

16-720 Computer Vision: Augmented Reality with Planar Homographies

Yanjia Duan

October 2020

Q1.1 Homography

For a camera that projects \mathbf{x}_π to \mathbf{x}_1 and a camera that projects \mathbf{x}_π to \mathbf{x}_2 :

$$\begin{aligned}\mathbf{x}_1 &\equiv \mathbf{P}_1 \mathbf{x}_\pi \\ \mathbf{x}_2 &\equiv \mathbf{P}_2 \mathbf{x}_\pi\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbf{x}_2 &\equiv \mathbf{P}_2 \mathbf{x}_\pi \quad \Rightarrow \quad \mathbf{x}_\pi \equiv \mathbf{P}_2^{-1} \mathbf{x}_2 \\ \therefore \mathbf{x}_1 &\equiv \mathbf{P}_1 \mathbf{P}_2^{-1} \mathbf{x}_2 \equiv \mathbf{H} \mathbf{x}_2\end{aligned}$$

So, there exists a homography \mathbf{H} , which is $\mathbf{P}_1 \mathbf{P}_2^{-1}$.

Q1.2 Correspondences

1. \mathbf{H} is a 3×3 matrix so \mathbf{h} has 9 elements. Since in homogeneous coordinates, the scale doesn't matter. The degrees of freedom of \mathbf{h} is 8.

2. Since \mathbf{h} has 8 degrees of freedom, we need 8 equations to solve it. Each pair of 2D points can offer 2 equations. So we need 4 point pairs.

3. Suppose

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

$$\mathbf{x}_1^i \equiv \mathbf{H}\mathbf{x}_2^i$$

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

$$x_1^i = \frac{h_{11}x_2^i + h_{12}y_2^i + h_{13}}{h_{31}x_2^i + h_{32}y_2^i + h_{33}} \quad y_1^i = \frac{h_{21}x_2^i + h_{22}y_2^i + h_{23}}{h_{31}x_2^i + h_{32}y_2^i + h_{33}}$$

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

We plug in 4 points ($i \in \{1, 2, 3, 4\}$) to form the equation:

$$\begin{bmatrix} x_2^1 & y_2^1 & 1 & 0 & 0 & 0 & -x_2^1 x_1^1 & -y_2^1 x_1^1 & -x_1^1 \\ 0 & 0 & 0 & x_2^1 & y_2^1 & 1 & -x_2^1 y_1^1 & -y_2^1 y_1^1 & -y_1^1 \\ x_2^2 & y_2^2 & 1 & 0 & 0 & 0 & -x_2^2 x_1^2 & -y_2^2 x_1^2 & -x_1^2 \\ 0 & 0 & 0 & x_2^2 & y_2^2 & 1 & -x_2^2 y_1^2 & -y_2^2 y_1^2 & -y_1^2 \\ x_2^3 & y_2^3 & 1 & 0 & 0 & 0 & -x_2^3 x_1^3 & -y_2^3 x_1^3 & -x_1^3 \\ 0 & 0 & 0 & x_2^3 & y_2^3 & 1 & -x_2^3 y_1^3 & -y_2^3 y_1^3 & -y_1^3 \\ x_2^4 & y_2^4 & 1 & 0 & 0 & 0 & -x_2^4 x_1^4 & -y_2^4 x_1^4 & -x_1^4 \\ 0 & 0 & 0 & x_2^4 & y_2^4 & 1 & -x_2^4 y_1^4 & -y_2^4 y_1^4 & -y_1^4 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \mathbf{0}$$

Therefore,

$$\mathbf{A} = \begin{bmatrix} x_2^1 & y_2^1 & 1 & 0 & 0 & 0 & -x_2^1 x_1^1 & -y_2^1 x_1^1 & -x_1^1 \\ 0 & 0 & 0 & x_2^1 & y_2^1 & 1 & -x_2^1 y_1^1 & -y_2^1 y_1^1 & -y_1^1 \\ x_2^2 & y_2^2 & 1 & 0 & 0 & 0 & -x_2^2 x_1^2 & -y_2^2 x_1^2 & -x_1^2 \\ 0 & 0 & 0 & x_2^2 & y_2^2 & 1 & -x_2^2 y_1^2 & -y_2^2 y_1^2 & -y_1^2 \\ x_2^3 & y_2^3 & 1 & 0 & 0 & 0 & -x_2^3 x_1^3 & -y_2^3 x_1^3 & -x_1^3 \\ 0 & 0 & 0 & x_2^3 & y_2^3 & 1 & -x_2^3 y_1^3 & -y_2^3 y_1^3 & -y_1^3 \\ x_2^4 & y_2^4 & 1 & 0 & 0 & 0 & -x_2^4 x_1^4 & -y_2^4 x_1^4 & -x_1^4 \\ 0 & 0 & 0 & x_2^4 & y_2^4 & 1 & -x_2^4 y_1^4 & -y_2^4 y_1^4 & -y_1^4 \end{bmatrix}$$

4. A trivial solution for \mathbf{h} is $\mathbf{h} = \mathbf{0}$. The matrix \mathbf{A} is not full rank. Because \mathbf{h} has 8 degrees of freedom, \mathbf{A} needs to be rank 8 to solve 8 unknowns and account for different scaling. Also, if \mathbf{A} is full rank, \mathbf{A} will have no rows of 0's and the number of equations equals the number of unknowns in \mathbf{h} , and there will only be a trivial solution $\mathbf{h} = \mathbf{0}$. Since \mathbf{A} is not full rank, some eigenvalues will be 0. Suppose \mathbf{A} has rank r , then $9 - r$ eigenvalues are 0. The eigenvectors corresponding to the zero eigenvalues are in the nullspace of \mathbf{A} , namely, the solutions for \mathbf{h} .

Q1.3 Homography under rotation

$$\begin{aligned}\mathbf{x}_1 &\equiv \mathbf{K}_1 [\mathbf{I} \quad \mathbf{0}] \mathbf{X} = \mathbf{K}_1 [\mathbf{I} \quad \mathbf{0}] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}_1 \mathbf{X} \\ \mathbf{x}_2 &\equiv \mathbf{K}_2 [\mathbf{R} \quad \mathbf{0}] \mathbf{X} = \mathbf{K}_2 [\mathbf{R} \quad \mathbf{0}] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}_2 \mathbf{R} \mathbf{X} \\ &\Rightarrow \mathbf{X} = \mathbf{R}^{-1} \mathbf{K}_2^{-1} \mathbf{x}_2 \\ &\Rightarrow \mathbf{x}_1 \equiv \mathbf{K}_1 \mathbf{R}^{-1} \mathbf{K}_2^{-1} \mathbf{x}_2 \equiv \mathbf{H} \mathbf{x}_2\end{aligned}$$

So, there exists a homography \mathbf{H} , which is $\mathbf{K}_1 \mathbf{R}^{-1} \mathbf{K}_2^{-1}$.

Q1.4 Understanding homographies under rotation

Suppose on the initial position, the projection equation is

$$\mathbf{x}_1 \equiv \mathbf{K} [\mathbf{I} \ 0] \mathbf{X} = \mathbf{K} [\mathbf{I} \ 0] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{KX}$$

Then after rotation, the projection equation is

$$\mathbf{x}_2 \equiv \mathbf{K} [\mathbf{R} \ 0] \mathbf{X} = \mathbf{K} [\mathbf{R} \ 0] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{KRX}$$

So,

$$\begin{aligned} \mathbf{X} &= \mathbf{K}^{-1} \mathbf{x}_1 \\ \Rightarrow \mathbf{x}_2 &\equiv \mathbf{KRK}^{-1} \mathbf{x}_1 \equiv \mathbf{Hx}_1 \\ \Rightarrow \mathbf{H} &\equiv \mathbf{KRK}^{-1} \\ \therefore \mathbf{H}^2 &= \mathbf{HH} = \mathbf{KRK}^{-1} \mathbf{KRK}^{-1} = \mathbf{KR}^2 \mathbf{K}^{-1} \end{aligned}$$

Since

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

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

Therefore, \mathbf{H}^2 is the homography corresponding to a rotation of 2θ .

Q1.5 Limitations of the planar homography

If the camera rotates too much based on the object, the two projected images will have little in common, and there will be no matching points between them.

Q1.6 Behavior of lines under perspective projections

Suppose that three 3D points \mathbf{X}_1 , \mathbf{X}_2 and \mathbf{X}_3 lie on a line l . Then

$$\lambda\mathbf{X}_1 + (1 - \lambda)\mathbf{X}_2 = \mathbf{X}_3$$

When applying projection \mathbf{P} on \mathbf{X}_1 , \mathbf{X}_2 and \mathbf{X}_3 ,

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{P}\mathbf{X}_1 & \mathbf{x}_2 &= \mathbf{P}\mathbf{X}_2 & \mathbf{x}_3 &= \mathbf{P}\mathbf{X}_3 \\ \lambda\mathbf{P}\mathbf{X}_1 + (1 - \lambda)\mathbf{P}\mathbf{X}_2 &= \mathbf{P}\mathbf{X}_3 \\ \lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 &= \mathbf{x}_3\end{aligned}$$

\mathbf{x}_1 , \mathbf{x}_2 and \mathbf{x}_3 lie on a same line. Therefore, projection preserves lines.

Q2.1.1 FAST Detector

FAST detector samples a circle of pixels around a pixel p , doing a binary intensity comparison, determining p to be corner if some contiguous pixels are brighter/darker than p . **Harris corner detector** sums over pixel value changings in x and y directions within a small neighborhood, classifying big changes in both directions as corners.

FAST detector recognizes corners by directly looking at changes of brightness of neighboring pixels, unlike Harris corner detector that performs derivative to see the amount of changes in directions. Harris will need to calculate the first order derivative on both x and y directions over all neighboring pixels, forming Harris matrix to find the largest and smallest eigenvalues and determining whether they are both large and approximately equal. Therefore, FAST is more computationally efficient than Harris.

Q2.1.2 BRIEF Descriptor

BRIEF descriptor randomly samples some pixels within the neighborhood of the detected point, performing binary intensity comparison between the sampled pixels and the detected point, forming a bit vector as the descriptor for the detected point. **Filterbanks**, like **Gist descriptor**, divides the whole image into several bins, applying Gabor filters of different scales to each bin, averaging the filter responses in each bin, and concatenating all filter responses as the descriptor.

Unlike filterbanks method that extract global features of the image and match scenes that have similar global patterns, BRIEF descriptor matches point that has similar relative pixel intensities around the detected point. BRIEF pays more attention to details in image.

It will not work if using only one of the filter banks as a descriptor, because one filter only extract one kind of response over the whole image, which could not describe the whole image and is not discriminative as a feature.

Q2.1.3 Matching Methods

Hamming distance: For two binary string of 1s and 0s, *Hamming distance* measures the number of bit positions where the two bits are different

Nearest Neighbor: For point represented as a string of 1s and 0s, *Nearest Neighbor* matches it with another point that has the smallest distance with it.

Hamming distance is more computationally efficient than Euclidean distance. *Hamming distance* can use the bit operation XOR on two string representations, followed by a bit count to get the distance. Bit operation is fast to compute. However, Euclidean distance has to take the square root of the sum of the squares of the difference between each bit position, which is much more expensive than calculating *Hamming distance*.

Q2.1.4 Feature Matching

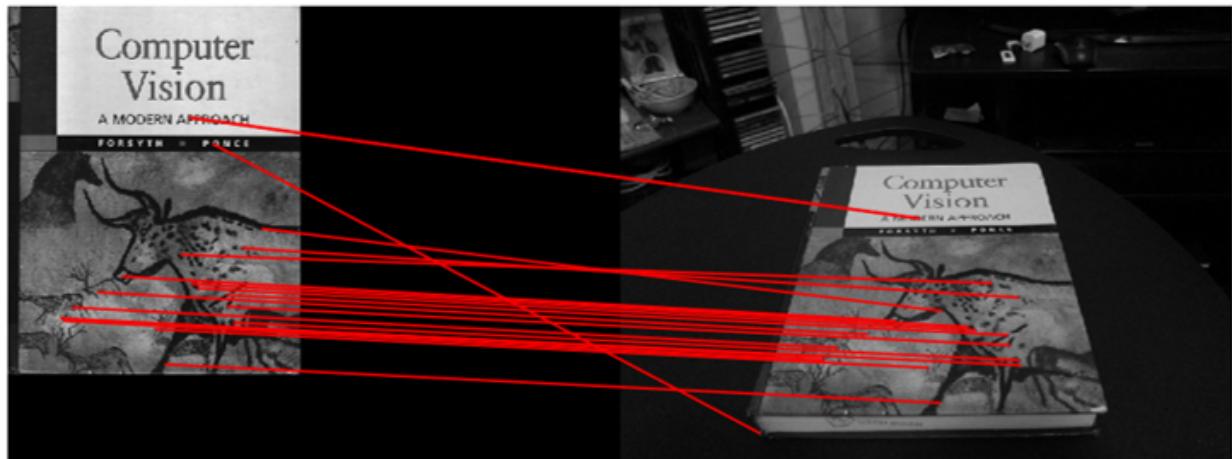


Figure 1: Visualization of matched points between cv_cover.jpg and cv_desk.jpg.

Q2.1.5 Feature Matching Parameter Tuning

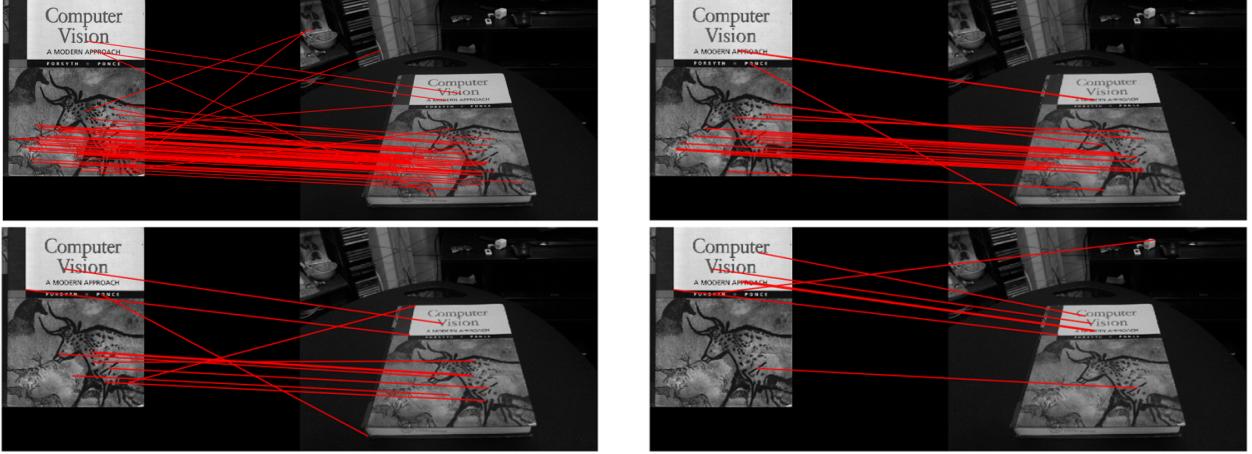


Figure 2: Matching points under `ratio=0.7` and `sigma=0.1, 0.15, 0.2, 0.25` separately.

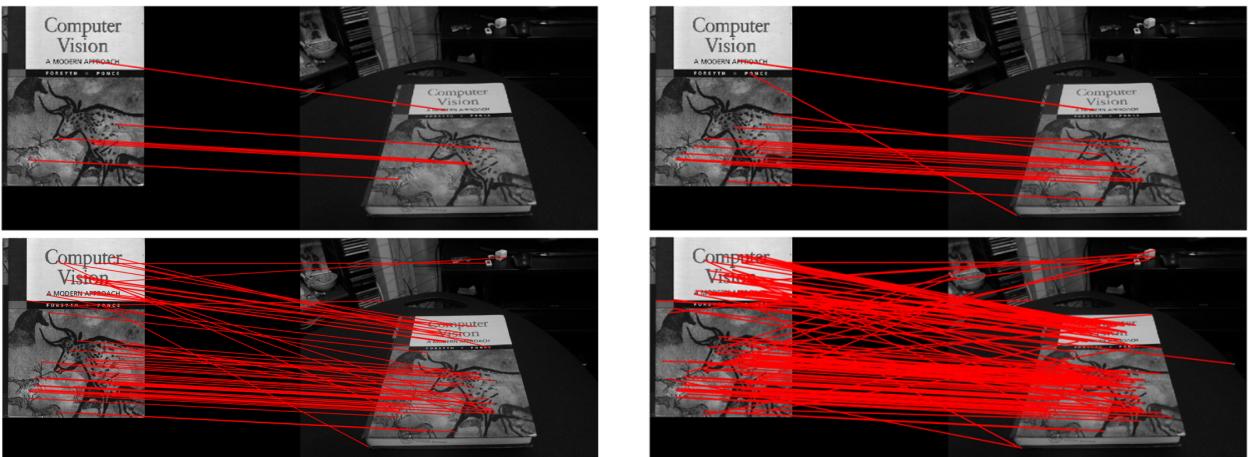


Figure 3: Matching points under `sigma=0.15` and `ratio=0.6, 0.7, 0.8, 0.9` separately.

<code>ratio=0.7</code>	<code>sigma=0.1</code>	<code>sigma=0.15</code>	<code>sigma=0.2</code>	<code>sigma=0.25</code>
# Matches	60	19	11	7

Table 1: The number of matched points under different `sigma` values, fixing `ratio=0.7`.

<code>sigma=0.15</code>	<code>ratio=0.6</code>	<code>ratio=0.7</code>	<code>ratio=0.8</code>	<code>ratio=0.9</code>
# Matches	6	19	62	130

Table 2: The number of matched points under different `sigma` values, fixing `sigma=0.15`.

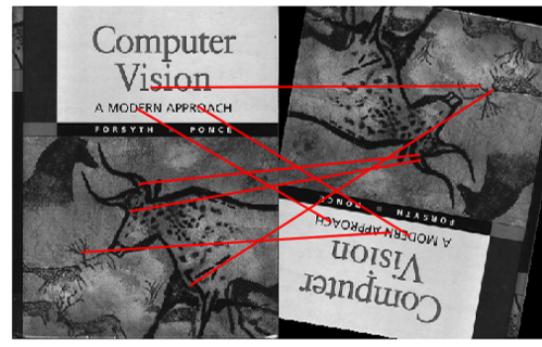
As Figure 2 and Table 1 show, fixing `ratio` and increasing `sigma` will decrease the number of matched points. It is because that `sigma` is the threshold in FAST detector, and FAST groups neighboring pixels if their intensity value $I_c > (detected\ point\ intensity\ value\ I_p + threshold)$ or $I_c < (I_p - threshold)$. If the threshold is bigger, the neighboring pixel intensity has to be more distinct than the detected point in order to be grouped together. This constraint is harder and harder to satisfy as `sigma` grows, since many corners in the image are not so distinct from their backgrounds. Therefore, less and less corners will be detected and the number of matched points decreases.

As Figure 3 and Table 2 show, fixing `sigma` and increasing `ratio` will increase the number of matched points. It is because that the `ratio` is the (distance of the closest match) / (distance of the second closest match). If `ratio` is small, it requires the second smallest matching distance to be bigger than the smallest matching distance, which means that the closest matching should be distinctively similar to the detected point. As `ratio` becomes bigger, this constraint is more relaxed. So more and more matched points will be found.

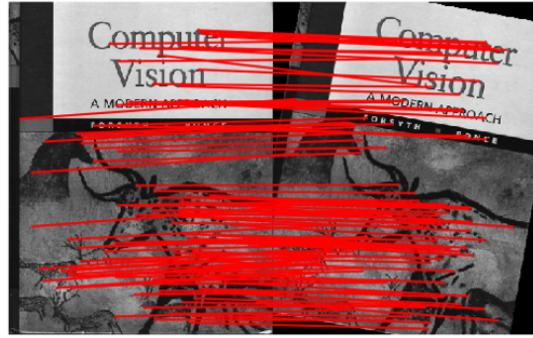
Q2.1.6 BRIEF and Rotations



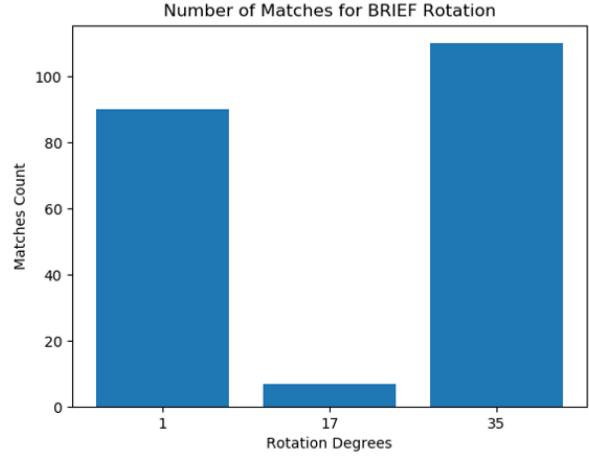
(a) Matches with itself rotated 10 degrees.



(b) Matches with itself rotated 170 degrees.



(c) Matches with itself rotated 350 degrees.



(d) The histogram of the count of matches for 10, 170 and 350 degrees of rotations.

BRIEF has lots of matching points when the image deviates little from the origin image, but has much less matching points when the image deviates a lot. This is because BRIEF samples some pixels within the neighbor of a detected point, doing a binary comparison between the sampled pixels and the detected point, using the resulting bits as a descriptor, counting the number of flipped bits between two descriptors as the distance. If the image rotates such that the pattern around the detected points is nearly up-side-down (or totally not similar to the origin pattern), the brighter-darker relationship between the sampled points and the detected points will be messed up, and there will be lots of flipped bits which makes the hamming distance big. So, many should-be-matching points are not viewed as "matching". Therefore, BRIEF has little matching points when the image deviates a lot from the origin image.

Q2.2.4 Automated Homography Estimation and Warping

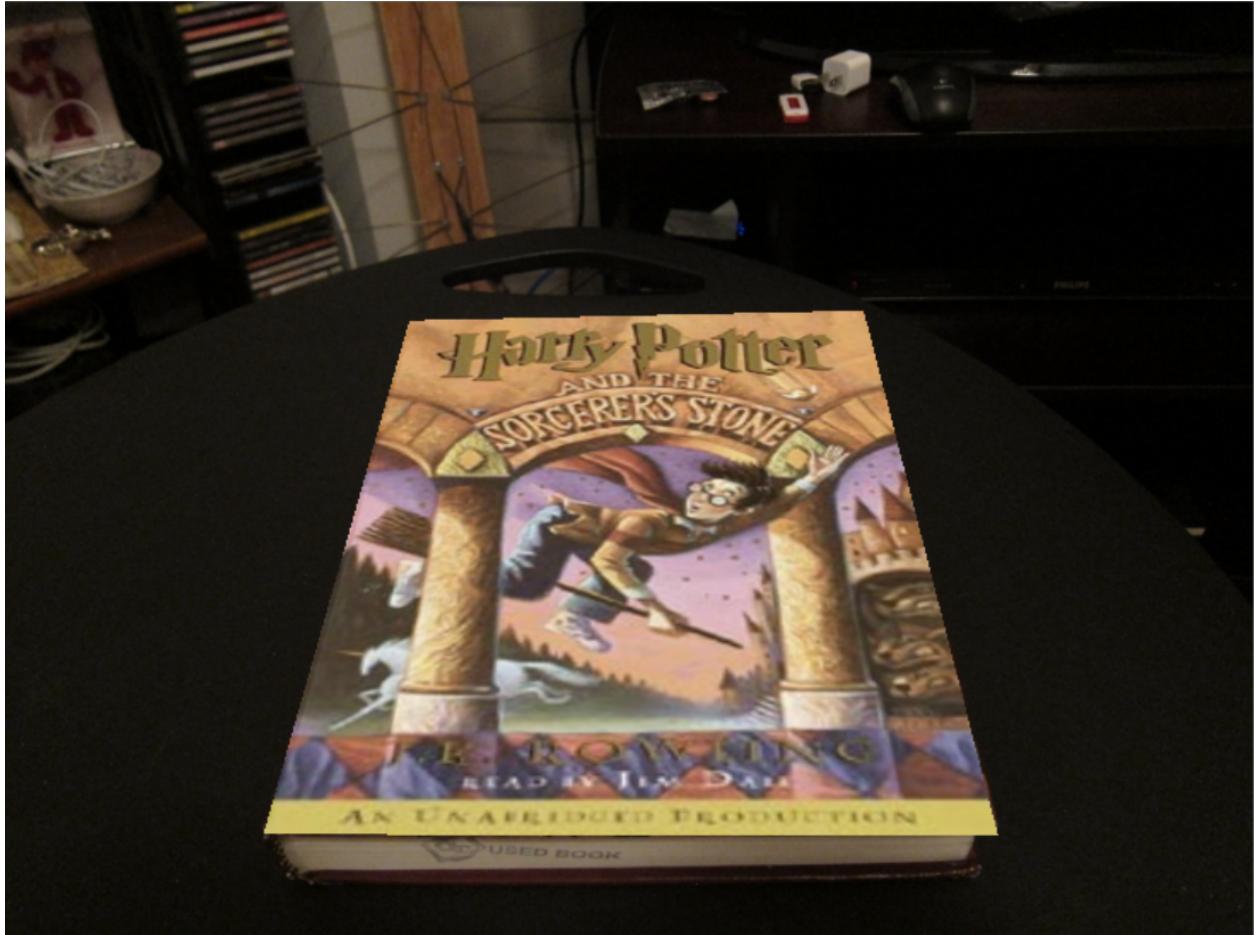


Figure 5: HarryPotterized text book.

The warpped Harry Potter cover is smaller than the desk book cover, since `hp_cover.jpg` is (295, 200, 3) and `cv_cover.jpg` is (440, 350, 3). The homography is the mapping between the cv cover and the desk book cover. Therefore, after applying homography, Harry Potter cover is smaller than the desk book cover. In order to fix this issue, we need to resize `hp_cover.jpg` to the same size as the `cv_cover.jpg`. So that after applying homography they can end at the same size.

Q2.2.5 RANSAC Parameter Tuning

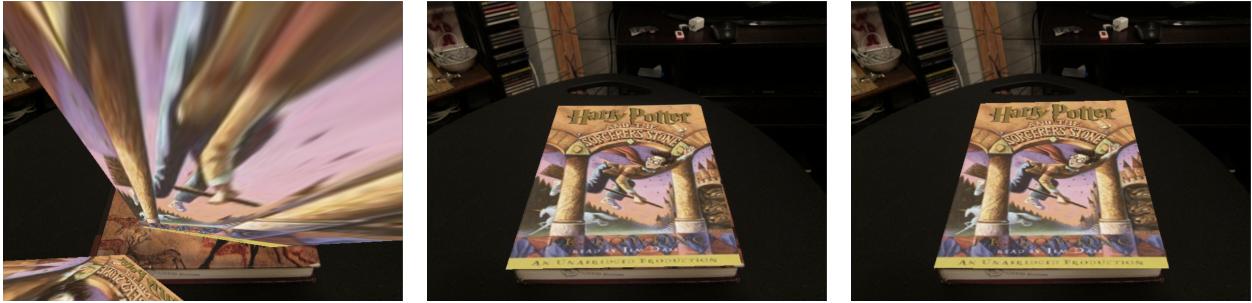


Figure 6: HarryPotterized text book under `inlier_tol=2.0` and `max_iters=6, 10, 30` separately.

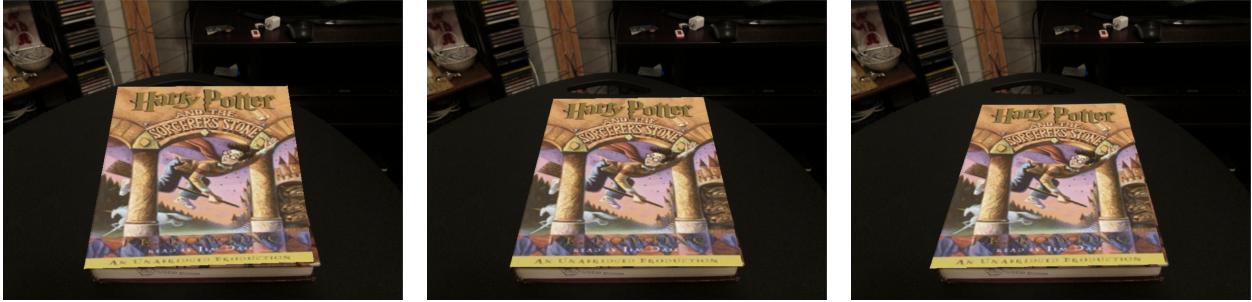


Figure 7: HarryPotterized text book under `max_iters=500` and `inlier_tol=0.1, 0.3, 5.0` separately.

<code>inlier_tol=0.2</code>	<code>max_iters=6</code>	<code>max_iters=10</code>	<code>max_iters=30</code>
# Inliers	4	12	13

Table 3: The number of inliers under different `max_iters` values, fixing `inlier_tol=0.2`.

<code>max_iters=500</code>	<code>inlier_tol=0.1</code>	<code>inlier_tol=0.3</code>	<code>inlier_tol=5.0</code>
# Inliers	5	7	13

Table 4: The number of inliers under different `inlier_tol` values, fixing `max_iters=500`.

As Figure 6 and Table 3 show, fixing `inlier_tol` and increasing `max_iters` will increase the number of inliers and the projection quality. There are 16 pairs of matched points in total. RANSAC randomly samples 4 pairs of points at each iteration, calculating the homography and measuring the number of inliers under that homography. If the number of iterations is too small, RANSAC will hardly pick all inliers at one iteration and will generate a bad homography. As the number of iteration increases, it is more possible that in some rounds RANSAC will pick all inliers and generate a good homography. For a good homography, the resulting inliers will be a lot, and the projection quality will be good. Therefore, increasing `max_iters` will increase the number of inliers and the projection quality.

As Figure 7 and Table 4 show, fixing `max_iters` and increasing `inlier_tol` will increase the number of inliers, and the projection quality will increase first and then decrease. `inlier_tol` considers the maximum distance between the points and their projected points that could be considered "inliers". When `inlier_tol` is very small, it is restrict that the homography need to project the 3D points almost exactly to the 2D points. There's no perfect homography, and the numbers of inliers will be very small which is not a good guidance for generating homography. So the projection quality is bad. As `inlier_tol` increases, the number of inliers will grow and be closer and closer to the best amount for estimation. So the projection quality will increase. When `inlier_tol` is very big, many outliers are considered inliers, which makes the consensus measurement not so accurate, and may misguide the choice of homography. So, the projection quality decreases after then.

Q4.2x Create a Simple Panorama



(a) Original image (left)



(b) Original image (right)



(c) Panorama