Object detection

Liuxuefei

1. Assignment requirement

For this assignment, you will implement a version of the Object Detection technique. See Figure 1, for an example. Figure-1-(a) is the object we want to detect, Figure-1-(b) is the scene which containsthe object we want to detect. What we want to do is detecting and localizing the object.

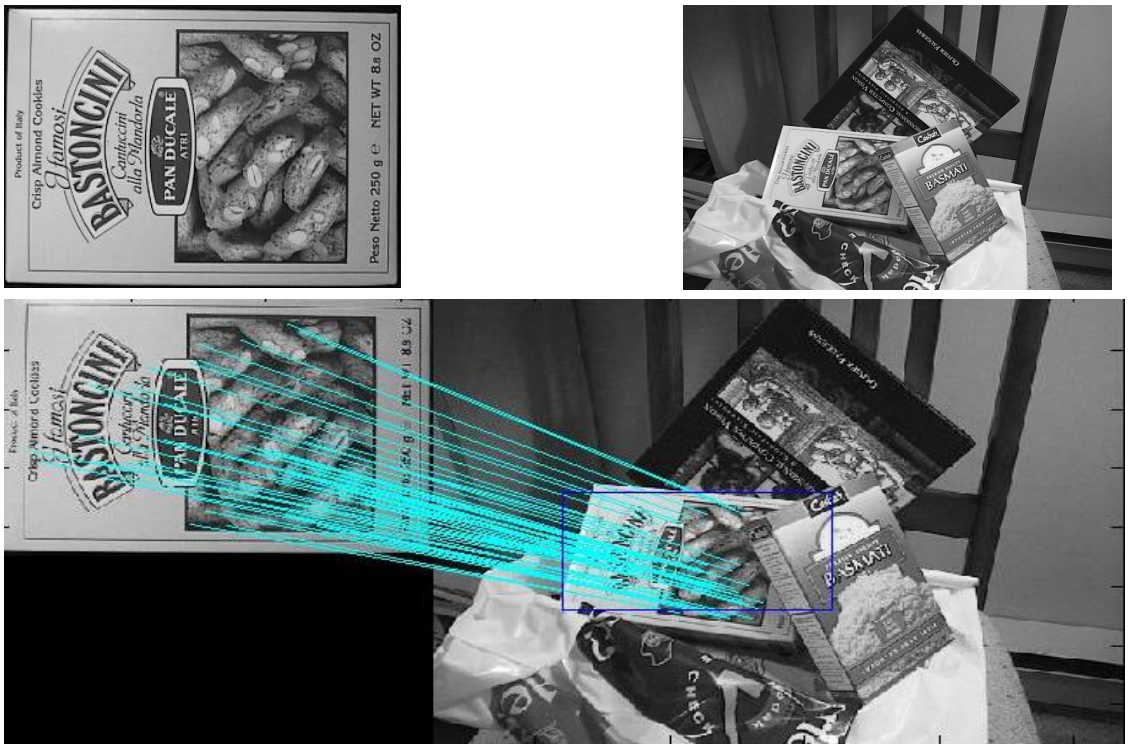
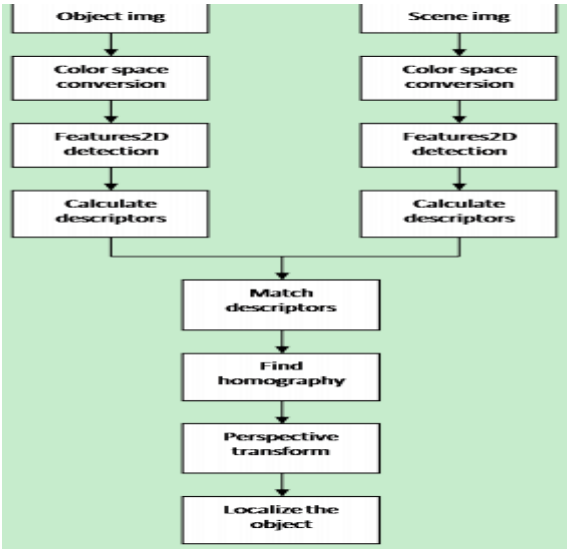


Figure 1

The whole framework of the implementation for object detection is shown in Figure 2, it may serves a reference for my assignment. Next, I will introduce the implementation and requirement of eachpart of my framework.



Step1.

Input two maps.

Step2.

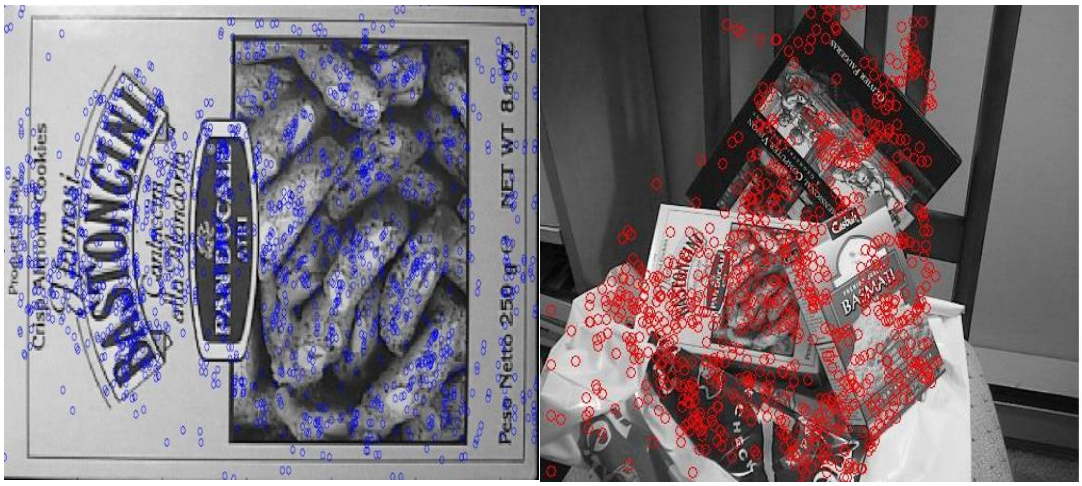
Input: The input color image (m * n * 3 matrix), m is the width of the input image, n is the height of the input image.
Output : m * n matrix.
Implementation : This step determines the kind of color space you want to use in step 2, and you should converse the color space from RGB to GRAY. As the given map is gray, we can jump this

step3. Features2D detection (sift algorithms)

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: Detect the keypoints using sift function and store the result. And using the showkeys function to show the keypoints.
Input parameters: image name.
Returned:
% image: the image array in double format
% descriptors: a K-by-128 matrix, where each row gives an invariant descriptor for one of the K keypoints. The descriptor is a vector of 128 values normalized to unit length.
% locs: K-by-4 matrix, in which each row has the 4 values for a keypoint location (row, column, scale, orientation).



Step 4: Calculate and match descriptors

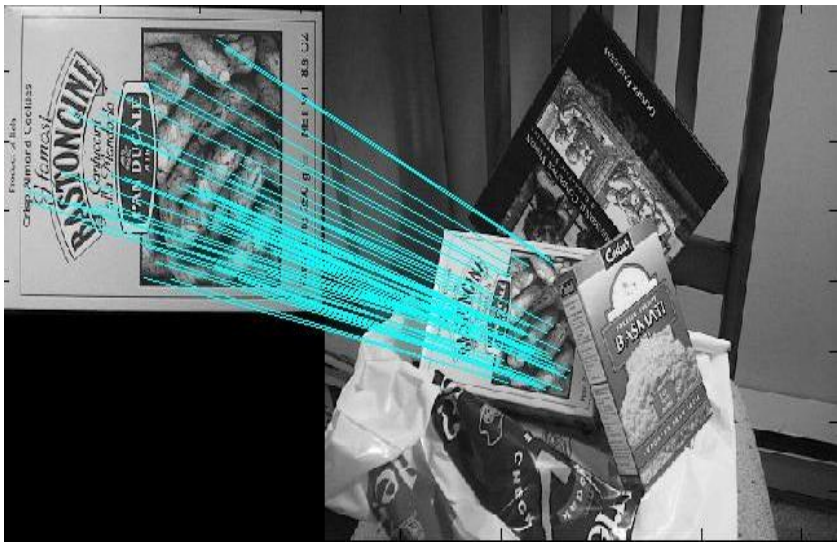
Input: The gray image (m * n * 1 matrix), m is the width of the input image, n is the length of the

input image.The keypoints detected in Step 3.

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

Implementation: For efficiency in Matlab, it is cheaper to compute dot products between unit vectors rather than Euclidean distances. Note that the ratio of angles (acos of dot products of unit vectors) is a close approximation. distRatio=0.5: Only keep matches in which the ratio of vector angles from the nearest to second nearest neighbor is less than distRatio.
to the ratio of Euclidean distances for small angles.
For each descriptor in the first image, select its match to second image. Then Check if nearest neighbor has angle less than distRatio times 2nd.

```
if (vals(1) < distRatio * vals(2))
      match(i) = indx(1);
   else
      match(i) = 0;
   end
```



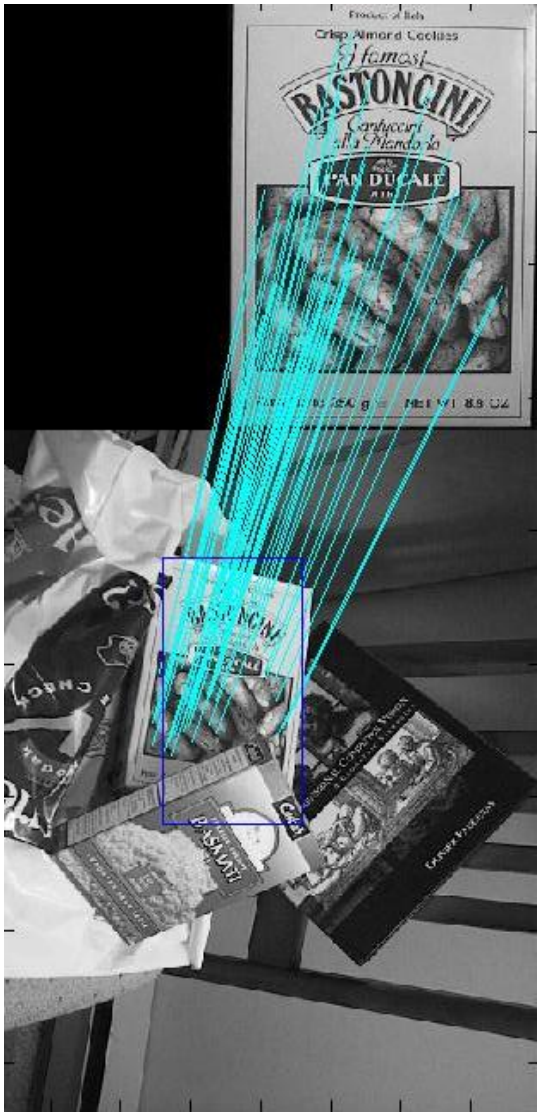
Step5. Find homography transformation

Input: Keypoints of object image and scene image.

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

Implementation: using the [H corrPtIdx] = findHomography(matchLoc2',matchLoc1') function, get the H matrix.

Step6. : Perspective transform and localize the object


Crisp Almond Cookies
famosi
BASTONCINI
PAN DI CALE