



Point cloud from Scan-Net

RE. CONSTRUCTION SE. GMENTATION AU. GMENTATION

ADVANCED TOPICS IN COMPUTER VISION

GROUP II

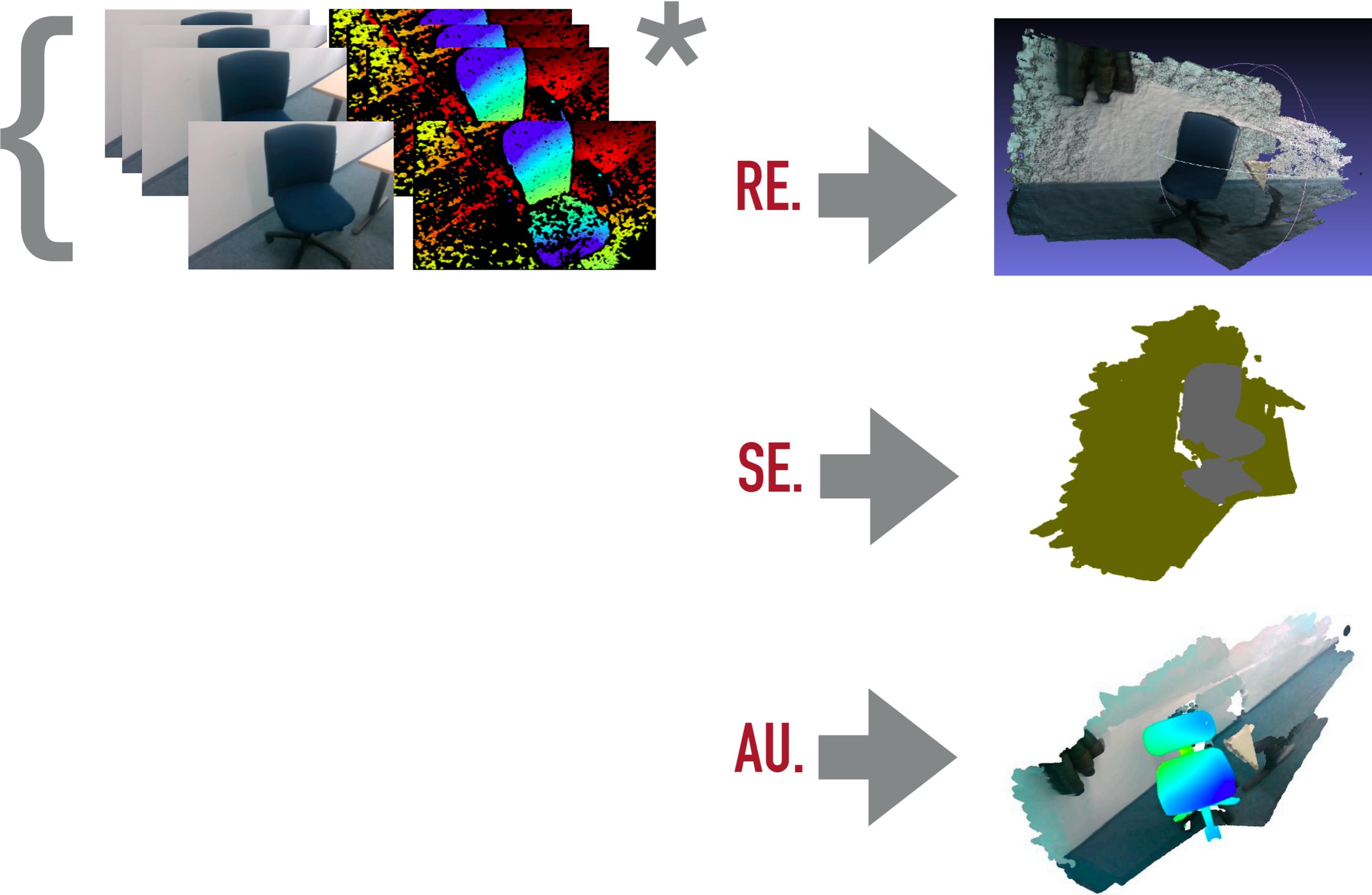
-
- REPLACE CUSTOMER'S FURNITURES**
 - CREATE NEW EXPERIENCES**
 - LET CUSTOMER HAVE A BETTER UNDERSTANDING OF HIS LIVING PLACE.**

Who We Are.

WHAT WE DO

- ▶ We Capture a stream of RGB-D images and 3D-Reconstruct the area.
- ▶ We use the reconstructed data and segment the objects inside.
- ▶ We use data above to provide customer some augmentation facilities, e.g. changing and removing furnitures.

INTRODUCTION



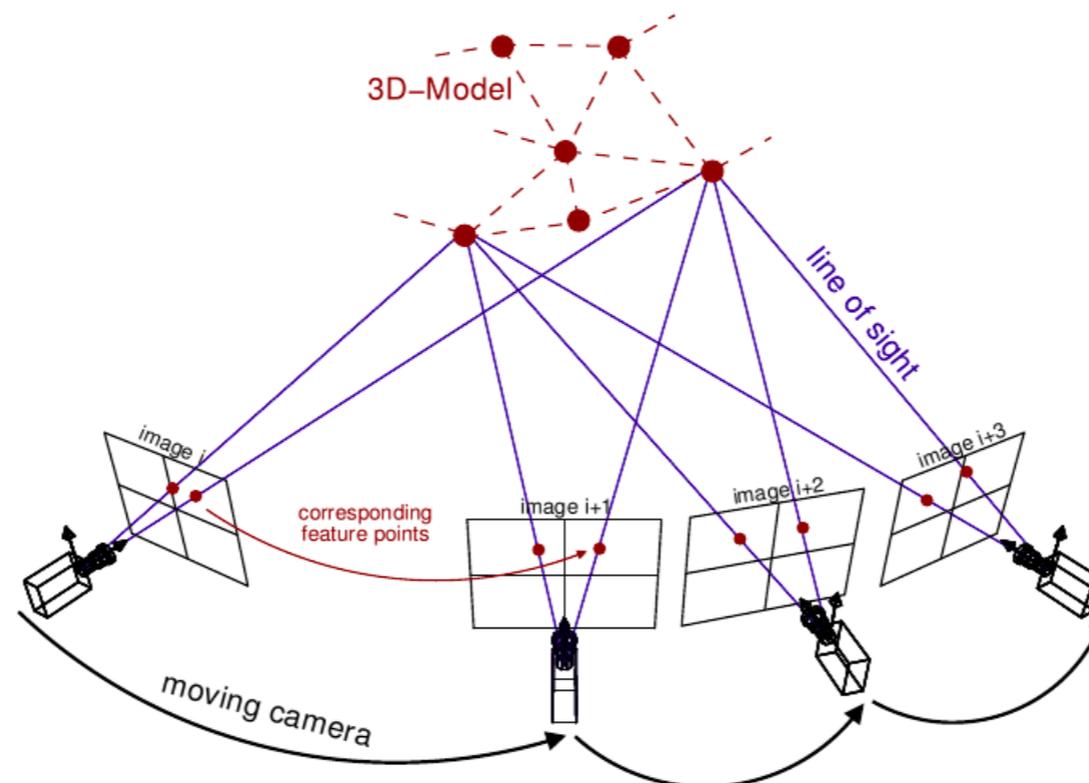
LET'S TALK DETAILS

RE

CONSTRUCTION.

WHAT IS RECONSTRUCTION?

- ▶ Camera calibration and multiple image acquisition.
- ▶ Matching features detection between the images.
- ▶ Image registration (triangulation) and 3d model extraction.



HOW WE ACTUALLY DO IT?

- ▶ Step 1: Capture RGBD frames using RealSense depth camera D435
- ▶ Step 2: Extract local geometric surfaces from images
- ▶ Step 3: Register geometric fragments (loop closure)
- ▶ Step 4: Refine registration
- ▶ Step 5: generate a 3d model for the scene

STEP 1: CAPTURE FRAMES USING REALSENSE DEPTH CAMERA D435

- ▶ RealSense depth camera D435
- ▶ Real sense SDK Python Wrapper
- ▶ Reduce framerate 15frames/s

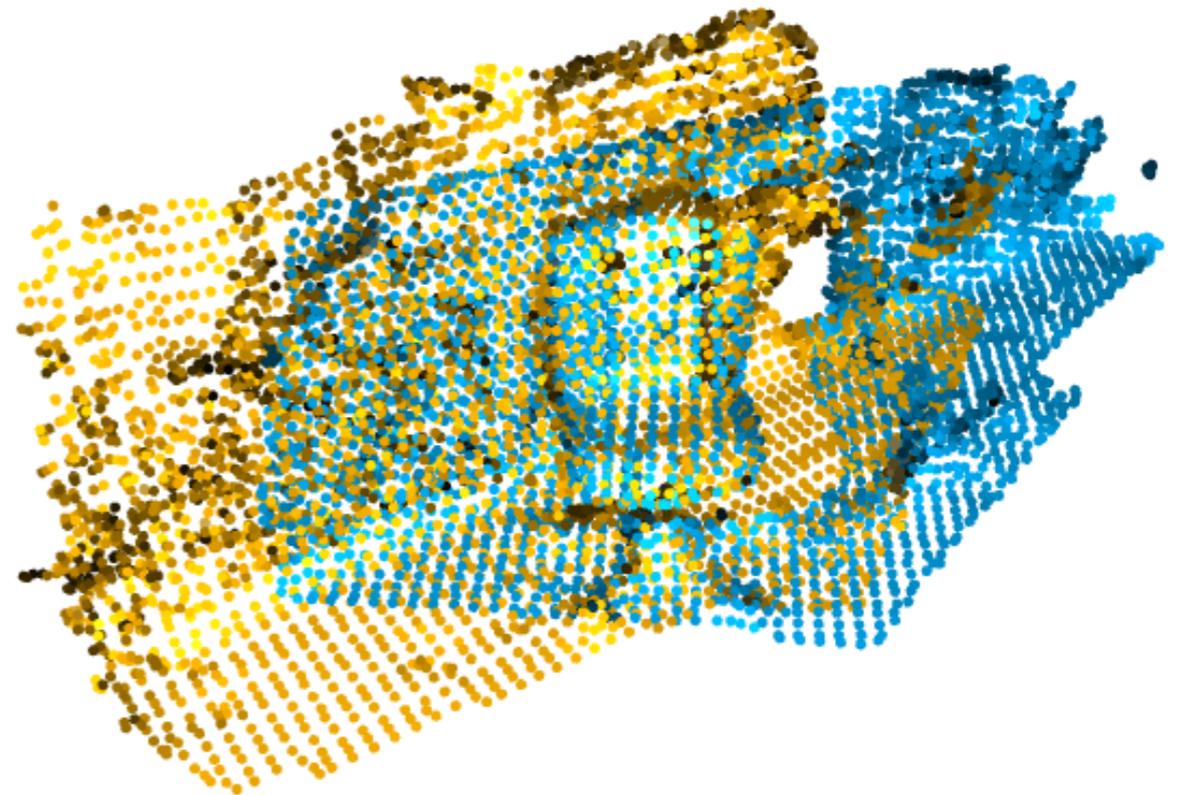
HOW YOU CAPTURE THE FRAMES MATTERS!

STEP 2: CREATE GEOMETRY FROM IMAGE

- ▶ Finds camera movement between two frames
- ▶ Make Posegraph
- ▶ Make Rough Global Alignment of Images
- ▶ Create a Coloured Pointcloud Fragments

STEP 3: REGISTER GEOMETRIC FRAGMENTS (LOOP CLOSURE)

- ▶ Downsample pointclouds
- ▶ Compute rough alignment between two fractions
- ▶ Make Posegraph
- ▶ Compute rough global alignment of all fractions



STEP 4: REFINE REGISTRATION

- ▶ Pairwise registration on the pairs detected by register fragments.
- ▶ Update Posegraph.
- ▶ Compute global alignment of all fragments.



*** INTEGRATE ALL RGBD FRAMES USING THE POSEGRAPH
OBTAINED IN THE PREVIOUS STEP.**



SHOW TIME!

SE

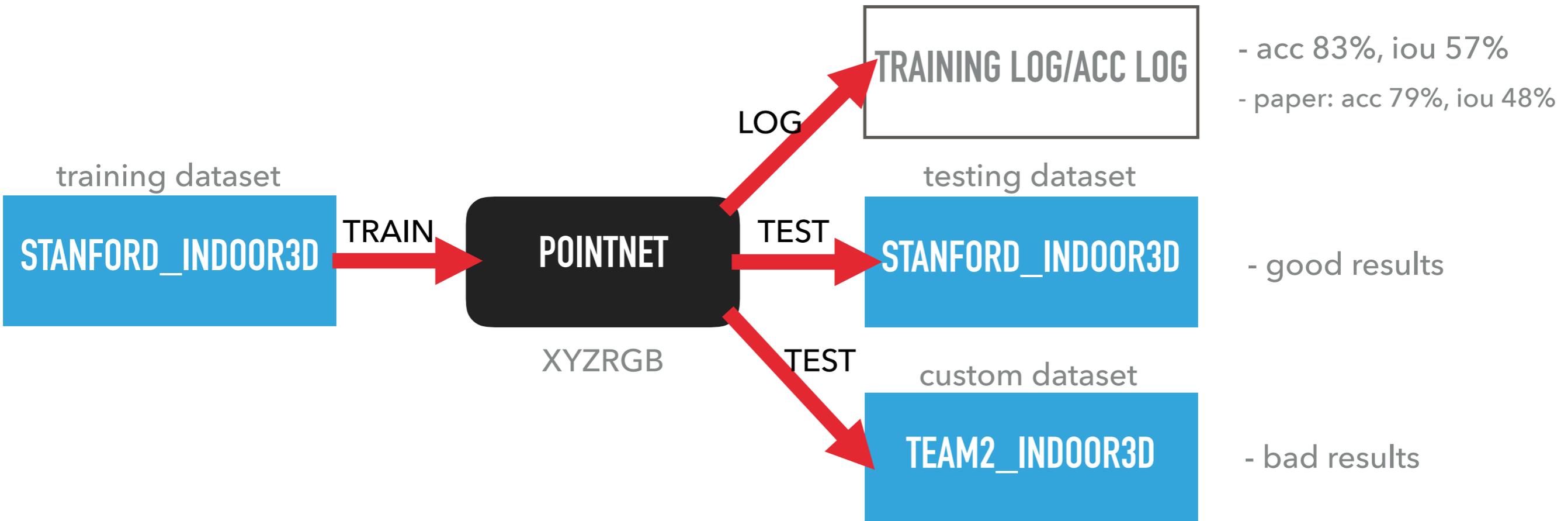
GMENTATION.

COMPONENTS

- ▶ PointNet and PointNet++
 - ▶ very well-known, a lot of resources.
- ▶ ScanNet, Stanford_indoor3D, team2_indoor3D
- ▶ Python, Tensorflow, Open3D

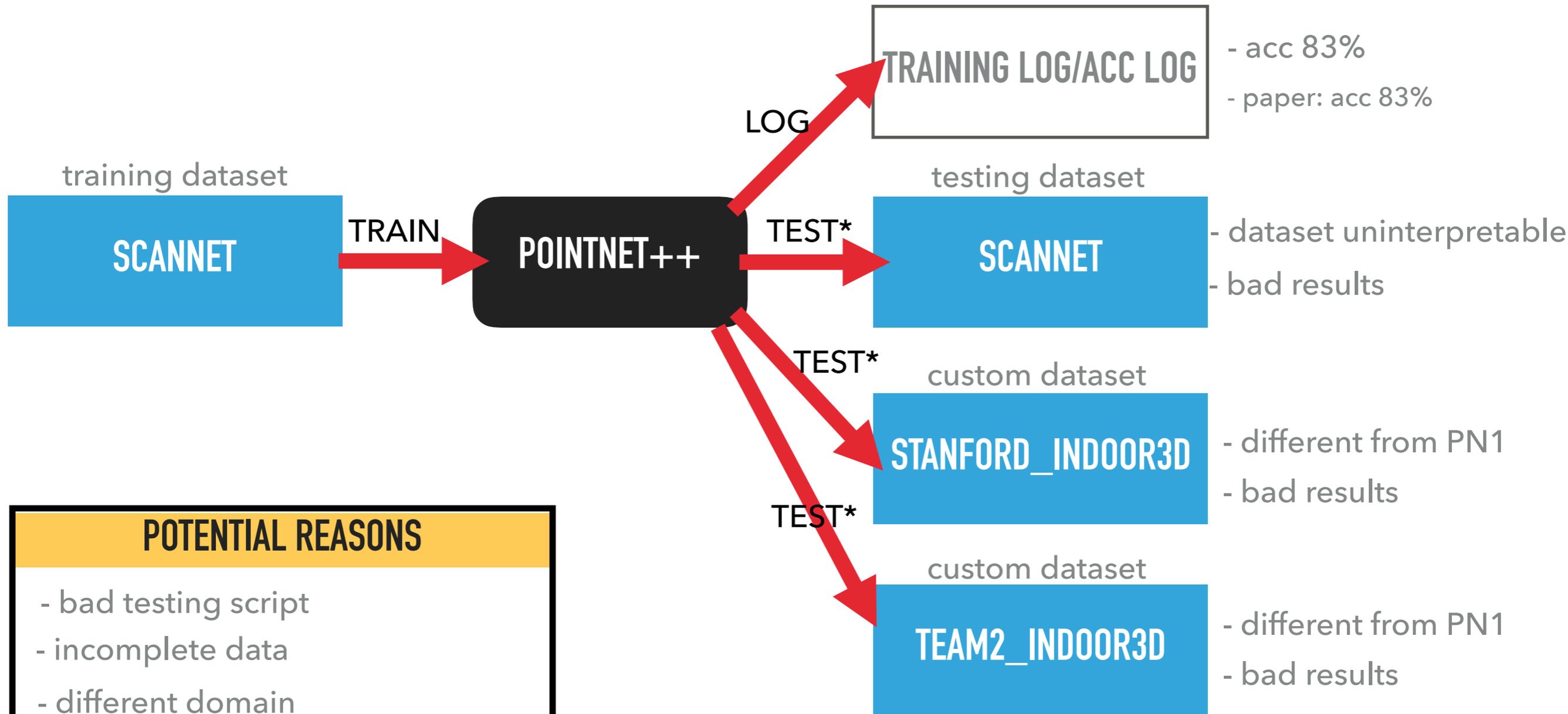


POINTNET



POTENTIAL REASONS
- different domain

POINTNET++

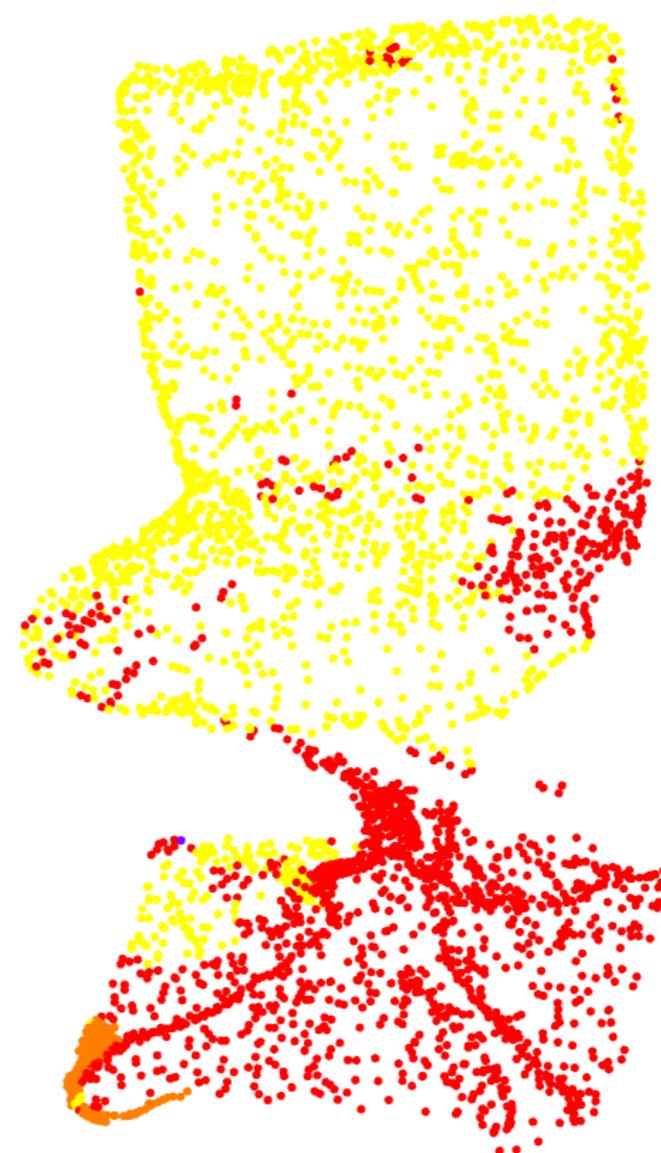


POTENTIAL REASONS

- bad testing script
- incomplete data
- different domain

TAKEAWAYS

- ▶ Verify availability of `train()` and `test()`
 - ▶ check GitHub > repo > issues
- ▶ Try Domain Adaptation
- ▶ Variance in performance
- ▶ DL has a long way to go

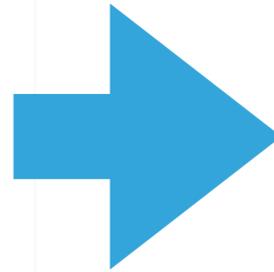
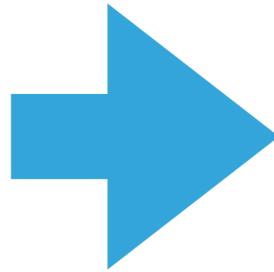


SHOW TIME!

AU

GMENTATION.

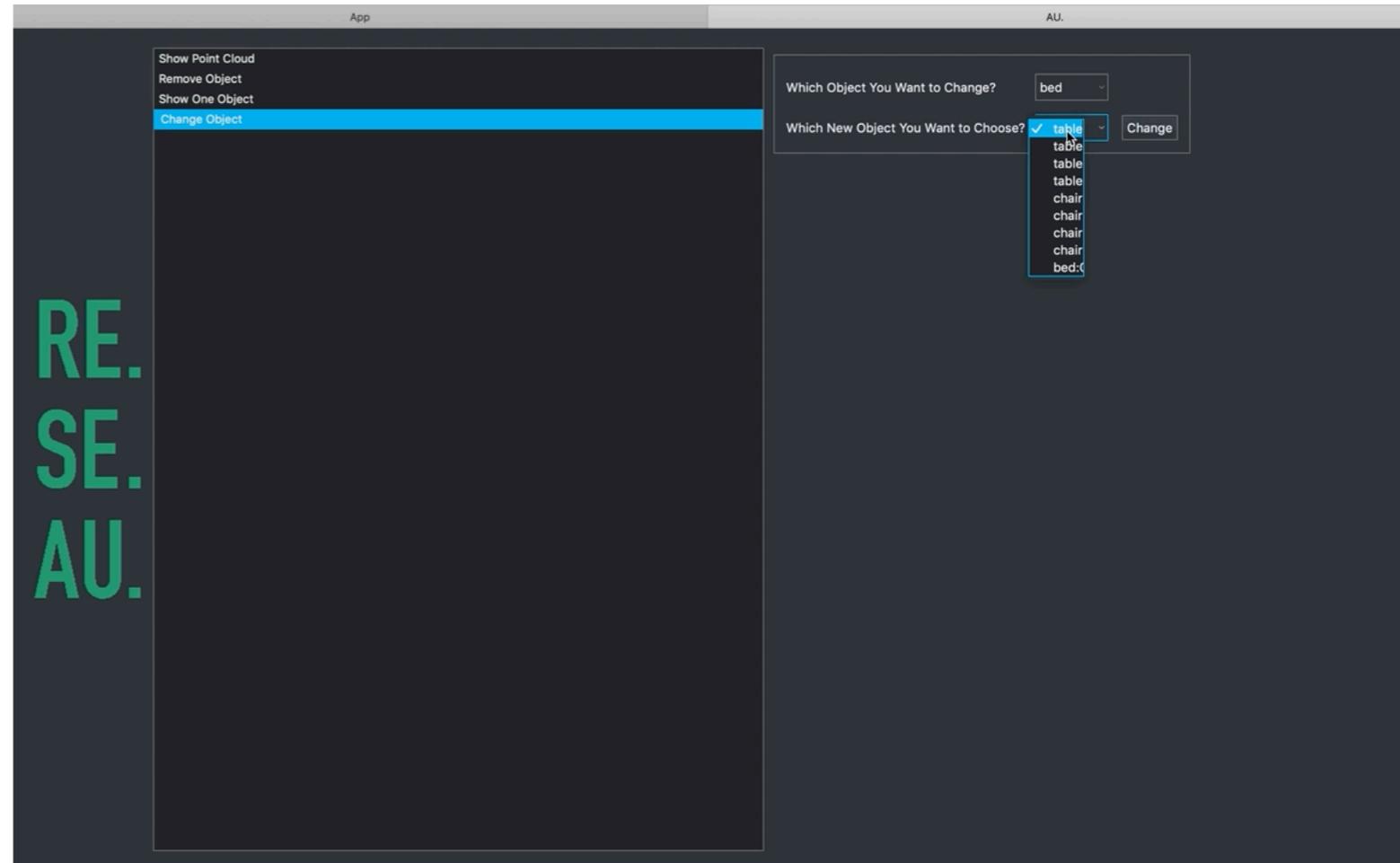
OVERVIEW



Point cloud from Scan-Net

WHAT WE PROVIDE

- ▶ Visualisation of Segmented and Raw Point Cloud of scene.
- ▶ Ability to **Remove** a Segmented Object from scene.
- ▶ Ability to **Change** a Segmented **Object** (Furniture) from the scene, with a new one, with a **good** pose.



TECHNICAL DETAILS

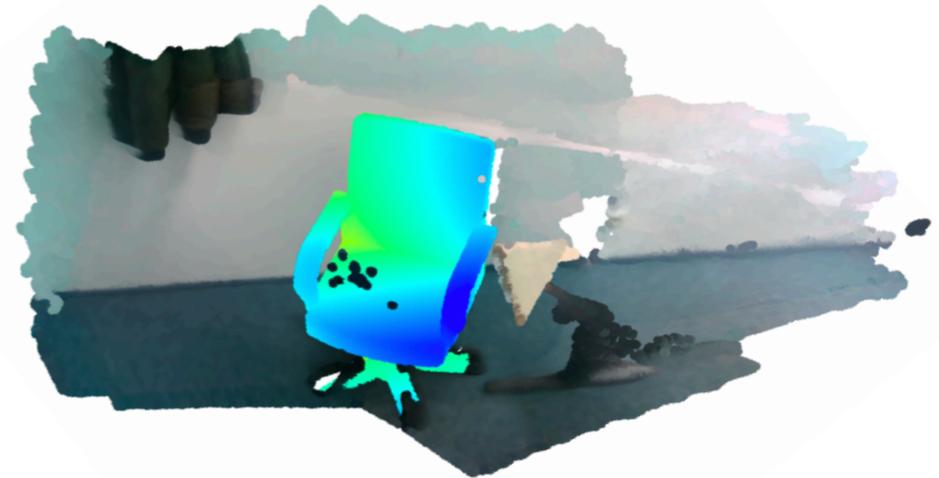
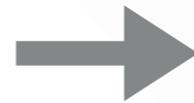
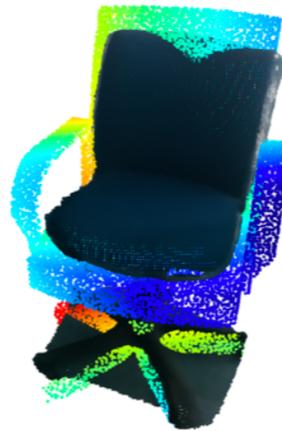
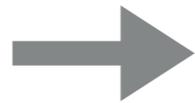
- ▶ When an object is going to be replaced with a new one, we need to find out the POSE of the object.
- ▶ When we have the pose, we modify the new object with the pose, so it will end up in same pose as the old object.
- ▶ But how could we find the pose?

TECHNICAL DETAILS (CONTD.)

- ▶ Step 1: Separate the chosen object from the whole scene.
- ▶ Step 2: Run a Registration between the separated Point Cloud, and the Point Cloud of the new object.
- ▶ Step 3: Refine the estimated Pose with RANSAC.
- ▶ Step 4: Apply the transformation on the new object, remove the old object, add the new transformed object to the scene.

STEPS

*



WE RELY ON THE FACT THAT OBJECTS FROM SAME FAMILY SHARE A SIMILAR TOPOLOGY.

WHICH MEANS: WE STILL COULD NOT CHANGE A CHAIR WITH A TABLE.

SHOW TIME!

USE OF OPEN3D FUNCTIONS BASED ON TWO PAPERS

- ▶ https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Choi_Robust_Reconstruction_of_2015_CVPR_paper.pdf
- ▶ http://openaccess.thecvf.com/content_ICCV_2017/papers/Park_Colored_Point_Cloud_ICCV_2017_paper.pdf

UI IMPLEMENTATION WITH PYQT

- ▶ [PyQT](#)

SEGMENTATION BASED ON POINTNET AND POINTNET++

- ▶ [pointnet and pointnet++](#)

POINT CLOUDS USED DURING IMPLEMENTATION AND ON DEMO

- ▶ [scan-net](#)
- ▶ [RGB-D Scenes Dataset v2](#)



OUR PARTNERS

github.com/shervn/at3dcv2018

QUESTIONS?

THANKS!