


Overview

В данном репозитории находится необходимый код для запуска ROS узлов для задач компьютерного зрения в пайплайне. В частности, здесь находятся узлы для:

- публикация списка категорий и их номеров из плана LLM
- сегментации с помощью YoloV8 (только кубики)
- сегментации с помощью OpenSeeD
- 2D-Трекинга объектов
- визуализации облаков точек всех объектов
- извлечения облака точек объекта
- 3D трекинга объекта
- определения 6DoF позы объекта с использованием 3D моделей объектов

Проект развёрнут в docker-контейнере, запускается с помощью документа `docker_run.sh`, одновременно монтируется папка `resources` :



```
1 xhost +;
2 DOCKER_BUILDKIT=0 docker build -t noetic_full . ; \
3 docker build -t noetic_full . ; \
4 docker run -it --name ros --rm -e DISPLAY=:1 \
5     -v /tmp/.X11-unix:/tmp/.X11-unix --gpus all \
6     --mount type=bind,source=/home/cogmodel/ROS/docker_data/,target=/docker_data \
7     --mount type=bind,source=/home/cogmodel/ROS/resources/data,target=/resources/data \
8     --mount type=bind,source=/home/cogmodel/ROS/resources/models,target=/resources/models \
9     --mount type=bind,source=/home/cogmodel/ROS/resources/cache_clip,target=/root/.cache/clip \
10    --mount type=bind,source=/home/cogmodel/ROS/resources/cache_torch,target=/root/.torch/iopath_cache/ \
11    noetic_full
12
```

В папке `ROS/resources` необходимо добавлять два каталога с моделями и весами.

Узлы подключаются последовательно (важно, поскольку для трекинга 3d и определения 6dof позы необходим трекинг 2d (`bot_sort_node`) и сегментация. Для 6dof-позы также заранее требуется активация сервиса экстракции PointCloud объекта по `tracking_ID`):

```
source /opt/ros/noetic/setup.bash
```

```
source /sources/catkin_ws/devel/setup.bash
```

```
# Bag 1
```

```
export FIXED_FRAME=local_map_lidar
```

```
export TARGET_BAG=/resources/data/output.bag
```

```
export RVIZ_CONF=/resources/data/testbag_full.rviz
```

```
export DEPTH_TOPIC=/realsense_gripper/aligned_depth_to_color/image_raw
```

```
export IMAGE_TOPIC=/realsense_gripper/color/image_raw
```

```
echo "Run roscore"
```

```
roscore &
```

```
sleep 1
```

```
echo "Run bag play"
```

```
rosbag play $TARGET_BAG -i &
```

```
sleep 1
```

```
echo "Run object_point_cloud_extraction_node.py"
```

```
python /sources/catkin_ws/src/husky_tidy_bot_cv/scripts/object_point_cloud_extraction_node.py
```

```
-vis &
```

```
sleep 1
```

```
echo "Run object_pose_estimation_node.py"
```

```
python /sources/catkin_ws/src/husky_tidy_bot_cv/scripts/object_pose_estimation_node.py -vis &
```

```
sleep 1
```

```
echo "Run rviz"
```

```
roslaunch rviz rviz -d $RVIZ_CONF &
```

```
sleep 1
```

```
python /sources/catkin_ws/src/husky_tidy_bot_cv/scripts/text_query_generation_server.py
```

```
&
```

```
while :
```

```
do
```

```
    if rostopic list -v | grep -q "/segmentation_labels \[husky_tidy_bot_cv/Categories\] 1
```

```
publisher"; then
```

```
    echo "Found /segmentation_labels with 1 publisher"
```

```
    break
```

```
    else
```

```
    echo "Waiting for /segmentation_labels..."
```

```
    sleep 1
```

```
    fi
```

```
done
```

```
python /sources/catkin_ws/src/husky_tidy_bot_cv/scripts/opencv_node.py -vis &
while :
do
    if rostopic list -v | grep -q "/segmentation_vis \[sensor_msgs/Image\] 1 publisher"; then
    echo "Found /segmentation_vis with 1 publisher"
    break
    else
    echo "Waiting for /segmentation_vis..."
    sleep 1
    fi
done

rosbag play $TARGET_BAG -r 0.5 -s 5
# rosbag play $TARGET_BAG -r 0.5 -s 5 -u 5
exec /bin/bash
```

Трекинг и сегментация работают в режиме постоянной публикации, а определение 6dof-позы осуществляется через обращение к сервисам.

В rviz прописывать `fixed_frame` из тех названий, которые публикуются в `tf`, считать центром, относительно которого рассчитывать координаты - СК камеры.

Описание топиков RealSense

1. `/realsense_gripper/aligned_depth_to_color/image_raw`

- **Тип сообщения:** `sensor_msgs/Image`
- **Описание:** Этот топик содержит данные глубины, выровненные с цветным изображением. Он предоставляет информацию о расстоянии до объектов в сцене, совпадающей с цветным изображением, что упрощает сопоставление данных цвета и глубины.
- **Поля:**
 - **header:** Заголовок сообщения, включающий время публикации и фрейм, в котором оно было опубликовано.
 - **seq:** Номер последовательности сообщения.
 - **stamp:** Временная метка, когда данные были захвачены.
 - **frame_id:** Идентификатор системы координат, с которой связаны данные.

- **height**: Высота изображения в пикселях.
- **width**: Ширина изображения в пикселях.
- **encoding**: Формат кодирования пикселей, **8UC3** (8-битное, трехканальное изображение глубины)
- **is_bigendian**: Порядок байтов (0 — little-endian, 1 — big-endian).
- **step**: Количество байт в одной строке изображения.
- **data**: Данные изображения, представленные в виде массива байтов.

2. **/realsense_gripper/aligned_depth_to_color/camera_info**

- **Тип сообщения:** **sensor_msgs/CameraInfo**
- **Описание:** Этот топик содержит параметры калибровки камеры и другую информацию, связанную с камерой глубины. Эта информация важна для преобразования пиксельных координат в мировые координаты.
- **Поля:**
 - **header**: Заголовок сообщения.
 - **height**: Высота изображений, получаемых с камеры.
 - **width**: Ширина изображений, получаемых с камеры.
 - **distortion_model**: Модель дисторсии, например, **plumb_bob**.
 - **D**: Вектор коэффициентов дисторсии.
 - **K**: 3x3 матрица внутренней калибровки камеры.
 - **R**: 3x3 матрица вращения (обычно единичная для монокулярных камер).
 - **P**: 3x4 проекционная матрица.
 - **binning_x, binning_y**: Коэффициенты биннинга по горизонтали и вертикали.
 - **roi**: Регион интереса (Region of Interest), определяющий область изображения.

3. **/realsense_gripper/color/image_raw**

- **Тип сообщения:** **sensor_msgs/Image**
- **Описание:** Этот топик содержит необработанные данные цветного изображения с камеры. Он передает полные данные цветного изображения
- **Поля:**
 - **header**: Заголовок сообщения.
 - **height**: Высота изображения.
 - **width**: Ширина изображения.
 - **encoding**: Формат кодирования пикселей, например, **rgb8**.
 - **is_bigendian**: Порядок байтов.
 - **step**: Количество байт в одной строке изображения.
 - **data**: Данные изображения.

4. **/realsense_gripper/color/image_raw/compressed**

- **Тип сообщения:** **sensor_msgs/CompressedImage**

- **Описание:** Этот топик содержит сжатые данные цветного изображения, что позволяет снизить объем передаваемых данных. Сжатие может быть выполнено с использованием различных методов, таких как JPEG.
- **Поля:**
 - **header:** Заголовок сообщения.
 - **format:** Формат сжатия, например, **jpeg** или **png**.
 - **data:** Сжатые данные изображения в виде массива байтов.

Трансформации /tf

Для определения 6D-позы (позиция и ориентация в пространстве) и трекинга в ROS используются трансформации, публикуемые через систему топиков **/tf** и **/tf_static**. Эти топики содержат информацию о преобразованиях координатных систем (фреймов) относительно друг друга, что позволяет соотносить данные из разных источников (камеры и манипулятора).

- **Тип сообщения:** **tf2_msgs/TFMessage**
- **Поля:**
 - **transforms:** Список преобразований.
 - **header:** Заголовок, включающий временную метку и родительскую систему координат.
 - **child_frame_id:** Идентификатор дочерней системы координат.
 - **transform:** Преобразование, состоящее из:
 - **translation:** Вектор смещения (x, y, z).
 - **rotation:** Кватернион, описывающий поворот (x, y, z, w).