

PCL学习笔记——NormalEstimation估计点云的法向量

主要Classes:

`pcl::NormalEstimation< PointInT, PointOutT > Class`

头文件:

`#include <pcl/features/normal_3d.h>`

Code:

```

1 // overlapped_points_normal_estimation.cpp: 定义控制台应用程序的入口点。
2 //
3
4 #include "stdafx.h"
5 #include<iostream>
6 #include<fstream>
7 #include<pcl/io/pcd_io.h>
8 #include<pcl/point_types.h>
9 #include<pcl/point_cloud.h>
10 #include<pcl/kdtree/kdtree_flann.h>
11 #include<pcl/kdtree/io.h>
12 #include<vector>
13 #include<pcl/features/normal_3d.h>
14
15 using namespace std;
16
17 int main()
18 {
19     pcl::PointCloud<pcl::PointXYZ>::Ptr cloudA(new pcl::PointCloud<pcl::Poi
20     pcl::PointCloud<pcl::PointXYZ>::Ptr cloudB(new pcl::PointCloud<pcl::Poi
21     pcl::PointCloud<pcl::PointXYZ>::Ptr overlapA(new pcl::PointCloud<pcl::F
22     pcl::PointCloud<pcl::PointXYZ>::Ptr overlapB(new pcl::PointCloud<pcl::F
23
24
25     if (pcl::io::loadPCDFile<pcl::PointXYZ>("pointA.pcd", *cloudA) == -1)
26     {
27         PCL_ERROR("Couldn't read the pointA.pcd\n");
28         return -1;
29     }
30     if (pcl::io::loadPCDFile<pcl::PointXYZ>("pointB.pcd", *cloudB) == -1)
31     {
32         PCL_ERROR("Couldn't read the pointB.pcd\n");
33         return -1;
34     }
35     if (pcl::io::loadPCDFile<pcl::PointXYZ>("overlapA.pcd", *overlapA) == -
36     {
37         PCL_ERROR("Couldn't read the pointB.pcd\n");
38         return -1;
39     }
40     if (pcl::io::loadPCDFile<pcl::PointXYZ>("overlapB.pcd", *overlapB) == -
41     {
42         PCL_ERROR("Couldn't read the pointB.pcd\n");
43         return -1;
44     }
45     //创建法线估计对象，并将输入数据集传递给这个对象
46     pcl::NormalEstimation<pcl::PointXYZ,pcl::Normal>ne1;
47     pcl::NormalEstimation<pcl::PointXYZ, pcl::Normal>ne2;

```

```
48
49     ne1.setInputCloud(overlapA);
50     ne1.setSearchSurface(cloudA);
51
52     ne2.setInputCloud(overlapB);
53     ne2.setSearchSurface(cloudB);
54
55     //以kdtree作为索引方式
56     pcl::search::KdTree<pcl::PointXYZ>::Ptr treeA(new pcl::search::KdTree<p
57     pcl::search::KdTree<pcl::PointXYZ>::Ptr treeB(new pcl::search::KdTree<p
58     ne1.setSearchMethod(treeA);
59     ne1.setSearchMethod(treeB);
60
61     //存储输出数据集
62     pcl::PointCloud<pcl::Normal>::Ptr overlapA_normals(new pcl::PointCloud<
63     pcl::PointCloud<pcl::Normal>::Ptr overlapB_normals(new pcl::PointCloud<
64
65     ne1.setRadiusSearch(0.4);
66     ne2.setRadiusSearch(0.4);
67
68     ne1.compute(*overlapA_normals);
69     ne2.compute(*overlapB_normals);
70
71     pcl::io::savePCDFFileASCII("overlapA_normals.pcd", *overlapA_normals);
72     pcl::io::savePCDFFileASCII("overlapB_normals.pcd", *overlapB_normals);
73
74     return 0;
75 }
76
77
```

Result:

```
overlapA_normals.pcd x overlapB_normals.pcd x
1 # .PCD v0.7 - Point Cloud Data file format
2 VERSION 0.7
3 FIELDS normal_x normal_y normal_z curvature
4 SIZE 4 4 4 4
5 TYPE F F F F
6 COUNT 1 1 1 1
7 WIDTH 31317
8 HEIGHT 1
9 VIEWPOINT 0 0 0 1 0 0 0
10 POINTS 31317
11 DATA ascii
12 -0.95084053 -0.23193903 0.2051989 0.088572256
13 0.99176824 0.094339527 0.086578175 0
14 -0.98646677 -0.15528122 0.052641395 0.060661763
15 0.93574834 0.019538369 0.35212678 0
16 0.95228642 0.19097812 0.23807161 0
17 -0.83992982 -0.35360715 0.41167936 0.1093794
18 0.96962583 0.025914276 0.24321628 0
19 -0.75219899 -0.27060819 0.60080606 0
20 -0.7878927 0.098950416 0.60781068 0
21 -0.96582764 0.15247923 0.20958792 0.054966029
22 -0.44480401 0.029779254 0.89513272 0
23 -0.9776867 0.19290064 0.083174713 0.025742527
24 0.9873293 -0.089407444 0.13109961 0
25 -0.73082024 0.16529368 0.66225356 0
26 -0.81718993 0.17678596 0.54858667 0
27 0.99380898 -0.057656094 0.094971344 0
28 0.96900618 -0.23852459 0.064289831 0
29 0.95195872 -0.30574962 0.017082244 0
30 -0.9242906 0.11712671 0.36327428 0.10433586
31 -0.82500994 0.45360929 0.33704174 0.01293295
32 0.98689371 -0.041893832 0.15583862 0
33 0.98279941 -0.0044520926 0.1846227 0
34 -0.98858988 0.073656999 0.13139459 0.044718735
```

智能推荐

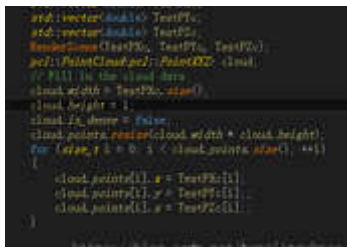


(/article/3736132098/)

PCL (点云库) 的源码安装 (/article/3736132098/)

pcl的官网 方法一(通过PPA安装) (本人没成功) 方法二(源码安装) 安装依赖: 官网: boost、Eigen、FLANN、VTK是必须的 (注意依赖版本) 根据自己环境安装, 我

只安装了必须的依赖 下载: 进入目录、创建build文件夹 创建makefile 默认编译、安装 (-j+线程数) 编译、安装的过程可能会出错, 我遇到的问题: (1) No rule to make target '/...



(/article/8407753023/)

PCL点云的保存和显示 (/article/8407753023/)

1.pcl点云的保存 1.首先将获得的点云x, y, z分别存放在三个vector中 2.pcl 是一个命名空间, 跟std类似, Point Cloud是类模板, <pcl::PointXYZ>是模板类实例化的类型

3cloud.height用来判断是否为有序点云, =1则是无序点云也可以使用如下函数代替: if (!cloud.isOrganized ()), 4. 对于无序点云来说: (...

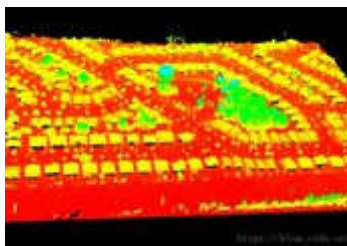


(/article/5575124166/)

PCL学习一 法线估计 (/article/5575124166/)

PCL法线估计 平面的法线是垂直于它的单位向量。在点云的表面的法线被定义为垂直于与点云表面相切的平面的向量。表面法线也可以计算点云中一点的法线, 被认为

是一种十分重要的性质。常常在被使用在很多计算机视觉的应用里面, 比如可以用来推出光源的位置, 通过阴影与其他视觉影响, 表面法线的问题可以近似化解为切面的问题, 这个切面的问题又会变成最小二乘法拟合平面的问题 解决表面法线估计的问题可以最终简化为对一个协方差...

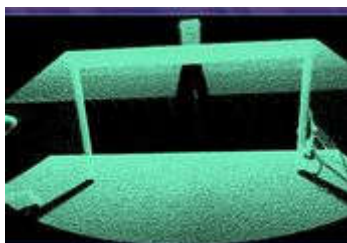


(/article/6499800/)

PCL中点云可视化: 坐标字段、随机、单一颜色、法向量 (/article/6499800/)

PCL中viewer添加并显示的点云过于简单, 现总结常见的几

种点云渲染方式, 便于点云结果的显示。(1) 按照点云坐标x、y、z坐标值中字段给点云进行赋值渲染 按照z高程渲染图: 按照x坐标渲染图: (2) 给点云单独赋予某一颜色 点云设置成绿色的结果: (3) 随机生成颜色 随机生成的颜色结果图: (4) 法向量的颜色表示 法向量估算渲染图: ...



(/article/5699935657/)

点云相关学习总结——估计一个点云的表面法线 (/article/5699935657/)

参考: http://www.pointclouds.org/documentation/tutorials/normal_estimation.php 转自: <https://blog.csdn.net/zhenxin066/article/details/38794195> 说明: 1) 本文

的内容部分来源于网络上, 部分来自本人自己的理解, 总结, 在这里写出来没有任何商业目的, 仅供交流、学习之用, 引用别...

猜你喜欢



(/article/99981109577/)

硬件：宽带猫（光猫）的基础知识 (/article/99981109577/)

1、宽带猫的概念 “猫”，又叫做:调制解调器(英文名 MODEM) (记住，这才是正经名字)，作用是把通过电话线或者光纤进入的信号还原为数据以及把网线产生的数据转换成模拟信号（电信号/光信号），所以，如果是电话线/光纤入户的，都需要安装调制解调器。首先你需要知道的是，常见的信号有两种，数字信号和模拟信号（光信号和电信号）。在计算机里运行的是数字信号，在网线（电话线/光纤）里...



(/article/900054011/)

ueditor编辑器上传文件超时报错 (/article/900054011/)

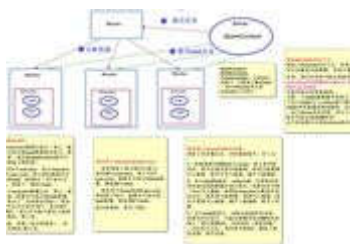
文章来自：源码在线<https://www.shengli.me/php/153.html> php 超时等待时间太长； 解决： php.ini中的max_execution_time为0；（http请求等待时间为永久） ueditor中设置了请求超时，提示：上传失败，请重试！ 解决： 打开webupload.js 修...



(/article/8645717021/)

React开发环境配置 (/article/8645717021/)

一、下载安装编辑器Atom <https://atom.io/> 安装完成：
二、安装插件 在这里首先推荐三个插件： 浏览器浏览功能 open-in-browser（可以设置快捷键） 分页展示html 效果 atom-html-preview（可以设置快捷键） 文件路径补全 autoComplete-path 在Atom上面安装插件步骤...



(/article/14081016626/)

spark性能调优 (/article/14081016626/)

spark性能调优的第一步，就是为任务分配更多的资源，在一定范围内，增加资源的分配与性能的提升是成正比的，实现了最优的资源配置后，在此基础上再考虑进行后面论述的性能调优策略。指定脚本： /usr/opt/modules/spark/bin/spark-submit -class com.atguigu.spark.Analysis -num-executors 80 &nd...



(/article/58751552424/)

面试被问到TCP IP问题 终于被我弄的明明白白的了 (/article/58751552424/)

目录 原理讲解 1.TCP报文格式 2.TCP三次握手 3.TCP四次挥手

面试题 为什么要TIME_WAIT 等待2MLS呢？ 讲一下对tcp/ip协议的理解 为什么连接的时候是三次握手，关闭的时候却是四次握手？ 为什么TIME_WAIT状态需要经过2MSL(最大报文段生存时间)才能返回到CLOSE状态？ 原理讲解 在说TCP IP之前，我们先理解一下报文头是甚麽意思。。。 1.TCP报文格式 ...