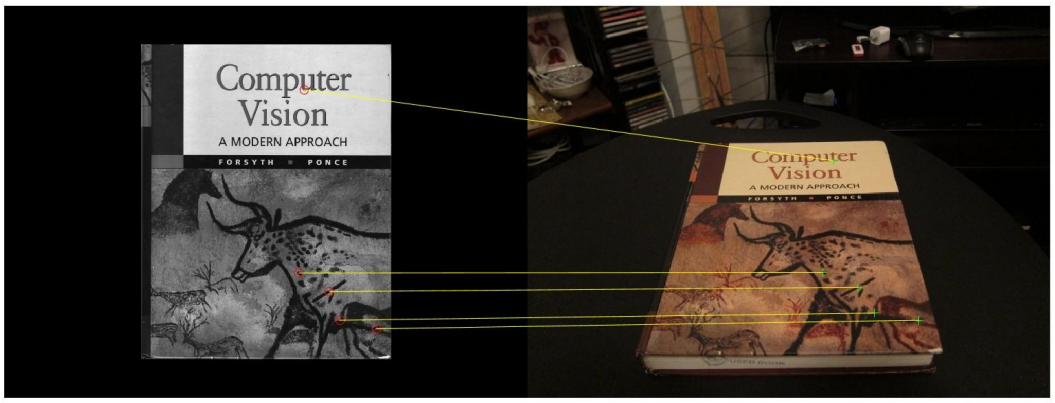
CMPT412 Project 4
Digit recognition with convolutional neural networks

Student: Sibei Zhou(301416917)
Computer ID: palamzad@sfu.ca
(use 1 free-late days)

1. Feature Detection, Description, and Matching

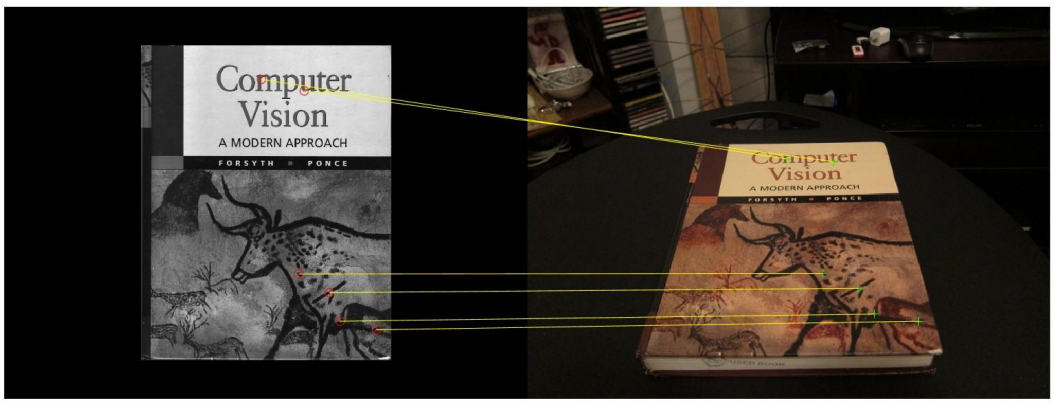
This section is implemented in a new script named “q1.m”, it is called the “matchPics.m” and shows the result.

(1) Default result with `'MatchThreshold'` are setted to 10.0

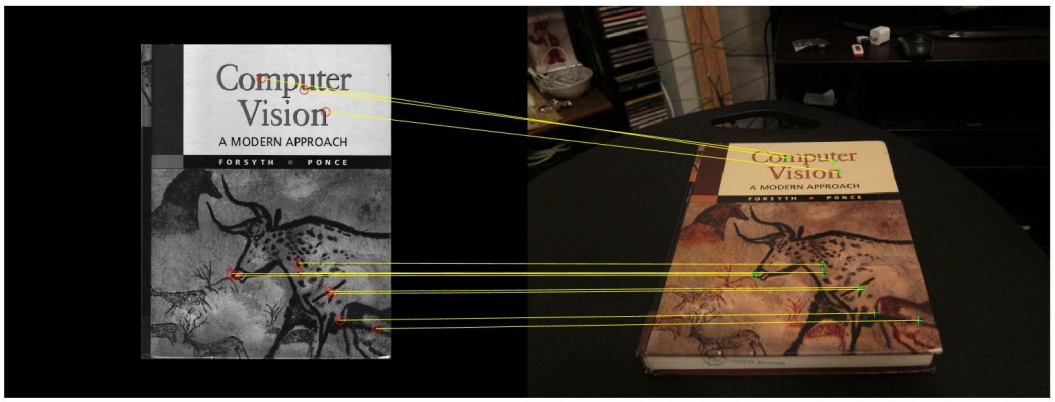


It only has 5 points matched. By increasing the MaxRatio parameter, I got a better feature matching.

(2) result with `'MatchThreshold'` = 10.0 and `'MaxRatio'`=0.65



(3) result with `'MatchThreshold'` = 10.0 and `'MaxRatio'`=0.68 (which is the best result we got)

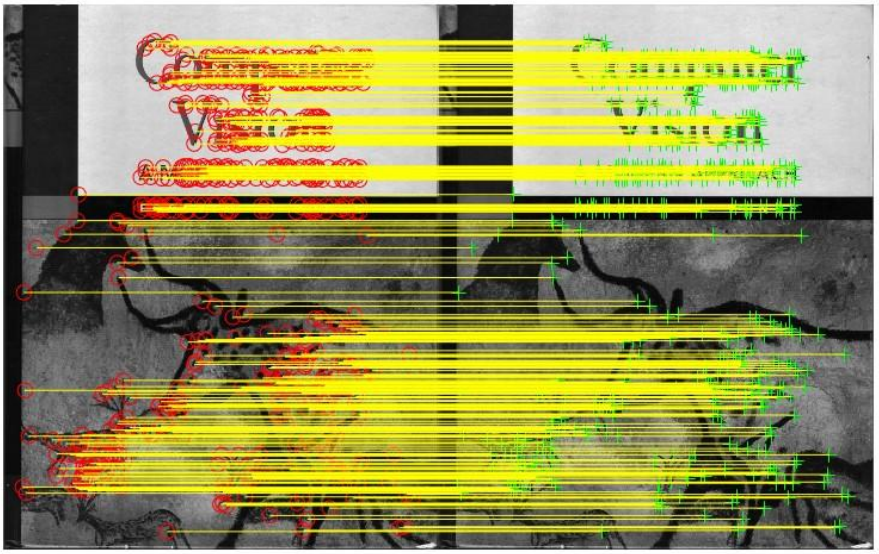


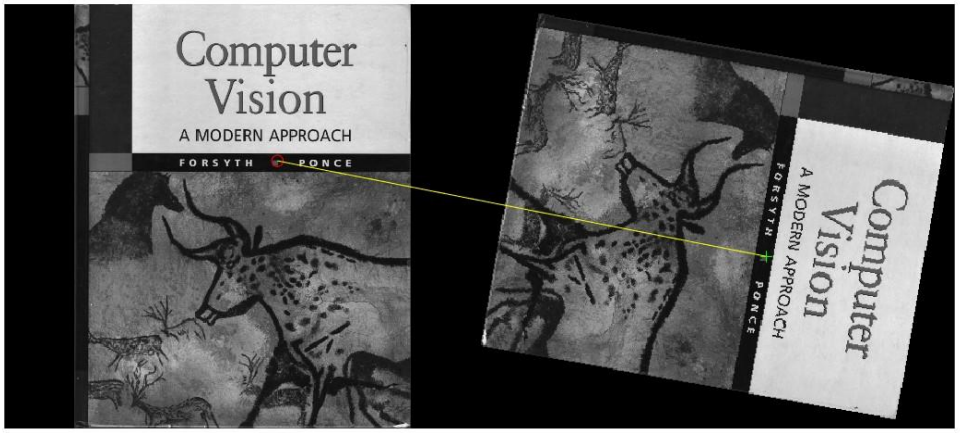
2. BRIEF and Rotations

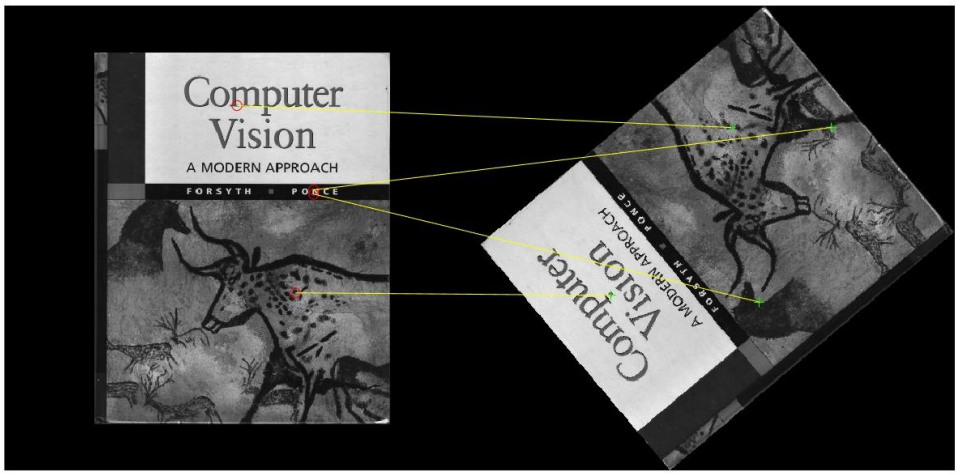
This section is implemented in “briefRotTest.m” and the first part of using detectFASTFeatures() and computeBrief() has been commented on.

(1) result using **detectFASTFeatures() and computeBrief()**:

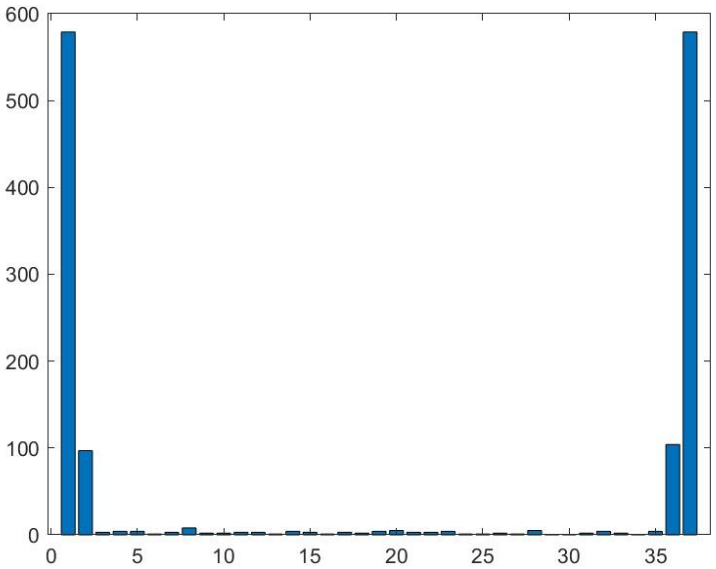
The feature matching **result at three different orientations**:





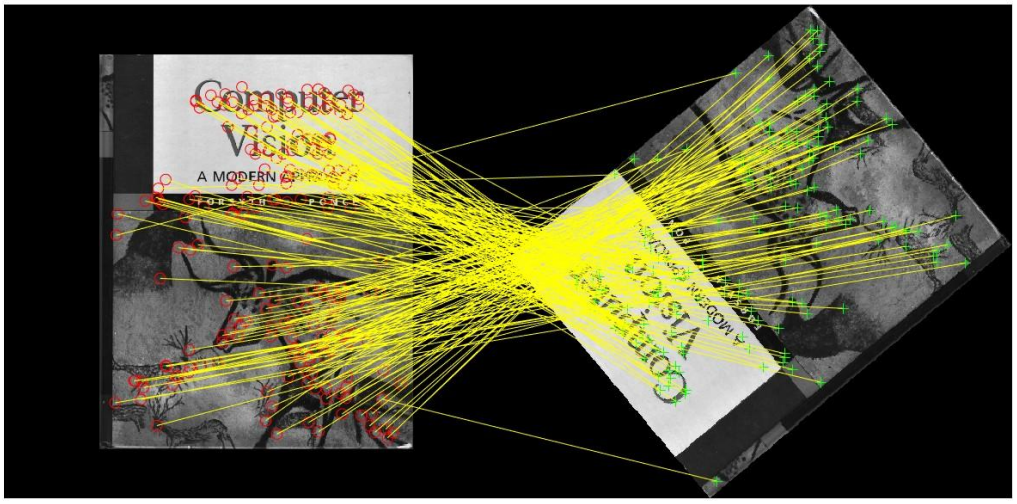


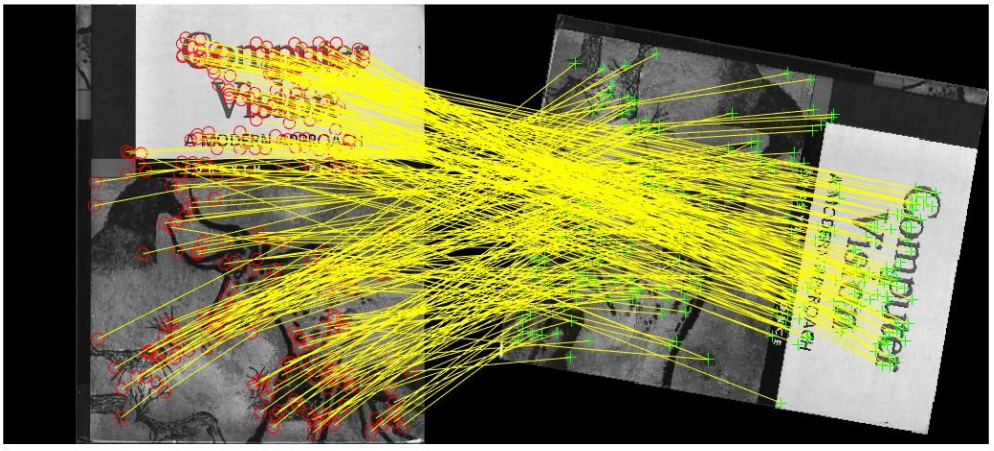
Plots the histogram:



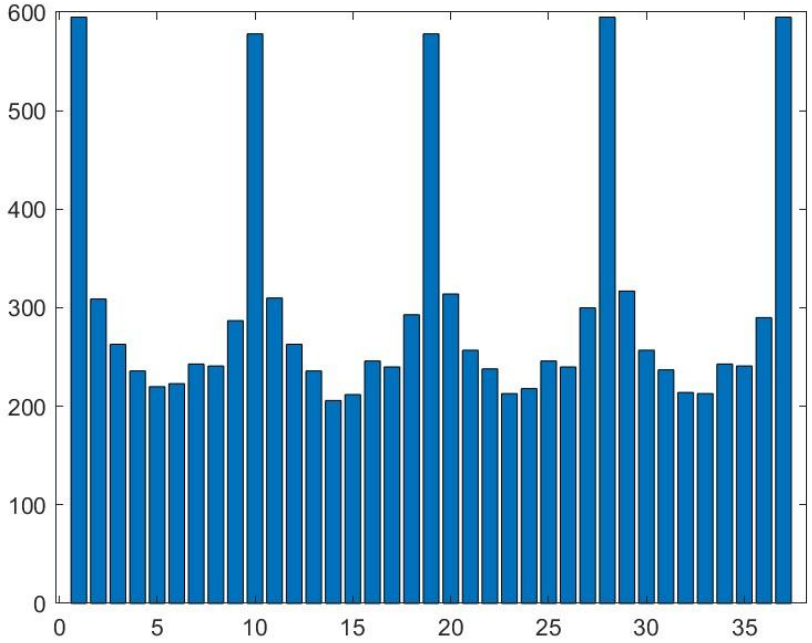
(2) result using **detectFASTFeatures() and computeBrief()**:

The feature matching **result at three different orientations**:





Plots the histogram:



It is obvious to see, SURF has a better performance during rotations, and seems to work the best when at every 90° of rotation (90°,180°, 270°).

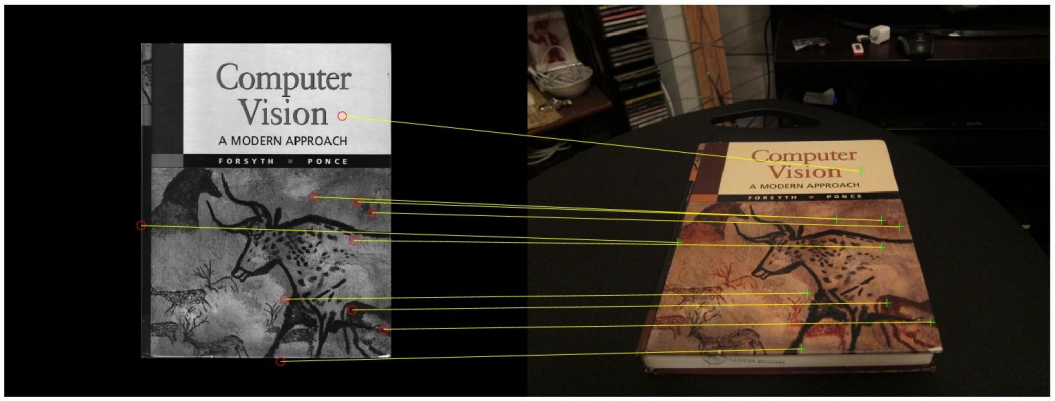
3. Homography Computation

To calculate the H matrix, using the formula below to form the left p matrix, then get the h vector by svd(), and finally change it to 3*3 matrix H2to1.

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i'x_i & -y_i'y_i & -y_i' \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0 \quad (20)$$

The function computeH is written in the “computeH.m” file, and after it’s done, the test for this part picks random 10 of points from the first image, and **shows the corresponding locations** in the second image after the homography transformation is in “q3.m”.

And the result is saved as “4_3.jpg” in /results and shown below.



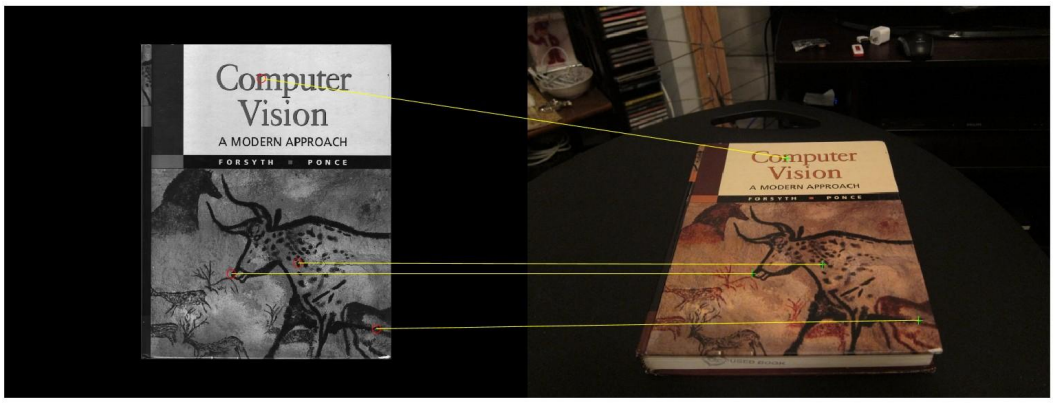
4. Homography Normalization

The function computeH_norm is written in the “computeH_norm.m” file, and after it’s done, the test for this part picks random 10 of points from the first image, and **shows the corresponding locations** in the second image after the homography transformation is in “q4.m”.

And the result is saved as “4_4.jpg” in /results and shown below.

5. RANSAC

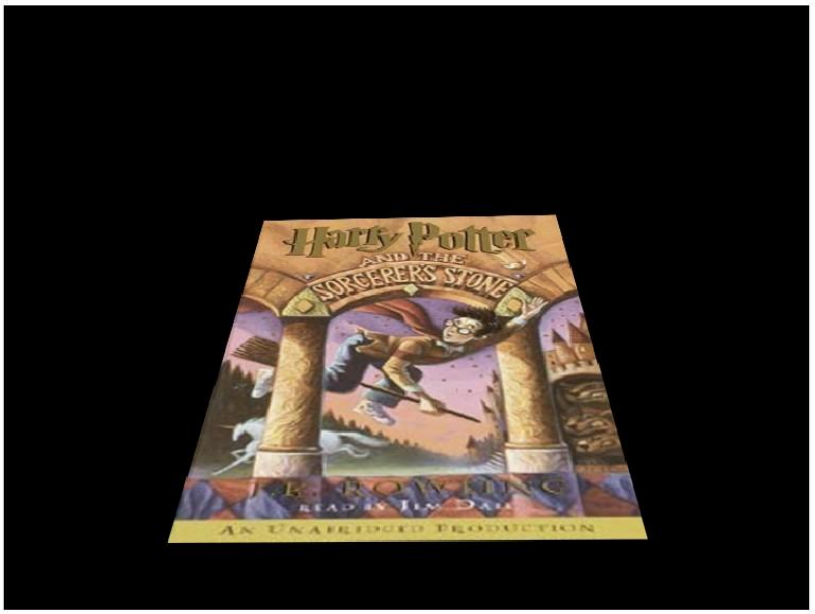
The function computeH_ransac is written in the “computeH_ransac.m” file, and the part of **visualizing the 4 point-pairs** that produced the most number of inliers is also written in “computeH_ransac.m” (the result is saved as “4_5(1).jpg” in /results and shown below).



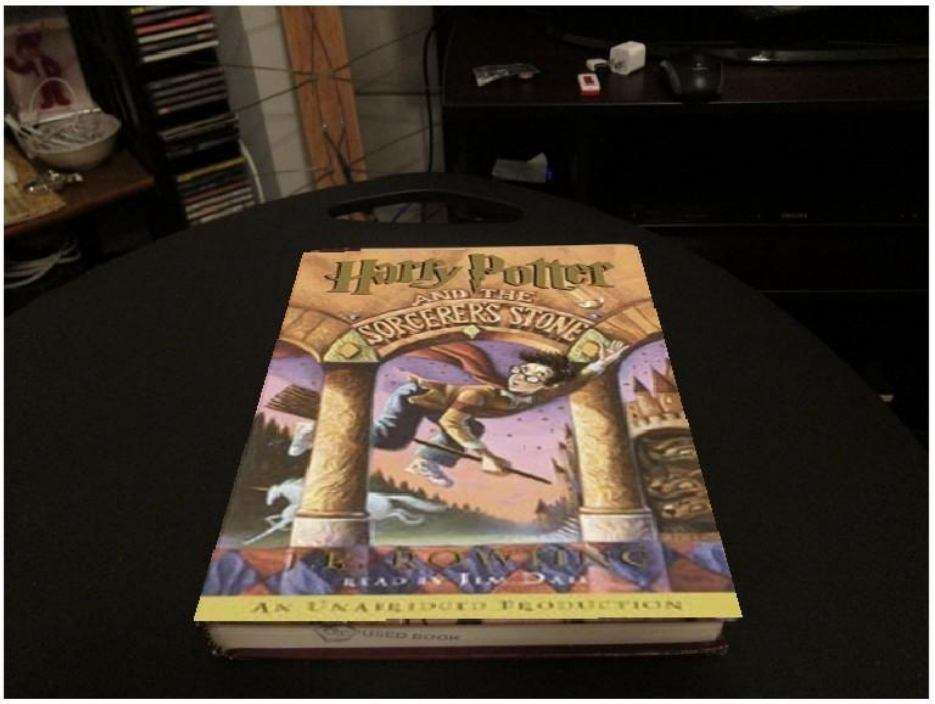
And the part that **visualizes the inlier matches** that were selected by the RANSAC algorithm is in “q5.m” with the result is saved as “4_5(2).jpg” in /results and shown below.

6. HarryPotterizing a Book

The result of **warpH(),** which is saved as “4_6(warpH).jpg” in /results and shown below**:**



The result of **compositeH()** which is complemented in script “compositeH.m”, which is saved as “4_6(compositeH).jpg” in /results and shown below**:**



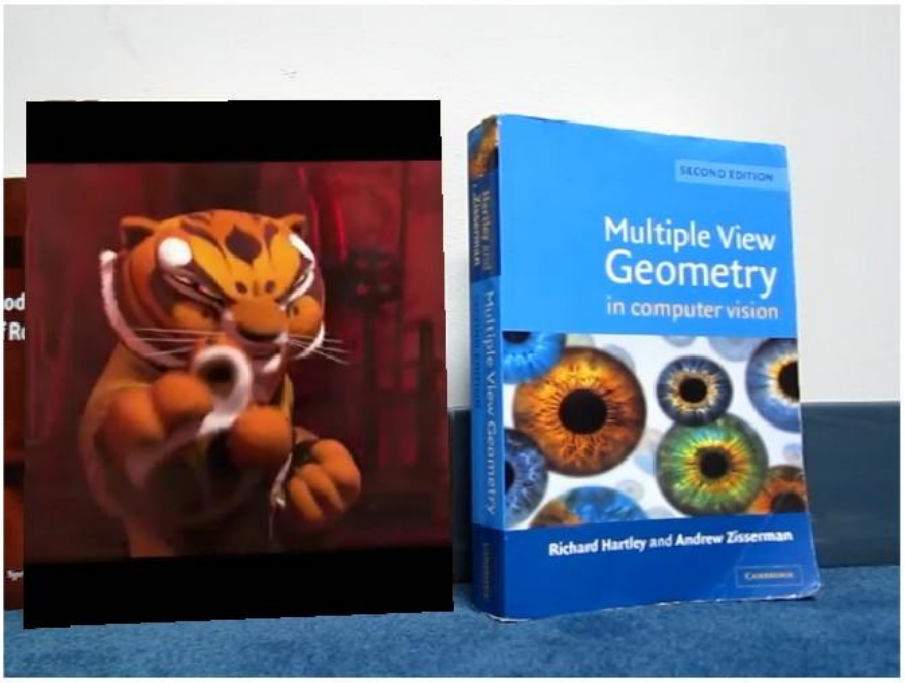
7. Creating your Augmented Reality application

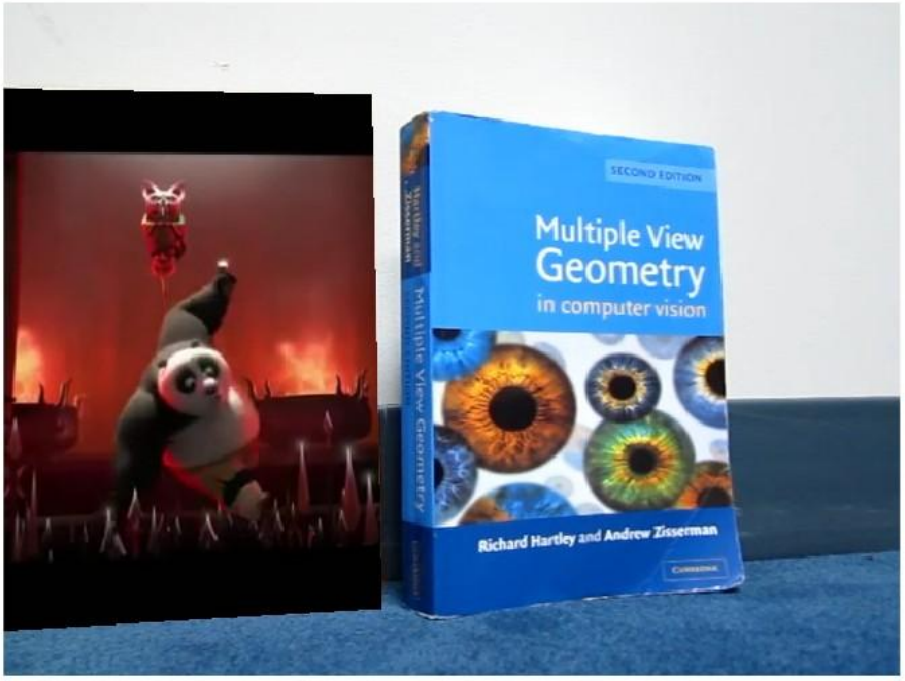
The first version of the result video is saved as **“ar1.avi”** in ../result/, and all of its frames are saved as .jpg file in ../result/ar1/, some of the frames are shown below:

But there are a few frames that are severely distorted, like:

Through a separate comparison of the frame, it is found that this is a corresponding error caused by too few feature points obtained by matchPics. In order to solve this problem, I optimized the original matchPics function (saved as matchPics1.m at the end) and improved it to use detectSURFFeatures and extractFeatures (saved as matchPics.m at the end) to get more and more accurate point pairs.

And the final version of the result video is saved as **“ar.avi”** in ../result/, and all of its frames are saved as .jpg file in ../result/ar/, some of the frames are shown below:


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


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