3.5.1 相机的安装

OAK-D-Pro 相机搭载了红外夜视 IR 模组、高像素 RGB 摄像头、深度摄像头模组，能够实现对目标的深度信息移动测距。首先检查机械臂相机支架安装，接着将相机与计算机连接好。如下图所示：

图 3-7 相机的安装

3.5.2 相机连接

由于 OAK 是 USB 设备，因此为了在使用 udev 工具的系统上与之通信，您需要添加 udev 规则以使设备可访问。

以下命令将向您的系统添加新的 udev 规则：

```
echo 'SUBSYSTEM=="usb", ATTRS{idVendor}=="03e7", MODE="0666"' | sudo tee /etc/udev/rules.d/80-movidius.rules
sudo udevadm control --reload-rules && sudo udevadm trigger
```

提示：第一次使用一定要配置此规则！

安装 depthai 驱动及其依赖(这一步已经安装好)：

```
git clone https://gitee.com/oakchina/depthai.git
cd depthai
python3 install_requirements.py
```

终端在 depthai 目录下，运行 Demo：

```
python3 depthai_demo.py
```

可见相机连接成功：

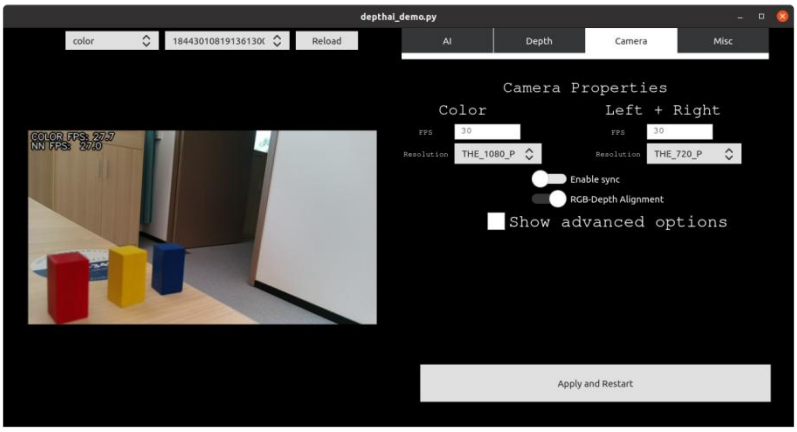


图 3-8 相机的成像

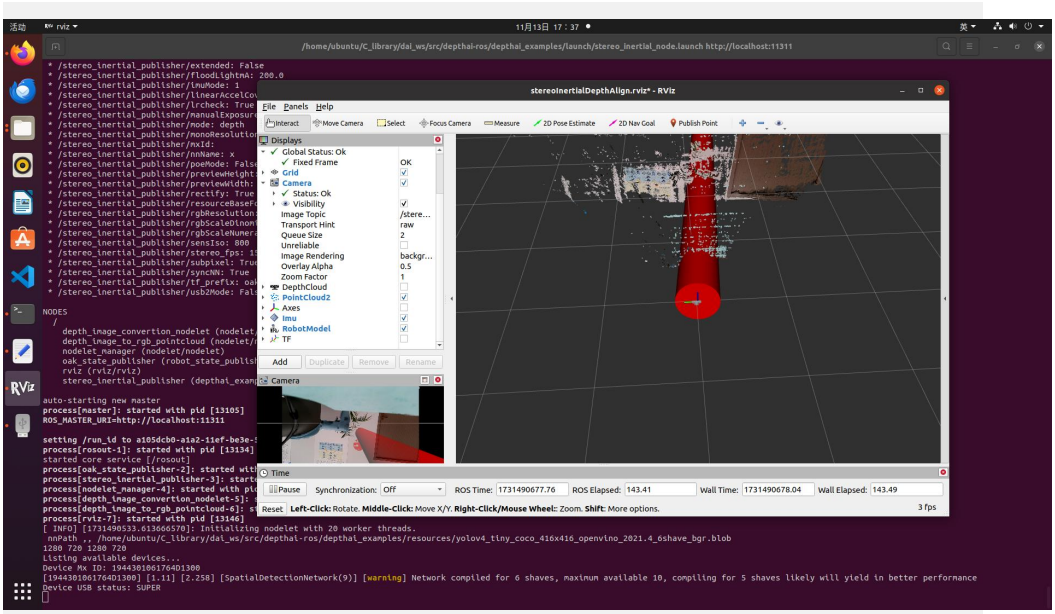
在 ROS 中使用相机：

安装驱动包(这一步已经安装好)：

```
mkdir -p dai_ws/src
cd dai_ws/src
git clone https://gitee.com/oakchina/depthai-ros.git
cd ..
source devel/setup.bash
source /opt/ros/noetic/setup.bash
sudo apt install ros-noetic-depthai-ros
sudo apt install ros-noetic-rviz-imu-plugin -y
catkin_make
source devel/setup.bash
```

执行示例

```
cd dai_ws
source devel/setup.bash
roslaunch depthai_examples stereo_inertial_node.launch
```



3.5.3 相机标定

3.5.3.1 相机标定（RGB 相机）

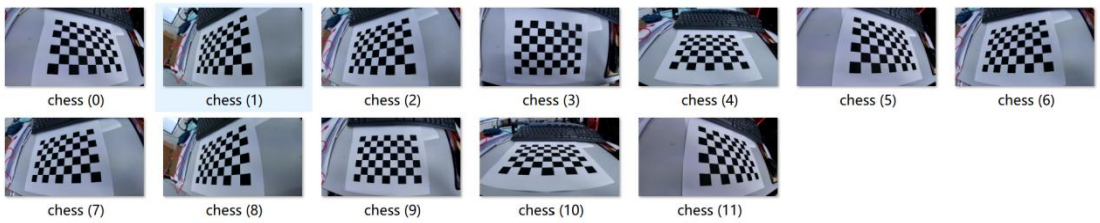
1、使用准备的棋盘格，棋盘格大小已知，把它贴在一个平面上，作为标定物。棋盘格生成来源（Camera Calibration Pattern Generator - calib.io）

2、通过调整标定物或摄像机的方向，为标定物拍摄一些不同方向的照片。

注意：

1) 从不同的角度和距离拍摄 10 张以上，越多越好；
2) 注意图像中棋盘格的方向不要改变，即横纵向不能转换；
3) 保证标定板的亮度足够且均匀；
4) 标定板成像不能过爆，过爆会导致特征轮廓的提取的偏移，从而导致圆心提取不准确；
5) 标定过程，相机的光圈、焦距不能发生改变，改变需要重新标定；
6) 标定板拍摄的张数要能覆盖整个测量空间及整个测量视场，把相机图像分成四个象限，应保证拍摄的标定板图像均匀分布在四个象限中，且在每个象限中建议进行不同方向的两次倾斜。

把图片保存至 chess 文件夹中，把原有文件删掉。图片需以 JPG 或 png 或 jpg 为后缀。



3、相机标定部分参考代码：

```
git clone https://gitee.com/tuotuowj/camera_calibration_tool.git
```

4、从照片中提取棋盘格角点进行检测，得到标定板角点的像素坐标值，根据已知的棋盘格大小和世界坐标系原点，计算得到标定板角点的物理坐标值；完成代码后，执行程序：

```
python3 calibration.py --image_size 1920x1080 --mode calibrate --corner 8x6 --square 20
```

注意修改参数，将以下参数替换成自己的：

image_size：图像分辨率，宽 x 高

corner：棋盘格角点，宽 x 高

square：棋盘格中每个格子的实际宽度，单位：mm

1、运行结束没有报错，相机参数将会保存在：camera_params.xml

2、理解 camera_params.xml 的参数。

3、学习用 camera_params.xml 的参数来矫正图像或者视频

3.5.3.2 相机标定（双目相机）（进阶实验，选做，不做要求）

我们通过将 charuco 板显示在电视或大型（平面，非曲面）显示器上。根据屏幕尺寸（对角线，以英寸为单位），我们建议全屏显示以下 charuco 板：

24 寸屏幕：charuco_24inch_13x7

28 寸屏幕：charuco_28inch_15x8

32 寸屏幕：charuco_24inch_17x9

36 寸屏幕：charuco_36inch_19x11

如果你用的是其他尺寸的屏幕，建议选比屏幕稍小的 charuco 板。

如果无法在显示器上显示 charuco 板，你也可以选择已经打印的 charuco 板。全屏显示 charuco 板时，标记和方块应清晰可见。注意以下几点：

- 屏幕不应太亮（或太暗），因为它会导致图像过度饱和，这将使相机更难检

测到标记。

- 不要让明亮的灯光/阳光直接照射在屏幕上。
- 全屏显示 charuco 板，使标记尽可能大。

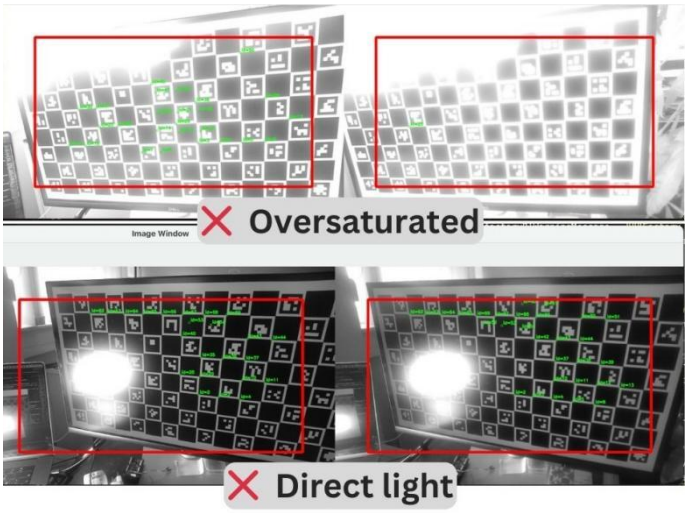


图 3-9 注意事项

设置给定的 charuco 板的正方形尺寸，配置 json 文件。

运行校准脚本：

将占位符参数值替换为有效条目：

```
python3 calibrate.py -s [SQUARE_SIZE_IN_CM] --board [BOARD] -nx [squaresX] -ny [squaresY]
```

例如，在 32‘’屏幕上校准 OAK-D-Pro，正方形尺寸为 3.76cm，应该运行如下命令：

```
python3 calibrate.py -s 3.76 --board OAK-D-Pro -nx 17 -ny 9
```

运行 python3 calibrate.py --help 可以查看更多参数解释。

我们建议从不同的角度和距离捕获校准，因为这将有助于校准算法找到最佳校准。

1、靠近屏幕：校准板几乎覆盖了整个视场。拍摄 5 张照片以覆盖相机的整个 FOV。

1）前视图，FOV 中间的校准板（参考下图）。

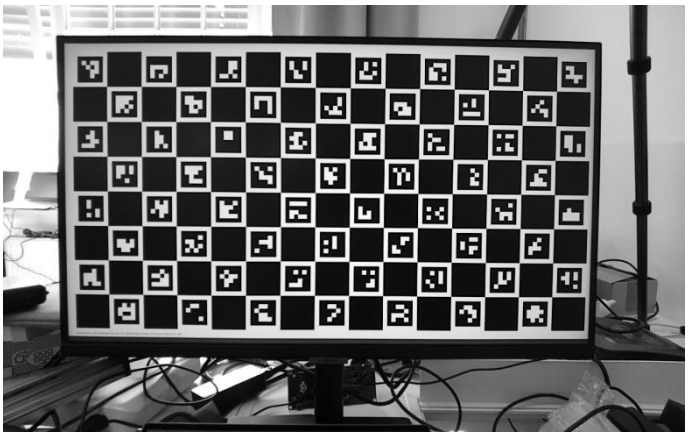


图 3-10 前视图

2）在不移动（平移）相机的情况下，只需旋转即可将相机 FOV 与校准板边缘对齐（例如：右下角、左上角、右上角、左下角）

2、靠近屏幕：从侧面。校准板倾斜的 4 张或更多图像，但仍覆盖大部分 FOV。将相机移动到屏幕的顶部、底部、左侧和右侧。你也可以使用不同的距离。

3、中距离：校准板覆盖 40% 的 FOV。拍摄 5 张图像以覆盖相机的整个 FOV。

1）前视图，FOV 中间的校准板。

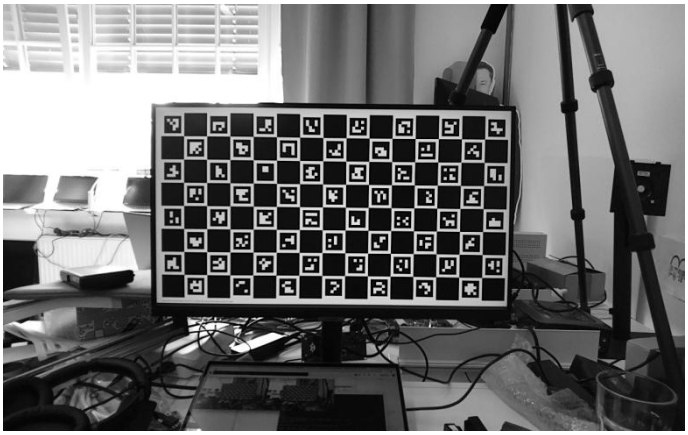


图 3-11 中距离前视图

2）与 2.靠近屏幕 一样，无需移动，仅通过旋转将相机 FOV 与校准板边缘对齐。

4、远离屏幕：校准板仅覆盖 FOV 的一小部分。总共拍摄 9 张图像以覆盖相机的整个 FOV。

1）正面视图，FOV 中间的校准板。

图 3-12 远距离前视图

2）与 2.靠近屏幕 类似，拍摄 4 张图像，将相机 FOV 对齐到所有 4 个边缘。

图 3-13 左侧顶部边缘对齐前视图

3）除了与所有 4 个边缘对齐外，还要拍摄 4 张与角对齐的图像（例如：顶部、底部、左侧、右侧）。

运行校准：

每个捕获的图像都保存在 dataset 文件夹中，捕获图像后，我们可以运行校准。

```
python3 calibrate.py -s 3.76 --board OAK-D-S2 -nx 17 -ny 9 -m proc
ess
```

校准结果存储在 resources 文件夹中，保存数据。