

## PCL之计算点云质心—pcl::compute3DCentroid()

`com put roi`

## 质心计算公式

质心指的是质量的中心，认为是物体质量集中于此点的假想点。

通常物体质心坐标

$$P_{[c]}$$

计算公式如下：

$$P_{[c]} = \frac{1}{M} \sum_{i=0}^n m_{[i]} r_{[i]}$$

其中，

$$r_{[i]} = (x_{[i]}, y_{[i]}, z_{[i]}), i=1, 2, \dots, n$$

为各质点的坐标，

$$m_{[i]}$$

为质点对应的质量。

## PCL函数原理

计算点云质心时，令上述公式中的

$$m_{[i]} = 1$$

即可，则点云质心坐标计算公式如下：

$$P_{[c]} = \frac{1}{n} \left( \sum_{i=0}^n x_{[i]}, \sum_{i=0}^n y_{[i]}, \sum_{i=0}^n z_{[i]} \right)$$

## PCL函数实现

```
1 Eigen::Vector4f centroid; //质心
2 pcl::compute3DCentroid(*cloud_smoothed, centroid); // 计算质心
```

该函数的原理即是使用上述公式计算点云质心坐标，接下来通过代码进行验证。

## 代码实现

```
1 #include <iostream>
2 #include <Eigen/Core>
3 #include <pcl/io/pcd_io.h>
4 #include <pcl/point_types.h>
5 #include <pcl/common/transforms.h>
6 #include <pcl/visualization/pcl_visualizer.h>
7
8 using namespace std;
9 typedef pcl::PointXYZ PointT;
10
11 int main(int argc, char **argv)
12 {
13     // 导入点云
14     pcl::PointCloud<PointT>::Ptr cloud(new pcl::PointCloud<PointT>);
15     // 点云文件地址:https://github.com/PointCloudLibrary/data/blob/master/tutorials/table_
16     if (pcl::io::loadPCDFile(" table_scene_1ms400.pcd", *cloud) == -1) { // 读取.pcd文件
17         cerr << "can't read file table_scene_1ms400.pcd" << endl;
18         return -1;
19     }
20
21     // PCL函数计算质心
22     Eigen::Vector4f centroid; // 质心
23     pcl::compute3DCentroid(*cloud, centroid); // 齐次坐标. (c0,c1,c2,1)
24
25     // 按公式计算质心
26     PointT p_c;
27     p_c.x = 0; p_c.y = 0; p_c.z = 0;
28     for (auto p : cloud->points) {
29         p_c.x += p.x;
30         p_c.y += p.y;
31         p_c.z += p.z;
32     }
33
34     p_c.x /= cloud->points.size();
35     p_c.y /= cloud->points.size();
36     p_c.z /= cloud->points.size();
37
38     // 结果对比
39     cout << "pcl计算点云质心结果:" << centroid << endl;
40     cout << "按照公式计算点云质心结果:" << p_c << endl;
41     // 可视化
42     pcl::visualization::PCLVisualizer viewer;
43     viewer.addPointCloud(cloud);
44     viewer.addCoordinateSystem();
45     // 质心坐标
46     PointT center;
47     center.x = centroid(0);
48     center.y = centroid(1);
49     center.z = centroid(2);
50
51     viewer.addSphere(center, 0.03, 1, 0, 0, "sphere", 0);
52     while (!viewer.wasStopped())
53     {
54         viewer.spinOnce(100);
55     }
56     return 0;
57 }
```

## ● 结果

```
pcl计算点云质心结果: -0.0994279
-0.307723
-1.35414
1
按照公式计算点云质心结果: (-0.0994279,-0.307723,-1.35414)
```



