CMPT412 Project 4 Report

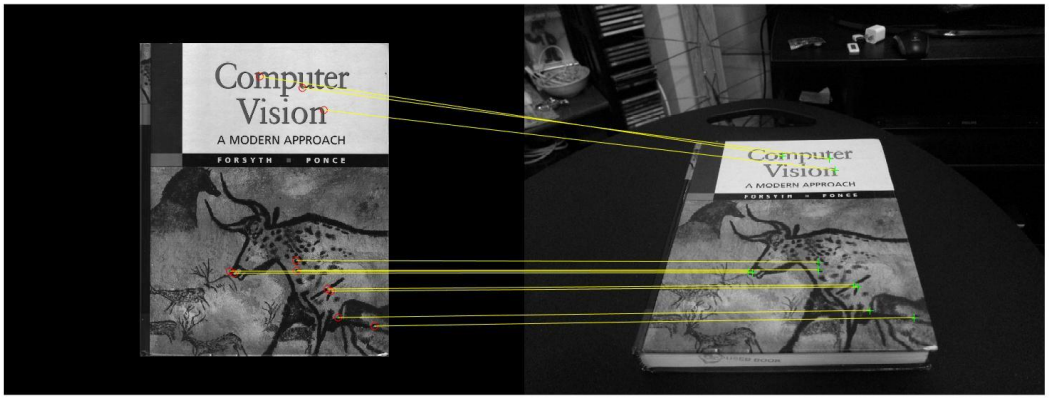
Ritika Goyal (301401516)

Project 4: Augmented Reality with Planar Homographies

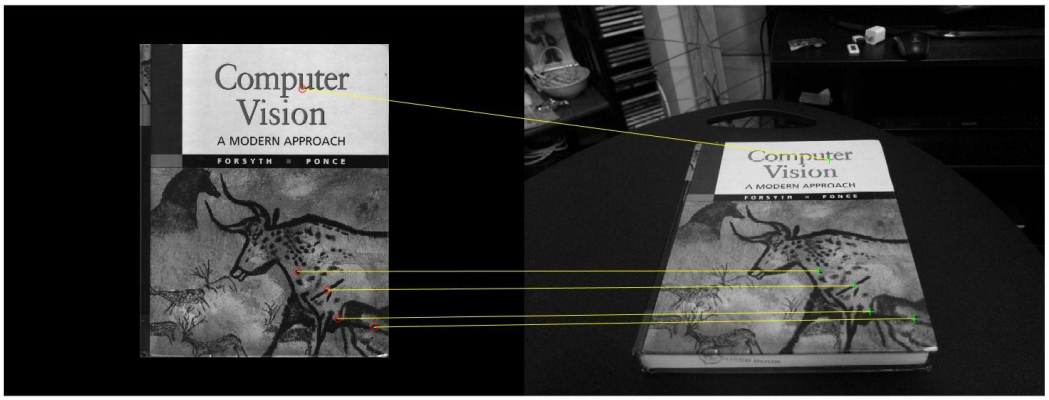
Computing Planar Homographies:

Part 4.1: Feature Detection, Description, and Matching

I tried various maxRatios in matchFeatures() function and the best results came with 0.68. I first converted the coloured images into grayscale and then applied matchFeatures on it. I used MATLAB's function detectFASTFeatures(). The image representation of matched features with MaxRatio of 0.68 is displayed below:



The default maxRatio is 0.6 and the number of match pairs appeared less. The results are displayed below with 0.6 MaxRatio:

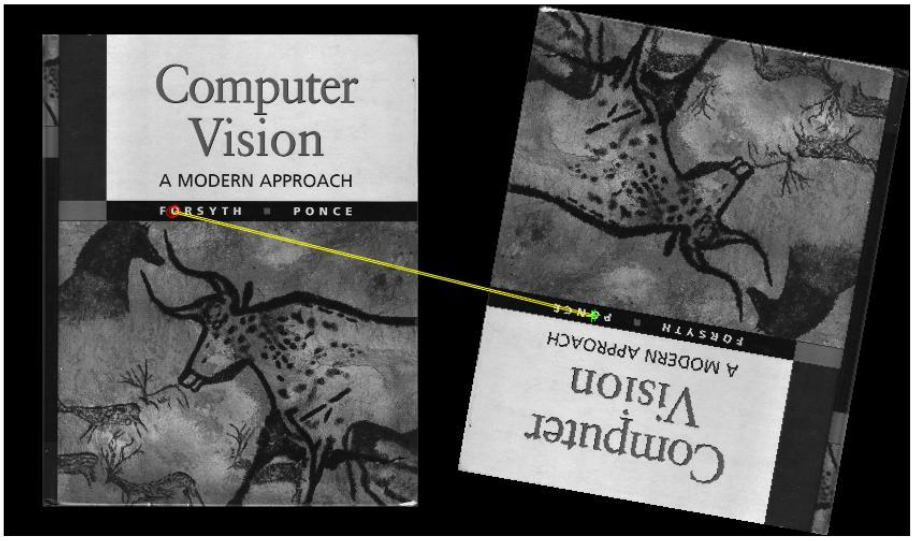


Part 4.2: BRIEF and Rotations

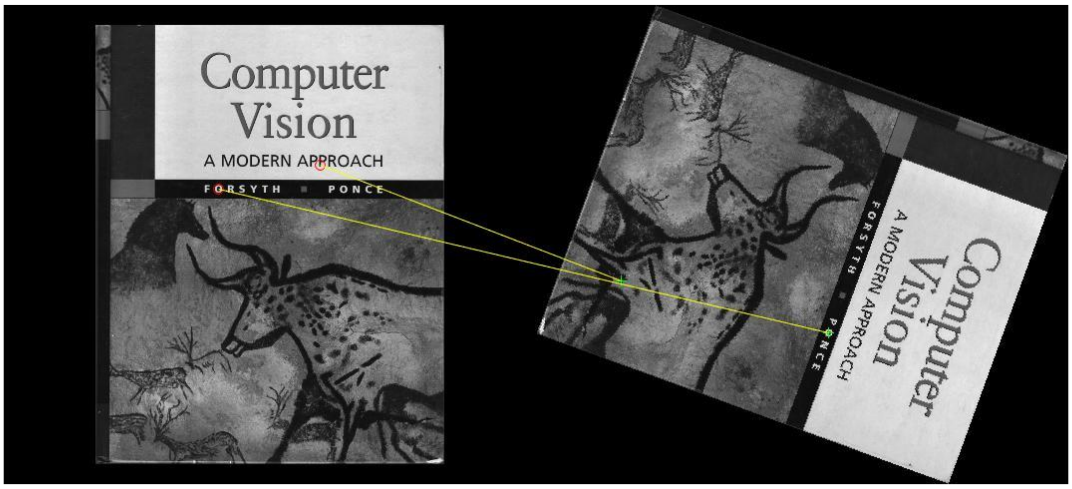
Results of rotating images using Brief Descriptor:

100 degrees rotation:

170 degrees rotation:

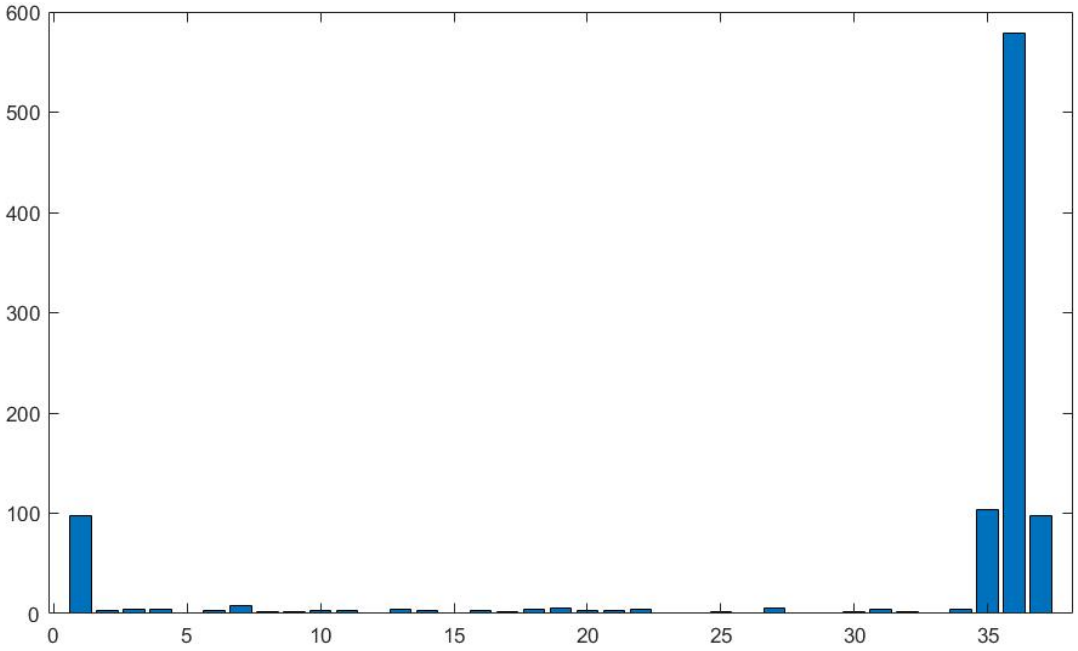


250 degrees rotation:



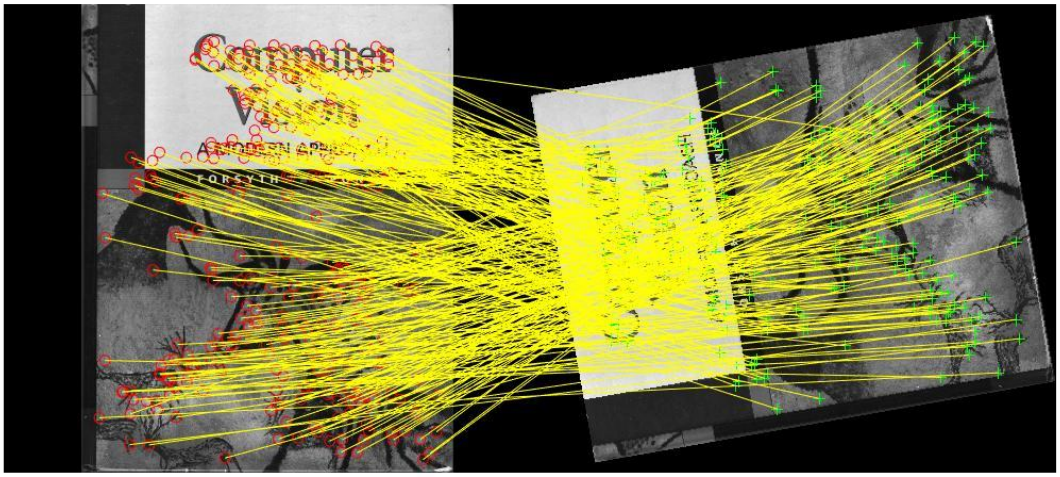
350 degrees rotation:

Histogram:

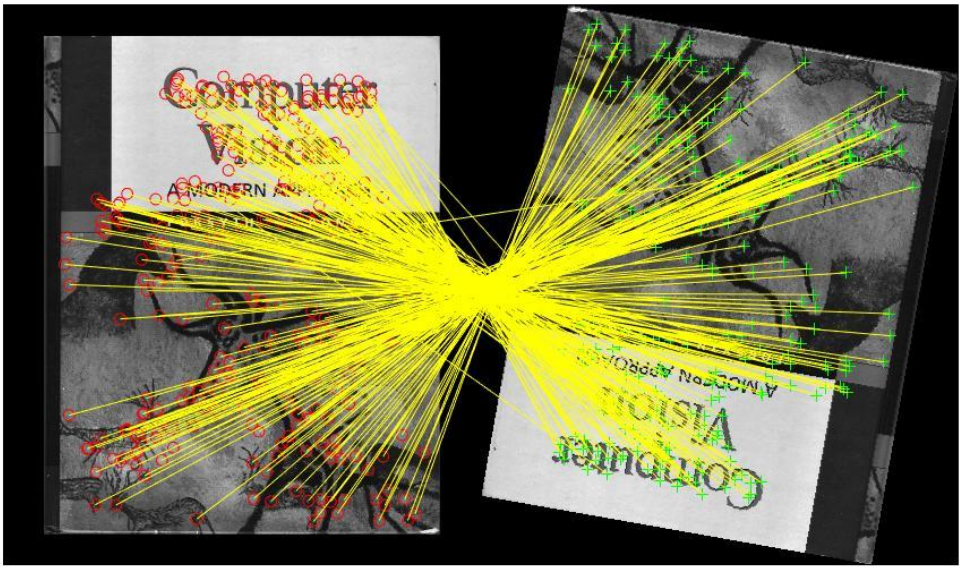


Results of rotating images using SURF Descriptor:

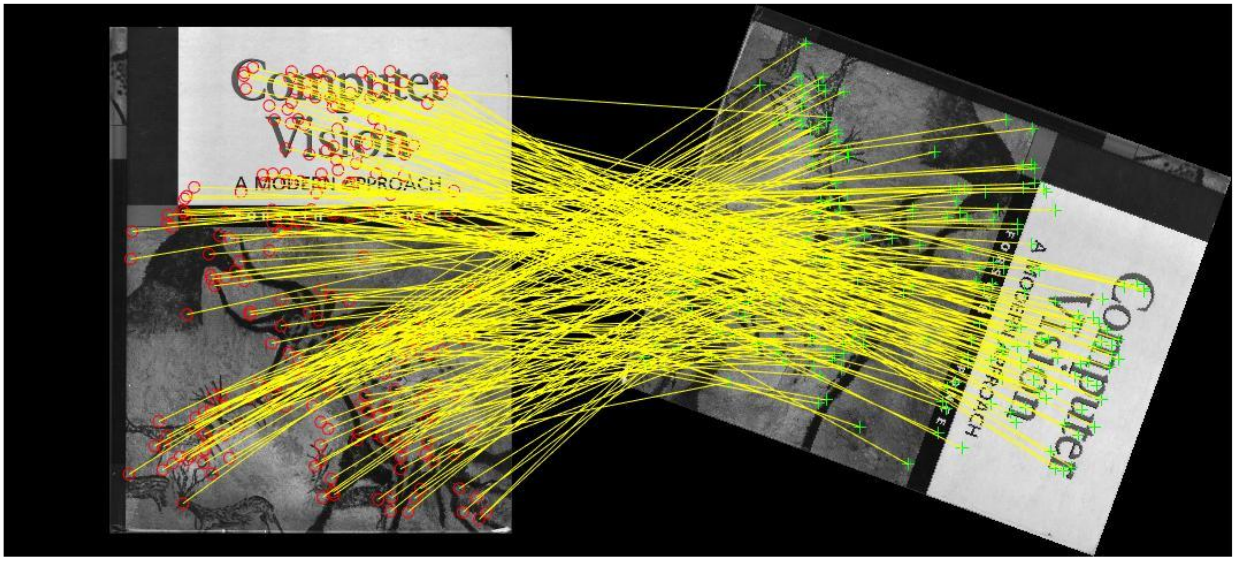
100 degrees rotation:



170 degrees rotation:

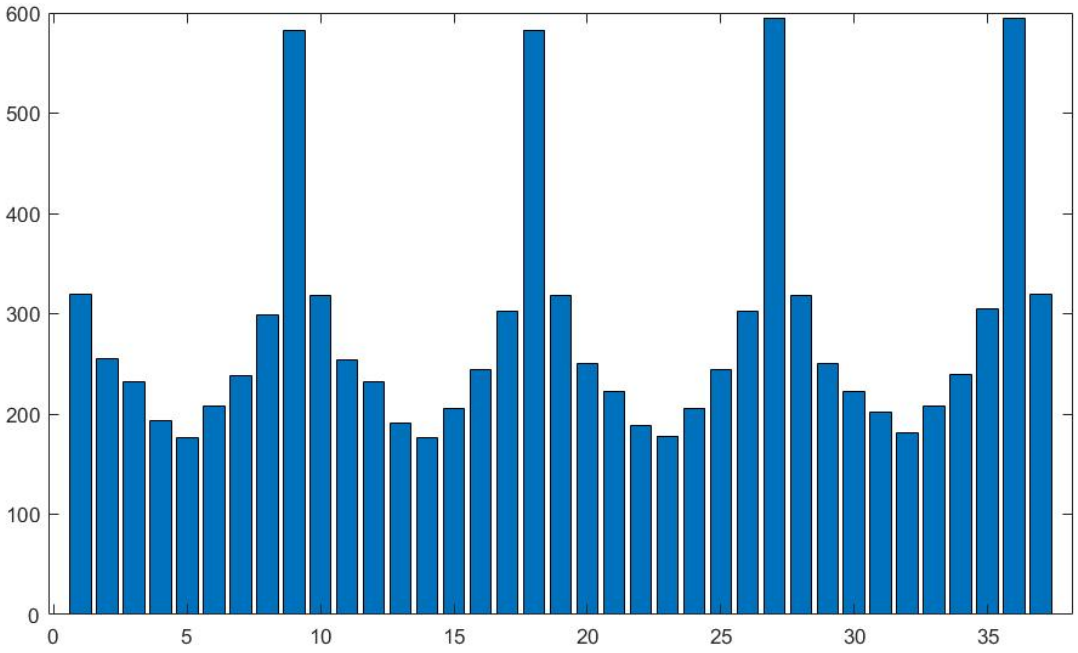


250 degrees rotation:



350 degrees rotation:

Histogram:

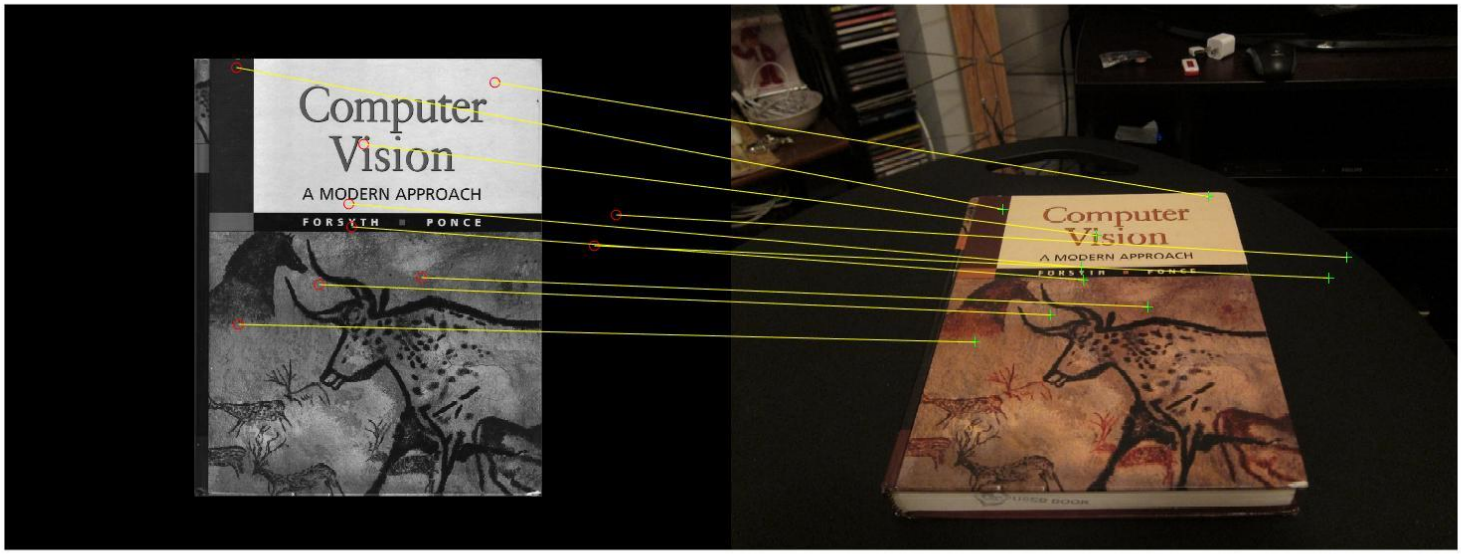


When looked at the result, It is clear that with rotation, SURF performs better than BRIEF. I plotted the histogram with angle of rotation increasing 10 degrees every time. The results show that Brief only works better with small degrees and some of the higher degrees. As seen in histogram, from 3- 34 degrees, it does not perform well at all.

SURF performs better than BRIEF and the highest performance is near the degrees 90-100, 180-190, 270-280, 350-360.

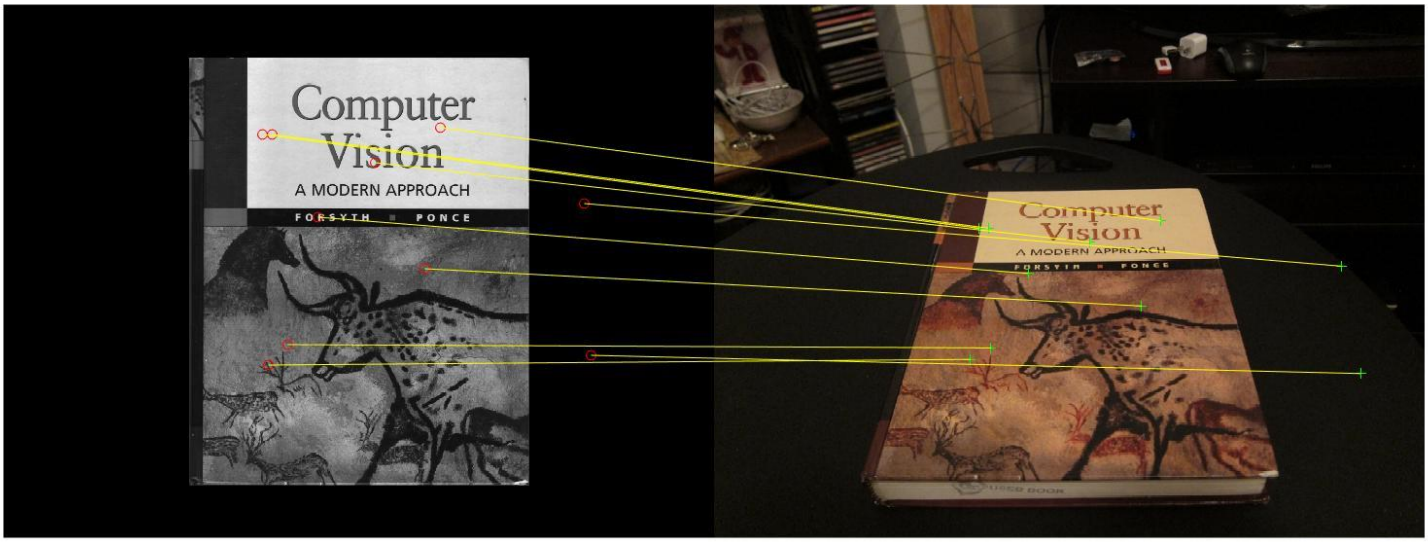
Part 4.3: Homography computation

First, I used matchPics to get the matched points on bith the images and then I computed homography matrix H using computH.m. After that I took 10 random points from the image and displayed the corresponding computed points from Homography on the other image. I made a separate script called result_homography_transform.m to displaye the points. The results are shown below:



Part 4.4: Homography Normalization:

I translated the mean of the points to the origin and then scaled the points so that the average distance to the origin is sqrt(2). I then got the transformation matrix using feature points. Then I took 10 random points from the image and displayed the corresponding computed points from Homography on the other image using computeH_norm.m. It is implemented in script result_homography_transform.m. The results are shown below:

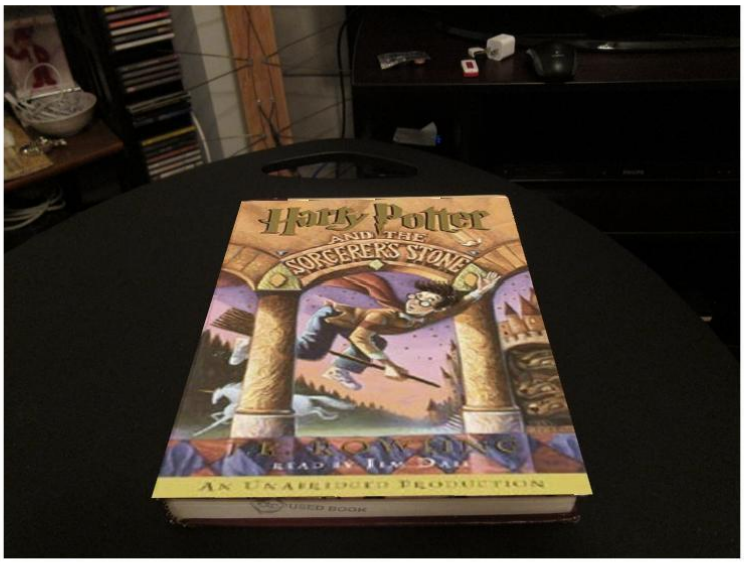


Part 4.5: RANSAC

I used Ransac algorithm to fit the noisy data. The results of using computeH_ransac.m is shown below and it is computed in result_homography_transform.m:

Part 4.6: HarryPotterizing a Book:

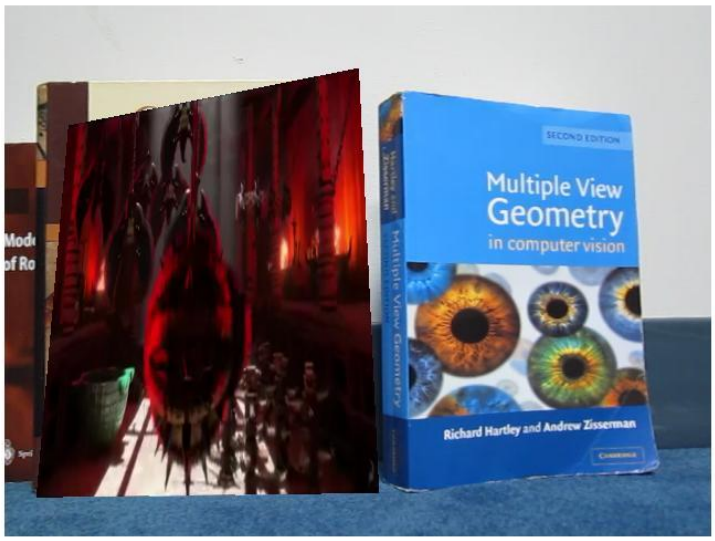
The results of displaying harry potter cover on the cv_desk using computH_ransac is shown below:



Part 5: Creating your Augmented Reality application

The Augmented Reality application is implemented in ar.m. The output video is saved in results folder as 'movie.avi'. I am also attaching some pictures below to describe the different frames of the video:


Multiple Vie
Geometr
in computer vis
Richard Hartley and Andrew Zis


SECOND EDITION
Multiple View
Geometry
in computer vision
Richard Hartley and Andrew Zisserman