CV2015Spring—Assignment #2

Name：Wang Wenzong　　Sutdent ID: 21140211056

1. The arrangement of experiment

The whole framework of the implementation for object detection is shown below.

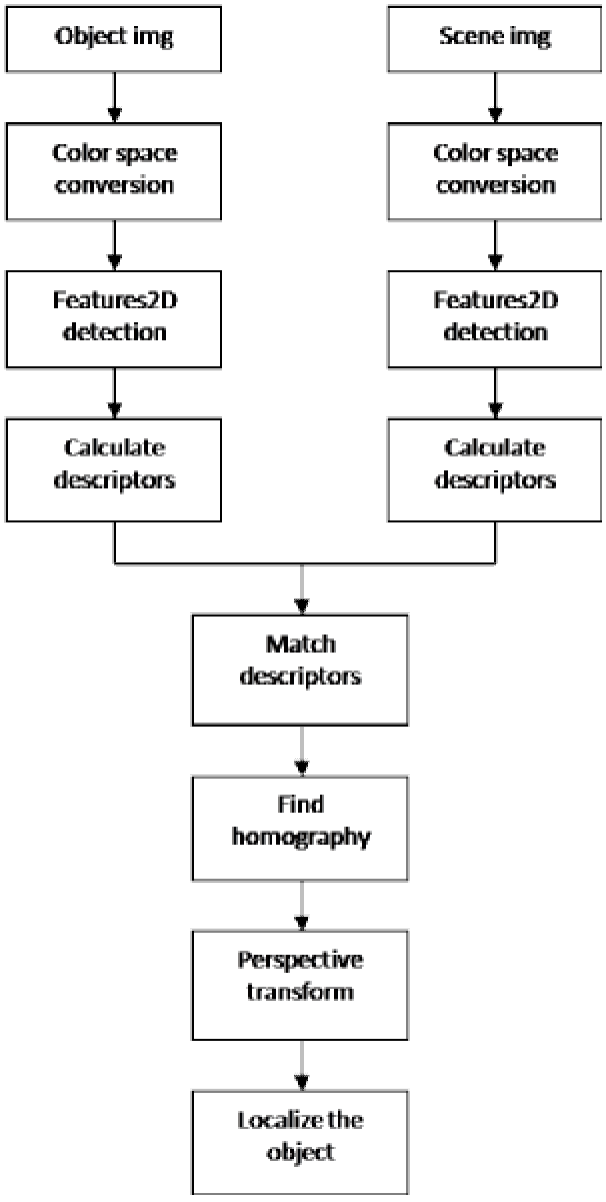

Figure 1: Framework of the implementation for object detection

1.1 Step 1: Input image

The object image and scene image is given. Firstly, I input the two images which are the object image and the scene image.

Figure 2: Input images

1.2 Step 2: Color space conversion (5 points)

Input: The input color image (m ×n ×3 matrix), m is the width of the input image, n is the height of the input image.

Output: m × n matrix.

Implementation: Here I use the cvtColor(i, i, COLOR_BGR2GRAY) to finish this job.

Figure 3: Input images in gray

1.3 Step 3: Features2D detection (15 points)

Input: The gray image (m×n×1 matrix), m is the width of the input image, n is the length of the input image.

Output: Keypoints of object image and scene image.

Implementation: There are lots of Features2D detection algorithms, such as SIFT, SURF, and so on. I've tried the two methods and the result were similar. Here I choose SURF to do this job.

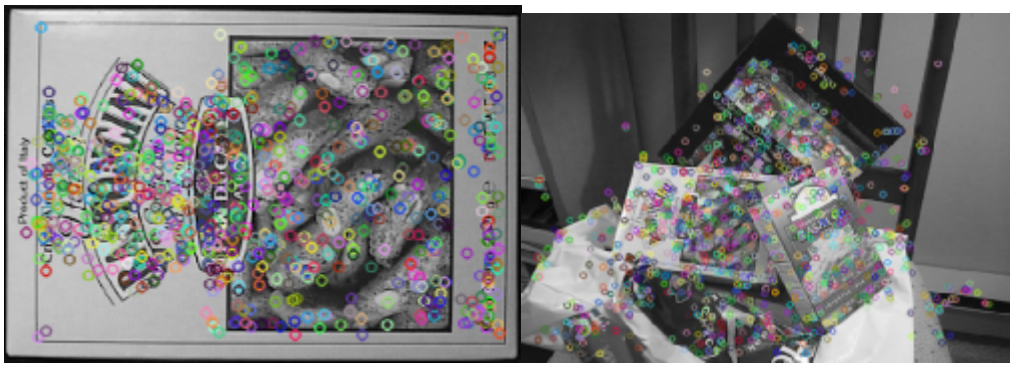

Figure 4: Features2D detection results

1.4 Step 4: Calculate descriptors (15 points)

Input: The gray image (m×n×1 matrix), m is the width of the input image, n is the length of the input image.The keypoints detected in Step 3.

Output: One matrix describes the keypoints in the object image and one matrix describes the keypoints in the scene image.

Implementation: Here I build two matries to store the keypoints of the images, one matrix is for the object image and the other matrix is for the scene image.

1.5 Step 5: Match descriptors (20 points)

Input: One matrix describes the keypoints in the object image and one matrix describes the keypoints in the scene image.

Output: M good matched keypoints between object image and scene image.

Implementation:

Step 5-1: The x×y distance(value) of the keypoints are quickly calculated(x, y are the number of keypoints) , each value indicates the similarity of different keypoints from the two images(the object image and the scene image).

Step 5-2: Fix one keypoint from object image and select the min or max distance between the keypoint and one keypoint from scene image to find the best match,then change the fixed keypoint and do the same.

Step 5-3: Only “good” matches were shown in the image below (i.e. whose distance is less than 3×min(distance)).

Hint: There are lots of distance calculation algorithms, such as Hausdorff Distance,

Euclidean Distance, and so on. I choose Euclidean Distance to calculate the similarity between two keypoints and find the best match which is fast and easy to implement.

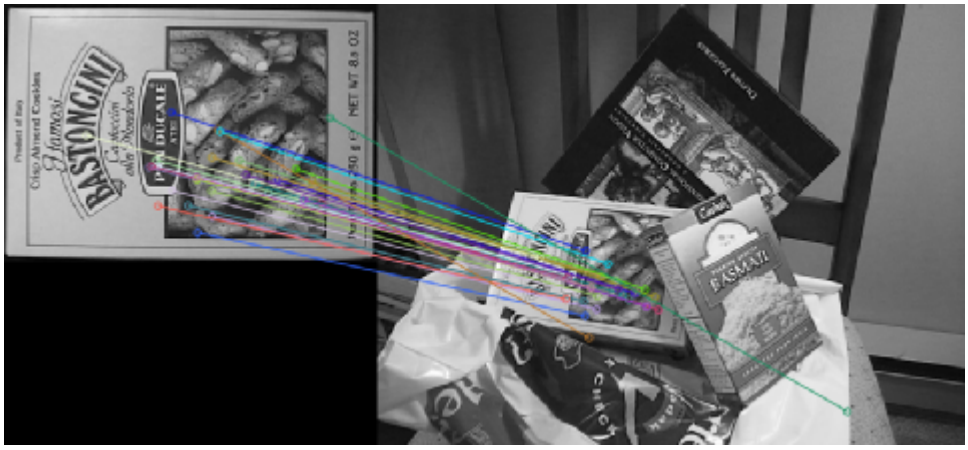

Figure 5: Matching results

1.6 Step 6: Find homography transformation (20 points)

Input: Keypoints of object image and scene image.

Output: Matrix of homography transformation between two different point sets.

Implementation:

Step 6-1: Get *N* keypoints from the good matches and store them in two different point sets.

Step 6-2: Calculate the matrix of homography transformation between the two different point sets.

Hint: There are lots of mapping transformation algorithms and in this algorithm the homography transformation is used to work out the matrix.

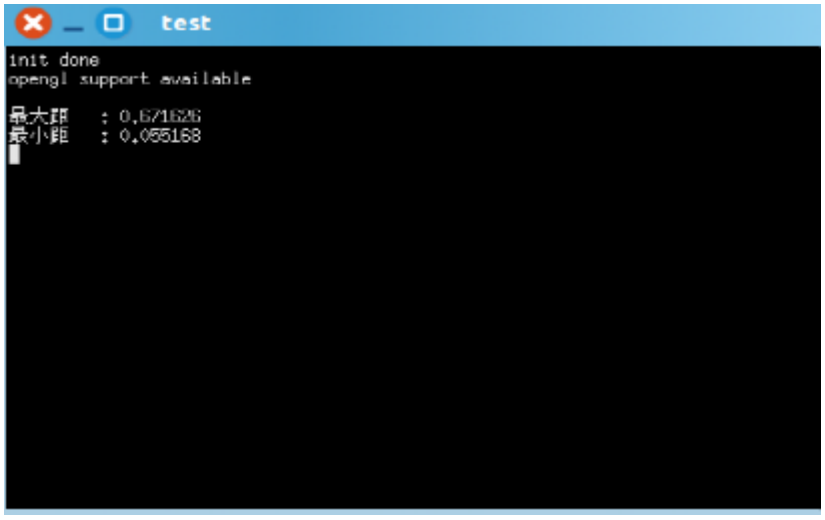

Figure 6: Find homography transformation.

1.7 Step 7: Perspective transform (20 points)

Input: Four corners of object image.

Output: Four points of scene image which are corresponding to the four corners.

Implementation:

I use the matrix of homography transformation which was got in Step 6 to perspective transform the four corners of object image to four points of scene image.

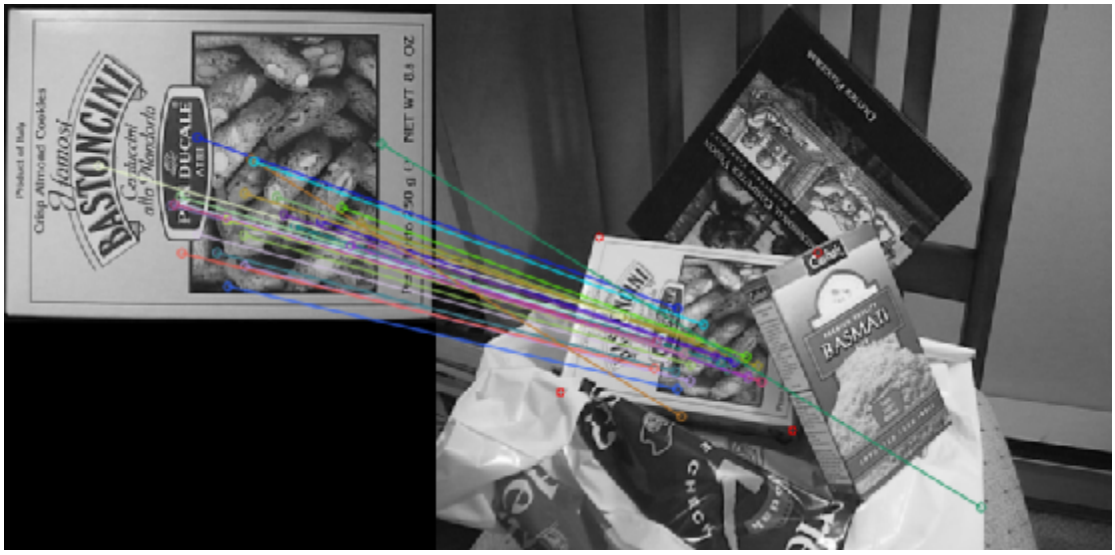

Figure 7: Perspective transform results

1.8 Step 8: Localize the object (5 points)

Input: The scene image which has four points (the mapped object in the scene image).

Output: The matrix box in the scene image which contains the object in object image.

Implementation

There are four lines which were drawn in the following image and the book was located in the green square (the mapped object in the scene image).

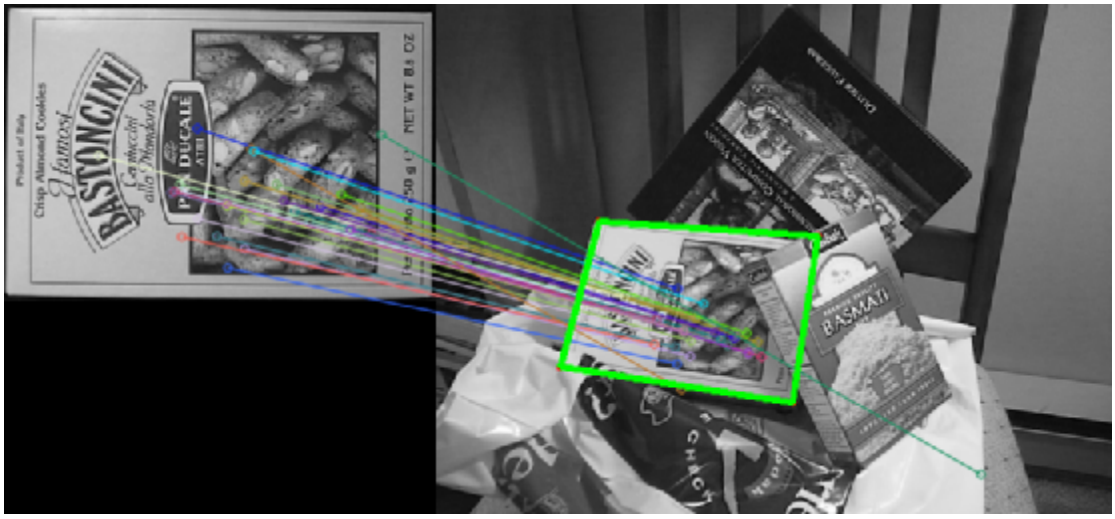

Figure 8: Localize the object