

激光雷达+相机标定

(编译工具源码+标定过程)

使用autoware1.10.0版本的标定工具箱进行标定

前提：安装autoware，编译安装autoware1.10版本，需要ubuntu16.04版本

参考链接：

- 标定工具安装

[使用calibration_toolkit进行相机和三维激光雷达的联合标定(详细步骤)]

(https://blog.csdn.net/xt dx_x ty/article/details/114903688)

[激光雷达和相机的联合标定-CSDN博客](https://blog.csdn.net/weixin_43199832/article/details/118884784)

[autoware 激光-相机联合标定-CSDN博客](<https://blog.csdn.net/nuo112/article/details/116404760>)

[Autoware 工具 CalibrToolKit 标定 Robosense 雷达和 ZED 相机!]

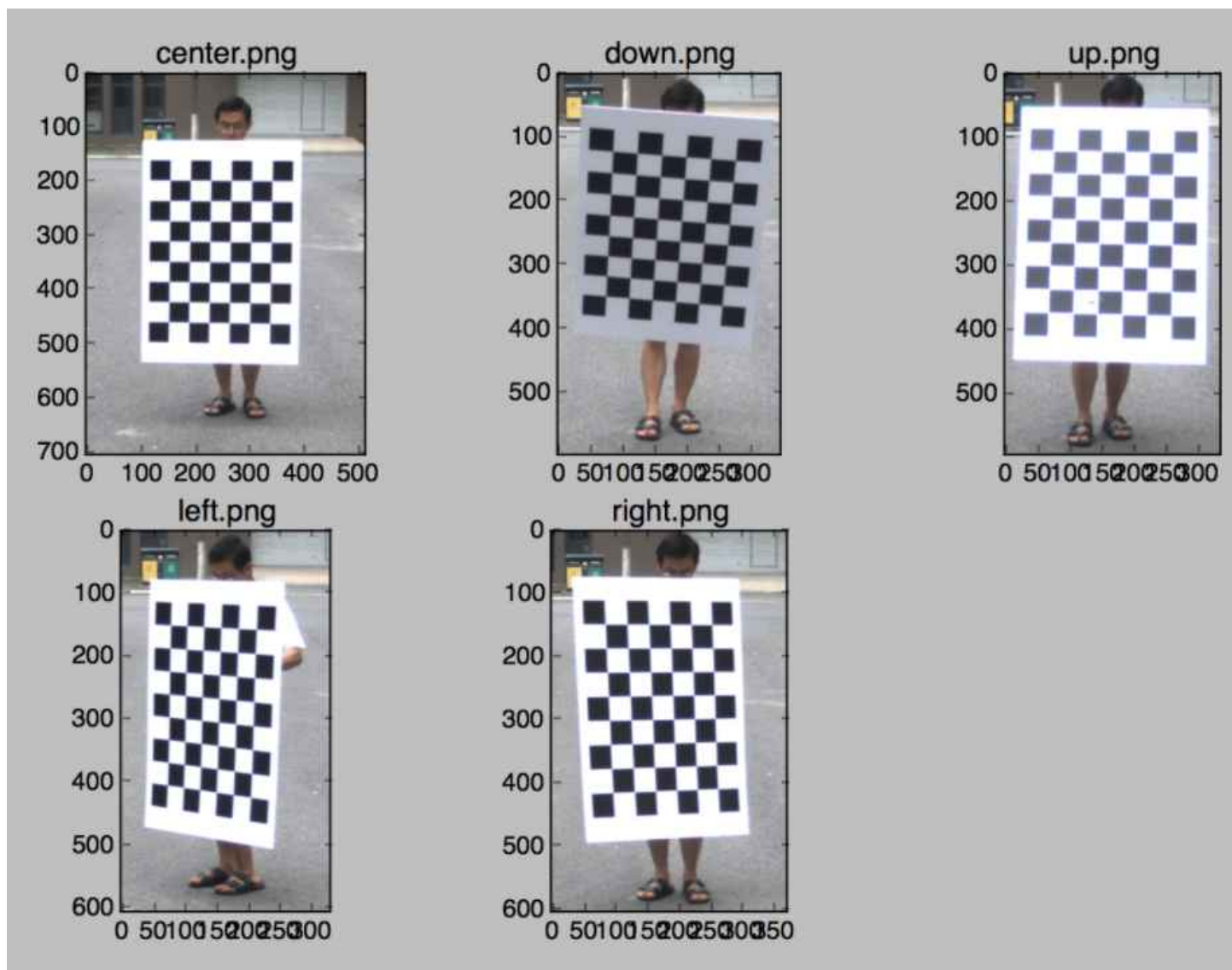
(<https://zhuanlan.zhihu.com/p/147638876/>)

[Calibration Toolkit - YouTube](https://www.youtube.com/watch?v=pfBmfgHf6zg&ab_channel=うすだひさし)

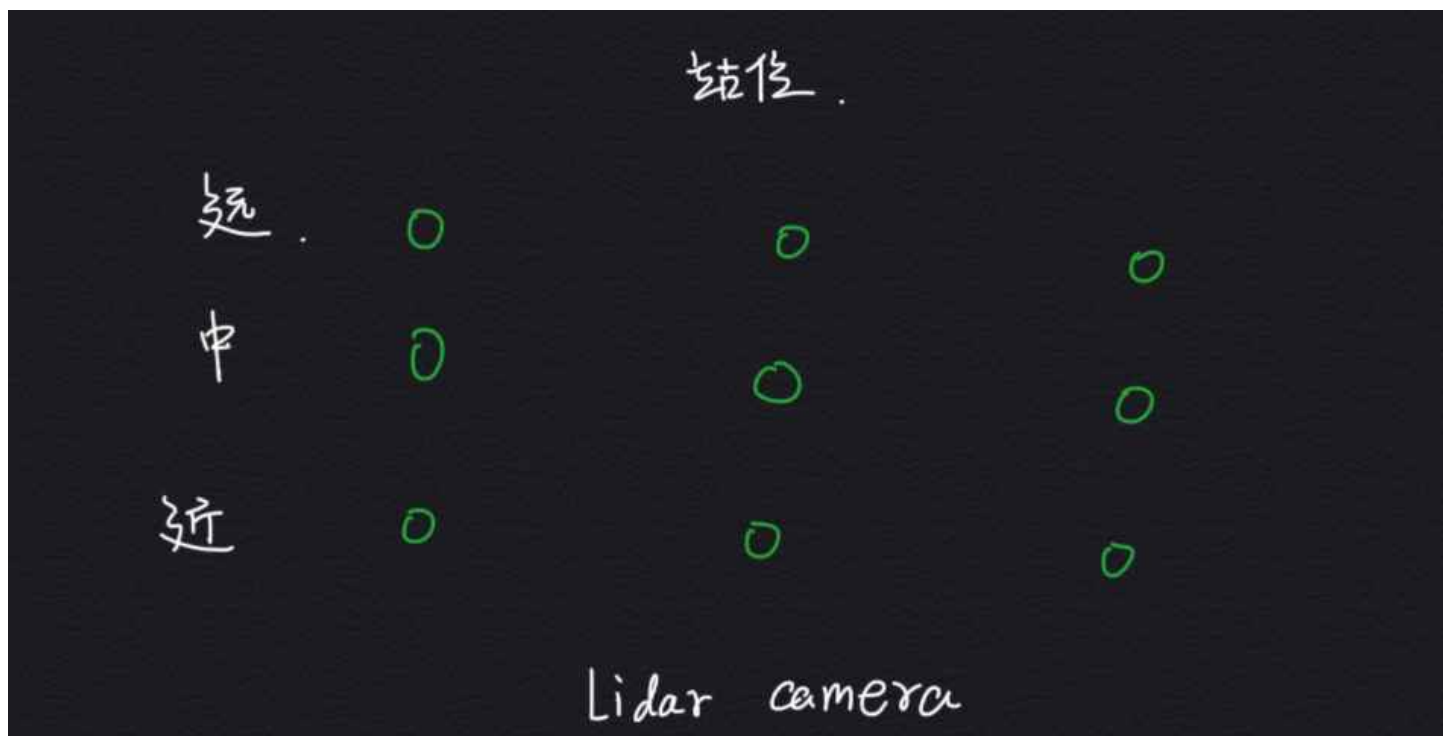
录制bag包时注意事项：

- 标定版要同时在相机和激光雷达的视野内

- 确定举标定板的站位和动作，标定版在摄像头下一般有9个站位，每个位置做相同的一组5个动作，5个动作分别为：正向，下俯；上仰，左偏，右偏，如图所示：



velody vlp16是16线雷达，所以拿标定板不能太远，不然就不能清楚地看到标定板了，录制时候站位如下：



每个位置移动标定板的姿态，每个姿势停留1-2s，防止模糊

- 录制bag包时，选择录制zed2相机的左目和右目话题，以及激光雷达的点云话题

```
1 rosbag record -o zed2_velodyne /zed2/zed_node/left/image_rect_color /zed2/zed_node/left/i
```

联合标定

录制好bag包之后，使用编译好的 autoware 1.10版本进行联合标定

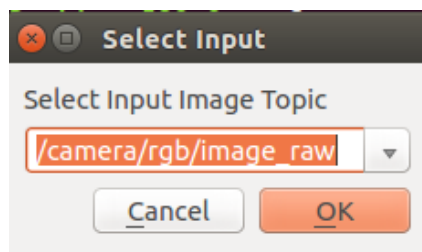
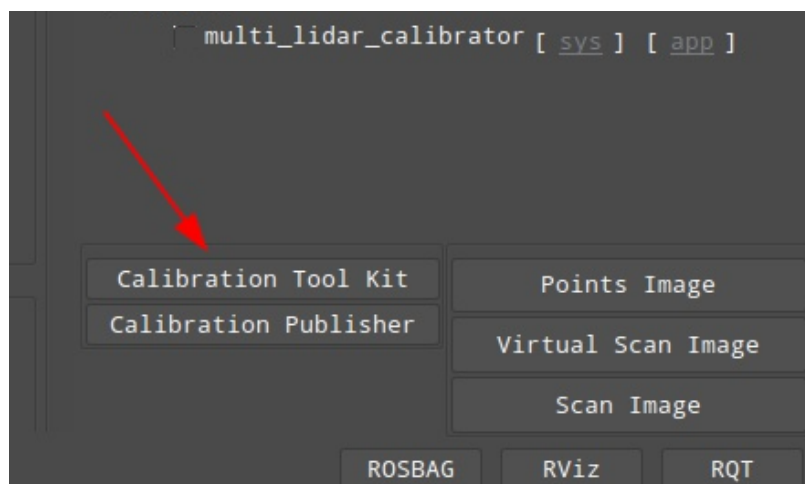
```
1 roscore
2 source ~/autoware_1.10/ros/devel/setup.bash
3 ./run
```

准备播放录制好的bag包

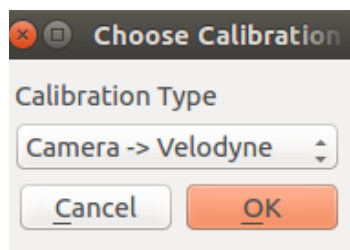
```
1 rosbag play zed2_velodyne.bag /velodyne_points:=/points_raw --loop --pause
```

可以按空格键暂停/播放，后面loop为循环播放的意思

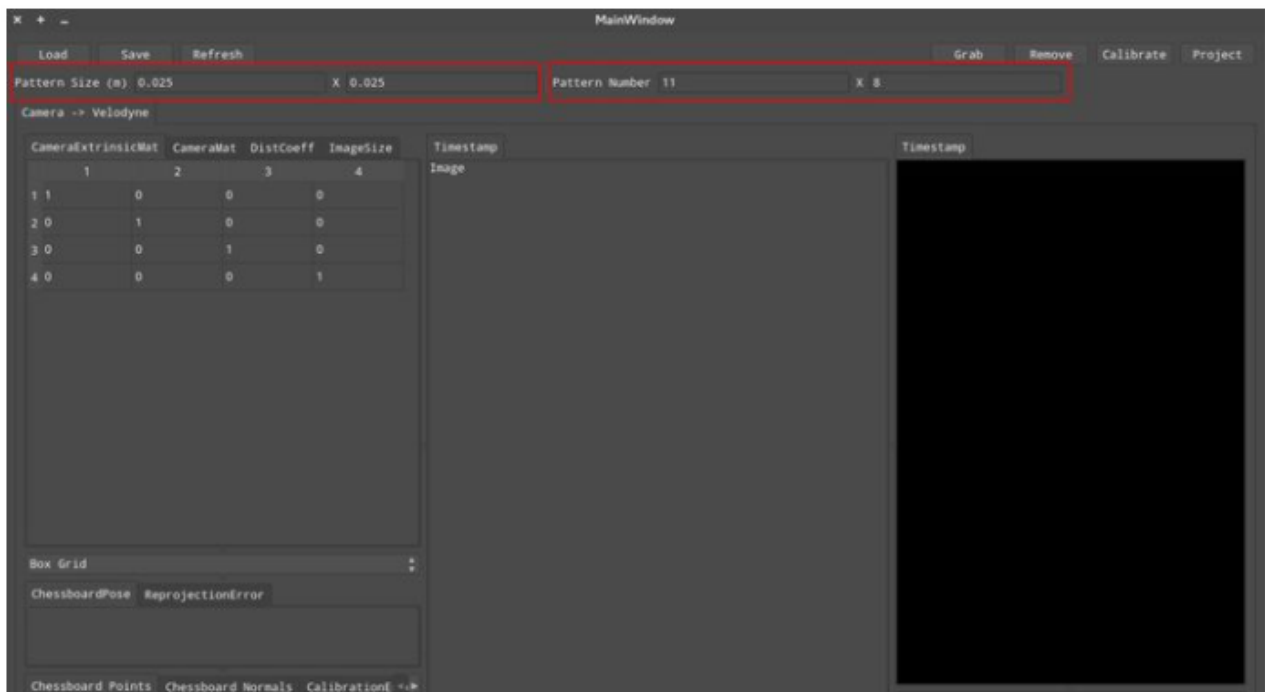
打开 Sensing下的calibration tool kit工具，选择相机话题，这里选择/zed2/zed_node/left/image_rect_color



选择/zed2/zed_node/left/image_rect_color话题



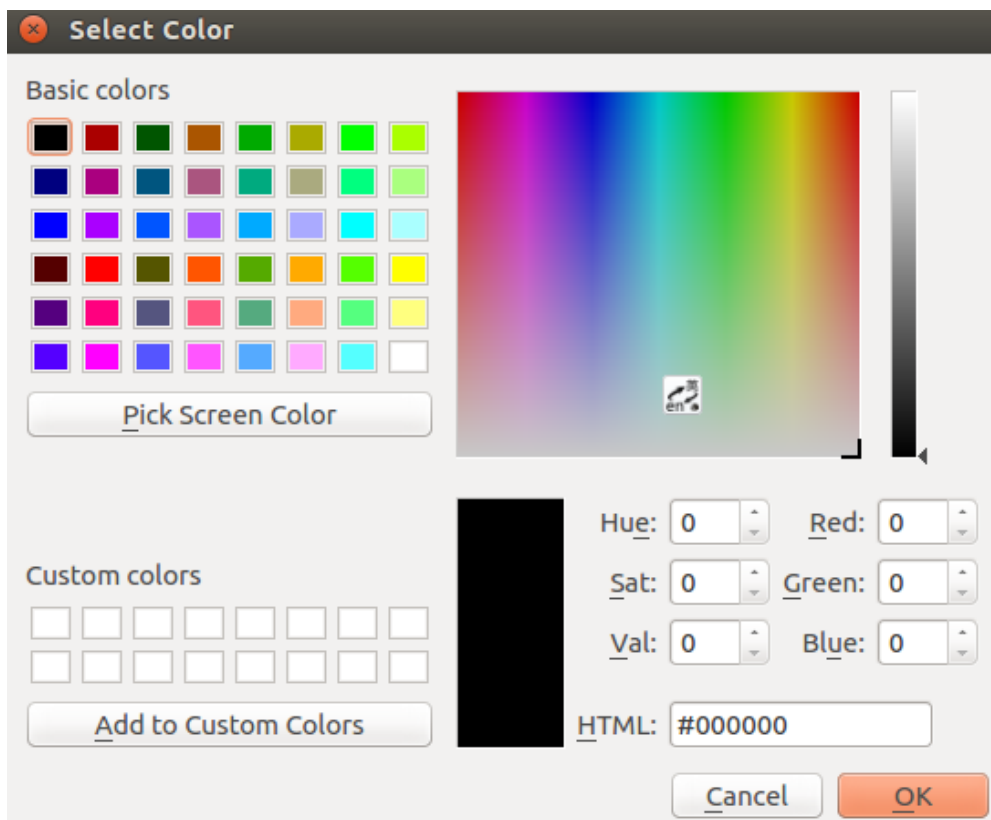
最后会弹出一个大界面，分为了几个区域，首先找到下图的一排位置，这排参数是标定板的尺寸。前面两个是标定板每个棋格的长宽，单位为m，此处我的标定板棋格大小为6cm，因此参数设置为0.06 * 0.06。后面是标定板的棋盘规格，我们使用的板子是12 * 9，具体情况根据自己的标定板对应修改。



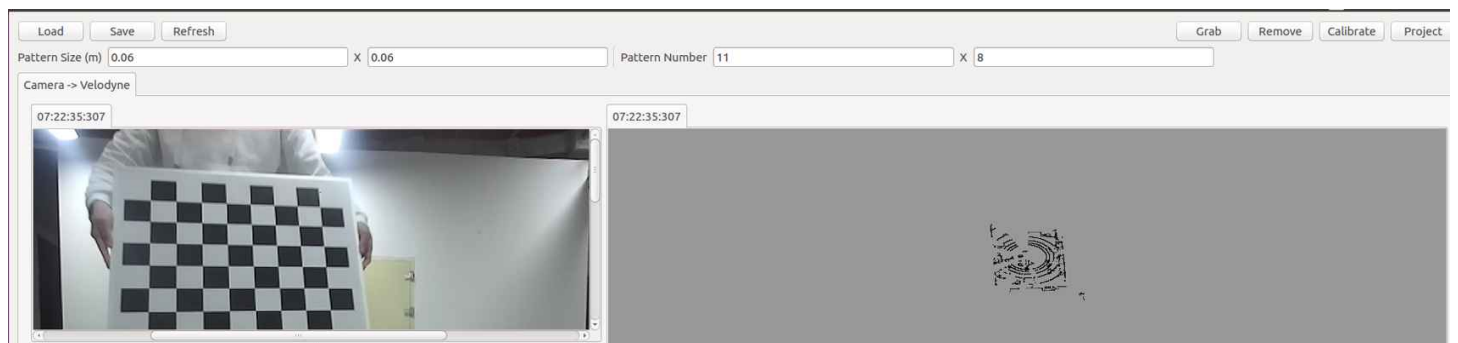
以回放bag包形式进行标定，所以不加载相机内参

打开回放 bag 终端，按空格继续回放数据，主界面会显示相机图像，但是刚打开时点云图像是黑色框，需要调整视角

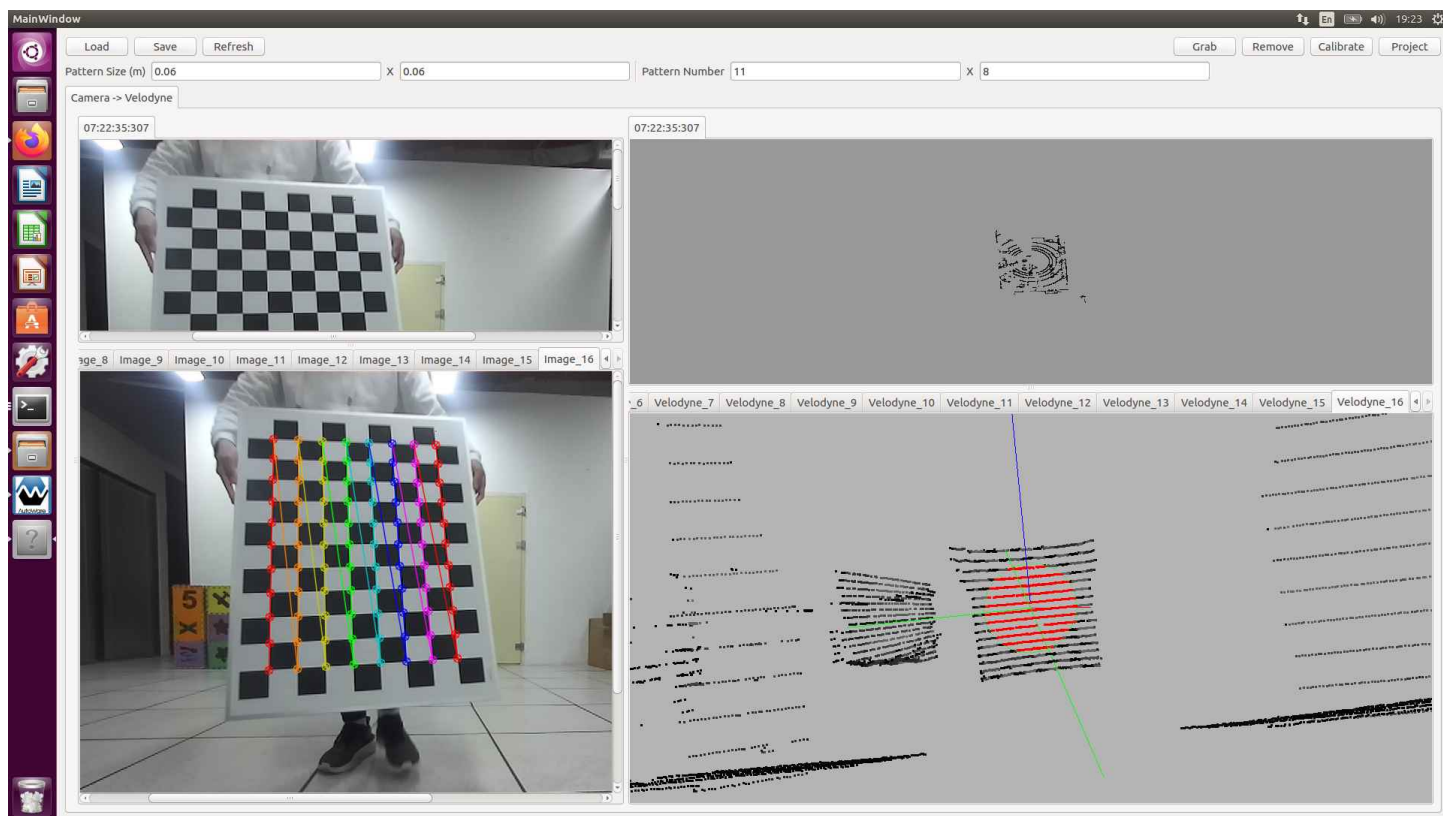
- 移动点云：上下左右方向键、PgUp、PgDn
 - 旋转点云：a、d、w、s、q、e
 - 切换模式：数字 1 和数字 2
 - 视角缩放：减号缩小、加号放大
 - 点云大小：o 键使用小点云、p 使用大点云
 - 改变点云窗口背景颜色：b
- 先调节点云界面，右上框的颜色，默认是黑色，点一下该框，按下B键，换成浅色调灰色（便于观察）。



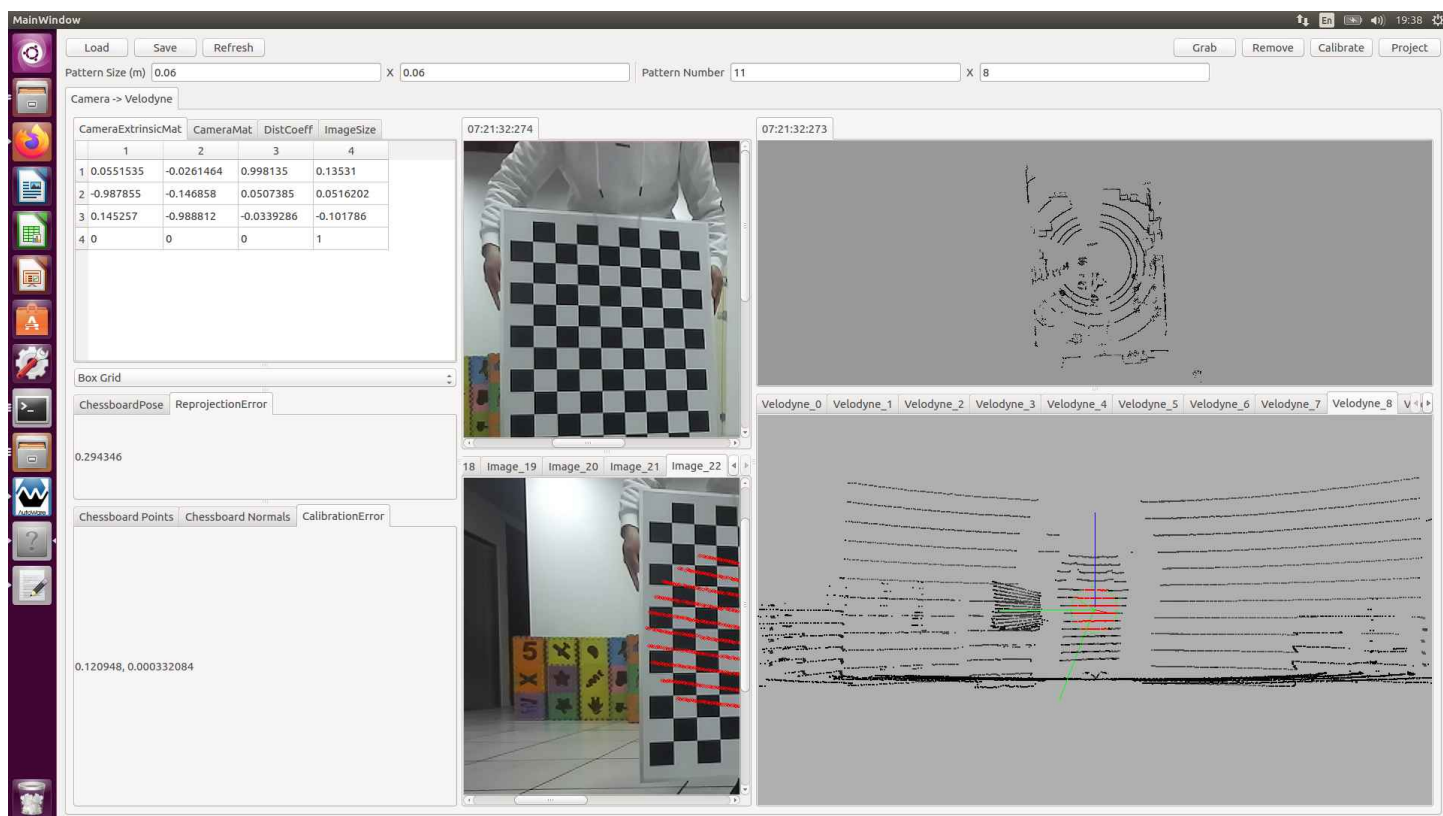
点击右边点云图，直接按数字2切换模式就可以看到点云了



有点云后，点击grab，然后把鼠标放到右下角捕获的点云窗口，选择一个棋盘格的中心位置区域，关于这个区域的选择，可以看参考链接的youtube视频，大概就是标定板的中心位置选择一个圆形的区域，尽量保证向外侧的平面法向量垂直于标定板平面：

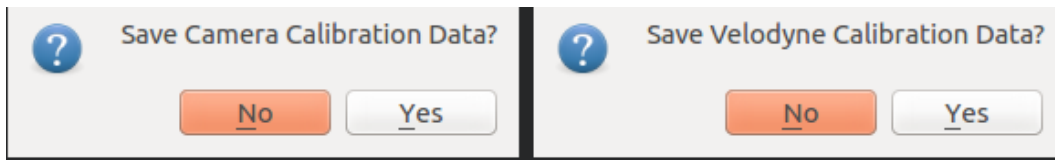


多重复几次，我这里grab二十多张点云图，然后点击右上角calibrate，计算完之后会在左边得到外参矩阵、重投影误差等参数（重投影误差越小越好）



然后点击右上角的“project”进行验证，会根据计算结果和激光雷达数据生成的图像对应位置，以红色散点表示，如果红色散点分布到标定板上，则说明正确，如果没有，就重新选点。

弄完之后没有问题，就点击左上方的“Save”，将结果保存到合适位置，保存过程中下面两个弹窗都选“NO”



标定结果

zed2_velodyne_calibr.yaml

```
1 %YAML:1.0
2 ---
3 CameraExtrinsicMat: !!opencv-matrix
4   rows: 4
5   cols: 4
6   dt: d
7   data: [ 5.5153500114773824e-02, -2.6146411694234428e-02,
8           9.9813549009170355e-01, 1.3531029419619076e-01,
9           -9.8785545439244893e-01, -1.4685777961060986e-01,
10          5.0738484357830960e-02, 5.1620165999774995e-02,
11          1.4525733252465411e-01, -9.8881199311262269e-01,
12          -3.3928595969549225e-02, -1.0178619198924406e-01, 0., 0., 0., 1. ]
13 CameraMat: !!opencv-matrix
14   rows: 3
15   cols: 3
16   dt: d
17   data: [ 5.2201708970269010e+02, 0., 6.4377972286274519e+02, 0.,
18           5.2286074789902261e+02, 3.6135606900407493e+02, 0., 0., 1. ]
19 DistCoeff: !!opencv-matrix
20   rows: 1
21   cols: 5
22   dt: d
23   data: [ -5.5469307283085682e-02, 6.1340617290566321e-02,
24           4.2243435379226758e-04, 8.0374552476056359e-04,
25           1.7713925213593901e-02 ]
26 ImageSize: [ 1280, 720 ]
27 ReprojectionError: 2.9434582277075816e-01
```

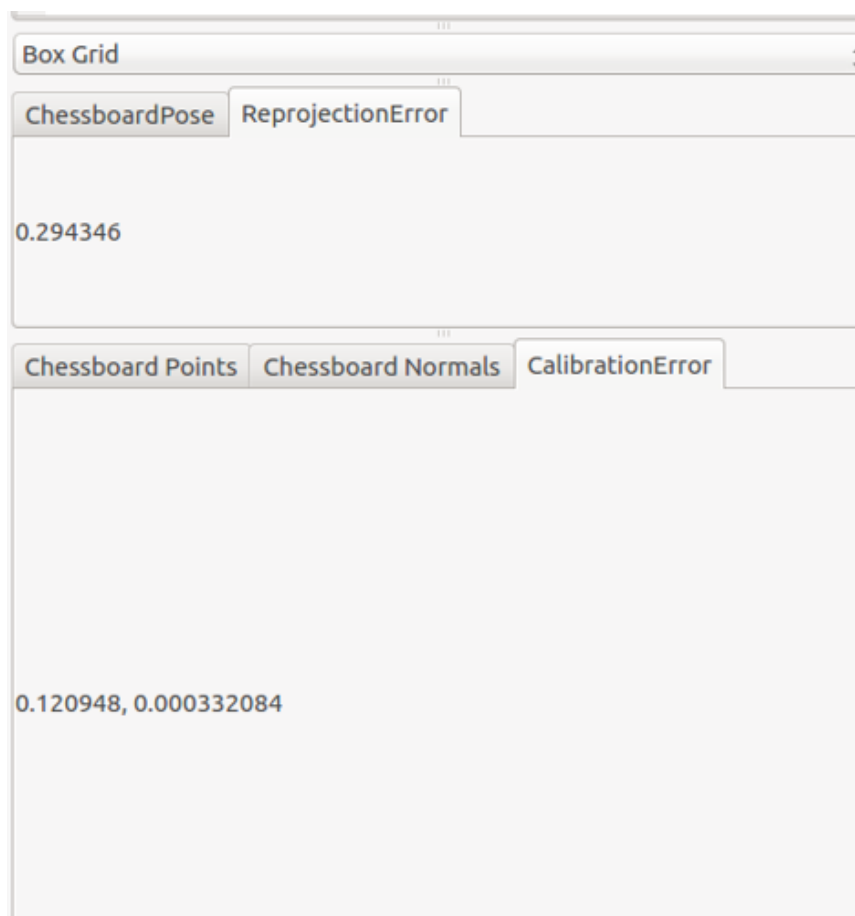
解释如下：

- 1 CameraExtrinsicMat: 【表示外参矩阵】 4*4 矩阵 左边3*3表示旋转外参，最右边那列表示平移外参
- 2
- 3 CameraMat: 【表示相机内参】 3*3矩阵，里面的系数依次为:
- 4
$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$
- 5
- 6
- 7
- 8 DistCoeff: 【表示畸变系数】， 1*5 矩阵，里面的系数依次为:
- 9 径向畸变系数 k1 k2 k3 以及切向畸变系数 p1 p2
- 10
- 11 ReprojectionError: 【重投影误差】

重投影误差 0.29[像素]，激光-相机标定误差0.12【像素】

解释:

- 1 重投影误差: 3D激光雷达点投影到标定板上与相机的观测2D点作差
- 2
- 3 标定误差: 相机的观测2D点补偿重投影误差后，与3D激光雷达点作差



查看融合标定结果

将我们标定的结果用上，将三维点云数据投影到图像上，此处借助Autoware自带的`Calibration Publisher`和`Points Image`节点能进行融合。如果是一体化的Autoware就在Calibration tool下面就有这两个节点选择，在Autoware/Sensing/中打开Calibration Publisher，并将保存好的内外参文件读取进来，打开Autoware/Sensing下的Point Image节点，在Rviz中，Panels/Add New Panel内添加ImageViewerPlugin，填写Image Topic，本例为/camrea/rgb/image_raw，Point Topic，本例为/points_image

继续播放上面录制bag包，可以看到融合标定图

