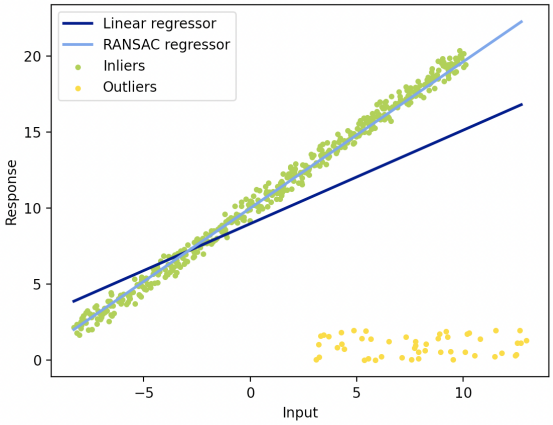
CV Assignment 3, RANSAC & MVS

Jingyu Liu

RANSAC

2.1.3. Write down the ground truth, estimation from least-squares and estimation from RANSAC:



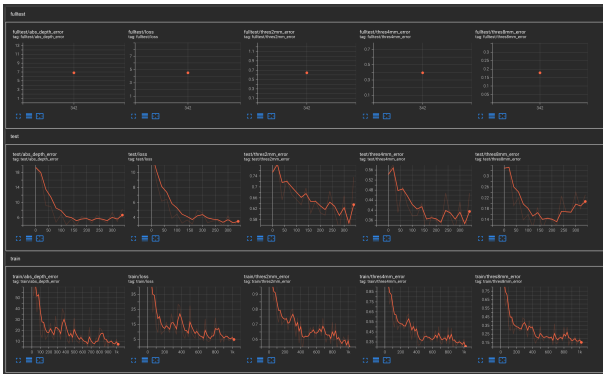
	True	Least-Squares	RANSAC
k	1	0.6159656	0.96438949
b	10	8.9617271	9.98292129

MVS

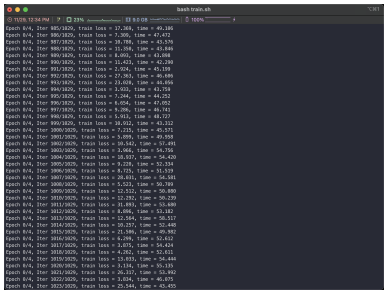
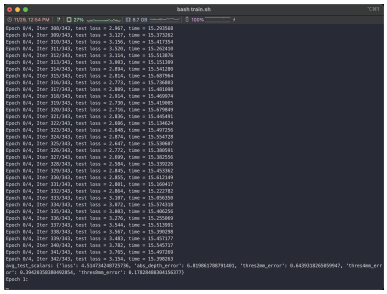
3.2.2. Write down the equation of corresponding pixel $p_{i,j} := p_i(d_j)$, first we convert the 2d coordinate into 2d homogeneous coordinate $p_{homo} = (p(x),\ p(y),\ 1)$. Compute the relative transformation from the source projection and reference projection, which is given by the code. Then we decompose the homography into rotation and translation. The final equation is:

$$p_{i,j,unhomo} = R_i(p_{homo} * d_j) + t_i,\ p_{i,j} = homogenize(p_{i,j,unhomo})$$

3.3. Screenshot of training:



Due to the computational complexity and my hardware problem, I only trained for one epoch but it is also obvious that the loss is decreasing very fastly and steadily. Each training iteration takes too long and training the whole 4 epochs on my machine will take days.

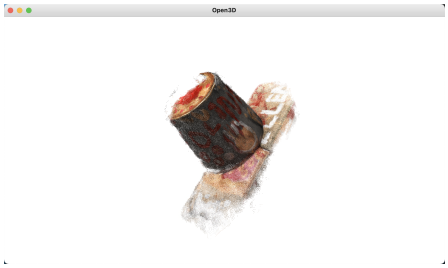
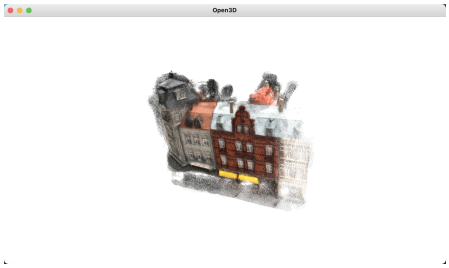
Training Log	Evaluation Log
	

3.4.

1. Explain what geometric consistency filtering is doing in the report.

When we're reconstructing dense 3d models, we want to make sure that the depth information is correct, which is ensured partially by the geometric consistency. Specifically, it first unprojects the images pixels from the reference image with the depth map, and then transforms it into the source image (just like warping) using the extrinsic and intrinsic matrices of both images. Then it uses the warped pixels to sample the depths of the source image, which then gets unprojected and transformed back to the original reference image coordinate. Finally, it checks two things to make sure that the depth information is correct: 1) it measures L2 differences between the "circular" transformed pixels and the reference pixels (like triangulation verification) and 2) calculates the relative difference between the depth information from the depth map and the depth values produced by the forward-reverse transformations. So that only those pixels with both errors below certain threshold will be used for constructing dense 3d models. At the end it also makes sure that a pixel is valid for further steps if there are at least three source images that can pass the geometric consistency test.

2. For all the scenes, visualize (visualize ply.py) and take screenshots of the point clouds in Open3D

Scene 001	Scene 009
	

3.5.

1. Taking uniform samples from the inverse range and then taking the inverse to get the depths might be helplful for lage-scale scenes.

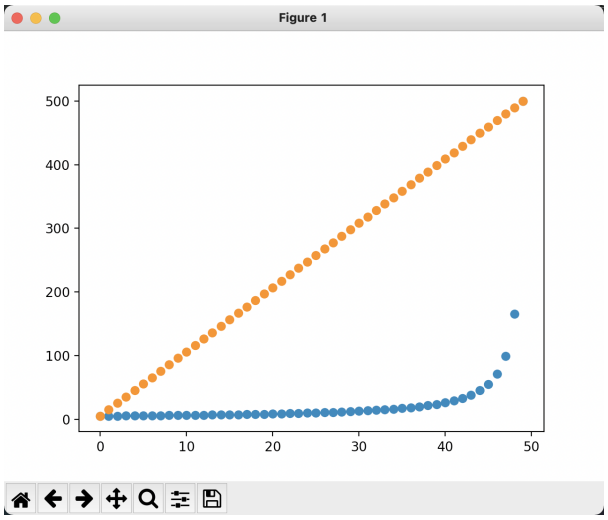


Figure 1: orange line is uniform samples in [depth_min, depth_max], blue line is uniform samples in the inverse range and than taking the inverse of the samples to get depths. Horizonal axis denotes sample index, and horizontal axis means the depth sample values.

The idea is that in the stereo setting, we have better accuracies for nearby objects and coarse estimations for objects far away. Just like being illustrated in the class, the disparity becomes less informative in far away scenes (also due to numerical issues), hence in large-scale scenes, using the new sampling methods (inverse one) can help get denser samples close to the camera, which is also shown in the plot.

2. It might not be robust because we can treat the pairs of images with occlusions as outliers as the matching similarity will likely to be low. The averages will not necessarily be the best statistics for describing the quality of the true matching similarity for the popularion

(or batch in our case). Maybe using other statistics robust for outliers such as median or percentile would help.