

캡스톤디자인(2) 진행 상황

Motus+er

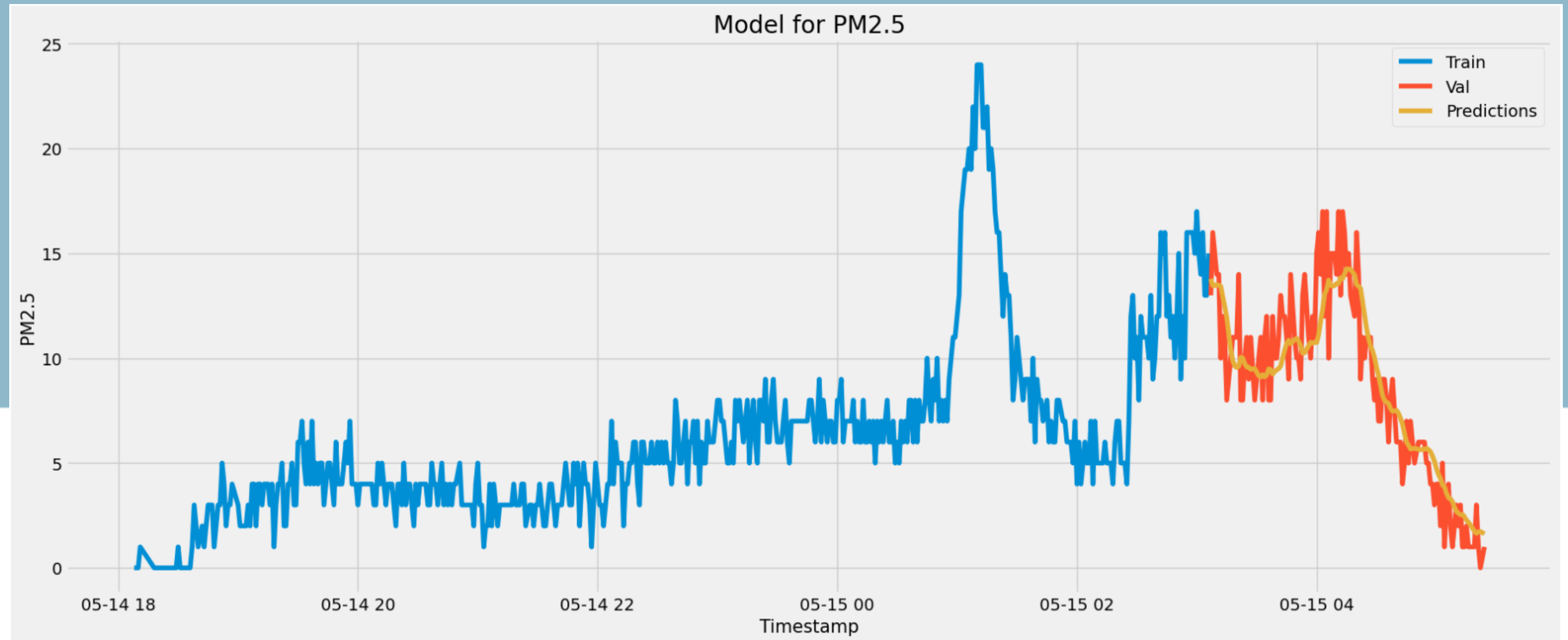
김민규 | 장하늘 | 임혜연

LSTM 모델 학습

교수님의 피드백을 반영하여 미세먼지만을 학습하여 모델을 완성

PM2.5 Predictions		
Timestamp		
2024-05-15 03:07:00	13.0	13.790129
2024-05-15 03:08:00	16.0	13.468938
2024-05-15 03:09:00	15.0	13.459887
2024-05-15 03:10:00	14.0	13.512075
2024-05-15 03:11:00	14.0	13.490475
...
2024-05-15 05:19:00	1.0	1.759684
2024-05-15 05:20:00	3.0	1.642843
2024-05-15 05:21:00	1.0	1.721553
2024-05-15 05:22:00	0.0	1.734786
2024-05-15 05:24:00	1.0	1.620316

실제 값과 예측치



실제값과 예측값 그래프

- 초기 우선순위는 방 이름의 오름차순으로 공기청정시작
- 케어버디가 방을 돌아다니면서 각 방별 공기질 데이터를 수집
이후 공기질 예측 모델을 활용하여 우선순위 선정

자동 매핑 - Active SLAM

Active SLAM의 목적

01

환경을 더 빠르고 효율적으로 매핑

로봇이 자신의 위치와 지도의 불확실성을
최소화하도록 움직일 수 있다.

02

로봇이 자율적으로 탐색하고 작업을 수행

따로 조작을 하지 않아도 센서를 통해 얻
은 정보를 통해
로봇이 스스로 경로를 계획하고 장애물을
피하며 목표 지점을 설정할 수 있다.

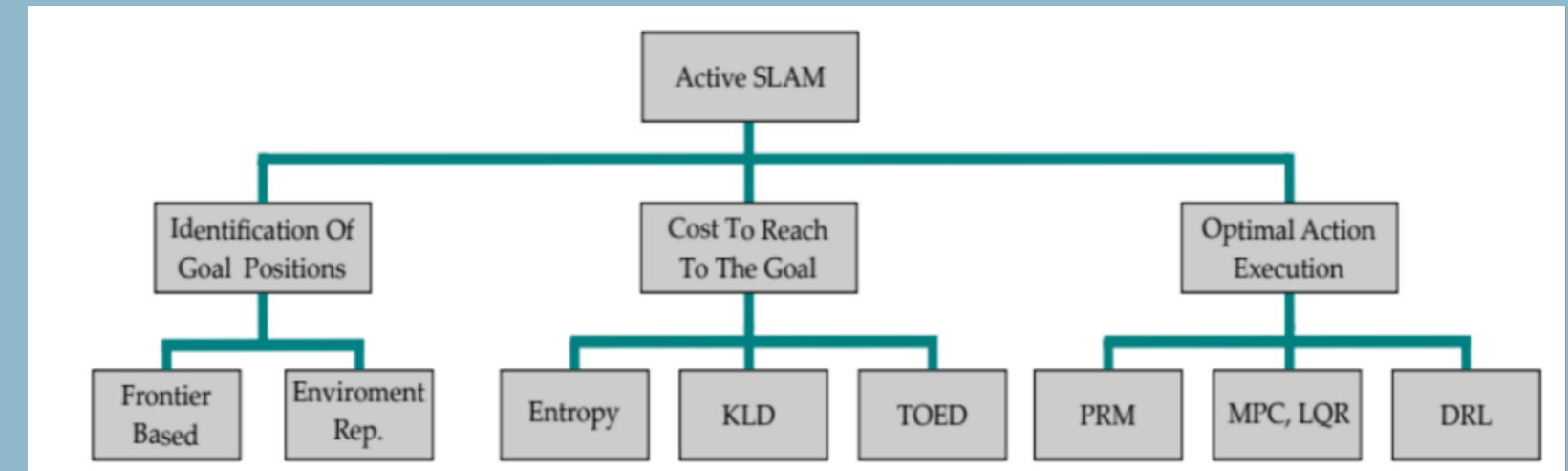
03

로봇의 위치 추정 및 환경 지도 작성의 불
확실성 개선

로봇이 불확실성이 높은 영역을 우선적으
로 탐색하여 데이터를 수집함으로써
위치 추정 정확도를 높이고 더 정확한 지
도를 생성할 수 있다.

자동 매핑 - Active SLAM

Active SLAM의 과정



01

로봇이 잠재적으로 탐색할 수 있는 여러 global waypoint(목표 지점) 후보들을 정한다.

가장 대표적인 알고리즘은 Frontier 기반 알고리즘이다.

02

Frontier까지 가는 cost를 계산해 global waypoint(목표 지점)를 정한다.

cost 기반으로 갈 곳을 정했다면, 그 다음으로는 언제까지, 어디까지 갈 것인지(목표를 그대로 유지하는 기간)에 대한 문제가 생긴다.

03

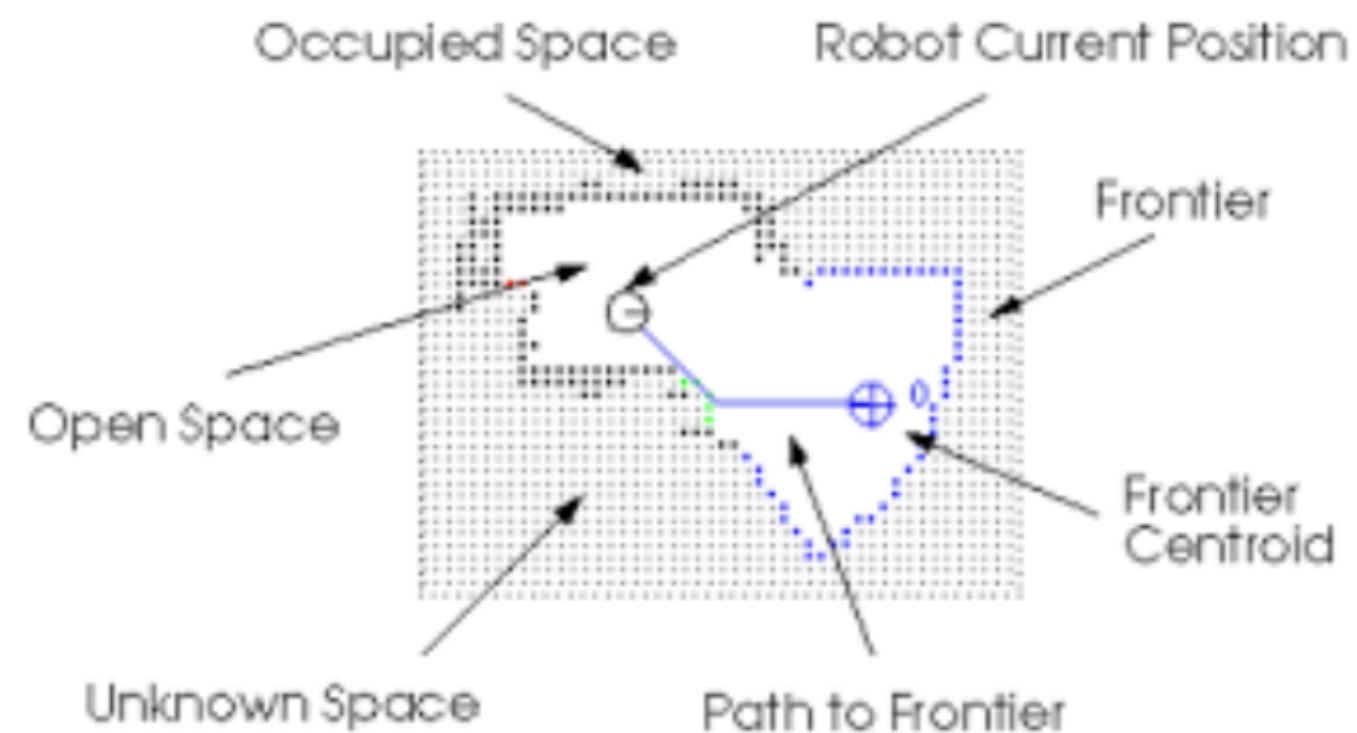
얼마 만큼 실제 이동할 (Action을 취할) 것인가를 정한다.

이 과정에서는 실질적인 Action을(local waypoint)정한다. localway point까지는 로봇이 100% 이동해야 한다.

정리하자면, global waypoint를 정한 후 local waypoint까지 이동한다.
이후에 다시 global waypoint를 정한 후 local wapoint까지 이동한다. 이 과정을 반복해서 사용자가 정한 닫힌 구간을 가장 빠르게 탐색한다.

자동 매핑 - Active SLAM

Frontier Based exploration



Frontier란 로봇이 가보지 않은 known 영역과 unknown 영역의 경계를 뜻한다.

즉, unknown과 known의 경계인 파란색 pixel이 Frontier이다. 이와 같이 복수개의 Frontier가 node로 저장되어 global waypoint를 정하는 데에 이용된다.

자동 매핑 - Active SLAM

'explore_lite' ros 패키지 사용



ROS.org

About | Support | Discussion Forum | Index | Service Status | ros @ R

Documentation Browse Software News

explore_lite

melodic noetic Show EOL distros: ☐

See explore_lite on index.ros.org for more info including anything ROS 2 related.

Documentation Status

Package Summary

✓ Released ✓ Continuous Integration ✓ Documented

Lightweight frontier-based exploration.

- Maintainer status: developed
- Maintainer: Jiri Horner <laeqten AT gmail DOT com>
- Author: Jiri Horner <laeqten AT gmail DOT com>
- License: BSD
- Source: git <https://github.com/hrnr/m-explore.git> (branch: noetic-devel)

Use GitHub to [report bugs](#) or [submit feature requests](#). [[View active issues](#)]

Package Links

- [Code API](#)
- [FAQ](#)
- [Changelog](#)
- [Change List](#)
- [Reviews](#)

[Dependencies](#) (12)

[Jenkins jobs](#) (10)

이 패키지는 탐욕적 프론티어 기반 탐사(greedy frontier-based exploration)를 제공한다.

노드가 실행 중일 때, 로봇은 더 이상 프론티어를 찾을 수 없을 때까지 환경을 탐욕적으로 탐사한다.

이동 명령은 move_base/goal 토픽으로 publish한다.

<탐사 과정>

- 경계 탐색 및 비용 계산: 'search_.searchFrom(pose.position)' 함수를 통해 경계를 탐색하고, 비용을 계산한다. 비용이 낮은 순서대로 경계를 정렬한다.
- 블랙리스트 필터링 : 로봇이 목표 지점에 도달하지 못하거나 진전이 없는 경우 블랙리스트에 넣는다. 'std::find_if_not' 함수를 사용하여 이 블랙리스트에 포함되지 않은 경계를 찾는다. 경계의 중심점이 블랙리스트에 포함된 경우 해당 경계를 건너뛴다.
- 목표 지점 설정 : 블랙리스트에 포함되지 않은 첫 번째 경계를 선택하고, 그 경계의 중심점을 목표 지점으로 설정한다.

위의 과정이 1초에 한번씩 반복된다.

자동 매핑 - Active SLAM

'explore_lite' ros 패키지 사용

4.1 explore

Provides exploration services offered by this package. Exploration will start immediately after node initialization.

4.1.1 Actions Called

move_base ([move_base_msgs/MoveBaseAction](#))

[move_base](#) actionlib API for posting goals. See [move_base#Action API](#) for details. This expects [move_base](#) node in the same namespace as [explore_lite](#), you may want to remap this node if this is not true.

4.1.2 Subscribed Topics

costmap ([nav_msgs/OccupancyGrid](#))

Map which will be used for exploration planning. Can be either costmap from [move_base](#) or map created by SLAM (see above). Occupancy grid must have got properly marked unknown space, mapping algorithms usually track unknown space by default. If you want to use costmap provided by [move_base](#) you need to enable unknown space tracking by setting `track_unknown_space: true`.

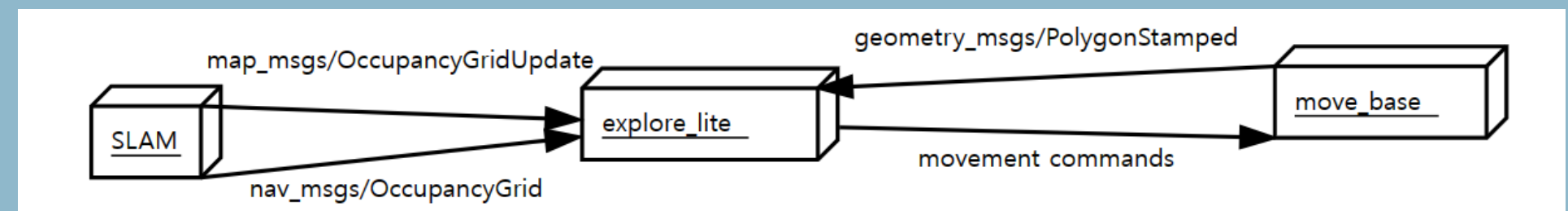
costmap_updates ([map_msgs/OccupancyGridUpdate](#))

Incremental updates on costmap. Not necessary if source of map is always publishing full updates, i.e. does not provide this topic.

4.1.3 Published Topics

~frontiers ([visualization_msgs/MarkerArray](#))

Visualization of frontiers considered by exploring algorithm. Each frontier is visualized by frontier points in blue and with a small sphere, which visualize the cost of the frontiers (costlier frontiers will have smaller spheres).



explore_lite 패키지를 사용하기 위해서는 action과 subscribe할 토픽이 필요하다.

Action(실제 이동)은 이동할 목적지를 알려주는 move_base_msgs/MoveBaseAction을 사용한다. (/move_base/goal 토픽 이용한다.)

그리고 노드가 실행되기 위해서 nav_msgs/OccupancyGrid 형태의 지도가 필요하다. (/map 토픽 이용한다.)

자동 매핑 - Active SLAM

explore_lite 패키지의 'explore' node 발행 정보

explore_lite는 탐사를 위해 move_base와 연동하여 작동한다.

```
/move_base_simple/goal
haneul@haneul-ubuntu:~$ rostopic info /move_base/goal
Type: move_base_msgs/MoveBaseActionGoal

Publishers:
 * /explore (http://172.20.10.4:46337/)
 * /move_base (http://172.20.10.4:46111/)

Subscribers:
 * /move_base (http://172.20.10.4:46111/)

haneul@haneul-ubuntu:~$
```

/move_base/goal을 통해 action을 publish 한다 (목표 지점을 설정해서 메시지로 보내는 과정)

```
haneul@haneul-ubuntu:~$ rostopic info /map
Type: nav_msgs/OccupancyGrid

Publishers:
 * /map_server (http://172.20.10.4:36015/)
 * /omo_r1_slam_gmapping (http://172.20.10.4:33365/)

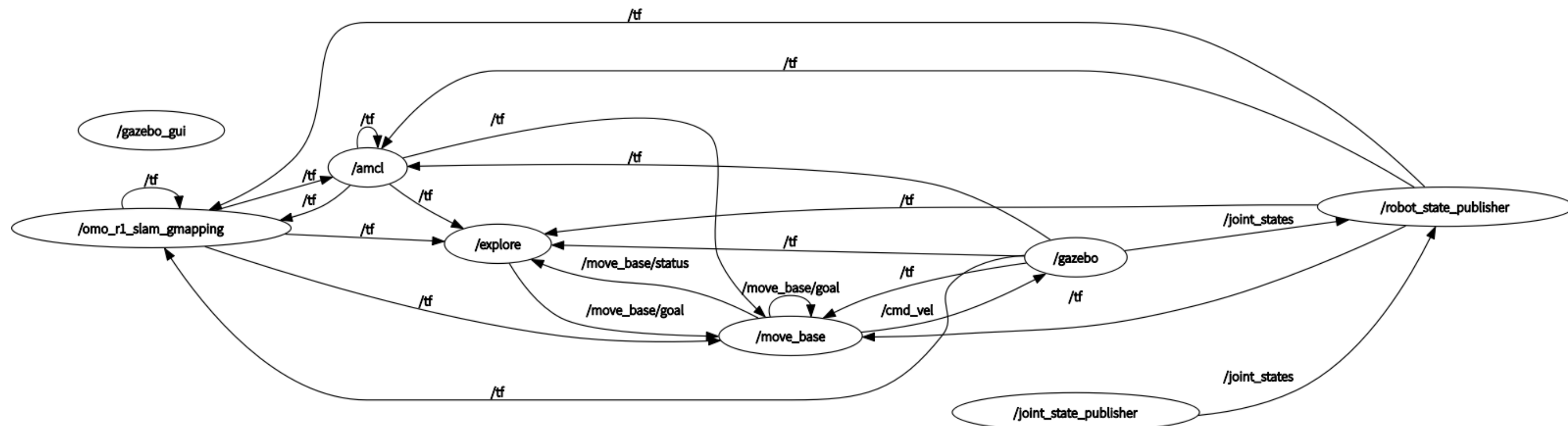
Subscribers:
 * /explore (http://172.20.10.4:46337/)
 * /rviz (http://172.20.10.4:40359/)
 * /move_base (http://172.20.10.4:46111/)
```

/map 토픽을 통해 OccupancyGrid 메시지를 받아 지도를 생성한다.

```
haneul@haneul-ubuntu:~$ rostopic echo /explore/frontiers
markers:
-
  header:
    seq: 0
    stamp:
      secs: 127
      nsecs: 301000000
    frame_id: "map"
  ns: "frontiers"
  id: 0
  type: 8
  action: 0
  pose:
    position:
      x: 0.0
      y: 0.0
      z: 0.0
    orientation:
      x: 0.0
      y: 0.0
      z: 0.0
      w: 0.0
  scale:
    x: 0.1
    y: 0.1
    z: 0.1
  color:
    r: 0.0
    g: 0.0
    b: 1.0
    a: 1.0
  lifetime:
    secs: 0
    nsecs: 0
  frame_locked: True
  points:
  -
    x: 2.2750001829117537
    y: 4.17500021122396
    z: 0.0
  -
    x: 2.3750001844018698
    y: 4.075000209733844
    z: 0.0
  -
    x: 2.2250001821666956
    y: 4.225000211969018
    z: 0.0
  -
    x: 2.2250001821666956
    y: 4.275000212714076
    z: 0.0
  -
    x: 2.1750001814216375
    y: 4.275000212714076
    z: 0.0
  -
    x: 2.1250001806765795
    y: 4.325000213459134
    z: 0.0
  -
    x: 2.0750001799315214
    y: 4.375000214204192
    z: 0.0
  -
    x: 2.0250001791864634
    y: 4.42500021494925
    z: 0.0
  -
    x: 1.9750001784414053
    y: 4.475000215694308
    z: 0.0
  -
    x: 1.9250001776963472
    y: 4.525000216439366
    z: 0.0
  -
    x: 1.8750001769512892
    y: 4.575000217184424
    z: 0.0
  -
    x: 1.8250001762062311
    y: 4.6250002179294825
    z: 0.0
```


자동 매핑 - Active SLAM

explore_lite 패키지의 'explore' node 발행 정보



/explore 노드가 각각의 노드에서 /tf 토픽을 받아와 위치를 파악하고 /move_base 노드에 goal 좌표를 보낸다

발표 흐름 예시

7분 (ppt 발표 + 시연 영상)

문제점 해결방안 기능소개

- 문제점 정의를 내리는 흐름
- 문제점 정의

- 사용자가 공간을 자유롭게 매핑
- Carebuddy가 매핑된 공간을 이동하며 공기질 데이터 수집
- 수집된 데이터를 바탕으로 LSTM을 통해 공기 케어 스케줄링 완성
- 스케줄링 및 사용자의 설정에 따라 공간을 이동하며 최적의 공기 상태 유지
- 긴급상황시 스케줄링 조정 가능

시스템 구조도

상세 설명

- how를 위주로
- 매핑, 네비게이션, 스케줄링, Segmentation, 긴급상황 (Demo내용)
- 시간상 넣을 수 있을지

시연 영상

기대 효과

공기청정기의 최적의 사용법을 모르는 사용자들이 건강하고 쾌적한 환경을 유지할 수 있다.

데모 시나리오

01

기존 위치

- 기존 위치에서 공기 청정

02

긴급 상황

- 연기 감지
- 해당 위치로 이동
- 공기질 센서를 통해 공기질 저하 확인 (가습기인 경우를 제외하기 위하여)
- 공기질 저하 확인시, 공기 청정

03

기존 위치 복귀

- 긴급상황 완료 후, 기존 위치로 돌아감

04

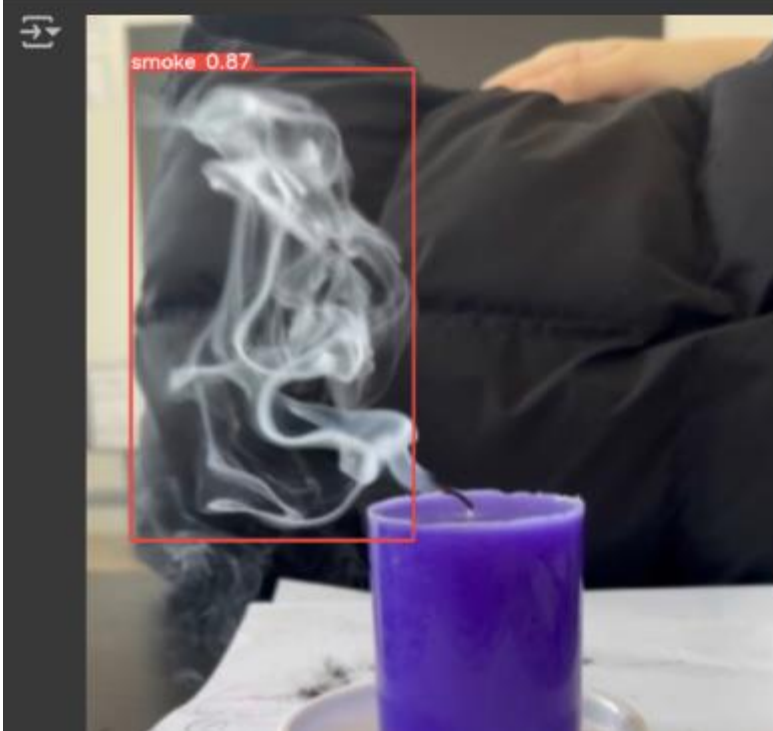
공간 이동

- 기존 위치의 공기 청정 완료 후, 스케줄링된 다음 공간으로 이동하여 공기 청정

데모 시나리오 : 연기 인식

```
[31] 1 %cd /content/yolov5/  
2 !python detect.py --weights runs/train/yolov5s_results/weights/best.pt --img 416 --conf 0.4 --source /content/candle_test.png  
  
/content/yolov5  
detect: weights=['runs/train/yolov5s_results/weights/best.pt'], source=/content/candle_test.png, data=data/coco128.yaml, imgsz=[416, 416], conf_1  
YOLOv5 v7.0-312-g1bcd17ee Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)  
  
Fusing layers...  
custom YOLOv5s summary: 182 layers, 7246518 parameters, 0 gradients  
image 1/1 /content/candle_test.png: 416x256 1 smoke, 80.8ms  
Speed: 0.5ms pre-process, 80.8ms inference, 503.5ms NMS per image at shape (1, 3, 416, 416)  
Results saved to runs/detect/exp4
```

```
[33] 1 import glob  
2 from IPython.display import Image, display  
3 from PIL import Image as PILImage  
4  
5 for imageName in glob.glob('/content/yolov5/runs/detect/exp4/*.png')[:10]:  
6     with PILImage.open(imageName) as img:  
7         width, height = img.size  
8         display(Image(filename=imageName, width=width//3, height=height//3))  
9
```



01

연기 데이터 학습

- roboflow 및 github를 통해 얻은 연기 데이터를 yolov5를 통해 학습
- 가중치 파일만 저장

02

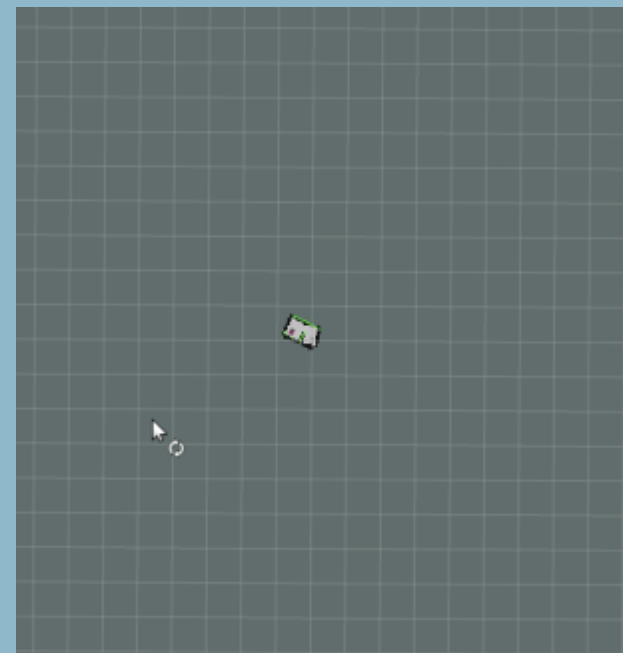
우분투와 로봇에 빌드

- 가중치를 우분투와 로봇에 빌드
- 연기를 감지하면 해당 위치로 가는 코드 작성

작은 공간 매핑 문제점

자동매핑, Segmentation

작은 공간에서 segmentation의 제약으로 인하여, 데모 시나리오 변경



60cmX90cm

01

미지 구역 없음



- 미지구역과 아는 구역의 경계선을 바탕으로 경로를 정하게 됨
- 라이다가 인식하는 범위보다 공간의 크기가 더 작아서 미지의 영역이 없다고 판단
- 자동매핑 불가능

YDLiDAR의 측정 범위 : 12cm~10m



90cmX114cm

02

Segmentation

- Segmentation될 만큼 공간을 나누면, 통로의 폭이 좁아서 로봇이 못 감
- 로봇이 갈 정도로 통로를 조금 넓게 하면 방으로 인식하지 못함

자동매핑 가능 공간



267cm

117cm

데모 공간 관련

- 5/20~5/24 제출하는 발표 동영상의 시연은 좌측 공간에서 진행할 예정
- 31일 판넬 부스 앞에서는 작은 공간+긴급 상황만 할 것인지

자동 매핑 - Active SLAM

자동 매핑 과정 중 발생한 문제

-> 로봇이 의자나 책상 다리와 같은 부분을 빙빙 도는 문제가 발생

예상되는 문제 이유 : 작은 프론티어 문제

로봇이 탐색 중에 너무 작은 경계(frontier)를 인식하여 비효율적인 탐색 경로를 선택하게 되는 상황을 의미한다.

알고리즘 코드 내의 'min_frontier_size' 파라미터를 설정하여 로봇이 탐색할 프론티어의 최소 크기를 지정함으로써
센서 잡음이나 오류로 인한 작은 프론티어를 무시하게 한다.

이를 통해 탐색 과정에서 발생할 수 있는 잘못된 목표 지점을 제거할 수 있을 것으로 본다.