Computer Vision - 16720-A

Kinjal Jain

Homework 2

February 21, 2020

1 Homographies

***Q*1.1** Prove that there exists a homography H that satisfies equation 1 given two 3 $\times$ 4 camera projection matrices P1 and P2 corresponding to the two cameras and a plane . You do not need to produce an actual algebraic expression for H. All we are asking for is a proof of the existence of H.

$$x_1 \equiv Hx_2 \tag{1}$$

Consider plane Q, and let X $\in$R^3 be the focus point in Q, such that:

$$x_1 = \alpha P1X \tag{2}$$

$$x_2 = \beta P2X \tag{3}$$

where α and β are scalars in $\mathbb{R}$ Then,

$$X = \frac{1}{\beta} P2^{-1} x_2 \tag{4}$$

$$x_1 = \frac{\alpha}{\beta} P1P2^{-1} x_2 \tag{5}$$

$$x_1 \equiv P1P2^{-1} x_2 \tag{6}$$

$$x_1 \equiv Hx_2 \tag{7}$$

where,

$$H = P1P2^{-1} \tag{8}$$

and this H is 3 $\times$ 3 matrix which exists. Hence, proved.

***Q*1.2** Correspondences

1. How many degrees of freedom does **h** have?
The degrees of freedom of h is 8.

2. How many point pairs are required to solve **h**?
Since the degrees of freedom of h is 8, we require 8 equations to solve this system. So, we will need 4 pair of points to obtain 8 equations. (as 1 point gives 2 equations).

3. Derive Ai.

$$x_1^i \equiv H x_2^i \tag{9}$$

Then, multiplying by scale factor λ_i,

$$\lambda_i x_1^i = H x_2^i \tag{10}$$

So,

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

Then, solving the matrix we will get,

$$\lambda_i x_1^i = h_{11} x_2^i + h_{12} y_2^i + h_{13} \tag{11}$$

$$\lambda_i y_1^i = h_{21} x_2^i + h_{22} y_2^i + h_{23} \tag{12}$$

$$\lambda_i = h_{31} x_2^i + h_{32} y_2^i + h_{33} \tag{13}$$

Equation [11 / 13] gives,

$$\frac{x_1^i}{1} = \frac{h_{11} x_2^i + h_{12} y_2^i + h_{13}}{h_{31} x_2^i + h_{32} y_2^i + h_{33}} \tag{14}$$

[14] can be written as,

$$x_1^i (h_{31} x_2^i + h_{32} y_2^i + h_{33}) = h_{11} x_2^i + h_{12} y_2^i + h_{13} \tag{15}$$

Equation [12 / 13] gives,

$$\frac{y_1^i}{1} = \frac{h_{21} x_2^i + h_{22} y_2^i + h_{23}}{h_{31} x_2^i + h_{32} y_2^i + h_{33}} \tag{16}$$

[16] can be written as,

$$y_1^i (h_{31} x_2^i + h_{32} y_2^i + h_{33}) = h_{21} x_2^i + h_{22} y_2^i + h_{23} \tag{17}$$

Solving [15],

$$h_{31}x_2^ix_1^i + h_{32}y_2^ix_1^i + h_{33}x_1^i = h_{11}x_2^i + h_{12}y_2^i + h_{13} \tag{18}$$
$$h_{31}x_2^iy_1^i + h_{32}y_2^iy_1^i + h_{33}y_1^i = h_{21}x_2^i + h_{22}y_2^i + h_{23} \tag{19}$$
$$\tag{20}$$

Equations [18] and [19] can be re-arranged into,

$$h_{31}x_2^ix_1^i + h_{32}y_2^ix_1^i + h_{33}x_1^i - h_{11}x_2^i - h_{12}y_2^i - h_{13} = 0 \tag{21}$$
$$h_{31}x_2^iy_1^i + h_{32}y_2^iy_1^i + h_{33}y_1^i - h_{21}x_2^i - h_{22}y_2^i - h_{23} = 0 \tag{22}$$

On re-arranging further,

$$-h_{11}x_2^i - h_{12}y_2^i - h_{13} + h_{31}x_2^ix_1^i + h_{32}y_2^ix_1^i + h_{33}x_1^i = 0 \tag{23}$$
$$-h_{21}x_2^i - h_{22}y_2^i - h_{23} + h_{31}x_2^iy_1^i + h_{32}y_2^iy_1^i + h_{33}y_1^i = 0 \tag{24}$$

The equations [23] and [24] can be written as a product of 2 vectors A_i and h, such that $A_ih = 0$, where,

$$A_i = \begin{bmatrix} -x_2^i & -y_2^i & -1 & 0 & 0 & 0 & x_2^ix_1^i & y_2^ix_1^i & x_1^i \\ 0 & 0 & 0 & -x_2^i & -y_2^i & -1 & x_2^iy_1^i & y_2^iy_1^i & y_1^i \end{bmatrix}$$

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

4. When solving Ah = 0, in essence you're trying to find the h that exists in the null space of A. What that means is that there would be some non-trivial solution for h such that that product Ah turns out to be 0. What will be a trivial solution for h? Is the matrix A full rank? Why/Why not? What impact will it have on the eigen values? What impact will it have on the eigen vectors?
Trivial solution for Ah = 0 is when h = 0 i.e.

$$h = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Rank (A) = 8 which is < 9 , so it is not full rank since A is a 3 $\times$ 3 matrix So, nullity of matrix A = 9 - 8 = 1 which is equal to the number of 0 valued eigenvalues of A.

***Q*1.3** Homography under rotation

$$x_1 = K_1 \begin{bmatrix} I & 0 \end{bmatrix} X \tag{25}$$

$$x_2 = K_2 \begin{bmatrix} R & 0 \end{bmatrix} X \tag{26}$$

Then,

$$x_2 = K_2 R \begin{bmatrix} I & 0 \end{bmatrix} X \tag{27}$$

$$K_2^{-1} x_2 = R \begin{bmatrix} I & 0 \end{bmatrix} X \tag{28}$$

So,

$$R^{-1} K_2^{-1} x_2 = \begin{bmatrix} I & 0 \end{bmatrix} X \tag{29}$$

Then,

$$x_1 = K_1 R^{-1} K_2^{-1} x_2 \tag{30}$$

which can be rewritten as:

$$x_1 = H x_2 \tag{31}$$

where H = $K_1 R^{-1} K_2^{-1}$ is the homography matrix which is 3 $\times$ 3 matrix since K_1, K_2 and R are all 3 $\times$ 3 matrices

***Q*1.4** Understanding homographies under rotation
Let θ be the angle between x_1 and x_2. Similarly, let θ be the angle between x_2 and x_3. So, the angle between x_1 and x_3 will be 2 θ.

Now,

$$x_1 \equiv Hx_2 \tag{32}$$

and

$$x_2 \equiv Hx_3 \tag{33}$$

So,

$$x_1 \equiv H(Hx_3) \tag{34}$$

which is

$$x_1 \equiv H^2x_3 \tag{35}$$

Therefore, Since, we know that angle between x_1 and x_3 is 2 θ, we can say that H^2 corresponds to 2θ.

***Q*1.5** Limitations of the planar homography
It assumes that all the correspondences lie on the same plane which is not always the case. It also fails when corresponding points are obfuscated in an image and visible in other.

***Q*1.6** Behavior of lines under perspective projections

2 Computing Planar Homographies

***Q*2.1.1** FAST Detector

Harris Corner Detector algorithm recognizes corners by calculating gradients of each pixel in the image. It marks the pixel as a corner if the absolute gradients in two directions are both mighty. However,FAST (Feature from an accelerated segment test) detector uses a Bresenhams algorithm for circle drawing with diameter of 3.4 pixels for trial pixel. The 16 pixels that lie on circumference of this circle are compared to the main pixel's value for a complete accelerated segment using a threshold. This pixel is marked a corner based on an accelerated segment test (AST) which there must exist at least a threshold number of pixels that have more brilliant or darker circle connection. FAST detector is computationally faster compared to the Harris corner detector.

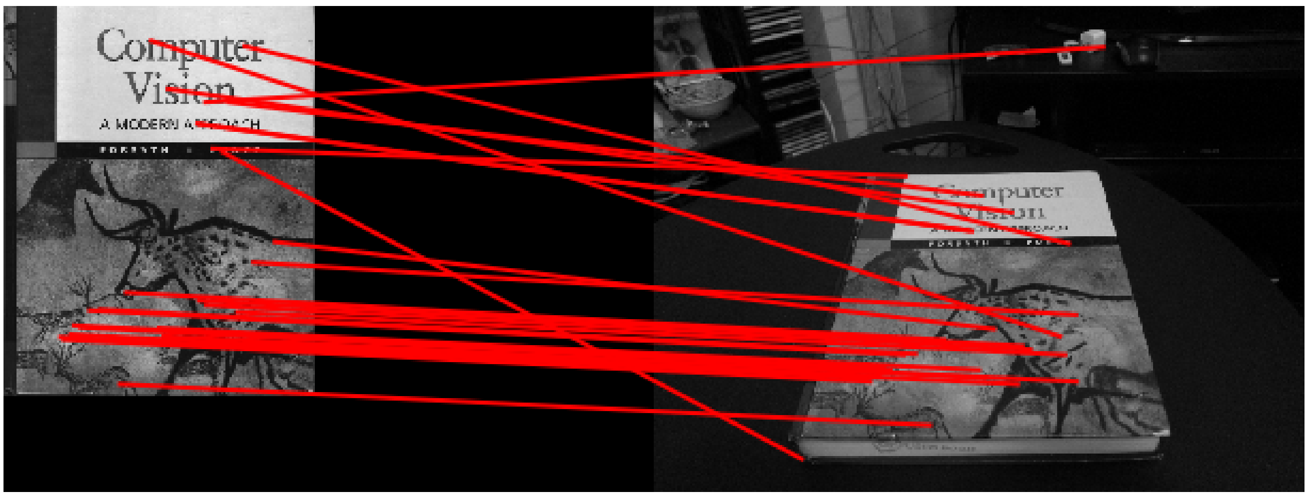
***Q*2.1.2** BRIEF Descriptor

BRIEF Descriptors are different from other filter banks as they don't really provide any method to find the actual features. which we can get using other filter banks like SIFT, Harris, SURF etc. BRIEF essentially provides a shortcut to find the binary strings directly without finding descriptors. It takes a smoothened image patch and then selects a set of n dimensional (x,y) location pairs and perform pixel intensity comparisons on these pairs, and then Hamming Distance can be used to match these descriptors. We can use other filterbanks as feature descriptors as well like Harris can detect corners and use them to get a map.

***Q*2.1.3** Matching Methods

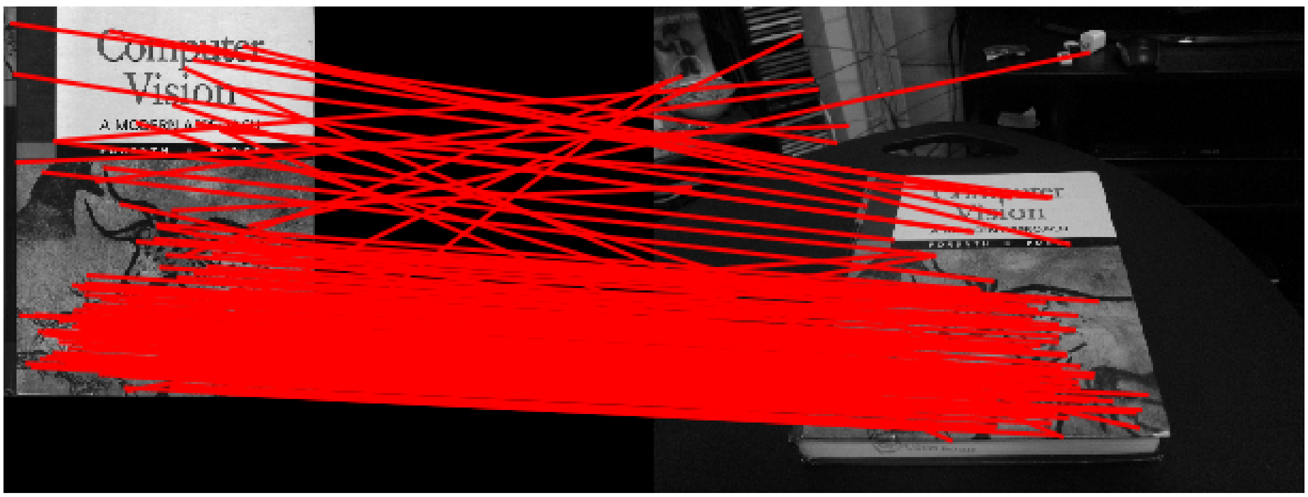
Hamming distance can be used to compare the binary vectors for matching points in the two images and this will basically count the number of points that differ in both. Similarly, Nearest Neighbor takes the two vectors of matching points from two images, and use the similarity measure to calculate distance to be the total closeness between them. The Hamming distance between two vectors whose elements are (0,1) is essentially same as the square of the Euclidean distance between them. It is advantageous to use Hamming distance in our setting because it is computationally much faster than Euclidean Distance in which extra time will be spend on taking difference and squaring it.

***Q*2.1.4** Feature Matching
Following is the matches outcome for default values of sigma = 0.15 and ratio = 0.7

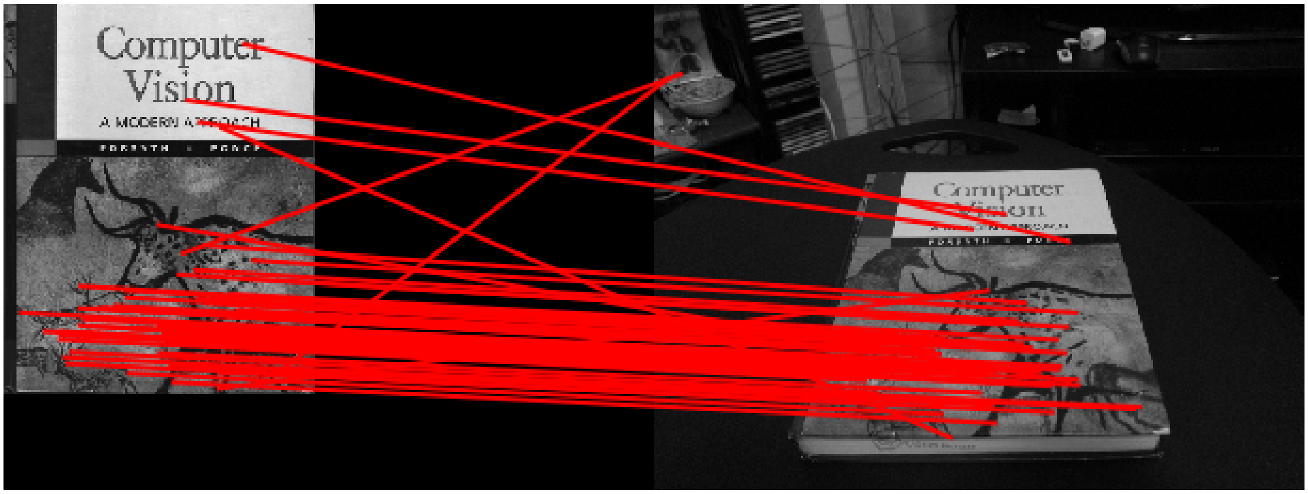


***Q*2.1.5** Feature Matching Parameter Tuning
Various sigma and ratio values were tried as follows:

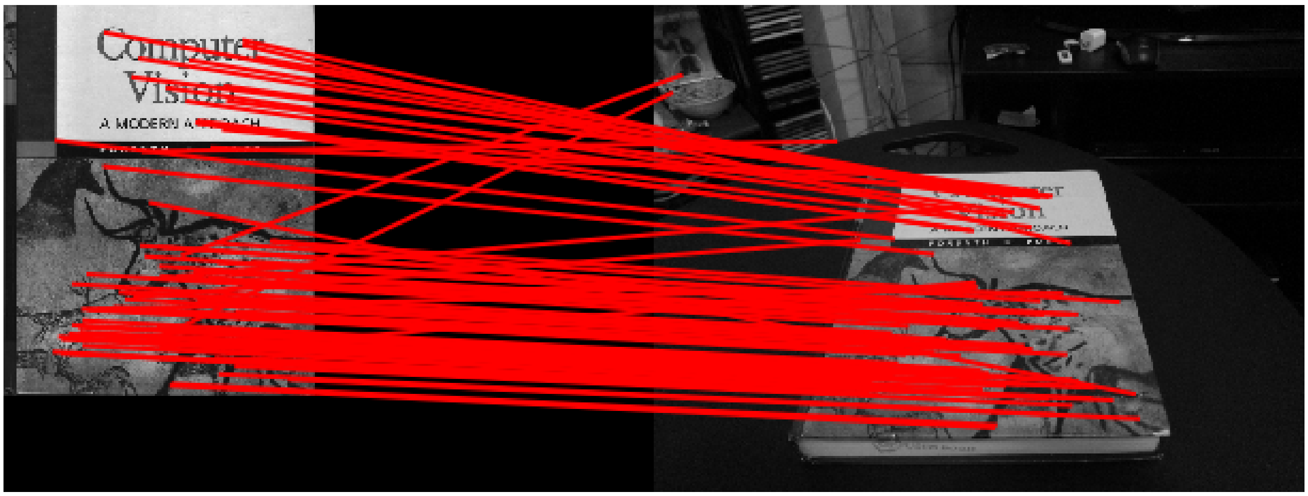
1. Descriptor matching: sigma=0.08, ratio=0.8



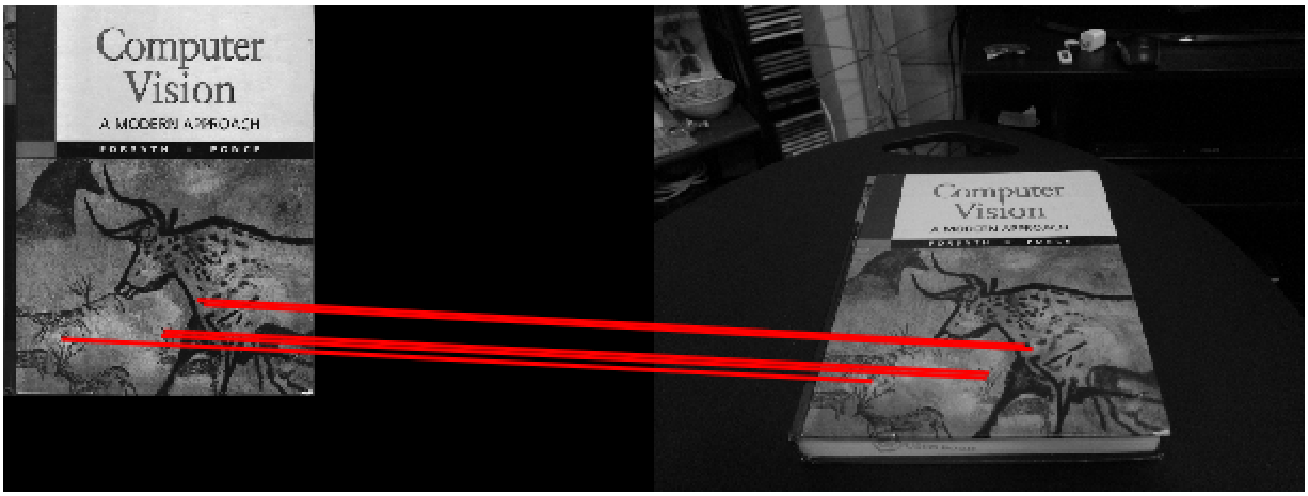
2. Descriptor matching: sigma=0.1, ratio=0.7



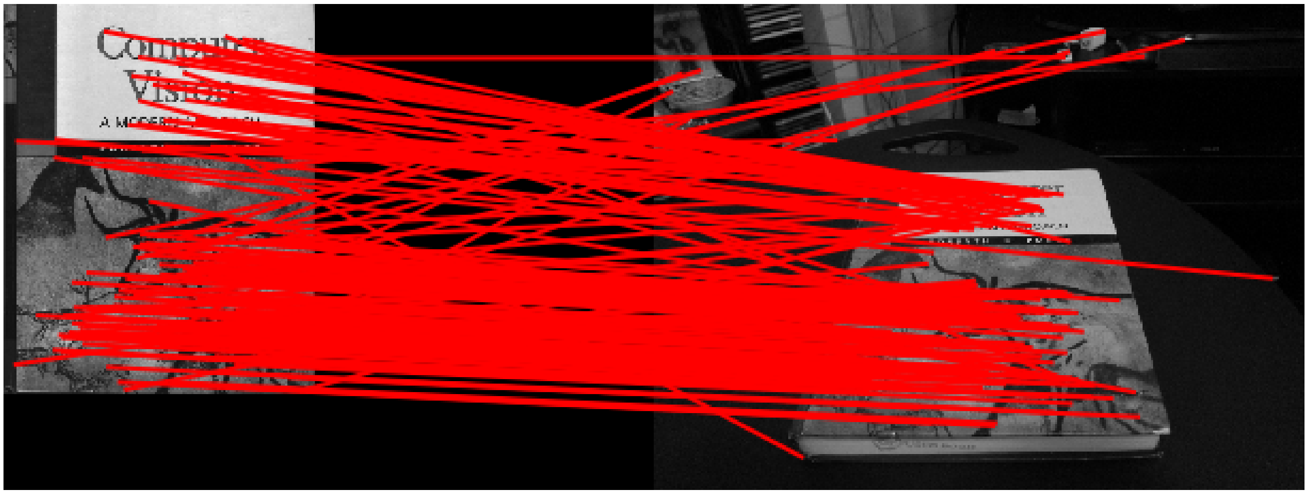
3. Descriptor matching: sigma=0.12, ratio=0.8



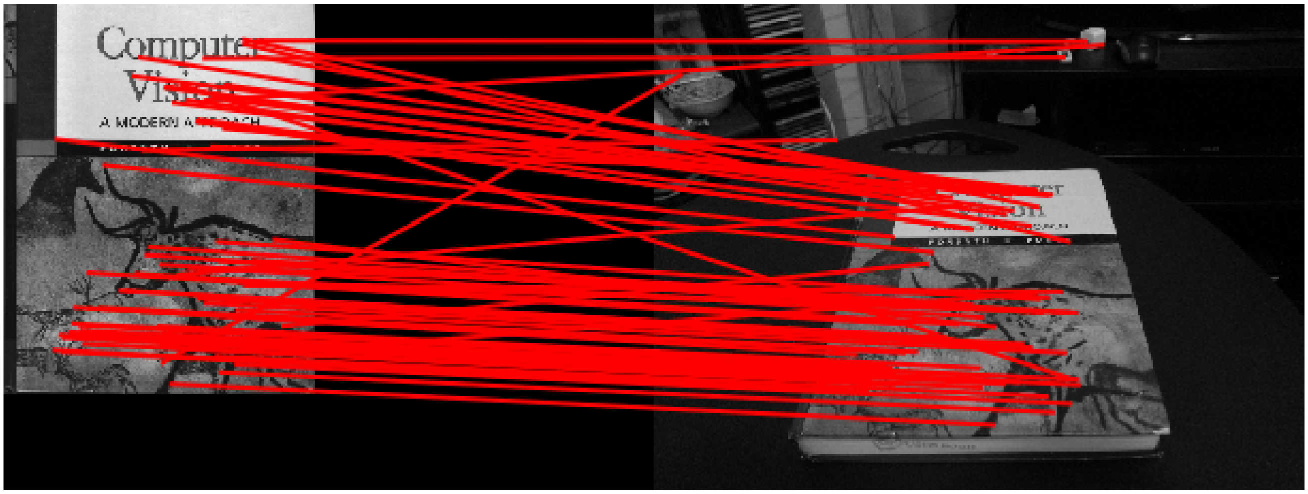
4. Descriptor matching: sigma=0.12, ratio=0.5



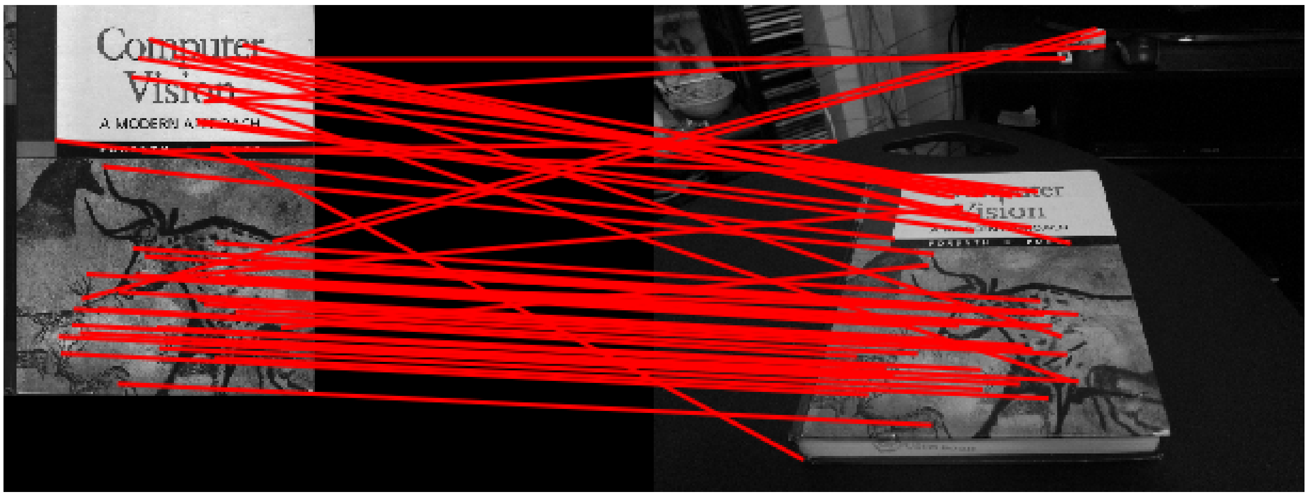
5. Descriptor matching: sigma=0.12, ratio=0.9



6. Descriptor matching: sigma=0.14, ratio=0.8



7. Descriptor matching: sigma=0.16, ratio=0.8

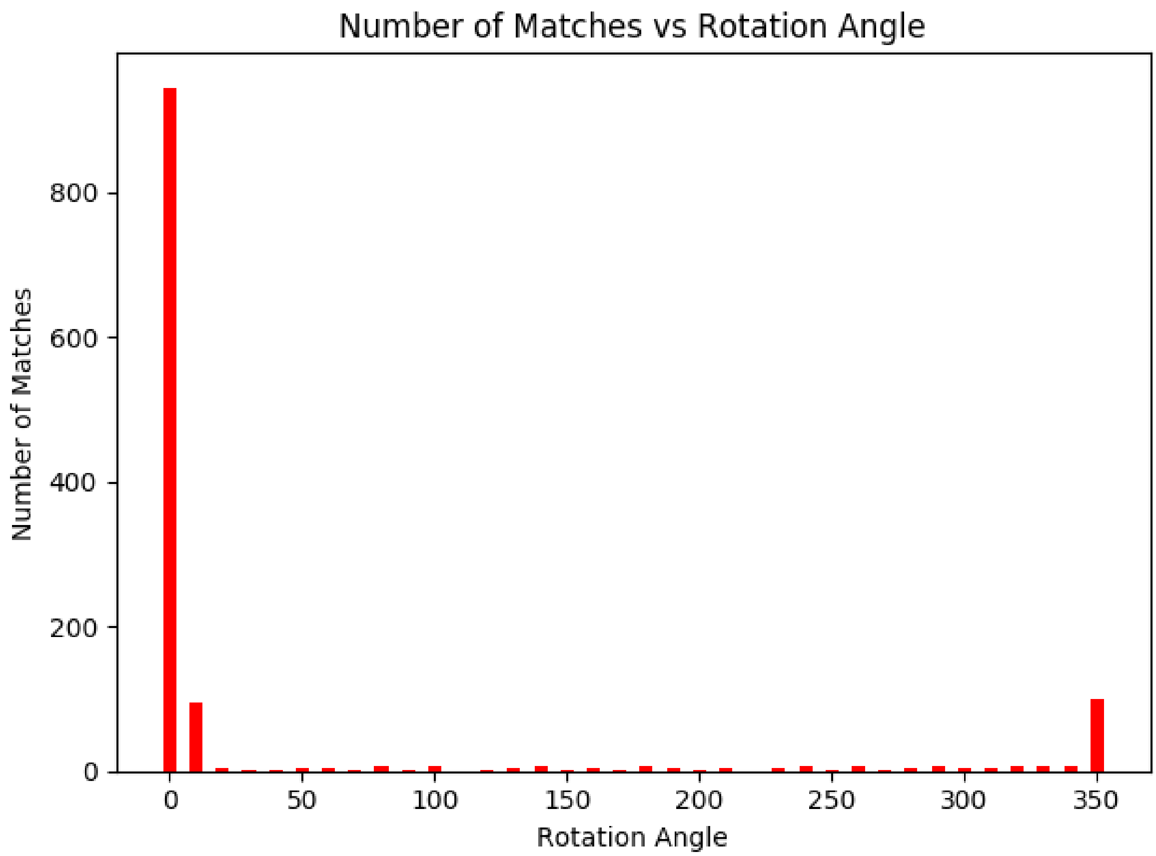


Based on the above ablation study it can be observed that higher sigma lower matches, and lower the ratio lower the matches. However, we shouldn't lower the value sigma and increase the value of ratio beyond a point after which it captures matches outside the object of interest.

***Q*2.1.6** BRIEF and Rotations

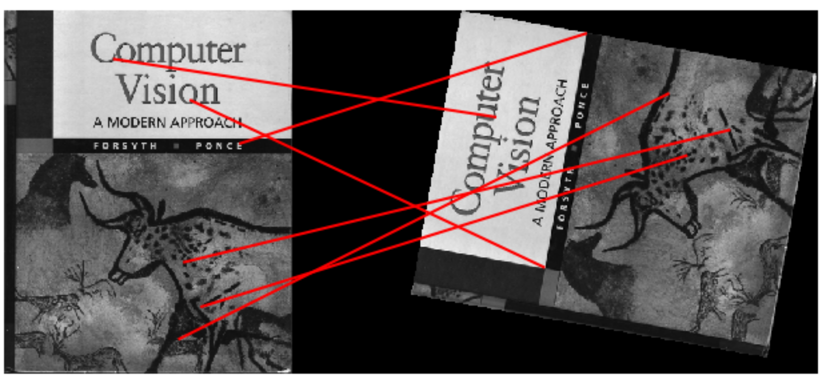
The BRIEF descriptor behaves this way because with rotations it is not able to get matching points as some points are lost when rotations are increased from an angle of 0 to 180. But after 180 degrees of rotation, the number of matches starts to increase again as the angle of difference starts decreasing the other way round. The most matches are observed at 0 degrees of rotation because the BRIEF algorithm can detect exact matches easily without any rotation (images are same).

Following is the histogram for the same:

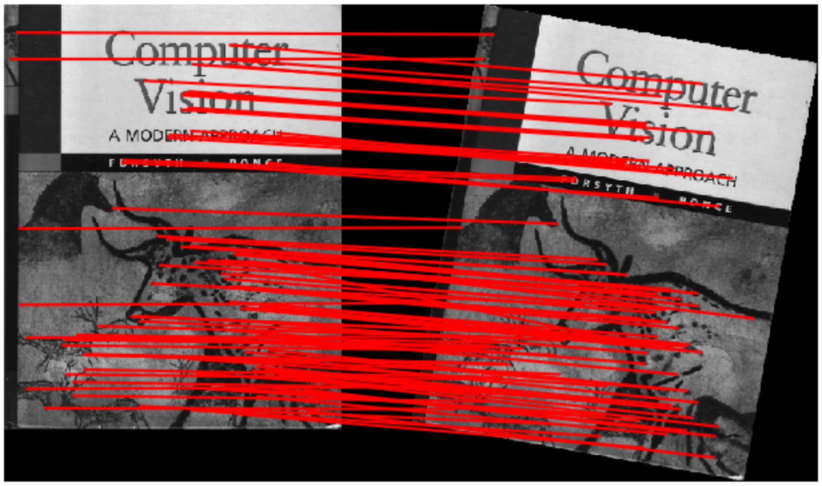


And 3 visualisations of matches at 0, 60, 350 degrees of rotation: For 0 degrees, the BRIEF descriptor gets about 900+ matches since the orientation is the same.

For 60 degrees, the BRIEF descriptor gets very few matches since the orientation is very different and number of points in rotated image are very less.

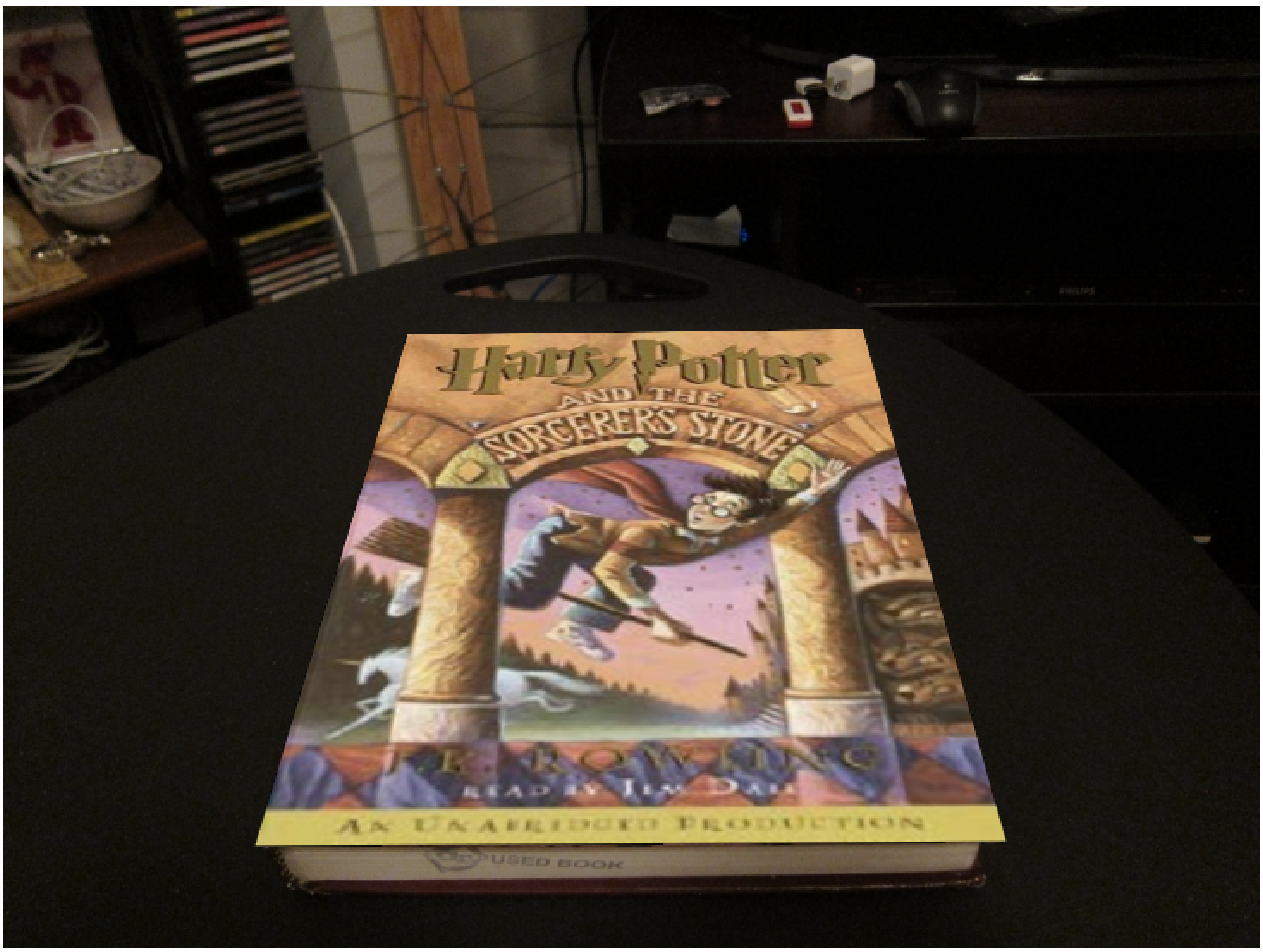


For 350 degrees, the BRIEF descriptor gets only about 100+ matches since the orientation is almost same as the one with 0 degrees rotation and it is able to get corresponding points easily.


Computer
Vision
A MODERN APPROACH
Computer
Vision
A MODERN APPROACH
FORSYTH
PONCE

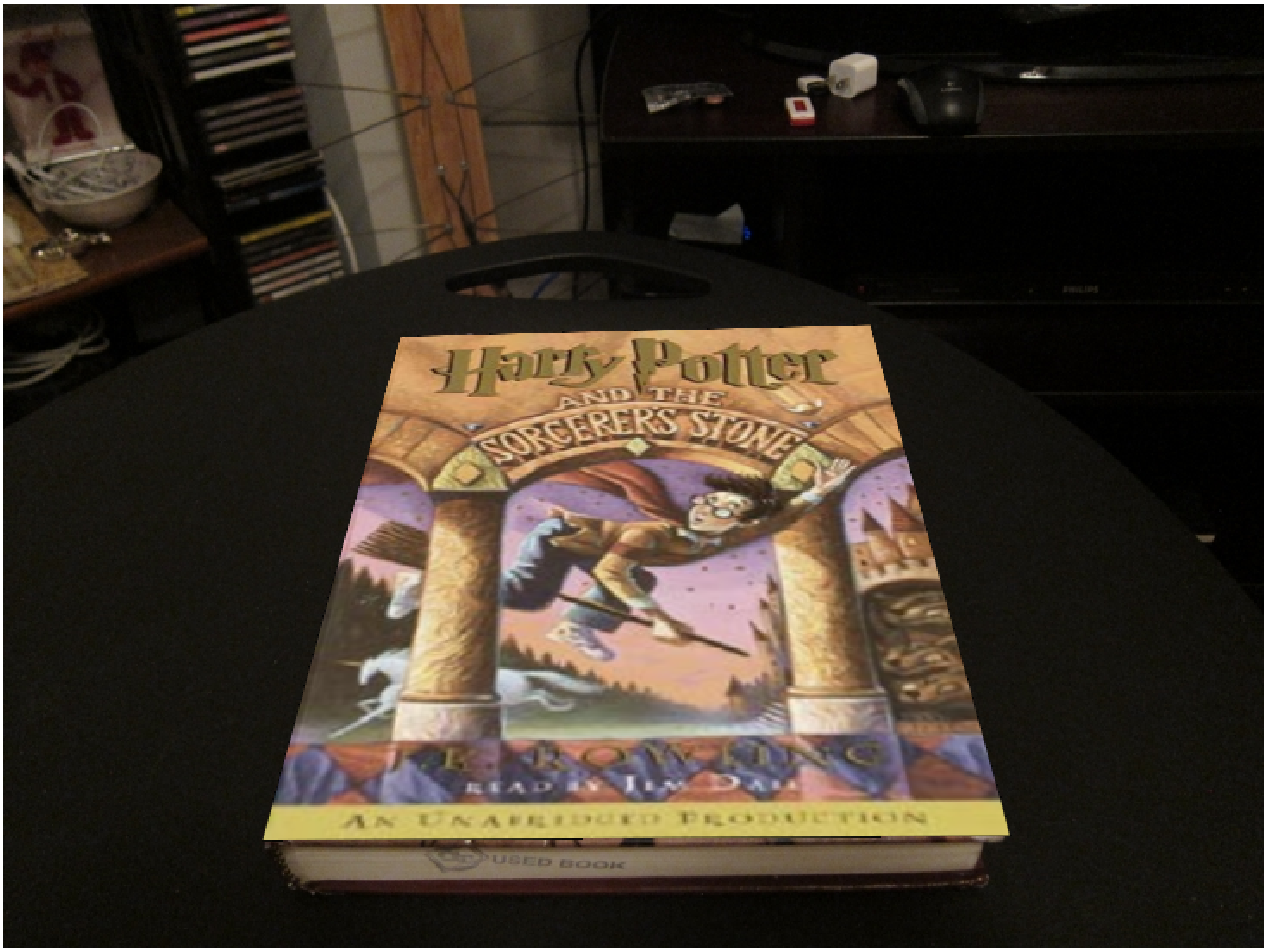
***Q*2.2.4** RANSAC - Putting it together
By default after performing RANSAC, if warp is done without resizing the $hp_{c}over.jpg$ image, it occupies smaller space in the output. This is because the homography was computer with larger imageSo, to overcome that, I resized the $hp_{c}over.jpg$ image to the size of $cv_{c}over.jpg$ image before performing the warp using $cv2.warp_{p}erspective$ function. The command used for resizing is:

$$hp\_cover = cv2.resize(hp\_cover, (cv\_cover.shape[1], cv\_cover.shape[0])) \tag{36}$$

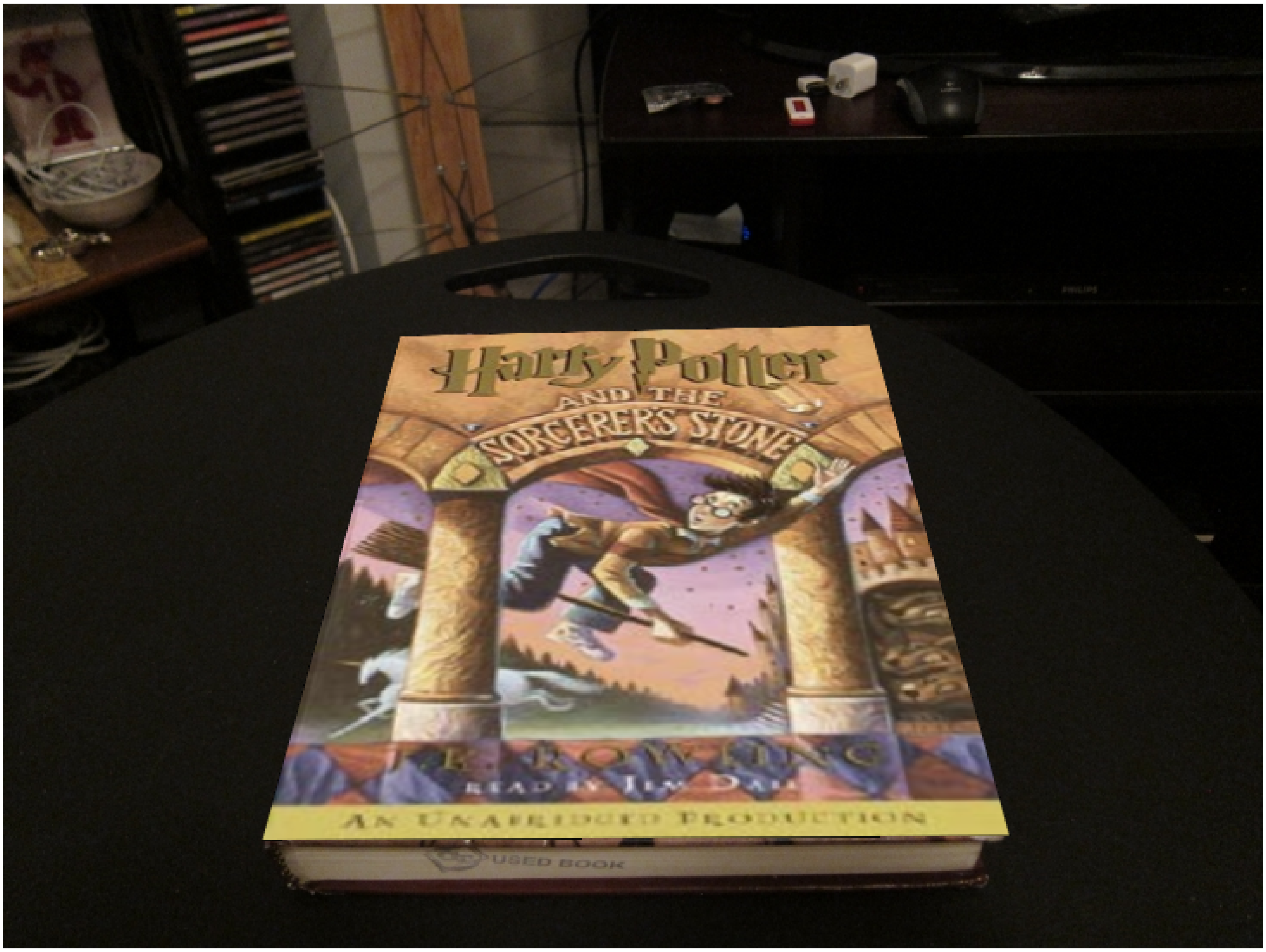


***Q*2.2.5** RANSAC Parameter Tuning
Various values for error_tolerance and max_iters are experimented as follows:

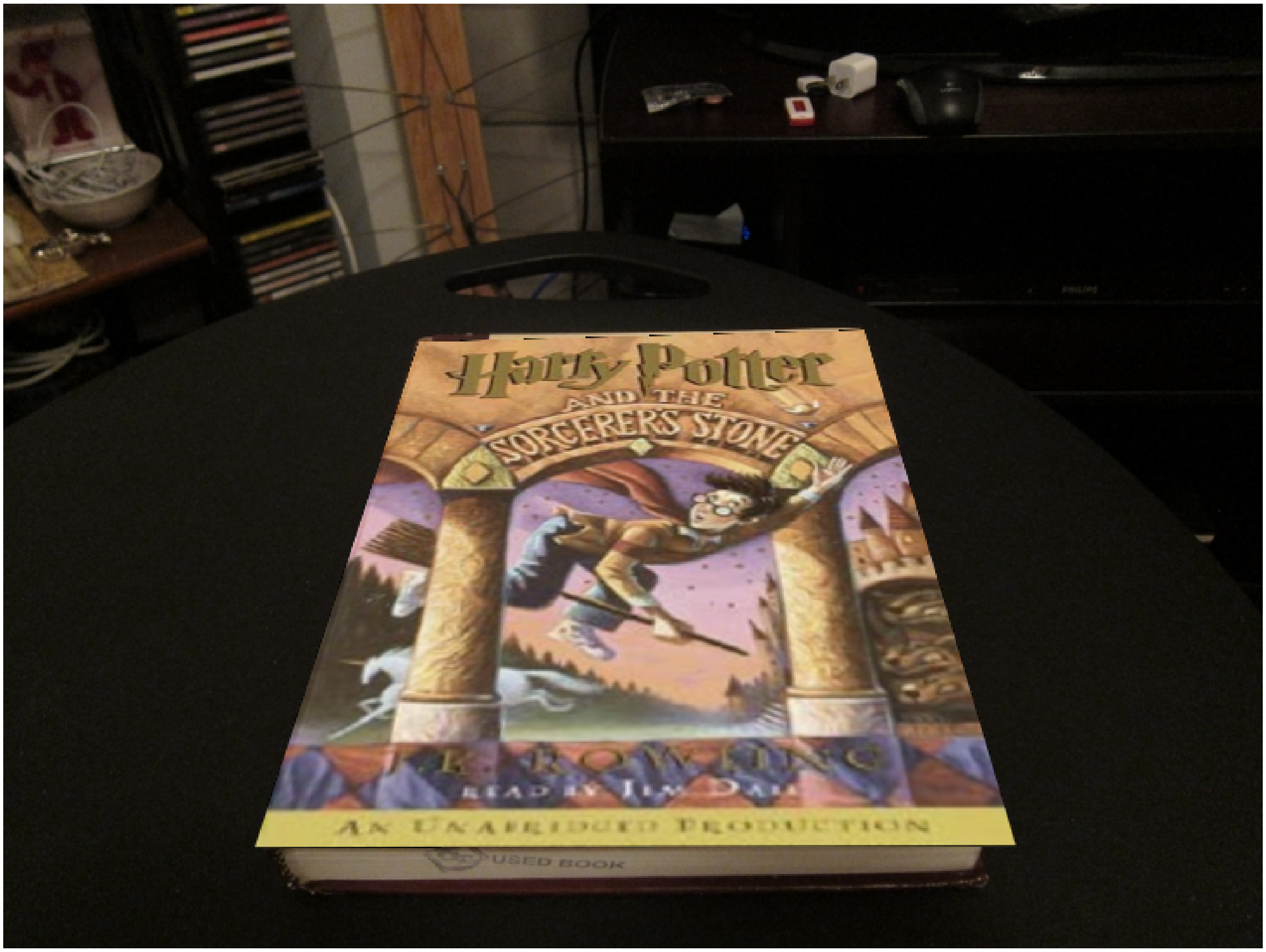
1. max_iters=500, error_tolerance=2



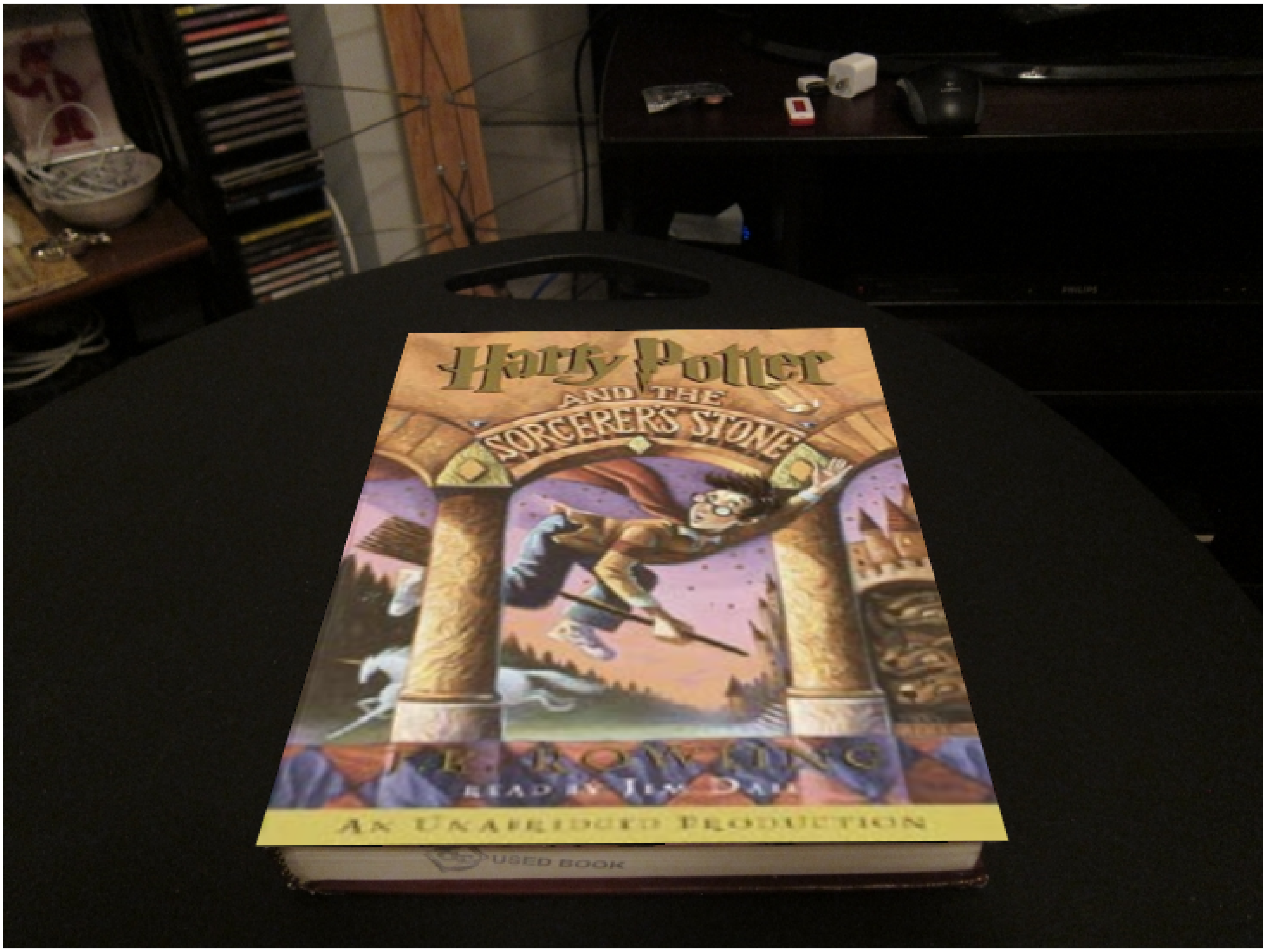
2. max_iters=1000, error_tolerance=2



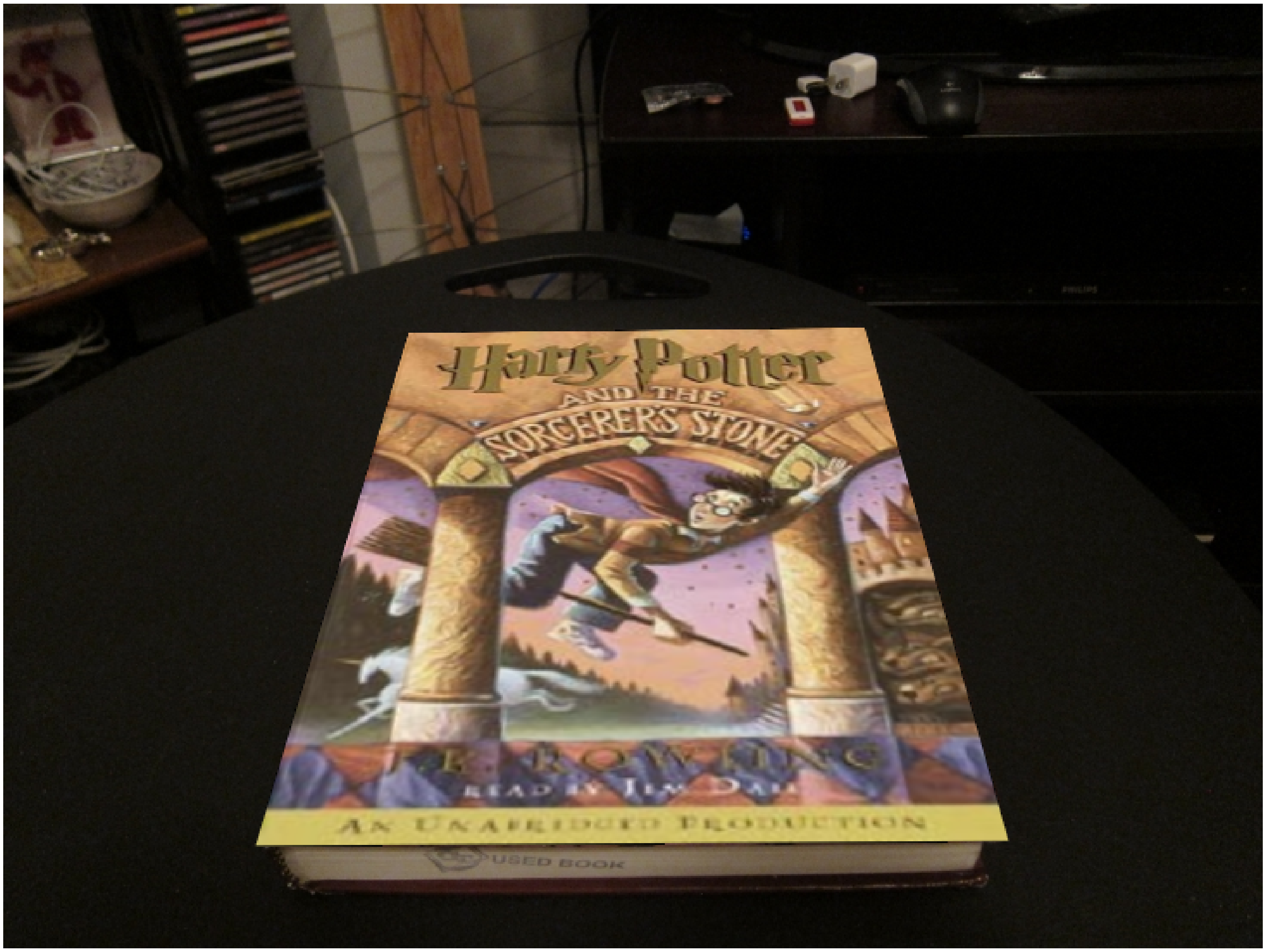
3. max_iters=1000, error_tolerance=1



4. max_iters=2500, error_tolerance=1



5. max_iters=5000, error_tolerance=1

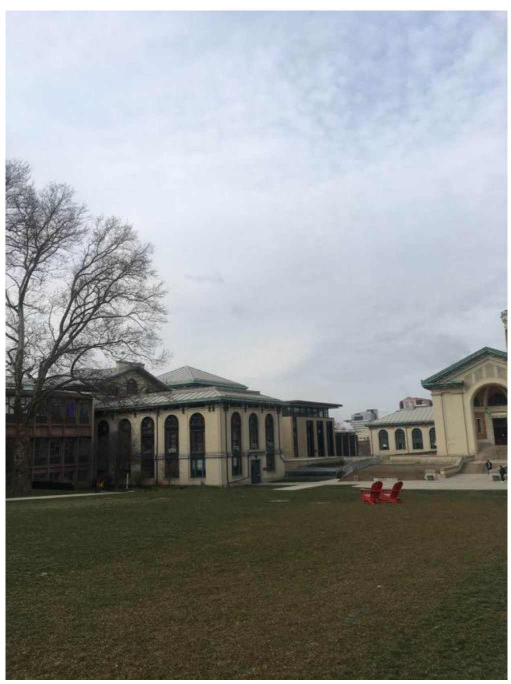


From the observations through the above ablation study, as you decrease the error tolerance, you get more accuracy, and as you increase number of iterations of RANSAC, it increases the accuracy with which the warped harry potter cover image is composed with the desk.

3 Extra Credit

Q4.2x Create a Simple Panorama Following is the panorama obtained for the two images taken in front of Wean Hall:
Left image-

Right image-



Panorama-

Following is the panorama obtained for given default images.
Left image-

Right image-

Panorama-