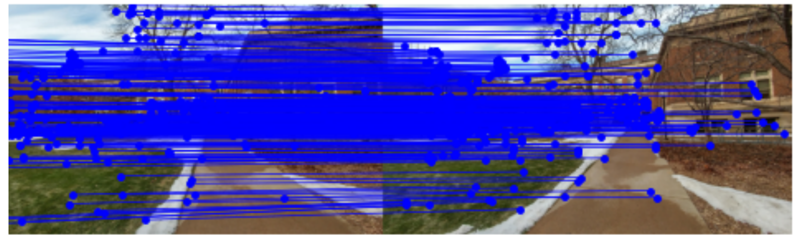
Asal Shavandi **Computer Vision HW5**

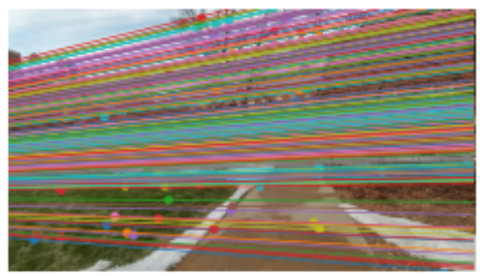
SIFT Feature Matching

Using OpenCV SIFT, extracted key points and descriptors. KNearestNeighbors is used to find matches in both images in both ways meaning right image to left image and left image to right image.
Later used distance threshold=0.7 to filter out some matches followed by bidirectional matching.
Image on the right is the visualization of matching points after thresholding and bidirectional filtering.



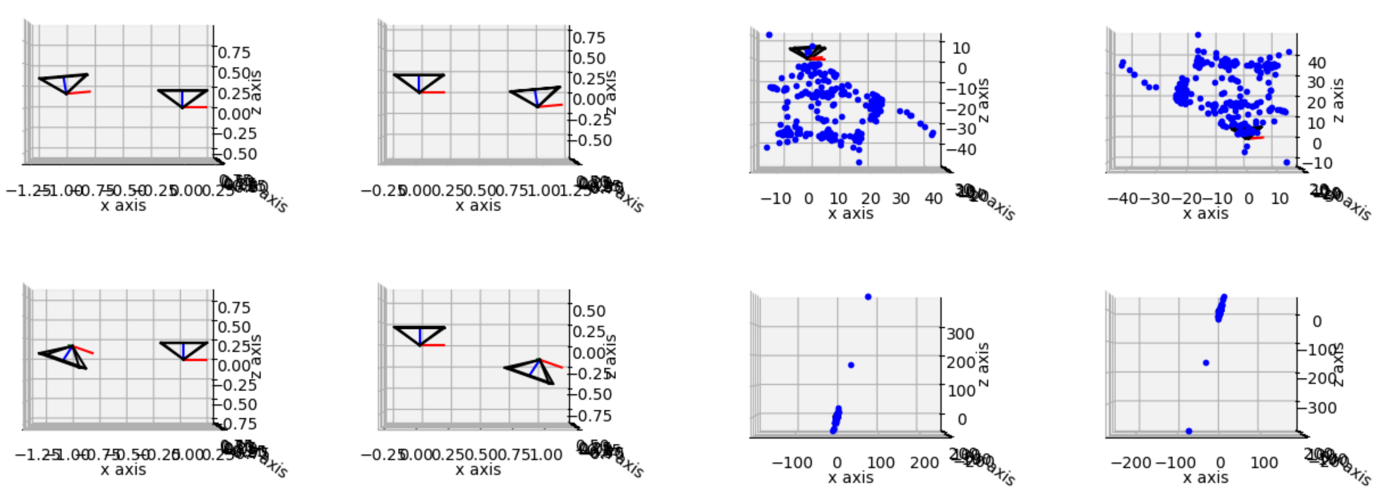
Fundamental Matrix Computation

Using RANSAC algorithm with 500 iterations, chose 8 random points from SIFT matches to form my matrix with. Then found the null space of the matrix and applied svd cleanup on the null space. Finally I can compute my Fundamental Matrix by F = u.d.vT . We know that vT.F.u=0 (F.u is our epipolar line) so at the end in order to find the best Fundamental Matrix, we will pick the one that the sum of vT.F.u for all matching points is the minimum and that matrix can be chosen as the best Fundamental Matrix. Image below is the visualization of the epipolar lines.



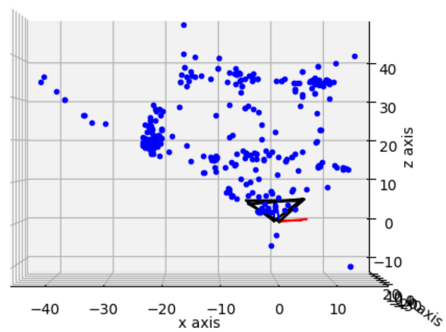
Triangulation

Given camera projection matrices and SIFT matching points, we need to reconstruct the 3D points.
For every matching points in both images, we compute the 3x3 skew-symmetric matrix followed by multiplying by camera projection matrices. We can then stack these points from both images in one matrix and find the null space [X 1]. Since my output should be 3 dimensional and last element should be 1, I divided all elements by fourth element of null space X.
Image on the left is visualization of our camera poses and image on the right is the same visualization with points after triangulation.



Pose Disambiguation

So far we've had 4 different camera poses and now we need to pick the best camera center and rotation. If R is my rotation matrix and C is camera center, for all 3D reconstructed points we need to check Cheirality condition r3.T (X-C)>0 meaning the more 3D points we have in front of our camera, the better the pose. With this condition we can count the number of valid points and finally pick the best one. The image below if camera pose and 3D points after disambiguation.



Compute Rectification

Given intrinsic and extrinsic parameters of the camera, we now need to find the homography matrices.
H = K @ Rrect @ R.T @ K-1
Which K is the intrinsic parameter and R is the given rotation matrix. Rrect is a 3x3 rectification rotation matrix that x-axis of the images align with the baseline.
Image below displays warped original images after rectification.

Disparity

Going form left image to right image and computing the minimum distance of images' descriptors using the given formula, we can visualize the disparity map.