Collaborators

Siddarth Ghodasara, Madhusha Goonesekera, Naman Menezes, Gaurav Sethia

Q1.1

We are given that P_1 and P_2 are 3x4 matrices, we can generalize this as:

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix}$$

And,

$$X = [X, Y, Z, 1]'$$

But since our points are planar, Z = 0.

$$X = [X, Y, 0, 1]'$$

We can hence ignore the 3rd row in the P matrix since it will always be multiplied by zero.

We can hence write the P as:

$$P = \begin{bmatrix} P_{11} & P_{12} & 0 & P_{14} \\ P_{21} & P_{22} & 0 & P_{24} \\ P_{31} & P_{32} & 0 & P_{34} \end{bmatrix}$$

And hence we can write P and X as:

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{14} \\ P_{21} & P_{22} & P_{24} \\ P_{31} & P_{32} & P_{34} \end{bmatrix} \quad X = [X, Y, 1]'$$

We can hence use this projection matrix as:

$$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{14} \\ P_{21} & P_{22} & P_{24} \\ P_{31} & P_{32} & P_{34} \end{bmatrix} * \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

From this, and the information in the question, we can conclude that:

$X_{\pi} = P_1 X$

And

$X_{\pi} = P_2 X^1$

We can now say that

$P_1 X = P_2 X^1$

$P_1^{-1} P_1 X = P_1^{-1} P_2 X^1$

$X = P_1^{-1} P_2 X^1$

Hence, we can say that

$H = P_1^{-1} P_2$

Q1.2

1. h has 8 degrees of freedom.
2. We need 4 point pairs to solve for h, since each point pair gives us 2 equations, and we need 8 equations to solve for the 8 degrees of freedom.
3. From the general form of a homography, we know that

$$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{14} \\ H_{21} & H_{22} & H_{24} \\ H_{31} & H_{32} & H_{34} \end{bmatrix} * \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

From this, we can say that:

$$U = \frac{X * H_{11} + Y * H_{12} + H_{14}}{X * H_{31} + Y * H_{32} + H_{34}}$$

Simplifying this, we can say that:

$$X * H_{11} + Y * H_{12} + H_{14} - U * X * H_{31} - U * Y * H_{32} - U * H_{34} = 0$$

Similarly for V:

$$V = \frac{X * H_{21} + Y * H_{22} + H_{24}}{X * H_{31} + Y * H_{32} + H_{34}}$$

$$X * H_{21} + Y * H_{22} + H_{24} - V * X * H_{31} - V * Y * H_{32} - V * H_{34} = 0$$

Generalizing these, we can say that:

$$A_i = \begin{bmatrix} X_i & Y_i & 1 & 0 & 0 & 0 & -U_i * X_i & -U_i * Y_i & -U_i \end{bmatrix}$$

$$A_{i+1} = \begin{bmatrix} 0 & 0 & 0 & X_i & Y_i & 1 & -V_i * X_i & -V_i * Y_i & -V_i \end{bmatrix}$$

4. The trivial solution for Ah = 0, is 0, but we can't use this solution as a homography matrix with an image. Hence, we choose to derive the non-trivial solutions using eigenvalue Decomposition (EVD) or Singular Value Decomposition (SVD).

While using EVD, our solution is the column of the eigen vector corresponding to the eigen value with the value 0. This means that one of the eigen values need to be zero, which is not possible if the matrix A is full rank. Hence, we can say that A is not full rank.

Q1.3

We are given the following equations

$$X_1 = K_1\,[I \quad 0]\,X$$

$$X_2 = K_2\;[R \quad 0]\,X$$

Where I is an identity matrix, R is the rotation matrix of the camera, K is the intrinsic matrix of the cameras, and X is a point in 3D space.

Simplifying the first equation, we get:

$$K_1^{-1}X_1 = \; K_1^{-1}K_1\,[I \quad 0]\,X$$

$$K_1^{-1}X_1 = \; K_1^{-1}K_1\;X$$

$$X = K_1^{-1}X_1$$

Simplifying the second equation, we get:

$$X_2 = K_2\;[R \quad 0]\,X$$

Combining these two simplifications, we get:

$$X_2 = K_2\mathrm{RX}$$

$$X_2 = K_2\,R\,K_1^{-1}X_1$$

Assuming K_2RK^{-1} as the homography matrix, we can say that X_2 = H X_1

Q1.4

For a rotation θ, we are given that

$$X_1 = H\,X$$

Similarly, rotating X_1 further by another θ, we can say that

$$X_2 = H\,X_1$$

Combining these two equations, we can say that:

$$X_2 = H\,(H\,X)$$

$$X_2 = H^2 X$$

Hence proving that H^2 is the homography corresponding to a rotation of 2θ.

Q1.5

Planar homography is not completely sufficient to map any arbitrary scene image to another viewpoint, because we make an assumption that the 3^{rd} coordinate of the point is 0 since it lies on a flat plane, which wont hold for most real-world scenes that have depth variations and non-planar surfaces.

Q1.6

We can write a general point in 3D space as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \lambda B$$

If we scale this line with a projection factor P, we can write it as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = P\left(\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \lambda B\right)$$

Simplifying this, we get:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = P\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + P\lambda B$$

Since this equation is still of the form Y = A + λB, we can conclude that the points are still on a line

Q2.1.1

The FAST detector operates by comparing pixel intensities around a pixel. If it finds a set of at least 8 neighboring pixels continuously above or below a certain threshold, it will be marked as a corner.

Harris corner on the other hand, uses the change in intensities (gradients) in multiple directions to comput a corner response function. A corner in the image will have a high response (energy), and is marked as one thereof.

The FAST detector is more computationally efficient than the harris corner detector since it involves simple intensity comparisons. At the same time, FAST is less robust in the presence of noise and more sensitive to lighting changes, than the Harris detector.

Q2.1.2

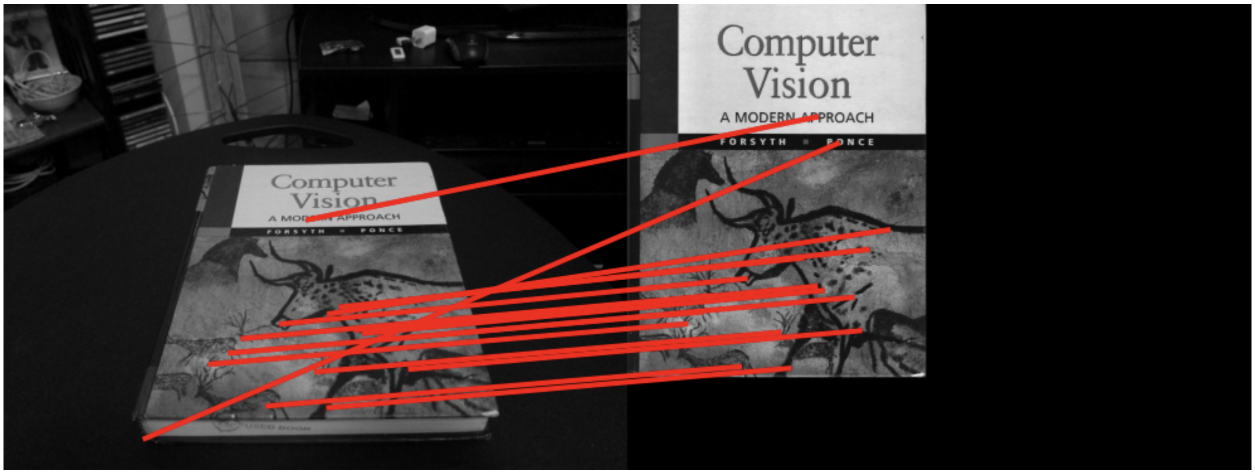
BRIEF is a binary descriptor that represents a feature point in an image as a binary string, representing a comparison of pixel intensities around the feature point. Filterbanks on the other hand are a collection of kernels which when convoluted with an image can help extract various features. Its a more general tool for filtering and extracting information from an image and their outputs are real-valued and not binary like BRIEF.

Filterbanks can be used instead of BRIEF, with the right modifications as descriptors. Since the output of each filter represents a different feature, it could be converted to a binary descriptor to be used instead of BRIEF

Q2.1.3

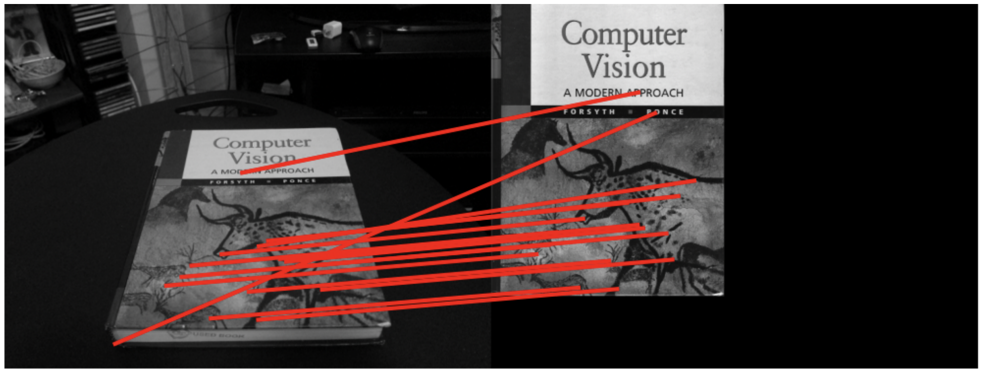
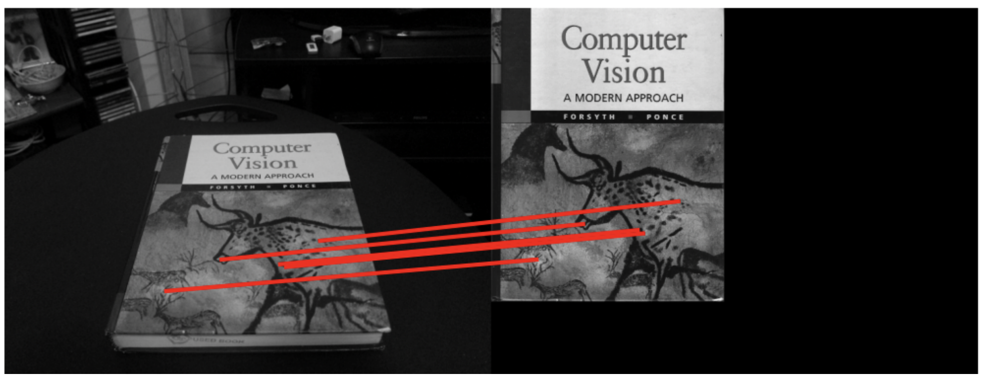
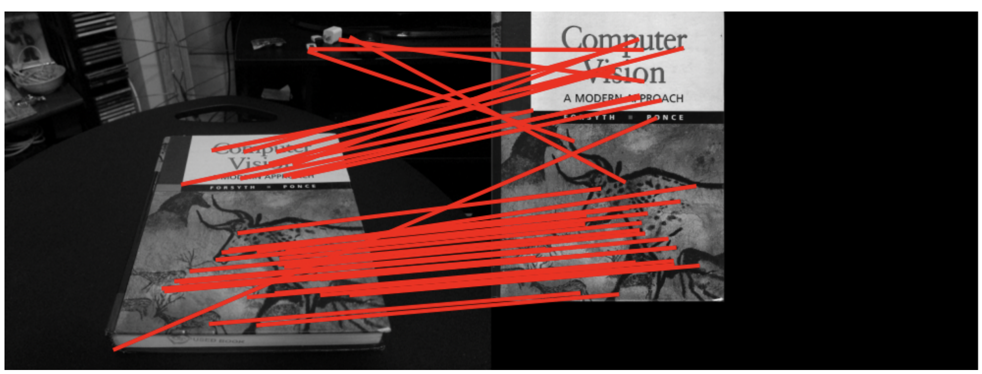
In the context of binary descriptors, the Hamming distance measures the distance by determining the disslimilarity between the descriptors by counting the number of differing bits. This makes it computationally lighter for finding descriptors for the same feature than other alternatives like euclidean distance which works with floating point math.

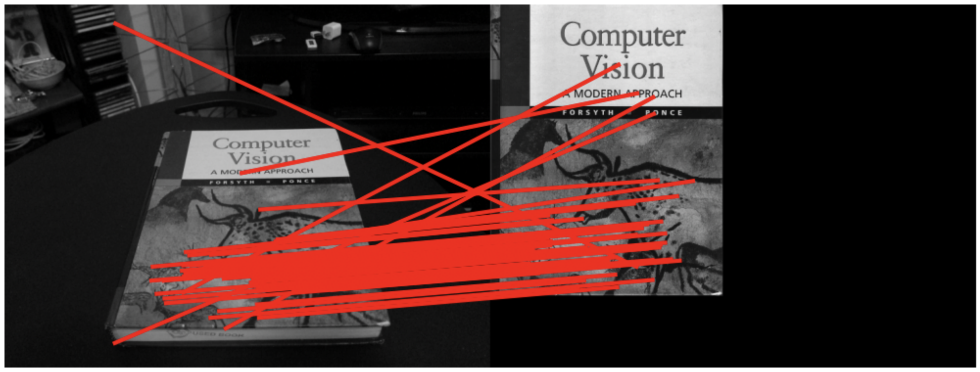
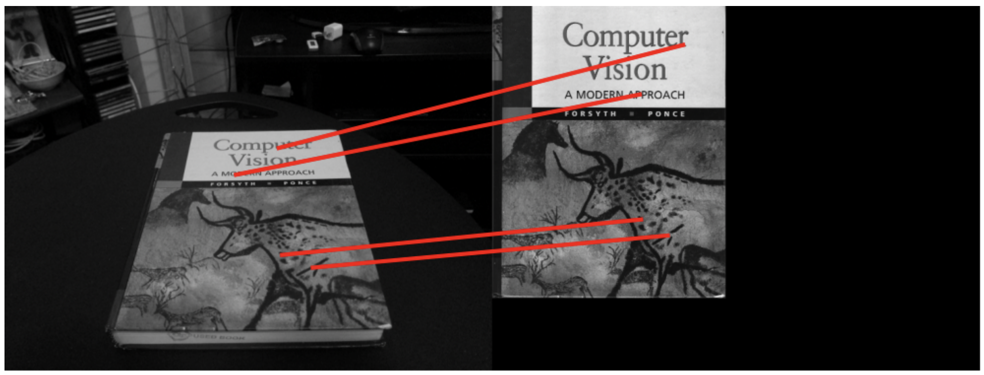
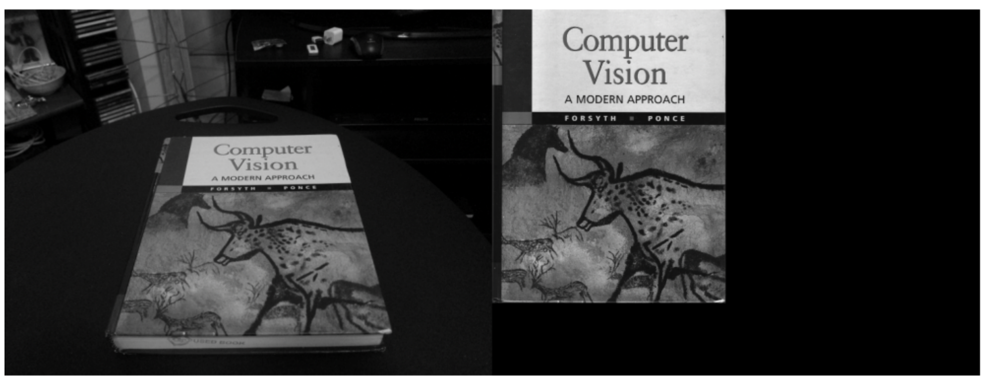
Q2.1.4



Screenshot of visualization of the matched points

Q2.1.5

Sigma	Ratio	Image
0.15	0.7	
0.15	0.6	
0.15	0.8	

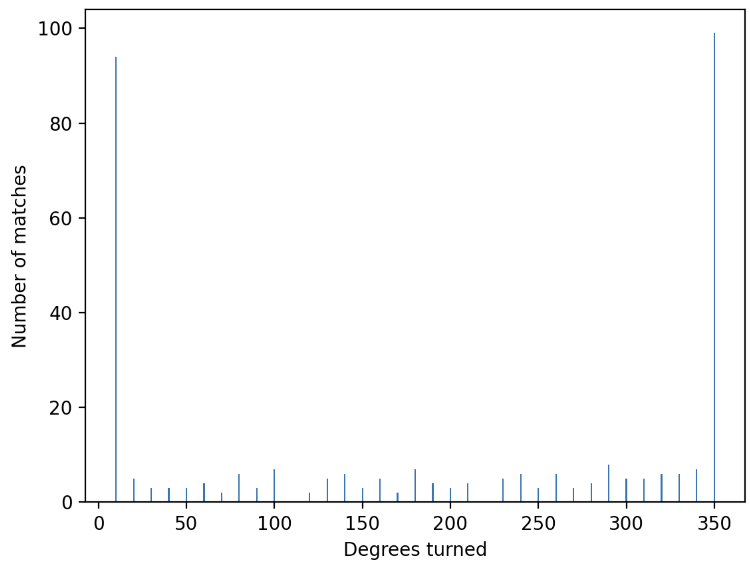
0.1	0.7	
0.2	0.7	
0.5	0.7	

Ablation study with different alpha and sigma values

The number of matches increases as the sigma is reduced or the ratio is increased. As the number of matches increases, the number of bad matches increases as well.

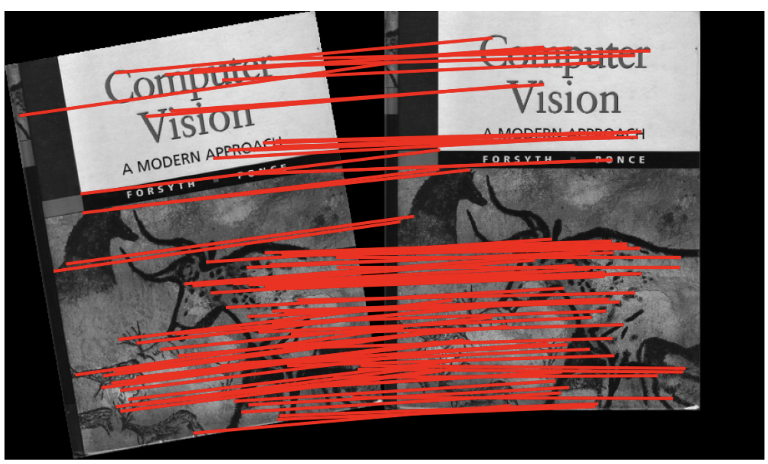
Reducing the ratio increases the quality of the matches as the number reduces, but this is not the case with increasing sigma.

Q2.1.6

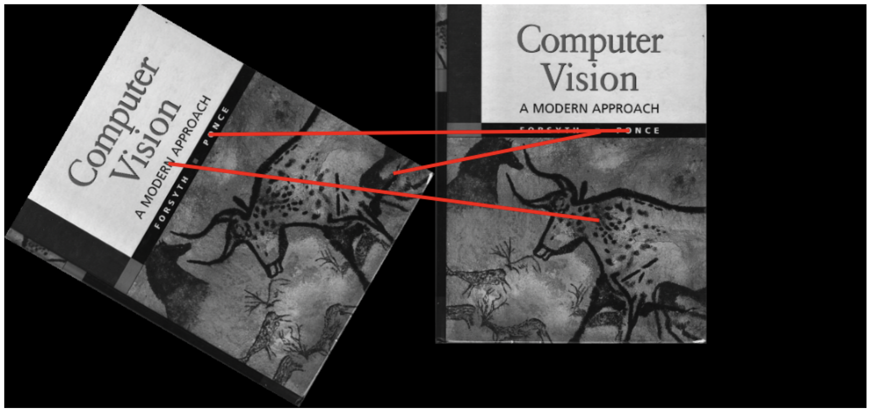


Histogram plot of the number of matches for the degrees the image is turned by

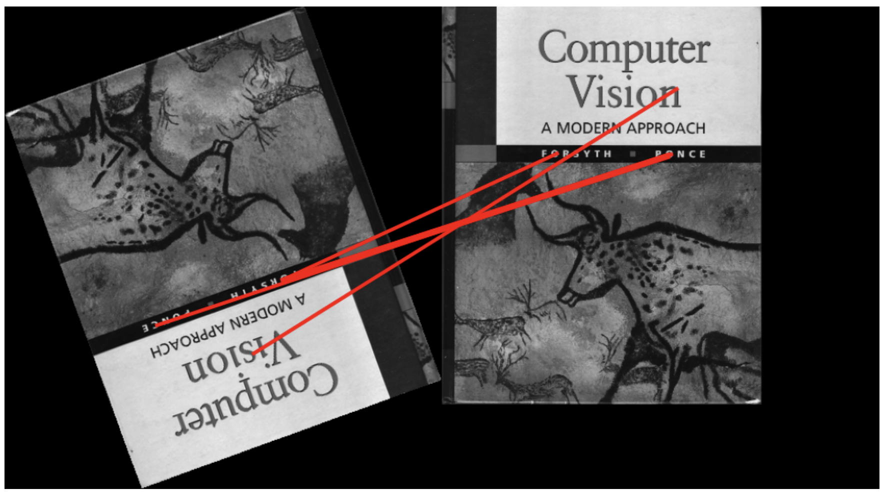
BRIEF as a feature detector is sensitive to translation and not rotation since it evaluates patterns based on intensities around a pixel which changes if the image is rotated. At 10 and 350 degrees, the rotation is small enough for this not cause a very large effect.



Plot of the features matched between the image and it rotated by 10°



Plot of the features matched between the image and it rotated by 100°

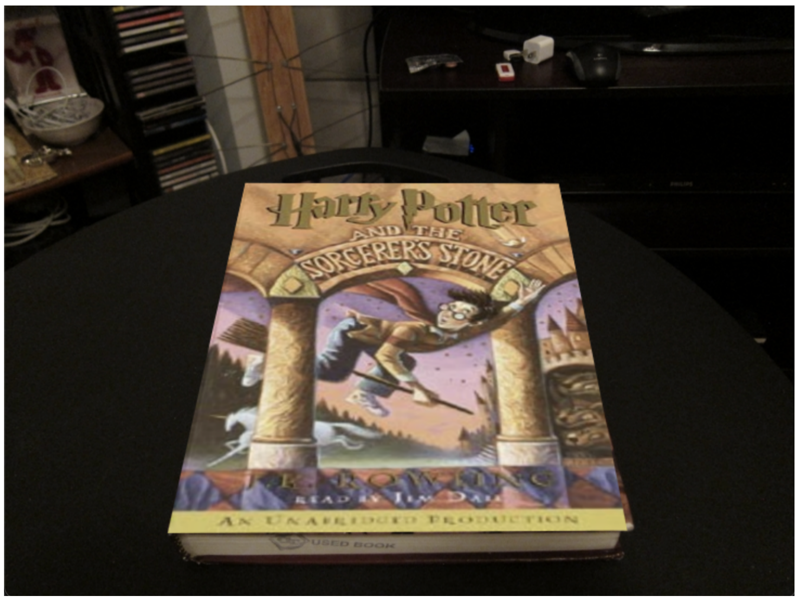


Plot of the features matched between the image and it rotated by 200°

Q2.2.4

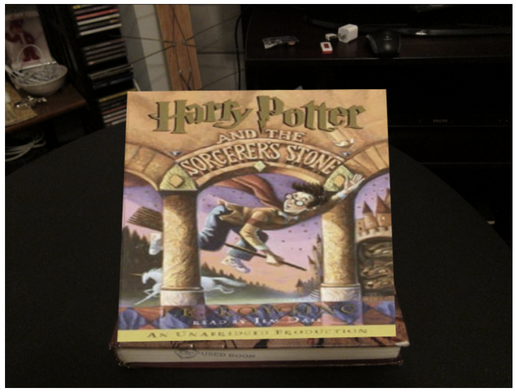
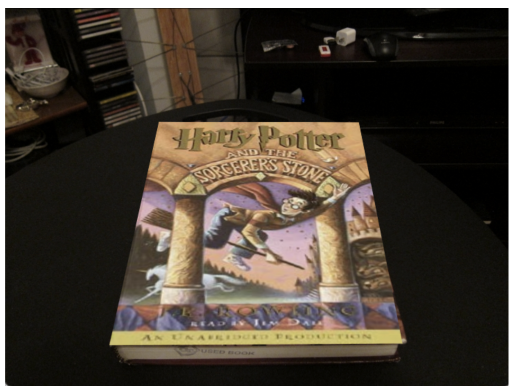
4) The image is not filling up the entire space since the harry potter cover is of a smaller size than the cv_cover image, which is used for evaluating the homography matrix. This can be rectified by resizing the harry potter image before warping it.

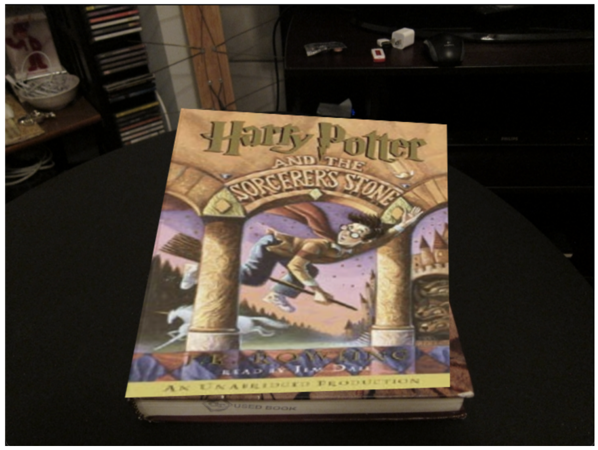
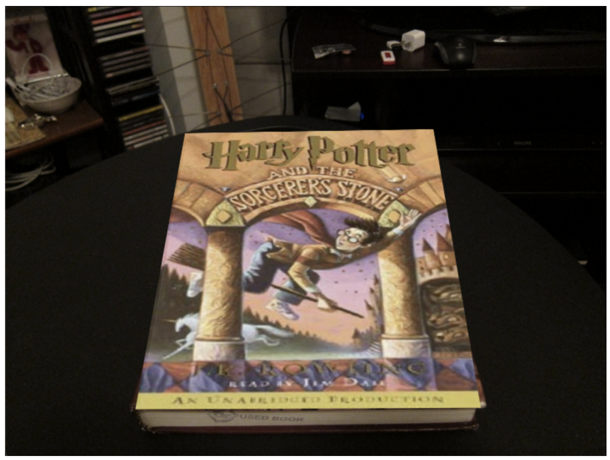
6)



Screenshot after HarryPotterizing the Computer Vision Text book

Q2.2.5

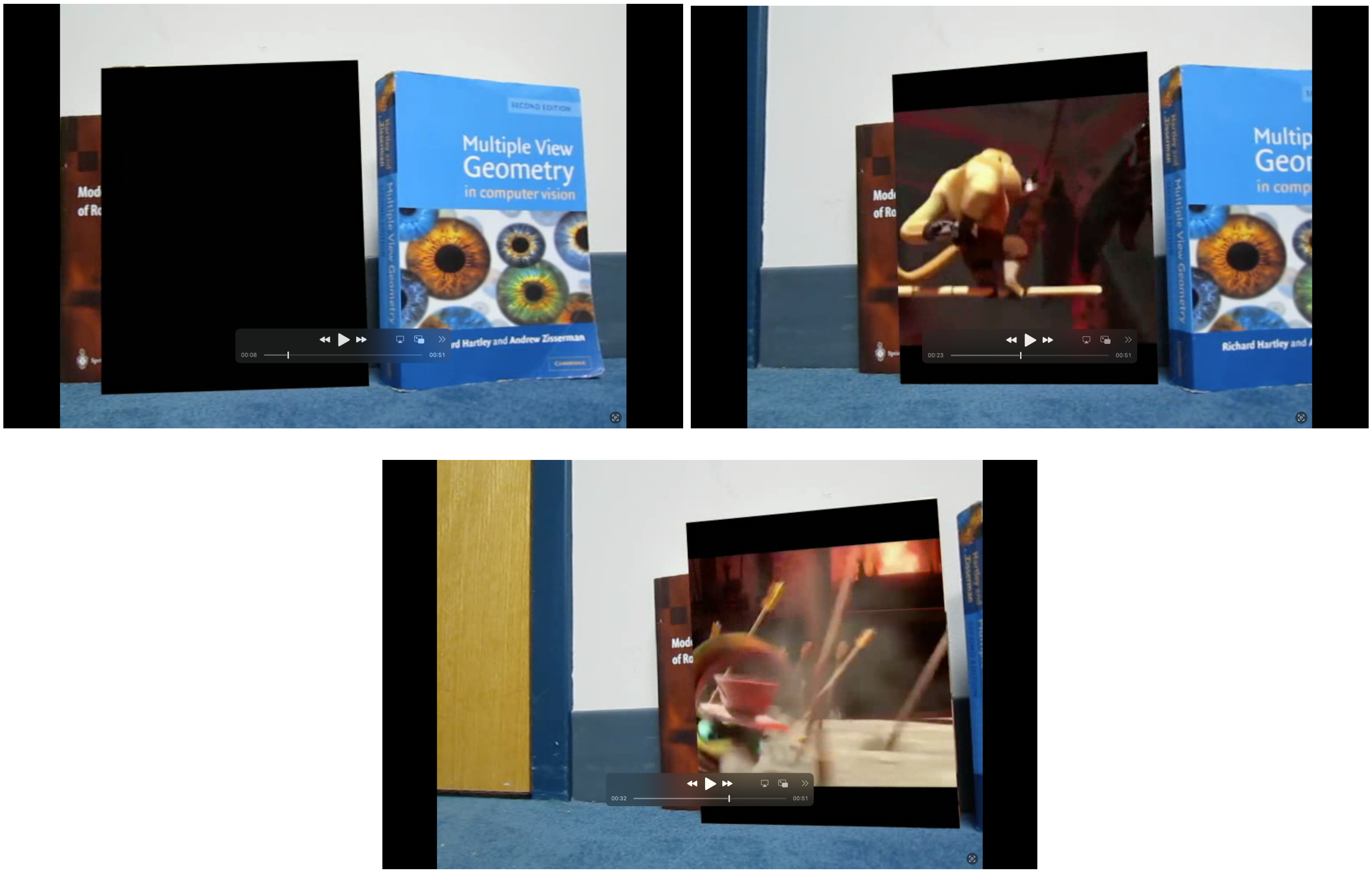
Max_iters	Inlier_tol	Image
500	1.0	
500	2.0	
500	50	

200	2	
2000	2	

If the inlier tolerance is increased by too much, bad matches are considered and it leads to a bad warp. On similar lines, if the inlier tolerance is decreased, not enough matches are considered leading to a bad warp.

If the number of iterations are reduced, the probability of getting a good result decreases, but if it is increased beyond a certain level, it reaches a point of diminishing returns since the result stays the same and only the amount of computation increases.

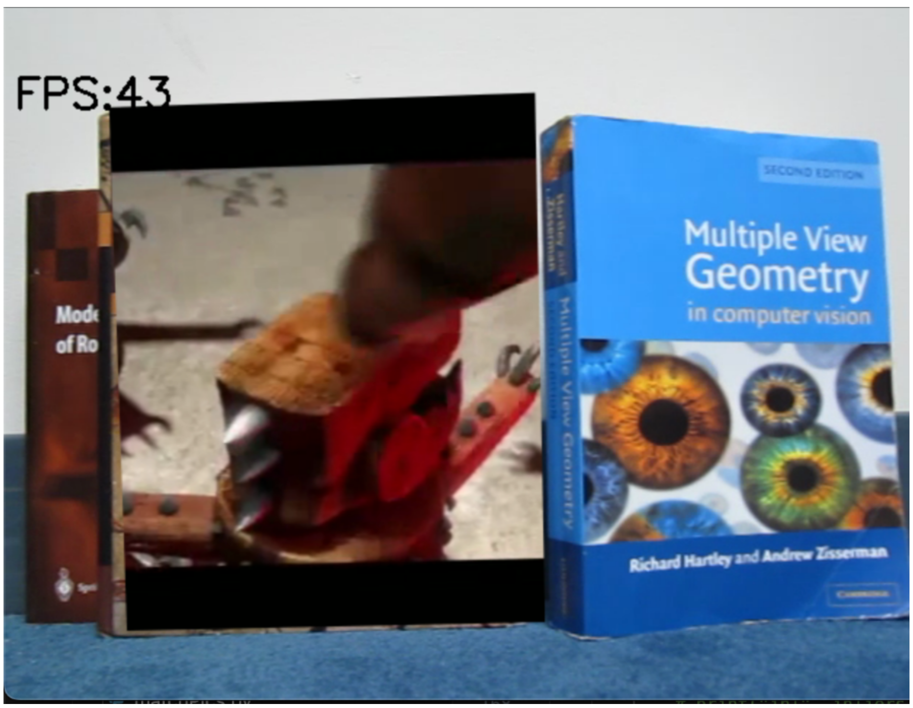
Q3



Screenshots at 3 distinct timesteps where the image is projected on the left, centre, and right of the frame

Q4.1x

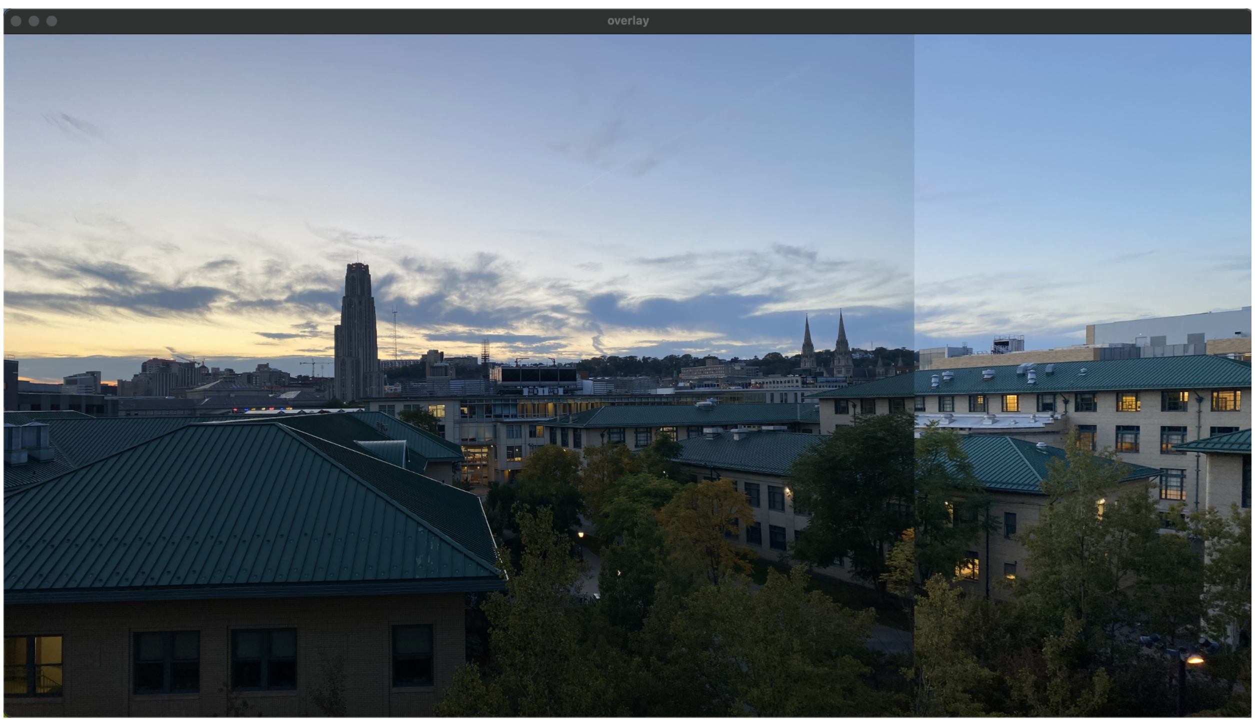
For making the AR real time, I started by reading the video frame-by-frame instead of loading all the frames into the memory since that takes a long time. I then replaced the FAST+BRIEF system with an ORB matcher. To further improve the speed, i replaced the computeH_ransac with the CV2 library function findHomography().
With these changes, the video reaches **35 FPS.**
But, the video is very jittery, so instead of ORB, I tried using SIFT as the feature descriptor. With SIFT, the video result is very stable, but since it is a computationally heavy task, it maxes out at 15 FPS.



Screenshot of the real-time video playback

Q4.2x

Images used as the left and right images in the panaroma



Resultant panaroma