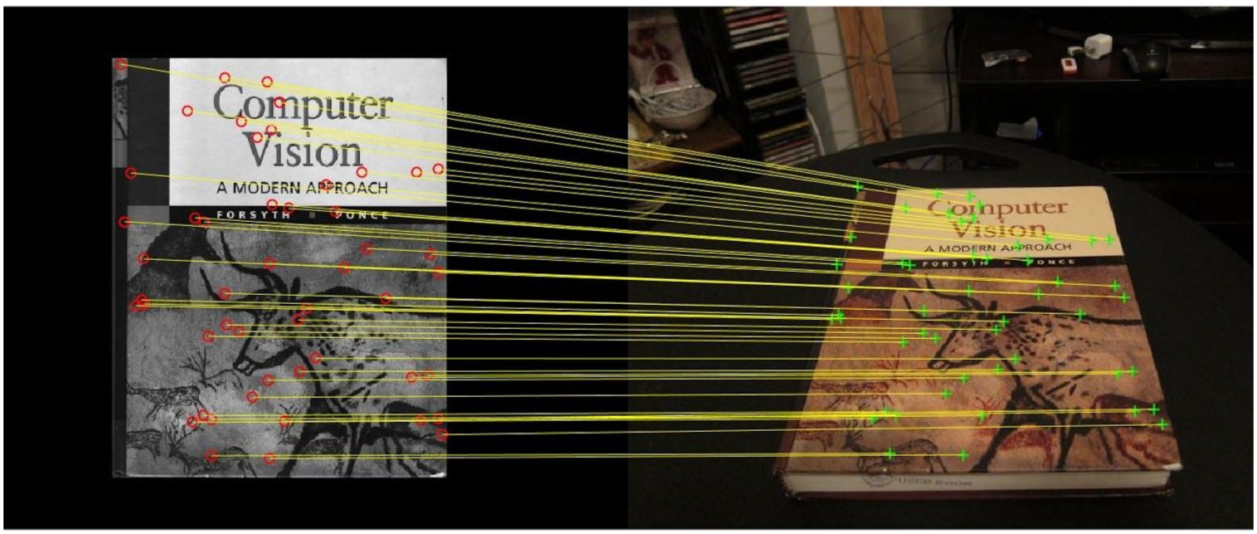
CMPT 412

Assignment 2

Augmented Reality with Planar Homographies



4.1 Feature Detection, Description, and Matching

In this part I simply convert both the images to grayscale using the rgb2gray function and follow the procedure in the handout to detect features, compute BRIEF descriptors, and finally match the features using the builtin matchFeatures function. Figure 4.1.1 is a visualization of the matched features. The ratio parameter was set to 0.6 to achieve the result below.

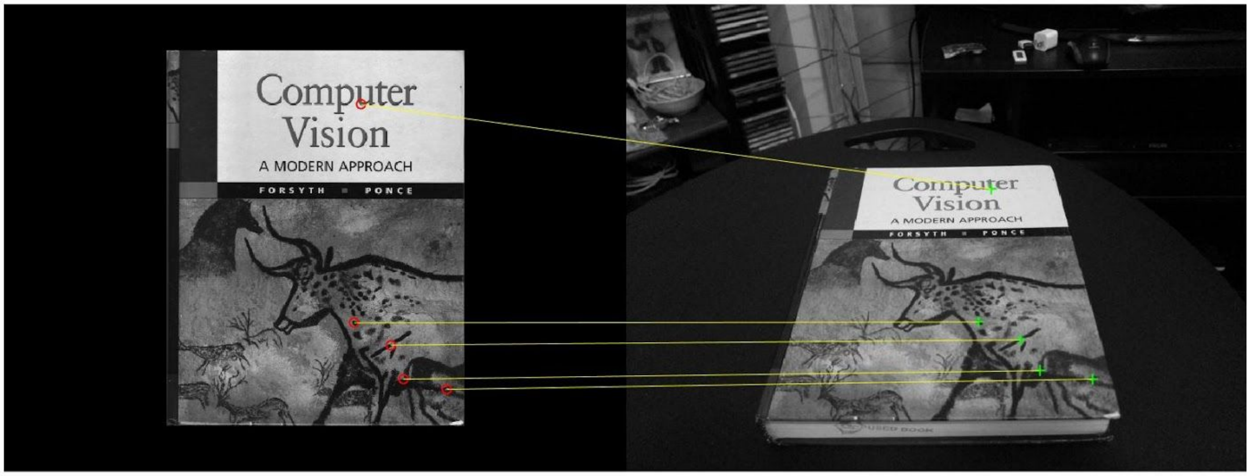


Figure 4.1.1

4.2 BRIEF and Rotations

In figure 4.2.1 we can see the plots for using BRIEF (right plot) and SURF (left plot) and analyzing their behaviour regarding rotation. We can see that SURF is much more robust when it comes to rotation and BRIEF fails to get good matches by even slight rotations. This is because BRIEF is rotation variant and does not take into account the orientation, whereas, SURF does. We can see the performance of SURF is much better at 90 degree intervals and decreases at other rotations, however, it is still much better than BRIEF.

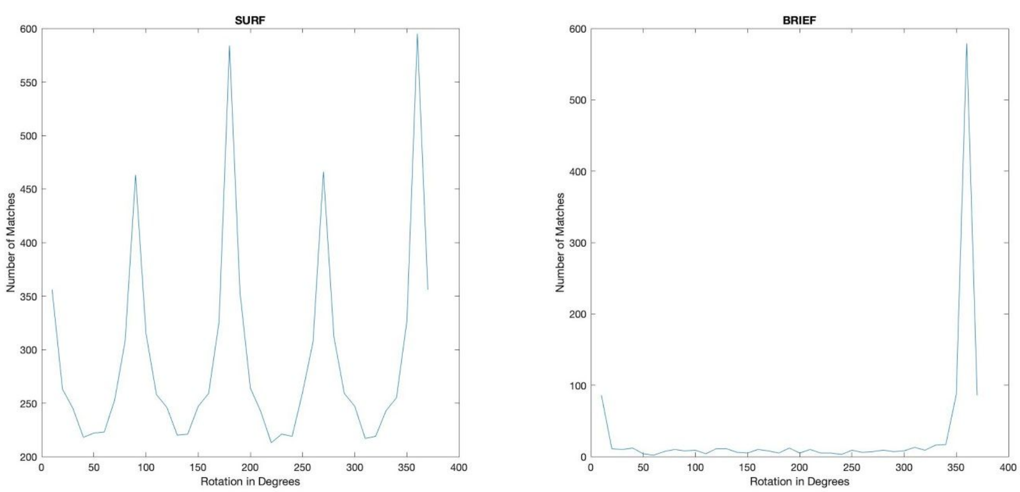


Figure 4.2.1

We will also compare the feature matching at 3 different orientations for both BRIEF and SURF. In figures 4.2.2, 4.2.3, and 4.2.4 we will visualize BRIEF at 45, 90, and 120 degree rotation respectively.

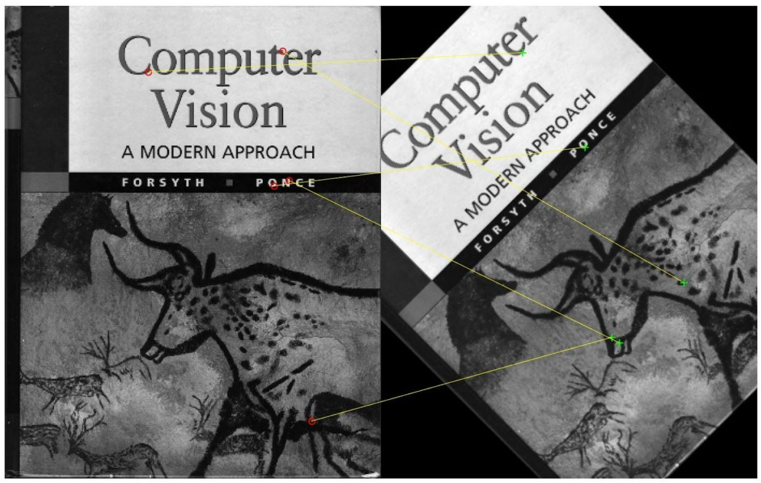


Figure 4.2.2

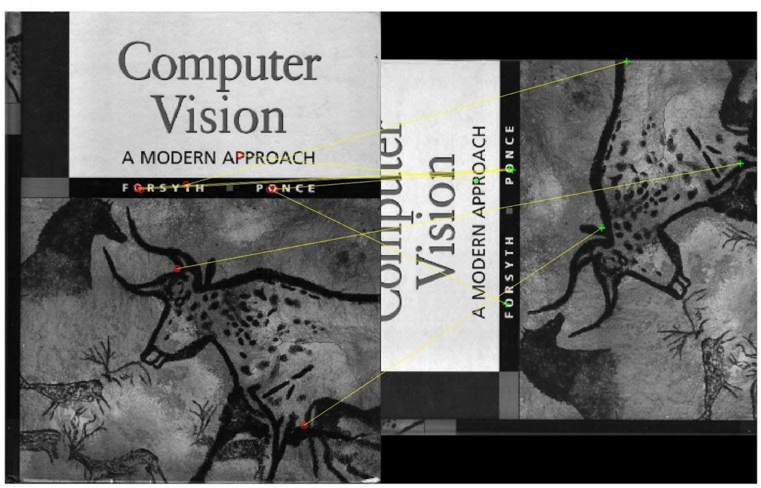


Figure 4.2.3

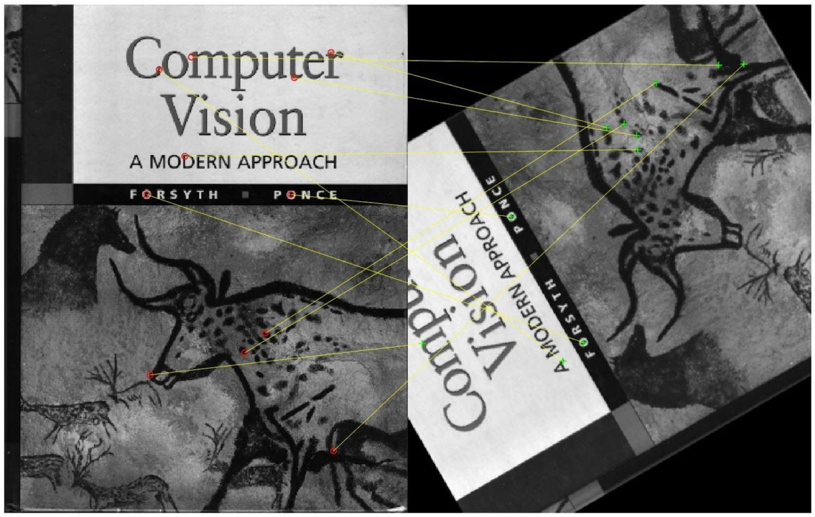


Figure 4.2.4

We can see that the number of matches detected by BRIEF after different rotations are quite low. In figures 4.2.5, 4.2.6, and 4.2.7 we will visualize the same rotations but by using SURF instead. We can see how the number of matches are much higher after rotation.

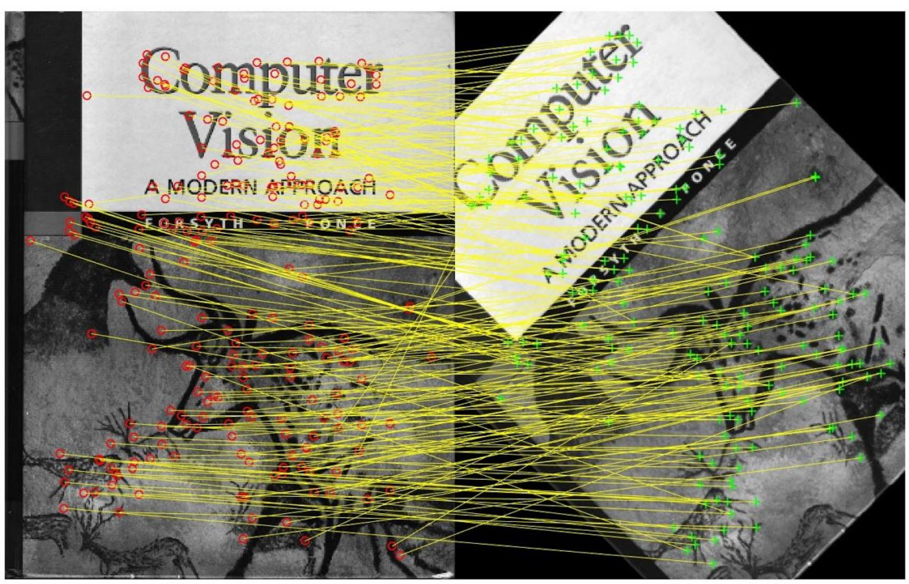


Figure 4.2.5

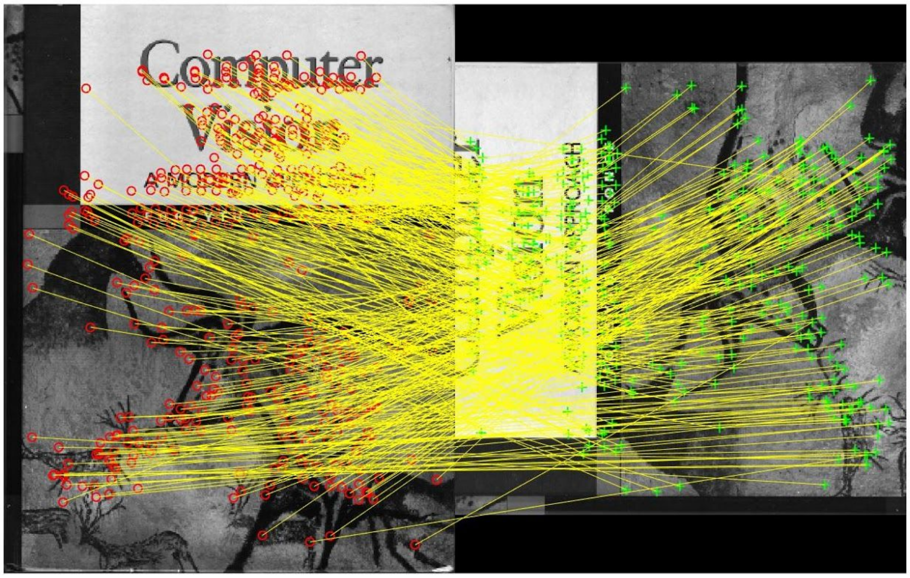


Figure 4.2.6

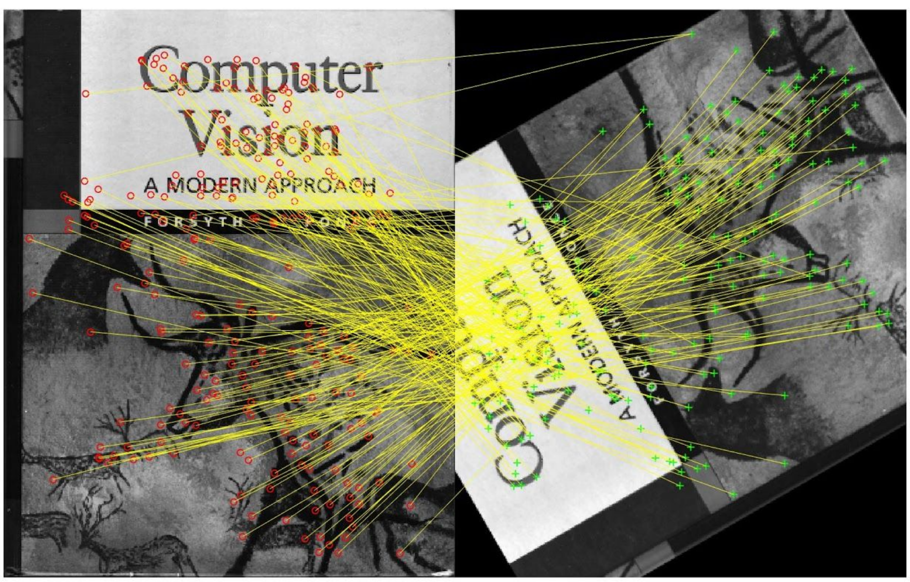


Figure 4.2.7

4.3 Homography Computation

In this section, we are going to estimate the homography transformation by using the SVD method. Below in figure 4.3.1 we will visualize the accuracy of the estimated homography by transforming 10 randomly chosen points from the first image and displaying them on the second image.

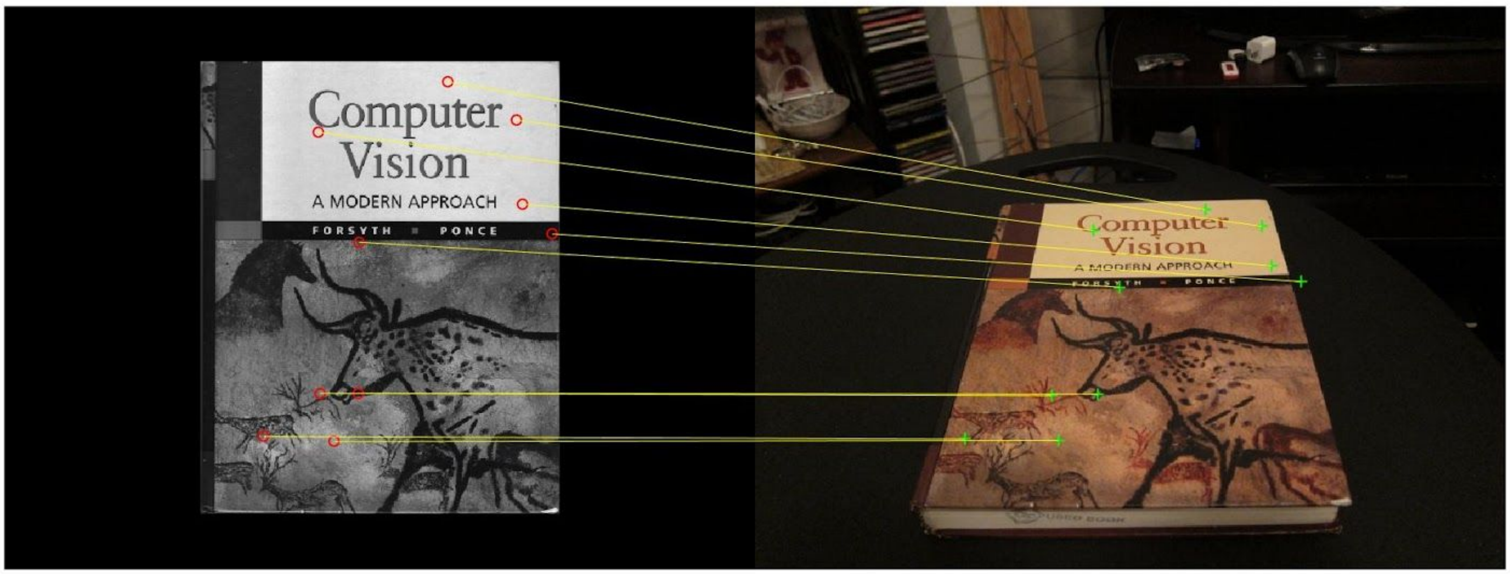


Figure 4.3.1

4.4 Homography Normalization

In this part, we are going to normalize the matched points and then compute the homography matrix H. Below in figure 4.4.1 we will visualize the accuracy of the estimated homography on normalized matched features by transforming 10 randomly chosen points from the first image and displaying them on the second image.

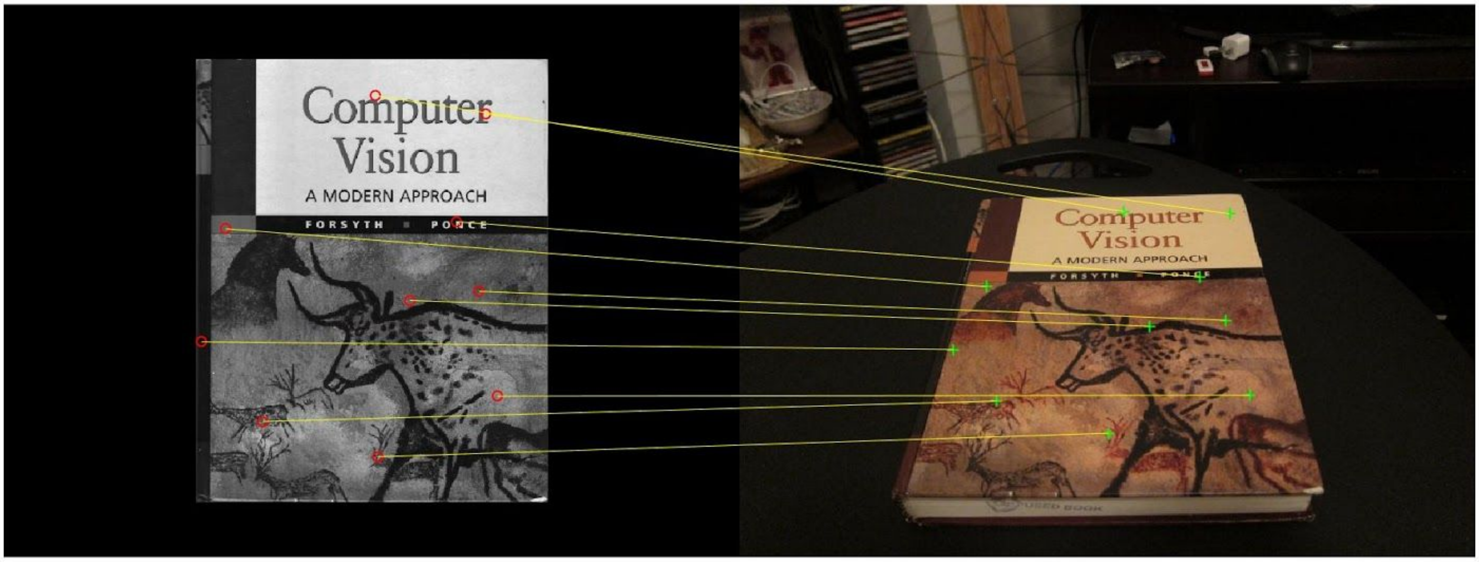


Figure 4.4.1

4.5 RANSAC

In this section, we are going to demonstrate the RANSAC algorithm. The number of iterations was set to 10000 for maximum accuracy. Lower iterations would sometimes produce unwanted homography. The threshold distance was set to 2 as well to get pinpoint accuracy on the transformation. In figure 4.5.1 we will see the 4 points that produced the maximum number of inliers. In figure 4.5.2 we will visualize all the inlier matches selected by the algorithm when those 4 points were chosen. In figure 4.5.3 we will also visualize 50 random points and their transformation using RANSAC. We can see how accurate the RANSAC algorithm actually is.

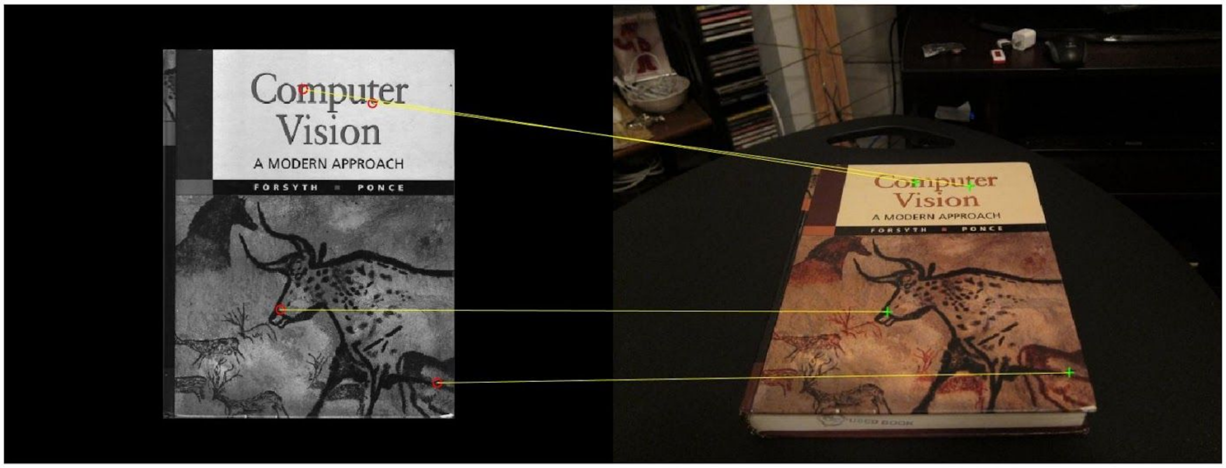


Figure 4.5.1

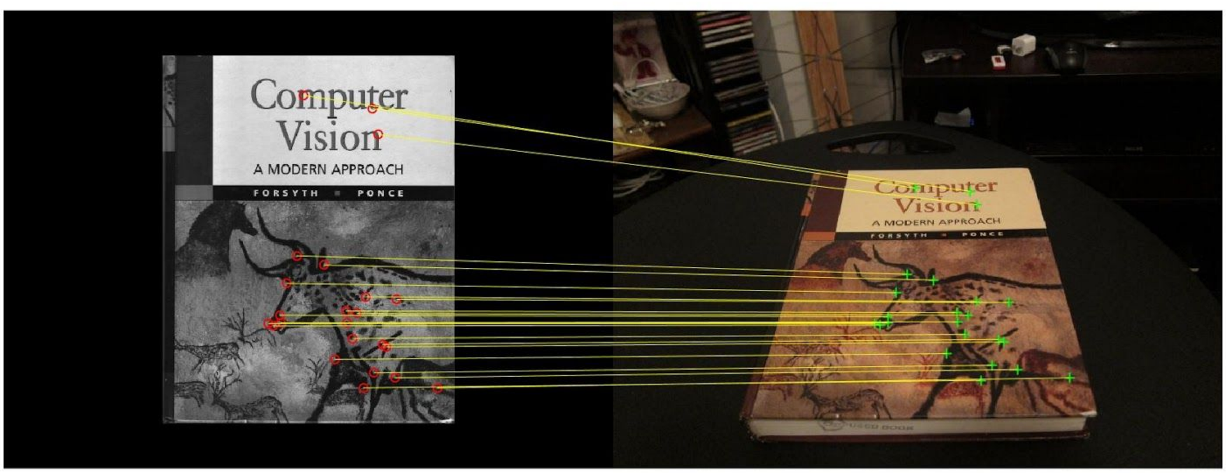


Figure 4.5.2

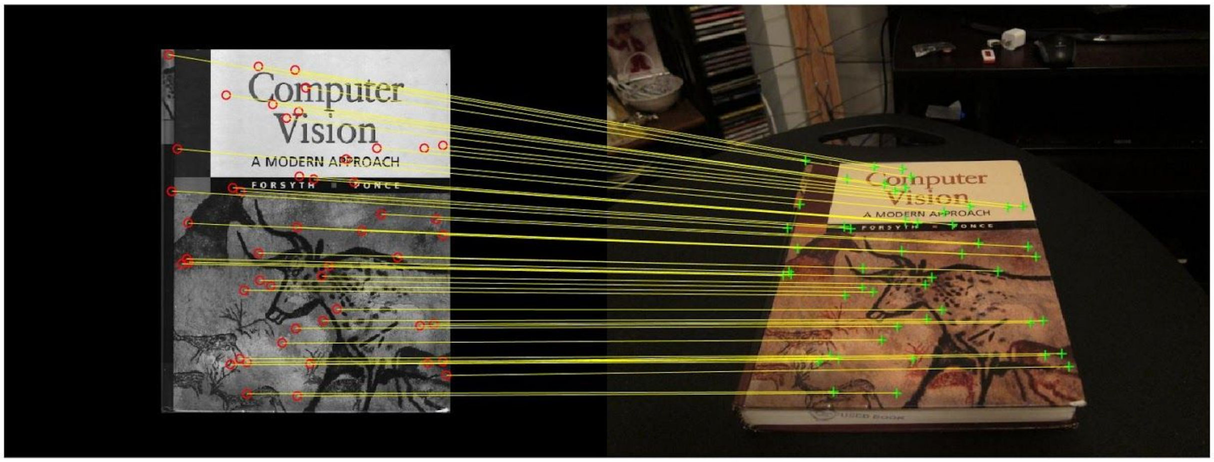


Figure 4.5.3

4.6 Harry-Potterizing a Book

In this part, we are going to use the RANSAC algorithm and harry potterize the book on the desk. The compositeH function was implemented to essentially replace the existing book on the desk with the harry potter book. This was done by using a mask the same size as the desk image. This mask has all its values set to 0 except the parts where the warped harry potter book is present, at which points it is equal to the harry potter book. Then we simply, make the composite image equal to the desk image at points where the mask is 0, and equal to harry potter book where the mask is set to 1. In figure 4.6.1 we can see the effect.

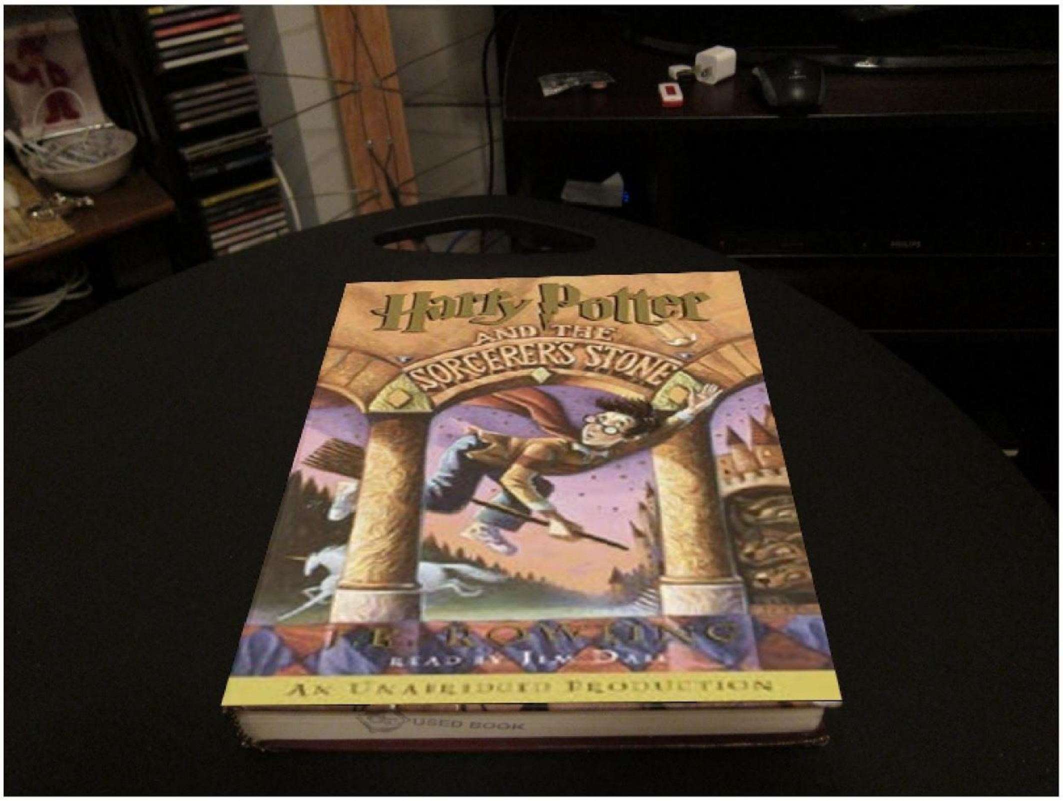


Figure 4.6.1

5. AR Application

In this part, we essentially do the same thing as above, but to every frame of the book movie. Moreover, instead of overlaying it with the harry potter book, we overlay it with the frames from the source movie. First, we detect where the black bars are in the source movie by running a simple loop and summing each row until we hit a row where the sum is high at which point we know we have reached the actual footage. Moreover, since the source has less frames than the book video, I simply start the source from the beginning when I run out of frames. Moreover, to deal with entirely black frames, I summed the entire frame over all dimensions and realized that the sum is not actually 0! Therefore, I sum every frame across all dimensions and if the sum is less than 500000, I set the whole frame to 0 to make the task easier on the mask and composite part. This was an anomaly, and although black frames exist in the source movie, the green channel seems to not be completely 0 and only red and blue channels are 0, Therefore, 500000 was a reasonable assumption for black frames and the results are very good as apparent in the final footage. Also for the horizontal cropping, I simply display divide the number of columns in the source frame by 3 and only display the middle third which is approximately from (213:426).