CMPT 762 Assignment 4

301474102 Sihui Wang

1. Feature Detection, Description, and Matching

1.1 Parameters for matching of BRIEF features:

MatchThreshold: 10.0

MaxRatio: 0.68

1.2 Visualization:

The visualizer is `q2_1_4.m`.

Here is the visualization of `cv_cover.jpg` and `cv_desk.png` under `data` folder:

2. BRIEF and Rotations

Note: `briefRotTest.m` is responsible for visualization of both BRIEF and SURF's performance in Section 2.1 and 2.2.

2.1 BRIEF

2.1.1 Visualization

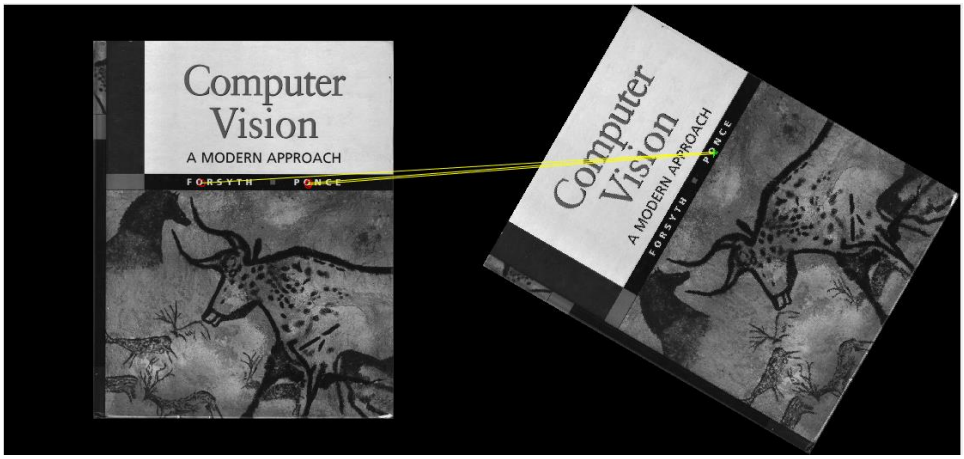
Here is the visualization of the BRIEF feature matching results for different rotation angles ($\theta = 60^{\circ}, 120^{\circ}, 180^{\circ}$).

Parameters Used for Matching of BRIEF features:

MatchThreshold: 10.0

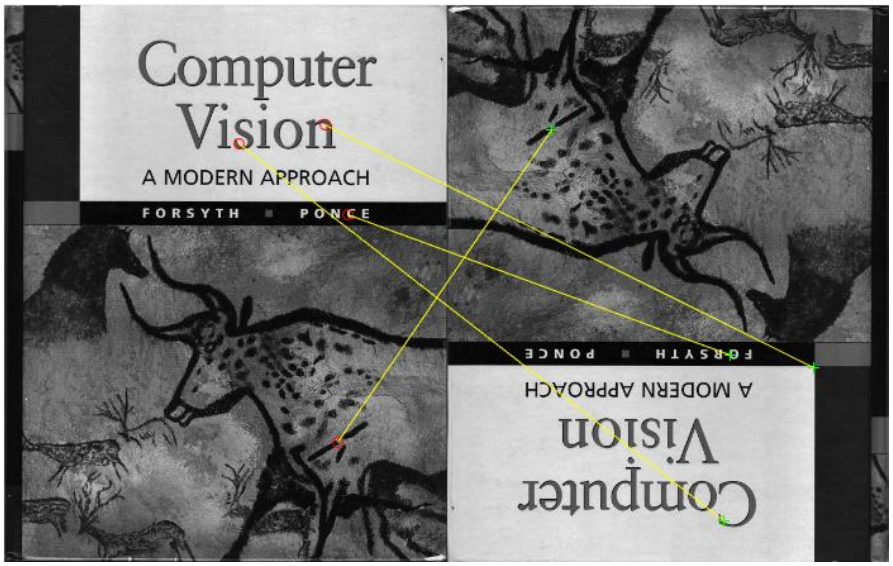
MaxRatio: 0.68

$\theta = 60^{\circ}$

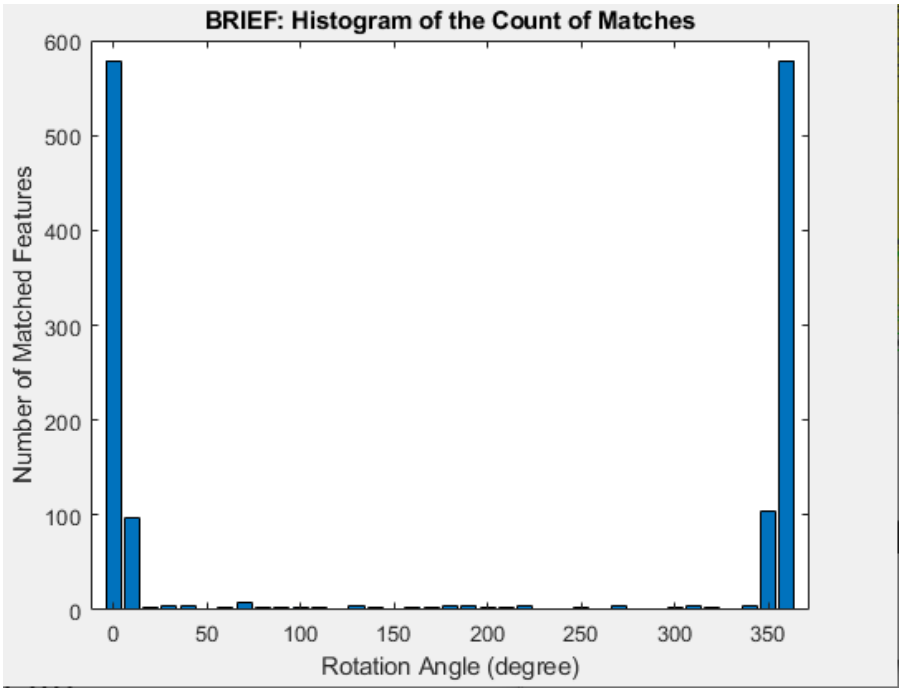


$\theta = 120^{\circ}$

$\theta = 180^{\circ}$



2.1.2 Histogram of the Count of Matches



2.1.3 Explanation

BRIEF descriptor relies on a small number of intensity difference tests to represent an image patch around the key point as a binary string. By design BRIEF is not invariant, or robust, to large in-plane rotations.

From the above visualization we can also confirm that BRIEF is not robust to rotations. When $\theta = 180^{\circ}$, BRIEF will consider the tail of 'S' and the head of 'C' to be similar, and it will consider the up-right corner of 'n' and up-right corner of the rectangle to be similar. This is because the intensity of the surrounding areas becomes similar after rotations. Since BRIEF is not robust to rotations, after rotations it will fail to recognize the matching points or make wrong matchings. This is why for BRIEF, the number of matched features declines sharply after rotations.

The tail of 'S' and the head of 'C' become similar after rotation

The corner of 'n' and the corner of the rectangle become similar after rotation

2.2 SURF

2.2.1 Visualization

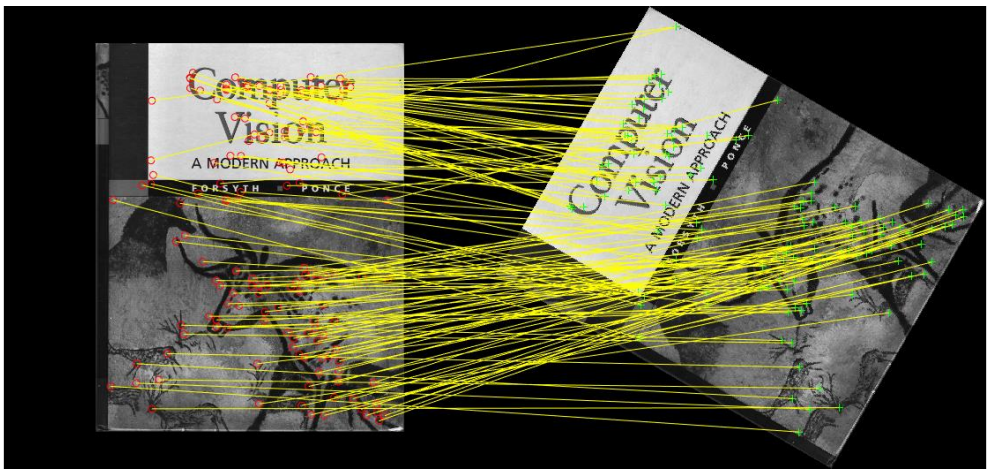
Here is the visualization of the SURF feature matching results for different rotation angles ($\theta = 60^{\circ}, 120^{\circ}, 180^{\circ}$).

Parameters Used for Matching of SURF features:

MatchThreshold: 1.0

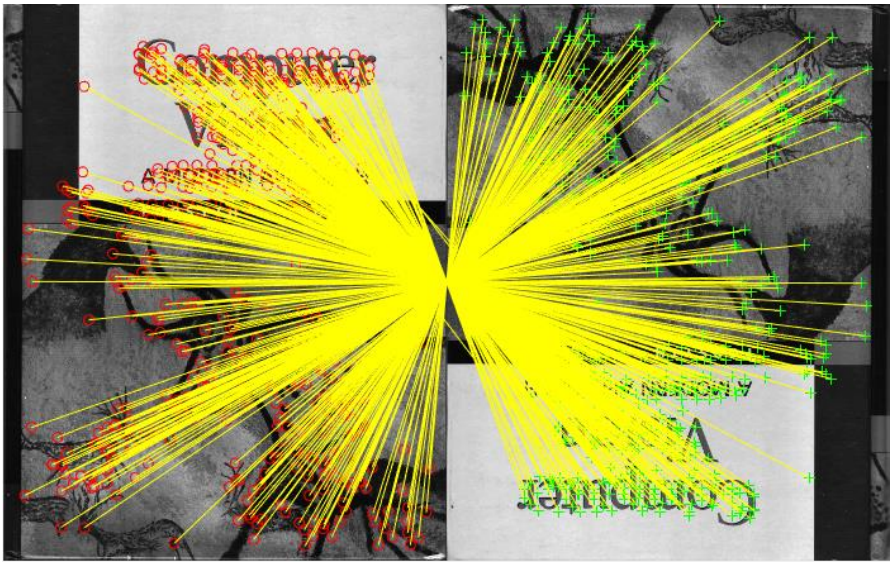
MaxRatio: 0.68

$\theta = 60^{\circ}$

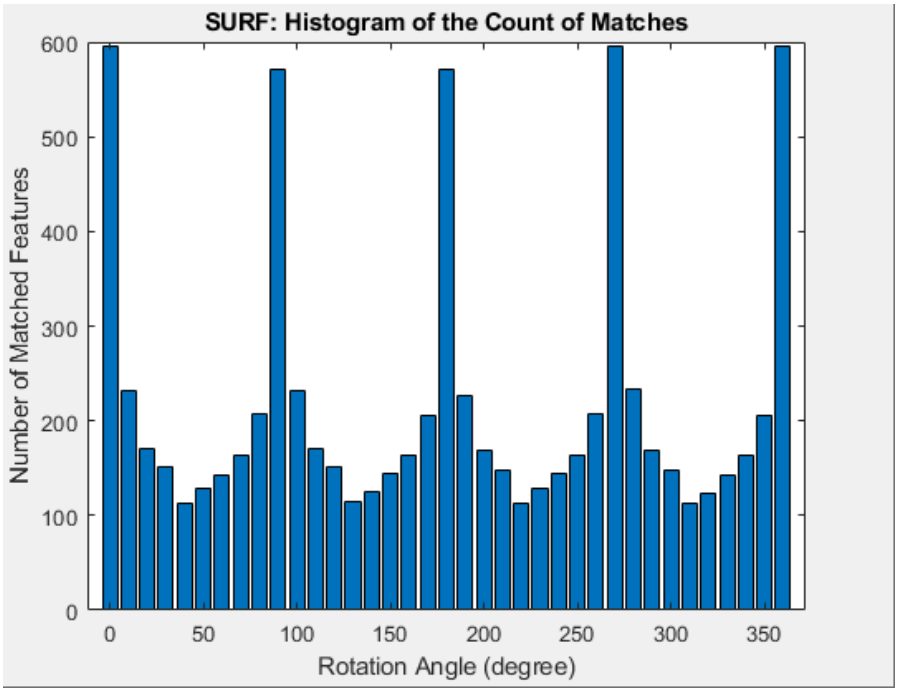


$\theta = 120^{\circ}$

$\theta = 180^{\circ}$



2.2.2 Histogram of the Count of Matches



2.2.3 Explanation

Compared with BRIEF, SURF's performance is much better in dealing with different orientations. When $\theta = 0^{\circ}$ with no rotation, BRIEF and SURF can recognize about the same number of matching pairs. However, BRIEF's performance deteriorates with even slight rotations. While SURF's performance also declines with rotations, it is much more robust and can still recognize a fair number of matching pairs correctly with rotations. Notably, SURF suffers virtually no performance loss when $\theta = 90^{\circ}, 180^{\circ}, 270^{\circ}$.

In SURF's implementation, the orientation of the point of interest needs to be found first, in order to achieve rotational invariance. Such mechanisms lead to SURF's greater robustness to rotations. Since SURF can recognize the change of orientations during the matching process, it is able to find much more pairs of matching features correctly when the images are rotated.

SURF computes horizontal and vertical Haar wavelet responses with respect to the orientation of the key points. Then, the wavelet responses dx and dy are summed up as $(\Sigma dx, \Sigma dy, \Sigma|dx|, \Sigma|dy|)$, which results in a 64-dimensional descriptor vector. When the image is rotated by $90^{\circ}, 180^{\circ}, 270^{\circ}$, the original image's Σdx can match exactly with the rotated image's Σdy, $-\Sigma dx$, or $-\Sigma dy$, and the original image's Σdy can perfectly match with the rotated image's Σdx, $-\Sigma dy$, or $-\Sigma dx$, which is the reason why SURF suffers no performance loss when the image is rotated by $90^{\circ}, 180^{\circ},$ or 270°.

3. Homography Computation

Note: `testH.m` is responsible for visualization of Section 3, 4, 5, that is, homography computation with 3 different methods.

3.1 Parameters:

Features Used: BRIEF

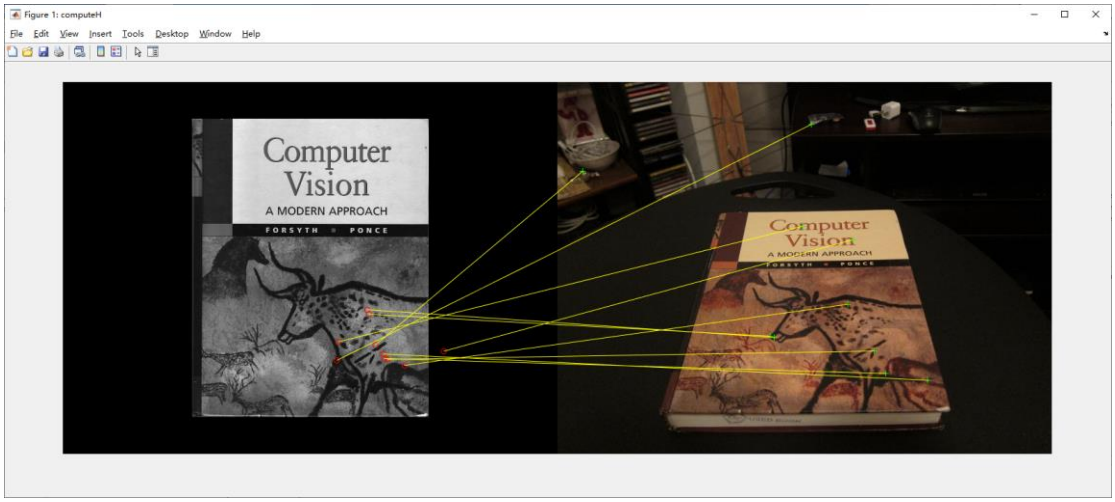
Parameters Used for Matching of BRIEF features:

MatchThreshold: 10.0

MaxRatio: 0.70 (*with former configuration* `MaxRatio = 0.68` *there will be no outliers for feature matching, and we won't be able to see the difference among 3 homography computation methods*)

Method for Homography Computation: Direct Linear Transform (DLT) with SVD

3.2 Visualization



4. Homography Normalization

4.1 Parameters:

Features Used: BRIEF

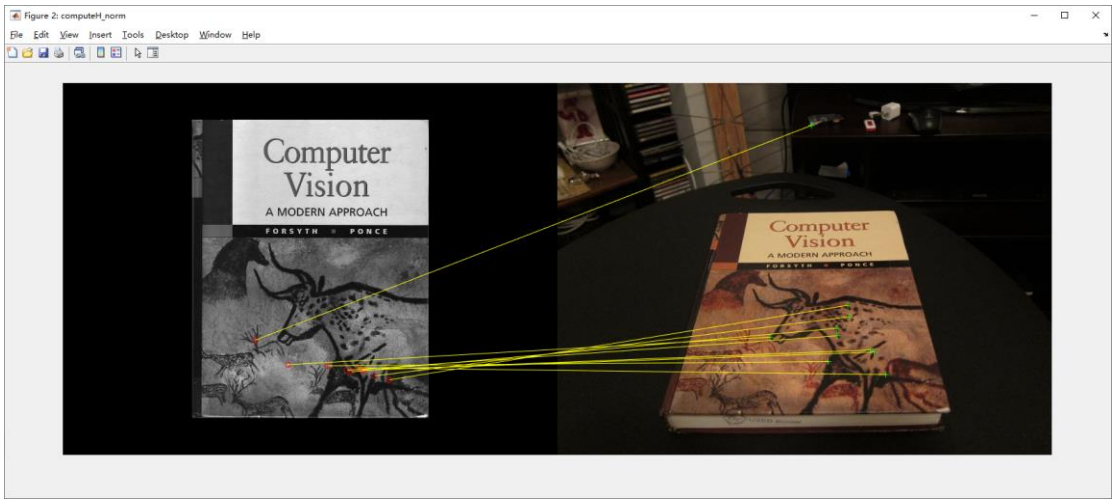
Parameters Used for Matching of BRIEF features:

MatchThreshold: 10.0

MaxRatio: 0.70

Method for Homography Computation: Data normalization is applied before Direct Linear Transform (DLT) with SVD

4.2 Visualization:



5. RANSAC

5.1 Parameters:

Features Used: BRIEF

Parameters Used for Matching of BRIEF features:

MatchThreshold: 10.0

MaxRatio: 0.70

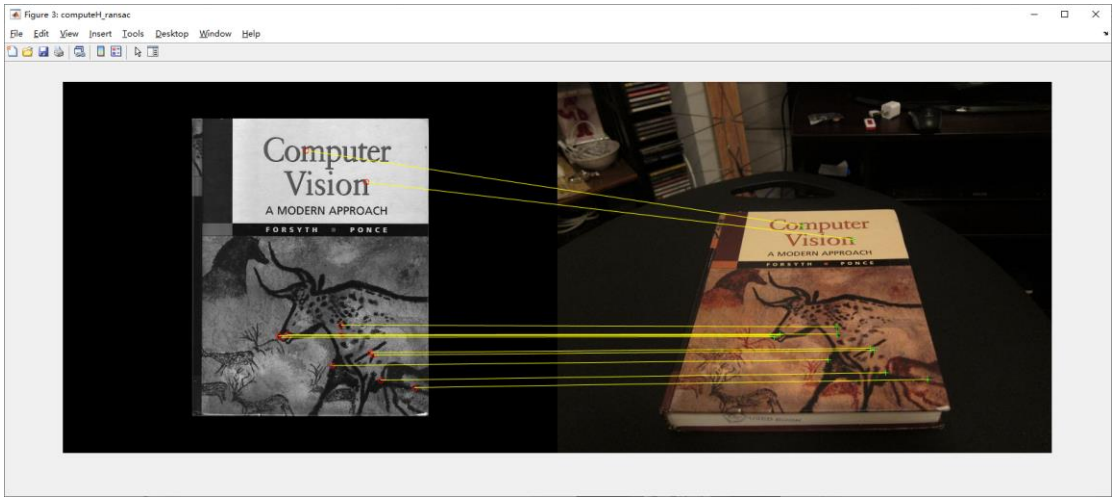
Method for Homography Computation: DLT with SVD is used for each iteration of RANSAC; data normalization is applied before DLT

Parameters for RANSAC:

Number of Iterations: 3000

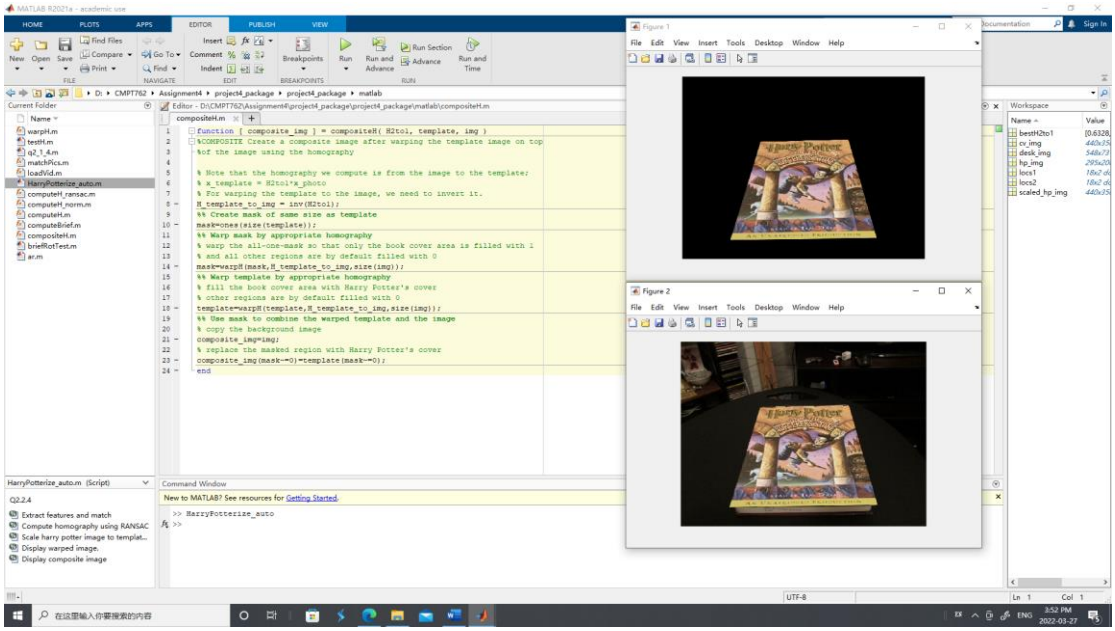
Threshold of error for inlier-outlier classification: 2.0

5.2 Visualization:



6. HarryPotterizing a Book

Below is the screenshot of the results. On the left is the implementation of `CompositeH.m`, and on the right is the visualization of the results. Figure 1 (top-right) shows warped book cover with no filling of the background, and Figure 2 (bottom-right) shows warped book cover with filling of desired background.



Note: In `HarryPotterize_auto.m`, the last line of code:

```
imshow(compositeH(inv(bestH2to1), scaled_hp_img, desk_img));
```

is not compatible with the design of `CompositeH.m`. `inv(bestH2to1)` should be replaced with `bestH2to1` to get the above outcome.

7. Creating an Augmented Reality Application

Note:

The *uncompressed* AVI is generated by the following matlab code:

```
result=VideoWriter('../result/ar.avi','Uncompressed AVI');
```

As the uncompressed AVI is large, I uploaded it on Google drive and here is the link: https://drive.google.com/file/d/1ZDzbeIbcY7soJsjw3cgtkudpkqxNmot5/view?usp=sharing

The *compressed* AVI is generated by the following matlab code:

```
result=VideoWriter('../result/ar.avi');
```

This result is saved under the `result` folder. Please use PotPlayer (Global Potplayer (daum.net)) to open this file (you will have problems opening it with Windows Media Player or 'Films & TV').

Parameters:

Features Used: BRIEF

Parameters Used for Matching of BRIEF features:

MatchThreshold: 10.0

MaxRatio: 0.70

Method for Homography Computation: DLT with SVD is used for each iteration of RANSAC; data normalization is applied before DLT

Parameters for RANSAC:

Number of Iterations: 3000

Threshold of error for inlier-outlier classification: 2.0