

# 1 Homographies

## Planar Homographies as a Warp

### Q1.1 Homography

Consider  $\mathbf{x}_\pi$  to be a point in 3D space given by the homogeneous coordinates  $\mathbf{x}_\pi = [X_\pi \ Y_\pi \ Z_\pi \ 1]^T$ , which is located on a plane  $\Pi$  defined by a 3D point with homogeneous coordinates  $\mathbf{a}_0 = [a_{01} \ a_{02} \ a_{03} \ 1]^T$  and two basis vectors given by homogeneous coordinates  $\mathbf{a}_1 = [a_{11} \ a_{12} \ a_{13} \ 0]^T$  and  $\mathbf{a}_2 = [a_{21} \ a_{22} \ a_{23} \ 0]^T$  respectively, as shown in the figure below.

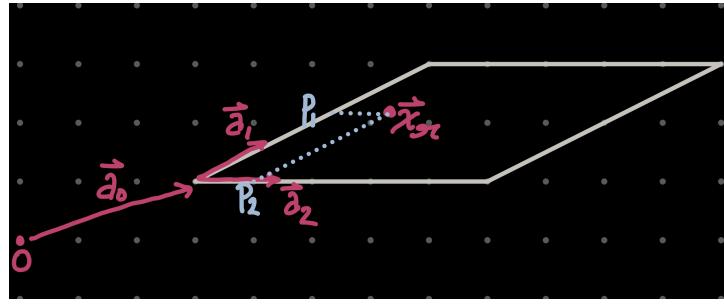


Figure 1: Depiction of point and basis vectors for plane  $\Pi$ .

If  $\mathbf{x}_\pi$  has 2D coordinates within plane  $\Pi$  with respect to basis vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$  given by the homogeneous representation  $\mathbf{p} = [p_2 \ p_1 \ 1]^T$ , then  $\mathbf{x}_\pi$  can be represented with respect to the planar coordinates  $\mathbf{p}$  as:

$$\mathbf{x}_\pi = \mathbf{a}_0 + p_1 \mathbf{a}_1 + p_2 \mathbf{a}_2 = [\mathbf{a}_2 | \mathbf{a}_1 | \mathbf{a}_0] \begin{bmatrix} p_2 \\ p_1 \\ 1 \end{bmatrix} = \mathbf{A}\mathbf{p} \quad | \quad \mathbf{A} = [\mathbf{a}_2 | \mathbf{a}_1 | \mathbf{a}_0]$$

If  $\mathbf{x}_\pi$  is seen through camera 2, which has projection matrix  $\mathbf{P}_2$ , as being at the 2D point given by homogeneous coordinates  $\mathbf{x}_2 = [x_2 \ y_2 \ 1]^T$ , then  $\mathbf{p}$  can be expressed in terms of  $\mathbf{x}_2$  as follows:

$$\mathbf{x}_2 \equiv \mathbf{P}_2 \mathbf{x}_\pi \equiv \mathbf{P}_2 \mathbf{A} \mathbf{p}$$

$$\therefore \mathbf{p} \equiv (\mathbf{P}_2 \mathbf{A})^{-1} \mathbf{x}_2$$

If  $\mathbf{x}_\pi$  is seen through camera 1, which has projection matrix  $\mathbf{P}_1$ , as being at the 2D point given by homogeneous coordinates  $\mathbf{x}_1 = [x_1 \ y_1 \ 1]^T$ , then  $\mathbf{x}_1$  can be expressed in terms of  $\mathbf{x}_2$  as follows:

$$\mathbf{x}_1 \equiv \mathbf{P}_1 \mathbf{x}_\pi \equiv \mathbf{P}_1 \mathbf{A} \mathbf{p}$$

$$\therefore \mathbf{x}_1 \equiv (\mathbf{P}_1 \mathbf{A} (\mathbf{P}_2 \mathbf{A})^{-1}) \mathbf{x}_2$$

Since  $\mathbf{A}$  is  $4 \times 3$  and  $\mathbf{P}_i$  is  $3 \times 4$ ,  $\mathbf{P}_i \mathbf{A}$  is  $3 \times 3$ , thus  $(\mathbf{P}_1 \mathbf{A} (\mathbf{P}_2 \mathbf{A})^{-1})$  is a  $3 \times 3$  and satisfies as a homography from  $\mathbf{x}_2$  to  $\mathbf{x}_1$ . Thus, there exists a homography  $\mathbf{H}$  such that  $\mathbf{x}_1 \equiv \mathbf{H} \mathbf{x}_2$  where camera 1 and camera 2 are observing a point on plane  $\Pi$ .

# The Direct Linear Transform

## Q1.2 Correspondences

$$\text{let: } \mathbf{x}_1^i = \mathbf{H}\mathbf{x}_2^i \quad | \quad i \in \{1 \dots N\}, \quad \mathbf{x}_j^i = [x_j^i \quad y_j^i \quad 1]^T, \quad \mathbf{x}_j^i \in \mathbf{x}_j$$

Let:  $\mathbf{A}_i \mathbf{h} = 0$  be an equivalent representation of the problem  $\mathbf{x}_1^i = \mathbf{H}\mathbf{x}_2^i$  where  $\mathbf{h}$  is  $\mathbf{H}$  reshaped into a column vector.

1. Since  $\mathbf{h}$  is, itself, a homogeneous representation, it has  $n_h - 1 = 3 * 3 - 1 = 8$  degrees of freedom.

2. Since each point pair produces 2 equations,  $\frac{8 \text{ DOF}}{2 \text{ equations per point pair}} = 4$  point pairs are required.

3.

$$\text{let: } \mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

$$\therefore \mathbf{x}_1^i = \mathbf{H}\mathbf{x}_2^i \rightarrow \begin{bmatrix} x_1^i \\ y_1^i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_2^i \\ y_2^i \\ 1 \end{bmatrix}$$

$$\therefore \begin{cases} x_1^i = \frac{h_{11}x_2^i + h_{12}y_2^i + h_{13}}{h_{31}x_2^i + h_{32}y_2^i + h_{33}} \\ y_1^i = \frac{h_{21}x_2^i + h_{22}y_2^i + h_{23}}{h_{31}x_2^i + h_{32}y_2^i + h_{33}} \end{cases} \rightarrow \begin{cases} (h_{31}x_2^i + h_{32}y_2^i + h_{33})x_1^i = h_{11}x_2^i + h_{12}y_2^i + h_{13} \\ (h_{31}x_2^i + h_{32}y_2^i + h_{33})y_1^i = h_{21}x_2^i + h_{22}y_2^i + h_{23} \end{cases}$$

$$\rightarrow \begin{cases} h_{31}x_1^i x_2^i + h_{32}x_1^i y_2^i + h_{33}x_1^i - h_{11}x_2^i - h_{12}y_2^i - h_{13} = 0 \\ h_{31}y_1^i x_2^i + h_{32}y_1^i y_2^i + h_{33}y_1^i - h_{21}x_2^i - h_{22}y_2^i - h_{23} = 0 \end{cases}$$

Extracting all  $h$  values gives that:

$$\mathbf{A}_i \mathbf{h} = 0 \quad | \quad \mathbf{A}_i = \begin{bmatrix} -x_2^i & -y_2^i & -1 & 0 & 0 & 0 & x_1^i x_2^i & x_1^i y_2^i & x_1^i \\ 0 & 0 & 0 & -x_2^i & -y_2^i & -1 & y_1^i x_2^i & y_1^i y_2^i & y_1^i \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}$$

$$\therefore \mathbf{A}_i = \begin{bmatrix} -x_2^i & -y_2^i & -1 & 0 & 0 & 0 & x_1^i x_2^i & x_1^i y_2^i & x_1^i \\ 0 & 0 & 0 & -x_2^i & -y_2^i & -1 & y_1^i x_2^i & y_1^i y_2^i & y_1^i \end{bmatrix}$$

4. • A trivial solution for  $\mathbf{h}$  will be any solution with 8 zeros (one for each degree of freedom) and 1 constant value indicating the scale (which we don't care about in this application). One example of this is  $\mathbf{h} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ h_{33}]^T$  for any  $h_{33}$ . Note: a solution where the scale is also zero and thus are 9 zeros would also be a trivial solution.
- Since  $\mathbf{h}$  only has 8 DOF,  $\mathbf{A}$  only needs to be  $8 \times 9$ , in which case  $\mathbf{A}$  being full rank means it must have full row rank (since there are fewer rows than columns) and  $\mathbf{A}$  will have full row rank (linearly independent rows) so long as all point pairs are unique and not scalar multiples of each other. Since  $\mathbf{A}$  is non-square and thus the row rank will always be less than than column rank, the dimension of the null space will always be at least colrank – rowrank =  $9 - 8 = 1$ . Thus, since finding a solution to  $\mathbf{A}\mathbf{h} = 0$  is akin to finding some  $\mathbf{h}$  which resides in the nullspace of  $\mathbf{A}$ , this means that there will be a nullspace for  $\mathbf{h}$  and thus there will be a non-trivial solution to  $\mathbf{h}$  so long as  $\text{colrank} > \text{rowrank}$ .
- Being non-square,  $\mathbf{A}$  won't have eigenvalues (that is, solutions to  $\mathbf{Av} = \lambda v$  because  $\mathbf{Av}$  and  $v$  would have different dimensions) but it will have at most  $\max \text{rank}(\mathbf{A}) = \max \text{rowrank}(\mathbf{A}) = 8$  singular values (square roots of eigenvalues of  $\mathbf{AA}^T$  or  $\mathbf{A}^T \mathbf{A}$ ). Since  $A$  has a nontrivial nullspace, there should be at least 1 zero eigenvalue.
- Likewise, being non-square,  $\mathbf{A}$  won't have eigenvectors (that is, solutions to  $\mathbf{Av} = \lambda v$  because  $\mathbf{Av}$  and  $v$  would have different dimensions) but it will have left and right singular vectors (eigenvectors of  $\mathbf{AA}^T$  and  $\mathbf{A}^T \mathbf{A}$ , respectively). Since  $A$  has a nontrivial nullspace, there should be at least 1 left singular vector corresponding to a zero eigenvalue; this left singular vector will be the solution  $h$ .

## Theory Questions

### Q1.3 Homography under rotation

$$\text{let: } \mathbf{x}_1 = \mathbf{K}_1 [\mathbf{I} \ 0] \mathbf{X}, \quad \mathbf{x}_2 = \mathbf{K}_2 [\mathbf{R} \ 0] \mathbf{X}$$

Based on the dimensions of matrices supplied,  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  and  $X$  is in homogeneous form. Thus,

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{K}_1 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}, \quad \mathbf{x}_2 = \mathbf{K}_2 \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \end{bmatrix} \mathbf{X} \\ \rightarrow \mathbf{x}_1 &= \begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} f_2 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix} \\ \rightarrow \mathbf{x}_1 &= \begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} f_2 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \\ \rightarrow \mathbf{x}_1 &= \mathbf{K}_1 \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}, \quad \mathbf{x}_2 = \mathbf{K}_2 \mathbf{R} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \\ \therefore \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} &= \mathbf{R}^{-1} \mathbf{K}_2^{-1} \mathbf{x}_2 = \mathbf{R}^T \mathbf{K}_2^{-1} \mathbf{x}_2 \end{aligned}$$

$$\text{Note: } \mathbf{K}_2^{-1} = \begin{bmatrix} 1/f_2 & 0 & 0 \\ 0 & 1/f_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \mathbf{x}_1 = \mathbf{K}_1 \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \mathbf{K}_1 \mathbf{R}^T \mathbf{K}_2^{-1} \mathbf{x}_2$$

Since  $\mathbf{K}_i$  is  $3 \times 3$  and  $\mathbf{R}$  is  $3 \times 3$ ,  $\mathbf{K}_1 \mathbf{R}^T \mathbf{K}_2^{-1}$  is a  $3 \times 3$  and satisfies as a homography from  $\mathbf{x}_2$  to  $\mathbf{x}_1$ . Thus, there exists a homography  $\mathbf{H}$  such that  $\mathbf{x}_1 \equiv \mathbf{H} \mathbf{x}_2$  where camera 1 and camera 2 are separated by a pure rotation.

#### Q1.4 Understanding homographies under rotation

From Q1.3, we know that  $\mathbf{x}_1 = \mathbf{K}_1 \mathbf{R}_\theta^T \mathbf{K}_2^{-1} \mathbf{x}_2$  or, more simply,  $\mathbf{x}_2 = \mathbf{K}_2 \mathbf{R}_\theta \mathbf{K}_1^{-1} \mathbf{x}_1$ ; thus, the homography between frames can be defined as:

$$\mathbf{H}_\theta = \mathbf{K}_2 \mathbf{R}_\theta \mathbf{K}_1^{-1}$$

Since the intrinsic parameters are constant,  $\mathbf{K}_1 = \mathbf{K}_2 = \mathbf{K}$

$$\therefore \mathbf{H}_\theta = \mathbf{K} \mathbf{R}_\theta \mathbf{K}^{-1}$$

This is a similarity transformation, thus

$$\mathbf{H}_\theta^2 = \mathbf{K} \mathbf{R}_\theta^2 \mathbf{K}^{-1}$$

For a camera rotating about its center about one axis (say, z), the rotation matrix can be defined as:

$$\mathbf{R}_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \mathbf{R}_\theta^2 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos^2(\theta) - \sin^2(\theta) & -2\cos(\theta)\sin(\theta) & 0 \\ 2\cos(\theta)\sin(\theta) & \cos^2(\theta) - \sin^2(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \cos(\theta)\sin(\theta) = \frac{1}{2}\sin(2\theta), \text{ and } \cos^2(\theta) - \sin^2(\theta) = \cos(2\theta)$$

$$\mathbf{R}_\theta^2 = \begin{bmatrix} \cos(2\theta) & -\sin(2\theta) & 0 \\ \sin(2\theta) & \cos(2\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{R}_{2\theta}$$

$$\therefore \mathbf{H}_\theta^2 = \mathbf{K} \mathbf{R}_{2\theta} \mathbf{K}^{-1} = \mathbf{H}_{2\theta}$$

Thus,  $\mathbf{H}_\theta^2$  is the homography corresponding to a rotation twice the size of that of  $\mathbf{H}_\theta$ .

### **Q1.5 Limitations of the planar homography**

The planar homography is not completely sufficient to map any arbitrary scene image to another viewpoint because if all the points in a scene aren't confined to a plane and there is translation involved in the extrinsic matrix between the viewpoints, the product of the intrinsic and extrinsic matrices won't be square and thus won't be invertible, meaning a homography can't be established between the two views. More simply, not all viewpoints are separated by a pure rotation and not all scenes contain points confined to a plane.

### Q1.6 Behavior of lines under perspective projections

Consider  $\mathbf{X}$  to be a point in 3D space given by the homogeneous coordinates  $\mathbf{X} = [X \ Y \ Z \ 1]^T$ , which is located on a line  $\mathcal{L}$  defined by a 3D point with homogeneous coordinates  $\mathbf{a}_0 = [a_{01} \ a_{02} \ a_{03} \ 1]^T$  and a basis vector given by homogeneous coordinates  $\mathbf{a}_1 = [a_{11} \ a_{12} \ a_{13} \ 0]^T$  as shown in the figure below.

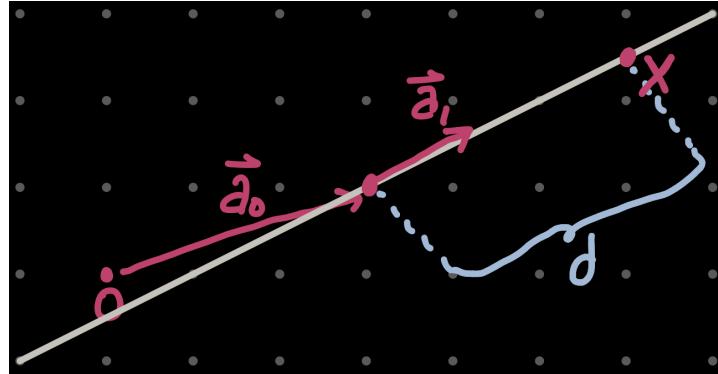


Figure 2: Depiction of point and basis vector for line  $\mathcal{L}$ .

If  $\mathbf{X}$  is a distance  $d$  away from  $\mathbf{a}_0$  along  $\mathbf{a}_1$ ,  $\mathbf{X}$  can be defined as:

$$\mathbf{X} = \mathbf{a}_0 + d\mathbf{a}_1 = \mathbf{A}\mathbf{D} \quad | \quad \mathbf{A} = [\mathbf{a}_1 | \mathbf{a}_0], \quad \mathbf{D} = \begin{bmatrix} d \\ 1 \end{bmatrix}$$

Under a perspective projection  $\mathbf{x} = \mathbf{P}\mathbf{X}$ ,

$$\begin{aligned} \mathbf{x} &= \mathbf{P}\mathbf{A}\mathbf{D} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{01} \\ a_{12} & a_{02} \\ a_{13} & a_{03} \\ 0 & 1 \end{bmatrix} \mathbf{D} \\ &\rightarrow \mathbf{x} = \begin{bmatrix} fa_{11} & fa_{01} \\ fa_{12} & fa_{02} \\ a_{13} & a_{03} \end{bmatrix} \mathbf{D} \\ &\rightarrow \mathbf{x} \equiv \begin{bmatrix} \frac{fa_{11}}{a_{13}} & \frac{fa_{01}}{a_{03}} \\ \frac{fa_{12}}{a_{13}} & \frac{fa_{02}}{a_{03}} \\ 0 & 1 \end{bmatrix} \mathbf{D} \end{aligned}$$

This expression therefore states that if a point  $\mathbf{X}$  lies at position  $d$  along on a line in 3D space defined by the point  $\mathbf{a}_0$  and vector  $\mathbf{a}_1$ , then the 2D point  $\mathbf{x}$  seen through perspective projection

$P$  with focal length  $f$  will be found lying at position  $d$  along on a corresponding line in 2D space defined by a point  $\mathbf{b}_0 = \left[ \frac{fa_{01}}{a_{03}} \quad \frac{fa_{02}}{a_{03}} \quad 1 \right]^T$  and vector  $\mathbf{b}_1 = \left[ \frac{fa_{11}}{a_{13}} \quad \frac{fa_{12}}{a_{13}} \quad 0 \right]^T$ .

## 2 Computing Planar Homographies

### Feature Detection and Matching

#### Q2.1.1 FAST Detector

While the FAST detector and Harris detector both broadly compare pixel intensities to the intensities of neighboring pixels in all directions and detect a corner where that difference is great for many directions, FAST simply looks at the intensities of pixels along a ring surrounding a candidate pixel rather effectively than shifting a window in all 8 directions (computing the eigenvalues of the sum of matrices of derivatives for shift multiple directions around a pixel). As a result, FAST requires drastically fewer calculations, most of which are just simple comparisons and, unlike Harris, these can be ordered such that fewer computations will be performed for center pixels which obviously could not pass the feature test (there are no pixels separated by the appropriate distance which are above or below the required threshold intensity). Thus, FAST should detect features (and ignore non-features) considerably faster than the Harris corner while offering similar performance / detecting similar features.

### **Q2.1.2 BRIEF Descriptor**

- Filterbanks allow one to record the response of an image (or a region of an image) to a small (on the order of 10) series of filters meant to accentuate certain features, such as edges. In contrast, the BRIEF descriptor records a large set (hundreds: 128,256,etc.) of binary comparisons of pairs of pixel intensities from a sampling geometry of the patch around a feature keypoint, meaning the BRIEF descriptions are more specific (higher dimension) than a filter response but have a less intuitive meaning. The big selling point of BRIEF is that since it represents each feature as a binary vector of pixel comparison results, it is very both computationally and memory efficient.
- Each of the filter responses could be collected into a vector to act as a fingerprint of a feature, in much the same way as textures were vectorized as "pixel tubes" in homework 1 where each element in a vector for a pixel recorded the corresponding pixel's value in the response to a filter. In theory, these vectors could be used to describe a feature, however, much of the structural information about neighbor relationships would be lost without a large and comprehensive filterbank; so, this technique is much better suited to identify textures which have a general composition rather than features which have a specific structure, especially since such a large and comprehensive filterbank would massively inflate the computation time and memory usage which, for many of the applications where feature descriptors are desirable, would be prohibitive.

### **Q2.1.3 Matching Methods**

- The Hamming distance between two binary vectors is simply the number of locations at which they differ in value, that is: the number of 1s in the bitwise XOR of the two binary vectors being considered. Nearest Neighbor is a search method used to efficiently find which of a set of vectors is closest to a desired vector according to some distance metric. Using the Hamming distance as this metric for a Nearest Neighbor search can be used to efficiently determine which of the features in an image being evaluated most closely matches a given feature in a source image according to their BRIEF descriptors which are binary strings.
- Since the Euclidean distance metric between two binary vectors will assume that they're vectors of ordinary numbers, it will require considerably more fundamental operations than Hamming distance for two binary vectors (N subtractions + N multiplications (squaring) + (N-1) additions vs N xor bit operations + N additions (a count)), allowing the matching process to be significantly more computationally and memory efficient when using the Hamming distance.

#### Q2.1.4 Feature Matching

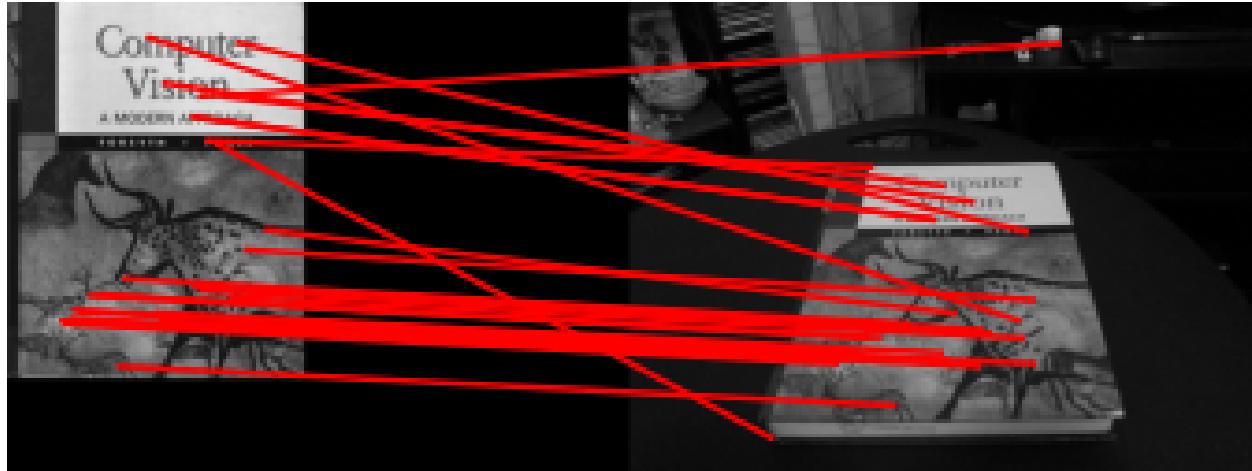
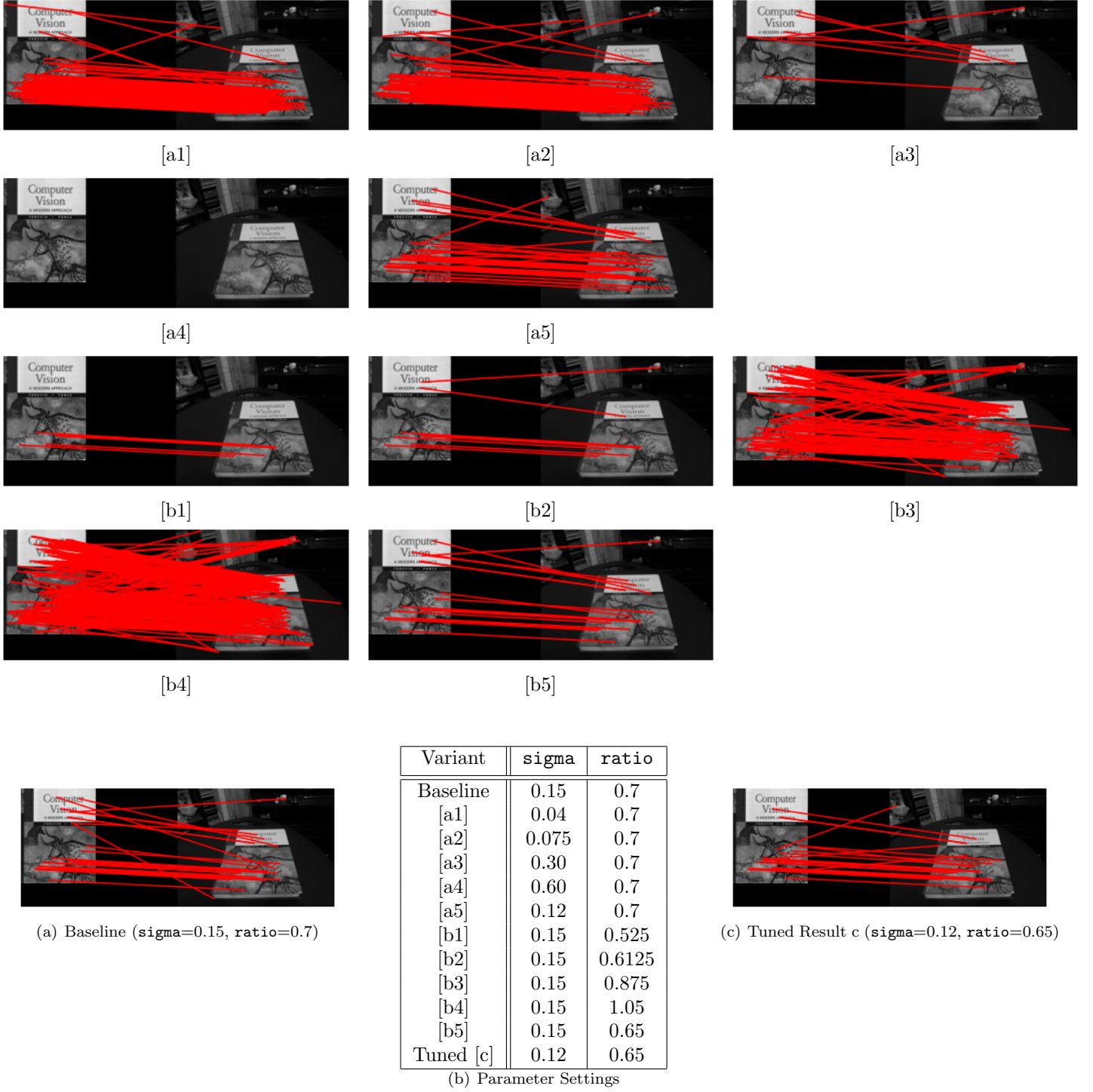


Figure 3: Image correspondences produced from `q2_1_4.py` for `sigma=0.15, ratio=0.7`.

### Q2.1.5 Feature Matching Parameter Tuning

In this parameter tuning, each parameter was varied independently with the other value fixed at its baseline value. `sigma` was tested at 1/4, 1/2, 2, and 4 times its baseline value and `ratio` was tested at 0.75, 0.875, 1.25, and 1.5 times its baseline value. The results of these tests were then qualitatively evaluated to determine a rough "best" value for the parameter in question which was then evaluated in a fifth test. Lastly, a final tuned trial was conducted which combined both of these "best" parameter values.

Figure 4: Results of each stage in parameter tuning



`sigma` controls the number of features detected, a lower `sigma` more detected more (lower quality) features and lead to increased computation time. `ratio` seemed to control how stingy the matching algorithm was with finding a match, a decreased `ratio` created fewer higher accuracy matches.

#### Q2.1.6 BRIEF and Rotations

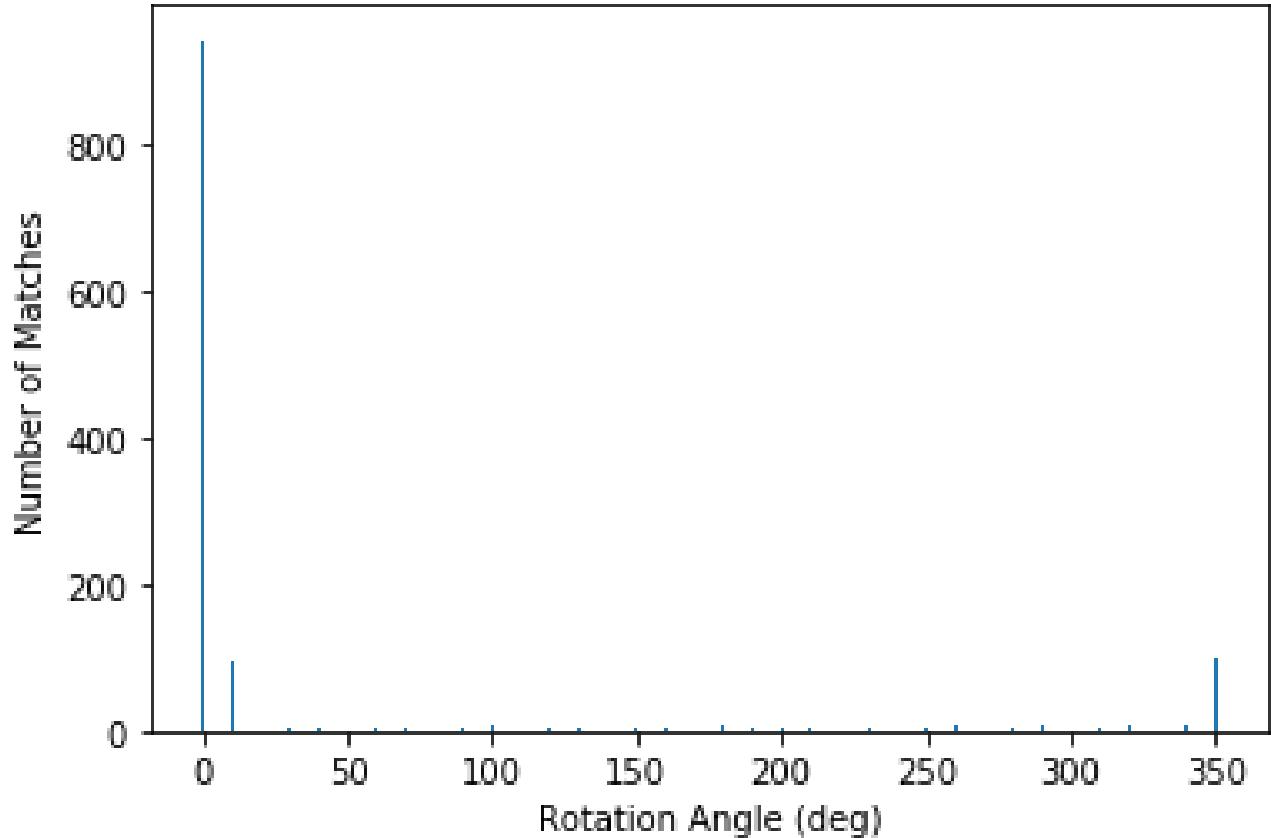
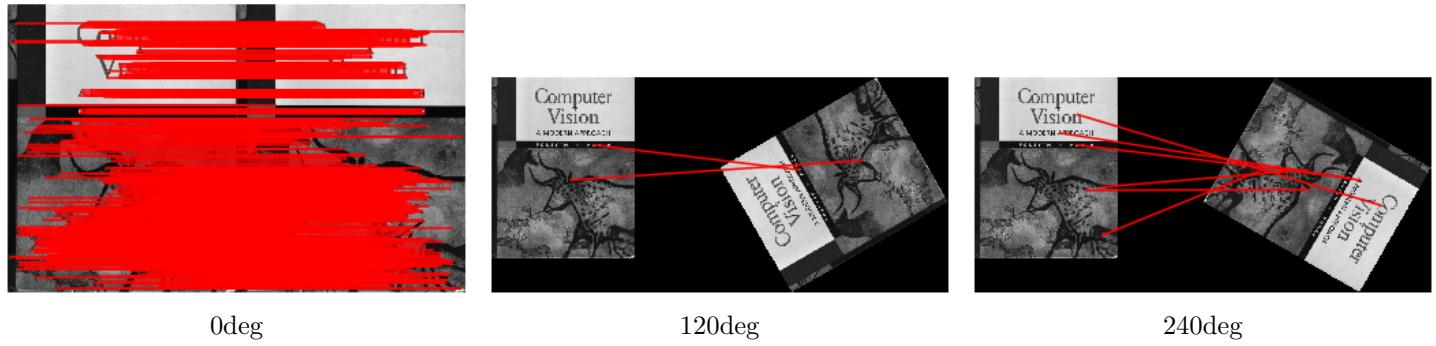


Figure 5: Histogram of Matches for Each Rotation Angle Trialed

Figure 6: Matching Results for 3 Different Orientations



Under rotation, BRIEF matches drastically fewer points, though it does still seem to match them to roughly the appropriate corresponding areas. This is likely because the BRIEF feature descriptor is not truly rotation invariant. Seeing as BRIEF works by randomly sampling and comparing the intensity of pairs of points within a sampling geometry which is initialized once, it makes sense that a rotated image would align differently underneath this sampling geometry and thus yield mostly different feature descriptors, making BRIEF not rotation invariant.

## Automated Homography Estimation and Warping

### Q2.2.4 Putting it together

As stated, it's been observed that the image is being warped but not to the correct location. This is likely due to the fact that the size and aspect ratio of the Harry Potter book template are not the same as those of the CV Book cover template (since the CV Book cover template is what the homography was built for). To remedy this, one should simply crop and adjust the Harry Potter book template to have the same dimensions as the CV Book cover template before warping.

5

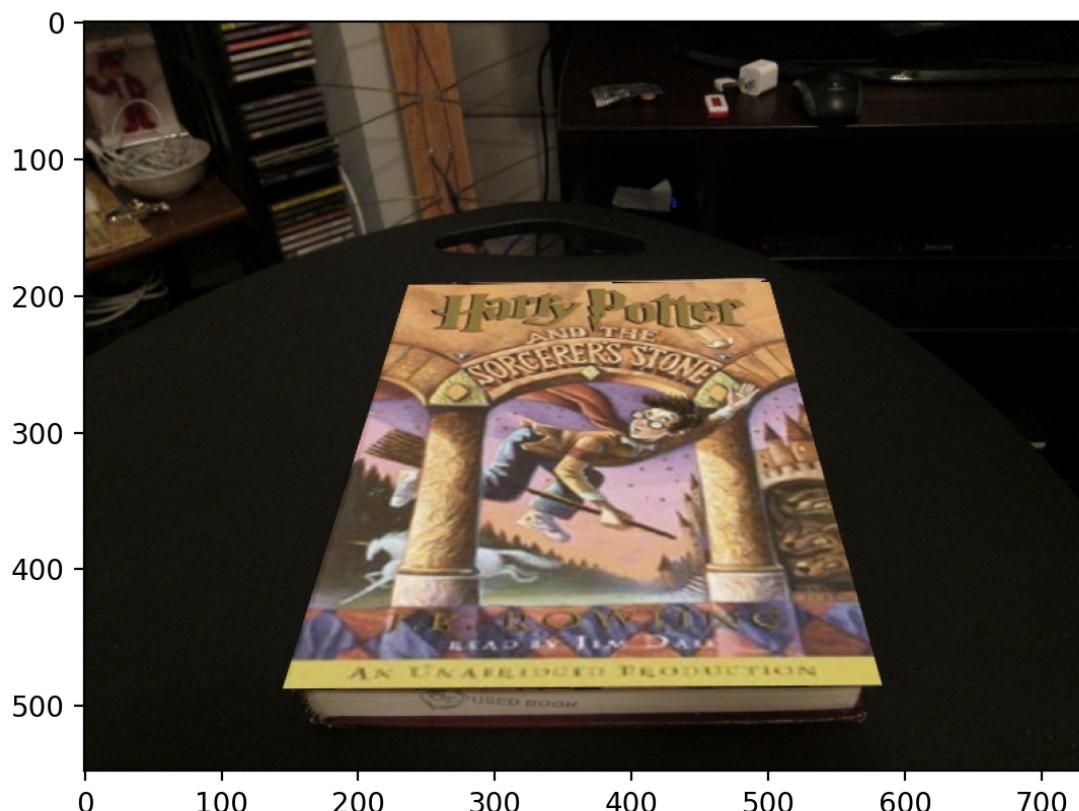
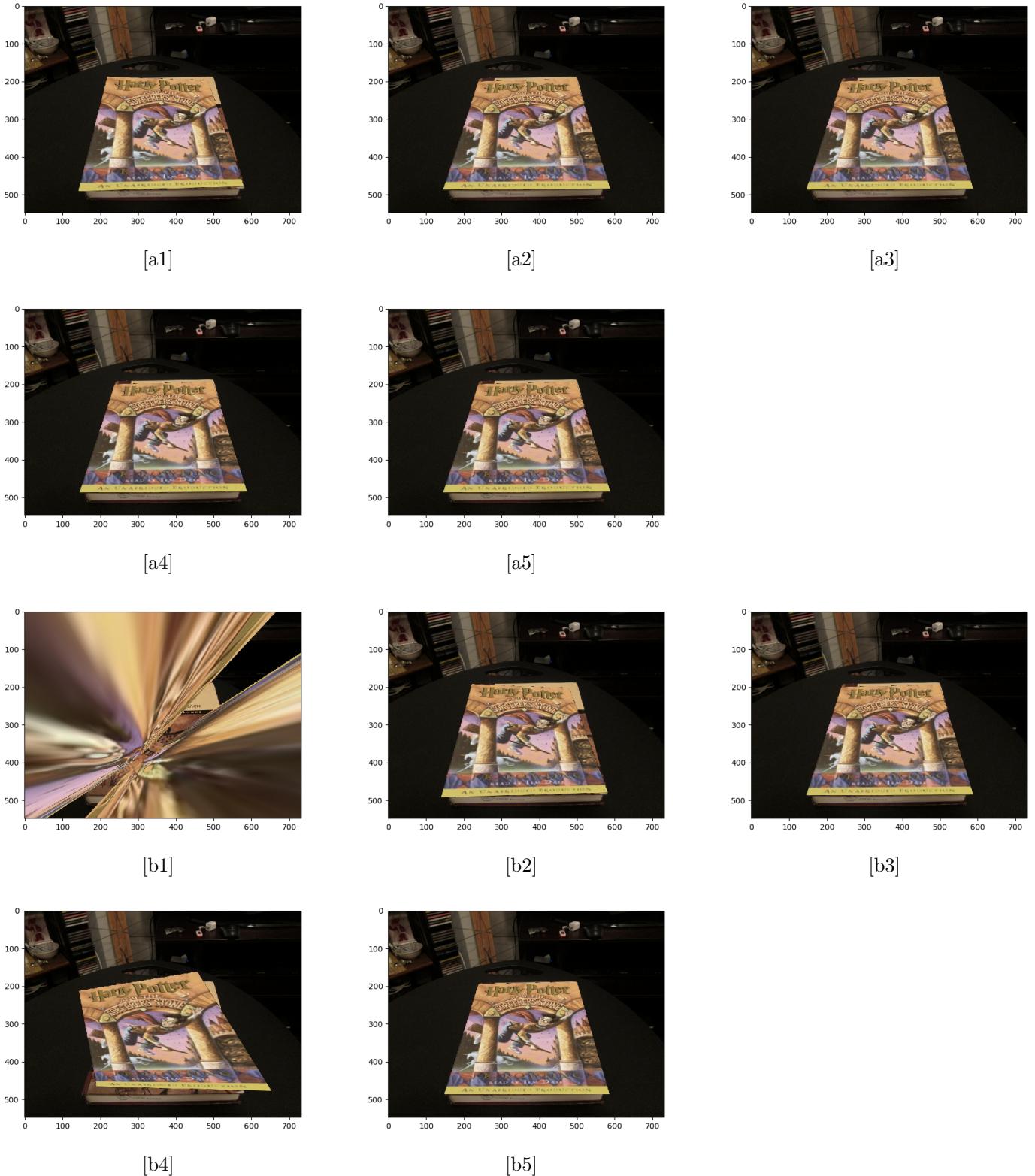


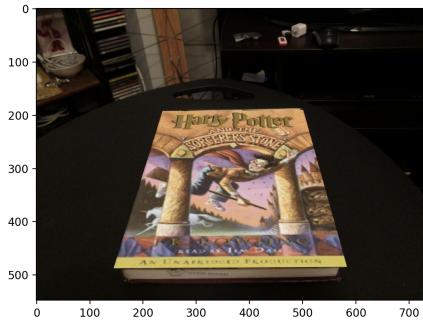
Figure 7: Composed (HarryPotterized) Image Result

## Q2.2.5 RANSAC Parameter Tuning

In this parameter tuning, each parameter (`max_iters` and `inlier_tol`) was varied independently with the other value fixed at its baseline value. All other hyperparameters (`sigma` and `ratio`) were left at their default values. The results of these tests were then qualitatively evaluated to determine a rough "best" value for the parameter in question which was then evaluated in a fifth test. Lastly, a final tuned trial was conducted which combined both of these "best" parameter values.

Figure 8: Results of each stage in RANSAC parameter tuning

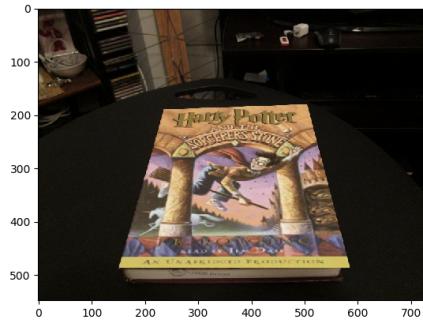




(a) Baseline

Variant	max_iters	inlier_tol	Time* [s]
Baseline	500	2.0	6.42
[a1]	10	2.0	6.28
[a2]	100	2.0	6.36
[a3]	1000	2.0	6.57
[a4]	10000	2.0	8.83
[a5]	250	2.0	6.38
[b1]	500	0.025	6.50
[b2]	500	0.5	6.43
[b3]	500	5	6.86
[b4]	500	50	6.42
[b5]	500	1.0	6.88
Tuned [c]	250	1.0	6.80

(b) Parameter Settings (\*note: time includes setup overhead)



(c) Tuned Result c

This tuned model was then combined with the tuned parameters `sigma` and `ratio` worked out in Q2.1.5 to determine produce the following result [d]

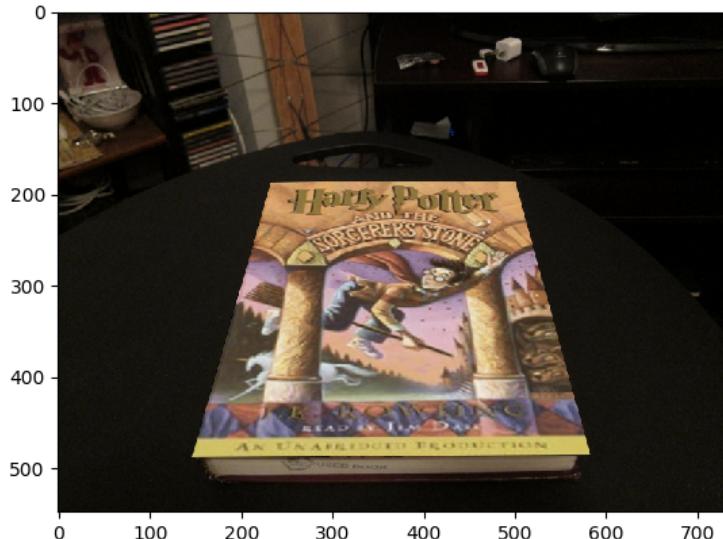


Figure 9: Tuned HarryPotterized Image

From this experiments, it's evident that increasing `max_iter` increases the fit of the model to the image but only marginally and seemingly not to a degree worth the marginal but present increase in computation time. In fact, one could get away with a substantially low `max_iter` on the order of 10 and still get a reasonable fit while saving noticeably on performance. Likewise, it's evident from these results that `inlier_tol` has a pretty substantial impact on fit while not directly (or at least linearly) affecting computation time, which makes sense since its mainly just used as a threshold comparator in this implementation of RANSAC. Excessively low values of `inlier_tol` prevent there from being enough points considered when the final model is built to get a close fit; whereas, overly high values of `inlier_tol` cause even poorly fitting homographies to register as satisfactory, thus allowing a result to be skewed towards outliers.