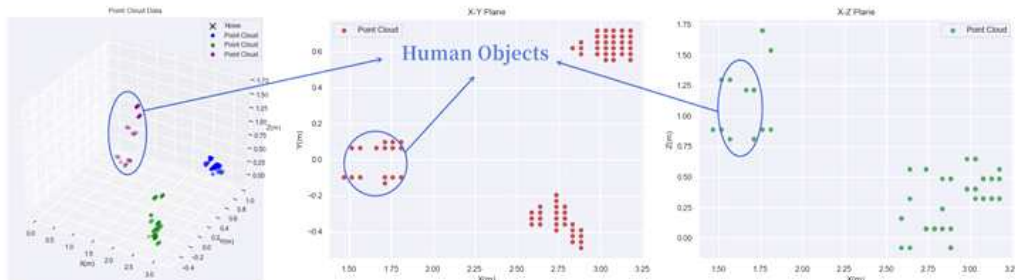


개선점 생각

1. DBSCAN 직접 구현 - 사이킷런 제공 DBSCAN은 좀더 일반적인 데이터까지 클러스터링 할 수 있게 구현이 되어있는지 직접 구현한 것과 비교해서 속도가 매우 느리다. 200개의 특징 2개의 데이터를 사용했을 때 직접 구현한 것에 비해 속도가 40배가 넘게 차이 나는 글도 있다. <https://zephyrus1111.tistory.com/356> 어차피 visualizer에 쓰는 DBSCAN은 어떤 특징을 쓰는지 확실하게 나오기 때문에 그 특징 수량 데이터 수에 맞게 정확하게 구현하면 속도가 더 빨라질 것이다.
2. DBSCAN 구현 과정에서 Z축 (vertical)에 대한 가중치 감소시키기 -



이 그림을 보면 클러스터에서 x-y 평면에서보다 x-z 평면에서 point들의 일관성이 없는 모습을 볼 수 있다. 따라서

$$D^2(p^i, p^j) = (p_x^i - p_x^j)^2 + (p_y^i - p_y^j)^2 + \alpha(p_z^i - p_z^j)^2$$

이 식처럼 z축에 가중치를 부여해서 유클리드 거리를 구해서 DBSCAN을 구현하는 것이 필요해보인다.

3. Tracking 알고리즘 개선하기

프레임 별로 클러스터들을 이어주는 방법을 지금은 그냥 연속된 두 프레임에 대해서 클러스터 중심의 유클리디안 거리가 가장 가까운 클러스터끼리 같은 클러스터라고 생각하고 이어지게 했다. 당연히 성능이 좋지 않다.

성능이 좋게 나온 논문에서 사용한 프로그램에서는 simplified Global Nearest Neighbour 알고리즘을 써서 트랙을 이어주고 정확한 트랙 예측을 위해서 칼만 필터를 사용했다고 한다. 하지만 논문에서도 이 부분을 그냥 이 알고리즘을 썼다 하고 넘겨버렸다... 그래서 이 부분을 찾아서 구현해보면 좋을 것이다. 이것 외에도 다른 tracking 방법들이 있을 것이다. 다음 팀은 이쪽 부분을 생각해서 만들어주면 좋겠습니다.

4. 외부로부터 들어오는 새로운 물체를 새로운 물체로 인식하기

지금은 외부에서 새로운 물체가 들어올 때 그 물체가 원래 이 안에 있던 물체인지 새로 들어온 물체인지 인식 하는게 알 수가 없게 되어있다. 이런 intruder 판단을 하는 방법에 대한 한글에서는 손실 함수에 center loss라는 값을 추가해서 random forest classifier에 넣는 방식으로 해결했다고 한다. 자세한 설명은 논문 설명을 써놓은 글을 같이 첨부할테니 읽어보면 좋겠습니다.