CAoCM

Semiautonomous Telerobotic Auscultation with Surface Normal Estimation

Presenters: Delun Zhang
Jeff Lin
Yuya Yuan
Zikang Liu

Motivation

1. Care professions shortage -> alternative: semiautonomous robotic arm
2. Perform remote diagnostic procedures precisly from arbitrary locations
3. At patient side: intergrate multi-sensors on robot to ensure and improve patient safety
4. At clinician side: share diagnostic workflow with external experts

Achievements

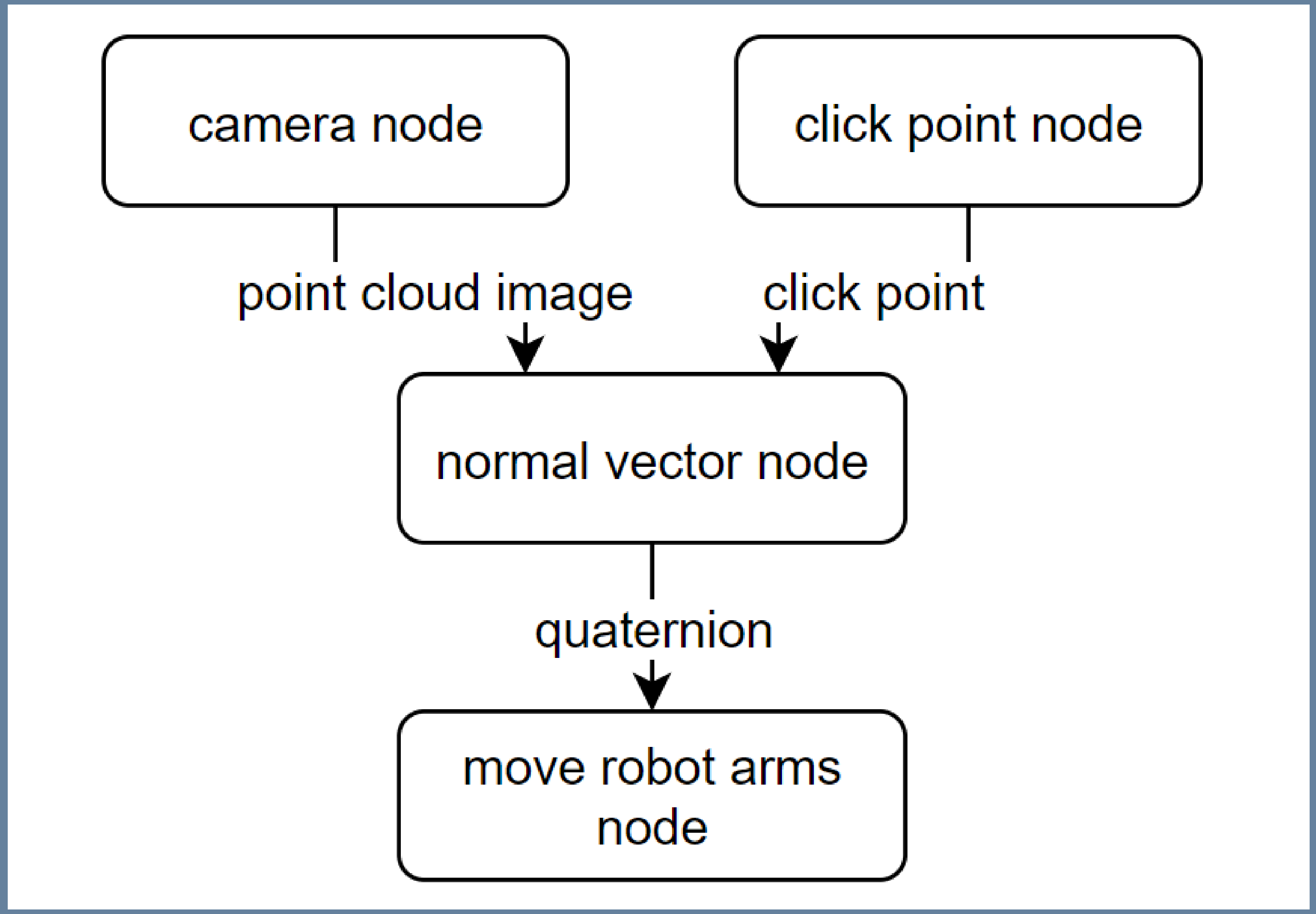
1. **Develop the robotic diagnostic system integrated with auscultation**
2. **Investigate the accuracy of surface normal estimation methods**

Workflow

1. **Use a sticker to mark a point on patient**
2. **Click a target point on the computer interface**
3. **Calculate the normal vector of the point**
4. **Robot places stethoscope perpendicular to the desired point.**

Pipeline

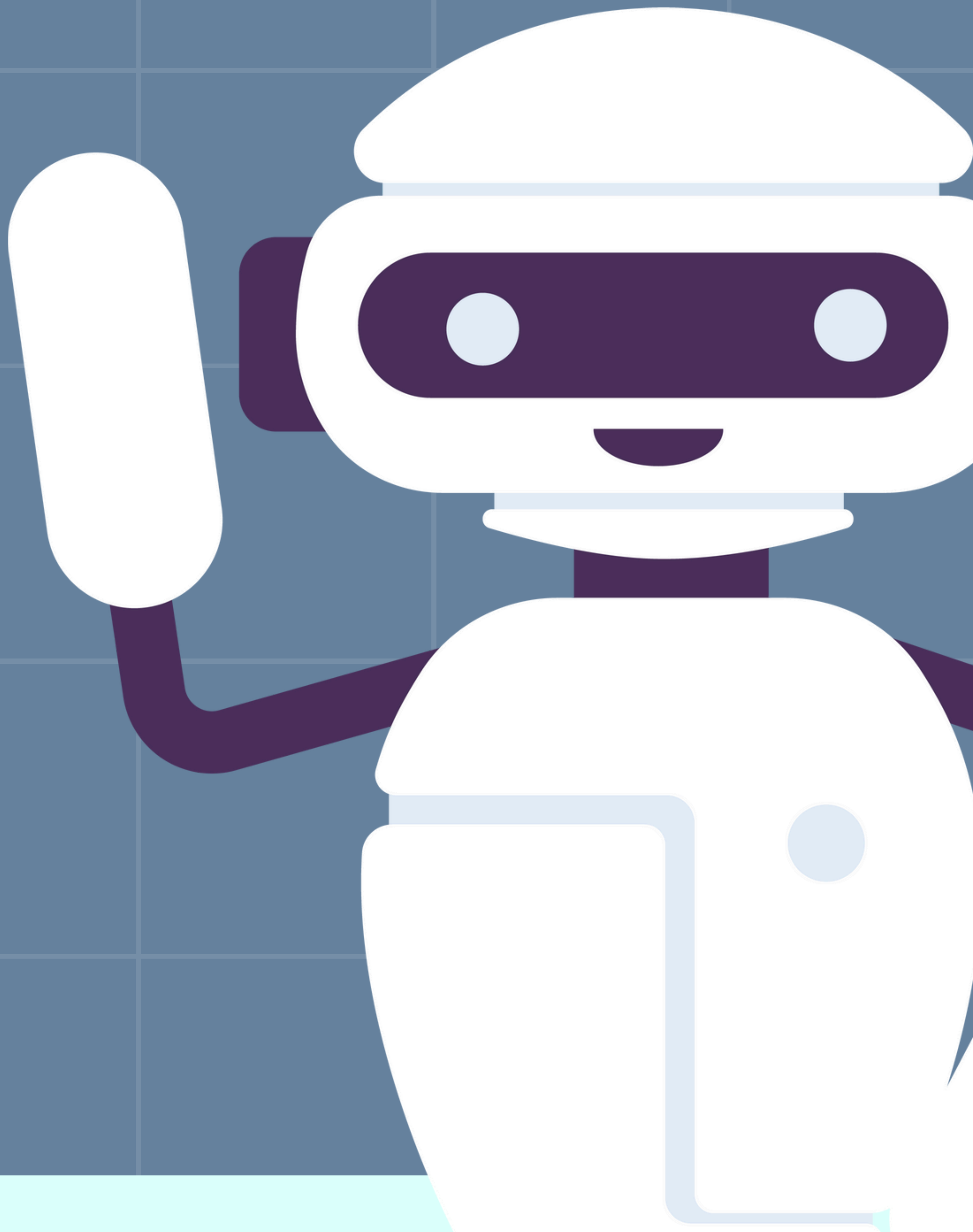
1. Camera capture the patient image and generate the point cloud image
2. Camera node and click point node send data to normal vector node
3. Calculate every point’s normal vector in point cloud data and pattern them
4. Transfer the normals from camera coordination into robot coordination by using linear transformation
5. Transform the 3 dimensional normal vector into quaternion only accepted by robot
6. Send quaternion to the move robot arms node



ROS2 code pipeline

Coding Details

1. ROS2 To operate the robotic arm
2. Downsampling
3. Calculate normal vectors by using open3d lib
4. Choose the closest point to the clicked point, which we call desired point



Estimate Surface Normal Vector

-with PointCloud data

Step 1:

Neighborhood Search

For point P_i , find a set of neighboring points.

Fix radius search or k-nearest neighbors search and get $\{P_{i1}, P_{i2}, P_{i3}, \dots, P_{ik}\}$.

Step 2:

Covariance Matrix Calculation and Eigenvalue Decomposition

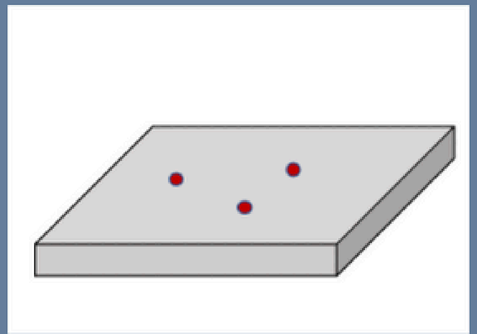
Compute the covariance matrix of the neighboring points.

Perform eigenvalue decomposition on the covariance matrix.

The eigenvector corresponding to the **smallest eigenvalue** is the normal vector.

$$\mathcal{C} = \frac{1}{k}\sum_{i=1}^{k} \cdot (\boldsymbol{p}_i - \overline{\boldsymbol{p}}) \cdot (\boldsymbol{p}_i - \overline{\boldsymbol{p}})^T, \ \mathcal{C} \cdot \vec{\mathbf{v}_j} = \lambda_j \cdot \vec{\mathbf{v}_j}, \ j \in \{0, 1, 2\}$$

Evaluation



Select and mark several check points on a flat plane and human model.

Manually align the probe with the check points and record the probe's end pose.

+

Use **algorithm** to predict the placement of the probe and control the robotic arm accordingly and record the probe's end pose. Repeat this procedure several times.

① **Standard Deviation - variability**

$$S = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

② **Error Percentage - error size**

$$\text{Error Percentage} = \left|\frac{\text{Measured Value} - \text{True Value}}{\text{True Value}}\right| \times 100\%$$

③ **t-Tests – influence of surface type and camera's perspective**

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{(n_1-1)S_1^2+(n_2-1)S_2^2}{n_1+n_2-2}\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}}$$

Select Surfaces and Check Points

Pose Recording

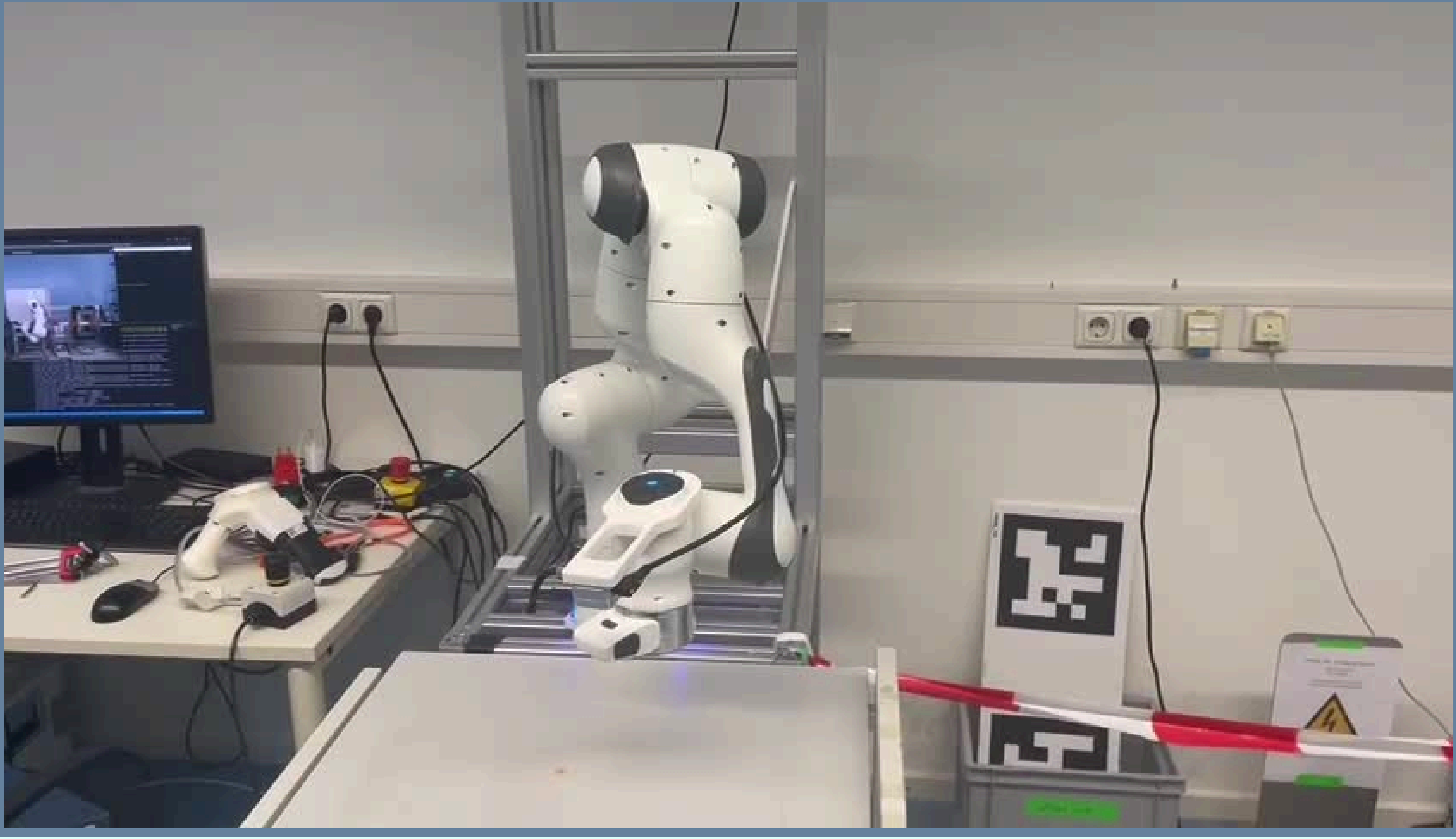
Statistical Analysis

Hardware Setup

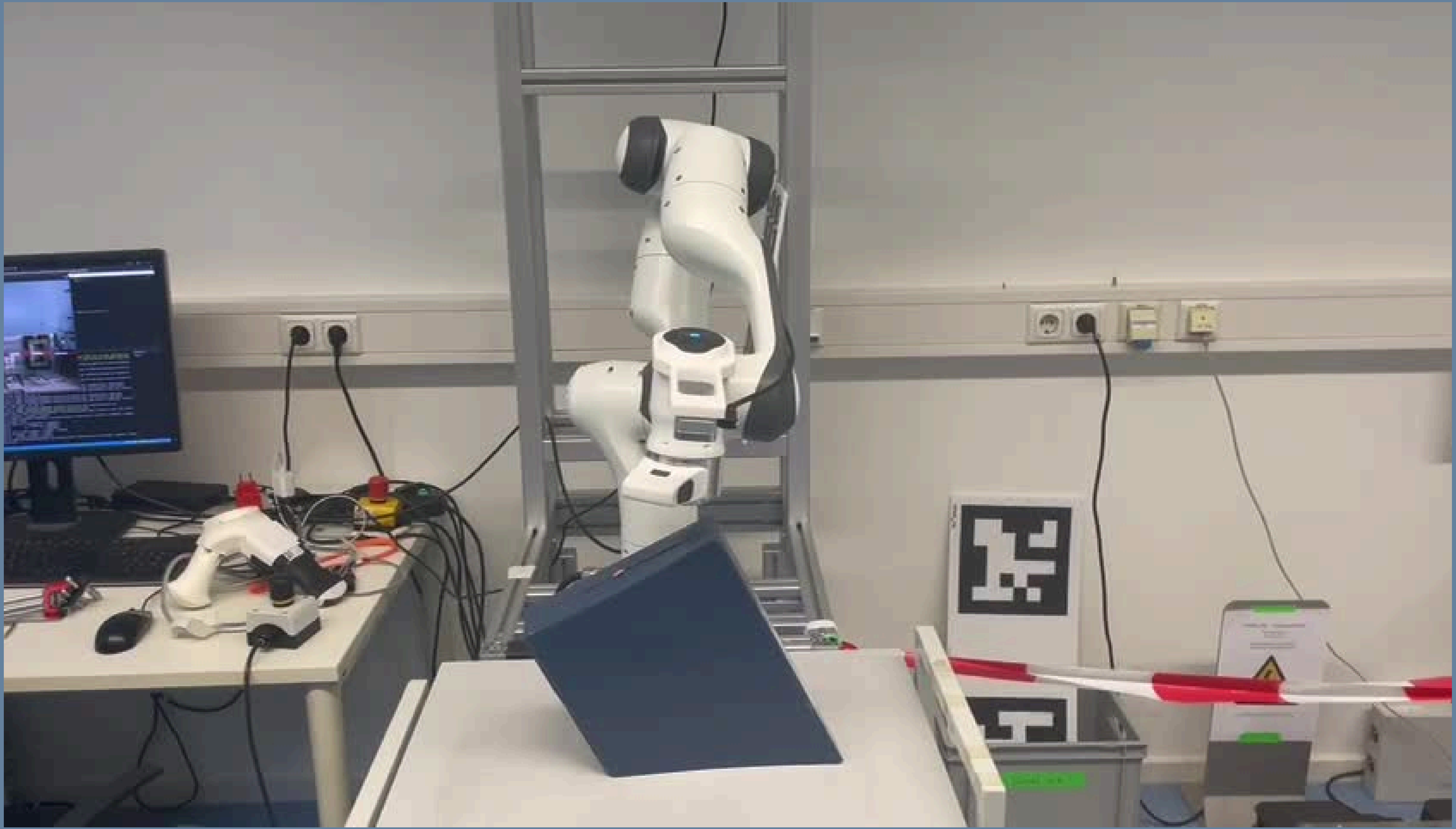
We use Franka Robotic arms to complete this project. By linking the robotic arms through the Internet, we can control it by codes.

Demo (normal vector)

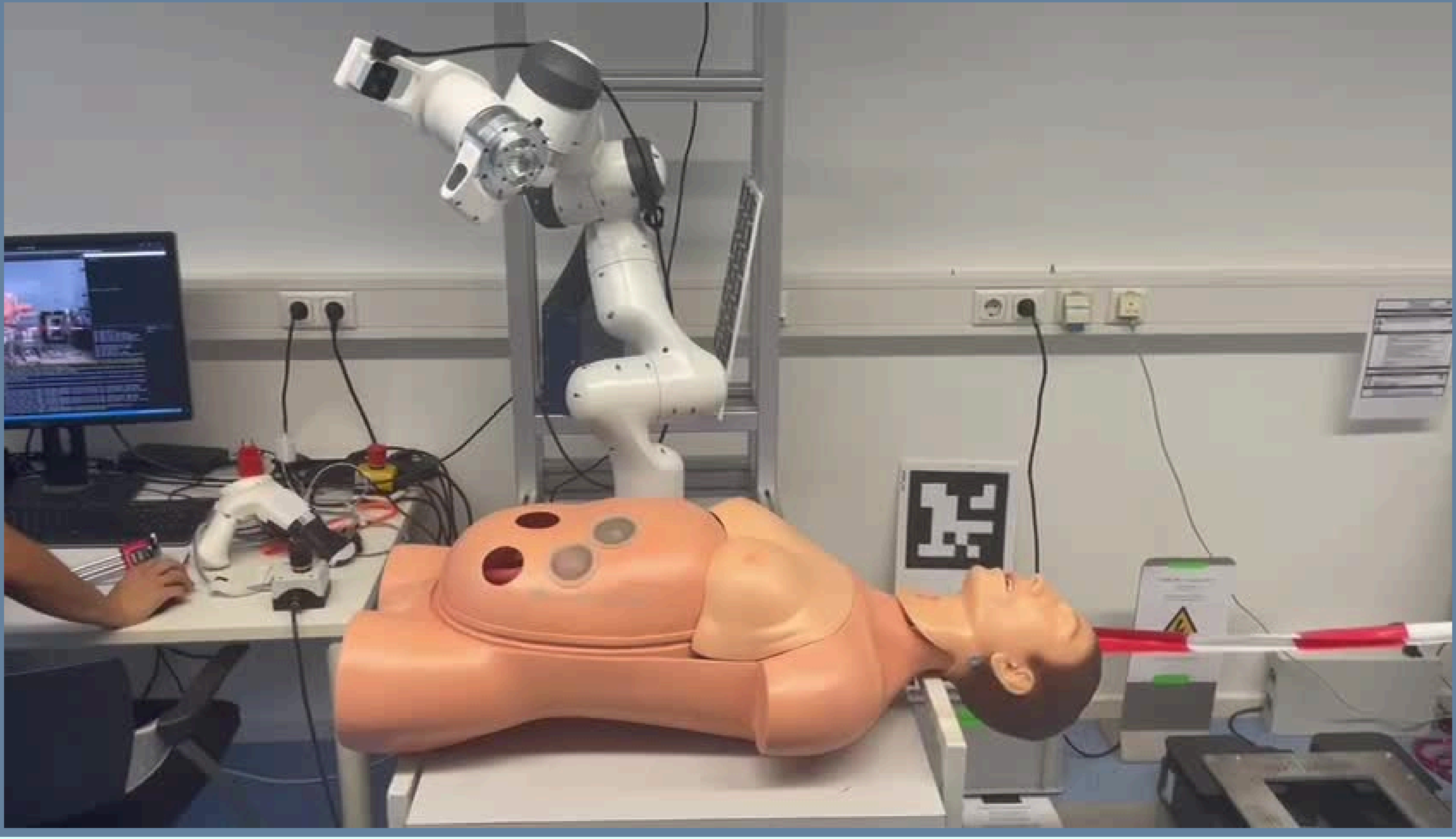
Demo (flat surface)



Demo (inclined surface)



Demo (complex surface)



Result and analysis

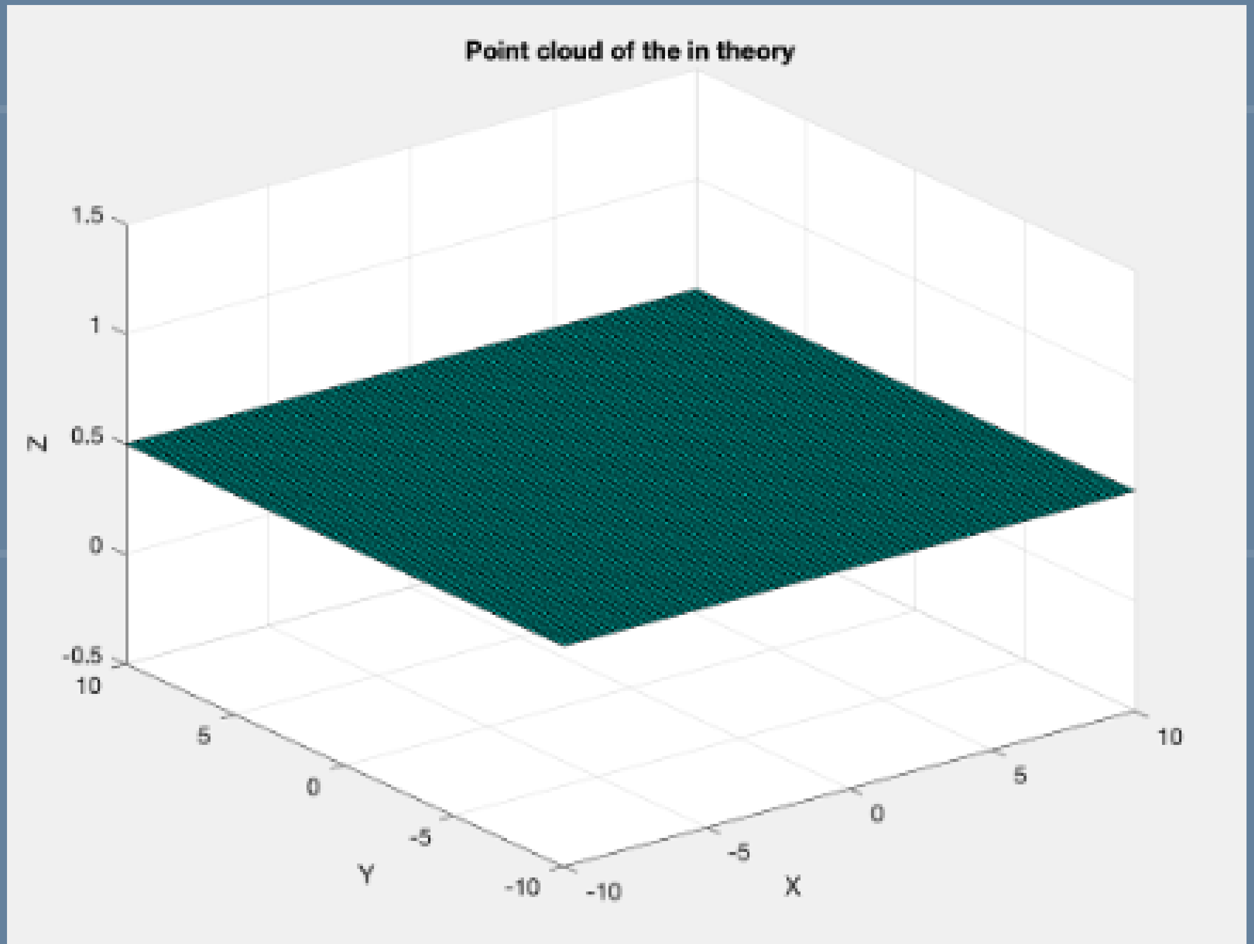
Error before applying Surface Normal Estimation Method (SNE): There are some noises when obtaining point cloud from camera

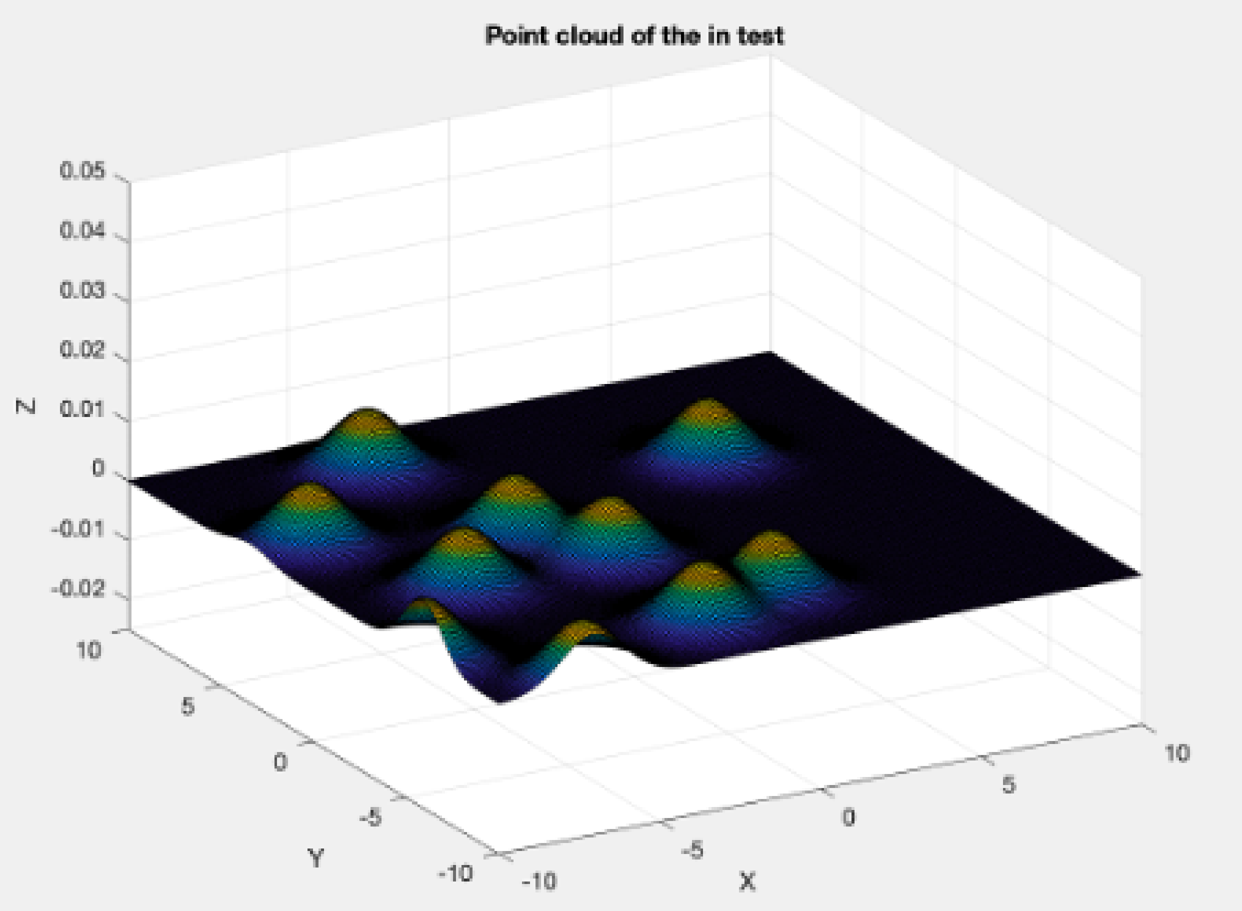
Error during applying SNE: Downsampling vs Clicked Point

- If voxel is large: Surface will be smoother, but contain less details
- If voxel is small: Surface will contain more details, but may have some noise

Best voxel we tested: 0.04

Error after applying SNE: The robotic arm requires quaternion datatype for the path planning. However, the transformation from 3D vector to quaternion may cause error.





Reference

- “Estimating Surface Normals in a PointCloud” [Online]Avaliable:https://pcl.readthedocs.io/projects/tutorials/en/master/normal_estimation.html
- “Smoothing and normal estimation based on polynomial reconstruction” [Online]Avaliable:https://pcl.readthedocs.io/projects/tutorials/en/master/resampling.html#smoothing-and-normal-estimation-based-on-polynomial-reconstruction
- “Normal Estimation Using Integral Images” [Online]Avaliable:https://pcl.readthedocs.io/projects/tutorials/en/master/normal_estimation_using_integral_images.html#normal-estimation-using-integral-images

Thank you