

实验 2 视觉 3D 点云图构建与深度测量

实验目的:

通过对 RGB-D 相机参数读取、画面帧的拼接验证,了解机器视觉中深度图像处理的一般工作流程,掌握 OpenCV、Eigen、PCL 等第三方图像图形库的使用方法。尝试自行搭建双目视觉系统,实现对视图中各物体的深度测量。

实验内容:

1. 矩阵运算库 Eigen 的使用
2. 彩色图、深度图的读取与参数提取
3. 相机参数加载与使用
4. 点云地图拼接
5. 双目视觉系统标定与深度测量(限实验方案设计)

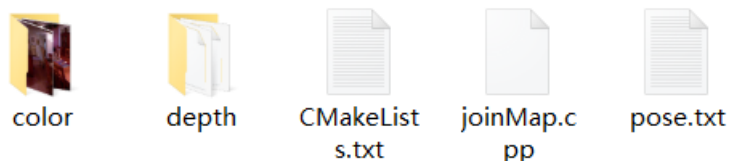
实验设备:

自带笔记本 PC 机摄像头/USB 摄像头, 安装 Ubuntu14.04 或以上系统, OpenCV 开发库

预备知识:

1. 深度相机原理
2. 深度图像采集过程及坐标转换关系
3. linux, C++编程基础, Cmake 编译器使用
4. Eigen 矩阵运算库, OpenCV 图像处理库

实验 1-4 步所需文件:



Color 和 depth 文件夹下分别放置了 5 张彩色和深度图,取自同一个 RGB-D 摄像机,按序号配对;

CMakeLists.txt 为编译配置文件, joinMap.cpp 为点云拼接主程序, pose.txt 为相机参数。

实验步骤:

1. 环境准备

安装 opencv3, 版本 3.4.0

(参考) 依赖:

```
sudo apt-get install build-essential cmake libgtk2.0-dev libvtk5-dev libjpeg-dev  
libtiff4-dev libjasper-dev libtbb-dev libopenexr-dev
```

实验例程使用 opencv3, 考虑到实验 1 已经安装过 opencv2, 可以参考以下链接,:

opencv2 和 opencv3 共存安装 <http://blog.csdn.net/thornsuge/article/details/71171272>

2. 熟悉线性代数运算库 Eigen

Eigen (<http://eigen.tuxfamily.org>) 是常用的 C++ 矩阵运算库, 具有很高的运算效率。大部分需要在 C++ 中使用矩阵运算的库, 都会选用 Eigen 作为基本代数库, 例如 Google Tensorflow, Google Ceres, GTSAM 等, 通过以下命令安装, 并查找例程, 定义矩

阵，实现矩阵加法，乘法，转置运算；

```
sudo apt-get install libeigen3-dev
```

3. 安装点云工具 pcl

```
sudo add-apt-repository ppa:v-launchpad-jochen-sprickerhof-de/pcl
```

```
sudo apt-get update
```

```
sudo apt-get install libpcl-all
```

4. 实现点云图像拼接

阅读例程代码并编译运行，理解代码原理，

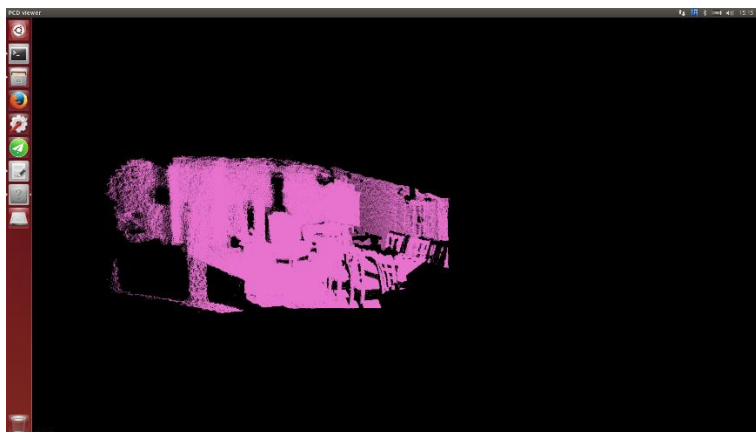
```
cmake .
```

```
make
```

执行 joinMap 生成 map.pcd

```
./joinMap
```

正确的实验结果应如图所示，可旋转点云图，查看不同视角下的拼接结果。



5. 双目视觉系统标定与深度测量

在前 4 步的基础上，参照网络资源链接如：

驱动双摄像头

<http://blog.csdn.net/hx1298234467/article/details/50478176>

读取并显示两个摄像头

http://blog.csdn.net/Jasmine_shine/article/details/51646532

双目测距 demo,附件代码文件及参考结果见 binocularDistance-master.zip

<https://www.cnblogs.com/zhanxiage1994/p/6673892.html>

- (1) 参照上述资源，给出用两个 USB 摄像头实现**简易测距**实验的原理、步骤，所需的关键代码；
- (2) 手动搭建简易双目相机，实现**深度测量(选做)**；

实验报告

1. 用自己的话给出本次实验的代码框架，代码分析和实验结果；
2. 单目相机内参、外参是指哪些参数？外参如何用矩阵表示？
3. 双目相机内参、外参是哪些参数？什么是双目相机的本征矩阵？

报告以 Word 或 PDF 形式提交，以“**Robotic_Lab2_学号_姓名**”格式命名，每人一份，提交至教辅系统，截至时间见教辅系统通知。