



智能无人机技术设计实践

--实验1指导书

于超

联系方式: yc19@mails.tsinghua.edu.cn

时间: 2019.9.28





目 录

- 1. 三维空间刚体运动的描述方式
 - 实验1.1: 四元数 q 与旋转矩阵 R
 - 实验1.2: 旋转矩阵 R 与变换矩阵 T
- 2. 坐标转换
 - 实验: 点云拼接



1 三维空间刚体运动的描述方式

◆ 实验1.1：四元数 q 与旋转矩阵 R

● 问题背景描述

四元数和旋转矩阵是常见的运动描述方式，我们直接给出二者的转换关系：
设四元数 $q = q_0 + q_1i + q_2j + q_3k$ ，对应的旋转矩阵 R 为

$$\begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 1 - 2q_1^2 - 2q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix}$$

反之，由旋转矩阵到四元数的转换如下。假设矩阵为 $R = \{m_{ij}\}, i, j \in [1, 2, 3]$ ，其对应的四元数 q 由下式给出：

$$q_0 = \frac{\sqrt{\text{tr}(R) + 1}}{2}, q_1 = \frac{m_{23} - m_{32}}{4q_0}, q_2 = \frac{m_{31} - m_{13}}{4q_0}, q_3 = \frac{m_{12} - m_{21}}{4q_0}.$$



1 三维空间刚体运动的描述方式

◆ 实验1.1：四元数 q 与旋转矩阵 R

● 任务

根据之前提供的数学转换关系，用Python语言实现两个函数，描述如下：

```
def q2R(q):
```

```
'''
```

功能：四元数转旋转矩阵

输入： q 1×4 的四元数向量，实部在前，虚部在后

返回值： R 3×3 的旋转矩阵

其中输入和返回值都用numpy结构

```
'''
```

```
def R2q(R):
```

```
'''
```

功能：旋转矩阵转四元数

输入： R 3×3 的旋转矩阵

返回值： q 1×4 的四元数向量，实部在前，虚部在后

其中输入和返回值都用numpy结构

```
'''
```



1 三维空间刚体运动的描述方式

◆ 实验1.2: 旋转矩阵 R 与变换矩阵 T

● 问题背景描述

旋转矩阵和变换矩阵的关系:

$$T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$$

● 任务

完成两个函数，实现旋转矩阵和变换矩阵间的相互转换，描述如下：

```
def R2T(R, t):  
    ...  
    功能：旋转矩阵变换矩阵  
    输入：R 3x3的旋转矩阵 t 3维的平移向量  
    返回值：T 4x4的变换矩阵  
    其中输入和返回值都用numpy结构  
    ...  
  
def T2R(T):  
    ...  
    功能：变换矩阵转旋转矩阵  
    输入：T 4x4的变换矩阵  
    返回值：R 3x3的旋转矩阵  
    其中输入和返回值都用numpy结构  
    ...
```




2 坐标转换

◆ 实验2：点云拼接（点云由所有像素在世界坐标系下的投影组成）

● 问题背景描述

熟悉相机内外参的使用方法以及像素坐标到世界坐标的转换。从实验1资料中下载提供的5张RGB-D图像，poses.txt文件中给定了5张图像对应的内参和外参。

● 任务

- (1) 计算任何一个像素在相机坐标系下的位置以及在世界坐标系下的位置。
- (2) 将图像拼接成点云。

● 任务分解提示

1. 利用OpenCV读取RGB图和深度图；
2. 使用之前所完成的函数，将所给的外参转换成变换矩阵；
3. 利用所给的相机内参和图像深度信息，将像素坐标转换成相机坐标；
4. 利用步骤2计算出的变换矩阵，将相机坐标转换成世界坐标；
5. 拼接点云，向ROS topic /points发布；
6. RVIZ 可视化点云。

```
from sensor_msgs.msg import PointCloud2, PointField
from sensor_msgs import point_cloud2
```



2 坐标转换

● 环境配置

完成本实验需要的库有

- Numpy
- OpenCV

`pip install opencv-python==3.1.0.0` (会同时安装numpy 1.11.1)

● 补充说明

代码中超参说明:

$$\text{内参矩阵 } K = \begin{pmatrix} 518.0 & 0 & 325.5 \\ 0 & 519.0 & 253.5 \\ 0 & 0 & 1 \end{pmatrix}$$

深度因子 $k=1000$

外参矩阵见poses.txt文件, 采用四元数表示 $[x, y, z, qx, qy, qz, qw]$, 其中, q_w 是四元数的实部, 下图是第一张图片的外参示意图。

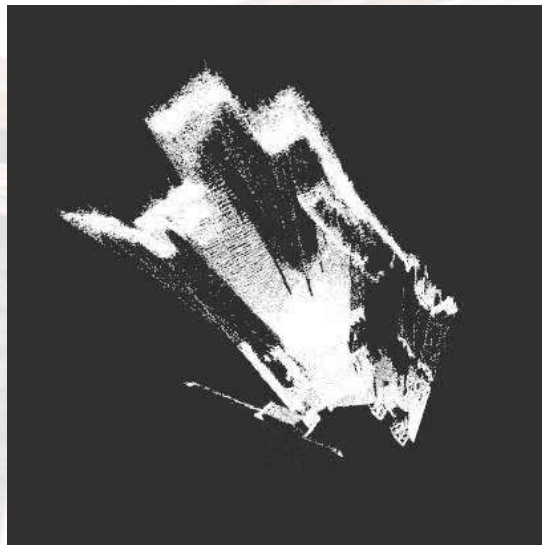
$[-0.228993, 0.00645704, 0.0287837, -0.0004327, -0.113131, -0.0326832, 0.993042]$.



作业提交

- 在网络学堂上直接提交一个python文件，助教会逐一运行。
- 该Python文件应该包括4个子函数（实验1.1和实验1.2），一个主函数（实验2）。
- 任选一张图片，任选一个像素，打印出其在**相机坐标系**和**世界坐标系**下的位置。
- 向ROS topic **/points**发布生成好的点云，采用PointCloud2的消息格式，用RVIZ可视化。（可以等ROS通信学习之后再完成这一步）

拼接后的
点云



对比原
图点云





谢谢!

答疑地点：双清大厦2号楼502