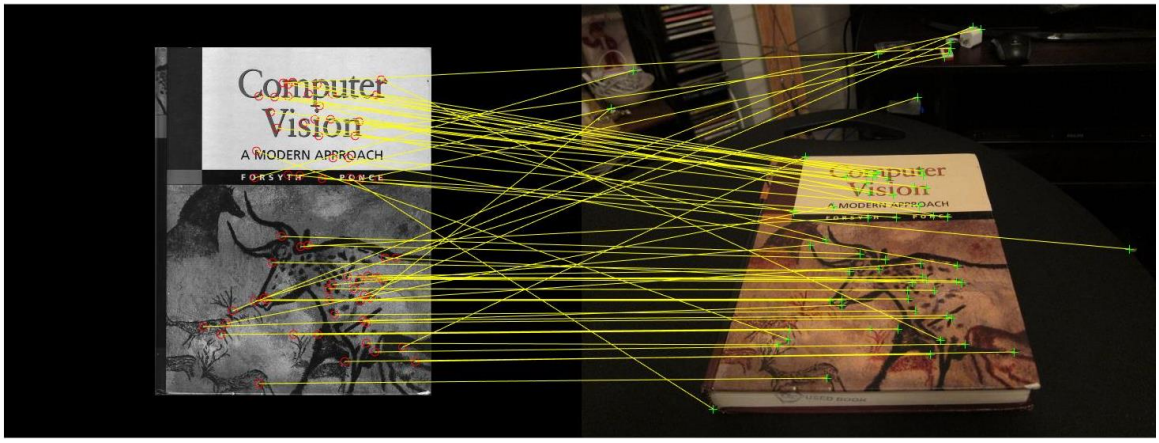
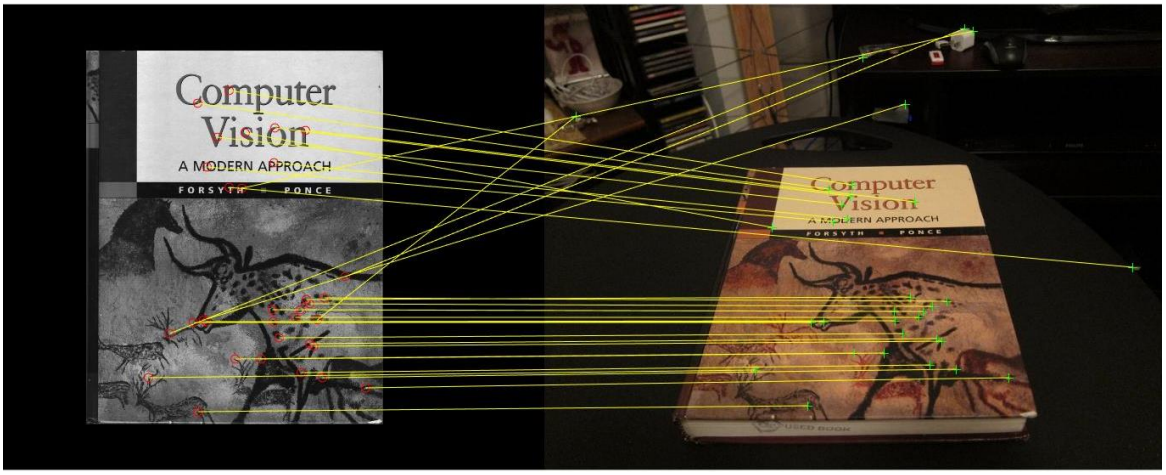
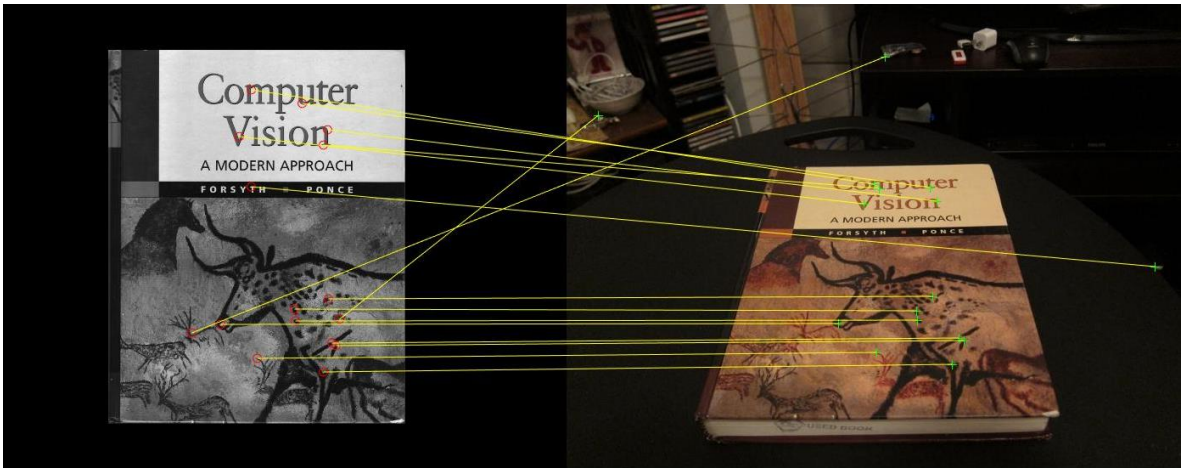
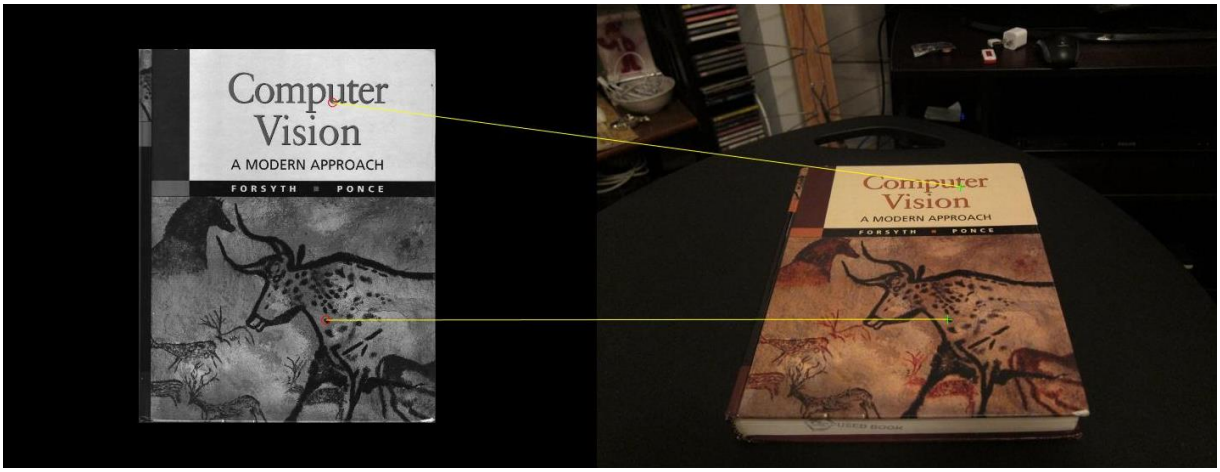
Part 2: Augmented Reality with Planar Homographies

Date: 2019-4-20

- Feature Detection, Description, and Matching

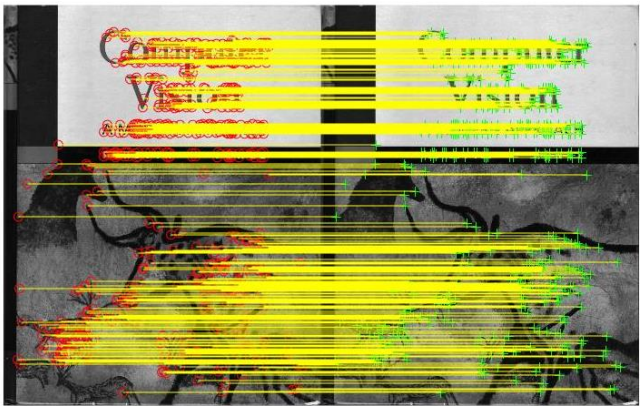
  In this section we need to transform the featureVectors that we get from the function "computeBrief" to binaryFeatures (or we will get an empty set of match features) , and carefully choose the parameter of the function " matchFeatures", in order to get the most correct results, the value of parameter 'Unique' should be true.

MatchThreshold =99 MaxRatio=0.9	
MatchThreshold =50 MaxRatio=0.8	
MatchThreshold =50 MaxRatio=0.7	

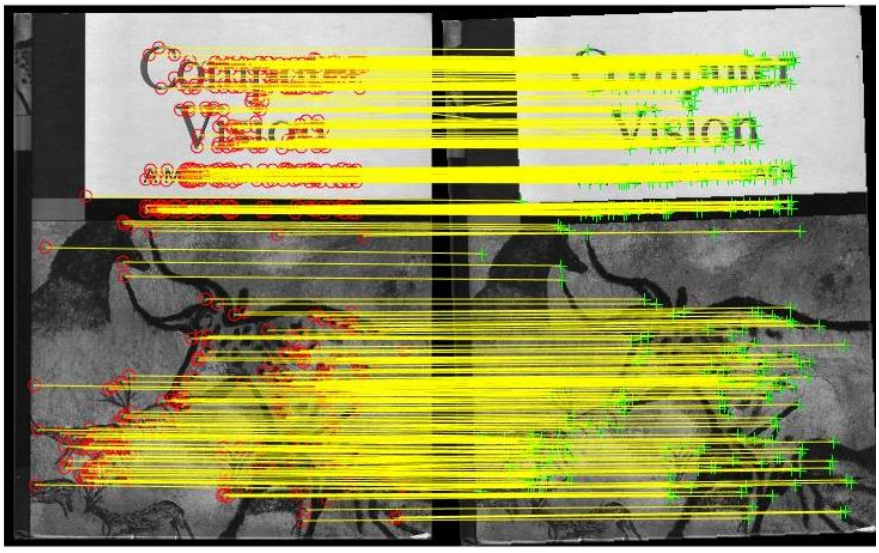
MatchThreshold =50 MaxRatio=0.6	

- BRIEF and Rotations
  - Fast:

0°

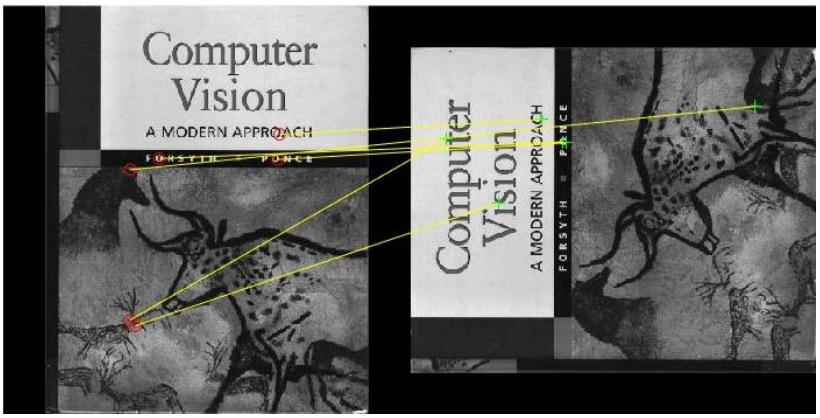


2°

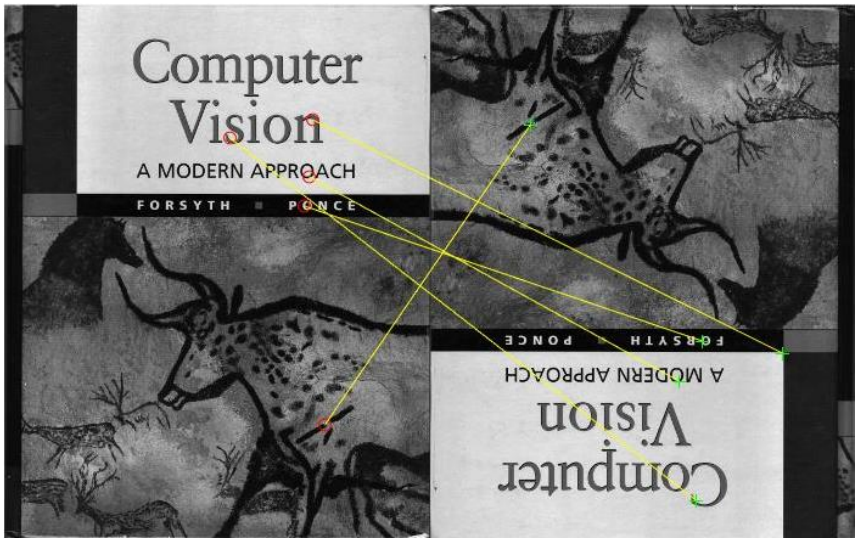


4°

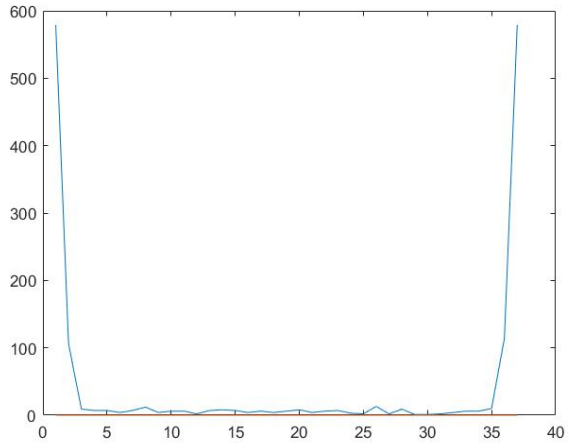
90°



180°



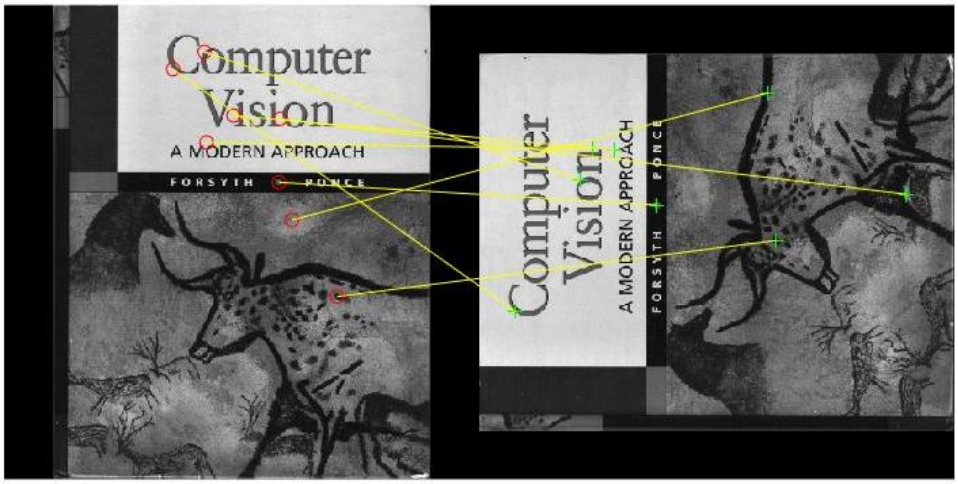
the count of matches



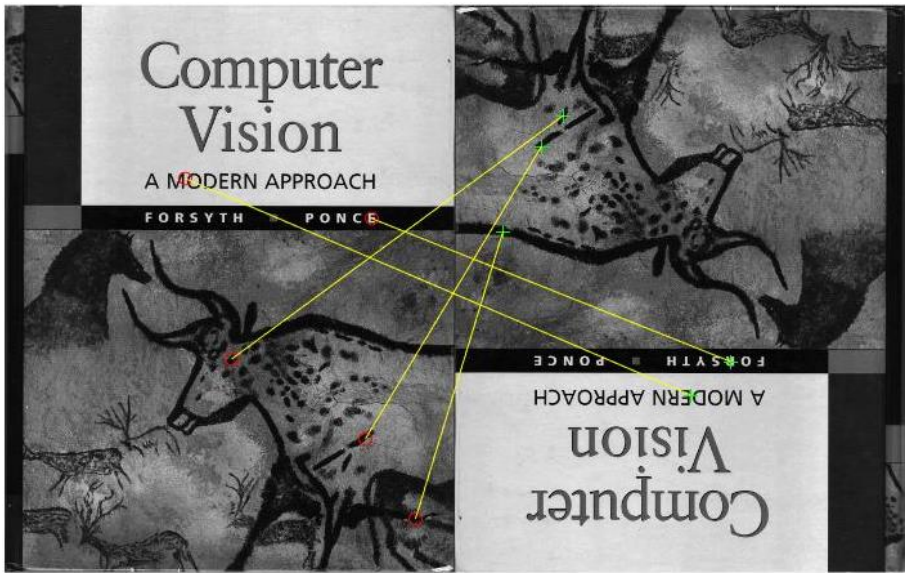
- SURFF

0°

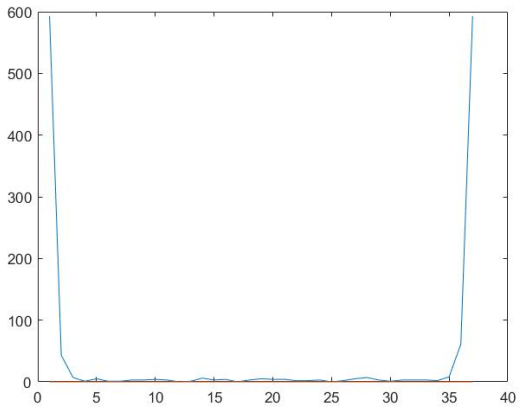
90°



180°

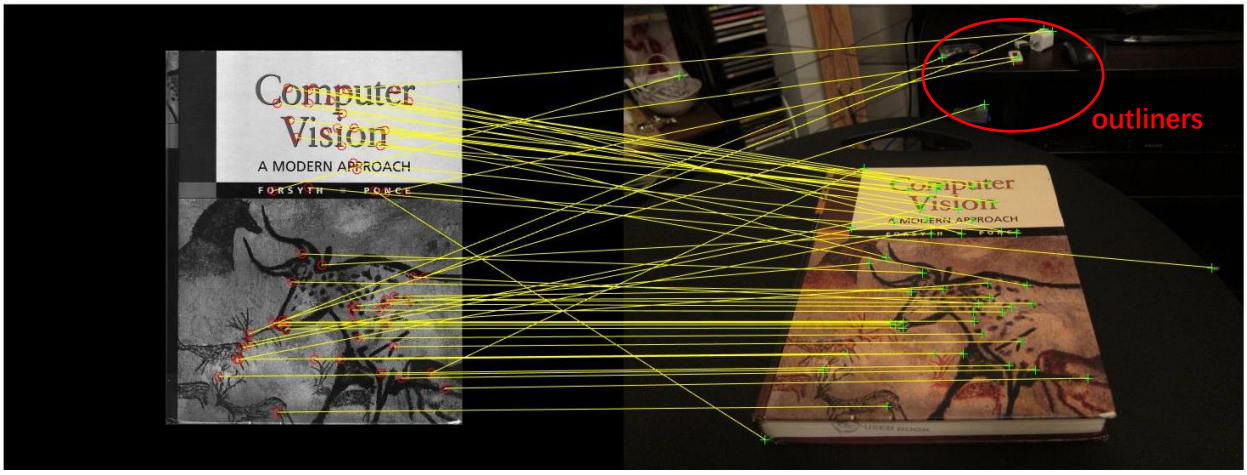


the count of matches

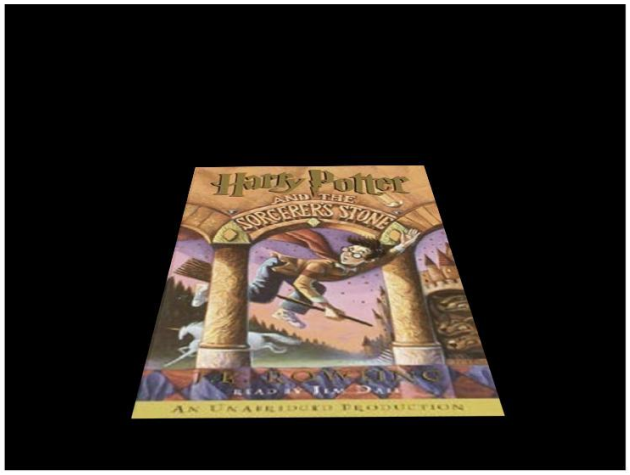


We can see that when the rotation angle is small, the BRIEF descriptor works well, but when the rotation angle is relatively big， the result is not idea, because when we choose the point pairs to get the descriptors, we set the origin as the key point, and the x axis as the horizontal direction, the y axis as the vertical direction, so after the image was rotated, the point pairs also changed, the descriptors changed too.

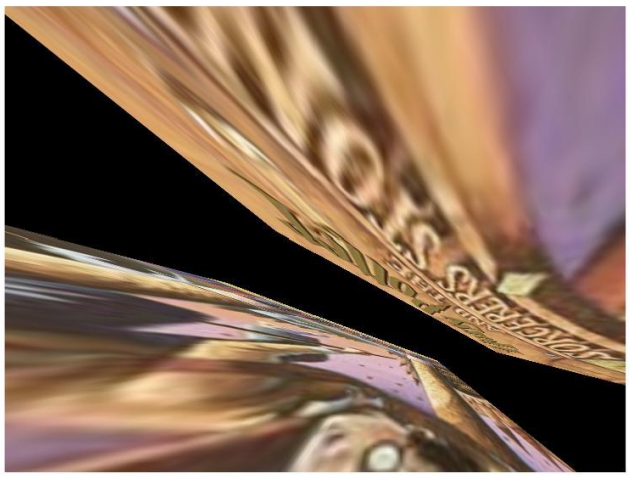
- Homography Computation and Normalization



In this section, I randomly choose 10 matchpoints to compute the Homography, the results are unstable , if all the points I choose are matched well, the result is ideal:



if the 10 pairs of points contain the outliners, I got a wrong result:

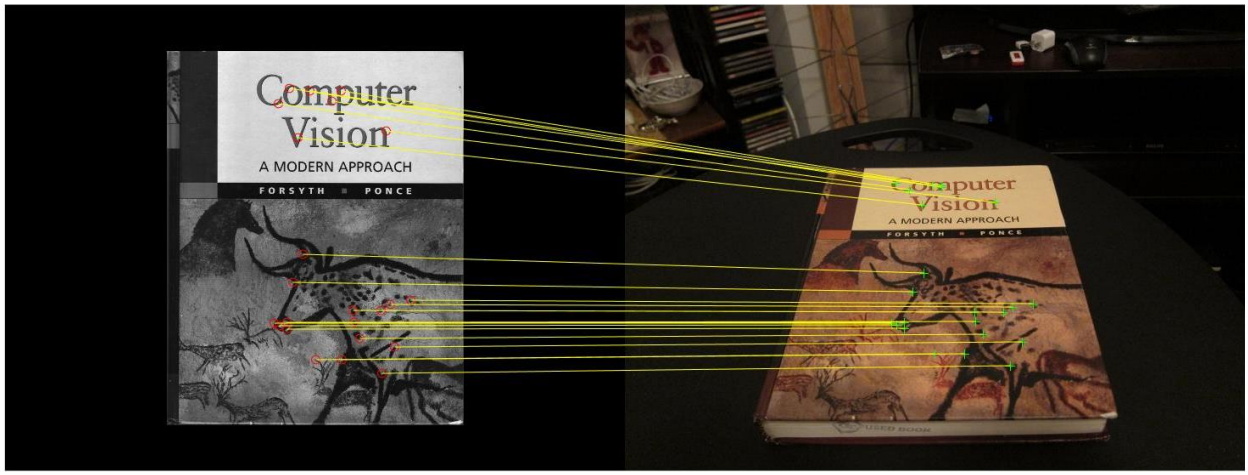


So a solution named RANSAC can solve such a problem.

- RANSAC

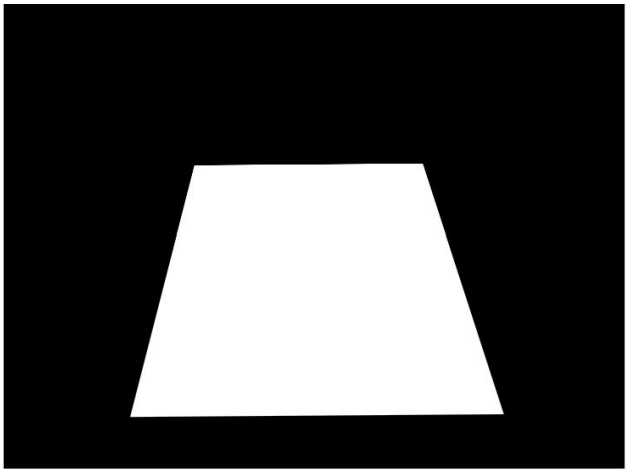
In this section, I set a threshold and sampled many times, compute the Homography, and use such a Homography to count the points that we regarded as “inliners” , that is, the error less than the threshold. Then recorded the Homography and the number of inliners, finally we can get the Homography that can get most inliners, then we use the inliners to compute the best Homography. And in order to get the best result, I set the parameter of matchFeatures as MatchThreshold=80, and MaxRatio=0.9 so that more feature can be abtained.

The inliners:

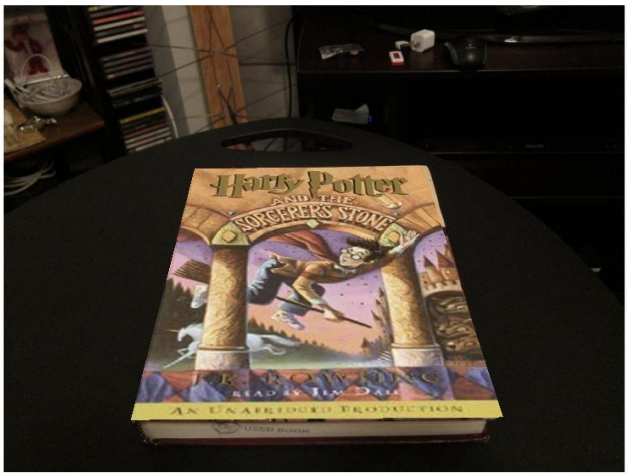


And last but not least, a mask is needed to get the composite image:

mask



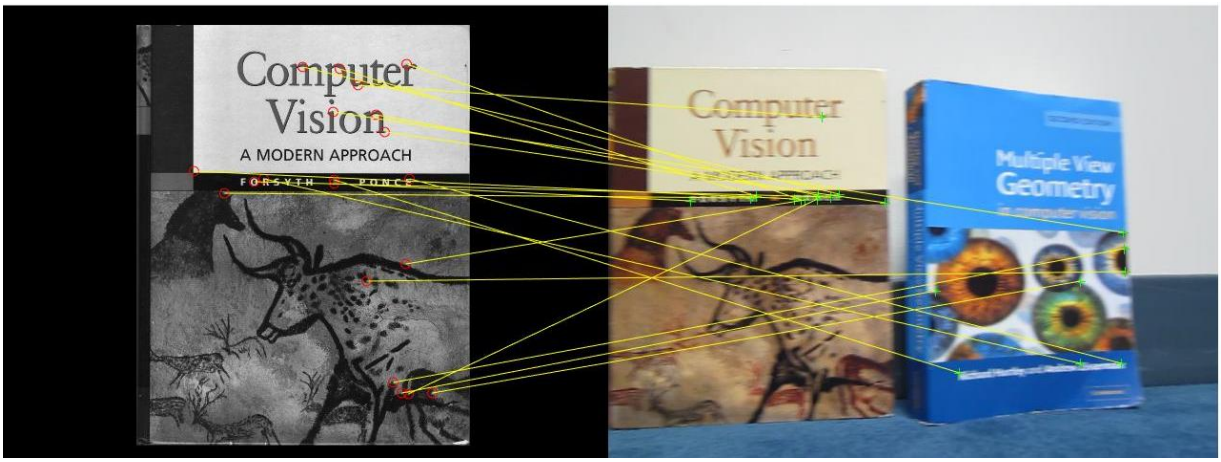
The composite image:



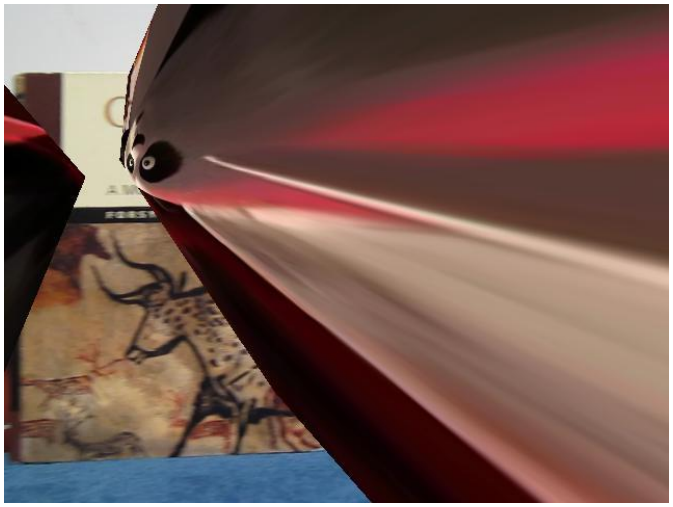
- Creating your Augmented Reality application

We can get the composite video by use the above solution to process every frames of the 2 video, there is also some problem:

As the following image shows, when the image is blurred （which is difficult to avoid）, the feature match can't work very well



And then the result is bad because maybe the outliners are more than inliners:



I solve this problem by simply replace those images with the ideal images that next to the frame. And the result.avi is in the file.