1. 2D Projective Transformations.
   (a) Implement MaSKS Algorithm 5.2 (The four-point algorithm for a planar scene), p. 139.
   (b) Use the four-point algorithm with \( n \geq 4 \) hand-clicked correspondences to remove the
       projective distortion from three images: building.gif, floor.gif, and one image of
       your own choice.


3. Prove MaSKS Corollary 5.23 (From essential matrix to homography), p. 142.


5. Reconstruction from Two Calibrated Views.
   (a) Implement MaSKS Algorithm 5.1 (The eight-point algorithm), p. 121.
   (b) Run the script make_scene.m to produce two views of a synthetic scene, \( \{x_i^j\}_{j=1}^n \), \( i = 1, 2 \).
       Use the eight-point algorithm to estimate the four possible decompositions \( (R, \hat{T}) \) for \( E \).
   (c) Estimate the depths of the points and the global scale factor by solving for \( \hat{X} \) in MaSKS
       Equation (5.21), p. 125 (Linear triangulation). Record the values of \( R, T \) and \( \gamma \) for which
       all the depths are positive.
   (d) Plot the estimated 3D coordinates of the pointset relative to each camera frame.
   (e) Compute the reprojection error using MaSKS Equation (5.23), p. 127.

6. Implement Hartley normalization as defined in MaSKS Equation (6.77), p. 212. Demonstrate it
   on a set of 100 random 2D points distributed uniformly on the rectangular area \([1, 128] \times [1, 192]\).

   (a) Implement MaSKS Algorithm 6.1 (The eight-point algorithm for the fundamental matrix), p. 212,
       with Hartley normalization.
   (b) Run your code on the stereo pair of desk1.gif and desk2.gif with \( n \geq 8 \) hand-clicked
       correspondences. Plot the epipolar lines \( \ell_1 \) and \( \ell_2 \) for at least three points in the first
       view, and verify that they pass through the corresponding points in the second view.
   (c) Solve for the coordinates of the epipoles \( e_1 \) and \( e_2 \).
   (d) Repeat the above two steps for another stereo pair of your own choosing.

8. Stereo Rectification.
   (a) Implement MaSKS Algorithm 11.9 (Epipolar rectification), p. 406.
   (b) Demonstrate your code on the image pair blocks{1,2}.gif.