ROBT502 – ROBOT PERCEPTION AND VISION

Laboratory Report 7

Z. Kuangaliyev, A. Amangeldi, S. Omirbayev and K. Kantoreyeva

11 April 2024

1. Introduction

This report presents the processes and results of working with cameras in ROS and creating a web video streaming.

2. Setting up rosbridge

The provided tutorial explains how to use rosbridge server to communicate with ROS using web browser.

First, we started by creating a workspace for our project on one of the laptops connected to two cameras located in the labyrinth:

```
mkdir -p ~/ros_ws/src
cd ~/ros_ws/src
catkin create pkg robot_gui_bridge --catkin-deps rosbridge_server
cd ~/ros_ws
rosdep install --from-paths src --ignore-src -r -y
```

We then have created a launch file to see if everything is working correctly:

```
cd ~/ros_ws/
catkin_make
source devel/setup.bash
roslaunch robot_gui_bridge websocket.launch
```

After that, we started working with the cameras.

3. Working with camera

After making sure that the cameras are connected and ready for access, we check one camera’s device node path:

```
ls /dev/video*
```

Knowing device node path of the camera, we then check the USB attributes of the camera using the command:

```
udevadm info --name=/dev/video2 --attribute-walk
```

We write down the needed parameters: `ATTRS{idProduct}` and `ATTRS{idVendor}`.

Knowing this information, we add a Udev rule for our camera by creating a file */etc/udev/rules.d/99-uvc.rules* with the following contents:

```
SUBSYSTEMS=="usb", ENV{DEVTYPE}=="usb_device", ATTRS{idVendor}=="1acf",
ATTRS{idProduct}=="2384", MODE="0666"
```

And then reload the Udev rules:

```
sudo udevadm control --reload-rules
```

We have installed libuvc_camera using the following command:

```
sudo apt-get install ros-noetic-libuvc-camera
```

Then, we run the minimal launch file provided in the tutorial, with some changes in the parameters.

```
roslaunch my_camera otp.launch
```

After this step, we see that it successfully streams the video from the camera into the web server.

4. Streaming from two cameras

We have tried different methods to stream from two cameras. We tried creating two launch files, group of nodes withing one launch file, stitching two cameras but none of them worked.

Our resulting launch files is provided below:

```
<launch>

  <!-- Nodelet Manager for Camera 1 -->

  <node pkg="nodelet" type="nodelet" name="nodelet_manager_cam1"
args="manager" output="screen"/>


  <!-- Camera 1 Nodelet -->

  <node pkg="nodelet" type="nodelet" name="libuvc_camera_cam1" args="load
libuvc_camera/driver /nodelet_manager_cam1" output="screen">

    <param name="frame_id" value="elp_camera1"/>

    <param name="vendor" value="0x1acf"/>

    <param name="product" value="0x2384"/>
    <param name="width" value="640"/>
    <param name="height" value="480"/>
    <param name="video_mode" value="vp8"/>
    <param name="frame_rate" value="30"/>
    <!-- Ensure the topic is uniquely named or namespaced for the camera -->
    <remap from="image_raw" to="/camera1/image_raw"/>
  </node>

  <!-- Nodelet Manager for Camera 2 -->
  <node pkg="nodelet" type="nodelet" name="nodelet_manager_cam2"
args="manager" output="screen"/>

  <!-- Camera 2 Nodelet -->
  <node pkg="nodelet" type="nodelet" name="libuvc_camera_cam2" args="load
libuvc_camera/driver /nodelet_manager_cam2" output="screen">
    <param name="frame_id" value="elp_camera2"/>
    <param name="vendor" value="0x2113"/>
    <param name="product" value="0x0145"/>
    <param name="width" value="640"/>
    <param name="height" value="480"/>
    <param name="video_mode" value="vp8"/>
    <param name="frame_rate" value="30"/>
    <!-- Ensure the topic is uniquely named or namespaced for the camera -->
    <remap from="image_raw" to="/camera2/image_raw"/>
  </node>

  <!-- MJPEG Server
  <node name="web_video_server" pkg="web_video_server"
type="web_video_server" output="screen"/> -->

</launch>
```

This launch file will open two viewers for two cameras. Here are the steps for executing the launch file using rqt:

1) `roscore`

2) `rqt --force-discover` (Run RosGUI)

3) `catkin_make`

4) `source devel/setup.bash`

3) `roslaunch my_camera otp1.launch`

4) In RosGUI, in the top bar menu, we go to plugins and select "Image viewer" 2 times and for each viewer we assign its own camera

The video of the streaming from two cameras can be found following this link:

https://drive.google.com/file/d/1NwyN244VgFQt9jcFfSAvEPlaZe3syzKm/view?usp=sharing