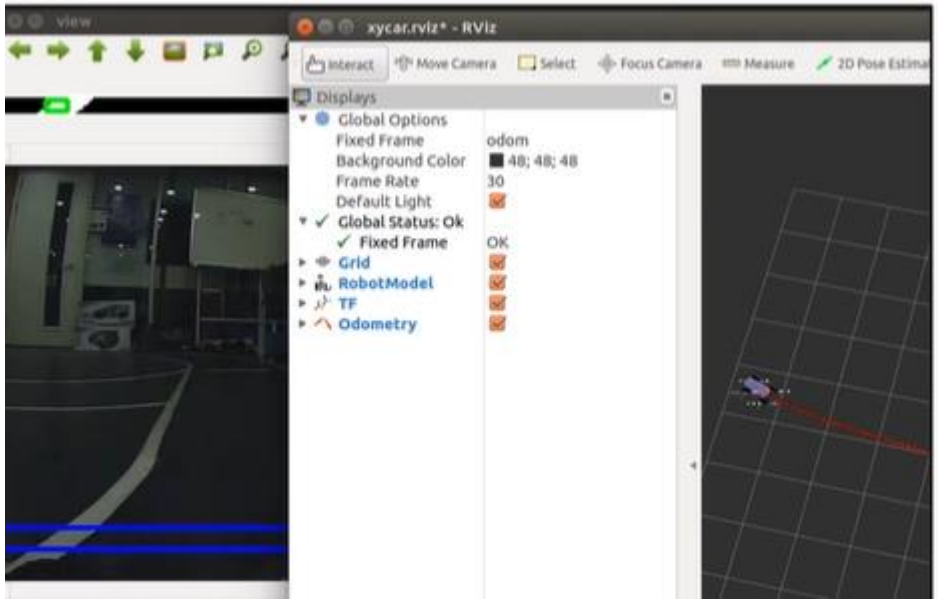
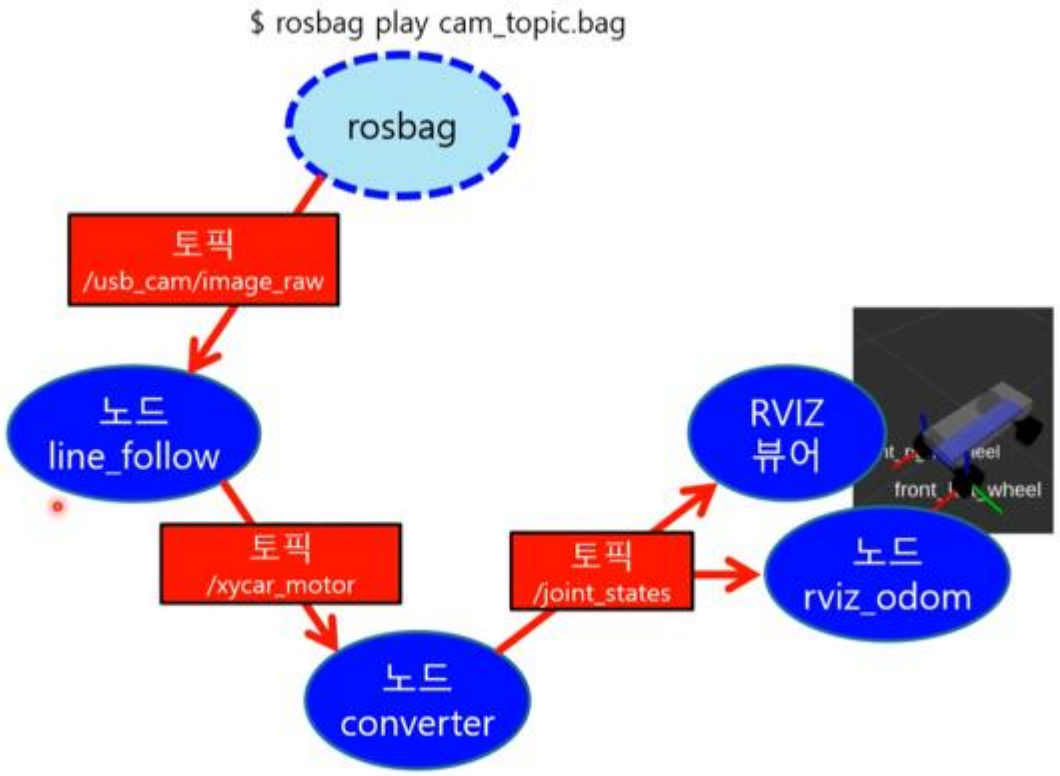
RVIZ차선 인식 주행

과제 설명

- OpenCV와 rosbag을 이용하여 차선인식 후 RVIZ에서 모형 차를 이용하여 차선을 따라 이동시키기



ROS 노드와 토픽 구성도



코드의 전체 프로그래밍 흐름도

/usb_cam/image_raw 토픽을 구독 -Subscribe

토픽 데이터를 OpenCV이미지 데이터로 변환

OpenCV영상 처리

ROI	관심영역 잘라 내기
GrayScale	흑백 이미지로 변환
Gaussian Blur	노이즈 제거
HSV - Binary	HSV 기반으로 이진화 처리

차선 위치 찾고 화면 중앙에서 어느 쪽으로 치우쳤는지 파악

핸들을 얼마나 꺾을지 결정(조향각 설정 각도 계산)

모터 제어를 위한 토픽 발행 - Publish

코드 설명 #1

```
#!/usr/bin/env python

import cv2, time
import numpy as np
import rospy, math, os, rospkg
from xycar_motor.msg import xycar_motor

from sensor_msgs.msg import Image
from cv_bridge import CvBridge

cap = cv2.VideoCapture('/home/soorim/xycar_ws/src/line_drive/src/track1.avi')
```

토픽을 발행하기 위한 메시지 타입

차선 추출을 위한 동영상 파일

코드 설명 #2

```
threshold_60 = 60

width_640 = 640
scan_width_200, scan_height_20 = 200, 20

lmid_200, rmid_440 = scan_width_200, width_640 - scan_width_200

area_width_20, area_height_10 = 20, 10

vertical_430 = 430


row_begin_5 = (scan_height_20 - area_height_10) // 2
row_end_15 = row_begin_5 + area_height_10
pixel_threshold_160 = 0.8 * area_width_20 * area_height_10
```

이미지 이진화에 이용할 명도 하한 값

차선 검출을 위한 ROI영역 크기

ROI영역에서 왼쪽과 오른쪽 검사가 끝나는 좌표(바깥에서 안쪽으로 검사)

흰 픽셀의 개수를 검사할 영역의 크기

ROI설정을 위한 세로 좌표

ROI 내에서 픽셀 검사 영역의 상대 좌표 시작과 끝

검사 영역을 차선으로 판단하는 흰 픽셀 비율의 하한 값

코드 설명 #3

```
motor_control = xycar_motor()

def pub_motor(angle, speed):
    global pub
    global motor_control

    motor_control.angle = angle
    motor_control.speed = speed

    pub.publish(motor_control)
```

각도와 속도 값에 대한 토픽을 발행하기 위한 함수

코드 설명 #4

```
def start():
    global pub

    rospy.init_node('line_follow')
    pub = rospy.Publisher('xycar_motor', xycar_motor, queue_size=1)
    rate = rospy.Rate(20)
```

Line_follow 이름으로 노드 선언

Xycar_motor로 토픽 발행

코드 설명 #5

```
while not rospy.is_shutdown():

    ret, frame = cap.read()
    if not ret:
        break

    if cv2.waitKey(1) & 0xFF == 27:
        break

    roi = frame[vertical_430:vertical_430 + scan_height_20, :]

    frame = cv2.rectangle(frame, (0, vertical_430), (width_640 - 1, vertical_430 + scan_height_20), (255,
0, 0), 3)
```

동영상에서 프레임들을 읽기

마지막에 도달하거나 ESC키가 눌릴 때 까지 반복

ROI 설정

설정 된 ROI의 둘레에 파란색 4각형 그리기

코드 설명 #6

```
hsv = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)

lbound = np.array([0, 0, threshold_60], dtype=np.uint8)
ubound = np.array([131, 255, 255], dtype=np.uint8)

bin = cv2.inRange(hsv, lbound, ubound)
view = cv2.cvtColor(bin, cv2.COLOR_GRAY2BGR)
```

Color를 HSV로 전환

밝기 지정

위에서 지정한 밝기에서 각 범위를 검은 점 혹은 흰 색 점으로 변환

Bin의 이미지를 color로 변환 후 차선 찾는 녹색 box를 그리기

코드 설명 #7

오른 쪽과 왼쪽 영역 모두 바깥쪽에서 안쪽으로 들어오면서 흰 픽셀 찾기

```
for l in range(area_width_20, lmid_200):
    area = bin[row_begin_5:row_end_15, l - area_width_20:l]
    if cv2.countNonZero(area) > pixel_threshold_160:
        left = l
        break

for r in range(width_640 - area_width_20, rmid_440, -1):
    area = bin[row_begin_5:row_end_15, r:r + area_width_20]
    if cv2.countNonZero(area) > pixel_threshold_160:
        right = r
        break
```

코드 설명 #8

왼쪽, 오른쪽 차선이 검출 되었으면 잘라낸 ROI이미지에 녹색 사각형 그림

```
if left != -1:
    lsquare = cv2.rectangle(view,
                            (left - area_width_20, row_begin_5),
                            (left, row_end_15),
                            (0, 255, 0), 3)
else:
    print("Lost left line")

if right != -1:
    rsquare = cv2.rectangle(view,
                            (right, row_begin_5),
                            (right + area_width_20, row_end_15),
                            (0, 255, 0), 3)
else:
    print("Lost right line")
```

코드 설명 #9

```
Speed = 10
if left !=-1 and right !=-1:
    Angle =(right+left)//2-320

    if Angle >-10 and Angle <10:
        Angle=0
    elif  Angle >10 :
        Angle//=3
    elif Angle <-10:
        Angle//=3
    pub_motor(Angle, Speed)
```

토픽으로 발행할 속도 지정

왼쪽, 오른쪽 차선이 검출되었는지 확인

중심 값을 기준으로 차선이 어느 쪽으로 치우쳐 있는지 계산

치우쳐 있는 정도가 좌우로 20 범위면 그냥 직진(Angle=0)

20범위가 넘어가면 왼쪽, 오른쪽에 따라 Angle을 현재 값의 절반 값으로 지정

토픽 발행

코드 설명 #10

```
            cv2.imshow("origin", frame)
            cv2.imshow("view", view)

            hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
            lbound = np.array([0, 0, threshold_60], dtype=np.uint8)
            ubound = np.array([131, 255, 255], dtype=np.uint8)
            hsv = cv2.inRange(hsv, lbound, ubound)
            cv2.imshow("hsv", hsv)

            time.sleep(0.1)

    cap.release()

    cv2.destroyAllWindows()


if __name__ == '__main__':

    start()
```

카메라를 이용하여 취득한 영상 표시와 파란 색 영역 표시

ROI를 잘라내어 이진화 한 영상 표시

카메라 영상을 이진화 한 영상 표시

While문이 종료되면 영상 화면 종료