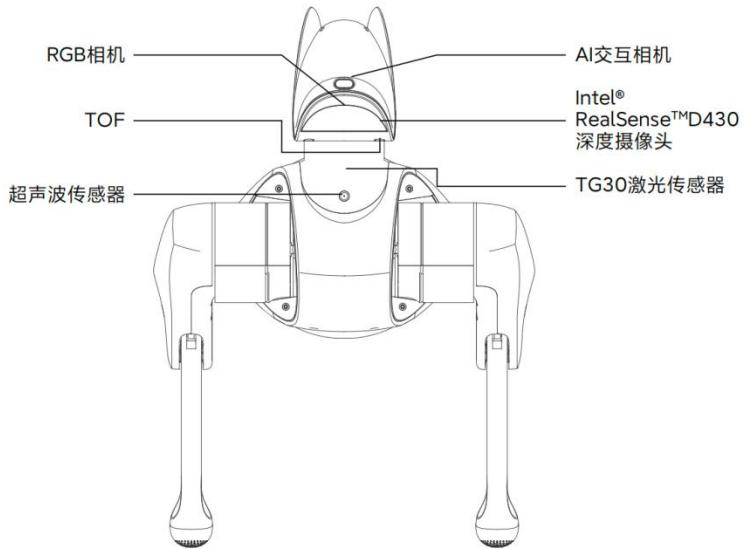


# Cyberdog 视觉功能

计算机视觉(Computer Vision)是一种利用计算机科学和数学方法处理、分析图像和视频等视觉数据的技术。在本次大赛中，计算机视觉可以为机器人提供感知能力，帮助其规划最优路径，避免障碍物并迅速穿越赛道。

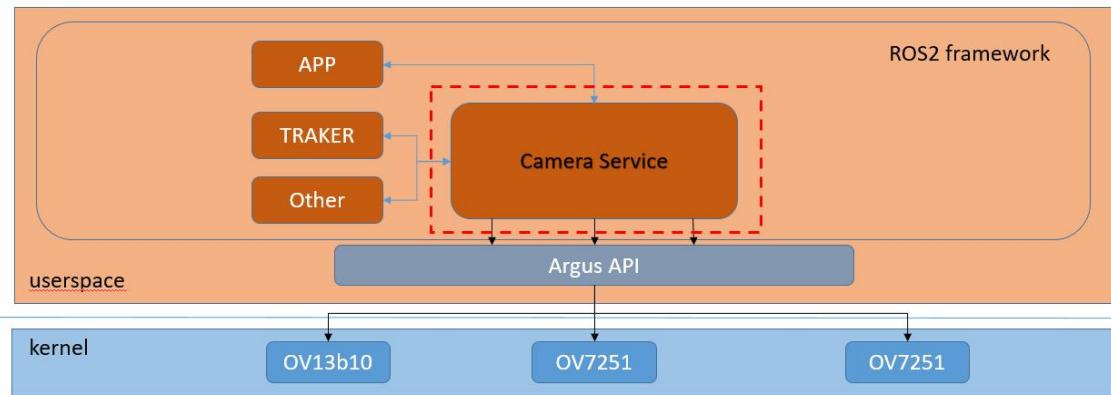
Cyberdog2 配备多个摄像头，涵盖了不同方向和视角，以全方位地捕获周围环境的图像。参赛队需要设计和实现系列算法，使机器人能够在图像中检测到不同类型的障碍物，识别后做出相应的动作调整。



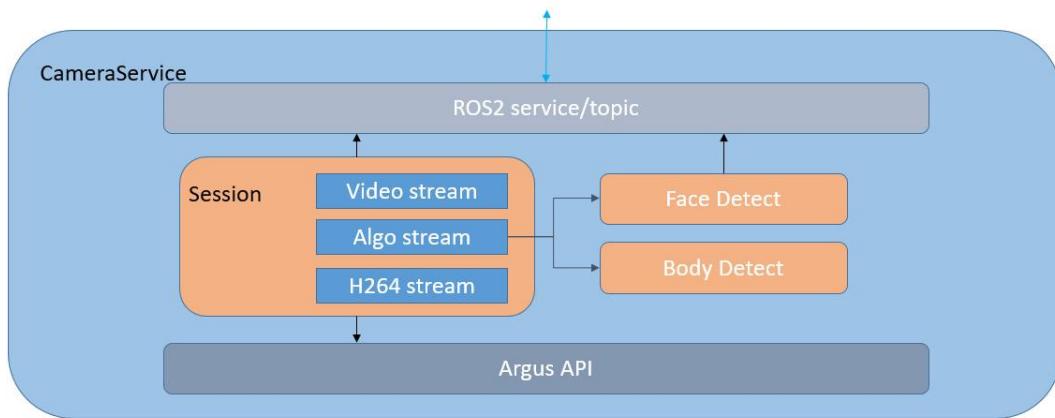
## 一、相机模块介绍

相机驱动模块使用 Argus API 提供的接口来操控 MIPI 相机硬件并实时捕获图像，使用 ROS2 提供的接口来管理相机节点，为外部模块提供交互接口。

模块整体架构图如下：



模块内部描述图如下：



## 1. 安装依赖项

编译本模块需要依赖若干外部软件包，编译前需按照下列命令安装：

1. **nvidia-l4t-jetson-multimedia-api**

```
sudo apt-get install nvidia-l4t-jetson-multimedia-api
```

2. **cuda-toolkit**

```
sudo apt-get install cuda-toolkit-10-2
```

3. **libavformat-dev**

```
sudo apt-get install libavformat-dev
```

## 2. 运行测试程序

测试程序地址:

[https://github.com/MiRoboticsLab/cyberdog\\_ros2/tree/main/main/cyberdog\\_interaction/cyberdog\\_camera/cyberdog\\_camera/src/camera\\_test](https://github.com/MiRoboticsLab/cyberdog_ros2/tree/main/main/cyberdog_interaction/cyberdog_camera/cyberdog_camera/src/camera_test)

基于相机 api 的相机测试程序，可以用来测试相机是否正常，亦可以作为 camera api 使用方式参考。

编译:

```
colcon build --merge-install --packages-up-to camera_test
```

运行:

```
./build/camera_test/camera_test cam_id width height rgb/bgr
```

例如，测试 camera 0， 640x480 分辨率 RGB 出图的话，使用如下命令：

```
./build/camera_test/camera_test 0 640 480 rgb
```

## 二、RGB 相机 Demo

### 1. 修改 gazebo.xacro

在 gazebo.xacro 中添加并保存

```
<gazebo reference="RGB_camera_link">
  <sensor type="camera" name="rgb camera">
    <always_on>true</always_on>
    <update_rate>15.0</update_rate>
    <camera name="rgb_camera">
      <horizontal_fov>1.46608</horizontal_fov>
      <image>
        <width>320</width>
        <height>180</height>
        <format>R8G8B8</format>
      </image>
      <distortion>
        <k1>0.0</k1>
        <k2>0.0</k2>
        <k3>0.0</k3>
        <p1>0.0</p1>
```

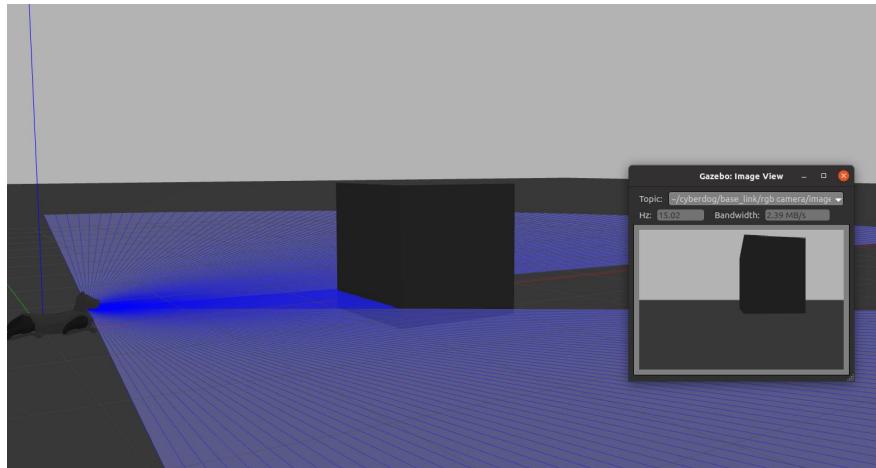
```

<p2>0.0</p2>
<center>0.5 0.5</center>
</distortion>
</camera>
<plugin name="rgb_camera_plugin" filename="libgazebo_ros_camera.so">
<ros>
<!-- <namespace>stereo</namespace> -->
<remapping>~/image_raw:=image_raw</remapping>
<remapping>~/camera_info:=camera_info</remapping>
</ros>
<!-- Set camera name. If empty, defaults to sensor name (i.e. "sensor_name") -->
<camera_name>rgb_camera</camera_name>
<!-- Set TF frame name. If empty, defaults to link name (i.e. "link_name") -->
<frame_name>RGB_camera_link</frame_name>
<hack_baseline>0.2</hack_baseline>
</plugin>
</sensor>
</gazebo>

```

## 2. 运行仿真程序

运行仿真程序后可通过 `window` 选项下的 `Topic Visualization` 中找到对应 topic 并打开, 可确认 RGB 相机正常运行



通过 `ros2 echo topic` 可确认 topic 正常发送

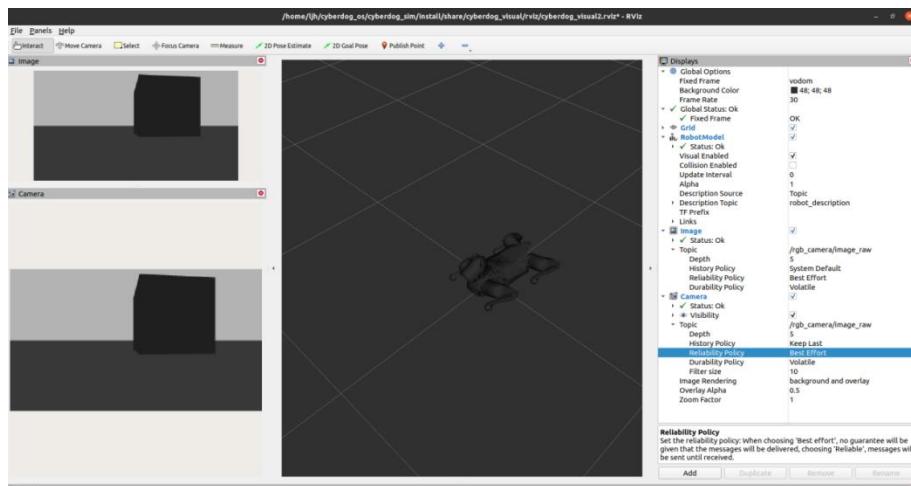
```

ljh@ljh-Precision-3640-Tower:~/cyberdog_os/cyberdog_sim$ ros2 topic list
/opt/ros/galactic/bin/ros2:6: DeprecationWarning: pkg_resources is deprecated as
an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import load_entry_point
/apply_force
/clock
 imu
/parameter_events
/performance_metrics
/rgb_camera/camera_info
/rgb_camera/image_raw
/rosout
/scan
/yaml_parameter

```

### 3. rviz 可视化

在 rviz2 中通过以下设置可将 topic 可视化



## 三、相机服务接口

CameraService.srv 配置

```

uint8 SET_PARAMETERS = 0

uint8 TAKE_PICTURE = 1

uint8 START_RECORDING = 2

uint8 STOP_RECORDING = 3

uint8 GET_STATE = 4

uint8 DELETE_FILE = 5

uint8 GET_ALL_FILES = 6

uint8 START_LIVE_STREAM = 7

```

```
uint8 STOP_LIVE_STREAM = 8

uint8 START_IMAGE_PUBLISH = 9

uint8 STOP_IMAGE_PUBLISH = 10

uint8 command

# command arguments

string args

uint16 width

uint16 height

uint16 fps

---

uint8 RESULT_SUCCESS = 0

uint8 RESULT_INVALID_ARGS = 1

uint8 RESULT_UNSUPPORTED = 2

uint8 RESULT_TIMEOUT = 3

uint8 RESULT_BUSY = 4

uint8 RESULT_INVALID_STATE = 5

uint8 RESULT_INNER_ERROR = 6

uint8 RESULT_UNDEFINED_ERROR = 255

uint8 result

string msg

int32 code
```

# RGB 鱼眼相机

```
# 打开、关闭 camera  
ros2 launch camera_test stereo_camera.py  
  
#查看 name space  
ros2 node list  
  
###如果是开机自启，注意 topic 前加上命名空间  
  
ros2 lifecycle set /stereo_camera configure  
  
ros2 lifecycle set /stereo_camera activate  
  
ros2 lifecycle set /stereo_camera deactivate  
  
ros2 lifecycle set /stereo_camera cleanupCopy to clipboardErrorCopied
```

## 获取图像

接口形式: ros topic

接口名称: camera\_server

topic 名称:

```
#使用 topic 订阅 rgb 相机、左鱼眼相机，右鱼眼相机  
  
#左鱼眼  
/image_left  
#右鱼眼
```

```
/image_right  
#rgb 相机  
/image_rgb Copy to clipboardErrorCopied
```

图像数据 topic 内容: sensor\_msgs::msg::Image

## AI 相机

```
# 打开、关闭 camera  
ros2 run camera_test camera_server  
  
##开启 AI 相机指令  
  
ros2 service call /camera_service protocol/srv/CameraService "{command: 9, width: 640,  
height: 480, fps: 0}"  
  
##关闭相机指令  
  
ros2 service call /camera_service protocol/srv/CameraService "{command: 10, args:  
"}"Copy to clipboardErrorCopied
```

## 获取图像

接口形式: ros topic

接口名称: camera\_server

topic 名称:

```
#使用 topic 订阅 AI 相机  
  
#AI 相机 topic  
  
topic: /imageCopy to clipboardErrorCopied
```

图像数据 topic 内容: sensor\_msgs::msg::Image

更多教程可参看官网[开发者手册](#)。