Coding Assignment 1: Recognzing specific objects with local feature matching

Manu Agarwal
UTEID: ma53767

1 Introduction

Several approaches have been proposed for object recognition. Most of these approaches fall into one of the two categories - local feature matching or global feature matching. In this experiment, we adopt the former approach and compute Scale Invariant Feature Transform (SIFT) descriptors of an image. These features are invariant to image scaling, translation and rotation and partially invariant to illumination changes and affine or 3D projection.
SIFT employs a staged filtering approach to identify stable points in scale space. Image keys are created using these features and a nearest neighbor indexing method identifies the candidate object matches. We employ this technique on several images and try to see what kind of images perform best and how invariant the SIFT features are to various kinds of transformations.

2 The Basic pipeline

Algorithm 1 Object recognition using SIFT features

Input: An object template and several test images some containing the object and some without the object
Output: Matched features between the object template and each of the test images and a boundary around the object if it is present in the test image

1: Extract SIFT descriptors in the object template
2: Repeat steps 3 to 8 for each test image
3: Extract SIFT descriptors of the test image
4: Apply a nearest neighbor algorithm to find out a potential matching between descriptors in the object template and the test image
5: Apply a raw threshold on the Euclidean distance to eliminate some false matches
6: Apply Lowe's ratio test to further eliminate some matches
7: Apply Random Sample Consensus (RANSAC) to further eliminate outliers
8: Based on a threshold on the number of inliers remaining, decide whether the object is present in the test image and draw and boundary around the object if it is present

3 Experiments

We use a threshold of 0.8 in the thresholded-nearest neighbor outlier rejection stage and a threshold of 0.6 in the Lowe's ratio test outlier rejection method. These thresholds were set by empirical observations on various images. We first ran the above algorithm with object-template.jpg as the template object and each of the three provided scene images - object-template-rotated.jpg, scene1.jpg and scene2.jpg as the test images. The matches between the SIFT descriptors after each stage of outlier filtering are shown in the Figure 1, Figure 2 and Figure 3.

We observe that there are a lot of SIFT descriptor matchings for object-template-rotated.jpg that survive the three outlier rejection techniques. This image was obtained with a simple rotation of object-template.jpg. This shows that SIFT features are invariant under rotation of objects. For scene1.jpg, we see that the algorithm does not perform as well. Although as we shall see in later sections, the algorithm succeeds in drawing a good bounding box around the tower in scene1.jpg, the number of matches drastically reduces after application of thresholded nearest neighbor and Lowe's ratio test outlier rejection methods. This might be attributable to the different lighting conditions in the two images and the distance from which the photograph was taken.

For scene2.jpg, out of 510 initial matches, only 133 remain after the thresholded nearest-neighbor test. This is understandable as there are a lot of outliers initially. The Lowe's test further brings this count down to 6 and after application of RANSAC, we get 3 matches. This is a decent result as scene2.jpg does not contain the desired object.

This experiment does confirm that SIFT features are invariant to rotation and atleast partially invariant to change in lighting conditions.

Next, we try detecting the presence of object-template.jpg in each of the three test images. The results for this are shown in Figure 4. We use a threshold of 3 on the number of inliers (after the three stages of outlier rejection) to classify an object as being present in the test images. This threshold was set by empirical observation on different images. We observe that identification is almost perfect for object-template-rotated.jpg. For scene1.jpg, the recognition is decent enough with a good boundary around the tower. One subtle thing to notice with scene1.jpg is that even if we crop a part of the tower and run our algorithm on the cropped image, the algorithm classifies the scene as containing the object. However, the bounding box drawn by the algorithm is quite improper. This shows that SIFT features match only local features. They have little sense of the image as a whole.

For scene2.jpg, the algorithm classifies the image as not containing the object which is in line with our expectations. After three stages of outlier rejection, only 3 matches remain between the image and the object-template which is below the

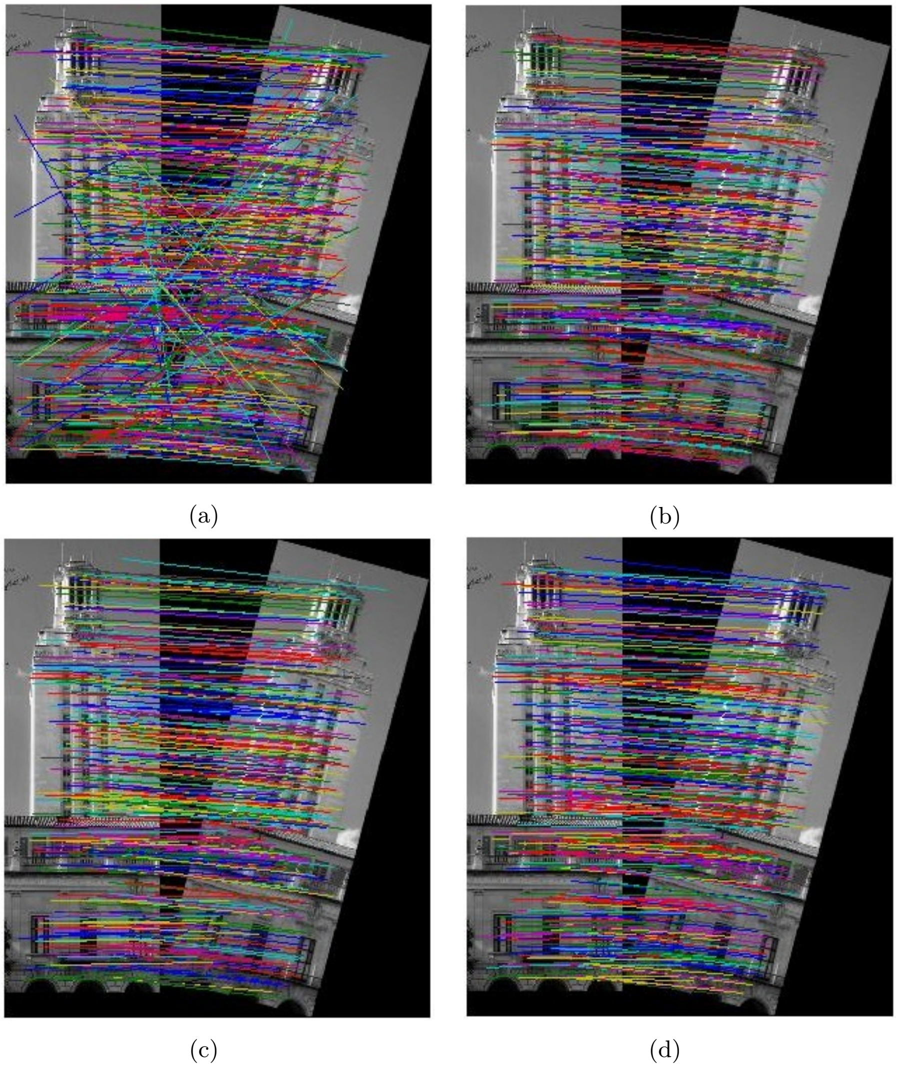


Fig. 1: Matches between object-template.jpg and object-template-rotated.jpg at various stages of the algorithm. (a) Initial matches (510) without any outlier rejection. (b) Matches (370) after application of thresholded nearest neighbors. (c) Matches (363) after application of Lowe's ratio test. (d) Matches (355) after filtering outliers with RANSAC

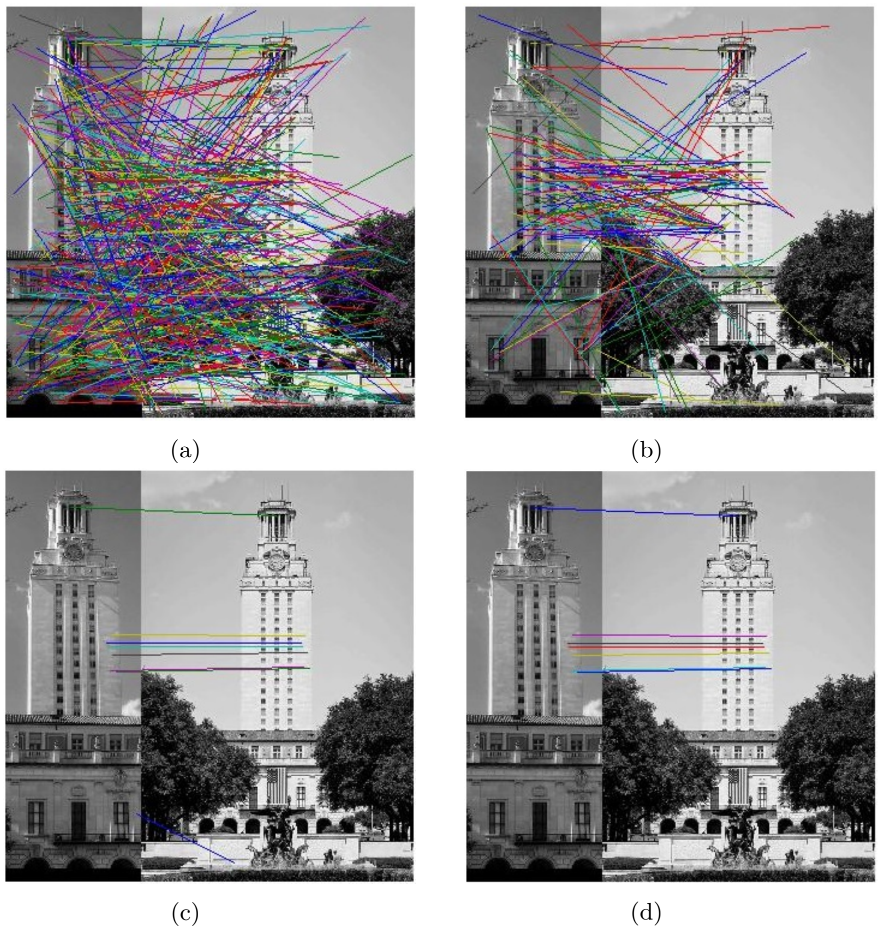


Fig. 2: Matches between object-template.jpg and scene1.jpg at various stages of the algorithm. (a) Initial matches (510) without any outlier rejection. (b) Matches (143) after application of thresholded nearest neighbors. (c) Matches (9) after application of Lowe's ratio test. (d) Matches (8) after filtering outliers with RANSAC

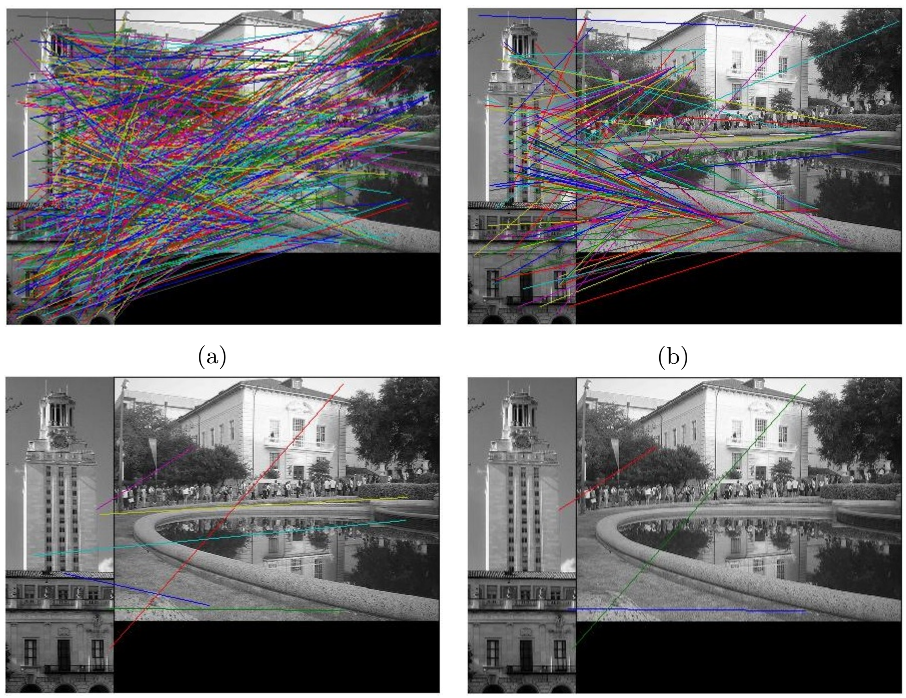
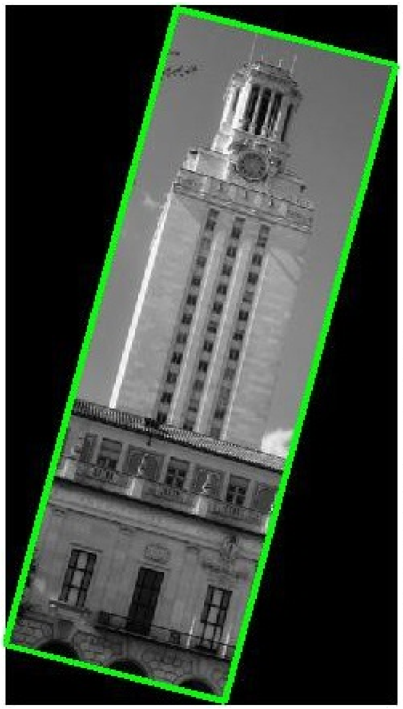


Fig. 3: Matches between object-template.jpg and scene2.jpg at various stages of the algorithm. (a) Initial matches (510) without any outlier rejection. (b) Matches (133) after application of thresholded nearest neighbors. (c) Matches (6) after application of Lowe's ratio test. (d) Matches (3) after filtering outliers with RANSAC

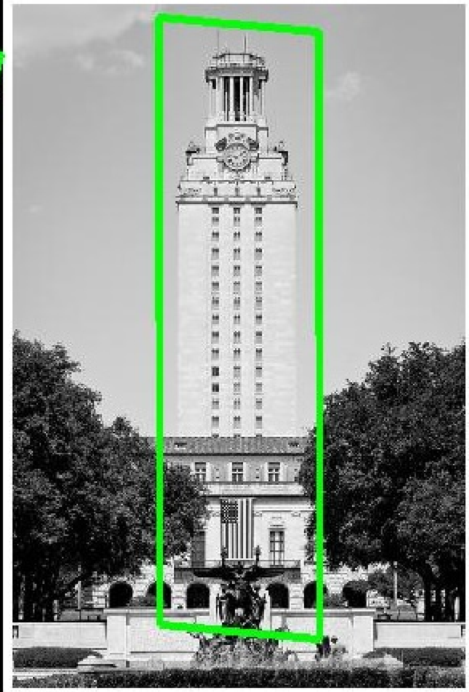
threshold. Hence, the algorithm classifies this scene as not containing the object.

We also tried our code on some random images taken from the internet and some hand-photographed images. One case that works well is of recognizing a bottle in a scene. The template object, that is, the bottle is rotated by almost 20 degrees in the scene. The scale varies largely in the template object and the scene. Also, there is a difference in the illumination of the two scenes with the template object taken without proper illumination while the scene was photographed in the presence of proper lighting. We see that the algorithm works pretty well drawing a good bounding box covering the bottle in the scene as shown in Figure 5. The number of SIFT descriptor matches is quite low though, only 4 remain after the three stages of outlier rejection. This reinforces the fact that SIFT descriptors are invariant under variation in scale, rotation and illumination conditions. The results of applying it to images of Taj Mahal is also shown in Figure 6. We also tried our code on two images of the Capitol taken under different lighting conditions and from different distances from the Capitol (Figure 7). Our algorithm fails to draw a proper bounding box in this case around the Capitol. The failure might be due to a big change in lighting conditions between the two images. It might also be due to the large variation in scale.

Finally, we try the algorithm on an image of Red Fort as the template image and that of Mysore Palace as the scene image (Figure 8). Here, Lowe's ratio test reduces the number of SIFT descriptors from 432 to 68. However, this is still a large number. RANSAC further reduces the number of matches to 5. Hence, we see that in this case, the Lowe's ration test does not reject a large number of outliers which are later rejected by RANSAC.



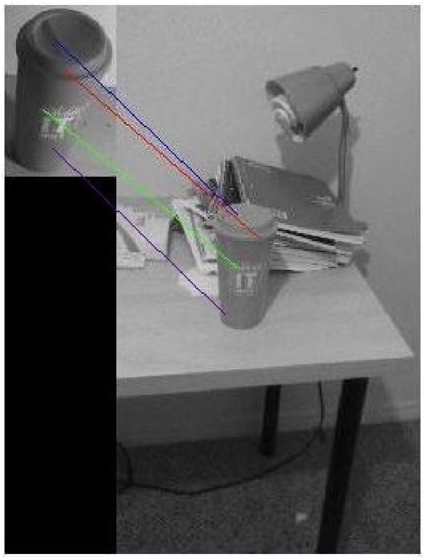
(a)



(b)

(c)

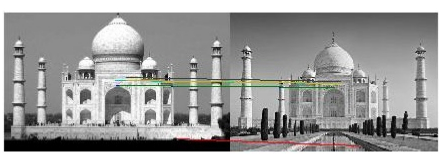
Fig. 4: (a) and (b)Images of object-template-rotated.jpg and scene1.jpg showing bounding boxes around the object present in the image. (c)scene2.jpg not containing any bounding box



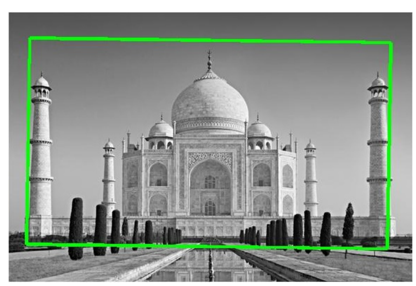
(a)

(b)

Fig. 5: (a)Matching between the template object - bottle and a scene containing the bottle. (b)The scene with a bounding box around the bottle



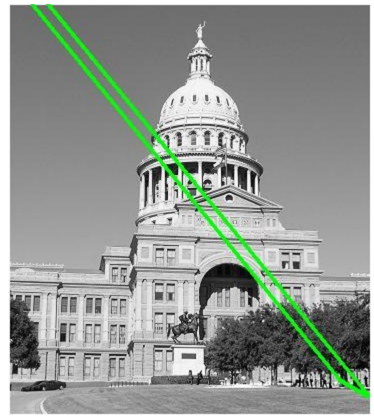
(a)



(b)

Fig. 6: (a)Matching between the template object - Taj Mahal and a scene containing Taj Mahal. (b)The scene with a bounding box around the Taj Mahal

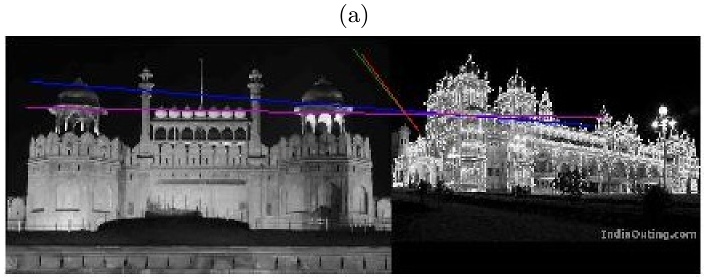
(a)



(b)

Fig. 7: (a)Matching between the template object - Capitol and a scene containing the Capitol. (b)The scene with an improper bounding box around the Capitol

(a)



(b)

Fig. 8: (a)Matching between the template object - Red Fort and a scene containing Mysore Palace after application of Lowe's ratio test (b)Matching between the template object - Red Fort and a scene containing Mysore Palace after application of RANSAC

References

1. David G. Lowe "Object detection from local-scale invariant features" [ICCV '99 Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2]
2. J. Philbin, O. Chum, M.Isard, J. Sivic, and A. Zisserman "Object Retrieval with Large Vocabularies and Fast Spatial Matching" [CVPR 2007]
3. K. Grauman and B. Leibe "Visual Object Recognition" [Synthesis Lectures on Artificial Intelligence and Machine Learning April 2011, Vol 5, No 2, pp 1-181]
4. Resources on the Visual Recognition course homepage
http://vision.cs.utexas.edu/381V-spring2016/