16-720 Homework 2: Augmented Reality with Planar Homographics

Austin Windham

October 5, 2023

1 Homographies

1.1 Q 1.1: Homography

If there are matrices P_1, P_2 for two camera planes and plane $\mathbf{\Pi}$, then the point on the plane that is multiplied by the P matrix could be written as $[x, y, 0, 1]$ if the z-axis is taken as perpendicular to the plane. This effectively makes P a 3x3 matrix that is multiplying coordinates $[x, y, 1]$. This P is then invertible, and if this is done to get the inverse of P_1 specifically, then the inverse can multiply the P_2 matrix to make a homography from x_2 to x_1.

1.2 Q 1.2 Correspondences

1. How many degrees of freedom does h have?

There are 8 degrees of freedom because the homographic equation 1 can be written out in equation 2 below:

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

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \tag{2}$$

So, then **h** values of the matrix can be written in a 9x1 array like below

$$h = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} \tag{3}$$

There are 9 variables, but the first 8 variables are normalized by the last one h_{33}, so $\boldsymbol{h}$ will have 8 degrees of freedom.

2. How many point pairs are required to solve h?

Only four point pairs are required to solve **h** .

3. Derive $\mathbf{A_i}$

If the matrix in equation 2 is multiplied out, then the following systems of equations is formed:

$$x_1 = h_{11}x_2 + h_{12}y_2 + h_{13} \tag{4}$$
$$y_1 = h_{21}x_2 + h_{22}y_2 + h_{23} \tag{5}$$
$$1 = h_{31}x_2 + h_{32}y_2 + h_{33} \tag{6}$$

Then by dividing equations 4 and 5 by equation 6 since equation 6 is just equal to one, these equations form.

$$\frac{x_1}{1} = \frac{h_{11}x_2 + h_{12}y_2 + h_{13}}{h_{31}x_2 + h_{32}y_2 + h_{33}} \tag{7}$$
$$\frac{y_1}{1} = \frac{h_{21}x_2 + h_{22}y_2 + h_{23}}{h_{31}x_2 + h_{32}y_2 + h_{33}} \tag{8}$$

By multiplying the denominators and grouping terms to one side, these two equations then form:

$$-h_{11}x_2 - h_{12}y_2 - h_{13} + h_{31}x_1x_2 + h_{32}x_1y_2 + h_{33}x_1 = 0 \tag{9}$$

$$-h_{21}x_2 - h_{22}y_2 - h_{23} + h_{31}y_1x_2 + h_{32}y_1y_2 + h_{33}y_1 = 0 \tag{10}$$

The above equations can be then expressed as $\mathbf{A_i h = 0}$, so $\mathbf{A_i}$ is derived as:

$$A_i = \begin{bmatrix} -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_1x_2 & x_1y_2 & x_1 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & y_1x_2 & y_1y_2 & y_1 \end{bmatrix} \tag{11}$$

4. Solving for h questions.

The trivial solution is when $\mathbf{h} = 0$. There is a non-trivial solution, so that means that A is not full rank. The impact of this is that there is a nontrivial h that is the eigen vector of $A^T A$ that corresponds to the least eigen value of $A^T A$.

1.3 Q 1.3 Homography Under Rotation

For the first camera, an inverse projection matrix to convert x_1 to X can be found by multiplying by the inverse of K_1 and the transpose of the $[I|0]$ matrix, $P = \left[\begin{smallmatrix} I & 0 \end{smallmatrix}\right]^T K_1^{-1}$. Since $X = Px_1$, Px_1 can be substituted in for X to get $\boldsymbol{H_{1to2}} = K_2[R|0]P$, which is the homographic matrix that relates x_1 to x_2, and the homographic matrix is invertible, so the inverse of that homographic matrix is the homographic matrix that satisfies $x_1^i \equiv Hx_2^i$

1.4 Q 1.4 Understanding Homographies Under Rotation

If **K** is constant and about the z-axis, then the rotation matrix by angle θ is given below. If you rotate twice, then it is the equivalent of multiplying the matrix by itself. With the trig identities given below, $\mathbf{R^2}$ is derived as given below, which shows that $\mathbf{H^2}$ is corresponding to a rotation of 2θ.

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$\sin 2\theta = 2\sin\theta\cos\theta \tag{13}$$

$$\cos 2\theta = cos^2\theta - \sin^2\theta \tag{14}$$

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

1.5 Q 1.5 Limitations of the Planar Homography

Planar homography is not sufficient to map any arbitrary scene image to another viewpoint because it assumes a planar scene and cannot map the variations in depth and non-planar deformations present in more complex 3D scenes.

1.6 Q 1.6 Behavior of Lines Under Perspective Projections

To show that perspective projection preserve lines, consider that there is a point in the real world frame defined by x, y, and z that can be projected to the 2D point given by x_p and y_p. So, a function for the line in $3D$ where $l = [a, b, c]$ is given below:

$$l \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0 \tag{16}$$

And the corresponding 2D coordinates are given below.

$$l \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \\ 1 \end{bmatrix} = 0 \tag{17}$$

And if $\frac{x}{z} = x_p$ and $\frac{y}{z} = y_p$, the equation becomes:

$$l \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = 0 \tag{18}$$

This indicates that the perspective projection preserves lines.

2 Computing Planar Homographies

2.1 Feature Detection and Matching

2.1.1 Q FAST Detector

FAST detector selects an area around a pixel and compares that pixel's intensity to the intensities of the pixels in the selected area in order to detect corners; whereas, Harris corner detector determines corners by analyzing pixel intensity gradients. Because of these approaches FAST is much more computationally faster that Harris.

2.1.2 Q BRIEF Descriptor

BRIEF applies a Gaussian filter to a region that is formed around a selected point and generates binary strings based on pixel intensity comparisons in order to match features. Filterbanks are just mainly used for feature extraction in our case. Gaussian is the preferred filterbank to use, but the others could also technically still be used.

2.1.3 Q Matching Methods

Hamming distance can be used with BRIEF descriptor to calculate the dissimilarity between binary strings that represents points in two different images. Then, nearest neighbor is used to find the smallest hamming distance to determine the matched point. Hamming distance is preferred to Euclidean distance because BRIEF descriptors are binary and hamming distance is much more efficient both in speed and in memory than Euclidean distance when working with binary strings.

2.1.4 Feature Matching

Feature matching image with default parameters sigma = .15 and ratio = .7 is below.

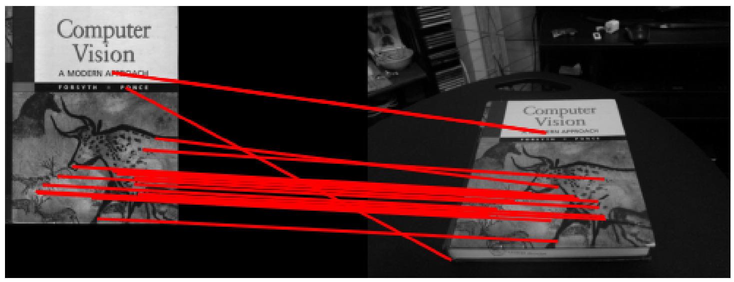
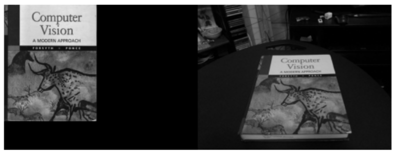


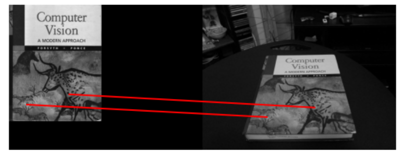
Figure 1: Feature Matching Result

2.1.5 Feature Matching Parameter Tuning

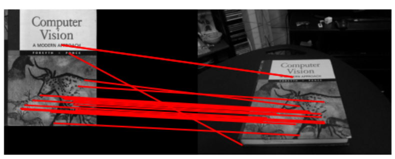
I used the default $sigma = 0.15$ and tuned $ratio$ to four different values. The results are shown below:



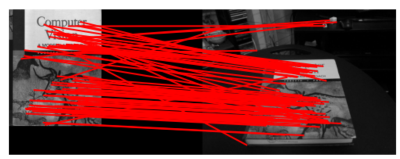
(a) sigma=0.15 ratio=0.25



(b) sigma=0.15 ratio=0.5

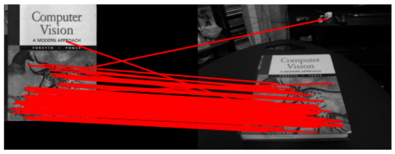


(c) sigma=0.15 ratio=0.7

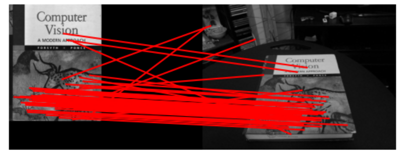


(d) sigma=0.15 ratio=0.85

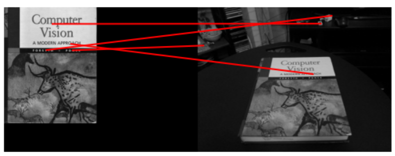
Then, I used the default $ratio = 0.7$ and tuned sigma to four different values producing the results below.



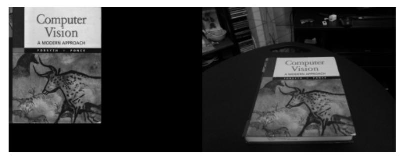
(e) sigma=0.05 ratio=0.7



(f) sigma=0.10 ratio=0.7



(g) sigma=0.40 ratio=0.7



(h) sigma=0.60 ratio=0.7

From the results, when decreasing the ratio, the number of matches decreases, and when increasing the ratio the number of matches increases. This is because less point pairs are considered matched when the ratio is lower. On the other hand, sigma has the opposite effect, when sigma is decreased, there are more matched points. This happens because sigma is the threshold for corner detection and lowering sigma lowers the threshold, so more points are considered matches.

2.1.6 BRIEF and Rotations

In this case, with BRIEF feature descriptor $ratio = 0.7$, The visualized match results for rotation angle $30, 170, 300$, images are shown below:

The histogram is also shown below.

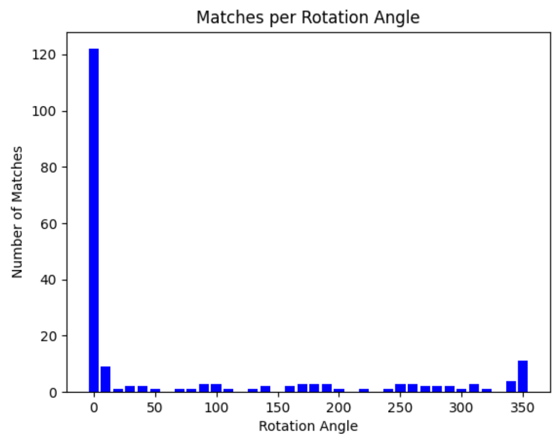
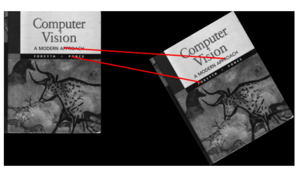
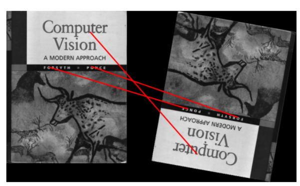


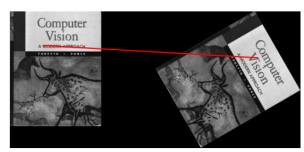
Figure 2: Histogram from Rotations



(a) Rotation Angle = 30



(b) Rotation Angle = 170



(c) Rotation Angle = 300

Brief descriptor is able to make the most matches by far when the pictures are in the same rotation. The number of matches drops sharply after just a 10 degree rotation. This is because BRIEF descriptor creates binary descriptors for each patch at randomly selected pixel locations, and when the image is rotated these points are going to give different descriptors for the same selected patch and not match its comparison.

2.2 Homography Computation, Normalization and RANSAC

2.2.4 Putting it together

Mapped image with default RANSAC parameters is shown below:

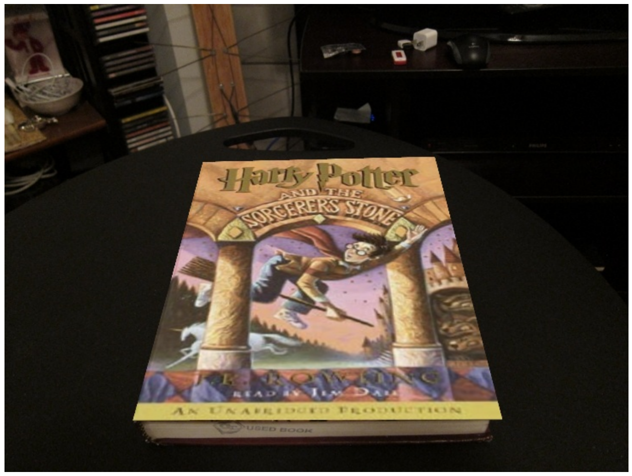
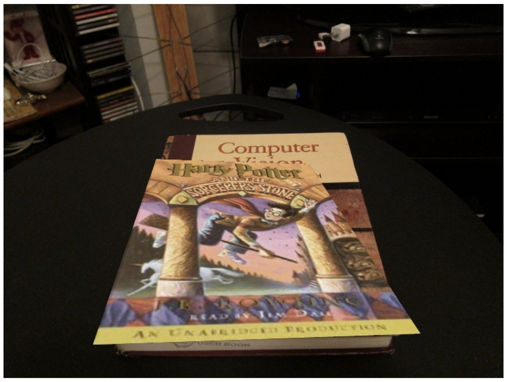


Figure 3: Mapped Harry Potter

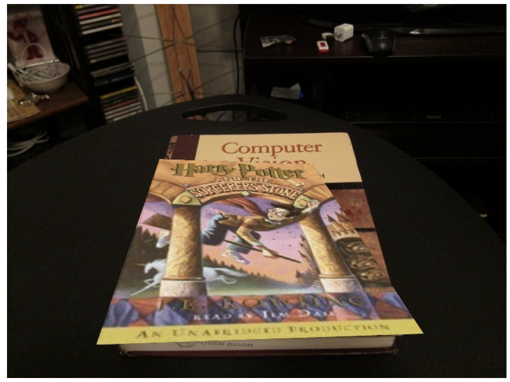
In order to get this fit, the image needed to be resized to match the size of CV book cover image.

2.2.5 RANSAC Parameter Tuning

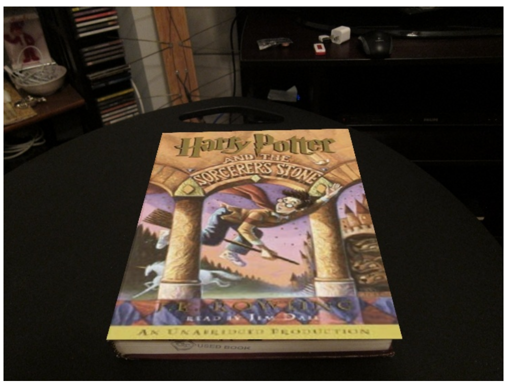
As you can see from the above image in 2.2.4, Harry Potter was mapped very accurately with the default parameters of 500 iterations and a tolerance of 2. I kept tolerance at 2 and varied the number of iterations.



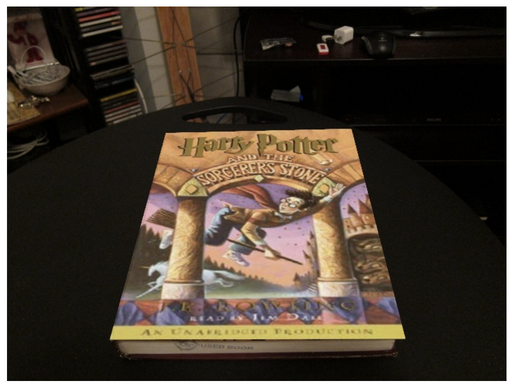
(a) iteration number=1, tolerance=2



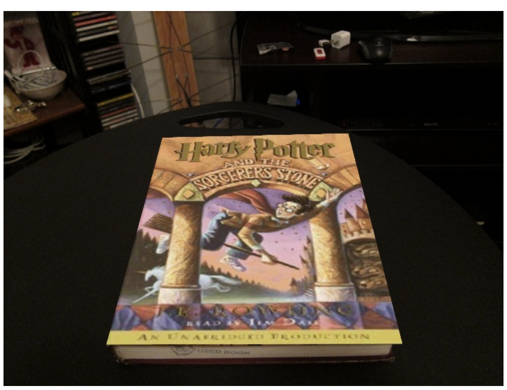
(b) iteration number=10, tolerance=2



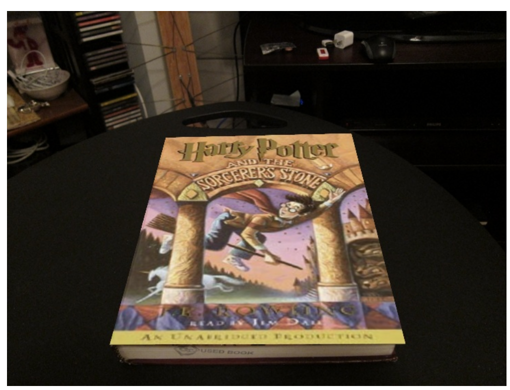
(c) iteration number=50, tolerance=2



(d) iteration number=100, tolerance=2



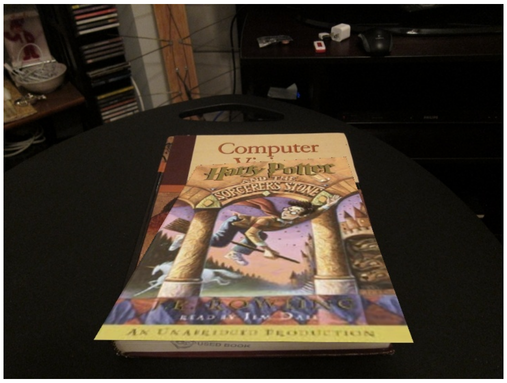
(e) iteration number=500, tolerance=2



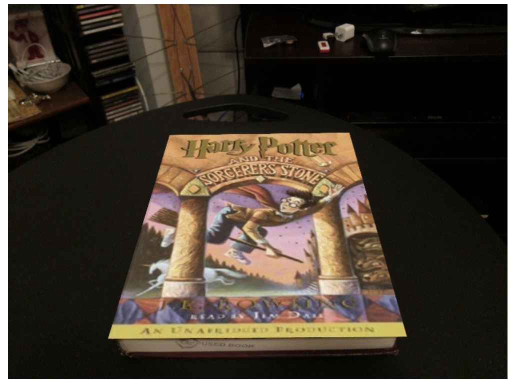
(f) iteration number=1000, tolerance=2

As the number of iterations decrease, the homography generated should be less suitable. And as shown, Harry Potter was mapped very well except for when the iteration number was 10 and 1. Even with 50 iterations, the homography is still suitable. The result was expected because more iterations means more chances to find a better homography.

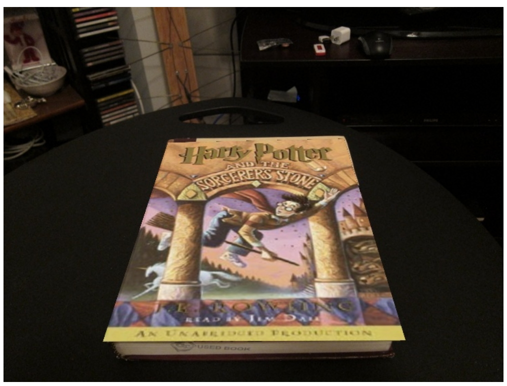
I then returned the number of iterations to the default of 500 and varied the tolerance, and the results are below.



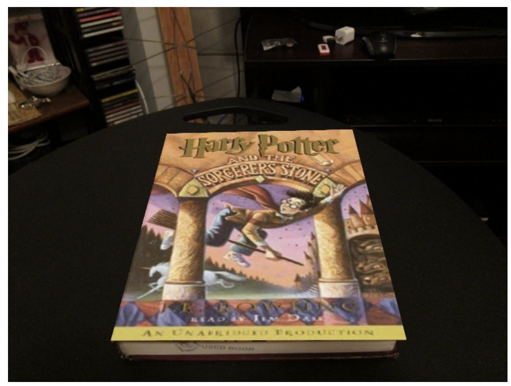
(g) iteration number=500, tolerance=.1



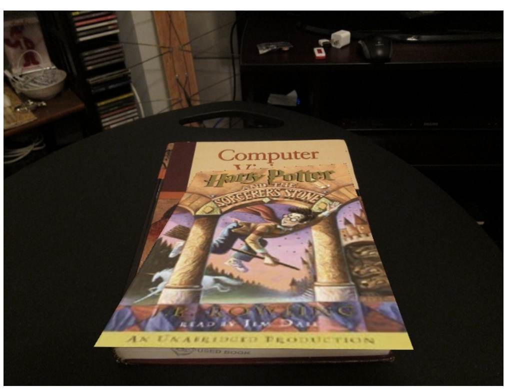
(h) iteration number=500, tolerance=0.5



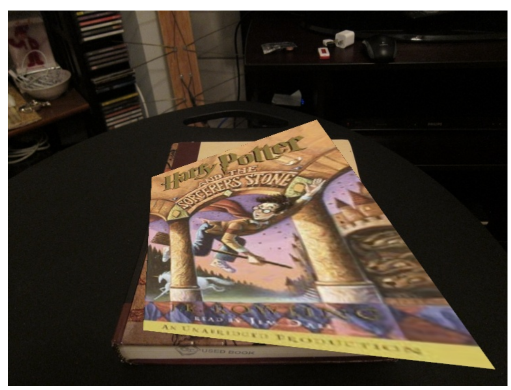
(i) iteration number=500, tolerance=1



(j) iteration number=500, tolerance=5



(k) iteration number=500, tolerance=10



(l) iteration number=500, tolerance=20

From the results, when tolerance increased to 10 and 20, the fit worsened as the tolerance was greater. This is because when the tolerance is greater more corresponding point locations are going to fall within the range. This causes the best homography not to be chosen because inaccurate homographies can still cause points to fall within the range when the range is large enough. On the other hand, when the tolerance is very small such as 0.1, few points are able to fall within the range, so a bad homography can be chosen because there were no homographies that could fit a lot of points within such a small range.

3 Creating your Augmented Reality Application

3.1 Incorporating Video

The left, center, and right images of frames from my ar.avi are below. As shown, the script was able to map the movie on to the book correctly.

Figure 4: AR Left Image

Figure 5: AR Center Image

Figure 6: AR Right Image

4 Extra Credit

4.2 4.2x Create a Simple Panorama

Left and right images are below with the panorama underneath them. The images were captured with my IPhone outside my Pittsburgh apartment.

(a) Left Image

(b) Right Image

Figure 7: Panorama Image