1 Homographies

1.1 Homography

Prove the existence of an H for the homography equation

- P_1, P_2: 3×4 camera projection matrices
- x_1, x_2: Points in homogenous coordinates
- Homogrpahy equation:

$$x_1 \equiv H x_2 \tag{1}$$

For a 3D point in homogenous coordinates we have $x_i = [X_i Y_i Z_i]^T$ The equation for P_1 would then be:

$$x_1 \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

A similar equation can be written for x_2 as well. Since the 3×4 projection are not square matrices, they are not invertible. One workaround this problem is to consider the XY plane only by setting $Z = 0$. This process yields two 3×3 matrices, one from each projection matrix. Assuming that these square matrices are invertible and are represented as A_i matrices, gives us the following result: $x_1 = A_1$ P $= A_1 A_2^{-1}$ $P \equiv$ H x_2 Hence proved.

1.2 Correspondences

1. **h** has 8 degrees of freedom (D.O.F) as the equation $A_i h = 0$ is only correct to a scaling factor.

2. A total of eight points, i.e. **four** pairs are required to solve for **h**.

3. Derivation for A_i:

We are given the equation $x_i^i \equiv Hx_2^i$, which can be expanded as follows:

$$x_1 \equiv \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \tag{3}$$

The equation above can be re-written as:

$$\begin{bmatrix} ku_1 \\ kv_1 \\ k \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \tag{4}$$

where k is a constant to eliminate the $\equiv$ relation. Now, to re-arrange the equation in the original form without constant k, we divide by the resultant for the third row as follows:

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} \frac{h_{11}}{h_{31}u_2} & \frac{h_{12}}{h_{32}v_2} & \frac{h_{13}}{h_{33}} \\ \frac{h_{21}}{h_{31}u_2} & \frac{h_{22}}{h_{32}v_2} & \frac{h_{32}}{h_{33}} \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \tag{5}$$

Now, writing the matrix in equation form gives the following equations:

$$h_{31}u_2u_1 + h_{32}v_2u_1 + h_{33}u_1 - h_{11}u_2 - h_{12}v_2 - h_{13} = 0 \tag{6}$$

$$h_{31}u_2v_1 + h_{32}v_2v_1 + h_{33}v_1 - h_{21}u_2 - h_{22}v_2 - h_{23} = 0 \tag{7}$$

Rearranging equations 6 and 7 yields the following result:

$$\begin{bmatrix} -u_2 & -v_2 & -1 & 0 & 0 & 0 & u_2u_1 & v_2u_1 & u_1 \\ 0 & 0 & 0 & -u_2 & -v_2 & -1 & u_2v_1 & v_2v_1 & v_1 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ \vdots \\ h_{33} \end{bmatrix} \tag{8}$$

Therefore,

$$\mathbf{A} = \begin{bmatrix} -u_2 & -v_2 & -1 & 0 & 0 & 0 & u_2u_1 & v_2u_1 & u_1 \\ 0 & 0 & 0 & -u_2 & -v_2 & -1 & u_2v_1 & v_2v_1 & v_1 \end{bmatrix} \tag{9}$$

4. The trivial solution for **h** would be $h = \left[[0 \quad 0 \quad 0 \quad \dots \quad 0]^T\right]$ where h is a 9×1 matrix. **A** is not full rank because the matrix only has eight degrees of freedom. So,there exists a nullspace of the matrix for the zero (or close to zero) eigenvalue. The rank of **A** plus the nullity of **A** is also not equal to dimension of the square matrix **A** The nullity is the dimension of the kernel of the matrix, which is all vectors v of the form: $Av = 0 = 0v$. The kernel of A is precisely the eigenspace corresponding to eigenvalue 0. So, to sum up, the rank is n minus the dimension of the eigenspace corresponding to 0. If 0 is not an eigenvalue, then the kernel is trivial, and so the matrix has full rank n. The rank depends on no other eigenvalues.

1.3 Homography under rotation

For camera 1: $x_1 = K_1[I \ \ 0]X$
For camera 2: $x_2 = K_2[R \ \ 0]X$

X is a point in 3D space $\implies \mathbf{X} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

Eliminating the zero columns from K and [I 0] results in the following relation: $x_1 = K_1 \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

$$\implies x_1 = K_1 X \implies X = K_1^{-1} x_1 \tag{10}$$

Similarly,

$$x_2 = K_2[R \ \ 0]X \tag{11}$$

where, R is a rotation matrix as defined below: $\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Now, using equations 10 and 11, we can refer the following:

$$x_2 = K_2[R \ \ 0] \begin{bmatrix} K_1^{-1} x_1 \\ 1 \end{bmatrix} = K_2 R K_1^{-1} x_1 \tag{12}$$

Thus, $x_2 = H x_1 \implies$

$$H = K_2 R K_1^{-1} \tag{13}$$

1.4 Understanding homographies under rotation

From equation 13 we get, $H^2 = (K_2 R K_1^{-1})(K_2 R K_1^{-1})$ Since Ks are the same for this problem, $H = K R K_1^{-1}$

$$H^2 = KRRK^{-1} = K \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} K^{-1} \tag{14}$$

$$= K \begin{bmatrix} \cos^2\theta - \sin^2\theta & -2\sin\theta\cos\theta & 0 \\ 2\sin\theta\cos\theta & \cos^2\theta - \sin^2\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} K^{-1} \tag{15}$$

$$= K \begin{bmatrix} \cos 2\theta & -\sin 2\theta & 0 \\ \sin 2\theta & \cos 2\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} K^{-1} = KR(2\theta)K^{-1} \tag{16}$$

Thus, H^2 is a function of 2θ.

1.5 Limitations of planar homography

Planar homography is not completely sufficient to map any arbitrary scene image to another viewpoint because planar homography has limited capability when it comes to planes that have repeated patterns. Also, homography does not take z-axis plane into account and only analyses images as flat planes, which might lead to loss in critical information.

1.6 Behavior of lines under perspective projection

Perspective projection preserves lines, i.e. a line in 3D is projected to a line in 2D. This can be verified algebraically as follows: Let P be a projection matrix and **X** be a line in 3D with points [1,1,2], [2,2,2], and [6, 6, 2]. Let the Projection matrix be:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

Projection in 2D plane is as follows:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 6 \\ 1 & 2 & 6 \\ 2 & 2 & 6 \\ 1 & 1 & 1 \end{bmatrix} \tag{18}$$

When this projection matrix is multiplied with our points of the line, all the values in the z-coordinate go to zero, but the points in x and y are still preserved, thus depicting that the line is still preserved.

For this case we can assume that the camera frame has no translation or rotation. So the extrinsic and intrinsic parameters are unchanged. When these points are normalized from homogeneous coordinates and the projection matrix is multiplied to these points, the output is points that also a line but in 2D.

The rank is dimension minus the dimension of the eigenspace corresponding to 0. If 0 is not an eigenvalue, then the kernel is trivial, and so the matrix has full rank. Therefore, in this case we have the dimension of eigenspace as 1. For eigenvectors, as the number of linearly independent eigenvectors of the zero eigenvalue of a matrix N is equal to the nullity of the matrix, the rank and the nullity sum up to the number of columns of N. Hence, the number of linearly independent eigenvectors of the zero eigenvalue of N and the rank of N add up to the number of columns of M.

2 Computing Planar Homographies

2.1 Feature Detection and Matching

2.1.1 FAST Detector

Both FAST detector and Harris corner detector use change in intensity to detect corners in an scene. However, FAST detector has a lower computational cost than Harris corner detector as FAST samples a pixel as an interest points, sets a threshold value and then considers a circle of 16 pixels around the pixel of interest. From these pixels, the intensities of four pixels are checked to see if they are all above or below the threshold to establish the presence of a corner. On the other hand, Harris corner detector uses a sliding window based on the idea that shifting this window in **any direction** would give a large change in intensity. Depending on the size of the window and the image, Harris corner detector can ne computationally intensive.

2.1.2 BRIEF Descriptor

BRIEF provides a shortcut to finding binary strings directly without finding any descriptors. It is different from the filter banks we've seen in lectures in the sense that it selects location pairs from a smoothened image patch and compares the intensities. However, Filter banks we have used in the past such as the gaussian and Laplacian filters are not ideal for feature detection as they perform a single operation on the whole image irrespective of the variation in intensity or location only care about classifying the whole image.

2.1.3 Matching Methods

Hamming Distance Methods like hashing using LSH (Locality Sensitive Hashing) is used to convert SIFT descriptors to binary strings. These binary strings are used to match features using Hamming distance. Hamming distance is used in BRIEF descriptor to find the two closest points and the ratio of the distances is then used to classify the closest point whereas Nearest Neighbor only classifies the pixels based on the number of clusters and the k-nearest points irrespective of the distance between these points.

Hamming distance provides faster feature matching as computing Hamming distance only involves applying XOR and bit count operations. Thus, Hamming distance is faster than other conventional operators such as Euclidian.

2.1.4 Feature Matching

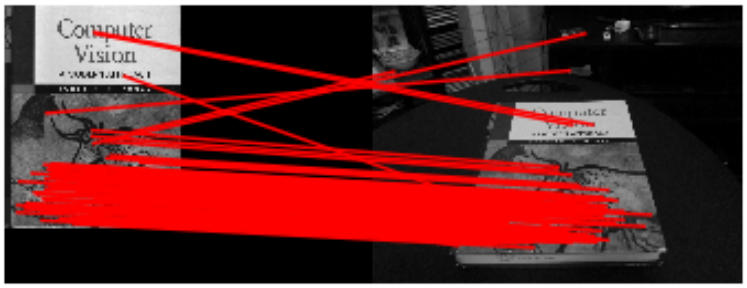

Figure 1: Responses for default ratio and constant sigma(σ)

2.1.5 Feature Matching Parameter Tuning

Sigma (σ)	Ratio
0.05	0.7
0.15	0.7
0.7	0.7
0.15	0.1
0.15	0.5
0.15	0.9

Table 1: Iteration results from feature matching parameter tuning

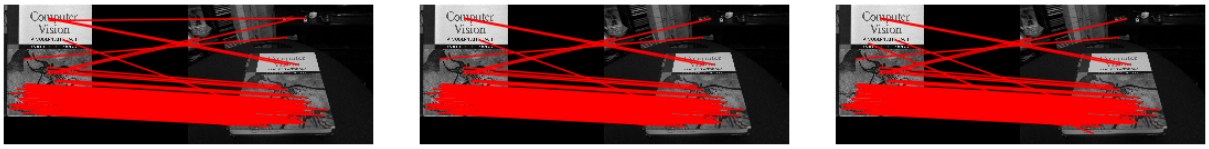

Figure 2: Responses for different values of sigma(σ) at a constant ratio

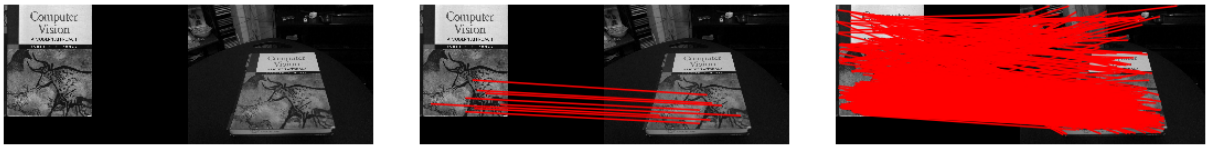

Figure 3: Number of matches as the image is rotated in increments of 10 degrees

Conclusions:

- From figure 7 it can be seen that at a low value of sigma (0.05) there were slightly more matches outside the actual book. Overall, the variation in matching was not pronounced when sigma was varied.
- Unlike the variation of sigma, changing the ratio impacted the matching efficacy significantly. At extremely low ratio value (0.1) there were almost no matches. Moreover, as the ratio values were increased the number of points matched increased, but this also increased incorrect matches.

2.1.6 BRIEF and Rotations

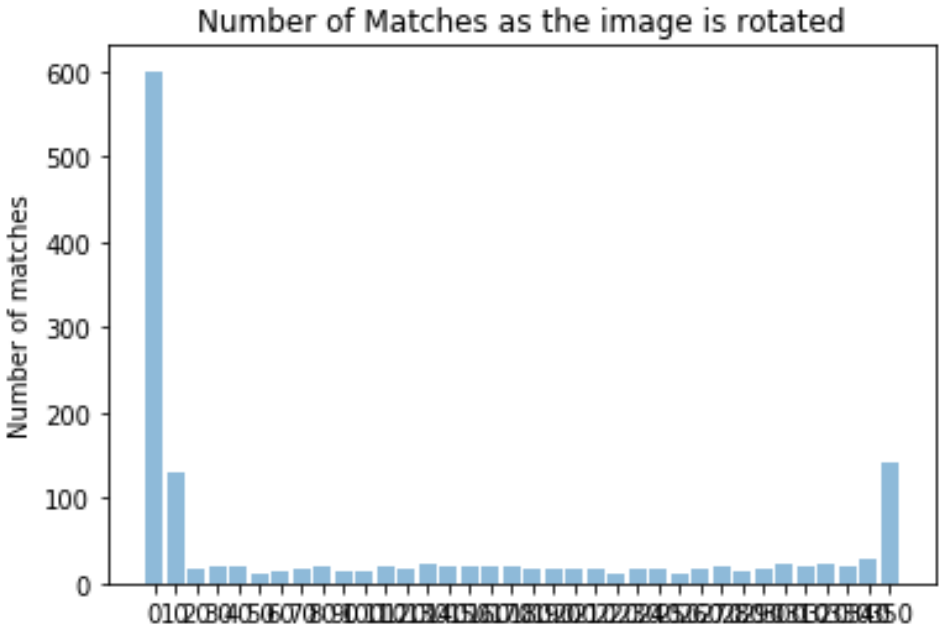

Figure 4: Number of matches as the image is rotated

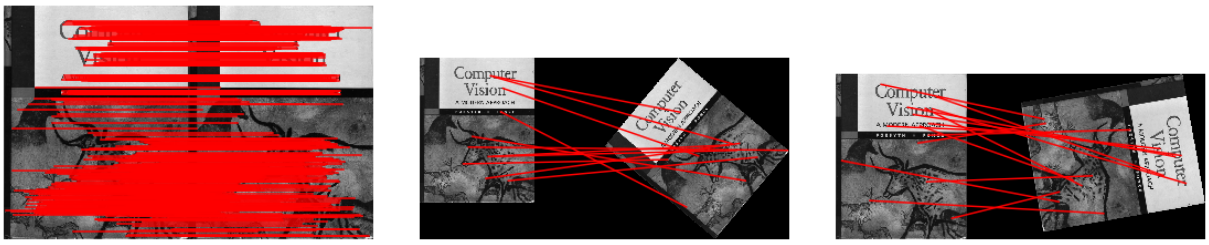

Figure 5: Responses for matches when the image is rotated in increments of 10 degrees.

The histogram above depicts that as the image is rotated clockwise, the number of matches dips significantly. This is because BRIEF and corner detection are not the best algorithms for feature matching. Thus, as soon as we rotate our image, the descriptor has a tough time recognizing the features it normally would if the image weren't rotated.

2.2.4: Automated Homography Estimation and Warping- Harry-Potterize

4. The image is being warped to the correct location but is not filling up the same space as the book as the Harry Porter image and the CV book image are of different shapes. One way to resolve this issue would be to resize the images to be of the same size. Moreover, it is possible that the homography being computed might not be ideal. The matching and homography process needs to be optimized by tuning parameters such as the σ, the ratio, and other RANSAC-specific parameters such as maximum iterations, and inlier tolerance.

6. **Implement the function:**

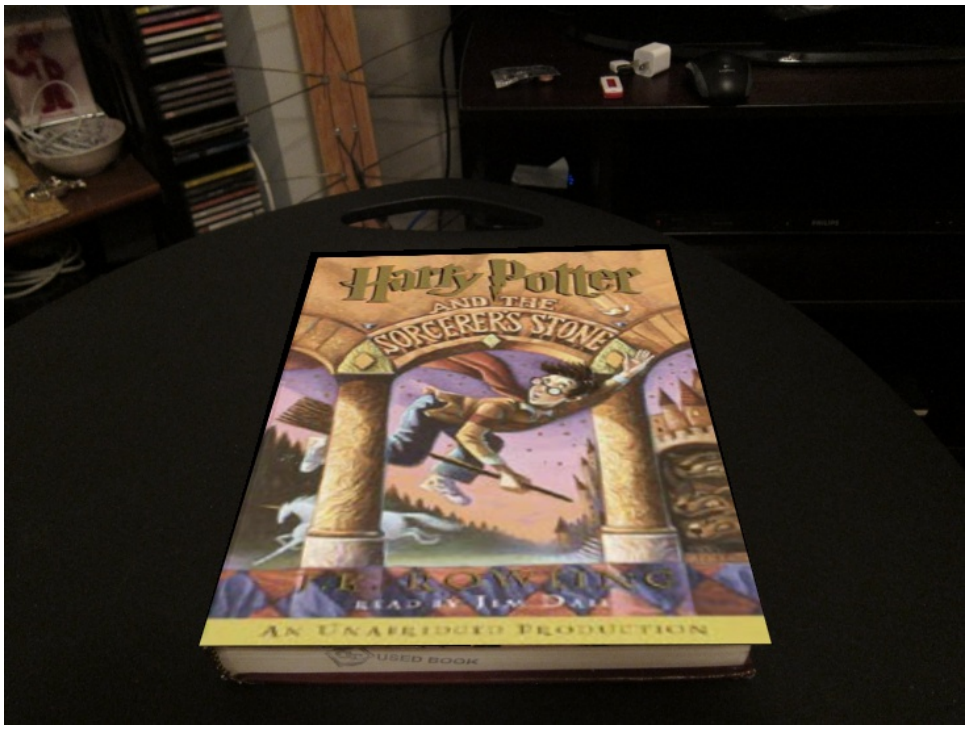

Figure 6: HarrryPoterize output from compositeH

2.1.7 RANSAC Parameter Tuning

Max Iterations	Inlier Tolerance	Inliers Found	Accuracy (%)
100	2.0	117	95.12
500	**2.0**	118	95.93
1000	2.0	118	95.93
5000	2.0	118	95.93
10000	2.0	118	95.94
5000	1.0	109	88.62
5000	5.0	119	96.75
5000	7.0	119	86.75
5000	10.0	119	96.75

Table 2: Iteration results from RANSAC Parameter Tuning
Results from Parameter Tuning

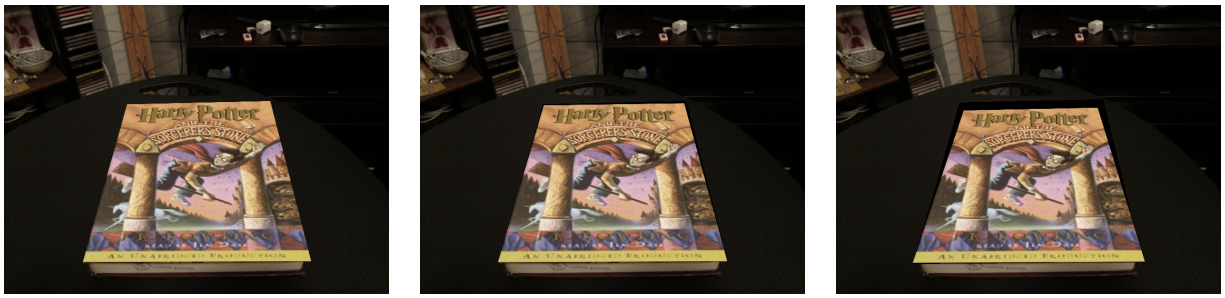

Figure 7: Responses for varying maximum iterations values at a constant tolerance and varying tolerance for fixed sigma.

Conclusions:

- Varying the maximum iterations for RANSAC did not have a huge impact on the accuracy of the RANSAC performance.
- Relaxing the tolerance for RANSAC did increase the apparent accuracy of RANSAC. However, the more we relax the model, the more susceptible the algorithm is to inaccurately matching descriptors.
- Tightening the constraints for tolerance decreased the accuracy of the model, as fewer points were matched. However, there is a higher chance that the matches were actually correct in this case. Overall, only the alignment of the Harry Porter with the book changes slightly as these parameters are varied one at a time.

3.1: Augmented Reality

Screenshot of the Video

Figure 8: Screengrab of the AR video frame