

# ROS 相机自动标定程序

## ROS标定程序启动流程:

### 启动ROS:

```
roscore
```

### 启动相机:

```
roslaunch /ud_perception_ws/params/ud-launch/ud_drivers/ros_camera1.launch
```

其中 `ros_camera*.launch` 对应需要标定的相机程序\*

### 启动标定程序\*:

```
roslaunch camera_calibration cameracalibrator.py --size 8x6 --square 0.025  
image:=/tztek/camera1 camera:=/tztek --no-service-check
```

### 参数设置:

`--size`: 棋盘格的角点数 (8x6角点数应对应9x7棋盘格)

`--square`: 棋盘格中每个方块的实际边长, 单位为米

`image`: ROS图像句柄, `camera*`应与启动相机命令中的相机号相对应

`camera`: ROS相机信息, 可通过 `rostopic list` 查看

`--no-service-check`: 关闭相机信息启动检查

### 标定流程:

1. 使标定板放置与摄像头视野中, 当标定程序识别出棋盘格后会将棋盘格角点以彩色圆圈标注\*
2. 保持标定板被程序识别出的状态下移动标定板
3. 通过将标定板沿着它的x轴和y轴平移, 使界面右侧上部 `x` 及 `y` 计量槽变绿并不再增加
4. 通过将标定板绕着它的x轴和y轴转动, 使界面右侧上部 `skew` 计量槽变绿并不再增加
5. 通过前后移动标定板, 使界面右侧上部 `size` 计量槽变绿并不再增加
6. 当四个计量槽全部变成绿色且不再增长后, 点击右侧下部 `CALIBRATE` 按钮开始执行ROS标定程序
7. 等待ROS标定完成后, 点击右侧下部 `SAVE` 按钮保存标定图片及ROS结果  
至 `/tmp/calibrationdata.tar.gz` 文件中, 不要点击其他按钮

### 重投影误差计算:

#### 执行误差计算程序:

```
python3 camera_calibration.py
```

等待误差计算程序运行完成后检查程序的打印输出项中的重投影误差 (`Opencv mean error:`)

标定合格判断标准：

-重投影误差应小于0.5，重投影误差越小标定效果越好

重投影误差计算程序将会自动复制标定文件至当前目录中，再以OPENCV算法计算标定的重投影误差并与ROS算法计算的重投影误差进行比较，之后将相机的内参根据相机名称以 yam1 格式保存至 `./calibration_result` 文件夹中，同时将标定图像文件备份保存至 `backup_data` 文件夹中。

备注事项:

\*如果相机无法启动，可尝试使用命令查看相机是否可用：

```
tztek-jetson-tool-camera-show-cuda /dev/video0 1920 1080
```

\*如果棋盘格未被识别出：

1. `--size 8x6` 命令中的 `x` 为小写英文字母x
2. 可能为棋盘格角点数错误，应当根据实际棋盘格角点数修改启动命令中的 `--size 8x6` 的参数后重新启动标定程序。

\*如果标定程序无法启动，可尝试使用命令安装相机标定功能包：

```
sudo apt-get install ros-melodic-camera-calibration
```

## 相机标定的注意事项:

1. 制作棋盘格标定板时，黑色方格与白色方格尺寸需要相同，所有方格尺寸一致；
2. 采集标定板图像时，应采集多幅不同位姿的标定板图像，采集图像数量不应太少；
3. 相机标定程序能够自动计算重投影误差，重投影误差越小表示相机标定的精度越高；
4. 标定板在图像中最好能够覆盖整个视野（覆盖图像的每个角落可以使得径向畸变系数更准确）；
5. 标定板覆盖较大的深度范围（将标定板绕它的x轴和y轴旋转或者放置在不同距离的位置上）；

## 程序错误解决方法:

**1.如果启动标定程序时遇到undefined symbol: v4l2\_open 错误，需要重新安装ffmpeg**

参考 (<https://forums.developer.nvidia.com/t/jetson-tx2-and-ffmpeg-cant-initialize-nvrm-channel/165629/3>)

### 手动更换ffmpeg版本:

版本过低会有open\_v4l2报错

```
sudo ffmpeg --version
```

```
sudo apt purge ffmpeg
```

解压 `./ffmpeg/ffmpeg-4.2.7` 版本源码到home文件夹：

```
tar xf ffmpeg-4.2.7.tar.xz -C ~/
```

ffmpeg-4.2.7 文件夹中配置安装参数：

```
cd /home/nvidia/ffmpeg/

./configure --enable-libv4l2 --enable-shared --extra-libs="-L/usr/lib/aarch64-
linux-gnu/tegra -lnvbuf_utils" --extra-cflags="-I
/usr/src/jetson_multimedia_api/include/"

sudo make

sudo make install
```

添加 LD\_LIBRARY\_PATH 中:

```
vim ~/.bashrc
```

在 bashrc 文件中添加此项:

```
export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
```

```
source ~/.bashrc
```

```
sudo ldconfig
```

**2.如果依然出现undefined symbol: v4l2\_open，使用libavcodec.so.58文件替换usr/local/lib/中的源文件**

```
sudo cp ./ffmpeg/libavcodec.so.58 /usr/local/lib/
```