



캡스톤디자인(2) 진행상황

Motus+er



INDEX PAGE

목차

- 1 Docker 컨테이너 생성
- 2 로봇 URDF 파일 생성
- 3 공간 분할 구현
- 4 4월 계획

현재 도커 컨테이너 현황

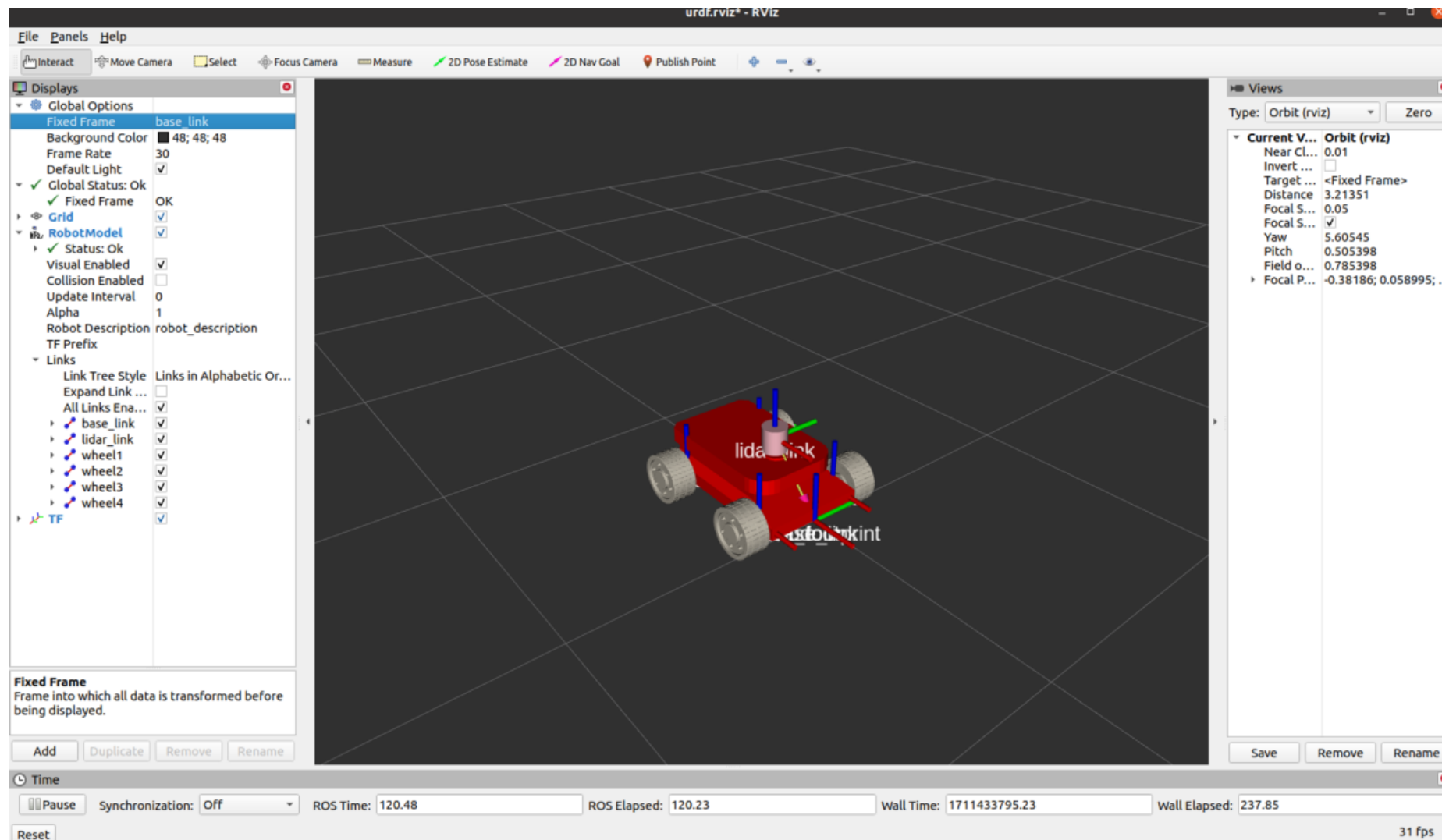


- Ubuntu 20.04
- ROS noetic
- mysql
- Git
 - ydlidar_ros_driver
 - rplidar_ros
 - hector_slam
 - map_server
 - rosbridge
 - roslibjs
 - ros2djs
 - Adafruit

ROS 패키지 실행 등 추가적인 테스트 필요

로봇 모델 구현

- urdf 파일 생성 및 Rviz 시각화 완료



로봇 GPIO 를 ROS로 제어하는 패키지 구현 필요

공간 분할 : 참고 문헌 및 라이브러리

IPA_ROOM_SEGMENTATION

필요 라이브러리



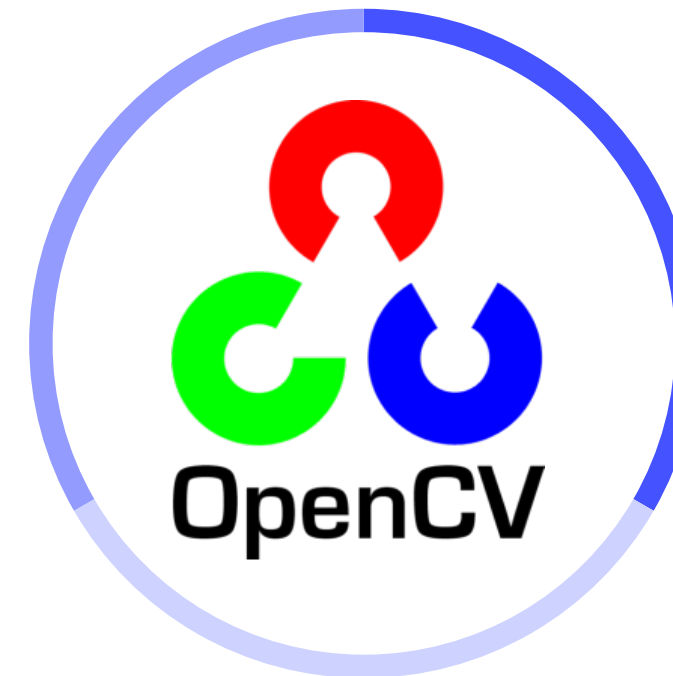
논문

R. BORMANN, F. JORDAN, W. LI, J. HAMPP, AND M. HÄGELE. ROOM SEGMENTATION: SURVEY, IMPLEMENTATION, AND ANALYSIS. IN PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2016.



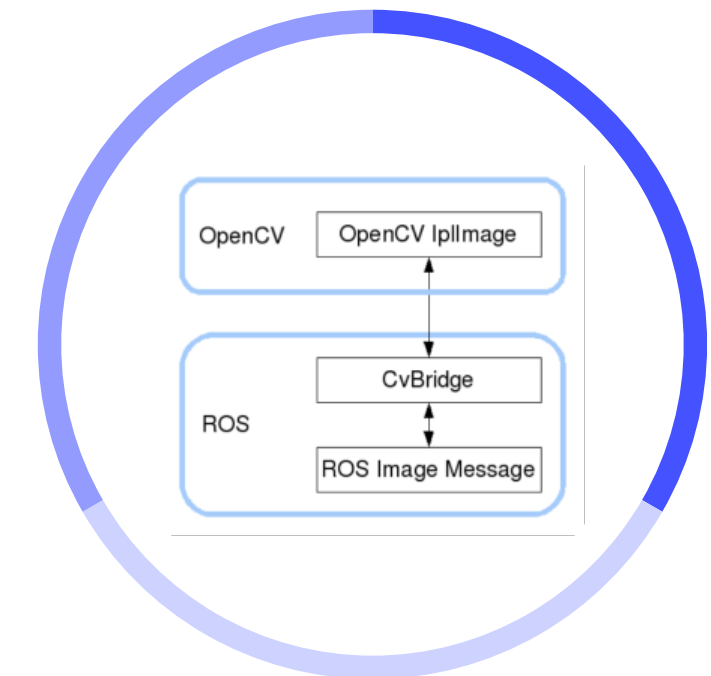
ROS WIKI

- IPA_ROOM_SEGMENTATION
- 깃허브 클론 후 빌드



OPENCV

지도 이미지를 불러올 수 있게 함



CV_BRIDGE

공간 분할 : 사용한 알고리즘

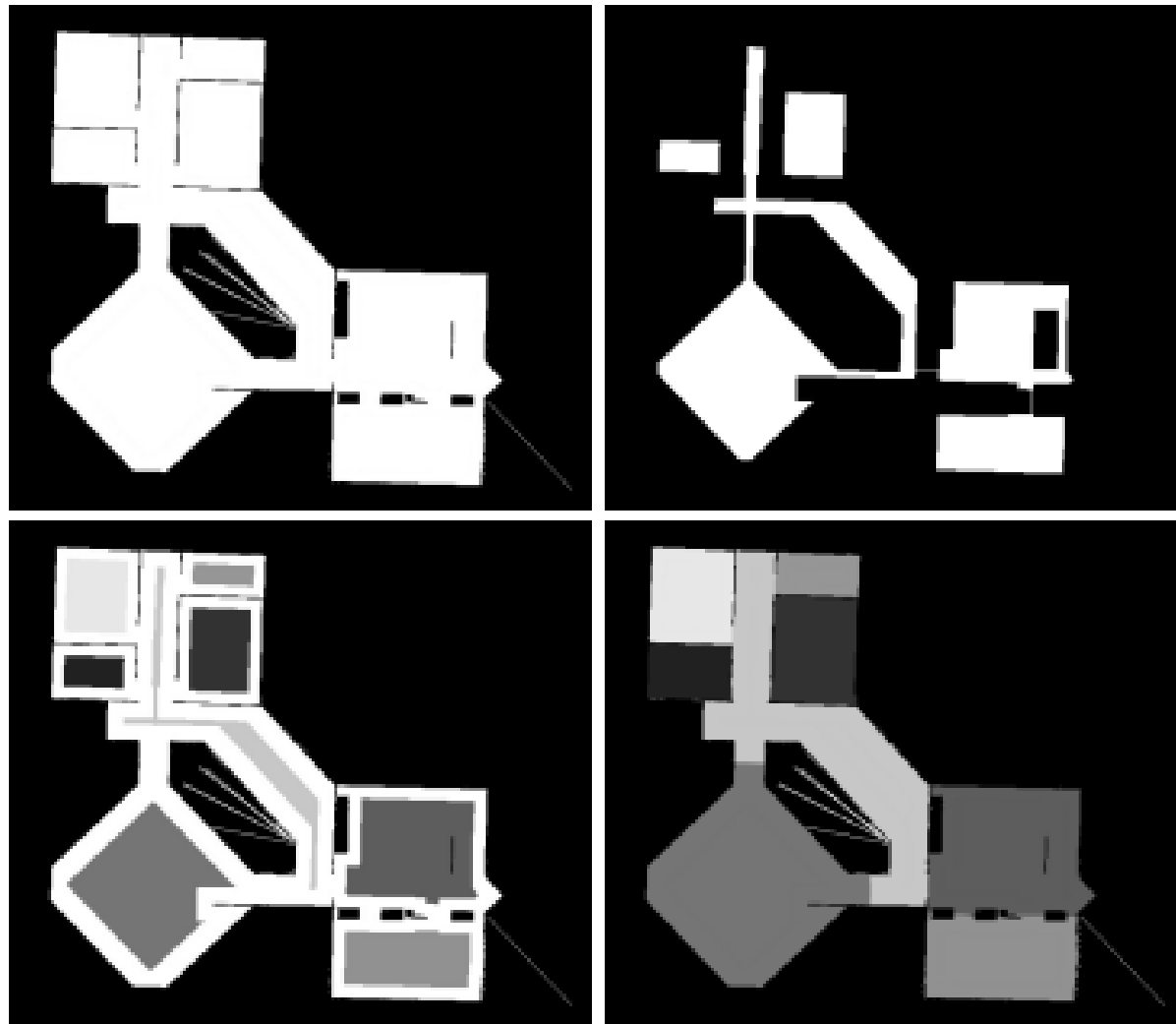


Fig. 2. Stages of the morphological segmentation algorithm: (i) initial floor map, (ii) iteratively eroded map, (iii) initial labeling of separated rooms, and (iv) segmentation after wavefront propagation.

MORPHOLOGICAL SEGMENTATION

1

- 지도 M1을 접근 가능(흰색), 접근 불가능(검은색)으로 설정

2

- 흰색 영역의 경계를 통해 조금씩 갉아내는 작업
- 넓은 흰색 영역을 점점 좁혀서 더 작은 부분으로 나눔

3

- 설정해둔 특정 크기의 분리 영역이 생기면, 영역을 식별 후 M2 지도에 표시
- 이는 M1에서 더 이상 고려되지 않음

4

- 모두 M1에서 고려되지 않을 때 까지 반복

공간 분할 : 설정 및 실행

설정

```
# this variable selects the algorithm for room segmentation
# 1 = morphological segmentation
# 2 = distance segmentation
# 3 = Voronoi segmentation
# 4 = semantic segmentation
# 5 = voronoi random field segmentation
# 99 = passthrough segmentation
# int
room_segmentation_algorithm: 1

# displays the segmented map upon service call
# bool
display_segmented_map: true

# publishes the segmented map as grid map upon service call
# bool
publish_segmented_map: true

# train the semantic segmentation and the voronoi random field segmentation
train_semantic: false
load_semantic_features: true
train_vrf: false

def segmentation_client(image_path, map_resolution):
    client = actionlib.SimpleActionClient('room_segmentation/room_segmentation_server', MapSegmentationAction)
    client.wait_for_server()

    cv_image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    if cv_image is None:
        rospy.logerr("Failed to load image from path: %s", image_path)
        return
    bridge = CvBridge()
    ros_image = bridge.cv2_to_imgmsg(cv_image, encoding="mono8")

    goal = MapSegmentationGoal()
    goal.input_map = ros_image
    goal.map_resolution = map_resolution

    goal.return_format_in_pixel = True
    goal.return_format_in_meter = False

    client.send_goal(goal)
    client.wait_for_result()

    return client.get_result()

if __name__ == '__main__':
    rospy.init_node('room_segmentation_client_py')

    parser = argparse.ArgumentParser(description="ROS Node to request map segmentation")
    parser.add_argument("image_path", type=str, help="Path to the input image")
    parser.add_argument("map_resolution", type=float, help="Map resolution in meters")
    args = parser.parse_args()

    try:
        result = segmentation_client(args.image_path, args.map_resolution)
        print("Segmentation result:", result)
    except rospy.ROSInterruptException:
        print("Program interrupted before completion.")
```

실행

액션 서버 실행

ROSLAUNCH를 사용하여
IPA_ROOM-SEGMENTATION 액션 서버 실행

알고리즘 설정

- 학습과정을 건너 뛰고, 사전에 학습된 모델로 학습과정 없이 동작하는 알고리즘
 - 1 : MORPHOLOGICAL SEGMENTATION (형태학적 분할)
- 학습을 하는 알고리즘도 사용해보았지만, 많은 시간 소요

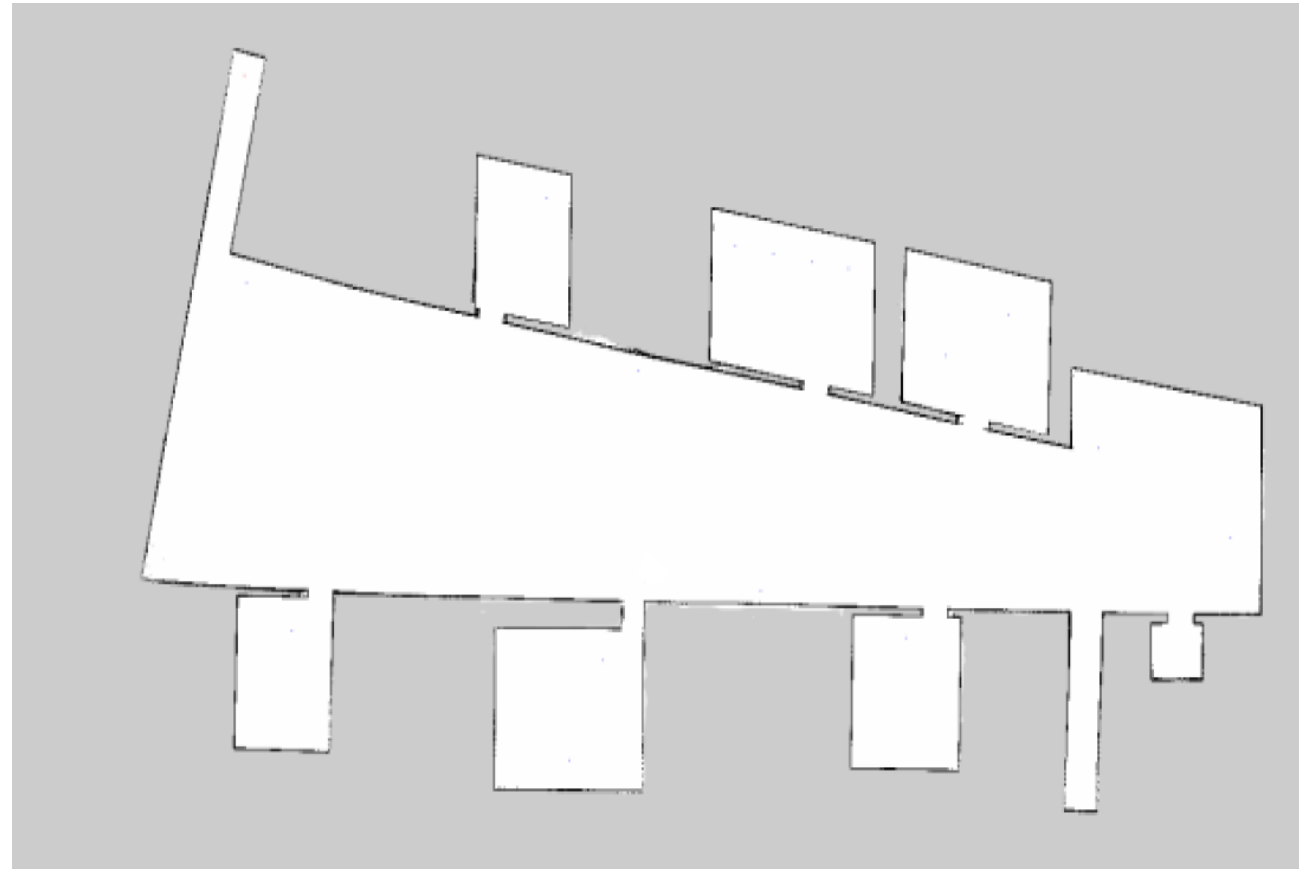
액션 클라이언트 파일 생성

- ROS 명령시 "저장경로"를 따로 설정할 수 있도록 코드 저장
- 해당 파일에 CHMOD로 권한 주기
- 이후 빌드

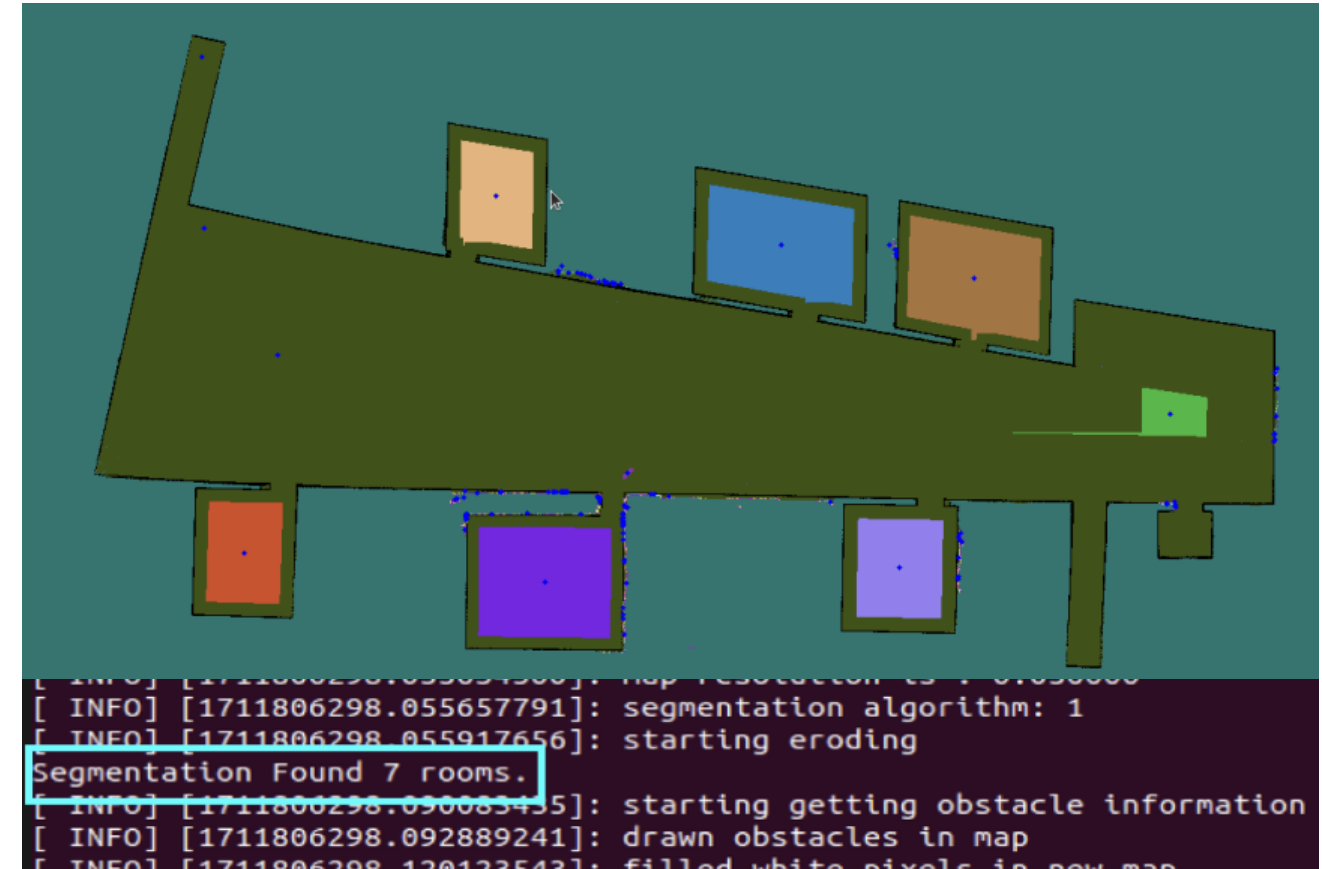
SEGMENTATION

- 액션 클라이언트 파일을 실행하여 SEGMENTATION요청
- ROS 명령어를 통하여 이미지 경로를 지정할 수 있도록 함

공간 분할 : 결과



기존 지도



분할된 지도

4월 계획

4월 말까지 최대한 완성을 목표로 함

- 네비게이션
- 자동매핑
- 공간 분할
- 카메라로 DNN 객체 인식
- 시계열 데이터화
- 공기질 스케줄링
- 구역별 공기 케어.

팀원 세부 계획 (4/1 ~ 4/28)				
	4/1~4/7.	4/8~4/14	4/15~4/21	4/22~4/28
김민규	도커 이미지 생성 완료 & DB 테이블 수정	센서 테스트 & 시계열 데이터화	시계열 데이터화	시계열 데이터화
임혜연	공간 분리 설정	공간분리 & 시계열 데이터화 통합 - DNN 객체감지 - DOCKER 사용	자동매핑 & 네이베이션.	자동매핑 & 네비게이션
장하늘	ROS로 로봇 제어	네비게이션	자동매핑	자동매핑

감사합니다

