



## Summary-Homework:5

The problem is to build a stereo reconstruction algorithm given the left and right views of a scene.

To this end I wrote codes in MATLAB compatible with the given specifications. The `main_stereo.m` file takes in the intrinsic parameters of the camera and the left and right views of the scene and produces the disparity map which gives us a notion of the depths at which different objects are located in the scene. The first step is to take the left and right views of the scene and compute the corresponding matched points. This is done using SIFT in the code `FindMatch.m`. After this, the Fundamental matrix is calculated which gives us a relationship between the two views of the scene. It has been implemented in the RANSAC framework by using Fundamental matrix as the consensus. 100000 RANSAC iterations have been used and the RANSAC threshold used is 0.01. Also, a SVD cleanup has been done to project it to a rank 2 matrix. Given the fundamental matrix, and the Intrinsic camera parameter, the 4 possible poses of the camera are computed and later one of them is filtered out by applying the Cheirality condition post the triangulation of the 2D points. Later, the left and right Homography matrices are obtained from the relative pose of the second camera w.r.t the first, the camera centre and the intrinsic parameters. The original images are then warped using the obtained Homography matrices to obtain the rectified images. These rectified images where the epipolar lines are horizontal are now used as input to obtain the disparity map. A dense SIFT feature is computed for every pixel using `vl_dsift`. To obtain features for every pixel the images has been zero padded. For every pixel in the left image it's corresponding pixel in the right image is found by scanning along the epipolar line. The obtained pixel position is the one where the norm of the difference between the pixel descriptors is the minimum.

The left and right rectified images and the disparity map have been visualized and shown above