UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Computer Vision, A.Y. 2017/2018
Master Degree in ICT for internet and multimedia

Homework 3 – LAB #6

Deadline: May 31, 2018

Kalakonda	Srikanth Reddy	1178232

LAB #6 REPORT

Lab session #6 is about the Keypoints, Descriptors and matching them with other images to detect the presence of it.

We have a set of datasets with set of objects to be computed and set of scenes to be marched with .Here we created a *class ObjectDetection( )* where every object is taken in a folder iterating through the scenes.

Step: I :The input image is now given to *SIFT* method which calculates and detects the keypoints and descriptors these are these features are stored in an array .we used *vector<KeyPoint> keypoints_object, keypoints_scene* to store those values respectively from these keypoints we also compute the detectors and stores in a matrix, as they have to be matched on the destination image .we used *Mat descriptors_object, descriptors_scene* these store the values respectively.

Step II: Now we have keypoints detected and descriptors extracted.we now use them to match them so as to detect the image , We used *FlannBasedMatcher* which is called as *Fast Library for Approximate Nearest Neighbors* , to detect the matches between the descriptor vectors of scenes and objects respectively.Along with the flann based matcher we used knnMatch method which compares the matches in objects and scenes descriptors using above , and stores in this *vector< vector< DMatch> > matches* , we used a threshold of 2 , to select only those number of matches.

Step III: Now we use these matches that are stored in the *matches,* and select only good matches so that we can filter unnecessary matches and select the exact image, This is checked iteratively over the rows of the descriptors of the scene with every match stored in matches.We try to filter out the unwanted matches using a condition, and that takes the first result only if its distance is smaller than 0.6*second_best_dist that means this descriptor is ignored if the second distance is bigger or of similar.Once detected they are later pushed/Stored into an array.

Step IV: After the stage III , we have certain good and strong matches which detect the object exactly so to show that we draw them to visualize the output, we use the drawMatch method which uses the object image, scene image, keypoints of both,the strong good matches we found above and draw them on a new image.

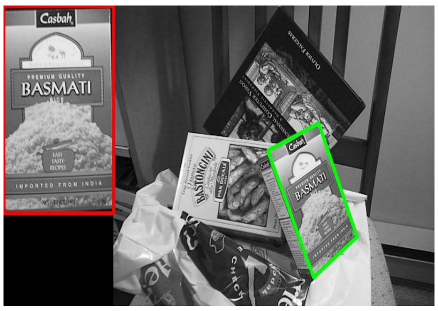
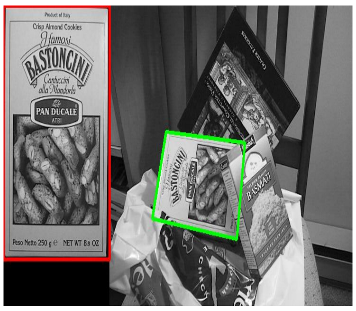
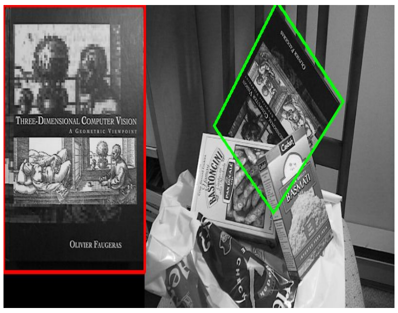
Step V: Now the last part is to localize the object image with the scene image and draw the boundaries of the detected image. So we created a method *localizeInImage* , in this method we use the keypoints of image and scenes which are compared with the good matches and are stored in an array again,later we use this values to find the homography using *findHomography(obj, scene, RANSAC);* this homography gives us the corners of the object image and that are to be drawn on the scene , so to draw the lines we use line method that draws lines on the corners specified with the color we need .

We now just throw the results on to the image and we derive the output.

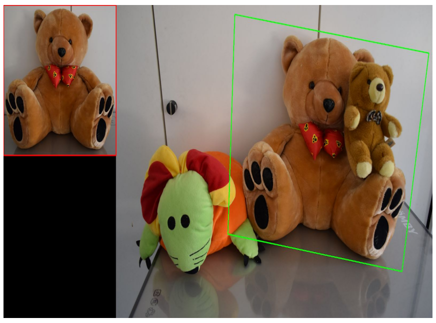
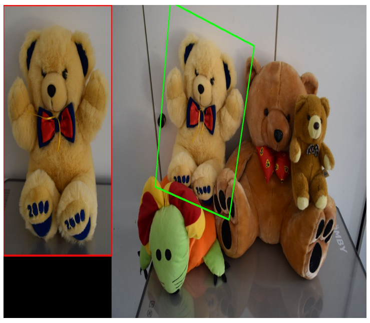
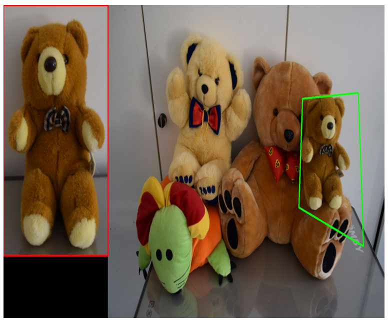
OUTPUT:

We would like to give certain samples of the output from the code we executed.

Dataset 1:

Dataset 2:

Dataset 3: