10/8 구글어시스턴트 서비스 설치 및 모터 좌우 실험

**구글 어시스턴스 서비스**

구글 어시스턴트 공식 문서에 따르면 python라이브러리 형태로 제공되던 assistant가 2019년 6월 부터 종료된다.

이에 alobo의 ai- speaker기능에 차질을 빚게 되었다. 그래서 ai speaker가 rpi에서의 성능만 보기위해서 구글 어시스턴트 서비스를 설치했다. 어시스턴트 서비스는 우리의 서비스와 별개로 동작하는 프로그램이다.

어시스턴트를 공식 문서에서 안내하는 대로 설치하였는데, portaudioerror -1 이라는 문구가 떴는데, 이 오류를 찾아보니 어시스턴트는 마이크와 스피커 없이는 실행조차 안되는 듯하다. 이로 인하여 금일 ai speaker성능 실험은 종료되었고, 11일 금요일 다시 한번 usb마이크를 가져와서 실험할 예정이다.

현재까지 우리가 따로 서비스를 변경하거나 우리의 서비스와 연결 시킬 수 있어 보이지는 않다.

**모터 좌우 실험**

모터의 좌우 회전 실험은 다음과 같은 순으로 진행하였다.

사람의 얼굴의 위치에 따라 오른쪽 혹은 왼쪽으로 회전 -> 얼굴이 화면 중앙과 가까우면 모터 정지 -> 얼굴을 화면 정중앙까지 모터 회전

(최초 실험은 황세연의 얼굴과의 각도 공식을 이용하였으며 추후 오류에 의해서 알고리즘이 변경되었다.)

**사람의 얼굴의 위치에 따라 오른쪽 혹은 왼쪽으로 회전**

Detect\_face로 부터 받은 얼굴 좌표에서 얼굴 정 중앙의 x좌표는 (x1-x2)/2 이다. 이 x좌표를 기준 하여 화면 정중앙의 x좌표 320보다 크다면 왼쪽에서 오른쪽으로 모터가 회전해야 하고, 320보다 작다면 오른쪽에서 왼쪽으로 모터가 회전 할 수 있도록 작성하였다.

**얼굴이 화면 중앙과 가까우면 모터 정지**

황세연 학생의 얼굴과의 각도 공식을 이용하여 만약 이 값이 3도 이하인 경우는 작동하지 않게 제어문을 사용하였다.

얼굴과의 각도를 구하는 방법은 다음과 같다.

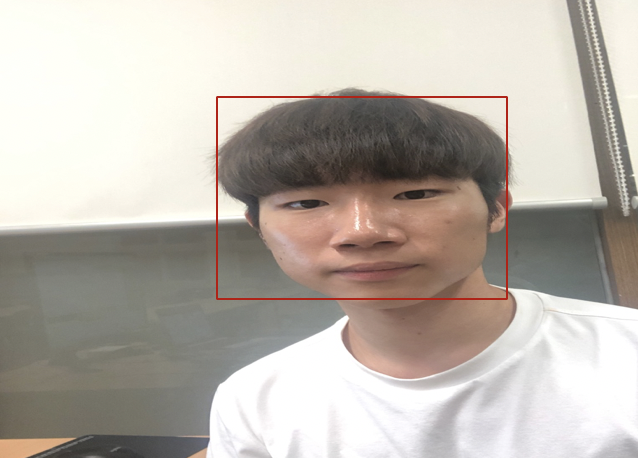
middlex = (x1-x2)/2

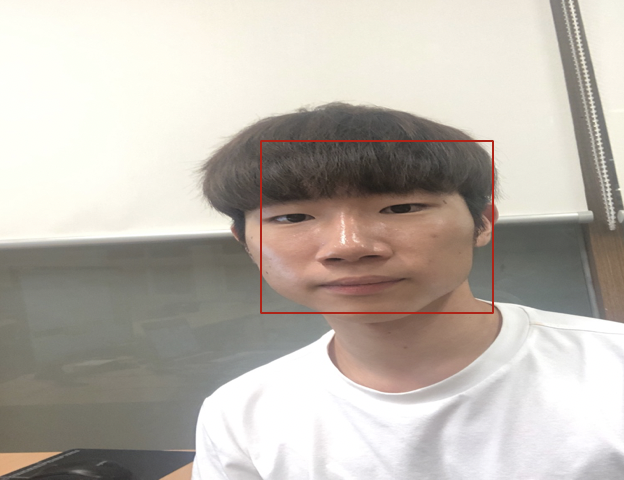
middley = (y1-y2)/2

xarctan = (math.arctan(middlex))/middley //리디안 값으로 반환된 세타 값

xartan = math.atan(xarctan) // degree로 반환된 세타 값

이를 이용하여 얼굴과 화면정중앙의 회전 각도를 구했다. 그러나 이 과정에서 우리는 치명적인 실수를 했다는 것을 깨달았다. 저 공식을 적용 해서 세타값을 확인해보니 시시각각 큰폭으로 세타값이 변하는 것이었다. 이유를 살펴보니 opencv가 얼굴을 실시간으로 탐지 할 때 얼굴의 일부분을 얼굴로 인식할 때도 있고 머리를 얼굴전체로 인식하는 경우도 있어 얼굴의 정중앙 좌표가 계속 변화하는 것이었다.





그렇기에 얼굴과의 각도가 시시 각각 변했고 화면 중앙에 얼굴이 위치해도 세타값이 계속 변하는 모습을 보였다. 또한 해당 공식은 y의 거리로부터 영향을 받기에 같은 위치에서 y좌표가 변화하면 최종 값도 심하게 차이 났다.

이로 인해서 x좌표 만큼 이동 할 수 있도록 알고리즘을 변경해야 만 했다.

**얼굴이 화면 중앙에 올 수 있도록 모터 회전**

변경된 알고리즘은 x좌표만 이용해서 x좌표가 화면 정 중앙(320)과 비교하여 340 보다 크다면 왼쪽으로 5step씩 주어 얼굴과 카메라의 위치를 계속 변경할 수 있도록 하고, 반대로 300 보다 작다면 오른쪽으로 5step 씩 주어 얼굴이 detect될 때 마다 모터가 올바른 방향으로 정해진 각도만큼 적게 움직이게 변경 하였다.

결과는 성공적이었으며, 한번에 모터를 필요한 step동안 움직이게 하는 알고리즘에 비해 다른 작업들을 쉽게 더 할 수 있게 바뀌었으며, 얼굴 정중앙까지 움직이는 카메라 이동도 정확해졌다.

(영상은 종호 폰)