**Project1 Report**

**Introduction**

In this project, we reconstruct a 3D scene from a bunch of 2D images taken in the same scene from several different positions. We also would like to create an object in our 3D scene, and then project it back to the 2D images. To approach our goal, we first use COLMAP to collect the features and reconstruct the 3D model. Then we use RANDSAC to find our desired plane in the 3D scene. Then we adjust plane using projection and rotation to make it parallel to the x-y plane. After that we put a cubic on that plane, and finally project the virtual box object onto 2D image.

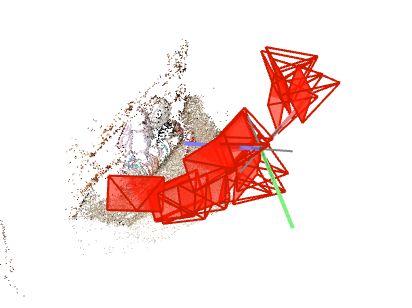
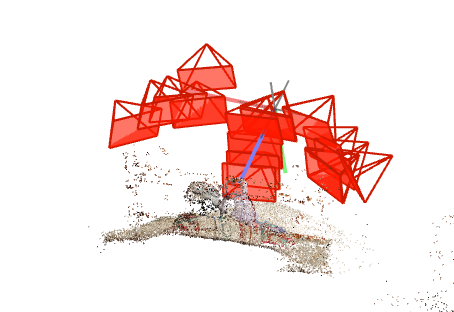
**Approach**

**Step 1:**

We take pictures using our cell phone of the apartment room from the floor. We have 30 pictures in total and use it. There are many items on the floor and the surface have a lot of texture. We also include some 3D structure besides the dominant plane. Notice that we use the same phone and the same focus to take all the pictures.

**Step 2:**

The second step is to get the COLMAP. We input the images from step 1 to the COLMAP. We process feature extraction and matching first then start the reconstruction. The program returns a 3D model. The red parts are the position and location of the camera. We export the model as test to get the detailed data.



**Step 3:**

To do step 3, we write a function named ”getDict” in our program. We read the x,y,z coordinates for each point and store them in a dictionary.

**Step 4:**

Step 5:

We will use the result of step 4. In step 4, we get the plane and the threshold. In our program, the function “plot3D“ takes the coordinates of all the points, the plane and the threshold as input. We draw the plane in the 3D model. For the points whose distance is smaller than the threshold, we use red points to represent them. For the points whose distance is higher than the threshold, we use blue points to represent them.

**Step 6:**

After we get the plane, the next step is to rotate this plane to become the plane where z=0. We compute the normal vector and compute the rotation matrix to rotate the normal vector of the plane to the normal vector of the plane where z= 0. Then we can use the rotation matrix to process all the points and rotate them to the plane where z = 0

**Step 7:**

In this step, we build a virtual cubic. The center of that rectangular bottom surface is centered at x=0, y=0, z=0. We use function “drawBox” to do this step. For debug purpose, we use different color for each surface.

**Step 8:**

For the read the carmera.txt and images.txt, we use the function in COLMAP “readModel.py”

**Step 9:**

We use the formulas to takes external and internal camera parameters and projects a given set of 3D points into a set of 2D pixel locations. Pu is the pixel location. K is the camera matrix. R is the rotation matrix. t is the translation vector . Pw is the world point.

Step 10: We draw back the cubic to the 2D image. For the debug purpose, we use different color for each surface.