

# 1. Instructions for using camera

## 1. Use opencv and usb\_cam to drive camera error

After using the launch command in the Orbbec\_SDK function package to drive the camera, you will find that the camera cannot be driven with usb\_cam, and an error will be reported as shown in the figure below.

```
process[image_view-3]: started with pid [10766]
[ INFO] [1697529235.315517776]: Initializing nodelet with 4 worker threads.
[ INFO] [1697529235.405725041]: Using transport "raw"
[ INFO] [1697529235.425513328]: using default calibration URL
[ INFO] [1697529235.426148155]: camera calibration URL: file:///home/yahboom/.ros/camera_info/head_camera.yaml
[ WARN] [1697529235.426534453]: [head_camera] does not match name narrow_stereo in file /home/yahboom/.ros/camera_info/head_camera.yaml
[ INFO] [1697529235.426583374]: Starting 'head_camera' (/dev/video0) at 640x480 via mmap (yuyv) at 30 FPS
[ERROR] [1697529235.426640067]: Cannot identify '/dev/video0': 2, No such file or directory
[usb_cam-2] process has died [pid 10765, exit code 1, cmd /opt/ros/noetic/lib/usb_cam/usb_cam_node __name:=usb_cam __log:=/home/yahboom/.ros/log/51dd48e6-6cc2-11ee-b19d-65f5d1e636b5/usb_cam-2.log].
log file: /home/yahboom/.ros/log/51dd48e6-6cc2-11ee-b19d-65f5d1e636b5/usb_cam-2*.log
```

Solution: Re-plug the camera, and then use usb\_cam to open the camera normally, including using opencv to open the camera. If the above error occurs, just re-plug it. As long as you use launch in Orbbec\_SDK to drive the camera and before using opencv to drive the camera, you need to unplug it again.

## 2. Virtual machine case demonstration instructions

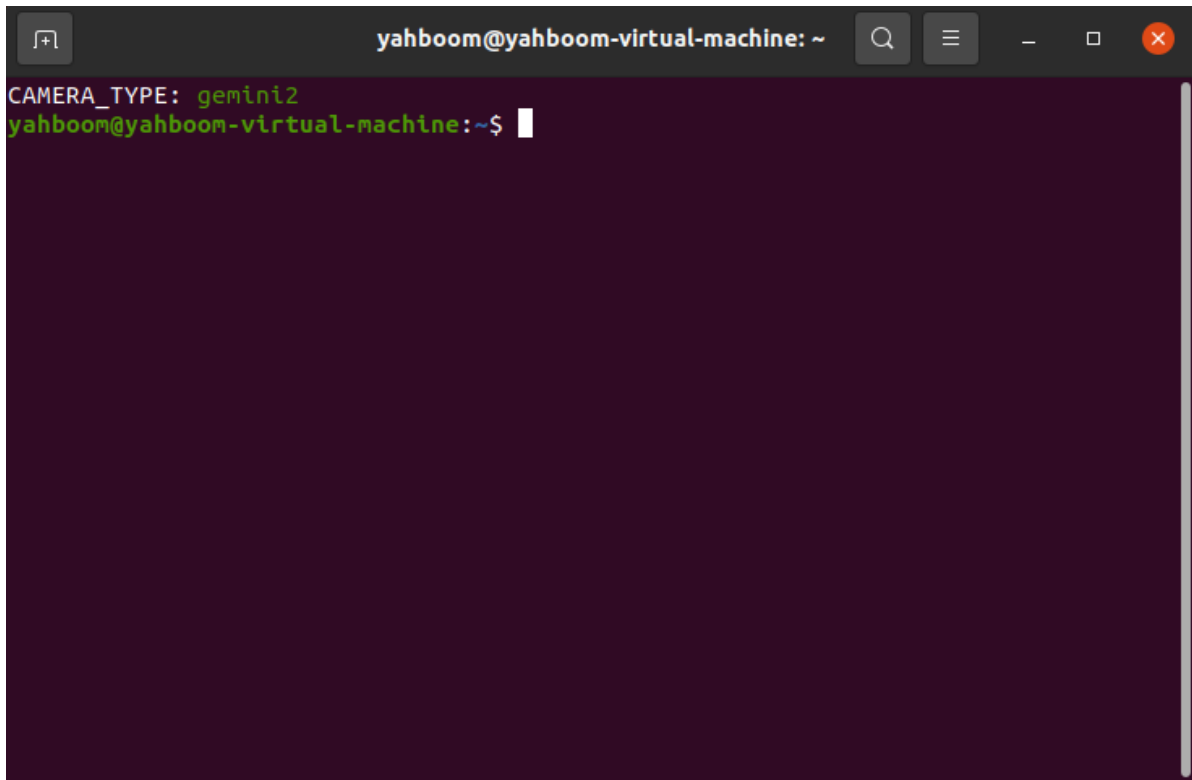
The cases in the virtual machine include SDKs for multiple cameras. Therefore, when matching the cases in the virtual machine experiment tutorial, you need to set the camera file of the car model through the ~/.bashrc file. You need to set [CAMERA\_TYPE] to the purchased camera model. Assume If you purchased a gemini2 camera, you need to modify the value of [CAMERA\_TYPE] to gemini2. Terminal input, (astrapropus camera is set to astrapropus)

```
sudo gedit ~/.bashrc
```

Line 140, [CAMERA\_TYPE] is set to gemini2.

```
export CAMERA_TYPE=gemini2
echo -e "CAMERA_TYPE: \033[32m$CAMERA_TYPE\033[0m"
source ~/ArTrack_ws/devel/setup.bash
source ~/orbbec_ws/devel/setup.bash
source ~/ros_ws/devel/setup.bash
export ROS_PACKAGE_PATH=${ROS_PACKAGE_PATH}:/home/yahboom/software/-ORB SLAM2/Examples/ROS/ORB SLAM2
```

Exit after saving and restart a terminal. The terminal will print out the set camera type.

A terminal window with a dark background. The title bar shows 'yahboom@yahboom-virtual-machine: ~' and standard window controls. The terminal content shows 'CAMERA\_TYPE: gemini2' on the first line and a prompt 'yahboom@yahboom-virtual-machine:~\$' on the second line with a cursor.

```
yahboom@yahboom-virtual-machine: ~  
CAMERA_TYPE: gemini2  
yahboom@yahboom-virtual-machine:~$
```

### 3. Source code description

We provide two sets of source code, one is the source code with only the camera SDK, and the other is the code with the camera SDK and some demos in the tutorial.

Orbbec\_ws\_src.tar.xz contains the function package to drive the camera and the function package to run the demo with the virtual machine.

Orbbec-ros-sdk.tar.xz only contains the function package to drive the camera.

opencv.zip contains only opencv related function packages