

UNITÀ

Incheon National University

2025.01.09

01월 09일 활동 일정

10:00 ~ 11:00 : 프로젝트 소개 및 강의. 가이드 라인 제시

1. 런치 파일 만들기
2. Rviz2 활용하기
3. TF
4. 카메라(내부 외부 파라미터란 무엇인가.)

11:00 ~ 12:00 : 경쟁형 퀴즈 : 개인전 (팀 우선 선택권을 위한)

12:00 ~ 13:00 : 점심시간

13:00 ~ 14:00 : 교육지도 인원 따라서 각각 스터디

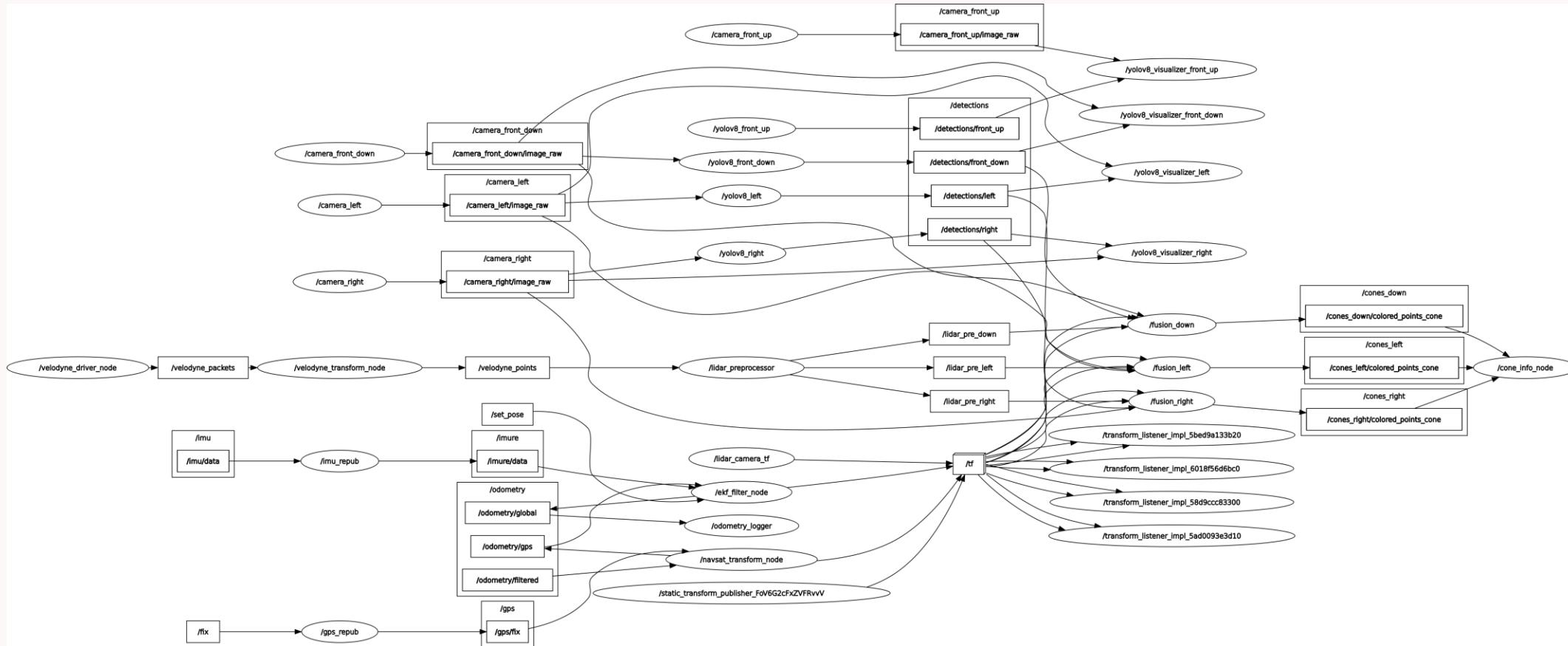
14:00 ~ 17:00 : 팀별 구현 미션 수행 (구현 실패 시, 주말 중에 구현해서 제출해야 함..!)

TEAM 1. 2D라이더 카메라 센서 퓨전

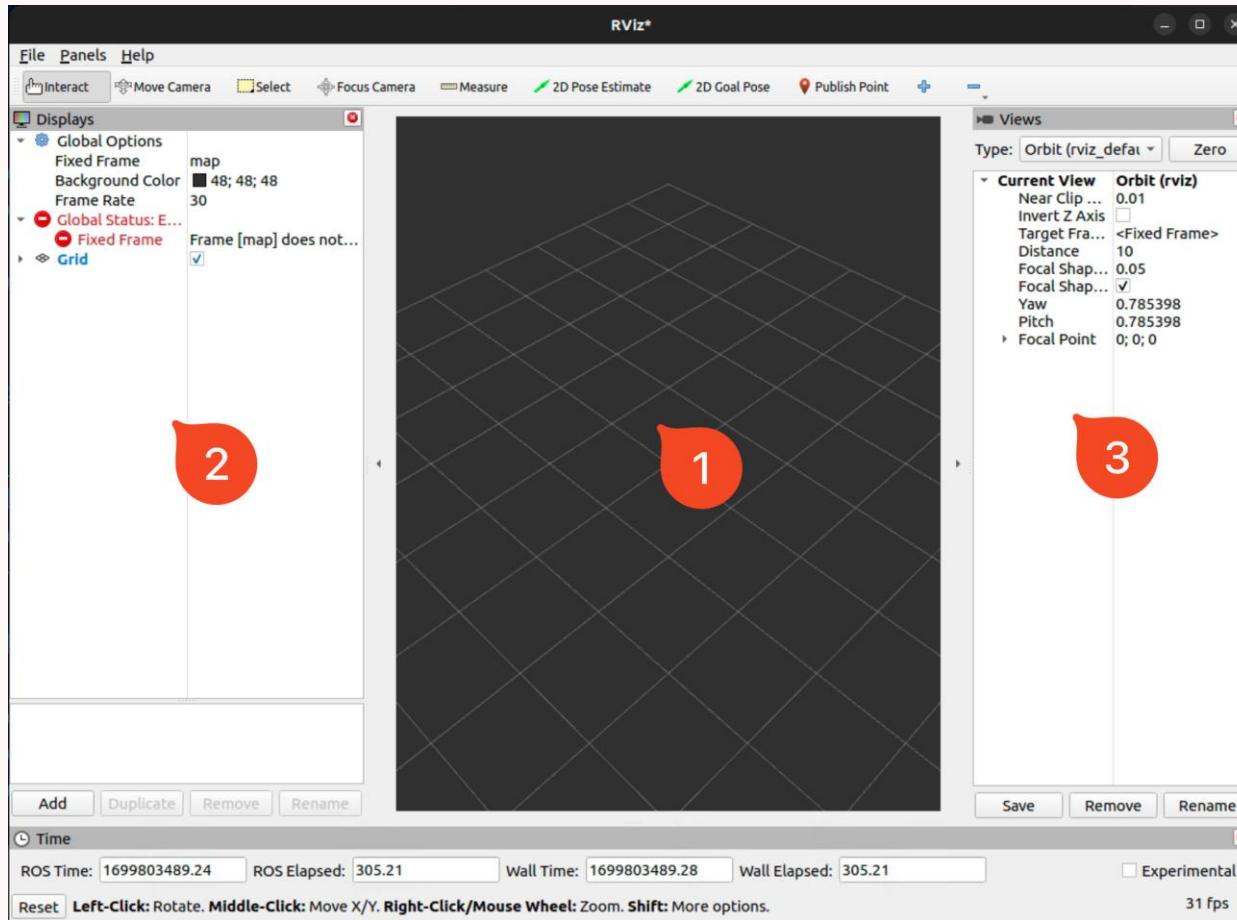
TEAM 2. 판단기술 선행 연구.

TEAM 3. urdf 만들기

Launch 파일을 사용하는 이유



Rviz2 활용하기



•(1) 중앙 패널

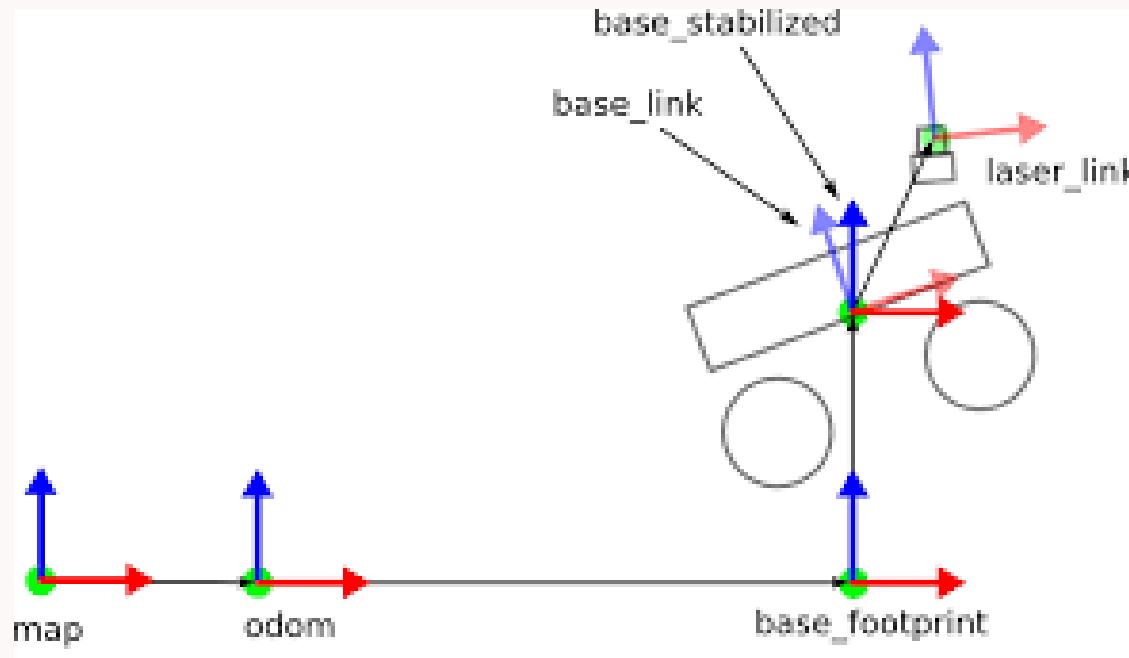
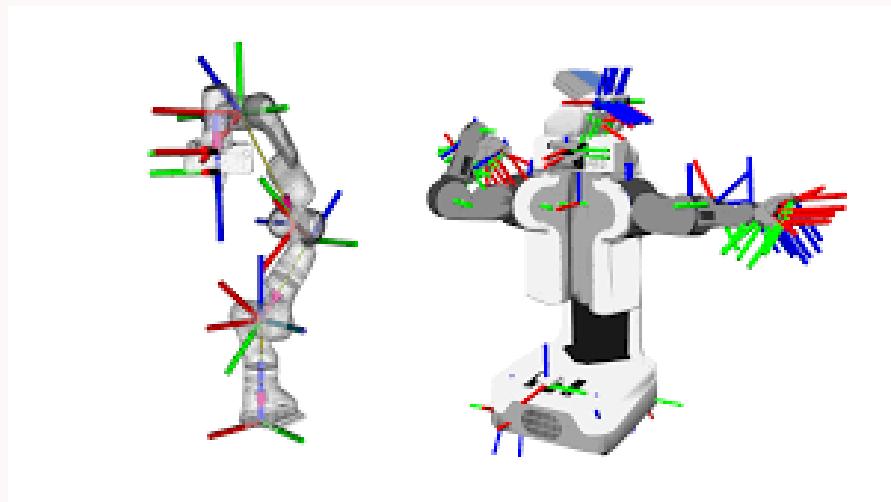
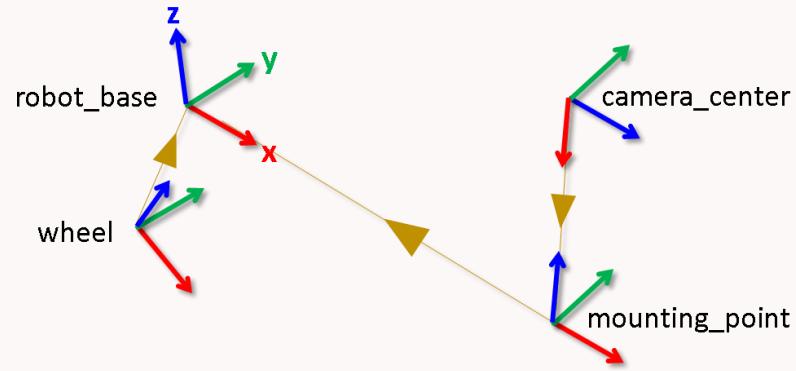
- 데이터가 표시되는 곳입니다.
- 회전(마우스 왼쪽 클릭 드래그),
- 이동(마우스 휠 드래그 또는 Shift + 왼쪽 클릭 드래그),
- 확대/축소(마우스 오른쪽 클릭 드래그)를 할 수 있는 3D 공간입니다.

•(2) 디스플레이 패널

- 중앙 패널에서 시각화하려는 모든 요소를 관리/구성할 수 있습니다.
 - Global Option에서 데이터 시각화 요구 사항에 가장 적합한 고정 프레임을 선택해야 합니다. 고정 프레임은 모든 데이터를 비교할 기준이 되는 프레임입니다.
 - Add 버튼을 클릭하면 RVIZ2에서 시각화할 수 있는 모든 유형의 요소가 표시됩니다.

•(3) 뷰 패널: 중앙 패널의 뷰를 전환할 수 있습니다.

TF



2025.01.09

TF의 발행과 구독

TF Broadcaster

좌표 변환을 “보내는 역할”

TF Broadcaster는 로봇의 각 링크나 센서 간 좌표 변환(TF)을 계산해서 퍼블리시하는 역할을 한다.

예를 들어 `base_link` → `camera_link`처럼 부모-자식 프레임 관계와 위치·자세 정보를 `/tf` 또는 `/tf_static` 토픽으로 계속 전달한다.

URDF, 로봇 상태 퍼블리셔, 센서 노드들이 대표적인 Broadcaster이다.



TF Listener

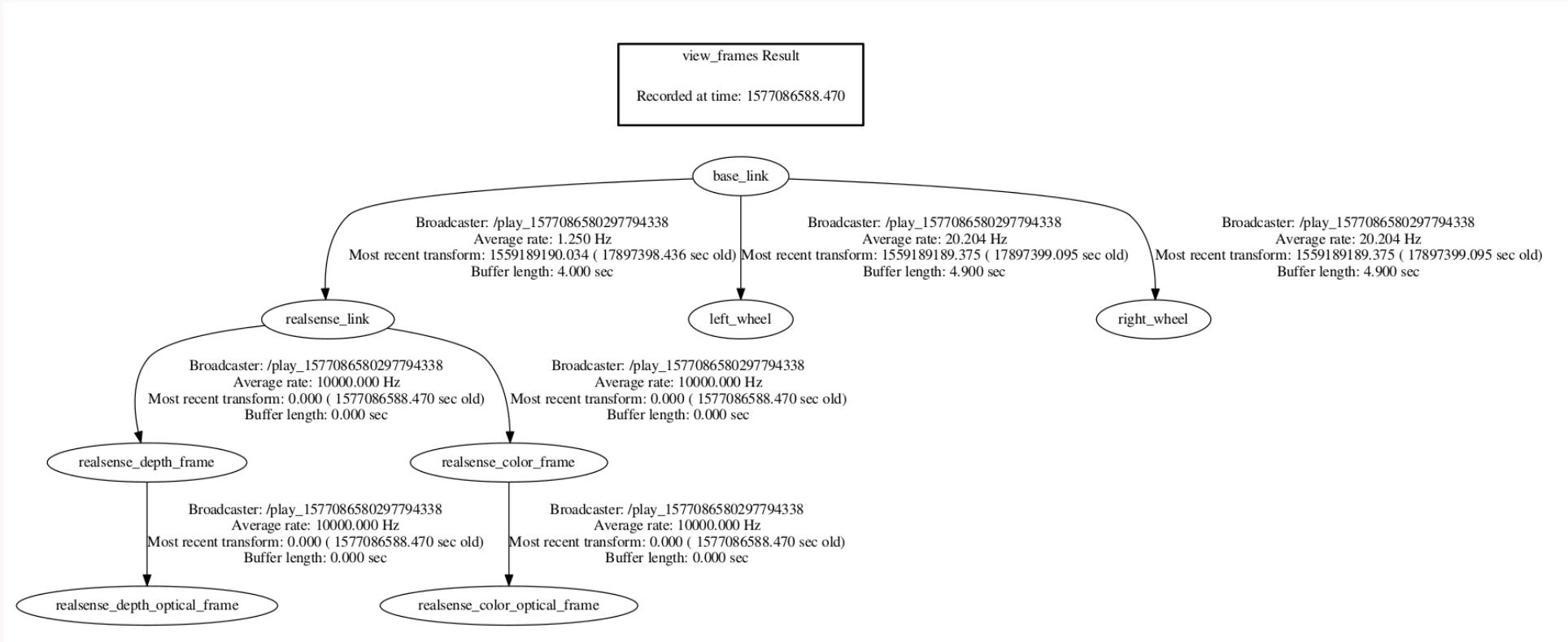
좌표 변환을 “받아서 사용하는 역할”

TF Listener는 Broadcaster가 퍼블리시한 TF 정보를 **구독(subscribe)** 하여,

“이 프레임에서 저 프레임으로 어떻게 변환되는지”를 요청해 사용하는 역할을 한다.

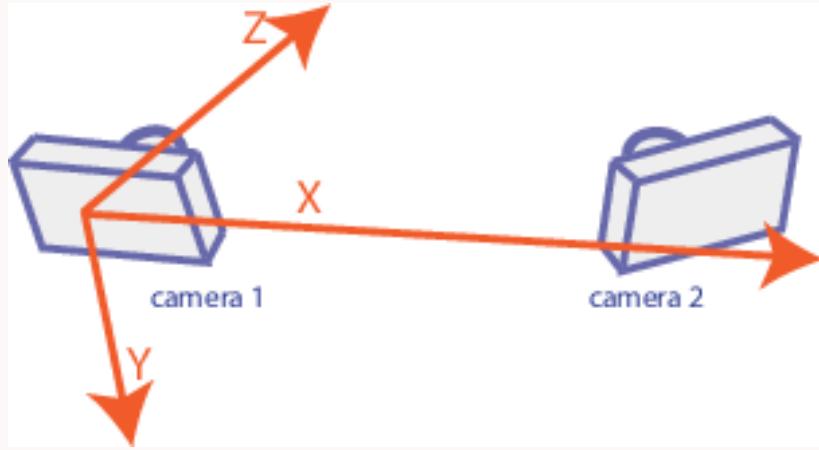
예를 들어, 인지 노드에서 카메라 좌표의 객체 위치를 `base_link` 기준으로 변환할 때 Listener가 사용된다.

TF tree



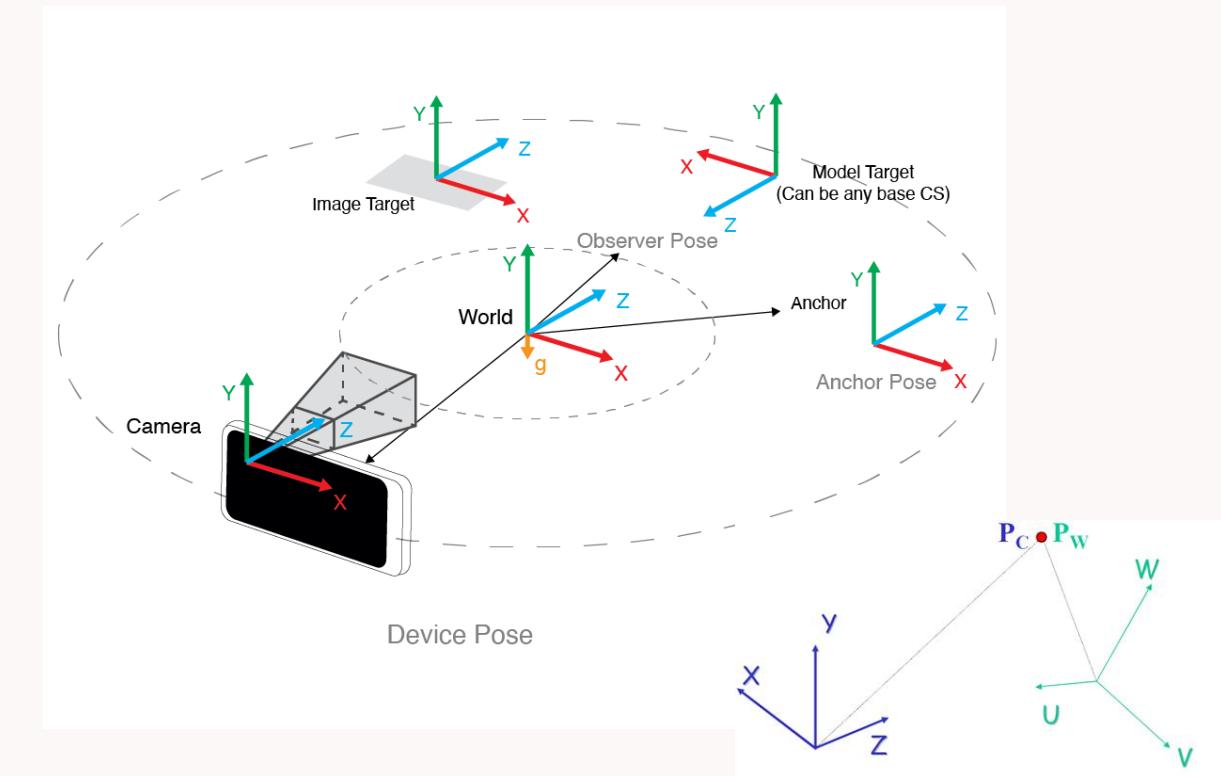
ros2 run tf2_tools view_frames <- tf tree를 보자.(pdf로 저장)

카메라 모델에서 사용되는 4가지 좌표계



1. Camera Coordinate System

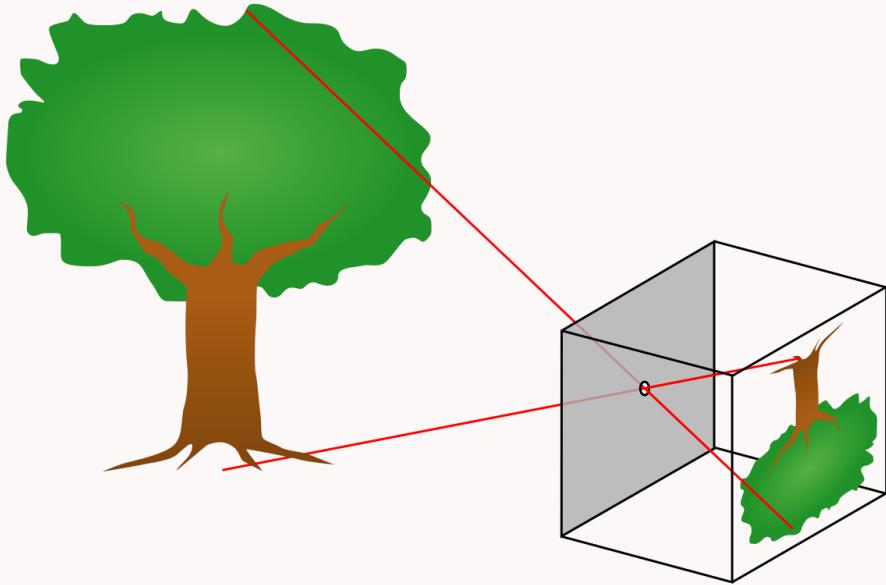
카메라 좌표계는 카메라의 광학 중심(optical center*)이 원점이 되는 3D 좌표계로, 카메라 기준에서 물체의 위치를 표현한다. X, Y, Z 축으로 구성되며, Z축은 카메라가 바라보는 방향(optic axis)을 의미한다. 이미지 평면(Image Plane)은 이 원점을 기준으로 Z축 방향으로 초점 거리(focal length)만큼 떨어진 위치에 놓인다.



2. World Coordinate System

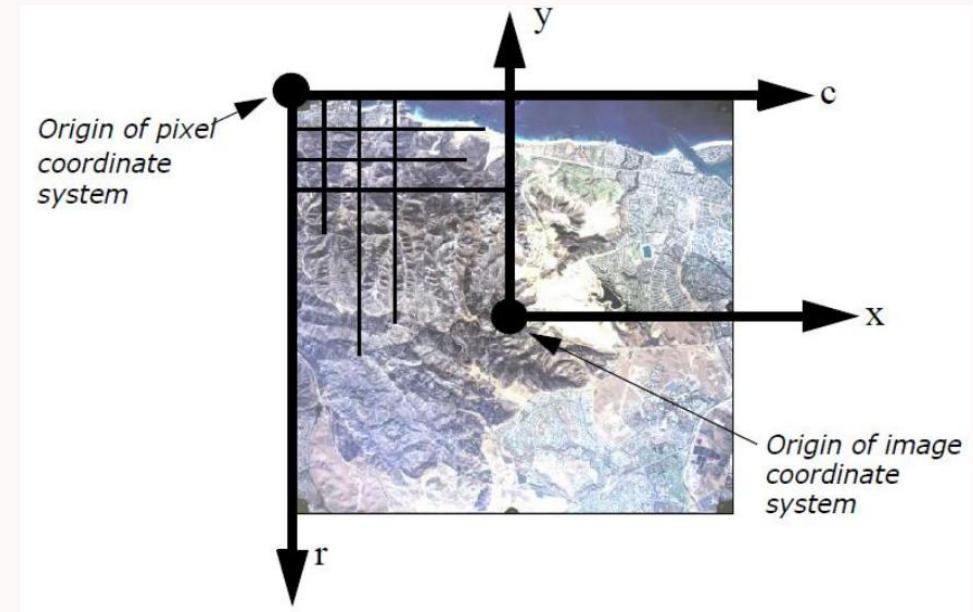
월드 좌표계는 여러 카메라와 객체의 공간적 관계를 표현하기 위해 정의된 기준 좌표계이다. 하나의 임의의 원점을 기준으로 U, V, W 축이 정의되며, 각 카메라의 위치와 자세는 이 좌표계를 기준으로 표현된다. 동일한 물체라도 월드 좌표계(P_w)와 카메라 좌표계(P_c)에서의 좌표값은 서로 다르게 나타난다.

카메라 모델에서 사용되는 4가지 좌표계



3. Image (Film) Coordinate System

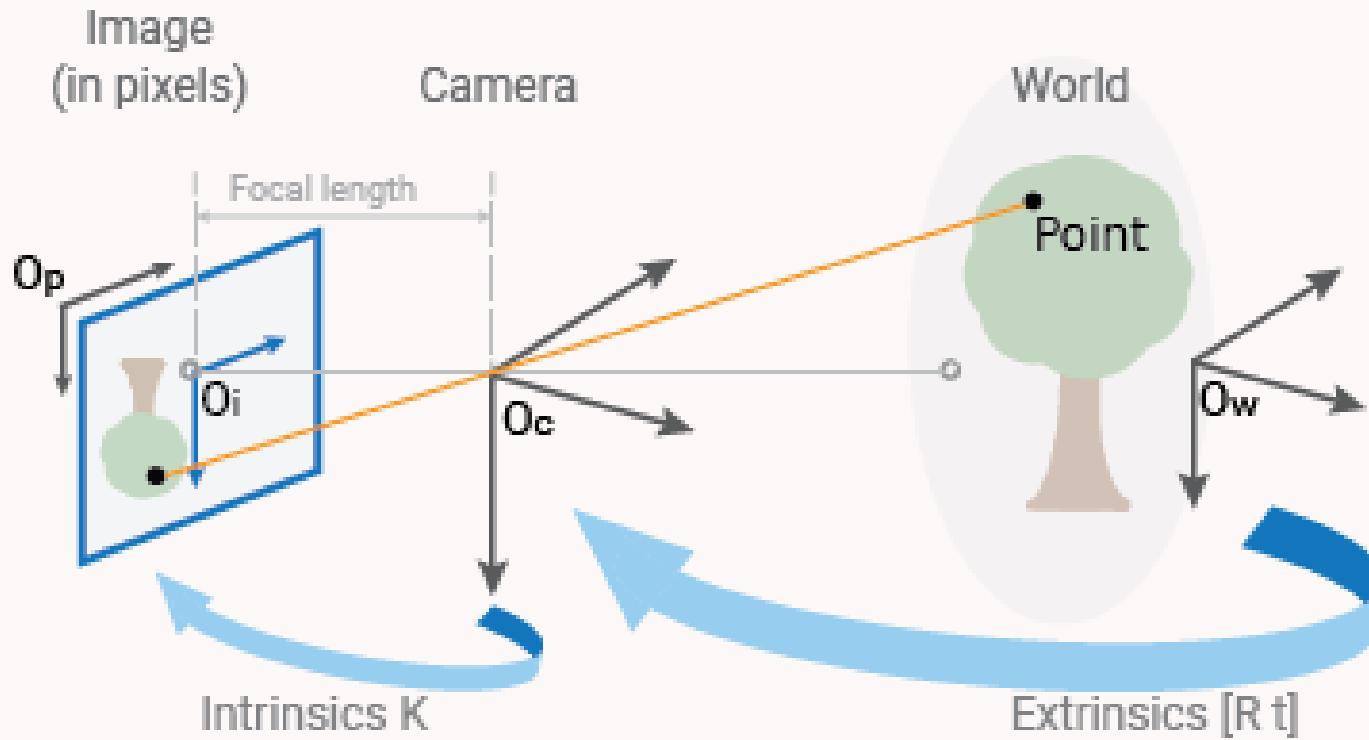
이미지 좌표계는 이미지 평면 위의 2D 좌표계로, 핀홀 모델에서 광축과 수직으로 만나는 지점인 Principal Point(주점)을 원점으로 사용한다. 이 좌표계는 카메라의 기하학적 투영 관계를 설명하는 데 사용되며, 렌즈 왜곡이나 내부 파라미터 분석의 기준이 된다.



4. Pixel Coordinate System

픽셀 좌표계는 실제 이미지가 저장·처리될 때 사용하는 2D 좌표계이다. 좌표의 원점 위치는 사용하는 그래픽/비전 라이브러리에 따라 다르며, **OpenCV는 좌상단**, **OpenGL은 좌하단**을 원점으로 사용한다. 이미지 좌표계를 픽셀 단위로 표현한 실용적인 좌표계라고 볼 수 있다.

Camera Calibration(카메라의 외, 내부 파라미터)



Intrinsic Parameter는 카메라 렌즈와 이미지 센서 구조에 의해 결정되는 내부 파라미터로, 3차원 공간의 점이 이미지 평면에 투영될 때 이미지가 얼마나 이동하는지(2D Translation), 얼마나 확대·축소되는지(2D Scaling), 그리고 얼마나 기울어져 표현되는지(2D Shear)를 정의한다

Extrinsic Parameter는 3D 공간에서 카메라가 실제로 어디에 위치해 있고(3D Translation), 어느 방향을 바라보고 있는지(3D Rotation)를 나타내는 파라미터로, 카메라 좌표계와 외부(월드 또는 로봇) 좌표계 사이의 공간적 관계를 설명한다.

Camera Calibration(카메라의 외, 내부 파라미터)

초점거리 (Focal Length: f_x , f_y)

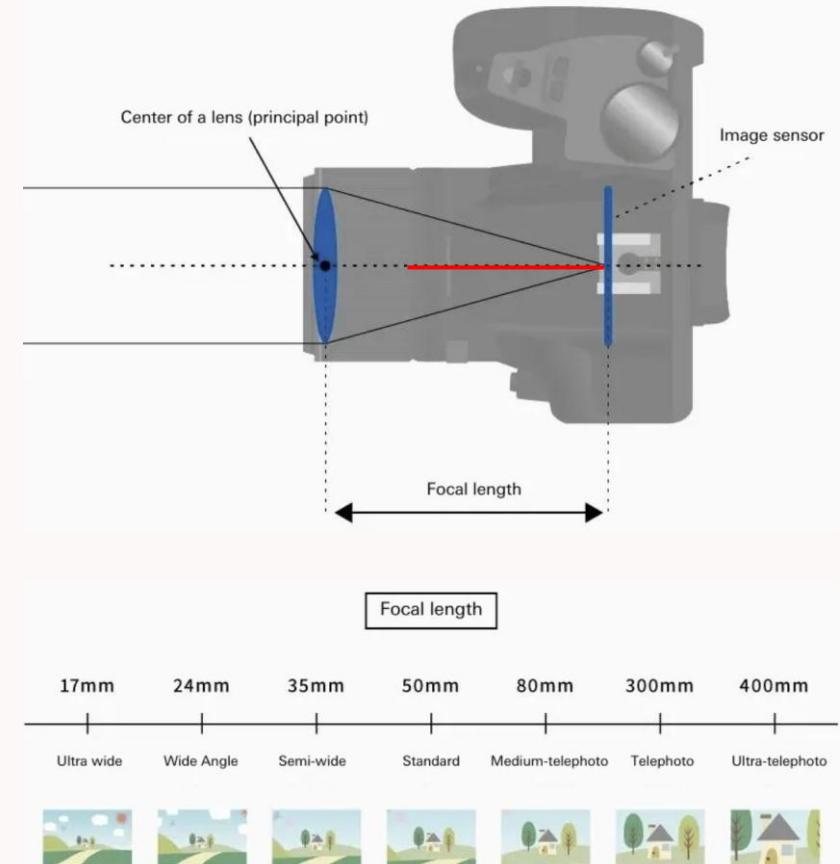
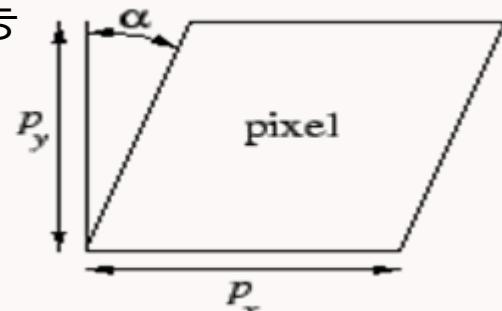
- 카메라가 얼마나 확대해서 보느냐를 결정
- f_x : 가로 방향 확대 비율
- f_y : 세로 방향 확대 비율
- 값이 클수록 더 줌(Zoom)된 이미지

주점 (Principal Point: c_x , c_y)

- 카메라 렌즈의 중심이 이미지에 맷히는 기준점
- 이상적으로는 이미지 정중앙
- 실제 카메라는 센서 오차로 중심에서 어긋나는 경우가 많음

비대칭 계수 (Skew Coefficient)

- 이미지의 x축과 y축이 완전히 직각이 아닐 때의 기울어짐
- 대부분의 현대 카메라에서는 거의 0
- 특수 센서가 아니라면 보통 무시 가능



Camera Calibration(카메라의 외, 내부 파라미터)

$$P = \overbrace{K}^{\text{Intrinsic Matrix}} \times \overbrace{[R \mid t]}^{\text{Extrinsic Matrix}}$$
$$= \underbrace{\begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Translation}} \times \underbrace{\begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Scaling}} \times \underbrace{\begin{pmatrix} 1 & s/f_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Shear}} \times \underbrace{\left(\begin{array}{c|c} I & t \end{array} \right)}_{\text{3D Translation}} \times \underbrace{\begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix}}_{\text{3D Rotation}}$$

경쟁형 퀴즈

이석빈, 박근호, 성현영 제외(펌웨어 마무리 할 것)

1. 진행 개요

본 퀴즈는 상대 팀의 코드를 분석·이해하는 능력 + 본인 프로젝트의 이해도를 평가하기 위한 경쟁형 퀴즈이다.
모든 문제는 상대 팀이 작성한 코드를 기반으로 출제한다.

2. 시간 구성 (20분 내)

상대팀 코드 분석 및 출제시간 (10분)

-상대 팀의 코드를 읽고 구조, 로직 등을 분석한다.

본인 프로젝트 문제 대응 시간(10분)

-분석한 코드를 기반으로 문제를 출제하고,
-출제자는 정답(답안)을 사전에 반드시 작성해야 한다.

1팀	2팀	3팀	4팀	5팀
장동혁	이기현	한주형	정가용	강민수
김민서	이원종	윤지윤	정규민	윤제호

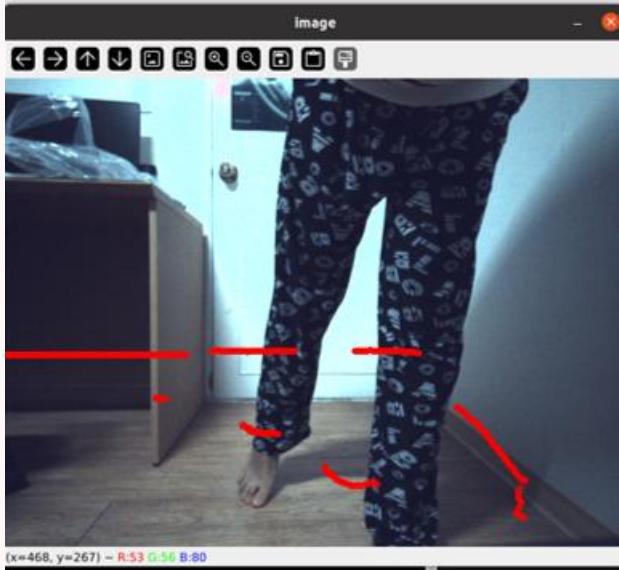
문제를 맞힌 팀 -> 문제당 1점

출제한 문제를 상대 팀 중 아무도 풀지 못한 경우 -> 출제 팀에게 2점

부분 점수 없음 (정답 / 오답만 판정)

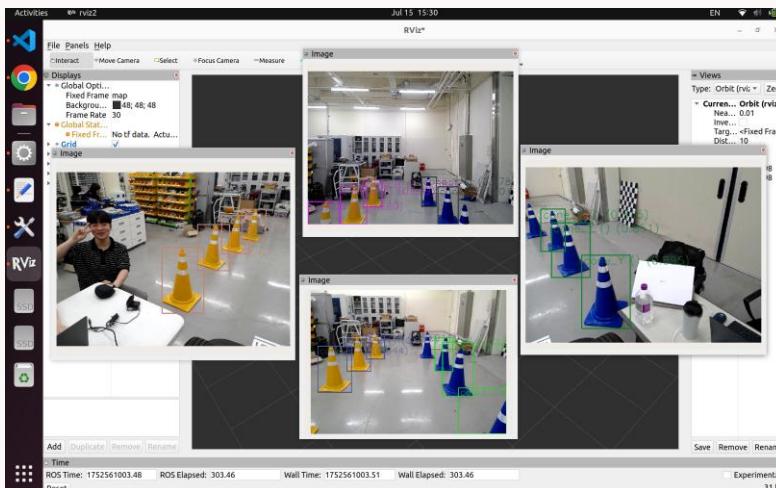
2025.01.02

2D라이다 카메라 센서 퓨전



Camera와 Lidar의 각 센서의 장점을 결합하여, 단점을 보완한 Sensor fusion을 수행할 것

1. 좌측 이미지와 같이, 이미지 위에 레이저 스캔 데이터가 투영되도록 해야함.
2. 학습 된 물체의 거리를 측정 가능하도록 해야함.
ex) 사람이 카메라에서 인식되고, 사람과 카메라 간의 거리를 구한다.,



참고는 가능, 하지만 모든 sw를 자체 개발할 것.

<https://velog.io/@jinhoyoho/%EC%9E%90%EC%9C%A8%EC%A3%BC%ED%96%89LiDAR-Camera-Calibration%EC%8B%A4%EC%8A%B5%ED%8E%8B>

https://github.com/jinhoyoho/rplidar_camera_calibration

2025.01.02

판단기술 선행 연구.

1. 트랙 주행을 위한 경로 생성,

2. 주차

3. 장애물 회피

4. 장애물 감지 시 정지

각 미션에 대한 기술 및 알고리즘을 조사함.

금일 내 발표 가능한 자료를 만들어서, 16시 30분이 되면 발표 시작해야 함.

단 우리가 활용 가능한 센서 기술 내에서 가능해야 함. (발표시간 15분 내.)

urdf 만들기

1. 센서 TF 반영

- 차량 바디 기준으로
- 카메라, 라이다, 초음파 센서의 TF를 반영한 URDF 작성

2. 차량 Mesh 적용

- H-Mobility class 코드의 차량 mesh 파일을 URDF에 적용
- RViz2에서 차량 외형이 실제 mesh로 시각화될 것

3. RViz2 시각화

- 단일 URDF 파일만으로 TF가 브로드캐스트되도록 구성
- RViz2에서 차량과 센서 프레임이 정상적으로 보일 것

최종 결과

차량 + 센서 TF가 포함된 **하나의 URDF 파일**
RViz2에서 차량 모델과 TF 트리가 정상 확인