

## **Table of Contents**

<b>Approach of My Algorithm .....</b>	<b>2</b>
<b>Results.....</b>	<b>3</b>
For Image Pair: (image_1, image_3) .....	3
For Image Pair : (image_1, image_4) .....	5
For Image Pair: (image_1, image_5) .....	7
For Image Pair: (image_2, image_3) .....	9
For Image Pair: (image_2, image_4) .....	11
For Image Pair: (image_2, image_5) .....	13
<b>General analysis .....</b>	<b>14</b>

## Approach of My Algorithm

- 1.) fetch images pairwise say (sourceImage, destinationImage).
- 2.) convert the images to grayscale say (source, destination)
- 3.) Find Keypoints and descriptors of each of the image using OpenCV function sift.detectAndcompute.
  - a. This returns list of keyPoints(KP) and descriptor(desc) for each of the images mentioned above.
- 4.) Draw the key Points of source image
- 5.) Find number of key points by calculating length of source KP.
- 6.) Then invoke instance of BFMatcher.
- 7.) Invoke match API of Brute Force Matcher(BFMatcher). This returns List of <DMATCH>
  - a. Each DMATCH is a tuple of <distance, imgIdx, queryIdx, trainIdx>
- 8.) Sort matches as per distances.
- 9.) Plot top 20 matches using drawMatches function.
- 10.) Find Homography function
  - i. For this, use the function findHomography. This returns H matrix and status. That is True/False value if corresponding keypoint is consistent or not.
- 11.) Then given the status received in step above, figure top 10 distances that are consistent.
- 12.) Display top 10 distances or matches on Image and find the number of consistent matches.

## Results

For Image Pair: (image\_1, image\_3)

SIFT KeyPoints	3253
KeyPoints Image	
Number of Matches Found	3253

Top  
match  
images

20



Top  
matches  
after  
Homography

10



Number  
Consistent of  
Matches

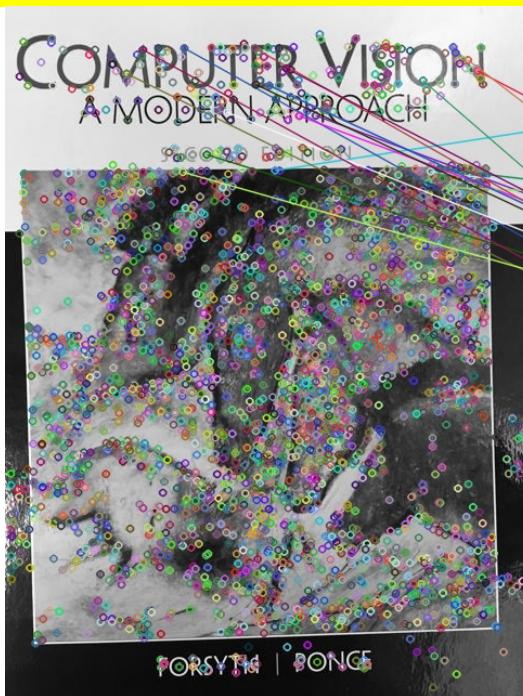
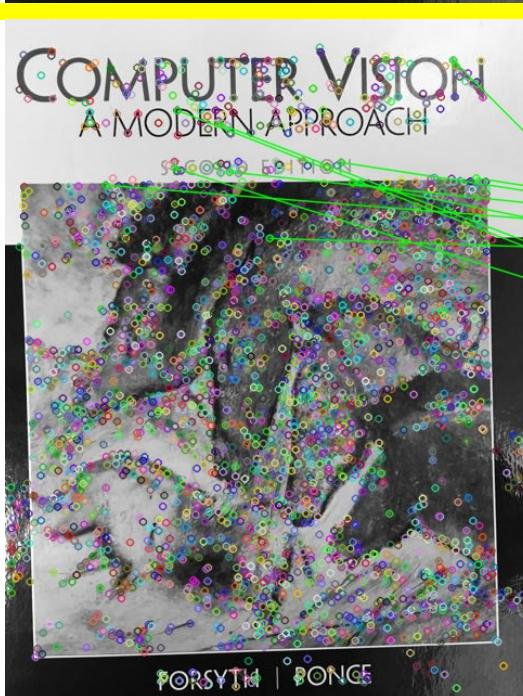
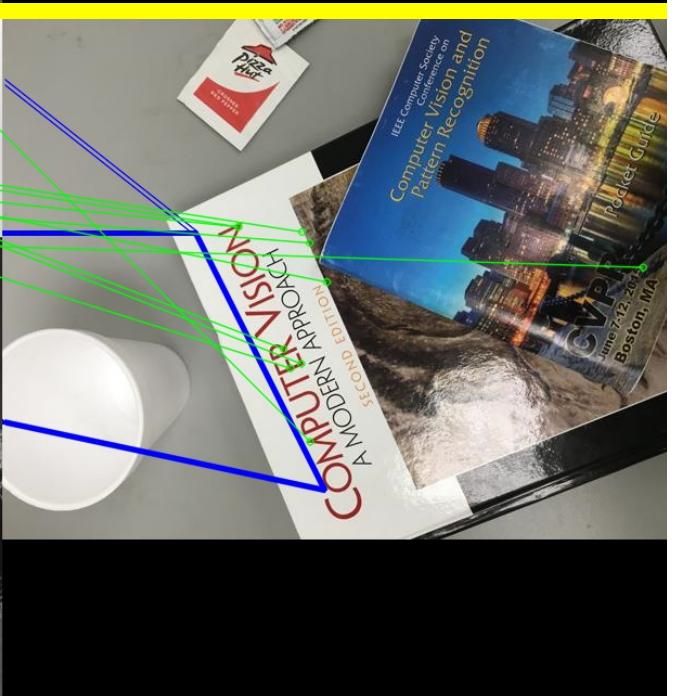
321

Homography  
Matrix

```
[[ 5.40424690e-01   6.92860070e-02   2.42151991e+01]
 [ -9.98585075e-02   5.32622716e-01   9.91874053e+01]
 [ -4.35700576e-05  -6.07857425e-05   1.00000000e+00]]
```

For Image Pair : (image\_1, image\_4)

SIFT KeyPoints	3253
KeyPoints Image	
Number of Matches Found	3253

Top match images	20		
Top matches after Homography	10		
Number Consistent of Matches	42		
Homography Matrix		[[ -2.23385013e-01 8.91160096e-02 2.97875960e+02 ] [ -4.66502178e-01 -6.00275198e-01 4.34869535e+02 ] [ 1.49700751e-04 -1.66603541e-03 1.00000000e+00 ]]	

For Image Pair: (image\_1, image\_5)

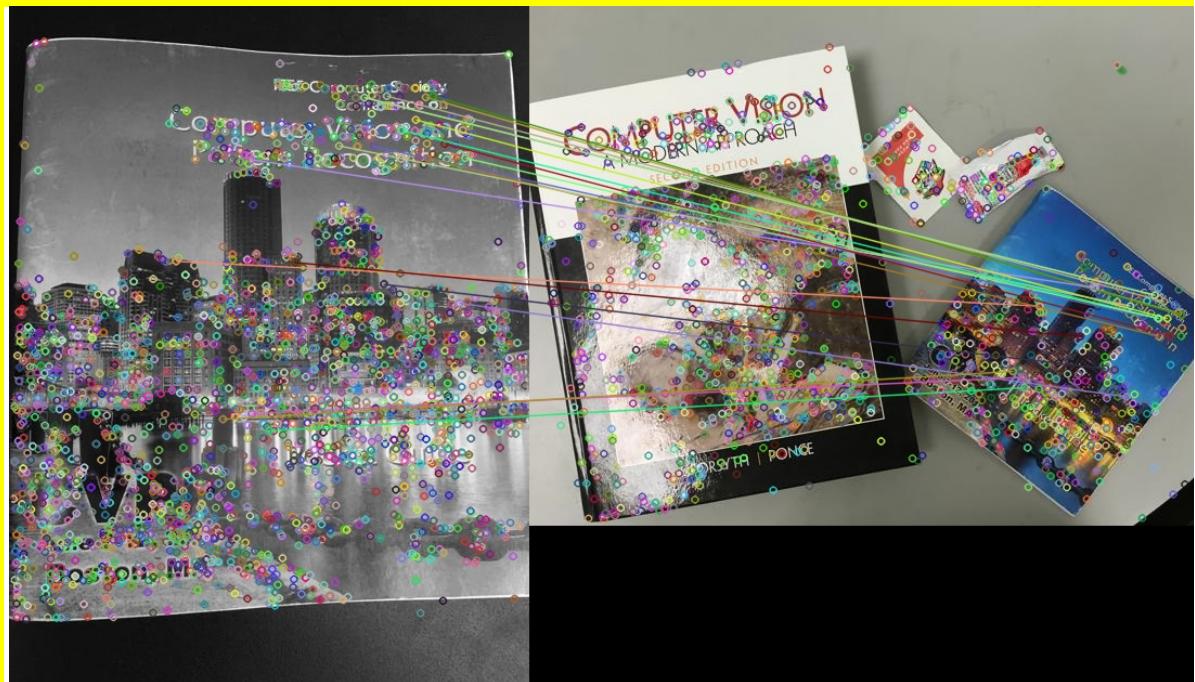
SIFT KeyPoints	3253
KeyPoints Image	
Number of Matches Found	3253

Top match images	20	
Top matches after Homography	10	
Number Consistent of Matches	602	
Homography Matrix		$\begin{bmatrix} [-1.17694313e-01 & 4.06723466e-01 & 2.74567690e+02] \\ [-4.76826475e-01 & -1.33253477e-01 & 3.45310634e+02] \\ [-5.15255905e-05 & -1.61819833e-04 & 1.00000000e+00] \end{bmatrix}$

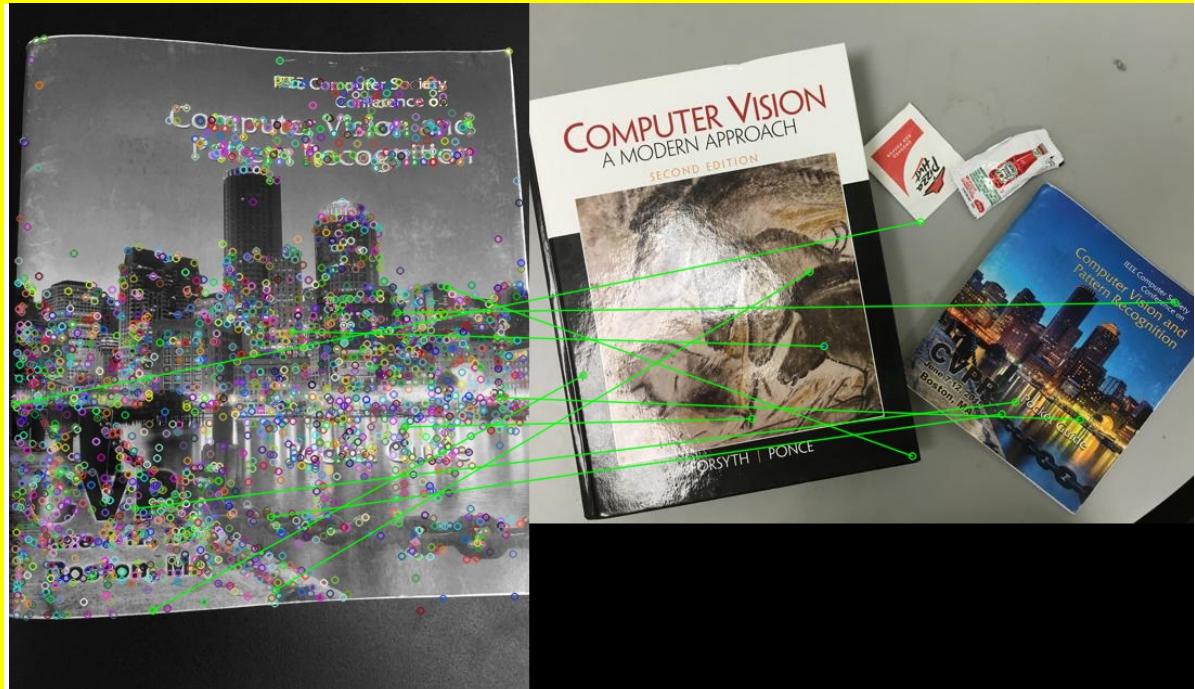
For Image Pair: (image\_2, image\_3)

SIFT KeyPoints	2815
KeyPoints Image	
Number of Matches Found	2815

Top match images 20



Top matches after Homography 10



Number Consistent of Matches

70

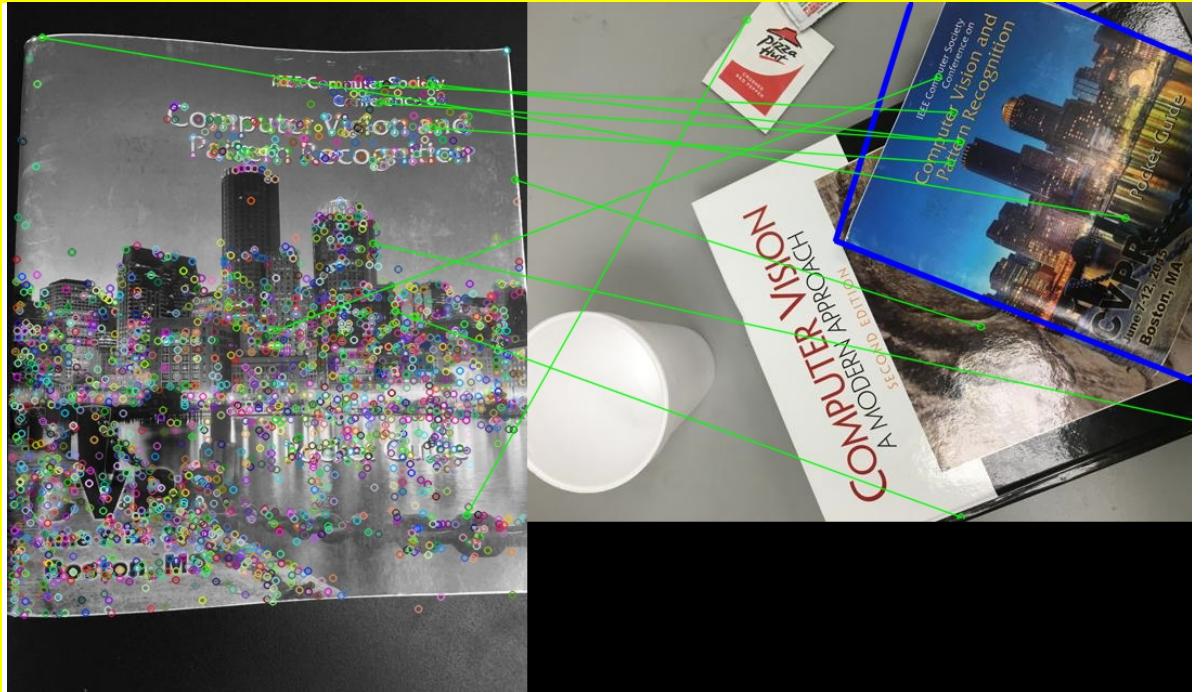
Homography Matrix

```
[[ -4.00647293e-01  -1.44709044e+00   2.71652827e+02]
 [ -4.66155999e-01  -1.68344544e+00   3.16040582e+02]
 [ -1.47501328e-03  -5.32638525e-03   1.00000000e+00]]
```

For Image Pair: (image\_2, image\_4)

SIFT KeyPoints	2815	
KeyPoints Image		
Number of Matches Found	2815	
Top match images	20	

Top 10  
matches  
after  
Homography



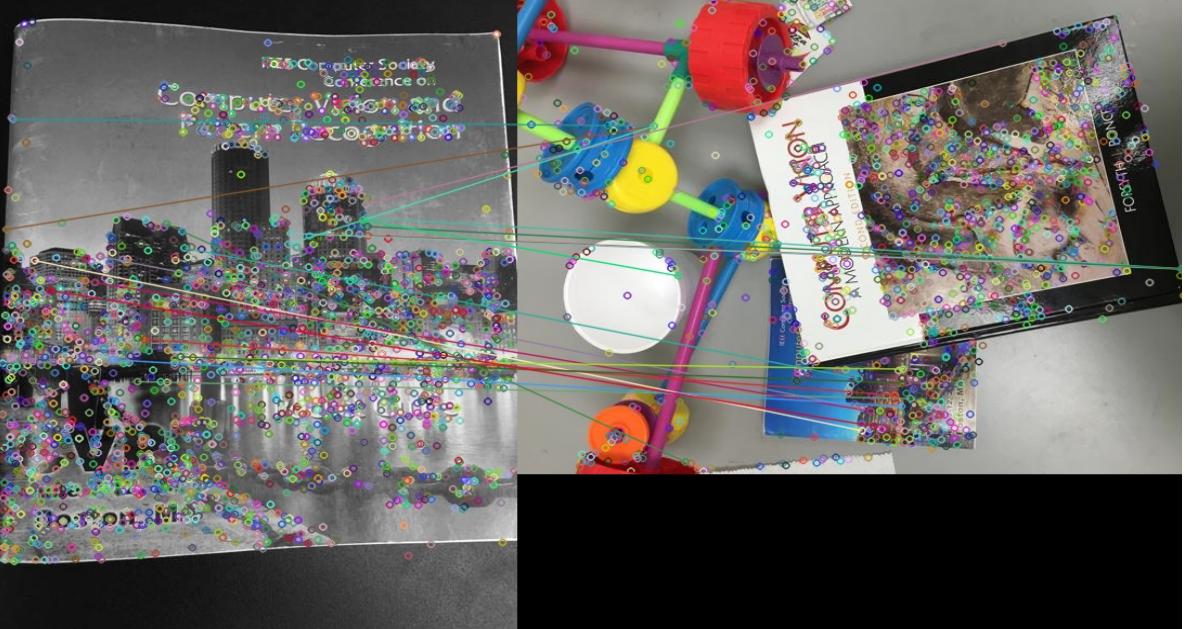
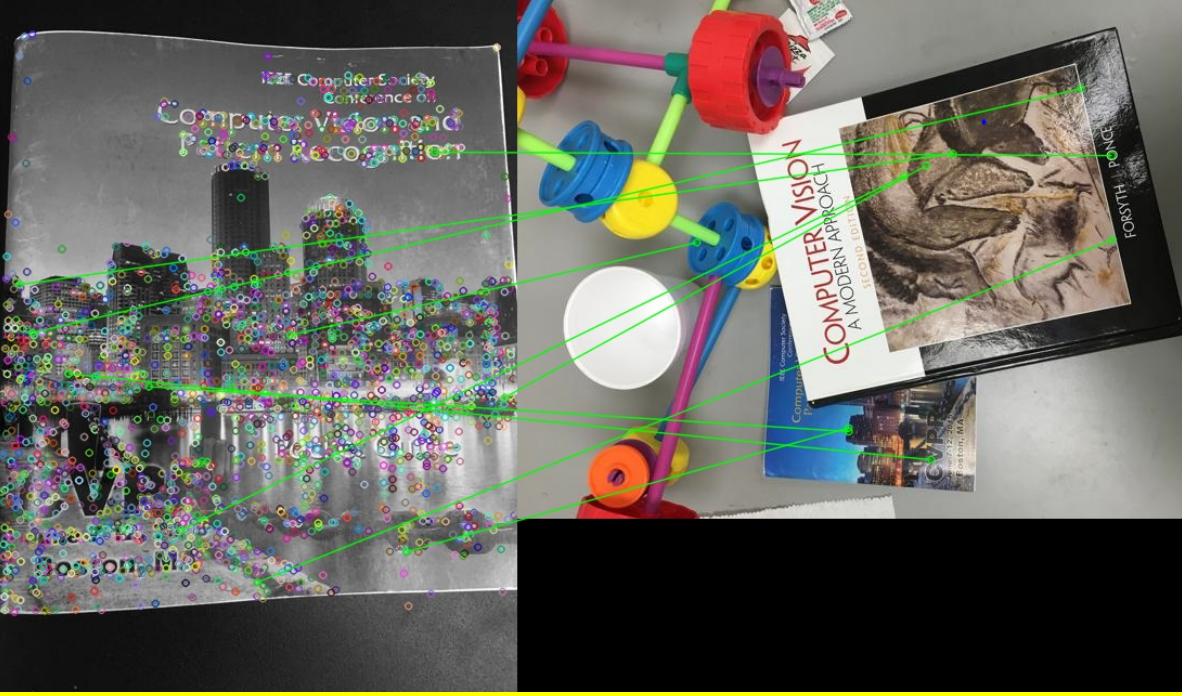
Number  
Consistent  
of Matches

128

Homography  
Matrix

```
[[ 7.97506707e-02  5.62813066e-01  2.84195832e+02]
 [ -5.56362693e-01  2.27663914e-01  2.20949575e+02]
 [ -2.68653312e-04  6.74283357e-05  1.00000000e+00]]
```

### For Image Pair: (image\_2, image\_5)

SIFT KeyPoints	2815
KeyPoints Image	
Number of Matches Found	2815
Top match images	20
	
Top matches after Homography	10
	
Number Consistent	43

of Matches	
Homography Matrix	<pre>[[ -1.31447015e-01 -9.45832596e-01  4.30775390e+02]  [ -3.45302000e-02 -2.48348513e-01  1.13112481e+02]  [ -3.05138548e-04 -2.19565536e-03  1.00000000e+00]]</pre>

## General analysis

The Algorithm in general performs very well and is rotation invariant. However, the algorithm doesn't seem to be well invariant to illumination. For ex: white portion on source cover page of the book tend to be confused with white portion of pizza hut sachets.

If the object is surrounded by objects that have intensity different from that of target object then the detection gets easier. In this case, the algorithm performs better as in image\_5.