CV2015Spring—Assignment #2

Due: Thursday, Apr 30 10:00 AM

Dai Jialun

Apr 30, 2015

My assignment is programmed by C++ based OpenCV 3.0.0 and the IDE is Code::Blocks.

1. The Arrange of This Assignment

1.1 Step 1: Input image

The object image and scene image are shown in Figure 1. Firstly, I input the two images and converse the color space from RGB to GRAY.

(a) (b)

Figure 1: Input Image

1.2 Step 2: Features2D detection

In this step, I choose SURF algorithm to detect keypoints in object image and scene image. And I adopt the "SURF" class in the "xfeatures2d" module. The initialization of "hessianThreshold" parameter

will affect the result in the end. Therefore, hessianThreshold is set between 100 and 500 to get a better effect. The detection result of keypoints in object image and scene image are shown in Figure 2.

(a) (b)

Figure 2: Keypoints Detection

1.3 Step 3: Calculate descriptors

In this step, I calculate two keypoints in object image and scene image and obtain two different descriptors of keypoints in two images perspectively. These descriptors are also stored in two different matrix perspectively.

1.4 Step 4: Match descriptors

In this step, I need to match keypoints between object image and scene image. FlannBasedMatcher and BFMatcher can be used to implement the task of keypoints matching. And FlannBasedMatcher is better in situations that the accuracy is not demanded exactly and speed is desired. Therefore, I choose it to match keypoints.

For the simplification of calculation and improve accuracy, I select some better keypoints to match. If two matching keypoints satisfies the condition that the distance of these two keypoints is less than $2.5*\min(distance)$, I may regard these twp keyponits as good matching keypoints and use these keypoints for the following process. The keypoints matching is shown in Figure 3.

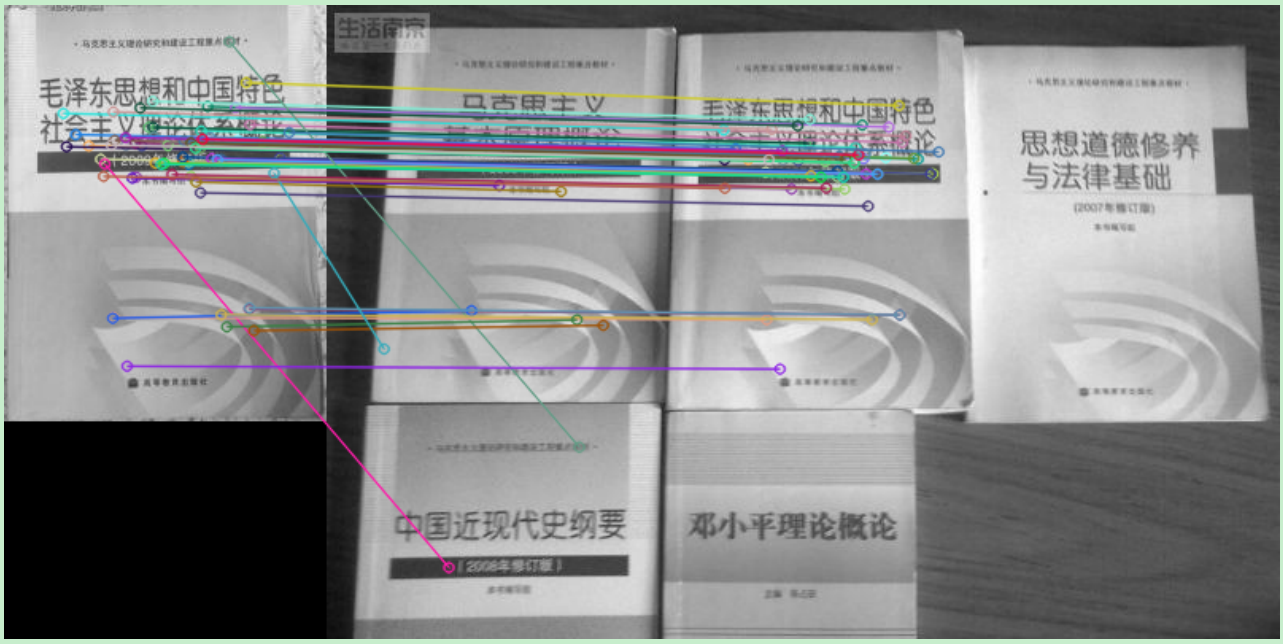

Figure 3: Keypoints Matching

1.5 Step 5: Find homography transformation

I need to find a matrix of homography transformation between two different point sets. I use homography transformation to obtain the matrix of homography transformation. The "findHomography" function can compute the connection of keypoint sets of oobject image and scene image and return the matrix of homography transformation correspoding to the keypoints sets quickly.

1.6 Step 6: Perspective transform

From Step 5, I have obtained the matrix of homography transformation. So I just use the "perspectiveTransform" function and input four corners of object image, the function will return the corresponding four corners of scene image. And draw lines in these corresponding points, the object in the lines means the mapped object in scene imge.

1.7 Step 7: Localize the object

Draw lines between the four points, the object in those lines means the mapped object in the scene image. The location of object in scene image is shown in Figure 4.

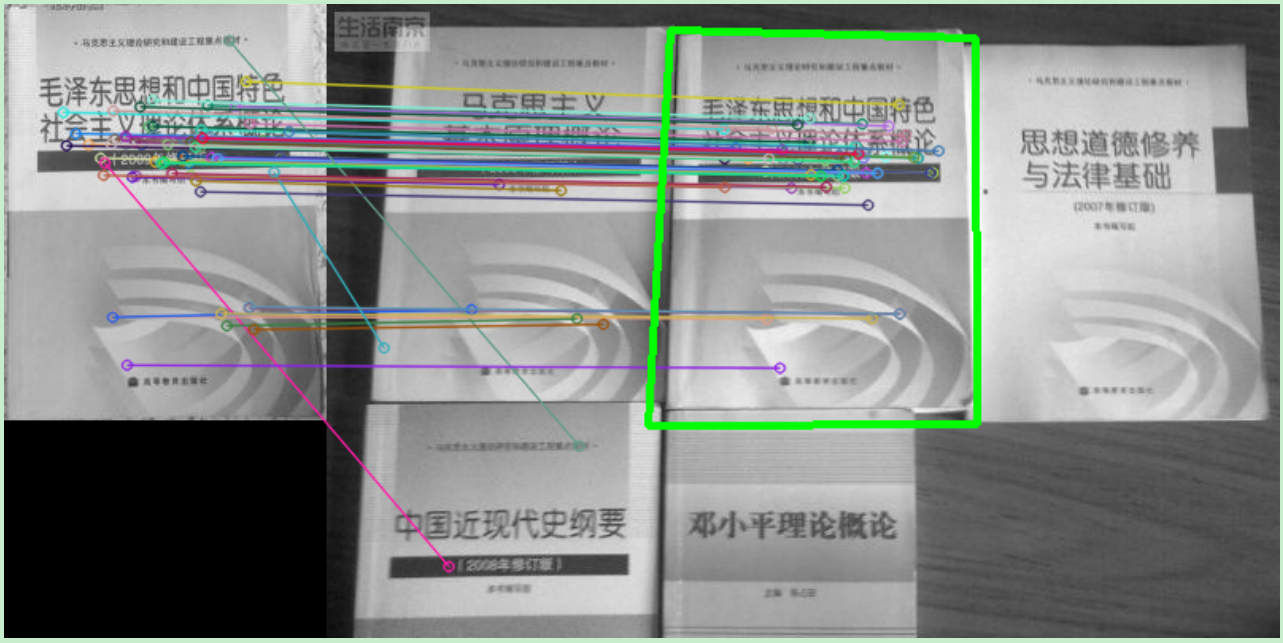

Figure 4: Location of Object