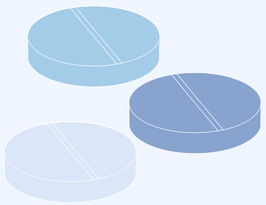
Presented by: Jeff Lin, Zikang Liu,
Yuya Yuan, Delun Zhang

MITI Research Group

CAOCM-Auscultation



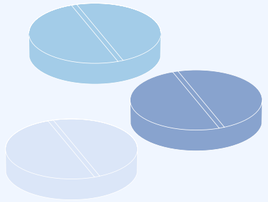
Estimating Surface Normals in a PointCloud

Option 1: Open3d

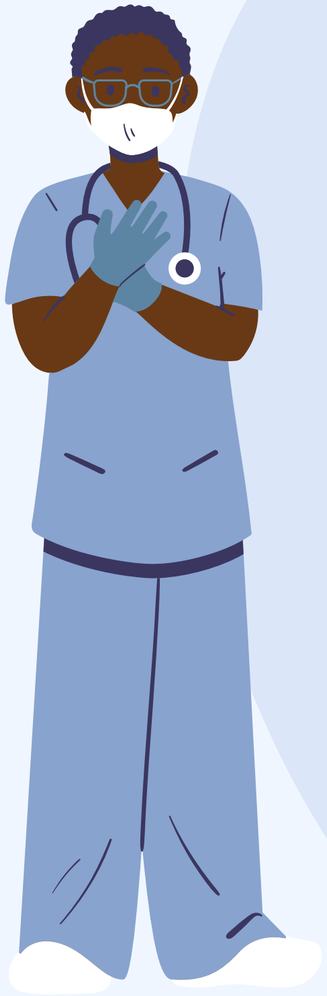
Open3d is a standalone open-source library designed for 3D data processing.
It is very popular because of python-friendly APIs.

Option 2: Pclpy

Pclpy is a python binding point cloud library. It is written in C++ and already have a lot of useful tools. However, it is also more complicated to use.



Smoothing and normal estimation based on polynomial reconstruction

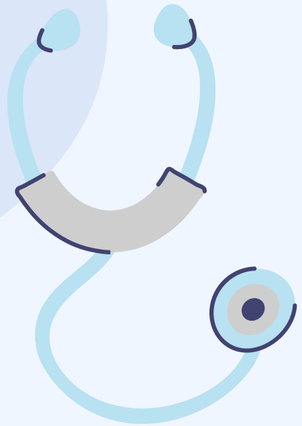


Using Pclpy library's function to implement this algorithm.
There are lots of factors that we can tune after we impelement our code on robotic arms. For example, how big the radius should be to determine the surface area.

Normal Estimation Using Integral Images

```
(cacom) D:\cacom-auscultation>python method3.py
[pcl::PLYReader] pointcloud\over1.ply:15: property 'list uint8 int32 vertex_indices' of element 'face' is not handled
[pcl::IntegralImageNormalEstimation::setInputCloud] Input dataset is not organized (height = 1).
[pcl::IntegralImageNormalEstimation::initCompute] Input dataset is not organized (height = 1).
```

Based on the error message, we can know that the format of our ply file isn't organized. Hence, we can not implement the Integral Image method.



VISUALIZE NORMAL VECTOR (Open3d)

We add functions to plot the calculated normal vector. Based on the image, the peformance looks good!!
(Becasue it's a 3D gif, I will run the code to show the result.)

VISUALIZE NORMAL VECTOR (Pclpy)

Becuase this is a C++ library, so I try to turn the C++ code I find it online into python code. However, in this visualization part, it fails.

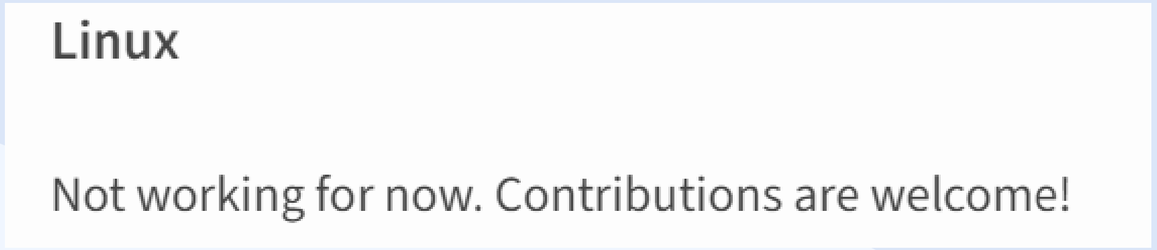
```
pcl::concatenateFields (*original_XYZ_cloud, *normals, *cloud_with_normals);
```

```
pcl.concatenateFields(cloud, normals, cloud_with_normals)
```

```
(cacom) D:\cacom-auscultation>python pclpy_normal.py
[pcl::PLYReader] pointcloud/over1.ply:15: property 'list uint8 int32 vertex_indices' of element 'face' is not handled
Traceback (most recent call last):
  File "pclpy_normal.py", line 24, in <module>
    pcl.concatenateFields(cloud, normals, cloud_with_normals)
AttributeError: module 'pclpy.pcl' has no attribute 'concatenateFields'
```

Laboratory

1. Know more about ros2
2. There are still some trivial problems when merging our algorithms into the actual working code.
3. Linux can't run the pclpy (probably we have to use C++ to run it)

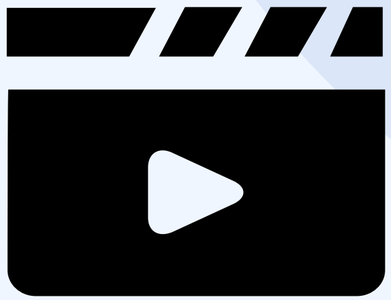


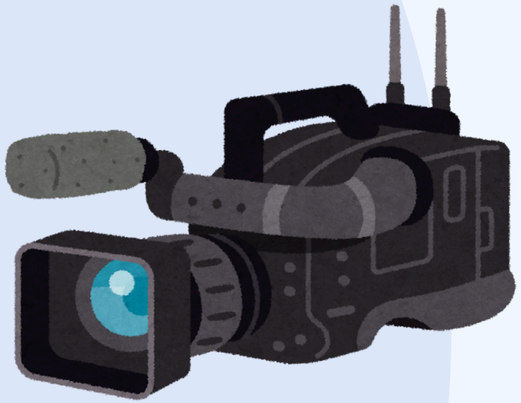
Conclusion

Due to time limitation, I decided to use Open3d to implement this project.
And use the parameters inside to tune its performances.
If we still have time, we can try pclpy in C++.

Video outline (7-8 MIN)

1. Introduction
2. Motivation - what're the uses of robotic arms
3. Our focus - what we want to achieve
4. Pipeline - how does the arm work
5. Surface Normal Evaluation - introduce the theory
6. Technical details
    a. local research outcome
    b. testing, data collection
    c. testing data analysis
7. Hardware setup
8. Demo - show how the robotic arms works with video
9. Limitations
10. Reference





Thank you for your attention

Notes & Advice