**Open Cv를 활용한 교내 키위 로봇**

**주제 : 지도 api 와 gps를 이용한 현재위치 검증을 통한 차량 경로 탐색**

**GPS와 지도**

삼성에서 몇년 전 발표한 사내 논문에서 발표한 내용에 따르면 자율주행자동차에 있어 가장 핵심적인 부분은 원하는 목적지로 이동하는 경로를 자율주행차량이 파악하고 경로에 맞게 이동하는 것이다. 이 핵심적인 부분을 담당 하는 역할은 정밀지도와 1m 급 이내의 정밀 gps이다.



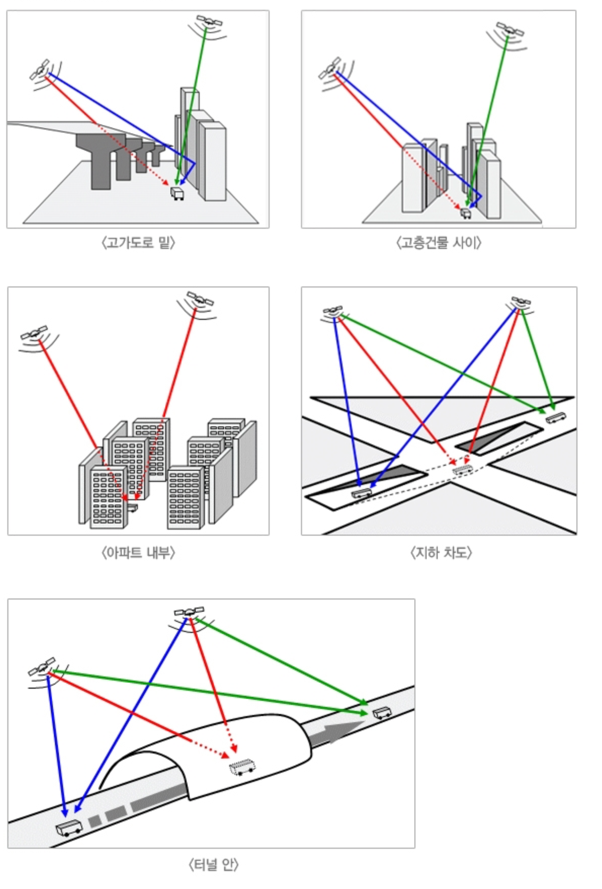
* 출처 : <https://1boon.kakao.com/HMG/5cbd9576ed94d200018e1b11>

그러나 정밀지도는 18년도가 되어서야 국토교통부에서 수도권 일부 도로를 대상으로 제작되었기에 자료가 매우부족하다. 그렇기에 Y&O는 쉽게 볼 수 있는 항법지도를 이용하여 키위로봇의 주행경로를 조정해야한다.

다음으로는 정밀gps이다. Gps는 현재 있는 위치를 파악해 가까운 교차로에서 키위로봇이 스스로주행경로를 재 탐색해야하기에 중요한 장치이며 gps로 부터 받는 절대 좌표는 현재 위치를 파악하는데, 중요한 역할을 하므로 gps의 성능은 이 키위로봇프로젝트에 지대한 영향을 미친다. Gps는 여러 위성들이 삼각측량이라는 원리를 이용하여 현재의 위치를 절대좌표 (X,Y)값으로 나타내는데, 이 X,Y값에 영향을 주는 것들로는 다음과 같은 것들이 있다.

다음장에서

**장소**



Gps 주변에 위성과의 교신을 방해하는 높은 건물혹은 구조물이 많다면 gps의 수신률이 낮아지게 되고 이로 인하여 현재 좌표 값에 대한 영향은 물론, gps가 현재 좌표를 위성으로부터 좌표를 받는 레이턴시가 높아져 현 위치와 좌표 간에 오차가 생긴다.

**날씨**

일반 적인 날씨는 gps에 큰 영향을 주지 않으나 구름이 많은 날씨 혹은 태양흑점 활동에 의한 위성신호 전파방해가 있을 경우 gps수신에 영향을 미친다.

**Gps진동 코어 성능**

100mHz  "$PMTK220,10000\*2F" // Once every 10 seconds, 100 millihertz

200mHz  "$PMTK220,5000\*1B"  // Once every 5 seconds, 200 millihertz

1HZ  "$PMTK220,1000\*1F"

5HZ  "$PMTK220,200\*2C"

10HZ "$PMTK220,100\*2F"

위의 값은 gps의 진동수에 따라 몇번의 계산이 가능한지를 보여주는 수치이다.

<http://shopping.interpark.com/product/productInfo.do?prdNo=5292362878&gclid=EAIaIQobChMI2IuDoLLS5AIVWLaWCh0rbg6eEAkYASABEgKzE_D_BwE>

이 GPS의 경우 default 1HZ부터 5HZ까지 지원하는데, 진동 수치가 높을 수록 이동시의 키위로봇의 실제좌표값과 gps로부터 받는 좌표값의 오차가 줄어든다.

<https://www.amazon.com/Expansion-Quad-Band-1900MHz-Raspberry-WIshioT/dp/B07NWKQ7ND/ref=sr_1_14?crid=37XCUZF4FIIBM&keywords=raspberry+pi+gps&qid=1568535423&sprefix=rasber%2Caps%2C1238&sr=8-14>

(이 제품의 경우 1900MHZ급 성능을 보여주는 듯 하다. )

**Gps의 성능에 대한 개인적인 생각**

현재 시장에 나와있는 gps들을 프로젝트에 구성하여 나온 결과를 보면 가만히 있을때는 오차값이 거의(1m)적은 정도로 정확한 모습을 보여주는데, 움직이는 경우 약 (1m ~ 3m)까지의 오차를 보여주는 듯하다.

이 오차는 해당 경로이탈오류와 직결될 수 있기에 중요하다. 또한 교차로를 키위로봇의 절대 좌표로 인식하기에 연속적인 우회도로가 나올 경우 올바르지 못한 교차로로 진입할 수 있다.

이러한 gps가 발생시키는 오차때문에 학교의 도로와 지형이 굉장히 중요하며 지도의 크기 또한 중요하다.

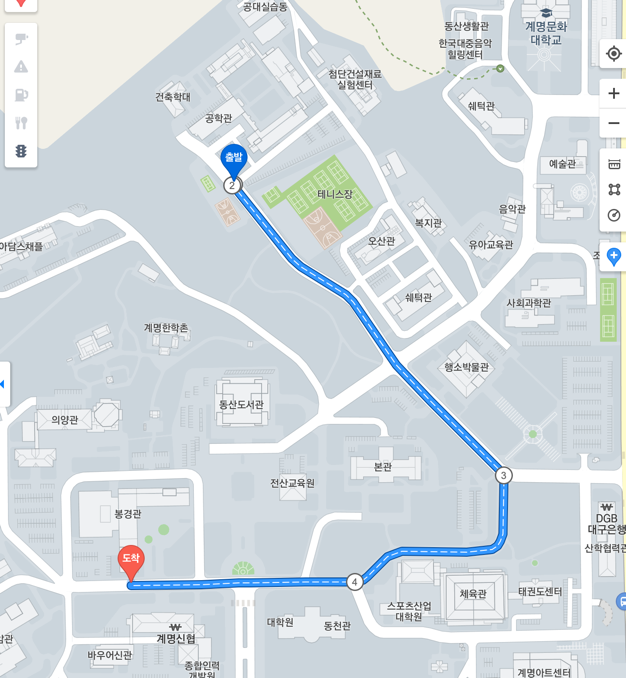
**MAP API와 네비게이션 API를 이용한 경로 탐색**

현재 주행경로를 탐색하는 방법으로는 구상한 것은 map api와 네비게이션 api를 이용한 경로 탐색이 있다.

현재로써는 플랫폼의 한계 때문에 두 방법 다 지도 혹은 네비게이션이 켜져있는 스마트폰을 웹캠을 이용하여 읽고 분석해야할 듯 하다. (이 방법의 경우 스마트폰과 웹캠의 간격 및 초점이 정확해야한다.)

**MAP API를 이용한 주행 경로 탐색**

Map api는 단순히 현재 위치와 도착할 곳을 정해주면 2차원 지도에 경로가 나타난다.



이 경로 이미지와 현재 위치를 나타내는 지도 이미지를 교차검증하는 방법을 사용하여 교차로를 판단하여 정확한 경로로 이동 할 수 있게 한다.

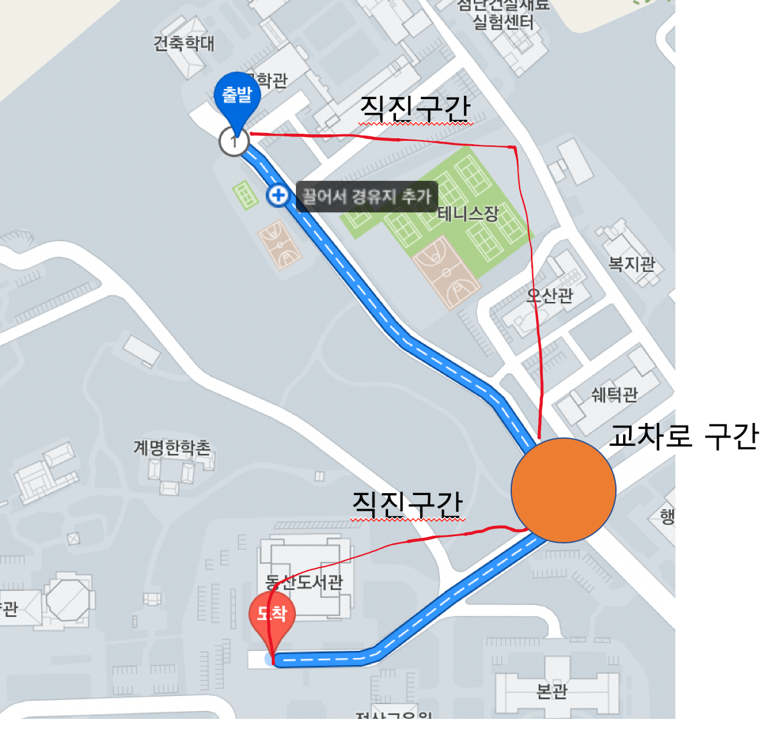
**교차로 판단**

카카오맵의 특이점으로 해당 거리까지 이동경로를 보여줄때 교차로를 번호로 표시해주는 기능이 있다.

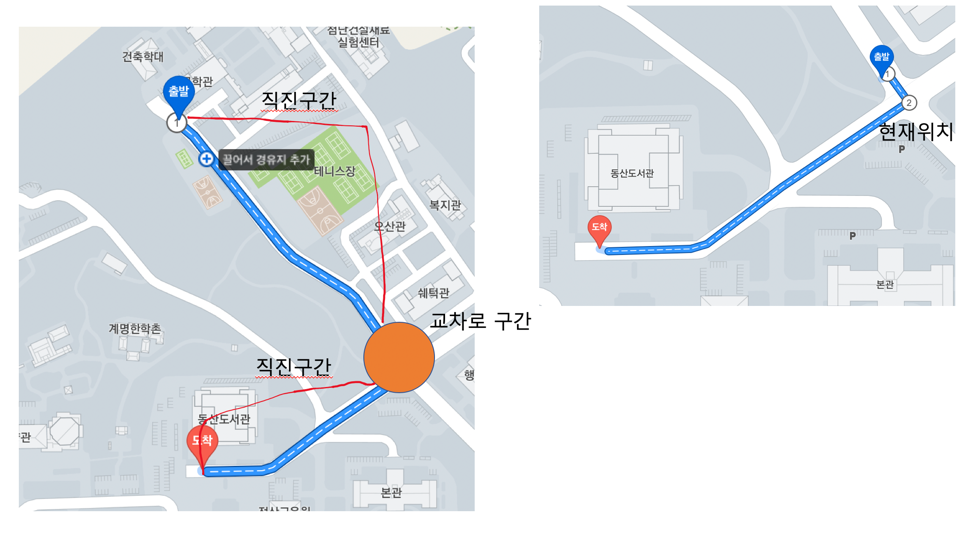
이를 이용하여 이 원의 반경 10m 거리 까지 도달 했을 때 특별한 우회전 , 자회전, 코드를 실행하는 것으로 조향을 제어합니다.

****

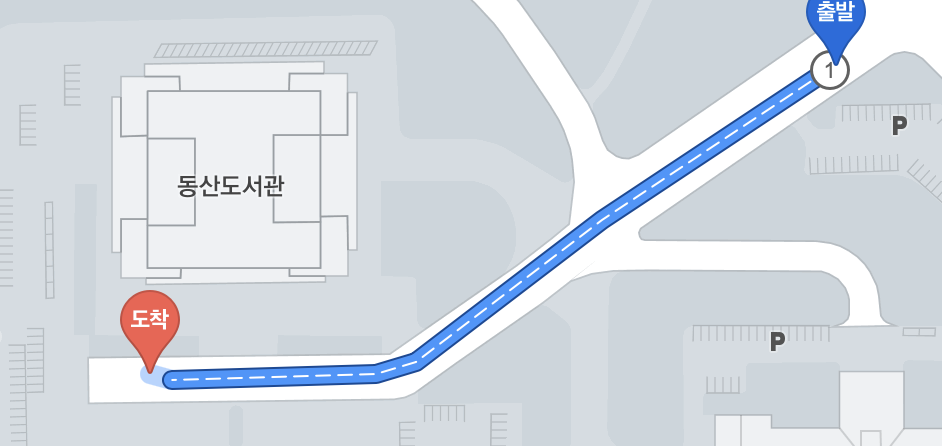
**예제 : 공학관부터 동산도서관 입구까지**

****

직진구간까지 키위로봇은 경로와 맵api로부터 받는 출발 아이콘을 교차 검증하며 직진 로직만 수행한다.

****

교차로 구간에 접어들게 되면 방향에따라 좌회전 , 우회전 로직을 수행하여 우회전을 끝마친후 직진로직을 실행하게 된다.



**네비게이션 API를 이용한 주행 경로 탐색**

네비게이션을 이용한 주행 경로 탐색은 정확도가 높으나 난이도가 한층 올라가는 작업이 되리라 예상된다.

이 방법은 다음과 같은 순서에 의해 작동된다.

1. 외부에서 호출 시 네비게이션 api에 현재 gps위치, 도착하는 곳 위치를 전달하여 네비게이션 앱을 활성화 시킨다.
2. 교차로 특정 반경 전 까지는 직진 로직만 수행한다.
3. 활성화 시킨 앱의 상단에 나오는 좌회전 우회전 신호 및 거리를 보고 특정 반경 도착시 우회전 , 좌회전 로직을 실행한다.



**개인적인 생각**

어느 두가지의 방법 전부 전면 차선을 인식하는 카메라 말고도 스마트폰을 보는 어플을 관찰해야 하니 cpu부하가 얼마나 되는지 전혀 감을 못 잡겠다. 또한 외부에서 스마트폰을 동작시키고 키위로봇을 작동시켜야하니 이는 얼마나 난이도가 높은지 모르겠다.

예상되는 로직으로는 키위로봇은 안드로이드에서 오는 신호를 기다리고 있다가 안드로이드가 자체제작 어플리케이션을 통해 서버에서 좌표와 목적지를 받고 이를 지도 혹은 네비게이션 API에 전송하고 켜지면 라즈베리파이를 작동시키는 것으로 시작해야 할 듯 하다.

하드웨어적으로 우려되는 점은 웹캠이 안드로이드폰을 보고 있을 때 거친 도로표면으로 인하여 초점이 소실되던지, 화면이 흔들려서 잘 못 인식될까 우려된다. 최대한 완충을 하여 시몬스급 편안함이 필요하다.

난이도가 좀 많이 어려울 것으로 예상되며 개발인력 2인정도가 더 필요할 것을 예상된다.