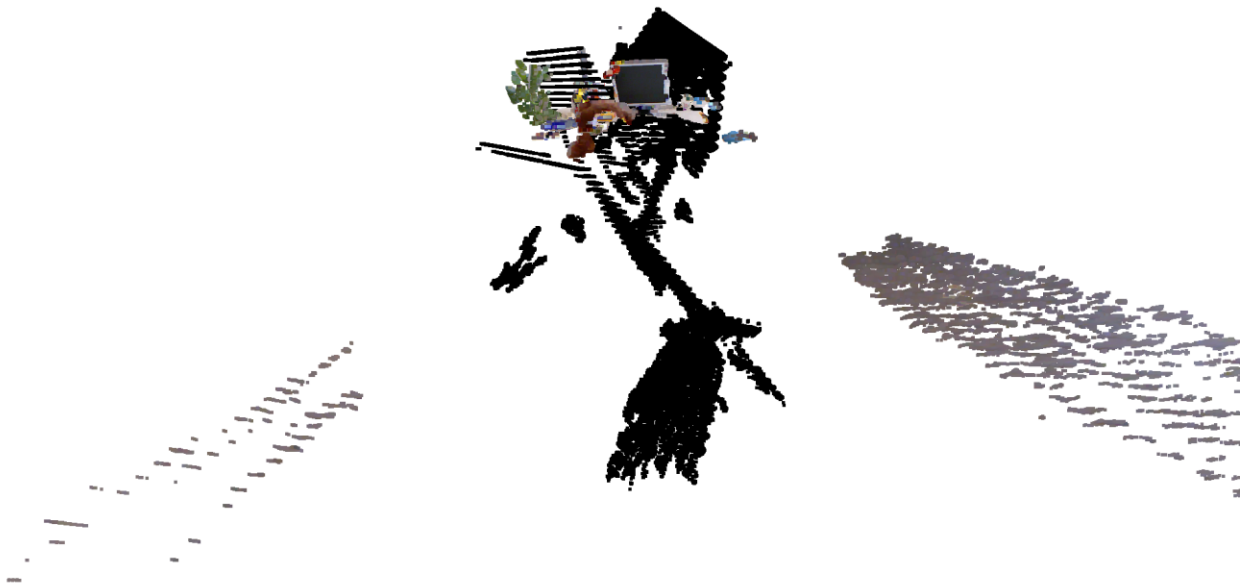
CS – 532 3D Computer Vision
Assignment 4

Problem 1

For problem 1, I have computed the parts sequentially with markdown for when one part has started. Since I was also in CS 558, some of the functions such as gaussian smoothing and getting corner response values I have taken from assignments and modified to fit the problem statement here.

The output is shown in the figure below. As can be seen, there are a lot of black points which might correspond to the depth 0 points. I would ask the professor to get insights about how to generate even better point cloud for future work.



Problem 2

For problem 2, I took the function that generates all 3D points from the previous problem and used here. The 3d points were in a shape of (307200, 3) and for the normal mapping, it was reshaped to match the image dimensions (480, 640, 3). The normal_mapping function takes in the input image and the 3d points along with verbose value which is mainly used to show output on jupyter notebook. The normal map is initialized by a zero array with same dimensions as the input image and is looped through entirely. Depth of 0 is discarded and a 7x7 neighborhood is checked if depth is not 0. Then with the neighborhood, the number of points are seen. If it is less than 3, then there are not enough collinear points for us to determine the plane so the normal is also 0 for them.

If we get the collinear points, then we calculate the plane and apply least squares to find the normal. Also mentioned in the problem, since there are 2 solution to the normal: n and -n, the plane parameters should be reversed which has been done. Following is the normal map generated.