\*데이터 저장 폴더 설명

[1]

~/Workplace/Point-Cloud-Registration/

~/Workplace/Point-Cloud-Registration/pairwise\_registration

[2]현재 리눅스 머신 아래 폴더에 깨진 헤드 스캔 데이터와 새로만든 헤드 스캔 데이터 저장

~/Workplace/reactor\_head\_scan\_crash\_head

~/Workplace/reactor\_head\_scan\_original\_head

→ 두개중 상황에 따라 폴더명을 ~/Workplace/reactor\_head\_scan으로 바꿔서 사용

( 처리하려는 데이터를 ~/Workplace/reactor\_head\_scan 폴더에서 실행해야함)

→ 현재 ~/Workplace/reactor\_head\_scan 폴더에 crashed head 데이터 들어있음

[3] 현재 리눅스 머신 아래 폴더에 깨진 헤드 스캔 데이터와 새로만든 헤드 스캔 데이터 정합한 데이터(interactive\_visualization\_reactor\_head\_r0.py실행 결과) 저장되어 있음

~/Workplace/Point-Cloud-Registration/pairwise\_registration\_for\_crashed\_head

~/Workplace/Point-Cloud-Registration/pairwise\_registration\_for\_original\_head

→ 처리하려는 데이터를 ~/Workplace/Point-Cloud-Registration/pairwise\_registration 으로 옮겨서 프로그램 실행해야함

----------------

\*프로그램 설명

[1] interactive\_visualization\_reactor\_head\_r0.py

pairwise\_manual\_registration (소스, 타켓, 파라미터 ) → 수동으로 점 세개 선택해서 두개의 점군 정합, 변환행렬은 ./pairwise\_registration폴더에 저장, 다운샘플링된 점군도 저장

|  |  |  |
| --- | --- | --- |
| code name / description | input | output |
| interactive\_visualization\_reactor\_head\_r0.py  /  수동으로 점 세개 선택해서 두개의 점군 정합 | ~/Workplace/reactor\_head\_scan/아래에 있는 점군중  source와 target을 지정 | ./pairwise\_registration/source\_pcd\_data\_clean\_downsampled.ply  ./pairwise\_registration/target\_pcd\_data\_clean\_downsampled.ply  ./source\_target\_transformations.npz |

[2] incremental\_registration\_reactor\_head\_r1.py

pcd\_data\_name = 소스데이터 이름

target\_pcd\_data\_name = 타겟데이터 이름

./pairwise\_registration/ 폴더에 있는 변환행렬을(scanN\_scan(N+1)\_transformations.npz) 읽어와서 딕셔너리에 저장, 색깔도 칠함

tf\_01\_02, …, tf\_12\_13 형태로 변환행렬 생성

하나로 합치면서 가시화

데이터가 16개면 tf\_01\_02 ~ tf\_15\_16까지 구한 다음에 데이터는 scan01~scan15까지 정합한다.

pairwise incremental registration 중간 단계 스냅샷 그릴 때 주의사항 → 개선 필요

: deepcopy를 안하고 그냥 트랜스포메이션을 적용하면서 그림 여러개를 그리다보면 변환이 누적되어서 맨 처음 그림만 제대로 나오고 뒤로는 변환이 누적된/틀어진 그림 그려짐

|  |  |  |
| --- | --- | --- |
| code name / description | input | output |
| incremental\_registration\_reactor\_head\_r1.py  수동으로 점 세개 선택해서 두개의 점군 정합 | ~/Workplace/reactor\_head\_scan/아래에 있는 점군중 source와 target을 지정  소스데이터 이름: pcd\_data\_name  타겟데이터 이름: target\_pcd\_data\_name  ./pairwise\_registration/ 폴더에 있는 변환행렬scanN\_scan(N+1)\_transformations.npz | (1) 정합후 그림  (2) 정합된 점군 파일  ./incremental\_registration/after\_registration.ply |

→ 개선사항

: ply 로 저장하지 말고 h5로 저장해보기

ply에 색상정보도 포함되나??