

# 从零开始一起学习SLAM | 点云平滑法线估计

## 点云滤波后为什么还需要平滑？

小白：师兄，师兄，上次你说的点云滤波我学会啦，下一步怎么把点云变成网格啊？

师兄：滤波只是第一步，在网格化前我们还需要对滤波后的点云进行平滑（smoothing）

小白：不是已经滤波了吗？怎么还要平滑啊？滤波和平滑不一样吗？

师兄：确实不太一样。我们用RGB-D，激光扫描仪等设备扫描物体，尤其是比较小的物体时，往往会有测量误差。这些误差所造成的不规则数据如果直接拿来曲面重建的话，会使得重建的曲面不光滑或者有漏洞，而且这种不规则数据很难用前面我们提到过的统计分析等滤波方法消除，所以为了建立光滑完整的模型必须对物体表面进行平滑处理和漏洞修复。

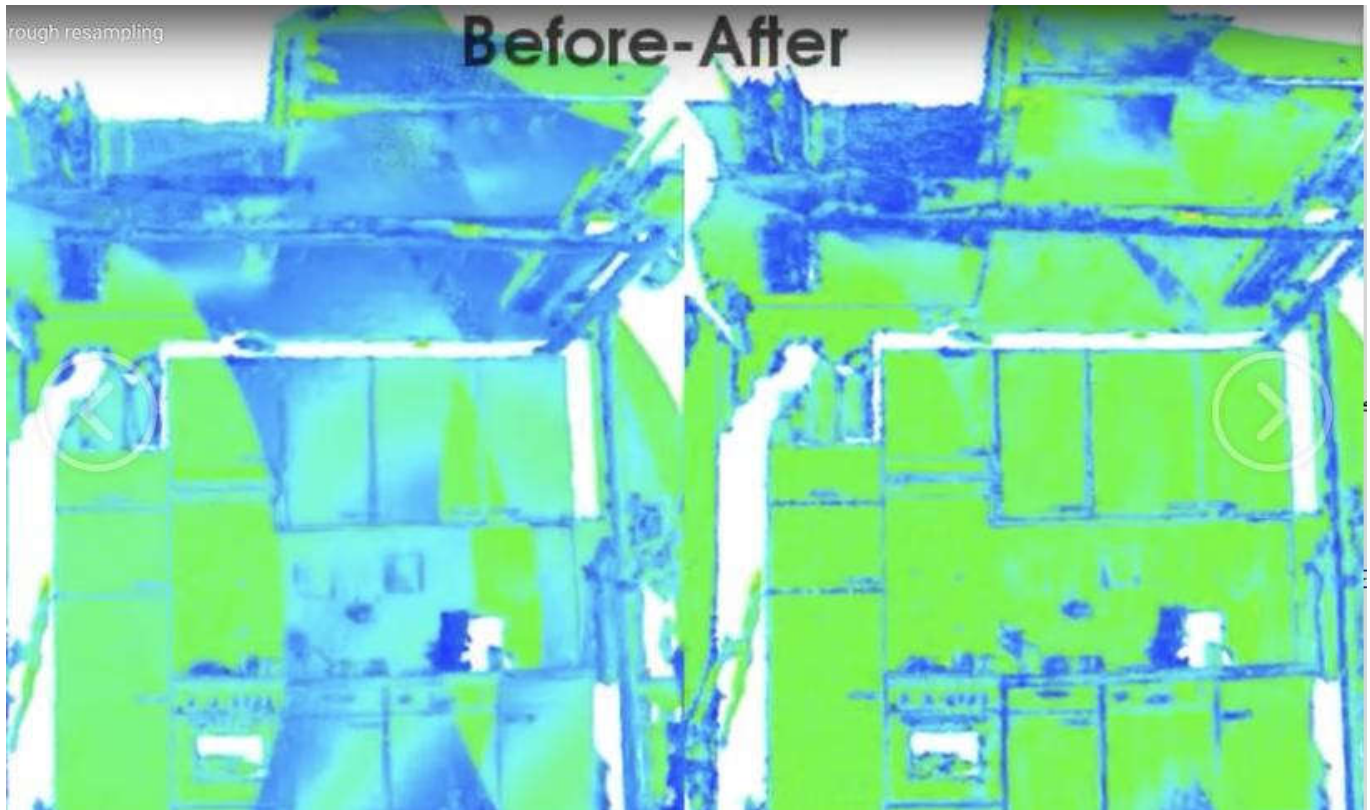
你看下面左边就是原始的扫描数据，右边就是用最小二乘法进行表面平滑后的结果



(<https://i.loli.net/2018/12/28/5c2580b5c1666.png>)

小白：从图上看，平滑确实效果很明显啊，左边杯子上黑色的是噪声吧，右边的结果来看经过平滑都消失了

师兄：对，除了上面说到的设备测量误差外，还有一种情况也需要对点云进行平滑。就是后处理过程中，比如我们对同一个物体从不同方向进行了多次扫描，然后把扫描结果进行配准，最后得到一个完整的模型，但是你配准的结果不一定准啊，比如下图中左侧就是配准后未经过处理的结果，同一面墙壁由于配准误差变成了“两面墙”，并不能完全重叠，你觉得这个数据可以直接用来进行表面重建吗？



(<https://i.loli.net/2018/12/28/5c2580b8d851e.png>)

小白：好坑啊，肯定不行，这样重建出的结果也是两面墙了吧

师兄：对，所以我们需要想办法把“两面墙”变成“一面墙”，如果这时候，我们没有条件重新扫描出更精确的结果，或者配准精度也无法提升，可以通过重采样的方法来实现点云的平滑，从而避免出现这样的问题。

小白：原来这个平滑这么重要啊！怎么用重采样来平滑呢？感觉迫不及待想要学习啦！

师兄：（既然胃口已经被吊起来了）那我们赶快开始切入正题吧

## 如何通过重采样实现点云平滑？

师兄：点云重采样，我们实际上是通过一种叫做“移动最小二乘”（MLS， Moving Least Squares ）法来实现的，对应的类名叫做：pcl::MovingLeastSquares，你知道怎么用吗？

小白：不知道，不过我还记得我们上次师兄给我说的方法，在PCL API documentation <http://docs.pointclouds.org/trunk/> (<http://docs.pointclouds.org/trunk/>) 上查询类名称，就能看到类的定义和用法啦

师兄：活学活用啊，哈哈，那我们现在去查一下看看吧

小白：嗯，我查到了，这个MLS类的定义在这里：

[http://docs.pointclouds.org/trunk/classpcl\\_1\\_1\\_moving\\_least\\_squares.html#a379330b0b1dacao668d165f94930749c](http://docs.pointclouds.org/trunk/classpcl_1_1_moving_least_squares.html#a379330b0b1dacao668d165f94930749c)

([http://docs.pointclouds.org/trunk/classpcl\\_1\\_1\\_moving\\_least\\_squares.html#a379330b0b1dacao668d165f94930749c](http://docs.pointclouds.org/trunk/classpcl_1_1_moving_least_squares.html#a379330b0b1dacao668d165f94930749c))

成员函数好多啊

师兄：对，看着是很多，但是很多我们不常用的，比如我们常用的一个用于重采样的示例代码如下，每行代码都给你注释好了，结合上面网址看很容易理解

```

1 // 对点云重采样
2 pcl::search::KdTree<PointT>::Ptr treeSampling (new pcl::search::KdTree<PointT>); /
3 pcl::PointCloud<PointT> mls_points; //输出MLS
4 pcl::MovingLeastSquares<PointT, PointT> mls; // 定义最小二乘实现的对象mls
5 mls.setComputeNormals (false); //设置在最小二乘计算中是否需要存储计算的法线
6 mls.setInputCloud (cloud_filtered); //设置待处理点云
7 mls.setPolynomialOrder(2); // 拟合2阶多项式拟合
8 mls.setPolynomialFit (false); // 设置为false可以 加速 smooth
9 mls.setSearchMethod (treeSampling); // 设置KD-Tree作为搜索方法
10 mls.setSearchRadius (0.05); // 单位m.设置用于拟合的K近邻半径
11 mls.process (mls_points); //输出

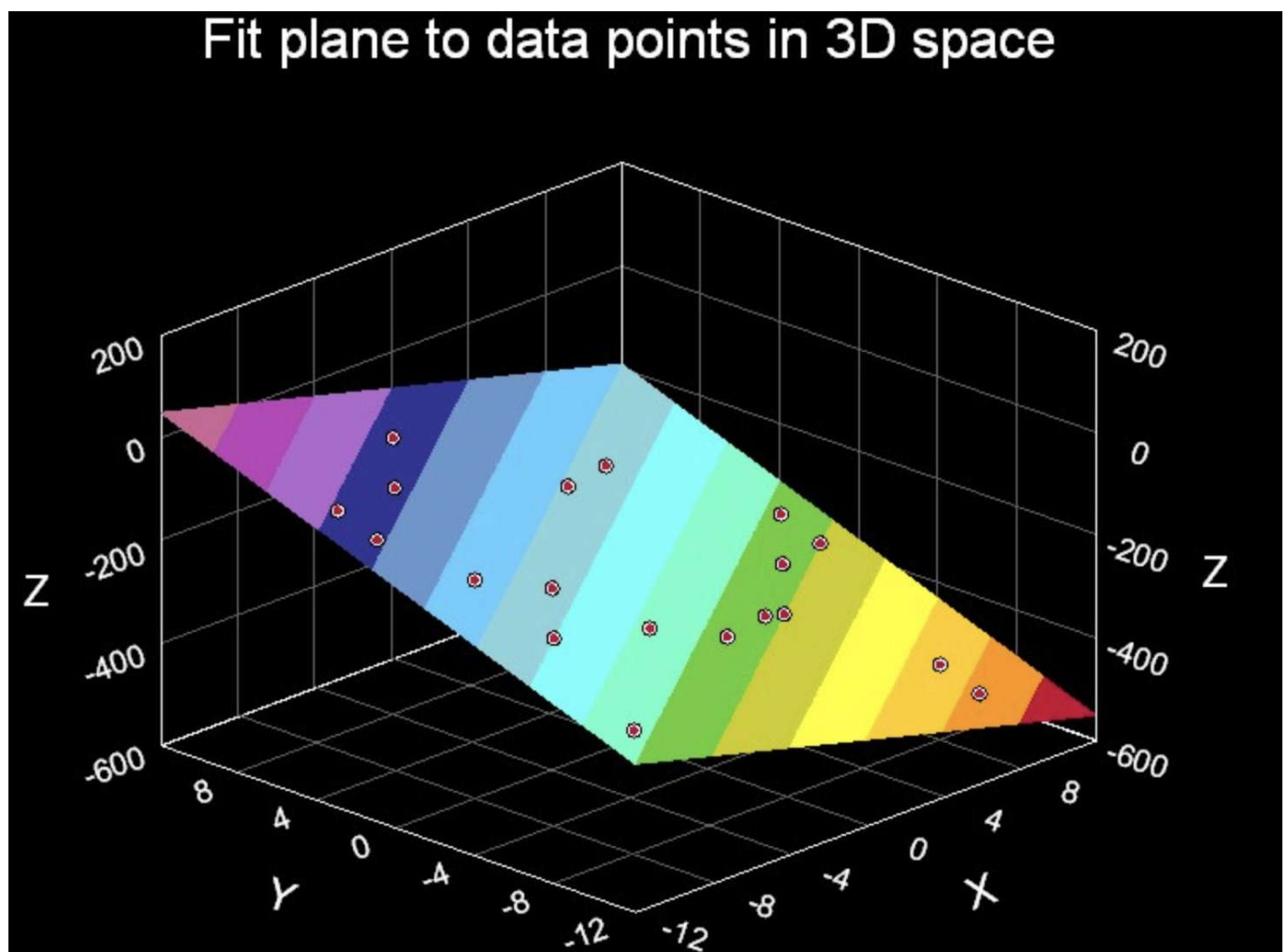
```

小白：师兄，这个代码里的KD-Tree是干嘛的？

师兄：Kd-Tree是一种数据结构，是空间二分树的一种特殊情况，可以很方便的用于进行范围搜索。在这里用KD-Tree就是为了便于管理、搜索点云，这种结构来可以很方便的找到最近邻点。

小白：原来如此，那上面mls.setSearchRadius (0.05) 的意思是不是就是搜索当前点以5cm为半径的空间中所有的点？

师兄：对的，然后把这些点用2阶多项式拟合~



(<https://i.loli.net/2018/12/28/5c2580b7c3d1f.png>)

小白：所以表面就变平滑啦！

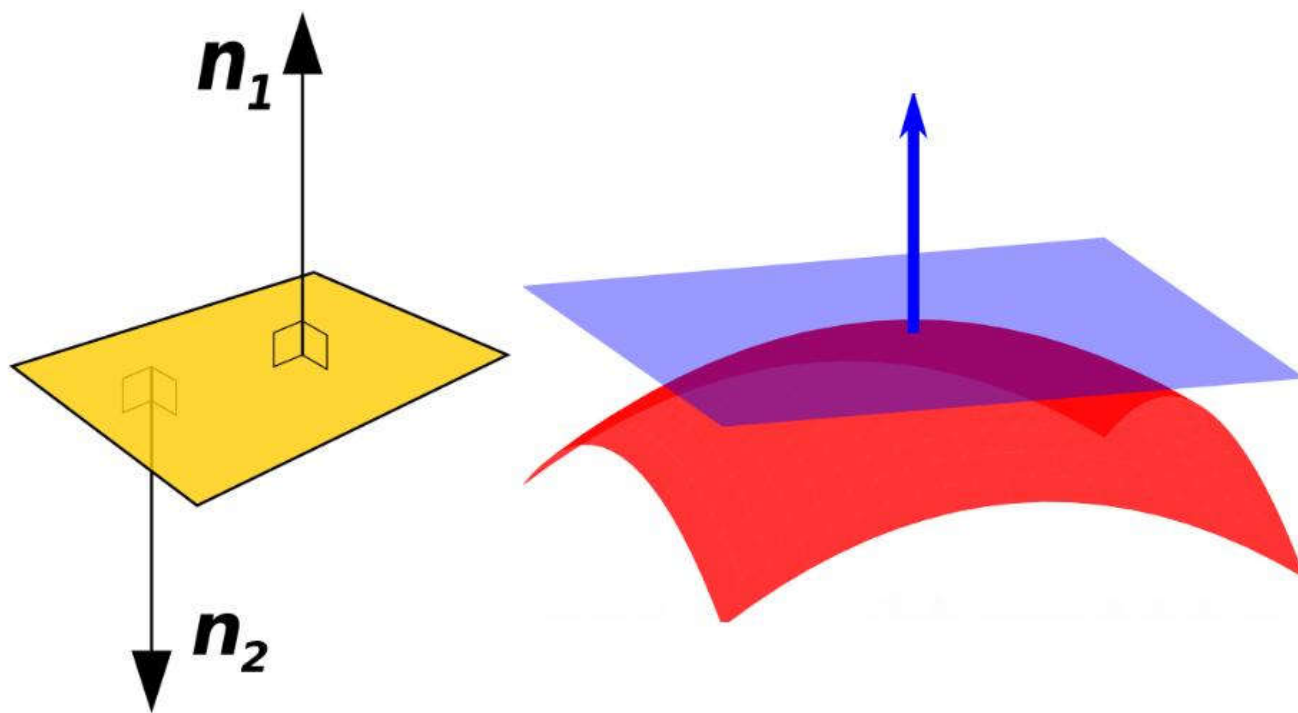
## 如何估计点云的表面法线？

小白：师兄，现在可以网格化了吗？

师兄：还不行。。。别急，网格化前我们还需要估计一下点云的表面法线（normal）

小白：啊，怎么又冒出来一个法线。。。

师兄：法线好像是中学就学过了，应该还记得平面的法线的定义吧，平面的法线是垂直于该平面的向量，如下图所示



(<https://i.loli.net/2018/12/28/5c2580ab0b402.png>)

你看上面右边那个图，对于曲面来说，曲面在某点P处的法线为垂直于该点切平面（tangent plane）的向量

小白：记得呢，不过这个法线有什么用？怎么就突然冒出来了

师兄：法线很有用的，尤其是在三维建模中应用非常广泛，比如在计算机图形学（computer graphics）领域里，法线决定着曲面与光源（light source）的强弱处理（Flat Shading），对于每个点光源位置，其亮度取决于曲面法线的方向。

小白：原来如此。不过好像平面或曲面的法线比较容易计算，方程  $ax + by + cz = d$  表示的平面，向量  $(a, b, c)$

就是其法线。而我们这里是点云呢！怎么算呢？

师兄：确实如此。点云的法线计算是稍微麻烦点，一般有两种方法：

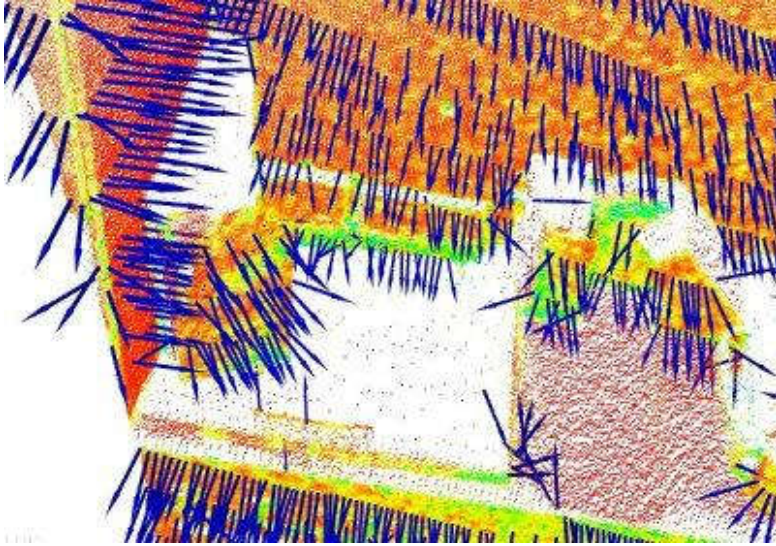
- 1、使用曲面重建方法，从点云数据中得到采样点对应的曲面，然后再用曲面模型计算其表面的法线
- 2、直接使用近似值直接从点云数据集推断出曲面法线

这里主要用第2种方法来近似估计点云中每个点的表面法线。

具体来说，就是把估计某个点的表面法线问题简化为：从该点最近邻计算的协方差矩阵的特征向量和特征值的分析，这里就不多做介绍了。PCL已经帮我们封装好了函数啦

我们计算出来点云的法线大概是这样的





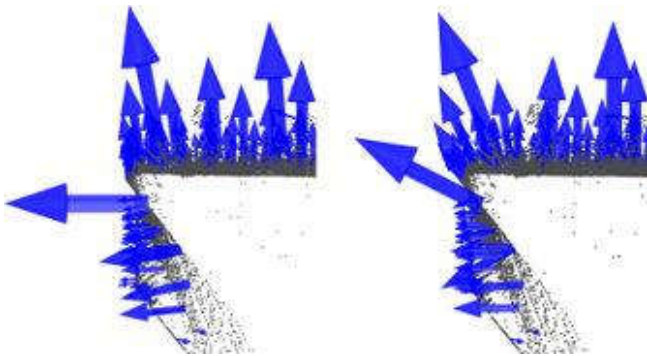
(<https://i.loli.net/2018/12/28/5c2580abc79d0.jpg>)

小白：那个箭头就代表法线吧？

师兄：对的，我们前面提到了，需要从该点的周围点邻域（也称为k邻域）估计一点处的表面法线，所以这个K邻域的选取也很关键

小白：这个K邻域选取会有什么影响吗？

师兄：有的，而且影响挺大的，K近邻的取值可以通过选择k个最近点，或者确定一个以r为半径的圆内的点集来确定，你看下面这个图是对同一个点云用不同尺度因子（k和r）进行法线估计的结果。左边部分表示比例因子选择的比较合适，估计的表面法线近似垂直于这两个平面，即使在互相垂直的边缘部分，也能明显看到边沿。而右边的尺度因子就选的有点大了，这样临近点集更大范围的覆盖临近表面的点，两个平面边沿处估计的法线就不准了，不能表达真实的情况。



(<https://i.loli.net/2018/12/28/5c2580a9dd4e8.jpg>)

小白：确实是这样啊，看来编程的时候要格外注意了。

师兄：法线估计的示例如下，我也给你注释好啦

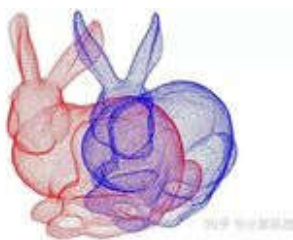
```

1 // 法线估计
2 pcl::NormalEstimation<PointT, pcl::Normal> normalEstimation;
3 normalEstimation.setInputCloud(cloud_smoothed);
4 pcl::search::KdTree<PointT>::Ptr tree(new pcl::search::KdTree<PointT>);
5 normalEstimation.setSearchMethod(tree);
6 pcl::PointCloud<pcl::Normal>::Ptr normals (new pcl::PointCloud<pcl::Normal>); //
7 // K近邻确定方法, 使用k个最近点, 或者确定一个以r为半径的圆内的点集来确定都可以, 两者选1即i
8 normalEstimation.setKSearch(10); // 使用当前
9 //normalEstimation.setRadiusSearch(0.03); //对于每一个点都用半
10 normalEstimation.compute(*normals); //计算法线

```

本文参考: PCL官网

(<https://creativecommons.org/licenses/by-sa/4.0/>)

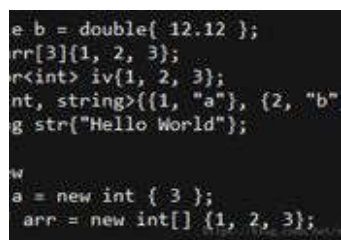


(/article/77831572993/)

## 从零开始一起学slam--icp (/article/77831572993/)

<https://zhuanlan.zhihu.com/p/66168444> 小白: 师兄, 最近忙什么呢, 都见不到你人影, 我们的课也好久没更新了呢  
师兄: 抱歉, 抱歉, 最近忙于俗事。我后面一起补上, 学习

劲头得向你们年轻人学习啊! 话说, 你最近在研究什么呢? 小白: 最近在看ICP, 十四讲上简单提了下, 还有点懵懵哒 师兄: 好, 那今天就说说ICP算法吧。搬个小板凳做好哈: ICP全称Iterative Clos...



(/article/35458801/)

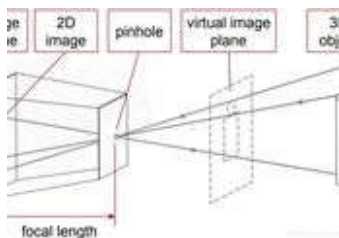
## 从零开始一起学习SLAM | C++新特性要不要学? (/article/35458801/)

学习SLAM, C++编程是必备技能。不过, 大家在学校里学习的书本一般比较老, 主要还是C++98那些老一套。本文所谓的C++新特性是指C++11及其以后的C++14、C++17增加的新关键字

和新语法特性。其中C++11是C++98以来最重要的一次变化, 而其后的C++14、C++17是在该基础上的完善和补充。那么, 问题来了, 如果我不想学习新特性, 还是像以前那样编程可以吗? 答案是: 可以, 不过这就像...

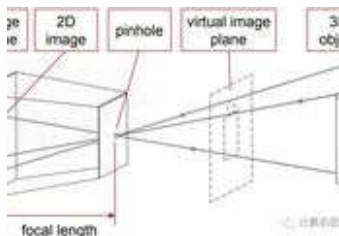
## 从零开始一起学习SLAM | 相机成像模型 (/article/626456023/)

上一篇文章《从零开始一起学习SLAM | 为啥需要李群与李代数?》以小白和师兄的对话展开, 受到了很多读者的好评。本文继续采用对话的方式来学习一下相机成像模型, 这个是SLAM中极其重要的内容, 必须得掌握哦~ 小白: 师兄, 上次听你讲了李群李代数, 有种“听君一席话胜读十年书”的赶脚~后来看书感觉容易理解多了呢! 师兄: 是吗?



(/article/626456023/)

那太好了，给你讲的过程也加深了我的理解呢 小白：那师兄今...



(/article/314770227/)

## 从零开始一起学习SLAM (8) 相机成像模型 (/article/314770227/)

文章目录 小孔成像 纷繁复杂的坐标系 针孔相机成像原理 相机畸变 此文发于公众号：计算机视觉life。 原文链接：从

零开始一起学习SLAM | 相机成像模型 上一篇文章《从零开始

一起学习SLAM (7) 为啥需要李群与李代数？》以小白和师兄的对话展开，受到了很多读者的好评。本文继续采用对话的方式来学习一下相机成像模型，这个是SLAM中极其重要的内容，...



(/article/427173703/)

## 从零开始一起学习SLAM (5) 为什么要用齐次坐标？ (/article/427173703/)

文章目录 为什么使用齐次坐标 什么是齐次坐标？ 使用齐次坐

标有什么优势？ 1 能否非常方便的表达点在直线或平面上 2 方便表达直线与直线，平面与平面的交点 3 能够区分一个向量和一个点 4 能够表达无穷远 5 更简洁的表达欧氏空间变换 作业 此文发于公众号：计算机视觉life。 原文链接：从零开始一起学习SLAM | 为什么要用齐次坐标？ 为什么使用...

## 猜你喜欢



(/article/2423378004/)

## 从零开始一起学习SLAM | 为啥需要李群与李代数？ (/article/2423378004/)

很多刚刚接触SLAM的小伙伴在看到李群和李代数这部分

的时候，都有点蒙蒙哒，感觉突然到了另外一个世界，很多都不自觉的跳过了，但是这里必须强调一点，这部分在后续SLAM的学习中其实是非常重要的基础，不信你看看大神们的论文就知道啦。关于李群李代数，其实高翔的《视觉SLAM十四讲》里推导什么的挺清楚了，本文就在高博的基础上用比较容易理解的语言讲述一下重点。首先，假装（也可能是真的）自己是个小白，我们假...



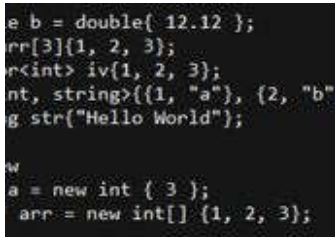
(/article/70721392617/)

## 从零开始一起学习SLAM (5) 为什么要用齐次坐标? (/article/70721392617/)

文章目录 为什么使用齐次坐标 什么是齐次坐标? 使用齐次

坐标有什么优势? 1 能否非常方便的表达点在直线或平面上 2 方便表达直线与直线, 平面与平面的交点 3 能够区分一个向量和一个点 4 能够表达无穷远 5 更简洁的表达欧氏空间变换 作业 此文发于公众号: 计算机视觉life。

原文链接: 从零开始一起学习SLAM | 为什么要用齐次坐标? ...



(/article/560870194/)

## 从零开始一起学习SLAM (4) C++新特性要不要学? (/article/560870194/)

文章目录 从零开始一起学习SLAM | C++新特性要不要学? 为

何要学习新特性? 常用的C++新特性 1 更方便的列表初始化 2 省心省力的自动类型推导 3 简洁舒服的循环体 4 简洁的lambda表达式 5 随心所欲可变长的参数模板 6 其他新特性 练习题 此文发于公众号: 计算机视觉life。 原文

链接: 从零开始一起学习SLAM | C++新特性...



(/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...