Who We Are.

- Replace Customer’s furnitures
- Create new experiences
- Let customer have a better understanding of his living place.
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
RECONSTRUCTION.
3D RECONSTRUCTION

WHAT IS RECONSTRUCTION?

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

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
3D RECONSTRUCTION

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 fractions.
* Integrate all RGBD frames using the posegraph obtained in the previous step.
SHOW TIME!
SEGMENTATION.
Components

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

POINTNET

- training dataset
- testing dataset
- custom dataset

TRAINING LOG/ACC LOG

- acc 83%, iou 57%
- paper: acc 79%, iou 48%

POTENTIAL REASONS

- different domain

- good results
- bad results
SEGMENTATION

POINTNET++

- training dataset
- SCANNET
- TRAIN
- POINTNET++
- TEST*

- testing dataset
- SCANNET
- LOG
- TRAINing LOG/ACC LOG
- - acc 83%
- - paper: acc 83%
- - different from PN1
- - bad results

- custom dataset
- STANFORD_INDOOR3D
- TEST*
- - different from PN1
- - bad results

- custom dataset
- TEAM2_INDOOR3D
- TEST*
- - different from PN1
- - bad results

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.
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?
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.
AUGMENTATION

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!
TOOLS & REFERENCES

USE OF OPEN3D FUNCTIONS BASED ON TWO PAPERS


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
QUESTIONS?
THANKS!