Nicholas Sun A0217609B

Report for Assignment 1

5.1 Method 1: Two-Step Stratification

Stage 1: Removing Projective Distortion.

1. Get two vanishing points, one each from intersection of a pair of parallel line.
2. Get vanishing line from the two vanishing point
3. Get equation of Hp using

$$H_p^{\top} = \begin{pmatrix} 1 & 0 & -l_1/l_3 \\ 0 & 1 & -l_2/l_3 \\ 0 & 0 & 1/l_3 \end{pmatrix}.$$

4. This homography matrix will transform the vanishing line to a line of inifinity, recovering some affine property
5. Warp the image with this predicted projective transformation

Stage 2: Removing Affine Distortion

1. From two pairs of orthogonal lines, construct

$$(l'_1m'_1, l'_1m'_2 + l'_2m'_1, l'_2m'_2)$$

   where each orthogonal line pair forms a constraint and two such constraints can be stacked to give a 2 x 3 matrix
2. The 2 x 3 matrix can be decomposed using SVD to get a 3 x 1 vector s
3. Since s is S written as a 3 x 1 vector, we can get back the 2 x 2 S by arranging s to get [[s1,s2],[s2,s3]]
4. Since S = K * K.T, we can find K using a Cholesky Decomposition of S
5. With K, we can get the homography Ha needed to remove the affine distortion given by,

$$H_A = \begin{bmatrix} K & 0 \\ 0^{\top} & 1 \end{bmatrix}$$

6. Warp the affinely rectified image with this predicted affine transformation

5.2 Method 2: One-Step Using $C\infty$

1. From 5 pairs of orthogonal line pairs, we can stack 5 constraints to form the left matrix given in this equation,

$$(l'_1m'_1, (l'_1m'_2 + l'_2m'_1)/2, l'_2m'_2, (l'_1m'_3 + l'_3m'_1)/2, (l'_2m'_3 + l'_3m'_2)/2, l'_3m'_3)\,\mathbf{c} = 0$$

2. Like before, we use SVD to get c, where $\mathbf{c} = a, b, c, d, e, f$ ⊤ is $C'\infty*$ written as a 6-vector

3. We can then get

$$C = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}$$

4. Using SVD on C, we can get C = UDV where U,D,V corresponds to

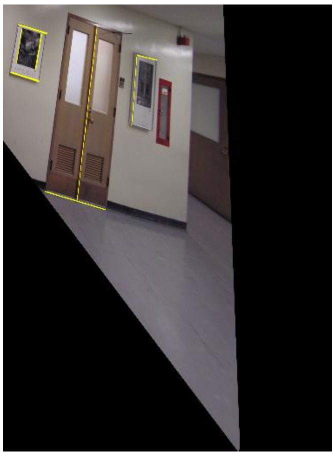
$$C_{\infty}^{*\prime} = U \begin{bmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{bmatrix} U^{\top}.$$

5. Get the rectifitying projectivity H = U up to a similarity square root of S

6. Warp the image using this predicted projectivity matrix

Result of method 1, stage 1

Result of method 1, stage 2



Result of method 2, when rotated looks similar to the above results

6.1 Robust Homography Estimation Using RANSAC

compute_homography_error( )

1. The error $d(x, x'; H) = \|x - H^{-1}x'\|^2 + \|x' - Hx\|^2$ can be calculated using numpy apis

compute_homography_ransac( )

Taken from Lecture 3 Slide 88,
For N number of tries,
a. Select a random sample of 4 correspondences and compute the
homography, H.
b. Calculate the distance d for each putative correspondence using the compute homography error method
c. Compute the number of inliers consistent with H by the number of
correspondences for which d < t
Choose the H with the largest number of inlier