平均调用时间/次数: 19.7142/10 毫秒.
66] truth: 18779, esti: 8933
93] precision: 0.908653, recall: 0.432238, fp: 816, fn: 10662
184] ==================
:32] 方法 Grid14 3D 多线程 平均调用时间/次数: 1.99317/10 毫秒.
66] truth: 18779, esti: 18779
93] precision: 0.908653, recall: 0.432238, fp: 816, fn: 10662
```

NEARBY14_1:

```
:173] ==================
n:32] 方法 Grid6 3D 多线程 平均调用时间/次数: 1.22435/10 毫秒.
:67] truth: 18779, esti: 18779
:94] precision: 0.911339, recall: 0.415997, fp: 760, fn: 10967
:179] ==================
n:32] 方法 Grid14 3D 单线程 平均调用时间/次数: 19.7359/10 毫秒.
:67] truth: 18779, esti: 9116
:94] precision: 0.88394, recall: 0.429096, fp: 1058, fn: 10721
:185] ==================
n:32] 方法 Grid14 3D 多线程 平均调用时间/次数: 2.07747/10 毫秒.
:67] truth: 18779, esti: 18779
:94] precision: 0.88394, recall: 0.429096, fp: 1058, fn: 10721
```

NEARBY14_2:

分析:

① NEARBY14_1准确率和召回率都略高于NEARBY14_2. 原因是NEARBY14_1中新增的8个体素相比与NEARBY14_2中的8个更接近中心点

② NEARBY14准确率低于NEARBY6. 原因是NEARBY14检测体素变多, 误检次数可能变多, 从而导致FP上升

③ NEARBY14召回率高于NEARBY6. 原因是NEARBY14检测体素变多, 漏检最近邻数量变少, 从而FN下降

④ NEARBY14耗时高于NEARBY6. 原因是NEARBY14检测体素多于NEARBY6

第二题

$$d^* = \arg\max_d ||Ad||_2^2$$

解

$$d^* = \arg\max_d ||Ad||_2^2 = \arg\max_d d^\top A^\top Ad, s.t. ||A|| = 1 \tag{1}$$

将矩阵$A^\top A$进行特征值分解, 有$A^\top A = V\Lambda V^{-1}$. V为正交矩阵, $V = [v_1, v_2, \ldots, v_n]$, 记$v_1, v_2, \ldots, v_n$构成一组单位正交基. 那么任意d可以被这组正交基线表出:

$$d = a_1 v_1 + a_2 v_2 + \ldots + a_n v_n$$

进一步地,

$$V^{-1}d = V^\top d = [a_1, a_2, \ldots, a_n]^\top \tag{2}$$

将(2)代入(1), 有:

$$\begin{aligned}
||Ad||_2^2 &= d^\top A^\top Ad \\
&= d^\top V\Lambda V^{-1} d \\
&= (V^{-1}d)^\top \Lambda (V^{-1}d) \\
&= [a_1, a_2, \ldots, a_n] \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_3 \end{bmatrix} \\
&= \lambda_1 a_1^2 + \lambda_2 a_2^2 + \ldots + \lambda_n a_n^2 = \sum_{k=1}^{n} \lambda_k a_k^2
\end{aligned}$$

而$||d|| = 1$, 意味着$a_1^2 + a_2^2 + \ldots + a_n^2 = 1$. 假设特征值$\lambda_k$是降序排列的. 要使(1)最大化, 则取$a_1 = 1, a_2, \ldots, a_n = 0$, 那么$d^* = v_1$, 即最优解为最大特征值向量.

第三题

将本节的最近邻算法与一些常见的近似最近邻算法进行对比，比如nanoflann，给出精度指标和时间效率指标

思路:

① PCL格式的点云转换成nanoflann支持的格式

② 初始化

```
nanoflann::KDTreeSingleIndexAdaptor<nanoflann::L2_Simple_Adaptor<double, PointCloud<double>>,PointCloud<double>, 3>;
```

③ 调用knnSearch()函数做KNN搜索

④ 使用par_unseq做并发

第三题

深蓝学院 shenlanxueyuan.com

实验结果

```
I20230611 10:16:40.717145  2676 sys_utils.h:32] 方法 Kd Tree build 平均调用时间/次数: 12.3001/1 毫秒.
I20230611 10:16:40.717408  2676 test_nn.cc:244] Kd tree leaves: 18869, points: 18869
I20230611 10:16:40.722388  2676 sys_utils.h:32] 方法 Kd Tree nanoflann build 平均调用时间/次数: 4.96259/1 毫秒.
I20230611 10:16:43.052353  2676 sys_utils.h:32] 方法 Kd Tree 5NN 多线程 平均调用时间/次数: 4.51399/1 毫秒.
I20230611 10:16:43.052465  2676 test_nn.cc:67] truth: 93895, esti: 93895
I20230611 10:16:48.846935  2676 test_nn.cc:94] precision: 1, recall: 1, fp: 0, fn: 0
I20230611 10:16:48.868860  2676 sys_utils.h:32] 方法 Kd Tree nanoflann 5NN 单线程 平均调用时间/次数: 21.8815/1 毫秒.
I20230611 10:16:48.868894  2676 test_nn.cc:67] truth: 93895, esti: 93895
I20230611 10:16:54.642343  2676 test_nn.cc:94] precision: 1, recall: 1, fp: 0, fn: 0
I20230611 10:16:54.645082  2676 sys_utils.h:32] 方法 Kd Tree nanoflann 5NN 多线程 平均调用时间/次数: 2.70152/1 毫秒.
I20230611 10:16:54.645105  2676 test_nn.cc:67] truth: 93895, esti: 93895
I20230611 10:17:00.440696  2676 test_nn.cc:94] precision: 1, recall: 1, fp: 0, fn: 0
I20230611 10:17:00.440730  2676 test_nn.cc:275] building kdtree pcl
I20230611 10:17:00.457770  2676 sys_utils.h:32] 方法 Kd Tree build 平均调用时间/次数: 6.97182/1 毫秒.
I20230611 10:17:00.457814  2676 test_nn.cc:280] searching pcl
I20230611 10:17:00.496543  2676 sys_utils.h:32] 方法 Kd Tree 5NN in PCL 平均调用时间/次数: 38.6346/1 毫秒.
I20230611 10:17:00.497155  2676 test_nn.cc:67] truth: 93895, esti: 93895
I20230611 10:17:06.303273  2676 test_nn.cc:94] precision: 1, recall: 1, fp: 0, fn: 0
I20230611 10:17:06.303306  2676 test_nn.cc:301] done.
```

nanoflann建树和搜索速度都要优于手写kdtree和PCL kdtree

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening