

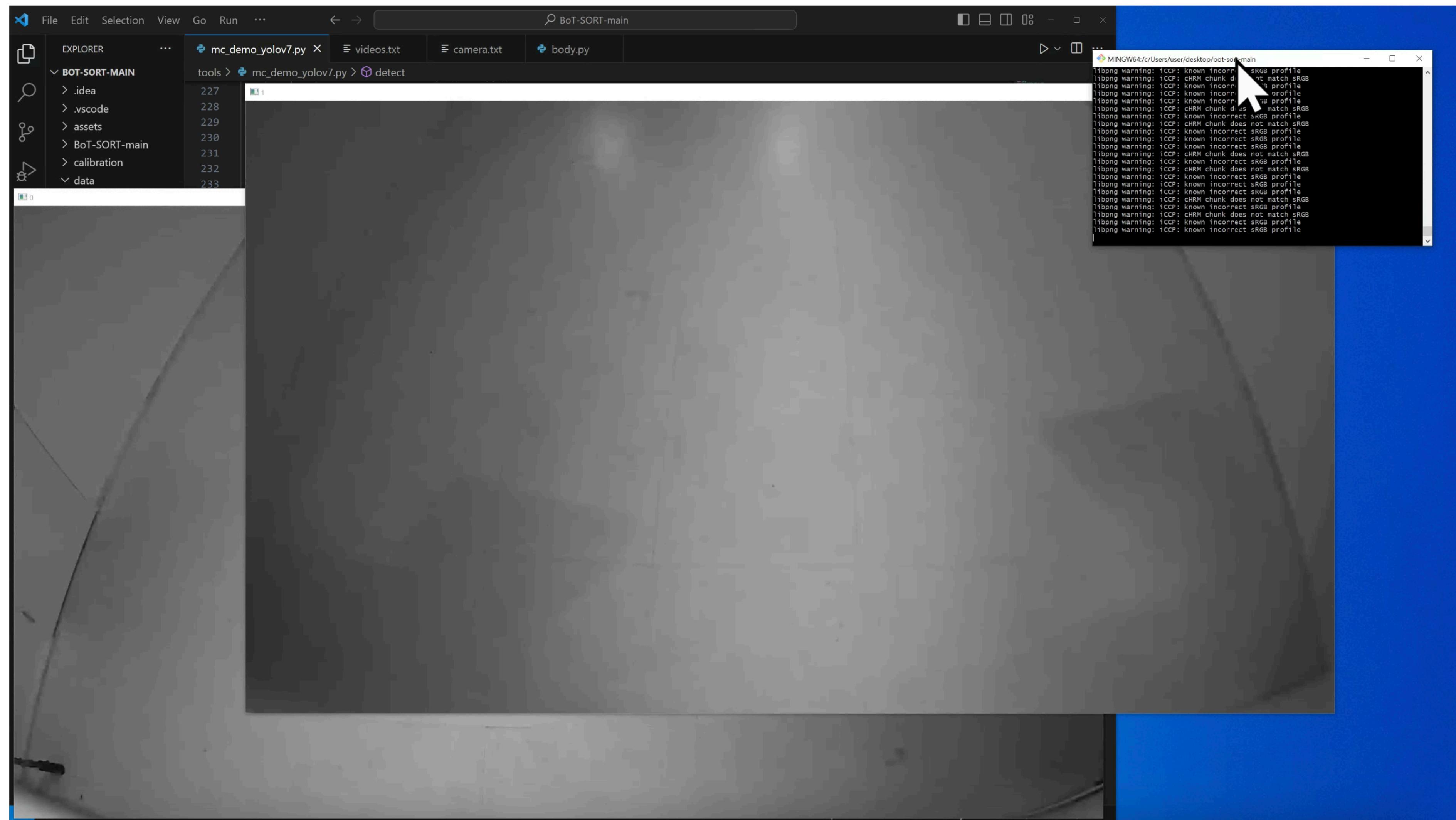
# Redirected Walking

231031 19101188 고은수

# Contents

- 영상에서만이 아닌 실시간 cam에서도 동작하도록
- openpose 적용
- 위치, 방향 제대로 전달

# 실시간 CAM에서 동작



# Openpose 적용

The screenshot shows a Jupyter Notebook interface with three main components:

- Output\_Keypoints:** A visualization of a person's body with 25 yellow circular keypoints numbered 0 through 24.
- output\_keypoints\_with\_lines:** A visualization of the same person's body with the keypoints connected by colored lines (green, red, blue) to form a skeleton.
- opencv – body.py:** A Python code editor containing the `body.py` script. The code uses OpenCV and DNN modules to process images and detect human poses.

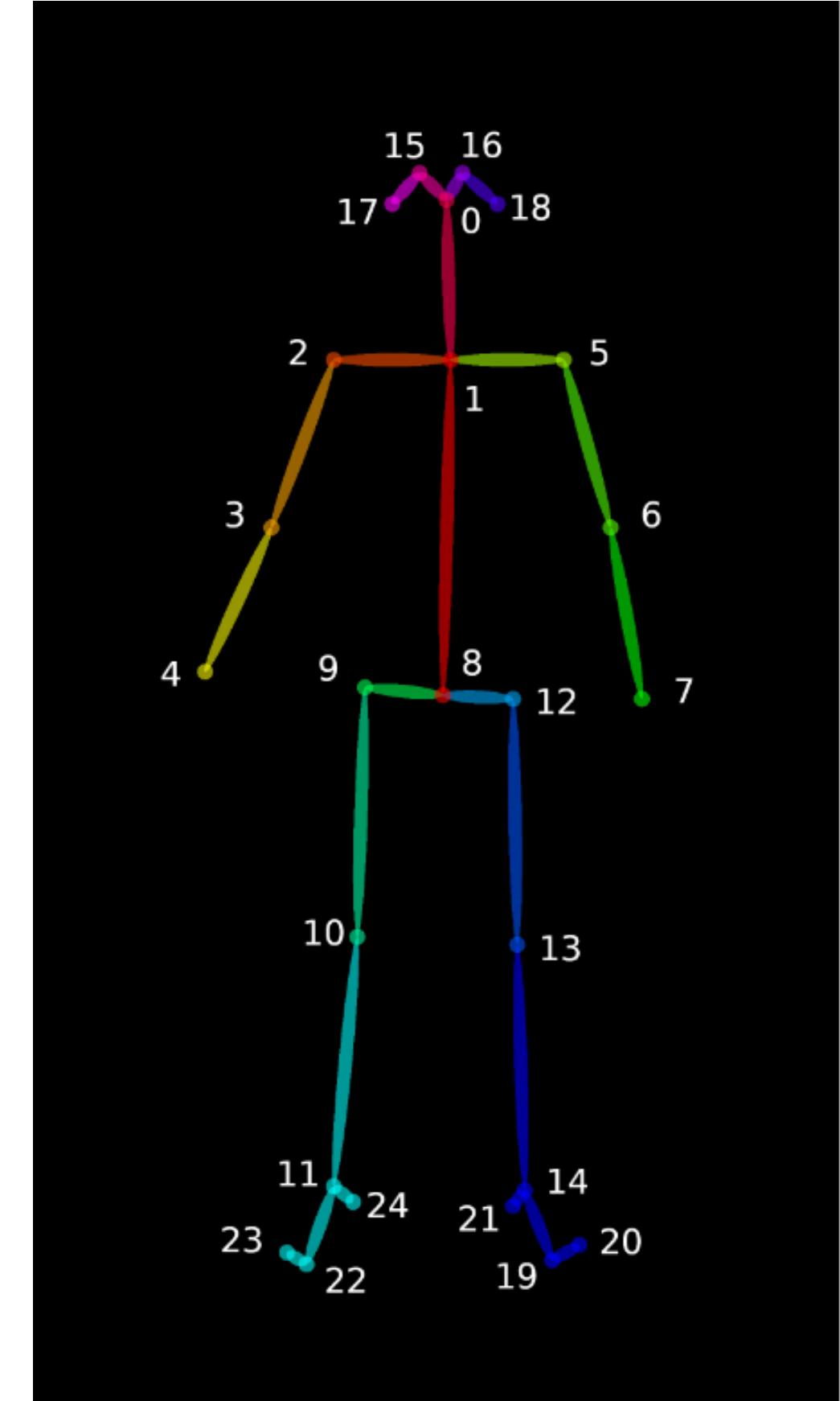
```
# 네트워크에 넣기 위한 전처리
input_blob = cv2.dnn.blobFromImage(frame, 1.0 / 255, (image_width, image_height))

# 전처리된 blob 네트워크에 입력
net.setInput(input_blob)

# 결과 받아오기
out = net.forward()

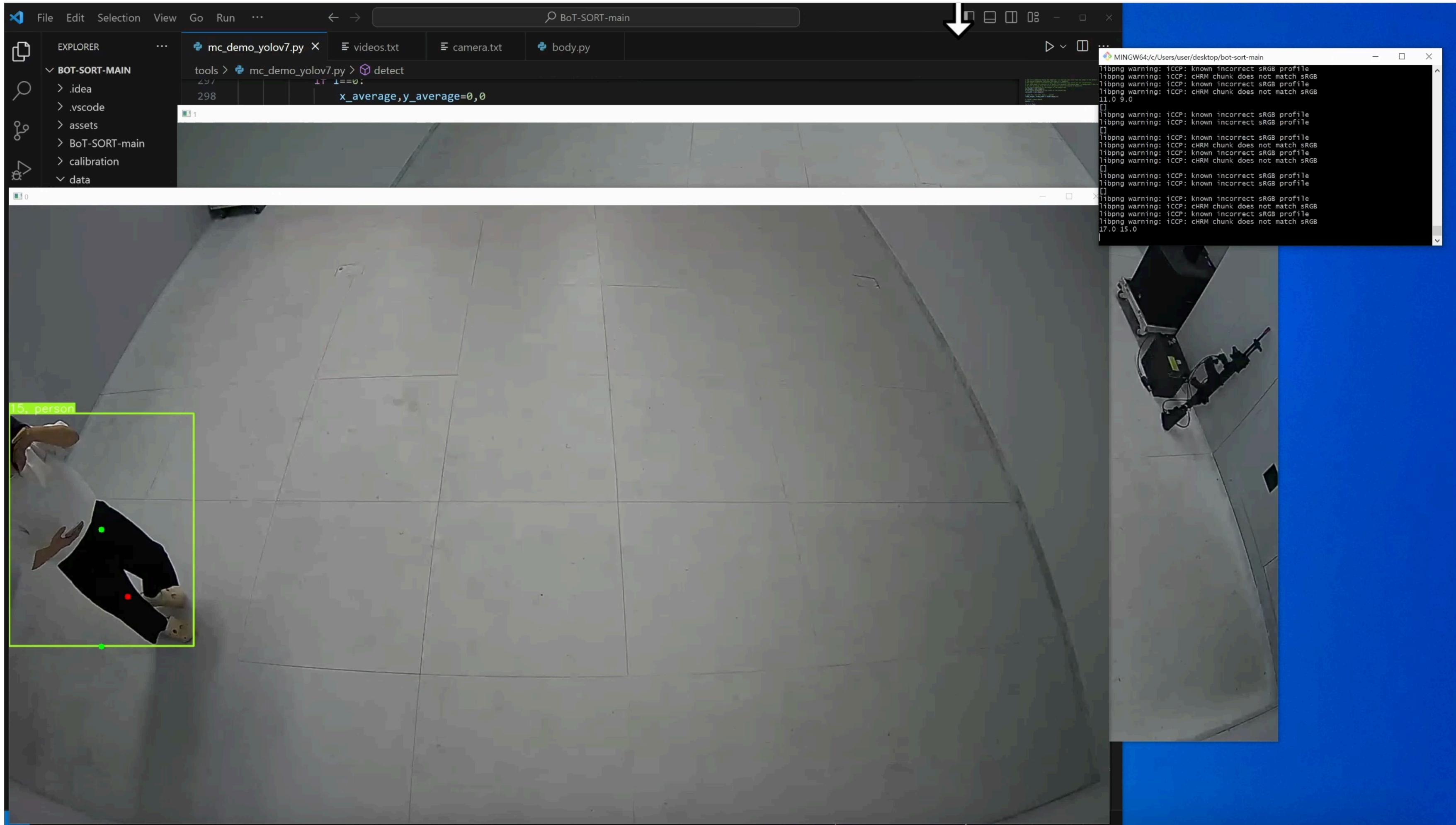
# The output is a 4D matrix :
# The first dimension being the image ID ( in case you pass more than one image )
# The second dimension indicates the index of a keypoint.
# The model produces Confidence Maps and Part Affinity maps which are all concatenated
# For coco model it consists of 57 points - 19 Keypoint_Confidence_Maps + 1 background
# map + 38 Part_Affinity_Maps + 1 background map

# output keypoints()
# [linked] 20 (39, 243) <=> 21 (47, 220)
# [linked] 11 (150, 228) <=> 24 (142, 228)
# [linked] 22 (165, 243) <=> 24 (142, 228)
# [linked] 23 (157, 243) <=> 24 (142, 228)
```



Body-25 model에서 발 좌표 이용

# Openpose 적용



너무 버벅거려서 사용하기 힘들어보임

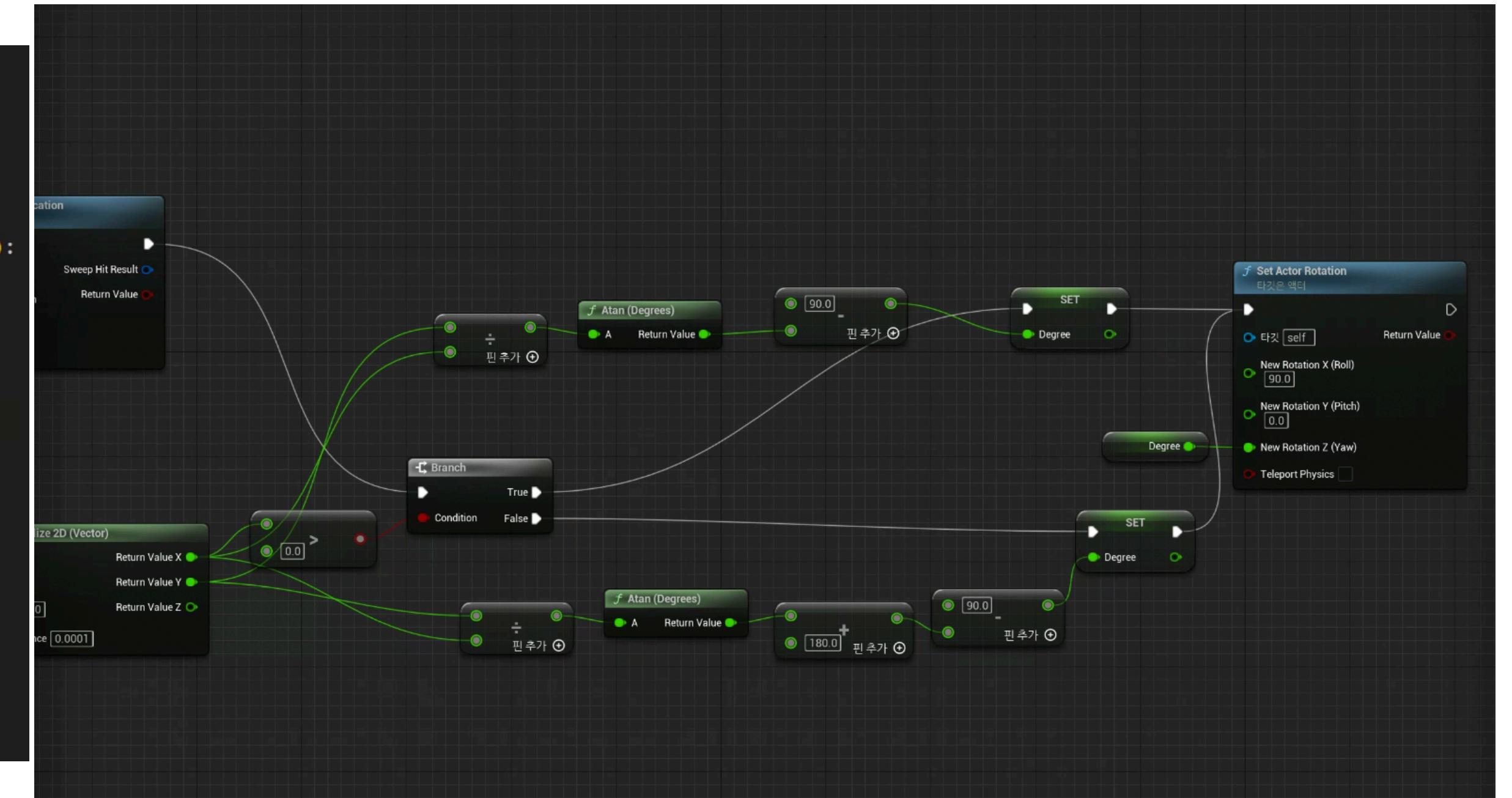
# 위치, 방향 제대로 전달

```
def Location(sid):
    global x_average,y_average,x_previous,y_previous,x_previous_vector,y_previous_vector
    while True:
        x_vector = x_average - x_previous
        y_vector = y_average - y_previous
        if x_average!=0 and y_average!=0:
            if not(x_vector==x_previous_vector and y_vector==y_previous_vector) and (x_vector!=0 and y_vector!=0):
                sio.emit('Location',f'{x_average},{y_average},{x_vector},{y_vector}',room=sid)
                x_previous = x_average
                y_previous = y_average
                x_previous_vector = x_vector
                y_previous_vector = y_vector

        eventlet.sleep(0.1)

@sio.event
def connect(sid, environ):
    logger.info('connect ' + sid)
    data_push_thread = eventlet.spawn(Location,sid)

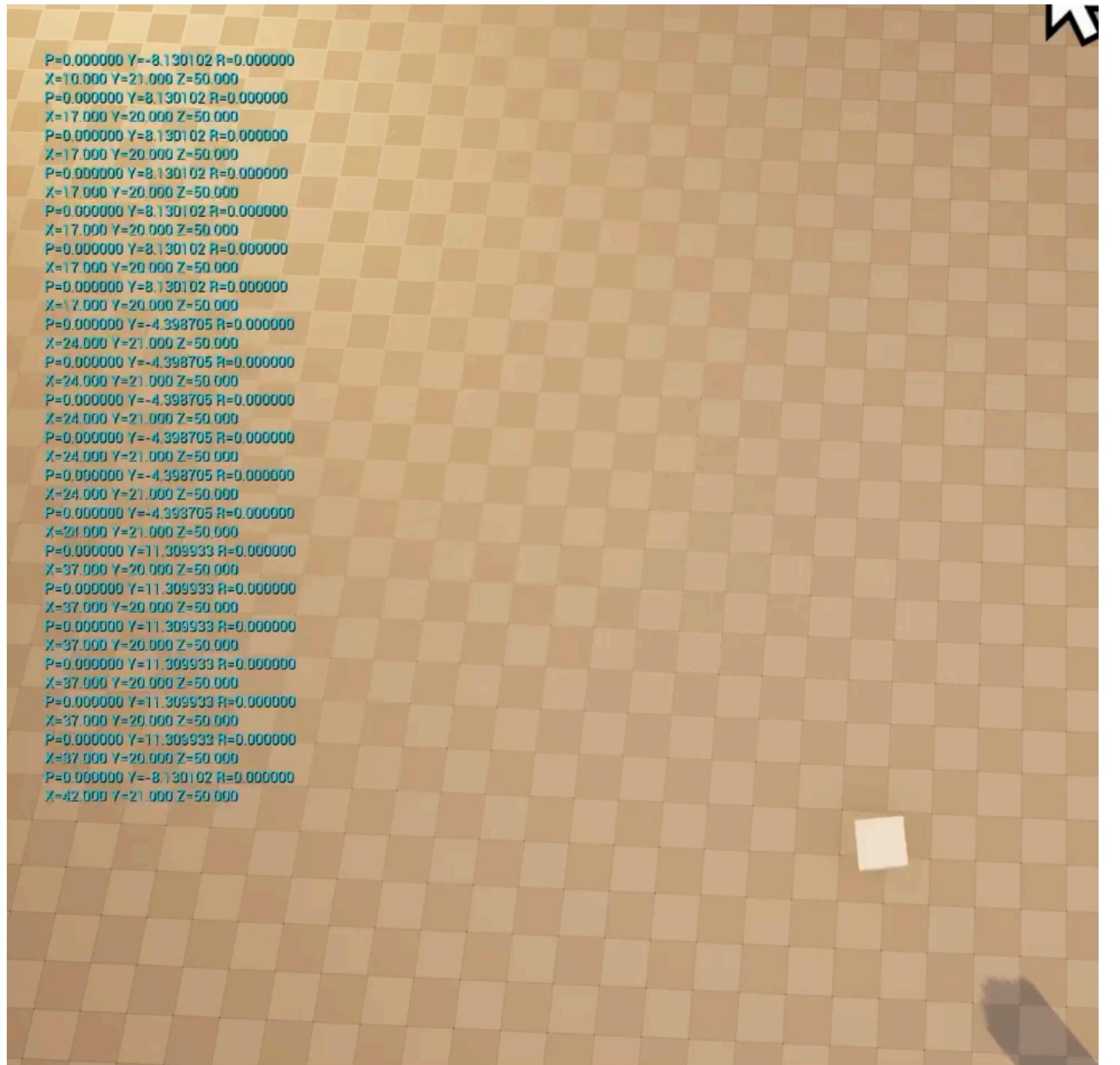
@sio.event
def disconnect(sid):
    print('disconnect ', sid)
```



로직 설정

방향벡터 좌표계에 맞게 변환

# 위치, 방향 제대로 전달



감사합니다