1. SIFT Features
    a. Using CV2's SIFT algorithm to go through grayscale image and compute keypoints
    b. sift.detect() function finds the keypoint in the images. You can pass a mask if you want to search only a part of image. Each keypoint is a special structure which has many attributes like its (x,y) coordinates, size of the meaningful neighbourhood, angle which specifies its orientation, response that specifies strength of keypoints etc.
    c. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html (Sift explanation)
2. Matching detected keypoints between images
    a. Tries to find the same keypoint and draw lines to connect them using k-NN or L2 Norm as a metric
3. Compute calibrated coordinates
    a. Fetching keypoints, making calibrated matrices
4. Least Squares Estimator
    a. Multiply x1 and x2 (grayscale images)
        i. Each column of X1 multiplied with X2.transpose
    b. Take the SVD of the above matrices stacked together
    c. Output is the least squares estimate of the essential matrix
5. RANSAC
    a. We create a least squares E using 8 sample points from both images
    b. Using the E, we can use our test indices to find inliers
    c. Inliers are determined by checking whether their residuals are lower than the specified eps
    d. We maximize the number of inliers over all iterations (20K default)
6. Plot Epipolar Lines
    a. Determine fundamental matrix using least squares E and calibration parameters
    b. Compute epipolar lines using fundamental matrix and uncalibrated points
7. Pose Candidates
    a. Decompose essential matrix into R and T using SVD
    b. T1 and T2 are given
    c. For a given estimate of E there are 2 possible solutions, obtained from the SVD
    d. If determinant of R_1 is negative, bottom of main diagonal in z matrix is determinant of U * V
8. Reconstruct 3D
    a. For each calibrated point, we triangulate a lambda using the given form

$$\underbrace{(q_i - Rp_i)}_{3\times 2} \underbrace{\begin{pmatrix} \mu_i \\ \lambda_i \end{pmatrix}}_{2\times 1} = \underbrace{T}_{3\times 1}$$

    b. The code filters out all the lambdas which are negative since negative depth isnt possible

   c. We then determine which of the R,T combinations have the highest added sum of lambas. The one with the highest sum is the combination that is chosen
9. Show reprojections
   a. P2 and p1 projections are
   b. For image 2: K * (inv(R)P2 - inv(R)T)
   c. For image 1: K * (RP1 + T)