

# Navigation

## 5.1 Navigation.

Navigation은 로봇을 목표지점으로 이동시키기 위해 굉장히 중요한 부분이다. 항법이라고 하는데 이는 선박이나 항공기, 차량 등을 한 장소에서 다른 장소로 이동시키기 위한 방법 또는 기술을 말한다.

### 5.1 Reactive navigation

빛을 추적하여 이동하는 방법, 지면에 선을 따라 이동하는 방법, 벽을 따라 이동하는 방법, 임의의 경로를 따라가는 방법등이 있다. 즉, 어떠한 변화에 반응하여 이동하는 방법이라고 말할 수 있다.

#### -5.1.1 Braitenberg vehicles

<https://www.youtube.com/watch?v=A-fxij3zM7g>

#### -5.1.2 Simple Automata

Braitenberg vehicle이랑 비슷한데 차이점은 주변 환경에 지나갈 수 없는 장애물이 존재한다는 점이다.



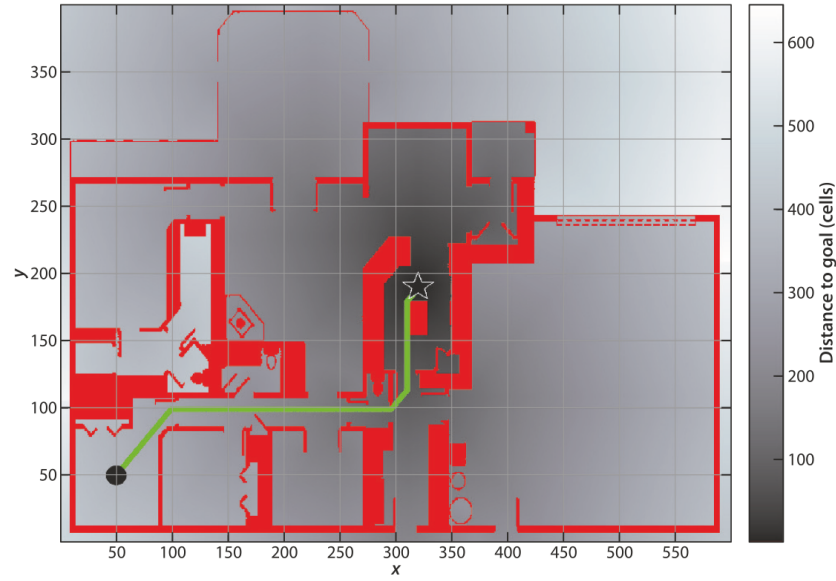
- 5.2 Map-Based planning

motion planning이라고도 한다. Reactive navigation과의 차이점은 reactive navigation은 센서로부터의 인식에 바로 반응하여 움직이는 반면, map-based planning은 **map에 기반**하여 시작점부터 목표점까지의 최적의 경로를 계획하여 움직인다는 점이다.

기본 조건이 있는데 하나는 robot이 grid world에서 작동한다는 것이고 다른 하나는 nonholonomic constraints를 가지고 있지 않다는 점이다. 즉 holonomic하다.

- 5.2.1 Distance Transform

Distance transform이란 거리 변환을 의미하는데 바이너리 이미지에서 픽셀값 0인 배경으로부터 픽셀값이 255인 영역을 표현하는 방법이다.

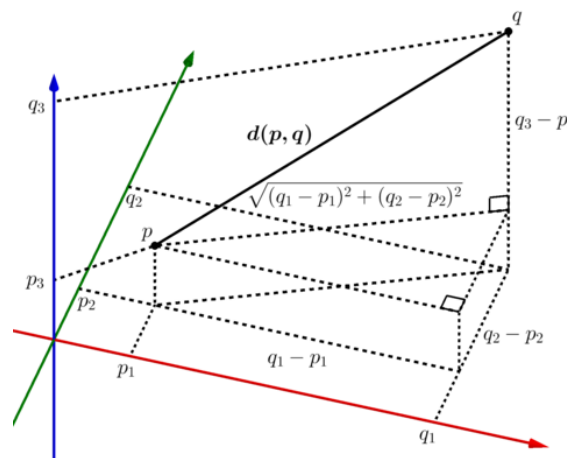


즉, 쉽게 말하면 그림과 같이 목표점 주위는 진하게 표시하고 그 주변부는 점점 멀어지게 표현하는 방법이다. 즉, 목표점은 픽셀값이 제일 낮고 주변부는 픽셀값이 높음을 알 수 있다. distance transform은 보통 이미지 프로세싱 기술이어서 ch.12에서 더 깊게 다룬다.

3차원 공간에서는 보통 출발점과 목표점의 거리를 계산할 때는 Euclidean distance를 사용한다.

<Euclidean distance>

3차원 공간에서 두점 사이의 거리를 구할 때 흔히 쓰는 방법인데

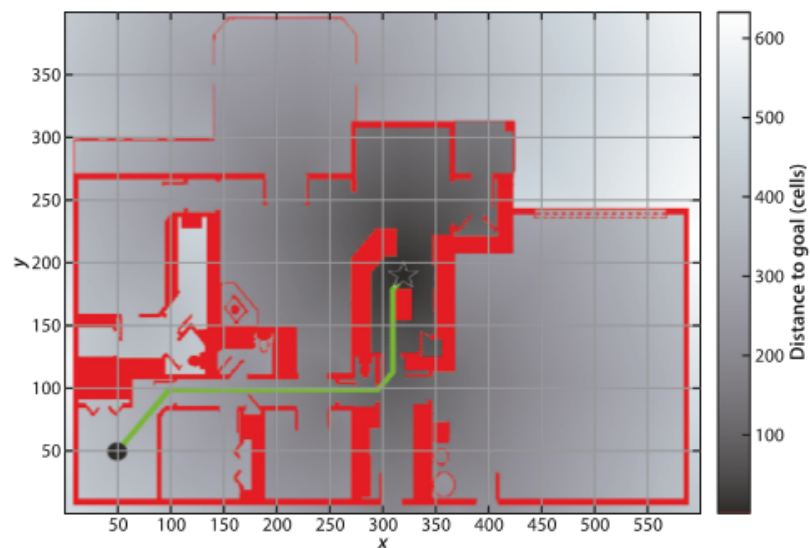


위의 유클리드 공간에서 p 점과 q점의 거리를

$d(p, q) = \text{root}((q_1 - p_1)^2 + (q_2 - p_2)^2 + (q_3 - p_3)^2)$  로 나타낸것이다.

#### -5.2.2 D\* ( finding minimum cost paths through a graph)

robot path planning의 가장 대중적 알고리즘이다. Incremental replanning을 사용하는 distance transform으로 이해하면 된다. Incremental replanning이란 말 그대로는 점진적 재설계이고 쉽게 이해하면 robot이 이동을 할 때 예기치 못한 변수나 장애물 등을 만날 수도 있기 때문에 그에 맞게 비용을 최소화하는 방향으로 재설계하여 이동한다고 생각하면 된다. 여기서 이제 비용은 우리가 어떤 것을 목적으로 삼는 가를 의미한다.



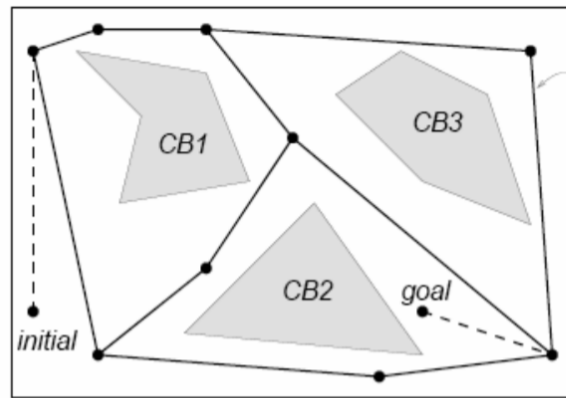
#### <D\*>

맨 처음에는 지형에 장애물이 없다고 가정을 하고 이러한 가정 하의 목표점까지의 최단 경로를 찾는다. 그러면 로봇이 경로를 따라 가는데 이제 새로운 정보 (예를 들면 알려지지 않은 장애물)가 관찰되면 해당 정보를지도에 추가하고 필요한 경우 현재 좌표에서 지정된 목표 좌표까지 새로운 최단 경로를 다시 계획한다. 목표 좌표에 도달하거나 목표 좌표에 도달 할 수 없을 때까지 프로세스를 반복한다.

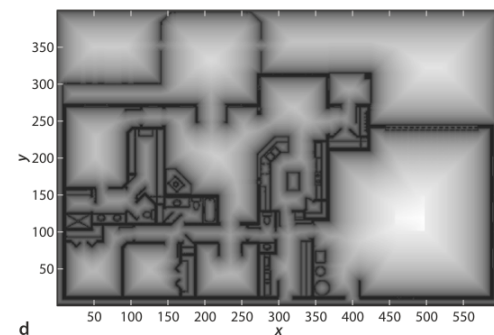
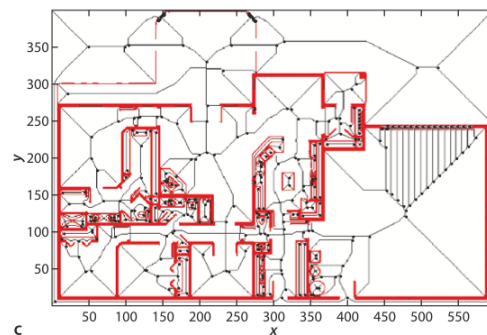
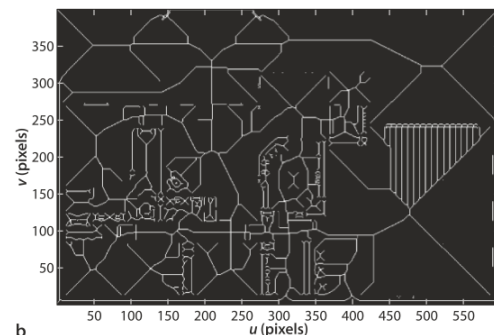
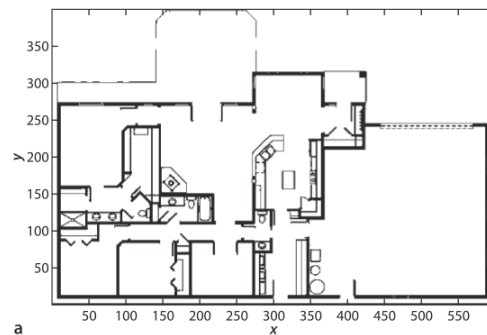
ros - <https://www.youtube.com/watch?v=jTt4lrsOqEU>

mit lecture - [https://www.youtube.com/watch?v=\\_4u9W1xOuts](https://www.youtube.com/watch?v=_4u9W1xOuts)

#### -5.2.3 Introduction to Roadmap Methods



장애물이 없는 free space에 있는 point끼리 길로 연결하는 방법.



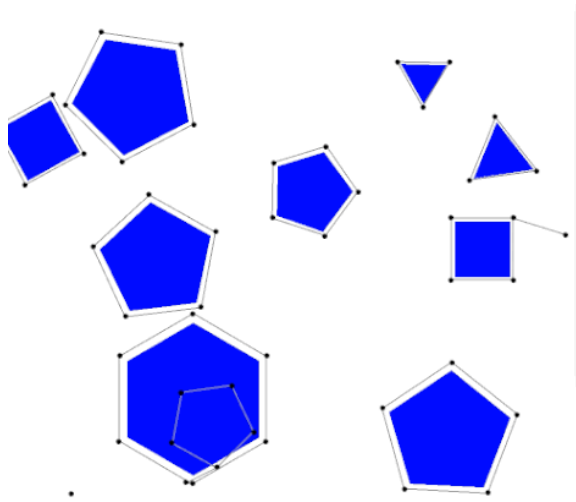
A는 위에서 봤던 지도를 단순화한 것이고, B는 A의 map을 skeletonization(topological skeleton)한 것이다. C는 B위에 원래의 map을 중첩한 것이고 D는 위에서 했던 distance transform을 적용시킨 것이다.

<skeletonization> //골격화?



#### -5.2.4 Probabilistic Roadmap Method (PRM)

확률적 로드맵이라고도 하는데 장애물이 없는 free space에 노드들을 랜덤으로 만들어서 이들을 서로 연결하는 방법이다.

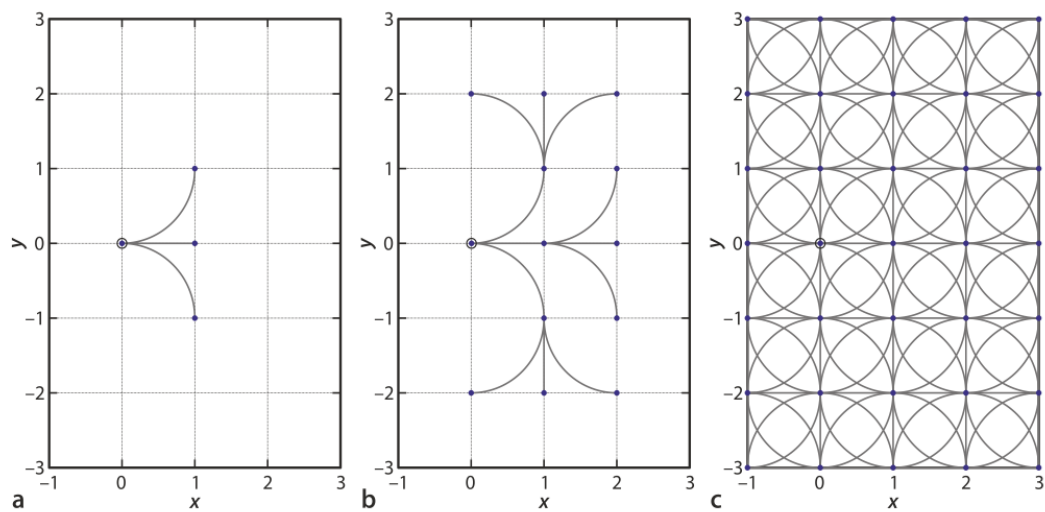




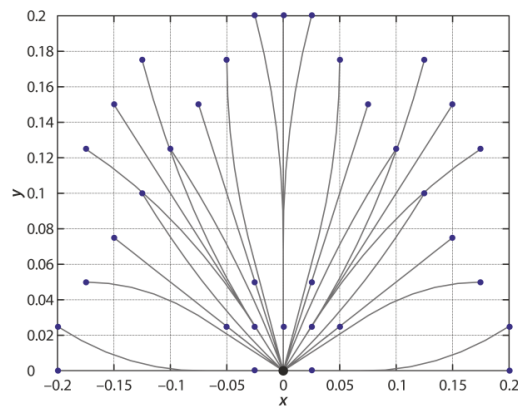
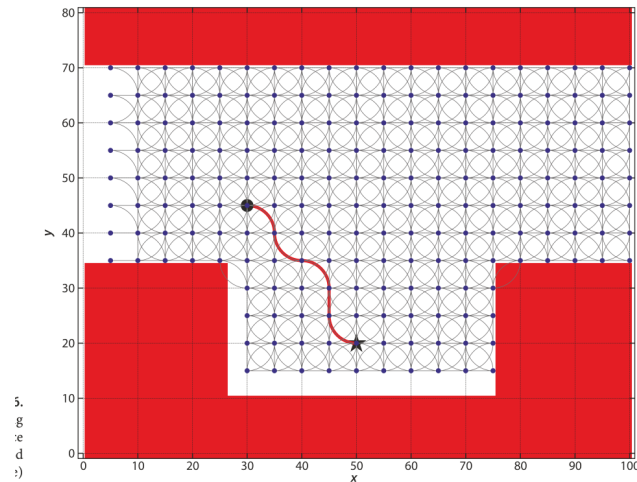
<https://www.youtube.com/watch?v=rKe6HO8LDu0>

### -5.2.5 Lattice Planner

앞서 나온 motion plan들은 vehicle constraints를 고려하지 않은 model을 대상으로 한 plan이고 lattice planner 와 RRT부분은 이 constraints를 고려한 plan이다.



A그래프를 보면 vehicle이 오른쪽을 보고 있을 때 갈 수 있는 방향이 3군데. b그래프에서는 도착한 각 node들마다 또 3군데 씩 갈 수 있기 때문에 이어서 계속 연결을 하면 최종적으로 c처럼 나오게 된다.

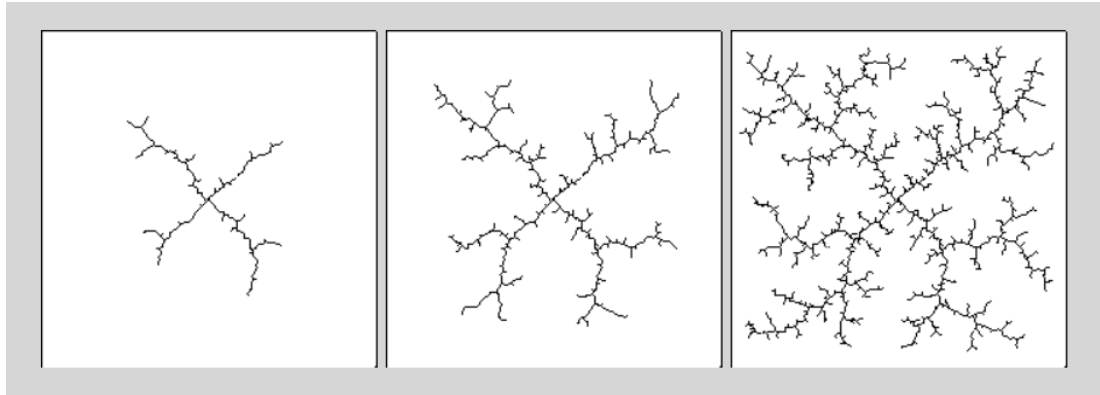


위의 그래프는 출발점부터 목표점까지 최단경로를 나타낸 것이다. 보면 알겠지만 조향을 급격히 변경해야하는 점을 보아 실제 vehicle이라면 타이어의 마모나 불편한 승차감이 있을 수 있다. 그래서 아래의 그래프처럼 arc를 더 크게 만들면 더 자연스럽게 정교한 motion을 할 수 있다.

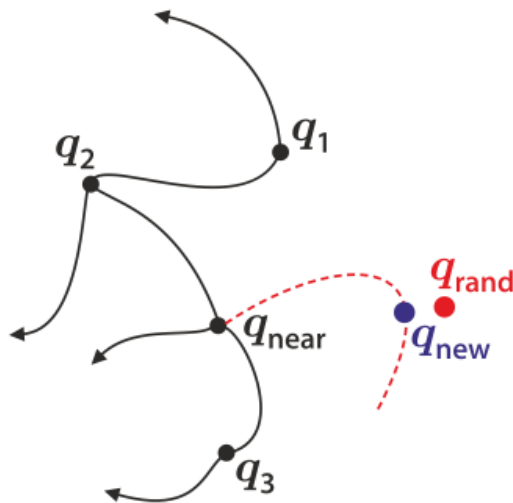
### -5.2.6 Rapidly-Exploring Random Tree (RRT)

RRT도 마찬가지로 vehicle의 motion을 나타내는데 앞서말한 PRM planning처럼 확률적 방법을 사용한다. 즉 격자 기반의 맵을 사용하지 않고 샘플링 기반의 path planning이다.





그림과 같이 나뭇가지처럼 퍼지는 모양으로 path plan을 하는 방법이다.



각 node들은 configuration  $q$ 를 나타내고  $q \sim (x, y, \theta)$ . 검정색 선의 initial configuration 상태에서 random configuration  $q_{rand}$ 이 생성된다. 그  $q_{rand}$ 이 free space에 존재하면 initial configuration에서  $q_{rand}$ 과 가장 가까운  $q_{near}$ 을 찾는다.  $q_{near}$ 에서  $q_{rand}$ 으로 향하는 길을 만들어 그  $q_{new}$ 를 생성한다.

<https://www.youtube.com/watch?v=xAmN8WnltRY>

[https://www.youtube.com/watch?v=cbYwxA9g\\_9k](https://www.youtube.com/watch?v=cbYwxA9g_9k)

<https://blog.naver.com/pasus/221982416404>

결론 :D\*,RRT

