

车载软件设计基础

松灵小车任务2



**2024至2025学年第 s2学期**

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 团队作用 |
| E2021201 | 宋智鑫 | Autoware标定和融合程序编写 |
| E2021146 | 王定彬 | Autoware标定和融合程序编写 |
| E2021206 | 熊健霖 | Ros程序标定和标定差异分析 |
| E2021198 | 陈干 | Ros程序标定和其余标定数据采样 |
| 任课教师 | 刘骥 | |
| 成 绩 |  | |

|  |  |
| --- | --- |
| 任务书 | |
| 任务内容 | 实验分组进行，每组人数不超过5人。在松灵小车上编写ROS程序，完成如下任务：  （1）通过/camera/color/camera\_info读取松灵小车的color相机的标定数据；  （2）用ROS官方标定工具对松灵小车的color相机进行标定，记录标定结果；  （3）用Autoware标定工具对松灵小车的color相机进行标定，记录标定结果；  （4）比较（1）（2）（3）所得结果的差异，分析原因；  （5）用Autoware标定工具实现松灵小车color相机到激光雷达的外参标定，记录标定结果；  （6）控制松灵小车运动（用遥控器），编写程序，程序显示运动过程中， color相机拍摄的图像（image）、激光雷达扫描的点云（point）和融合后的point-image图像。 |
| 程序规范 | （1）所有程序代码采用C++编写，使用git进行源代码管理；  （2）类名、变量名、函数名应符合C++的命名规范，并在代码中前后保持一致；  （3）涉及面向对象的程序，例如自定义的类，应符合面向对象的设计原则；  （4）正确使用头文件和源文件，自定义的头文件应符合头文件的编写原则，例如用条件宏定义确保头文件不被多次引用、不在头文件中进行类和函数的实现（模板除外）；  （5）项目必须是ROS项目，符合ROS的项目的规范，正确编写CmakeLists.txt等文件；  （6）程序能够在松灵小车上运行。 |
| 报告要求 | （1）报告至少应该包括人员分工、标定过程、标定结果分析、程序效果展示、总结分析5个部分；  （2）人员分工介绍组员各自的工作情况；  （3）标定过程描述color相机标定过程、color相机到激光雷达的外参标定过程；  （4）标定结果分析描述标定结果与实际结果的差异，分析产生差异的原因；  （5）程序效果展示描述point-image图像的生成程序，除了程序运行效果截图之外，应该有必要的文字说明；  （6）总结分析可以分析实现的效果与理想情况的差异，分析导致这些差异的原因，切忌不要写成心得体会；  （7）报告应该格式规范、排版整洁、少语病和错误。 |
| 作业提交 | （1）含有git仓库（有.git目录）的完整源代码；  （2）程序功能演示的讲解视频；  （3）任务报告。 |
| 评分标准 | 按照五级制打分，分为优秀、良好、中等、及格、不及格，各评分项占总成绩的比例为：  （1）任务完成情况占评分的60%；  （2）程序规范占评分的20%；  （3）报告占评分的20%。  评分老师根据各部分的完成情况，直接给出总成绩。 |

1. **人员分工**

* 宋智鑫
* 王定彬

**负责任务：**

1. **读取松灵小车的color相机的标定数据**：
   * 宋智鑫负责通过ROS的/camera/color/camera\_info话题读取松灵小车的color相机的标定数据。
2. **用ROS官方标定工具对color相机进行标定**：
   * 王定彬负责使用ROS提供的标定工具（如camera\_calibration包）进行标定，并记录标定结果。
3. **用Autoware标定工具对color相机进行标定**：
   * 宋智鑫和王定彬共同负责使用Autoware提供的标定工具进行标定，并记录标定结果。
4. **比较三种标定结果的差异，分析原因**：
   * 宋智鑫和王定彬共同负责比较步骤1、2、3的结果，并分析造成差异的原因。

* 熊健霖
* 陈干

**负责任务：**

1. **用Autoware标定工具实现color相机到激光雷达的外参标定**：
   * 熊健霖负责使用Autoware提供的工具进行color相机到激光雷达的外参标定，并记录标定结果。
2. **控制松灵小车运动，并显示图像、点云和融合后的图像**：
   * 陈干负责控制松灵小车使用遥控器移动，并编写程序在小车移动过程中显示color相机拍摄的图像、激光雷达扫描的点云以及融合后的point-image图像。
3. **标定过程**

* 1.可以在启动深度相机后直接通过rostopic echo /camera/color/camera\_info 命令查看松灵小车自身的标定结果。

文本

描述已自动生成

* 2.利用ROS官方标定工具对松灵小车的color相机进行标定。

启动深度相机：roslaunch realsense2\_camera rs\_camera.launch

通过rosrun camera\_calibration cameracalibrator.py --size 8x6 --square 0.108

image:=/camera/color/image\_raw --no-service-check 命令启动ROS官方标定工具。

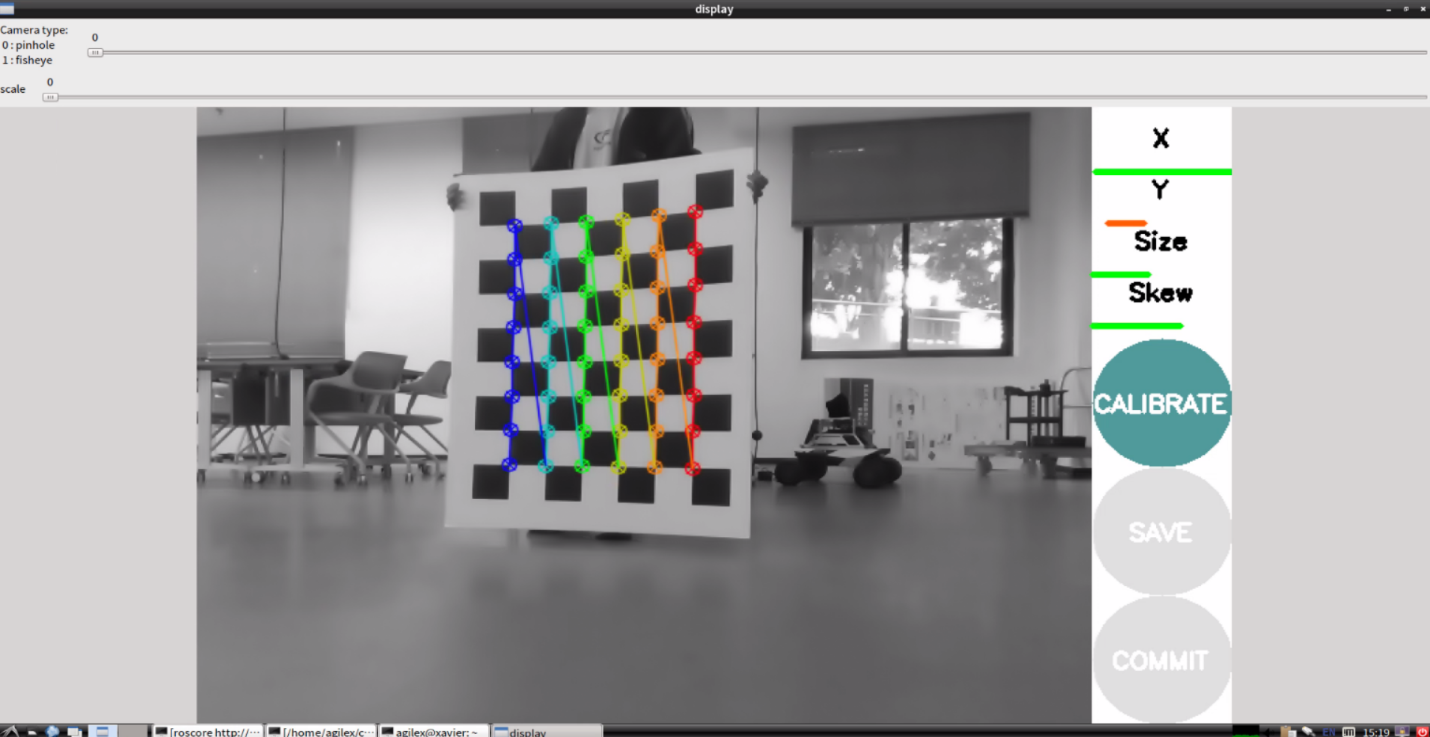
其中8x6代表棋盘的长和宽个数-1，即列数-1乘以行数-1，不能混淆了。

square 0.108 代表每个棋盘格子的长度。

image:=/camera/color/image\_raw 指定相机发布彩色图像的话题。

--no-service-check 不检测相机服务。

将标定板置于相机视野内，前后左右、平放、斜放等多个角度移动标定板，待CALIBRATE按钮亮起时，代表数据量足够，此时可以单击CALIBRATE查看结果。



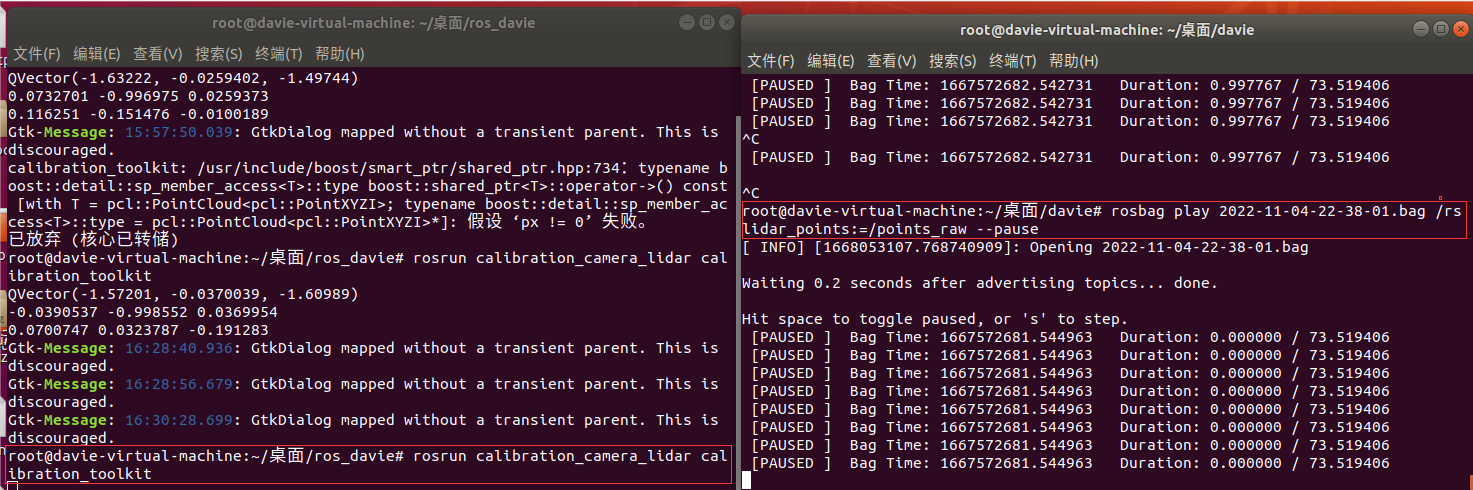
ROS官方标定工具标定结果如下：

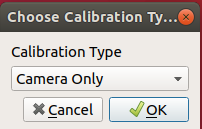
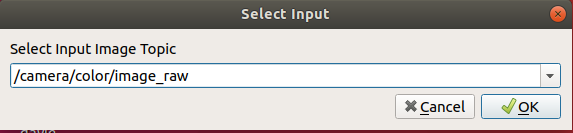
图形用户界面

描述已自动生成

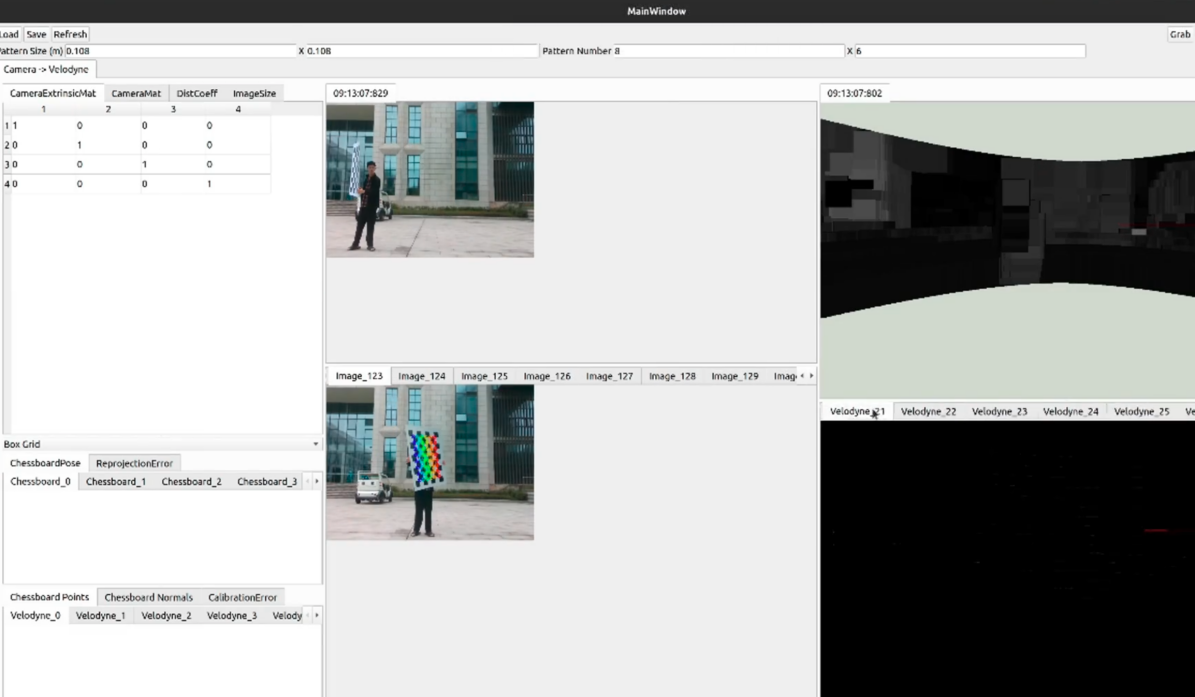
* 3.用Autoware标定工具对松灵小车的color相机进行标定。

首先启动autoware标定工具并播放之前录制的bag文件，注意需要指定激光雷达话题为autoware需要的话题名称。





当标定板出现在合适位置后，点击右上角Grab即可抓取

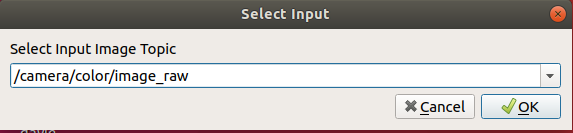


* 4. 用Autoware标定工具实现松灵小车color相机到激光雷达的外参标定。

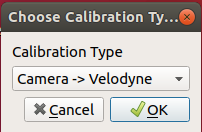
通过 rosrun calibration\_camera\_lidar calibration\_toolkit 命令启动autoware工具

通过 rosbag play xxxx.bag /rslidar\_points:=/points\_raw --pause 播放之前录制的bag文件

图像话题选择对应的相机topic



标定类型选择Camera->Velodyne，表示相机和激光雷达的联合标定。

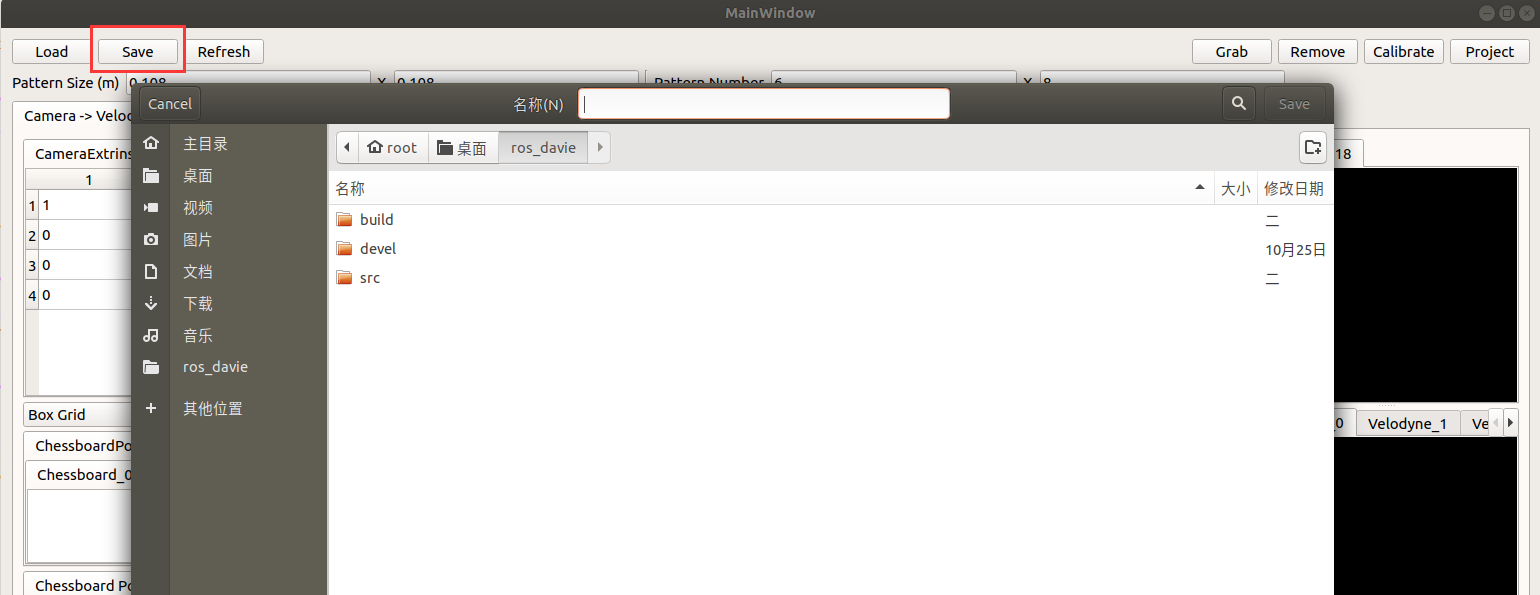


调整右上角黑窗口显示的激光雷达视角，确保标定板点云数据处于正前方。然后点击Grab按钮进行抓取，一般抓取20张左右，然后在右下角黑窗口进行标记，点击鼠标左键，使红色圆圈处于标定板正中间，鼠标右键取消。

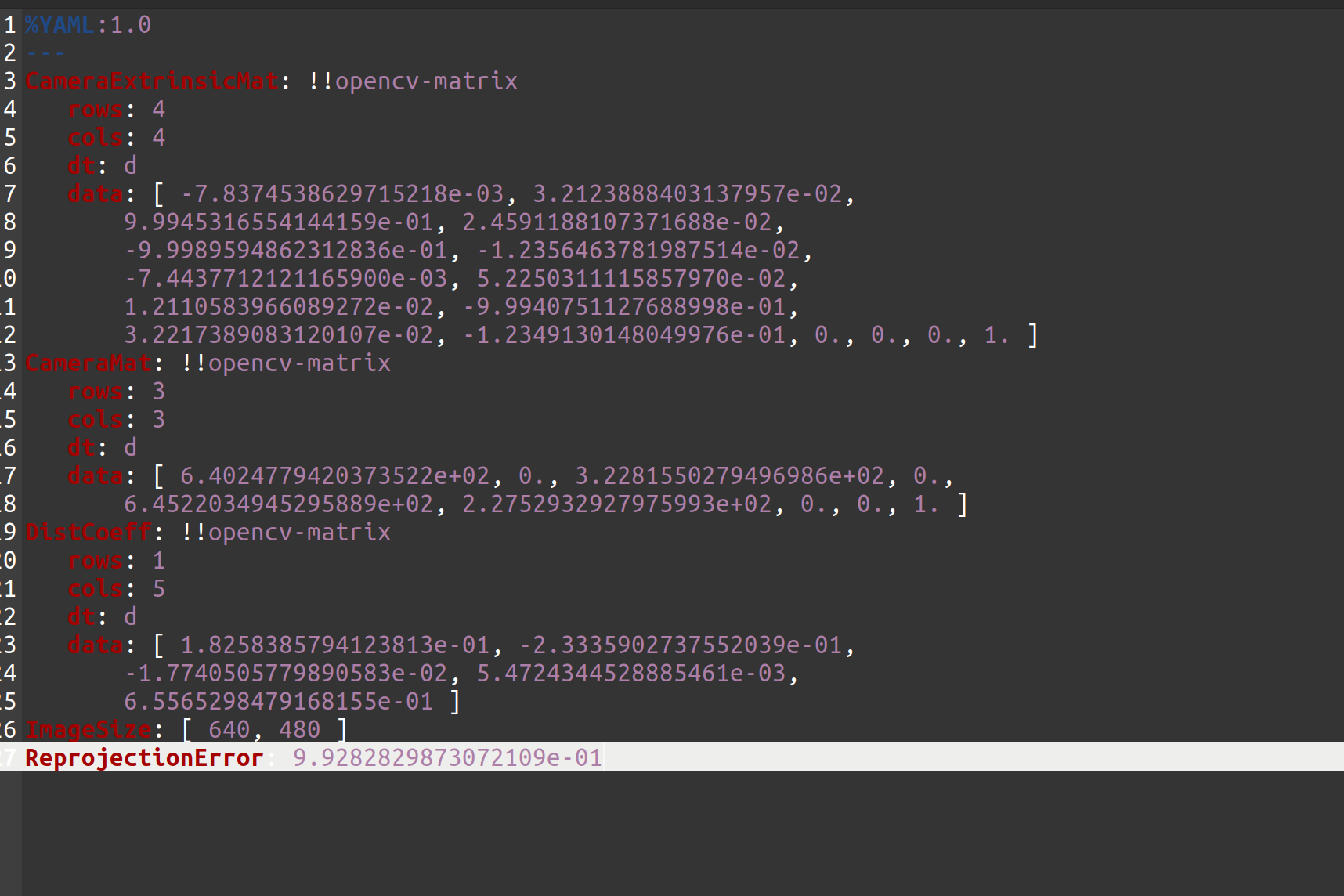
最后点击Calibrate、Project。查看标定效果，如果左下角窗口中红色线条处于标定板正中间说明标定效果较好，如果在边缘甚至超出标定板，则需要重新标定。



标定完成后点击左上角Save，输入文件名、选择路径进行保存即可。



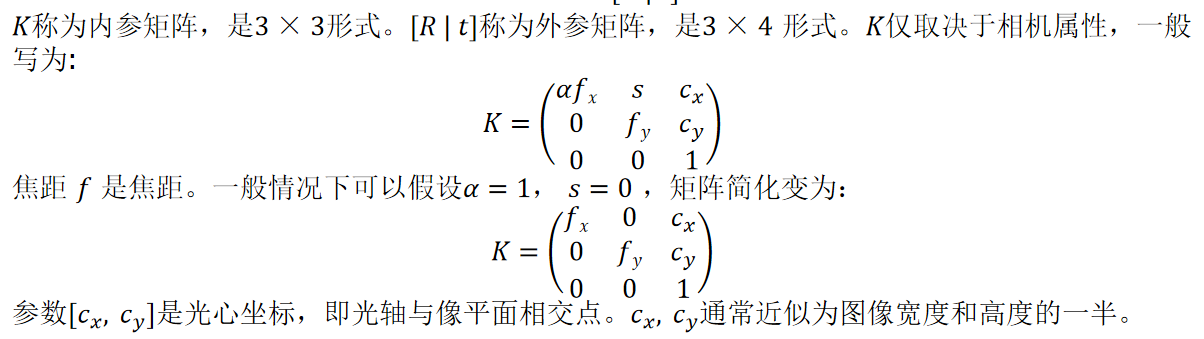
最后生成的yaml文件如下图：



1. **标定结果分析**

比较（1）（2）（3）所得结果的差异，分析原因

如下图所示K为相机内参——为相机本身属性，为主要测量对象。相机外参为相机的姿态。会随着相机姿态的改变而参数改变。所以我们主要分析的是简化情况下的相机内参的误差形成原因。



由上图可知由Ros官方标定工具和Autoware标定工具产生的相机内参矩阵中，fx，fy，Cx，Cy均偏大。Cx、Cy偏大原因未知，下面分析fx、fy偏大原因。



在内参简化矩阵中，由上式可知。影响fx，fy增大的原因可能为。dx、dy、dz偏大或dX、dY、偏小。因为我们使用的是标准标定板。dX、dY产生误差极小，所以基本可以排除。所以产生误差的原因为dx、dy、dz的偏大导致的。

1. dx、dy偏大

dx、dy为相机中以像素为单位的图像长和宽，导致数值偏大的原因可能为像素图像产生畸变导致图像的像素长与宽被拉大。导致fx、fy比实际值更大，导致相机内参偏大。

2.dz偏大

dz为相机到物体的实际距离。产生数值偏大的原因可能为，相机的深度信息不够准确。测量相机到物体的距离比实际距离稍大。导致fx、fy数值偏大，导致相机内参偏大。

由上可得，导致相机内参结果偏大的原因可能为。1.图像的畸变。2.图像的深度信息的测量误差。

**四、程序效果展示**





1. **总结和分析**

此次实验基本达到了预期结果，但在实验过程中也遇到了一些需要注意的细节问题。以下是具体的分析：

1. 标定准确性

在利用Autoware进行标定时，输入的标定板数据必须准确无误。如果输入的数据不准确，将会直接影响标定结果的正确性。例如，如果标定板的实际尺寸与输入的尺寸不符，那么生成的标定参数也将是错误的。因此，在进行标定前，务必仔细核对标定板的具体尺寸和位置信息。

2. YAML文件格式规范

Autoware生成的YAML文件具有严格的格式规范。每组数据的格式，包括每行的缩进、逗号后面的空格以及何时换行，都必须遵循固定的规则。一旦格式不符合规范，程序将无法正确读取文件，从而导致运行时出错。因此，在处理Autoware生成的YAML文件时，必须严格按照格式要求进行操作，确保文件的可读性和有效性。

3. 外参标定结果的处理

Autoware生成的标定结果在直接应用于程序之前需要进行一定的处理。特别是生成的4x4矩阵，其3x3部分代表了旋转矩阵，如果不对其进行适当的转置处理，将会导致激光雷达与相机的融合图像出现问题。例如，在未处理的情况下，融合图像中的点云线可能会呈现垂直状态，而不是期望的水平或正确的方向。正确的做法是在使用之前，对旋转矩阵进行适当的转置，以保证融合图像的一致性和准确性。

4. 标定方法的差异性

通过比较原始标定数据、ROS官方标定工具和Autoware标定工具的结果，我们发现不同方法得到的标定参数存在一定差异。具体表现为：

焦距（fx, fy）：ROS标定的结果焦距略有增加，而Autoware标定的结果焦距变化不大。

主点位置（cx, cy）：两种方法对主点位置的调整幅度不同。

畸变系数：三种方法得到的畸变系数有所不同，尤其是在径向畸变系数上。这些差异主要源于标定过程中使用的棋盘格图案数量、标定精度要求等因素的不同。为了获得更加一致的结果，可以考虑增加标定样本的数量，并优化标定过程中的参数设置。

5. 图像与点云融合的挑战

在控制松灵小车运动并显示图像、点云和融合图像的过程中，我们发现了一些挑战：

同步问题：确保图像和点云数据的时间同步非常重要，否则融合后的图像可能会出现时间差，导致数据不匹配。

数据处理效率：实时处理大量的图像和点云数据需要高效的算法和技术支持。在实验中，我们注意到处理速度有时会影响实时性的表现，因此优化数据处理流程是提高系统性能的关键。

总结

本次实验不仅验证了松灵小车color相机的标定数据的有效性，还展示了不同标定工具的效果差异。通过外参标定和图像点云融合技术的应用，进一步提高了系统对环境的感知能力。实验结果表明，选择合适的标定工具和参数优化策略对于实现准确的相机标定至关重要。未来的研究可以进一步探索更先进的标定技术和优化方法，以提升系统的鲁棒性和精度。

通过此次实验，我们认识到在实际应用中，不仅需要关注标定的准确性，还需要注意数据处理的细节和格式规范，以确保系统稳定可靠地运行。

1. **参考文献**

[1]Python实现激光雷达点云的向图像的重投影.http://t.csdn.cn/Y86jr,2020-01-20.

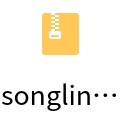
[2]autoware雷达相机联合标定得到的标定结果的使用.http://t.csdn.cn/cCzxx,2022-12-08.

[3]autoware标定工具进行固态激光雷达与相机的联合标定.http://t.csdn.cn/g8Wu5,2019-03-08.

[4]Scout mini 使用Ros例程控制.https://zhuanlan.zhihu.com/p/550920775,2021-03-18.

[5][美]Stephen Prata.C++ Primer plus[M].张海龙 袁国忠.北京:人民邮电出版社,2015:96-183.

**源码：**

****