# Assignment 5

**Siddhant Haldar (16EE35018)**

- Libraries required -
    - OpenCV
    - Numpy
- Python Version - 3.6
- Put the input images in the **"input"** folder. The output images are also saved in **"output".**
- An existing set of sample outputs has been provided in the **"Results"** folder.

## Instructions to run the code

- In order to run the code, just use the command :
    - *python3 Assignment5_16EE35018.py*

## Note to Teaching Assistants

The code takes a little long to run since all the results for all images are being generated in one go. A copy of each result is also stored in the "output" folder if you do not want to run the code.

## Explanation of Code Execution

### Part 1 : Function to compute image curvatures  and topology

The range image is loaded. First and second order image gradients are calculated using Sobel filters. These gradients are used to compute principal, mean and Gaussian curvatures of the range image and the corresponding topologies using formula and methods as described in the slides.

### Part 2 : Function to compute Neighbourhood Plane Set(NPS) at each pixel

The Neighbourhood Plane Set(NPS) is computed at each pixel using the algorithm described in the slides. A value is computed at each pixel by binarizing the presence of DNPs at each pixel. These values are saved in NPS_image.

**Part 3 : Function to segment image using region growing algorithm**

Connected components algorithm is applied on a feature image (curvatures or NPS image) and the corresponding labels and an image with lines separating each component is returned.

**Part 4 : Saving results**

The results corresponding to all range images are stored in the "outputs" folder.

## Results

The segmentation quality comparison for every feature scheme corresponding to every range image is listed below.

**Image 0**

NPS > Gaussian > Principal + Gaussian > Principal 1 = Principal 2

**Image 1**

NPS > Gaussian > Principal + Gaussian > Principal 1 = Principal 2

**Image 2**

NPS > Gaussian > Principal + Gaussian > Principal 1 = Principal 2

**Image 3**

NPS > Gaussian > Principal + Gaussian > Principal 1 = Principal 2

**Image 4**

NPS > Gaussian > Principal + Gaussian > Principal 1 = Principal 2