Sadaan Tahir - 25100224

Intel RealSense D435i as a ROS Node on Odroid-XU4 with Ubuntu 20.04 and ROS Noetic

Step 1. Launch the RealSense node (assuming the realsense2 package is installed):

```
roslaunch realsense2_camera rs_camera.launch
```

Step 2. Check for available topics:

```
rostopic list | grep camera
```

Step 3. These topics should be visible:

```
root@odroid:~# rostopic list | grep camera
/camera/color/camera_info
/camera/color/image_raw
/camera/color/metadata
/camera/depth/camera_info
/camera/depth/image_rect_raw
/camera/depth/metadata
/camera/extrinsics/depth_to_color
/camera/motion_module/parameter_descriptions
/camera/motion_module/parameter_updates
/camera/realsense2_camera_manager/bond
/camera/rgb_camera/auto_exposure_roi/parameter_descriptions
/camera/rgb_camera/auto_exposure_roi/parameter_updates
/camera/rgb_camera/parameter_descriptions
/camera/rgb_camera/parameter_updates
/camera/stereo_module/auto_exposure_roi/parameter_descriptions
/camera/stereo_module/auto_exposure_roi/parameter_updates
/camera/stereo_module/parameter_descriptions
/camera/stereo_module/parameter_updates
```

Step 4. To view the live feed, use these commands:

```
export LIBGL_ALWAYS_SOFTWARE=1
rosrun image_view image_view image:=/camera/color/image_raw
```

Step 5. Make a new workspace and a new package for this subscriber node using the following steps:

```
mkdir -p ~/testing_ws/src
cd ~/testing_ws
catkin_make

source devel/setup.bash
```

```
cd src
catkin_create_pkg marker_detector rospy std_msgs sensor_msgs cv_bridge
cd marker_detector

mkdir scripts
cd scripts
touch yolo_sub_test.py  # make the subscriber node python file
gedit yolo_sub_test.py  # paste the code given below
chmod +x scripts/yolo_sub_test.py   # make it executable
```

Step 6. The python script called **yolo_sub_test.py**:

```
import rospy
import cv2
import torch
import time
import contextlib
import numpy as np
from sensor_msgs.msg import Image
from cv_bridge import CvBridge

# Load YOLOv8 model
model_path = "~/Desktop/YOLOv8/best.pt"
model = torch.hub.load('ultralytics/yolov5', 'custom', path=model_path,
force_reload=True)

# Initialize CvBridge
bridge = CvBridge()

# Output directory
output_dir = "/home/odroid/Desktop/YOLOv8/"

# Image callback function
def image_callback(msg):
    try:
        # Convert ROS image message to OpenCV format
        cv_image = bridge.imgmsg_to_cv2(msg, "bgr8")

        # Run YOLO inference
        results = model(cv_image)

        # Process detection results
        for result in results:
            for box in result.boxes:
                conf = box.conf.item()
                if conf > 0.90:  # Confidence threshold
                    x1, y1, x2, y2 = map(int, box.xyxy[0])
                    center_x = (x1 + x2) // 2
                    center_y = (y1 + y2) // 2

                    # Draw bounding box and label
                    label = f"Marker {conf:.2f}"
                    cv2.rectangle(cv_image, (x1, y1), (x2, y2), (0, 255, 0), 2)
                    cv2.putText(cv_image, label, (x1, y1 - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

                    # Save snapshot
                    timestamp = int(time.time())
                    filename = f"{output_dir}/marker_{timestamp}.jpg"
                    cv2.imwrite(filename, cv_image)
                    rospy.loginfo(f"Marker detected! Confidence: {conf:.2f}, Image
saved: {filename}")

    except Exception as e:
        rospy.logerr(f"Error processing image: {e}")

# ROS node initialization
def main():
    rospy.init_node("marker_detector", anonymous=True)
    rospy.Subscriber("/camera/color/image_raw", Image, image_callback)
    rospy.loginfo("Marker detector node started. Waiting for images...")
    rospy.spin()

if __name__ == "__main__":
    main()
```

Step 7. Edit the CMakeLists.txt file to include:

```
find_package(catkin REQUIRED COMPONENTS
  rospy
  std_msgs
  sensor_msgs
  cv_bridge
)
```

Step 8. Modify package.xml to include these dependencies:

```
<depend>rospy</depend>
<depend>std_msgs</depend>
<depend>sensor_msgs</depend>
<depend>cv_bridge</depend>
```

Step 9. Build the package:

```
cd ~/testing_ws
```

```
catkin_make
source devel/setup.bash
```

Step 10. Run the subscriber node:

```
rosrun marker_detector yolo_sub_test.py
```