Virtual Cafe

3D Object Recognition By Using Google Tango Project And Creating Virtual World

Members

Virtual Cafe

Wan

Chatchawan Yoojuie

Benz

Natthakul Boonmee

Top

Kanin Kunapermsiri

Virtual Cafe

3D Object Recognition By Using Google Tango Project And Creating Virtual World

Advisor

Dr. Kwankamol Nongpong

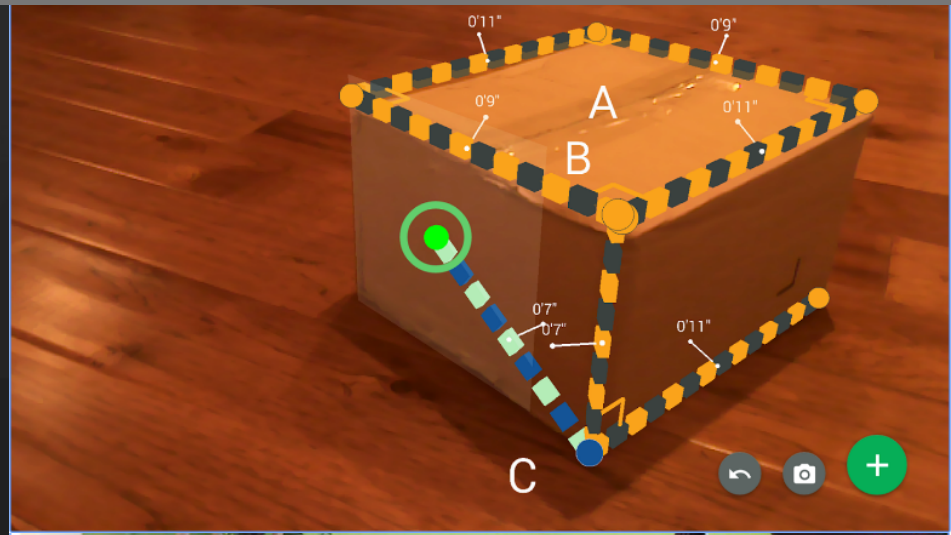
Senior Project

Semester 2/2016

Tango

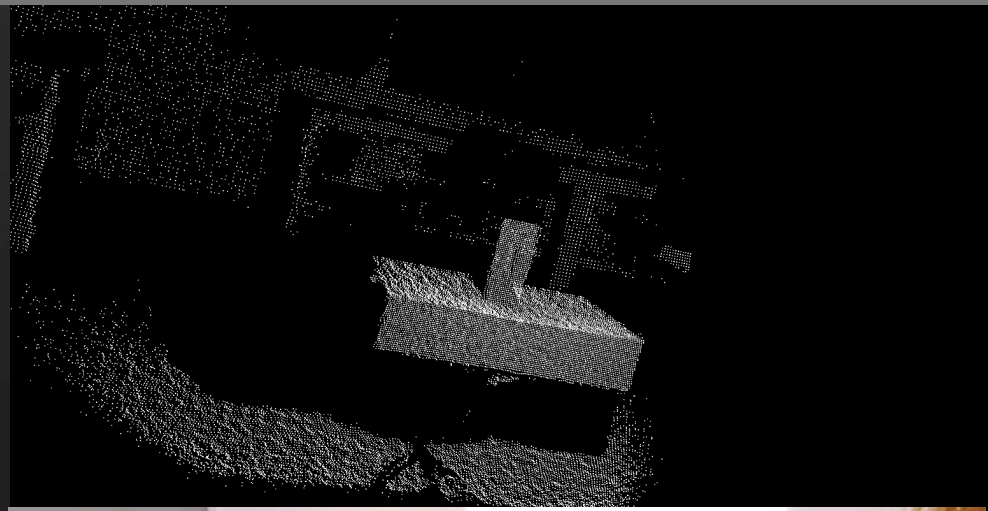
What Tango can do

- Create indoor navigator (without using GPS)
- Create accurate measurement tools
- Create augmented reality game



What Tango cannot do

- 3D object detection
- Create realistic augmented reality application



Goal of Project

- Learn the surrounding environments and transform the physical world into the virtual world

- Recognize the 3D object and display inside the virtual world

- Basic interaction with the objects inside the virtual world

Note: The environments and detected object will be unmovable.

Software

- Google Platform

  for capturing image and position

- Point Cloud Library

  for 3D image processing

- Unity

  for rendering virtual world

  and virtual object

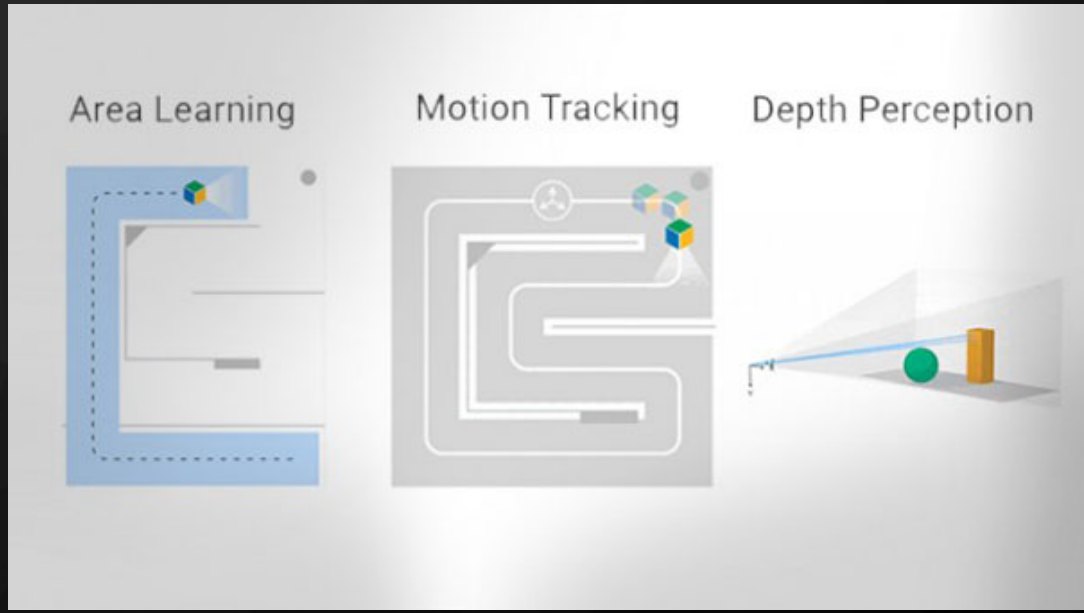
Hardware

- Lenovo Phab 2 Pro

Supported by Google Tango

Google Tango Platform

- It’s a computer vision platform that can do the Area Learning, Motion Tracking, and Depth Perception



Point Cloud Library

- Computer vision library for 3D image which is used for processing the data and recognizing objects using C++

PCL features

Learn more

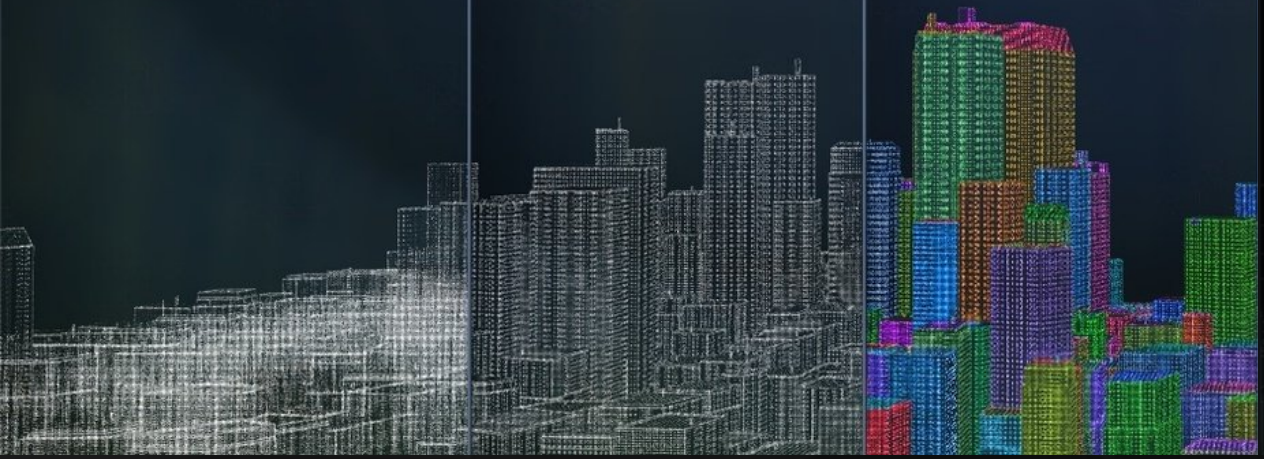
Initial point cloud data

Filtering

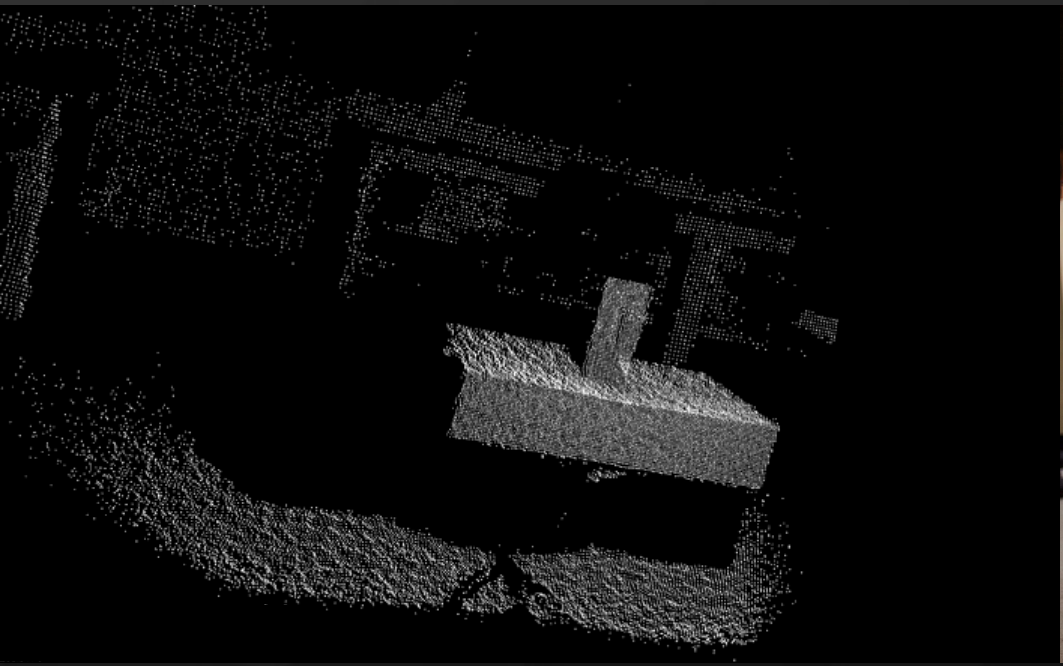
Segmentation

Surface reconstruction

Model fitting



Point Clouds

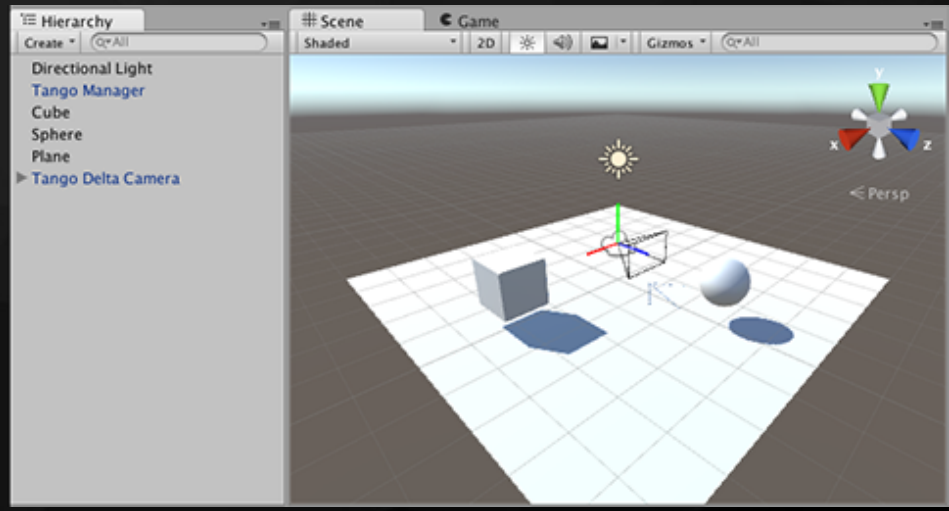


Point cloud is a set of points in the 3D coordinates system

It represents as 3D image and each point in the image contain x, y and z value

Unity

- The game engine which provides all necessary tools
- Coding in C#
- Use for rendering 3D objects and virtual world
- Google provided Tango API for unity



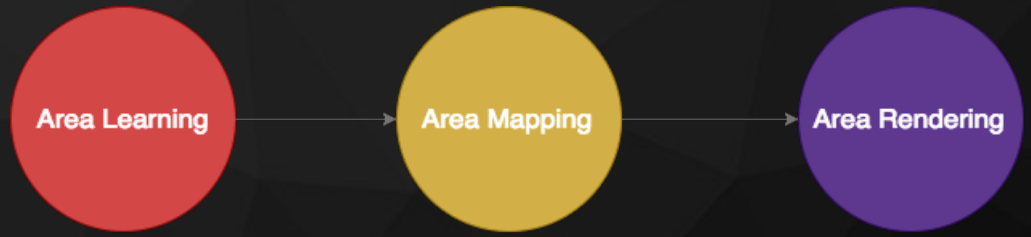
Lenovo Phab 2 Pro

- Android phone that supports Google Tango Platform
- Equipped with IR sensor for capturing point cloud

Framework

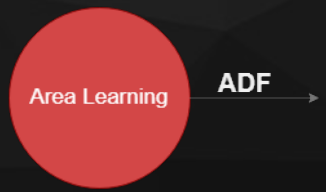
The design framework can be divided into two parts :-

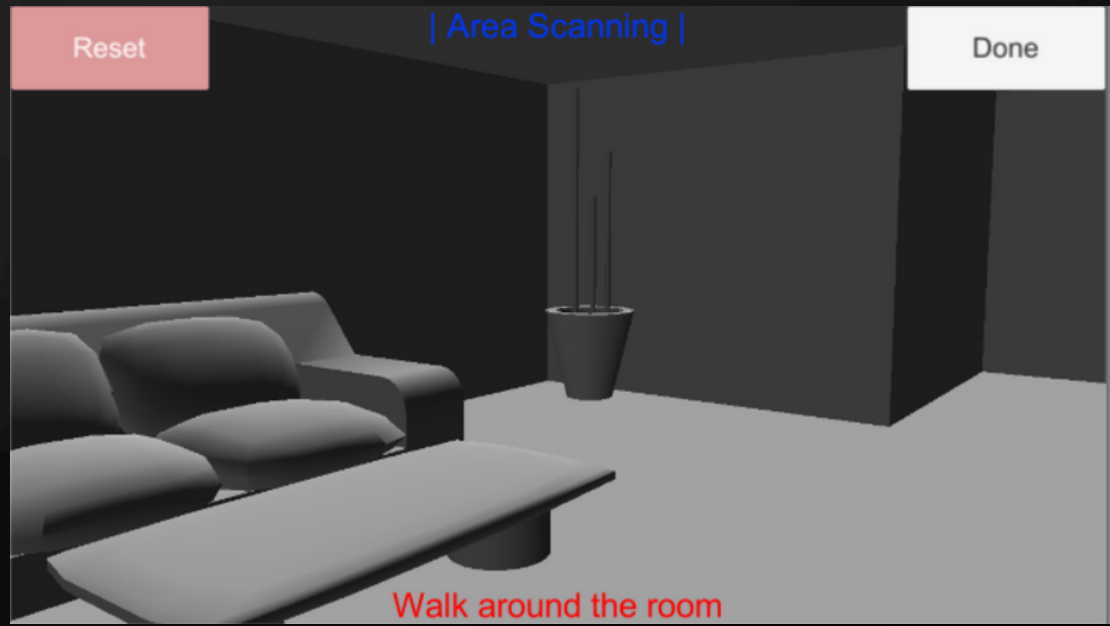
1. Area Mapping (Creating a room)



Area Learning

- Make application remember the room by scanning around the room
- Save into ADF file

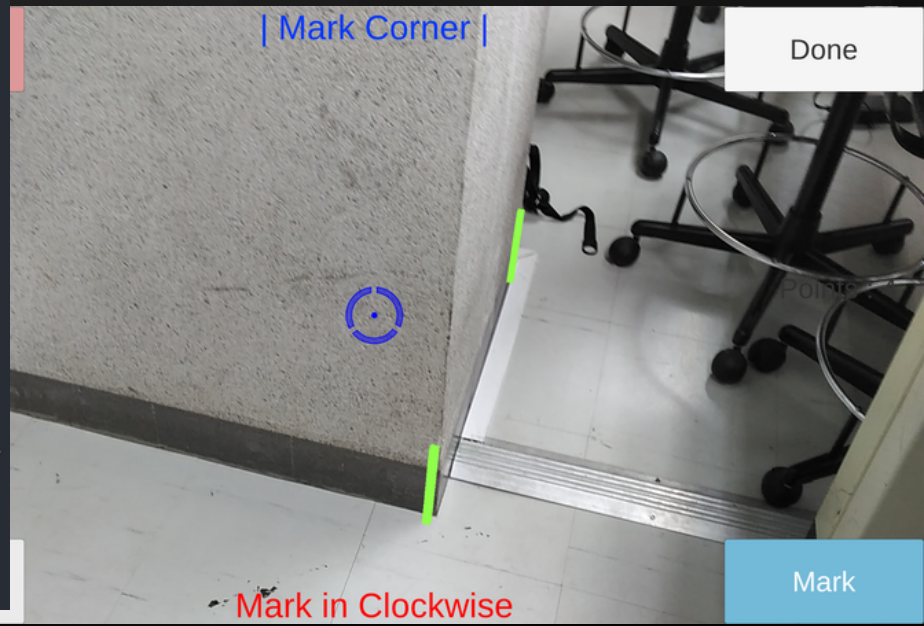




Area Mapping

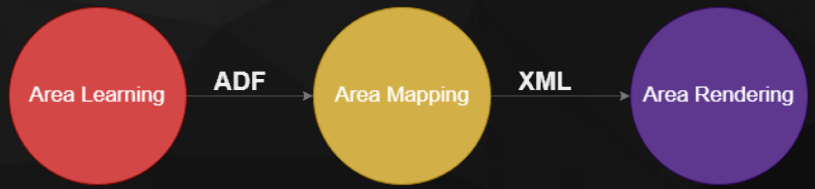
- Measure the actual room size by marking corners and look for distance
- Save into XML file

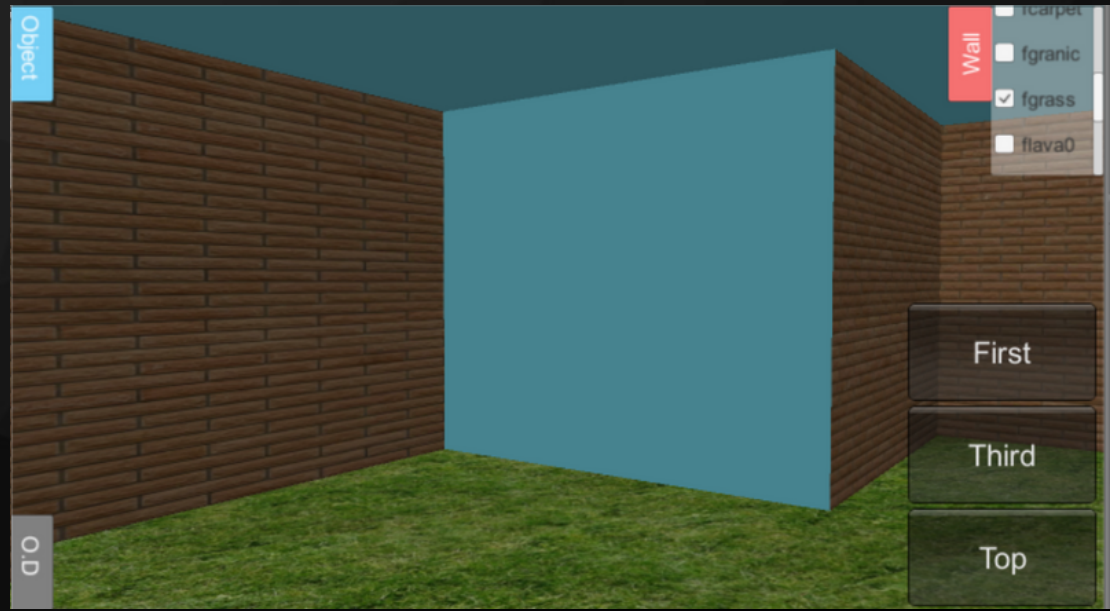

```
<MarkerData>
  <type>-1</type>
  <position>
    <x>1.89826262</x>
    <y>-1.30074608</y>
    <z>1.54177094</z>
  </position>
  <orientation>
    <x>1.03514473E-07</x>
    <y>-1.27317925E-08</y>
    <z>7.157416E-08</z>
    <w>1</w>
    <eulerAngles>
      <x>1.18618855E-05</x>
      <y>-1.45895513E-06</y>
      <z>8.201795E-06</z>
    </eulerAngles>
  </orientation>
</MarkerData>
```



Area Rendering

- Load the ADF file that we saved in area learning part along with the XML file that contains all the vertices representing the corners of the room
- Use Unity to render previous data into virtual room

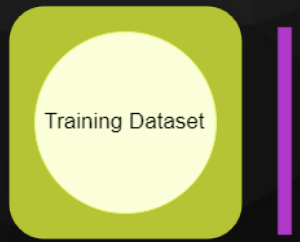




Framework

The design framework can be divided into two parts :-

2. Training Dataset and Object Recognition



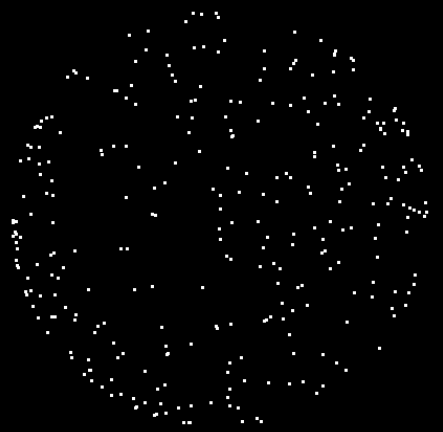
Training Dataset

- Create datasets that will be used in matching step
- Recognize the object with the dataset
- 6DOF pose estimation of the detected object
- Display object model in Unity

What is Centroid?

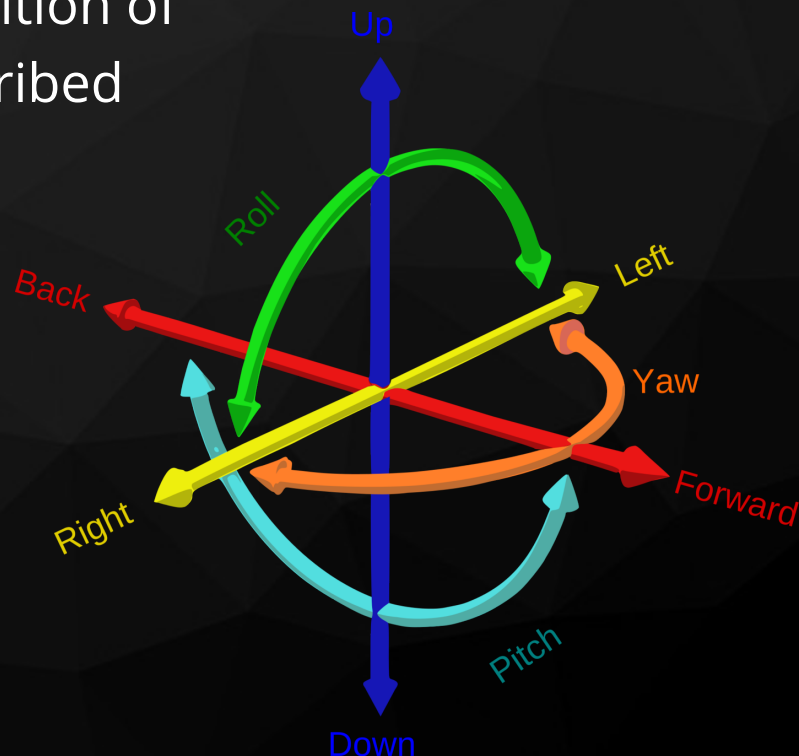
The centroid is a point of the result by calculating the mean value of all points in the cloud

It is a " Center of Mass "



What is 6DoF?

Six degrees of freedom (6DoF) refers to the position of the object in 3D dimension space which is described with translation and rotation



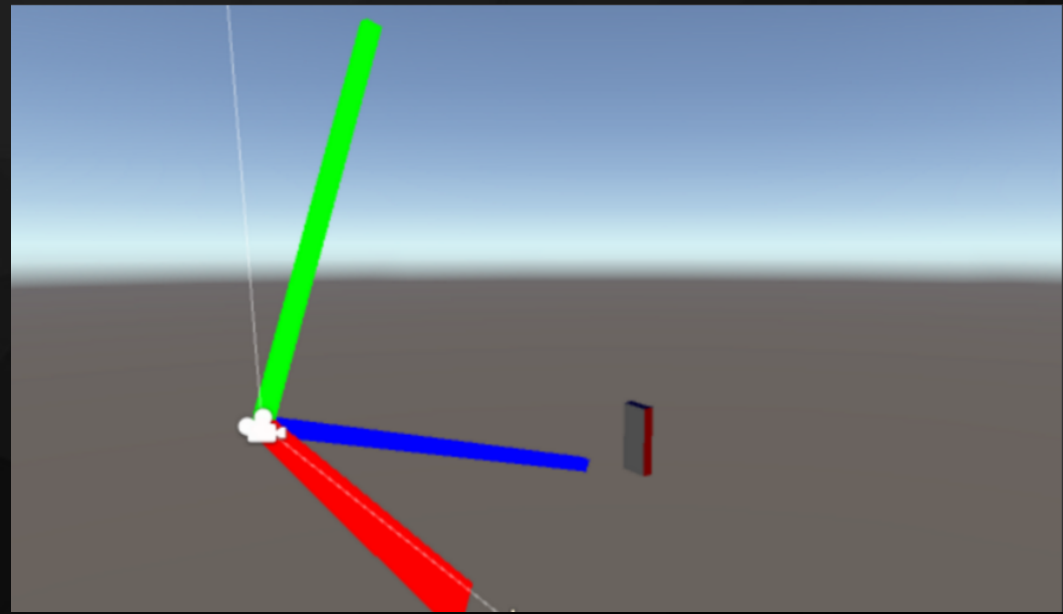
How can we find 6DoF of the object?

With ground-truth information of the object in unity coordinate system

It contains these information :-

1. Translation of the device (Vector format)
2. Rotation of the device (Quaternion format)
3. Translation of the object (Vector format)
4. Rotation of the object (Quaternion format)

```
0     0     0
-0.259     0.001     -0.004
-0.027     -0.082     0.0754
0     0.966     -0.259     0
```

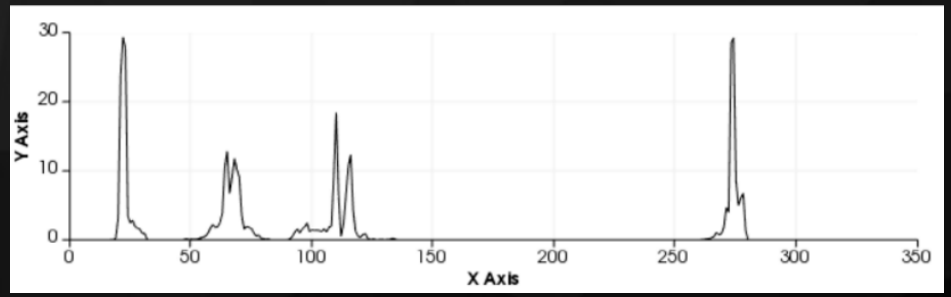


What is descriptor?

Feature extraction that encodes the information about the point cloud

Basically, there are two types of descriptor in PCL :-

1. Local - computed for individual points
2. Global - computed for the whole cluster that represents an object



VFH (Viewpoint Feature Histogram)

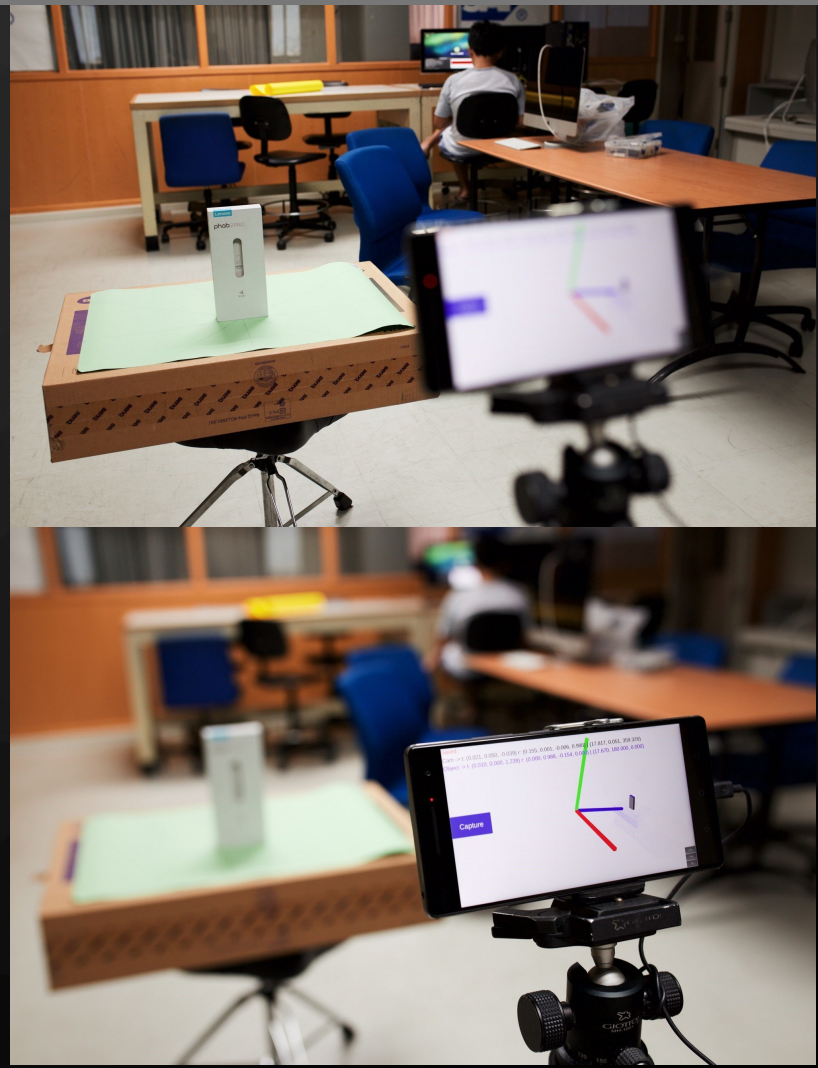
Setting Up

Tripod

Pan-tile

Collecting Dataset

- Capture the snapshots of the object along with ground-truth (pose) information of the object at every 40 degree
- As a result, we have a total of 9 different snapshots
- Then, we use these 9 snapshots as reference frames

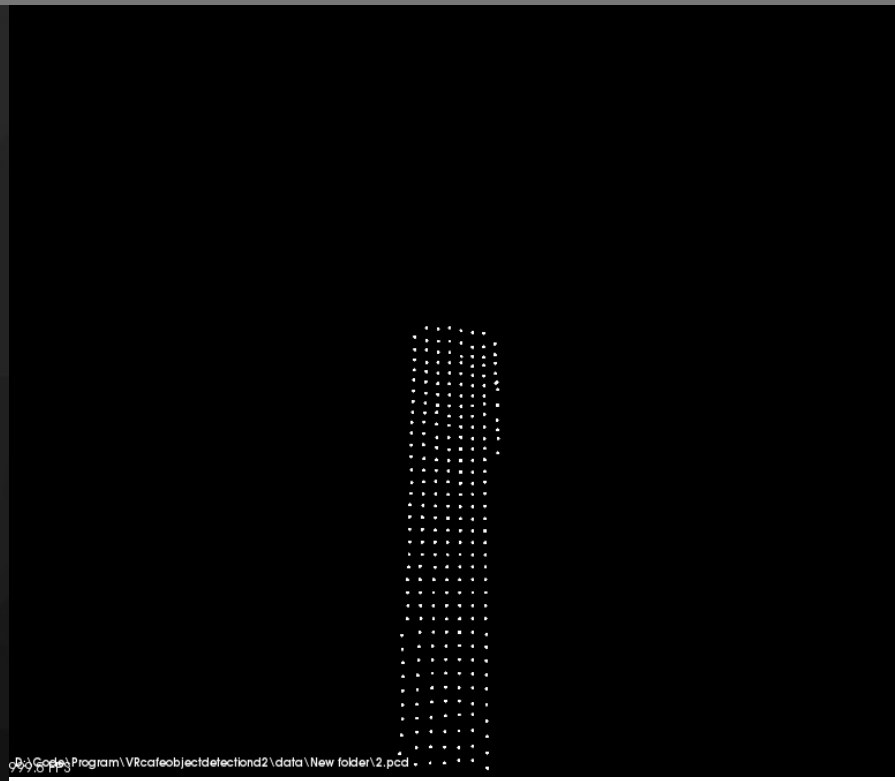


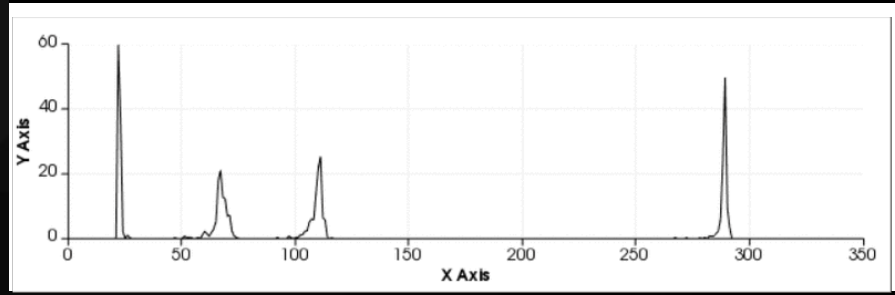
Collecting Dataset

- We can improve and extend the dataset using these reference frame
- Also, the descriptors must be computed for every snapshot in the dataset

Structure of the dataset :-

1. Object Snapshot (.PCD)
2. Descriptor (.PCD)
3. Ground-Truth (.TXT)

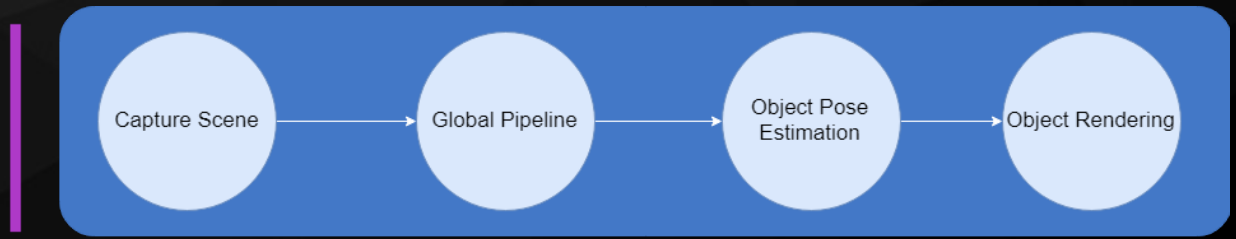




Framework

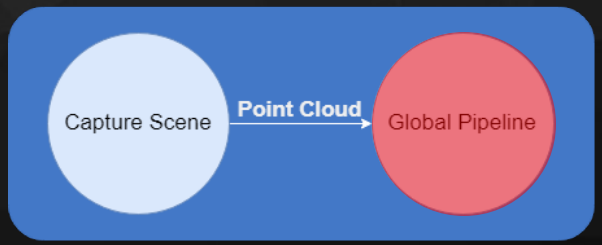
The design framework can be divided into two parts :-

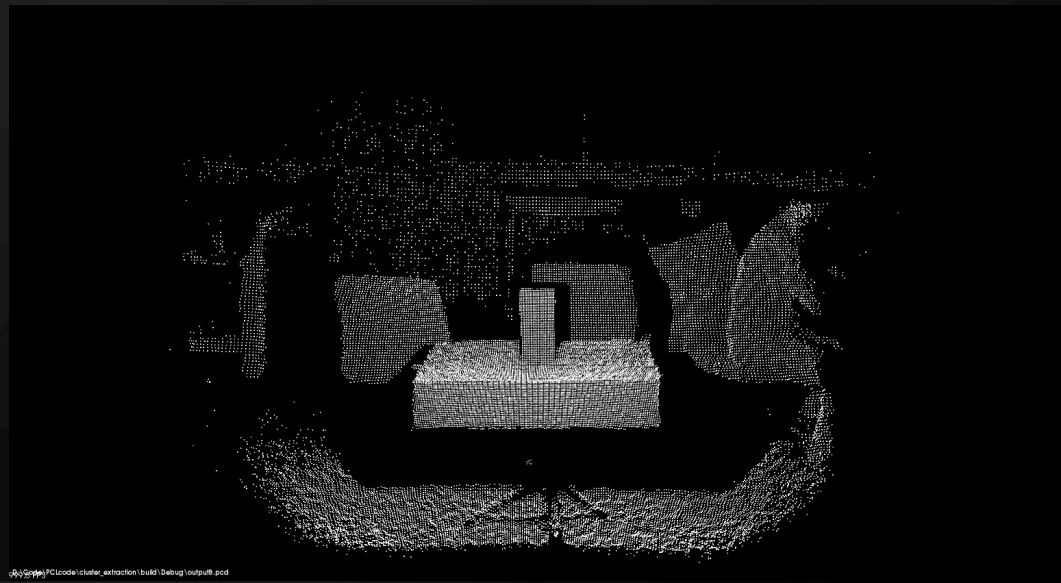
2. Training Dataset and Object Recognition



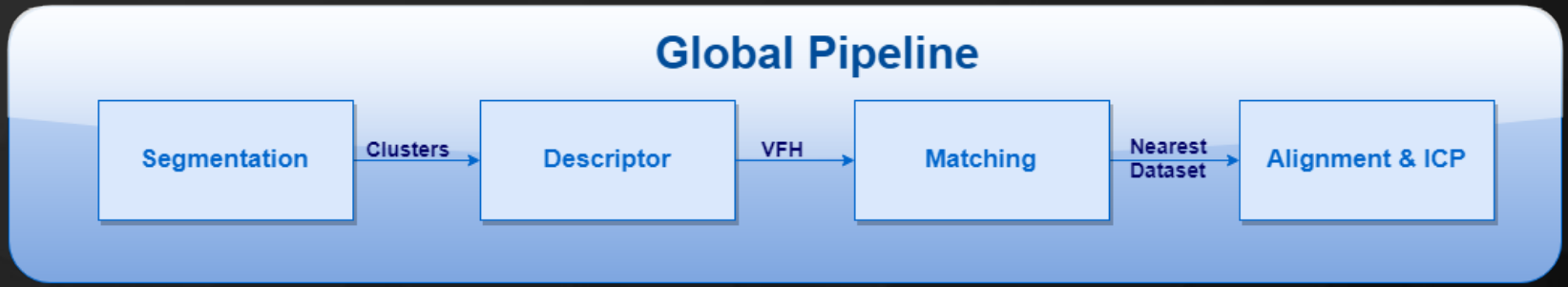
Capturing Scene

- Capture point cloud and send to the server via socket
- Then, follow the process of Global Pipeline





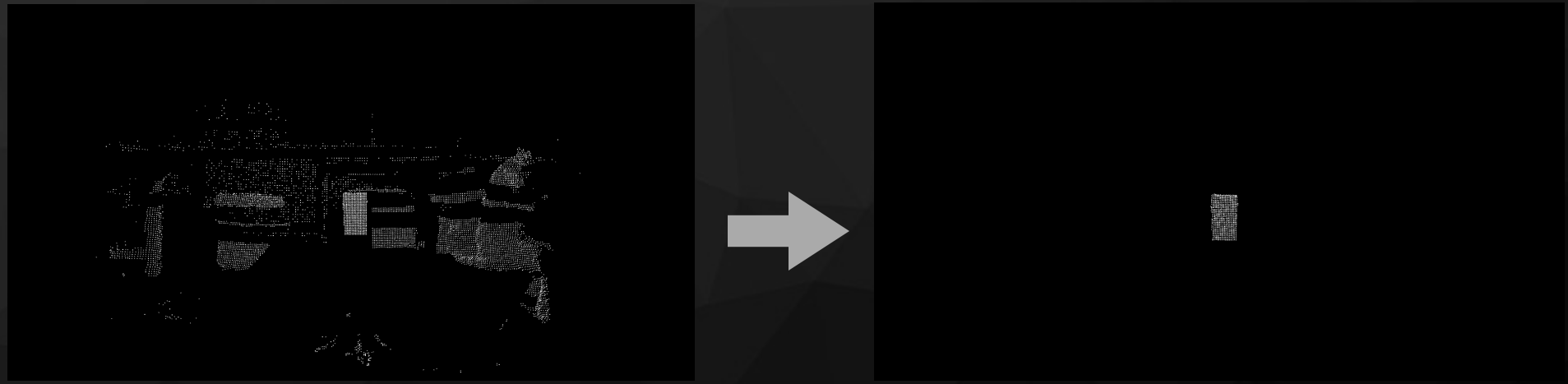
Global Pipeline



The global pipeline contains 4 steps

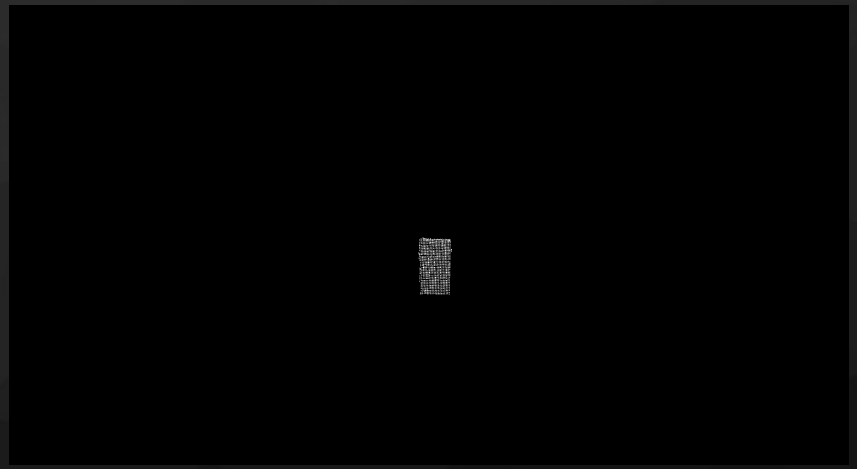
Segmentation

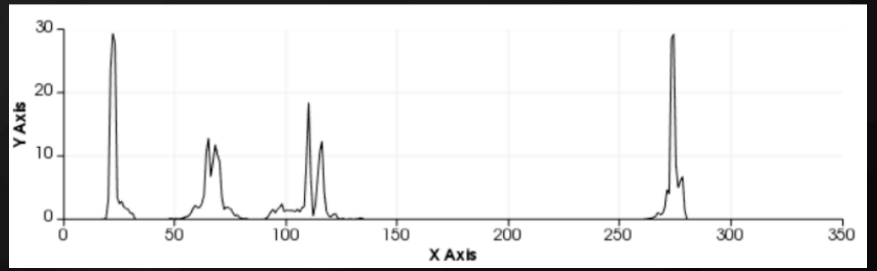
Perform segmentation on the cloud in order to retrieve all possible clusters on the plane surface

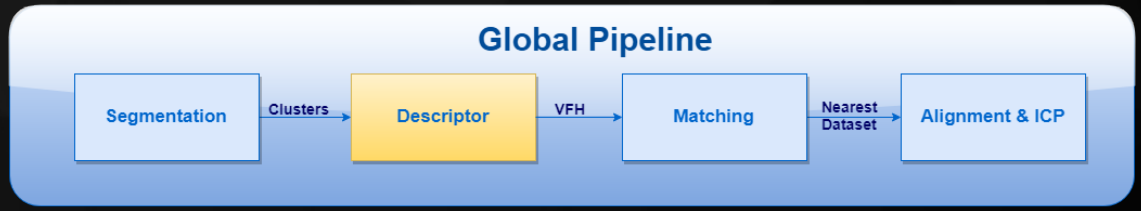


Descriptor

For every cluster that has survived in the segmentation step, a global descriptor must be computed

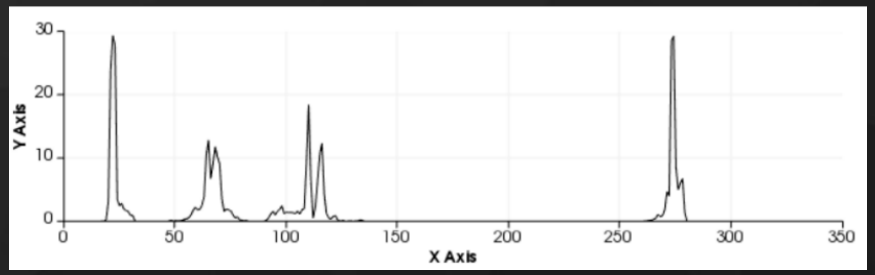


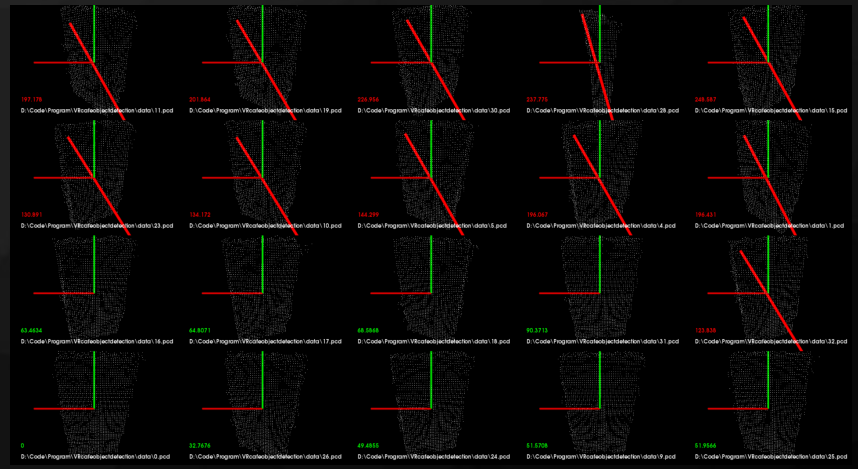


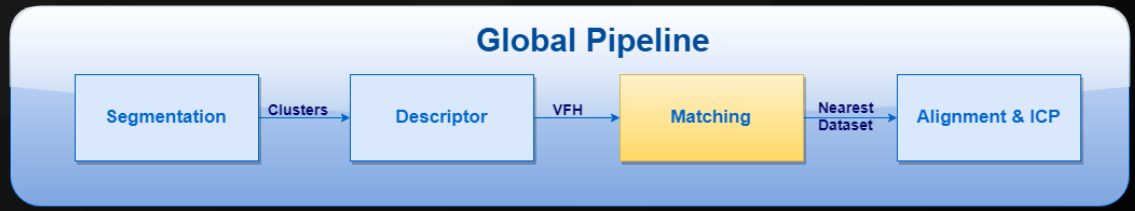


Matching

Use the descriptor to perform a search for their nearest neighbors in the database

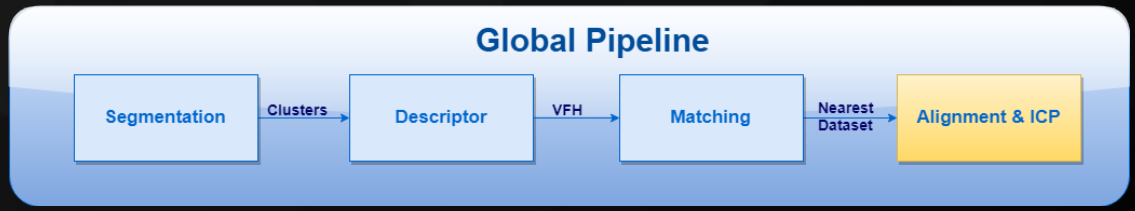




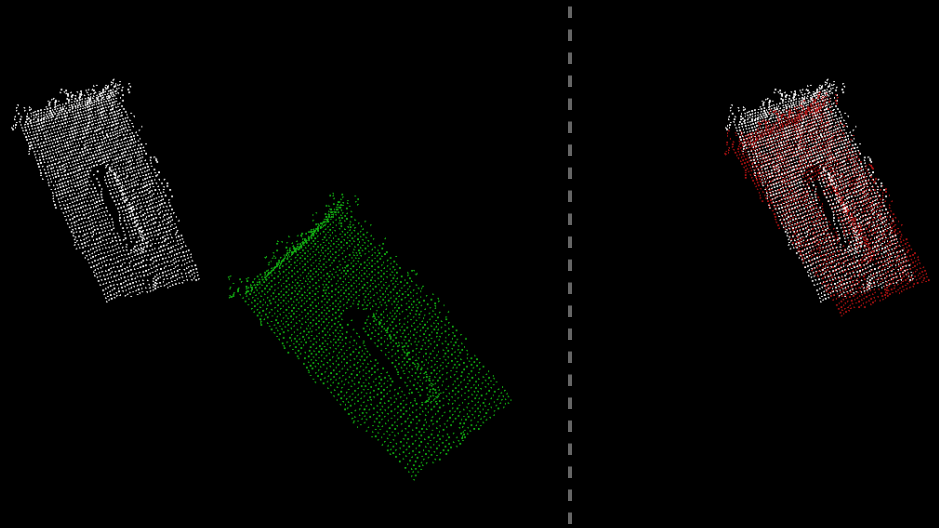


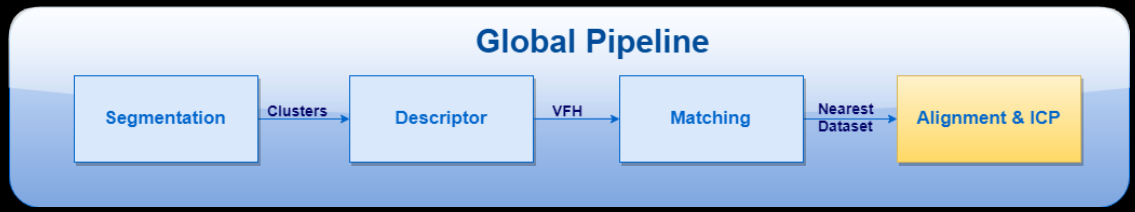
Alignment & ICP

- With ground-truth that saved along with the dataset
- Determine translation of object by computing and aligning the centroids of the clusters
- For the rotation, we can use ICP to compute and find the best transformation from source (given dataset from matching step) to target (current cluster)



Alignment & ICP



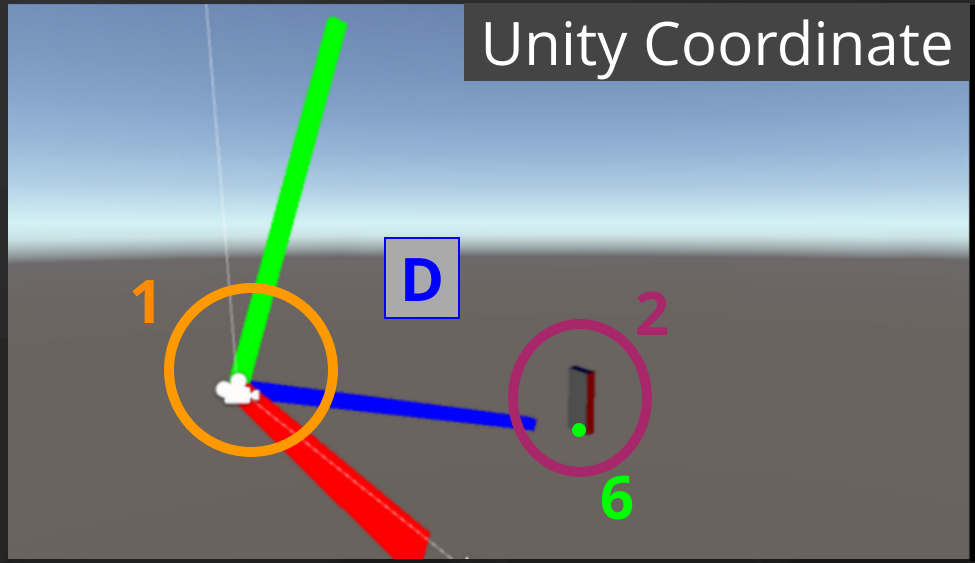


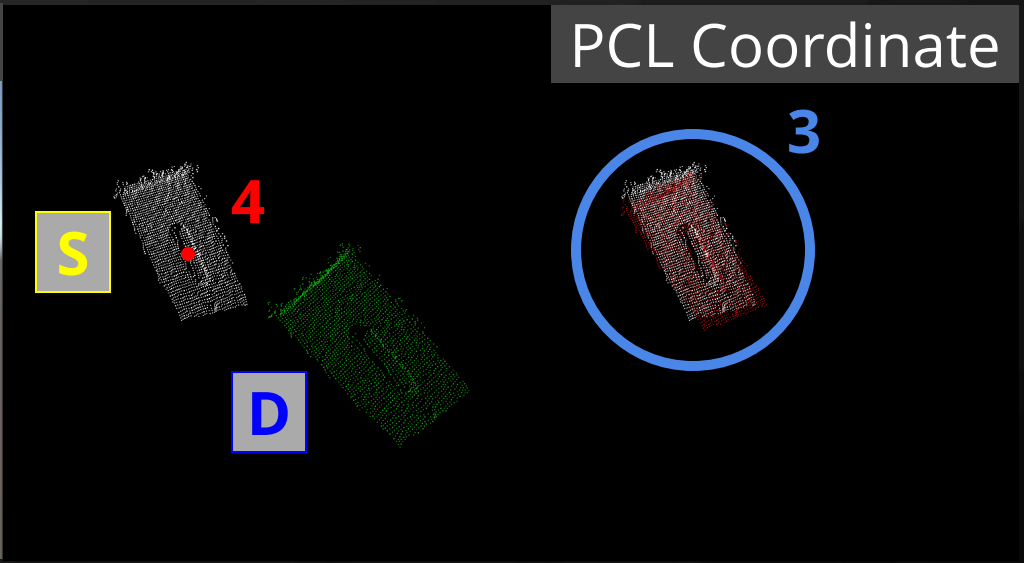
Object Pose Estimation

- The output of the global pipeline will be sent back to the device
- Output is in XML format
- Some calculation needed to extract those pose estimation of the object and display in Unity


```
<BOX>
    <DIS>53.336281</DIS>
    <DR>0.153000,-0.001000,0.002000,0.988000</DR>
    <OR>-0.100000,-0.887000,0.137000,0.429000</OR>
    <ICP>0.078709,-0.958991,0.272317,
         -0.060132,0.268100,0.961522,
         -0.995092,-0.092054,-0.036563</ICP>
    <SC>0.509644,0.283172,1.741762</SC>
    <DCO>-0.012734,-0.109291,0.079274</DCO>
</BOX>
```

Object Pose Estimation



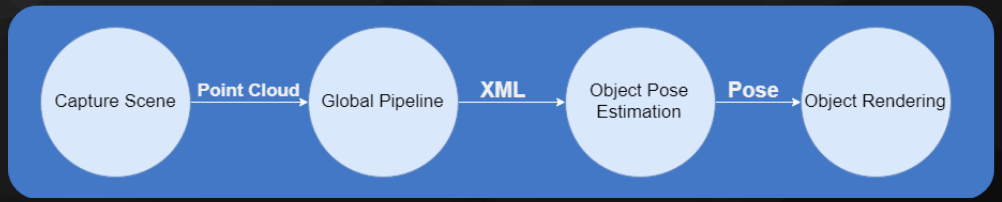


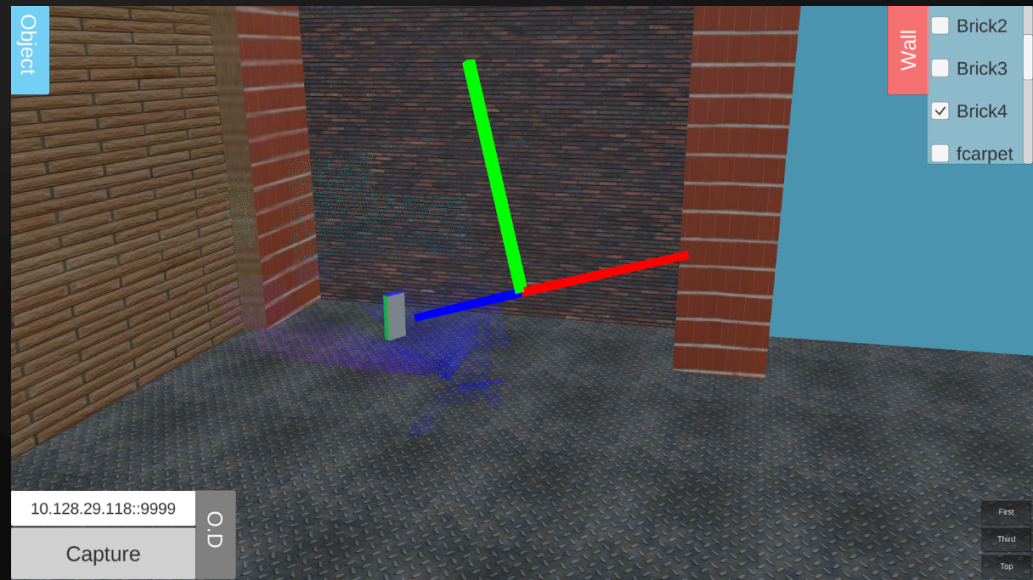
These are 5 pieces of information extracted from the output:

(1) DR = Unity ground-truth of D : device rotation in Quaternion format
(2) OR = Unity ground-truth of D : object rotation in Quaternion format
(3) ICP = ICP from S to D : transformation in Matrix format
(4) SC = Centroid of S in Vector format
(5) DCO = Database centroid offset, the offset of the centroid between SC(4) and Unity ground-truth of D : object centroid(6) in Vector format

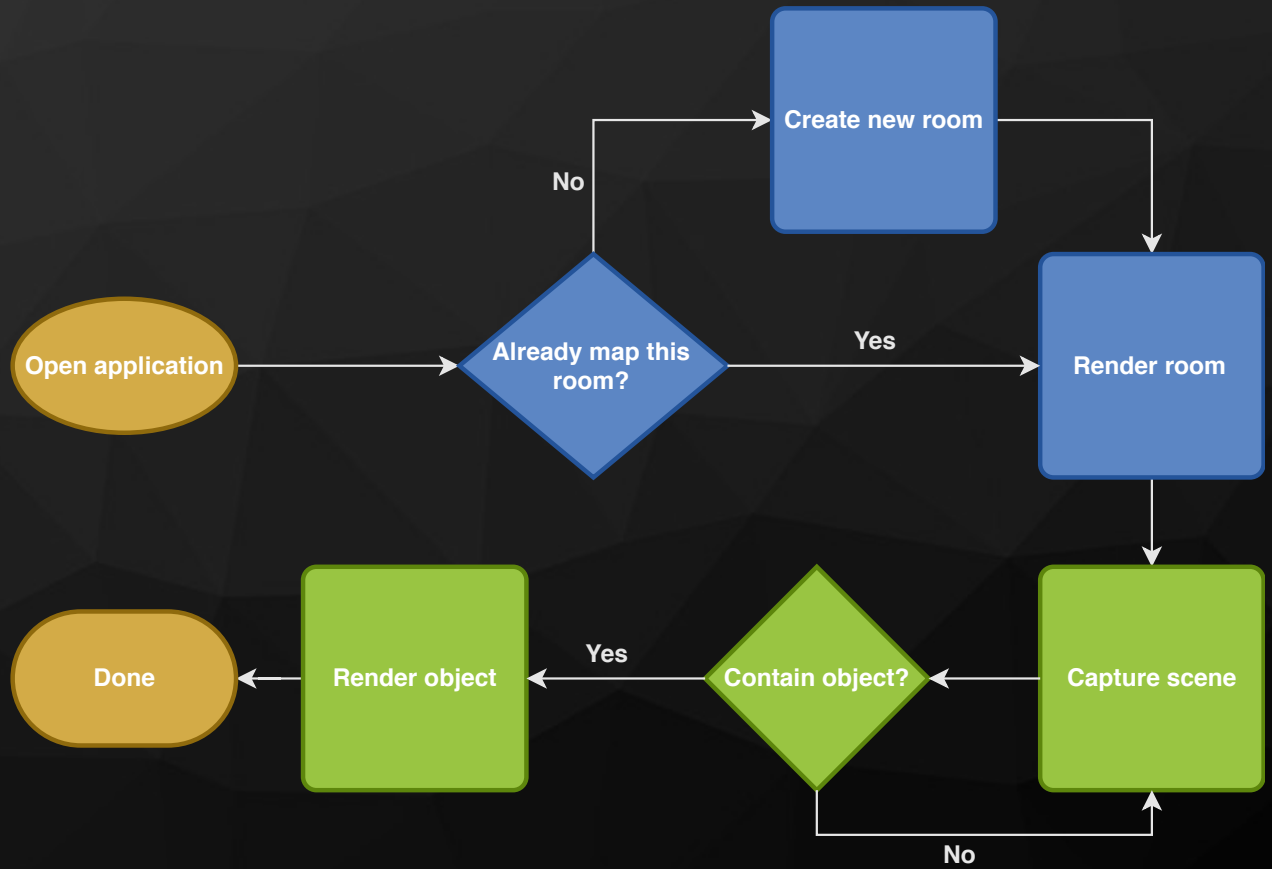
Object Rendering

Use Unity to render the detected object according to the data that extracted from previous





Flow of The Application



Evaluation

- Testing for precision, recall, and f-measure of object recognition
- Testing for how well it can get the correct pose
- Use a single white rectangle box for both testing

Preparing Dataset

Training dataset :

- There are 2 datasets which object is trained
- First dataset has 34 scenes, trained at range 0.9 metre
- Second dataset has 16 scenes, trained at range 1.5 metre

Preparing Dataset

Testing dataset :

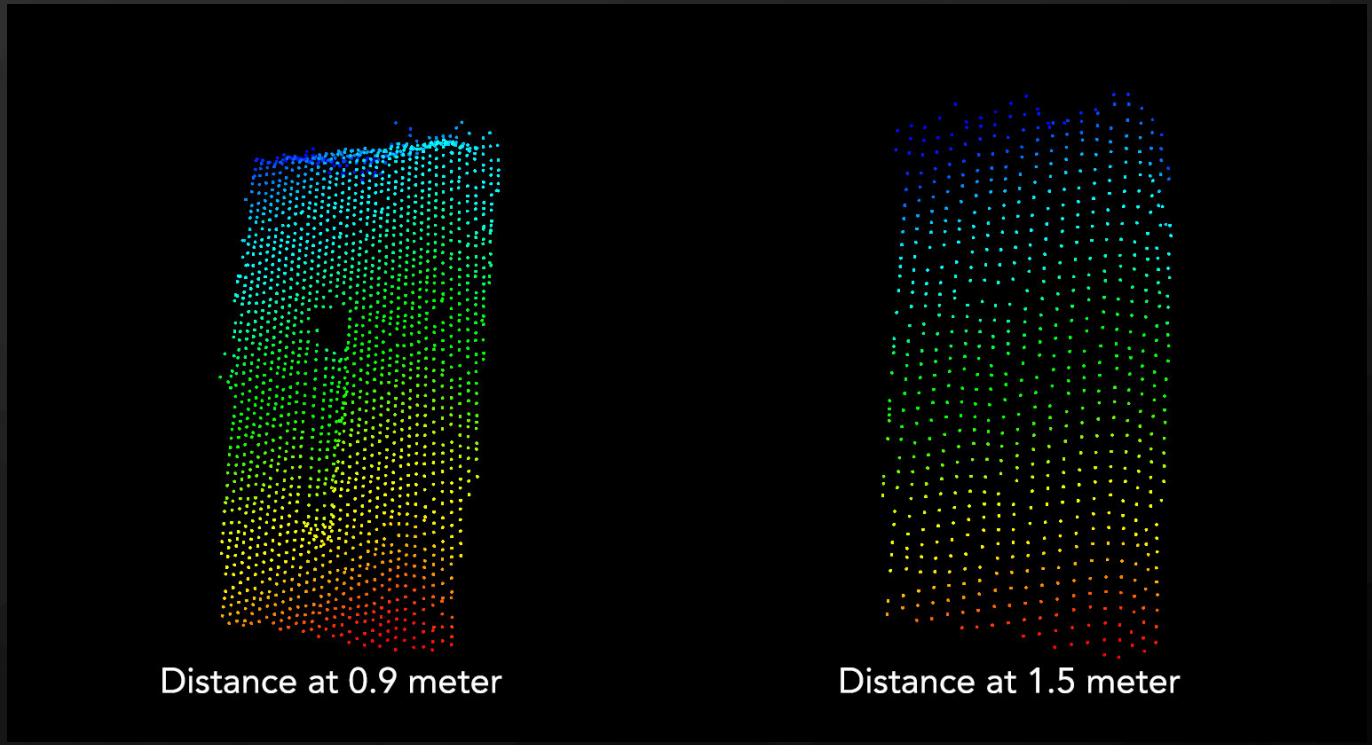
- Use 3 sets which object will be placed at range 0.5, 1.0, and 1.5 metre
- Each distance has 10 scenes that will contain the object at different viewpoint (rotate at every 40 degrees)
- Addition 5 more scenes without the object

Preparing Dataset

Total of tested scenes :

- With Object = 30 scenes
- Without Object = 15 scenes

Sample Point Clouds



Captured At Training

Environment

- Room with no sunlight passing through
- No mirror
- Use tripod to hold the device steady

Result - Detection Accuracy

		Ground Truth		Number of Scenes
		Box	No Box	
Test	Box	24	6	30
	No Box	2	13	15
Calculation	Precision	80.00%		
	Recall	92.08%		
	F-Measure	85.12%		

Dataset 1 At 0.9 metre

		Ground Truth		Number of Scenes
		Box	No Box	
Test	Box	16	14	30
	No Box	14	1	15
Calculation	Precision	53.33%		
	Recall	53.33%		
	F-Measure	53.33%		

Dataset 2 At 1.5 metre

- The performance from dataset 2 is significantly drops compared to dataset 1
- The threshold value of the matching is too large
- The quality and detail of the point cloud changed according to distance

Result - Pose Estimation

	Distance of box									Total Scenes
	At 0.5 metre			At 1.0 metre			At 1.5 metre			
	✓	x	F-measure	✓	x	F-measure	✓	x	F-measure	
Dataset 1 (at 0.9 m)	6	3	66.67%	7	2	77.78%	3	3	50.00%	24
Dataset 2 (at 1.5 m)	1	0	50.00%	4	4	50.00%	5	2	71.43%	16

- Distance of the object at the training stage have a huge impact on the accuracy of the recognition system
- At 0.5 metre is slight lower performance than at 1.0 metre.

DEMO

Challenges

- Limited access to Tango API
- Difficult to control variable
- The sensor is poor, so the distance can affect the details of the point cloud
- Compile Point Cloud Library for android
- Small community
- Less example
- Less guideline

Improvement

- Training dataset can be improved by using pan-tilt that can be rotated almost at all the angles i.e., x, y and z rotation
- Do some research on how to improve the quality of the point cloud

- Q&A -

Thank you