

Introduction to Computer Vision (ECSE 415)

Assignment 2: Image Features

Due date: 11:59PM, October 22nd, 2021

Please submit your assignment solutions electronically via the myCourses assignment dropbox. The submission should include a single jupyter notebook. More details on the format of the submission can be found below. Submissions that do not follow the format will be penalized 10%. Attempt all parts of this assignment. The assignment will be graded out of total of **100 points**. You can use OpenCV and Numpy library functions for all parts of the assignment except stated otherwise. Students are expected to write their own code. (Academic integrity guidelines can be found at <https://www.mcgill.ca/students/srr/academicrights/integrity>). Assignments received up to 24 hours late will be penalized by 30%. Assignments received more than 24 hours late will not be graded.

Submission Instructions

1. Submit a single jupyter notebook consisting of the solution of the entire assignment.
2. Comment your code appropriately.
3. Do not forget to run Markdown cells.
4. Do not submit input/output images. Output images should be displayed in the jupyter notebook itself. Assume input images are kept in the same directory as the codes.
5. Make sure that the submitted code is running without error. Add a README file if required.
6. If external libraries were used in your code please specify their name and version in the README file.
7. We are expecting you to make a path variable at the beginning of your codebase. This should point to your working local (or google drive) folder.
Ex. If you are reading an image as following: `img = cv2.imread('/content/drive/My`

Drive/Assi_2/images/Circles.png'), Then you should convert it into the following: path = '/content/drive/My Drive/Assi_2/images/'. Your path variable should be defined at the top of your jupyter notebook. While grading, we are expecting that we just have to change the path variable once and it should allow us to run your solution smoothly. Specify, your path variable in the README file.

8. Answers to reasoning questions should be comprehensive but concise.
9. Submissions that do not follow the format will be penalized 10%.

1 Harris Corner Detection (20 points)

Implement the Harris corner detector as described in class **primarily using NumPy** and show the corners as red circles overlapped on the original image. Refer to the checkerboard example below. (you are prohibited from using OpenCV or Scikit-Image Harris corner detector function) (10 points).

Note that you should convert all images to grayscale for this question.

The Harris corner detector has the following steps:

1. Apply gaussian filtering on the image.
2. Compute image derivatives using the Sobel operator.
3. Compute the square of the x- and y-derivatives.
4. Compute the cornerness function response on the output of step-3. ($\text{Determinant}(H) - k\text{Trace}(H)^2$), where $k=0.05$. (You can vary value of k for your application).
5. Perform non-maxima suppression on the output of step-4 (the cornerness function).
6. Threshold the output of the Non-maxima suppression at a user defined threshold (T). Please normalize the image using `cv2.normalize` before thresholding.

You will apply the Harris Corner Detector for three different images:

1. Checkerboard image Figure:1(a) (Input image). Change the threshold value to get detected corners similar to Figure:1(a) (Harris Corner). Observe and report the effect of changing threshold values. Show the detected corners for at least three different threshold values (T). (3 points)
2. Shape image Figure:1(b). Try different values of thresholds and report your observations. Show detected corners for at least three different threshold values. (3 points)
3. House image Figure:1(c). Try different values of thresholds and report your observations. Show detected corners for at least three different threshold values. (3 points)

2 Invariance of SIFT Features (28 Points)

You are given a reference image of a book as shown in Figure:2. Verify the invariance of SIFT features under changes in image scale and rotation. For this question, you are allowed to use in-built OpenCV/Scikit-Image functions.

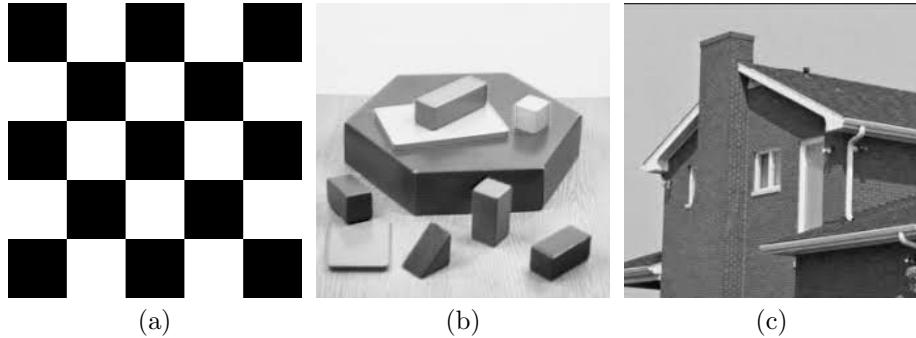


Figure 1: (a) Checkerboard, (b) Shape image Figure (c) House image Figure

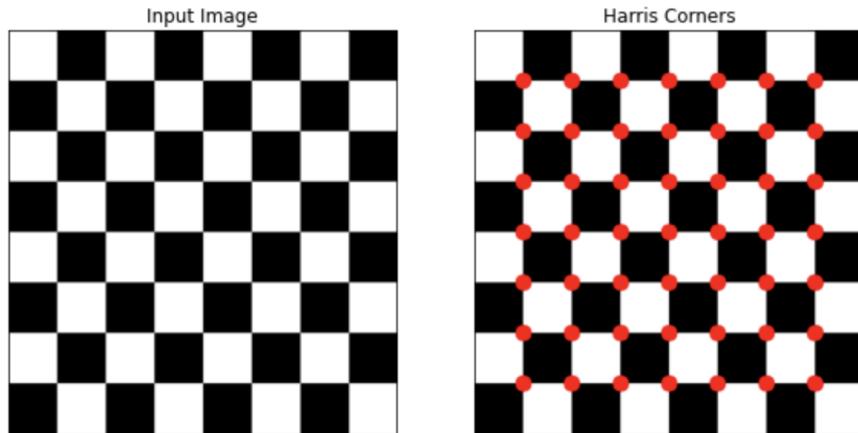


Figure 2: Harris Corners shown in checker-board example

2.1 Invariance Under Scale

1. Compute SIFT keypoints for the reference image. (2 points)
2. Scale the reference image using scaling factors of (0.2, 0.8, 2, 5). This should result in a total of 4 different transformed images. Display scaled images. (1 points)
3. Compute SIFT keypoints for all (total 4) transformed images. (2 points)
4. Match all keypoints of the reference image to the transformed images using a brute-force method (by examining all the possible pairs of key point matches). (2 points)
5. Sort the matching keypoints according to the matching distance. (2 points)

6. Plot the matching distance for the top 100 matched keypoints. Plot indices of keypoints on the x-axis and corresponding matching distance on the yaxis. (2 points)
7. Display the top ten matched keypoints for each pair of the reference image and a transformed image (the top matches are the ones with the smallest match distances). (2 points)
8. Discuss the trend in the plotted results. What is the effect of increasing the scale on the matching distance? Reason as to the cause. (1 points)

2.2 Invariance Under Rotation

1. Compute the SIFT keypoints for the reference image. (2 points)
2. Rotate the reference image at the angles of (10, 90, 130, 180). Display the rotated images. (1 points)
3. Compute the SIFT keypoints for all (total of 4) transformed images. (2 points)
4. Match all keypoints of the reference image to the transformed images using a brute-force method. (2 points)
5. Sort the matching keypoints according to the matching distance. (2 points)
6. Display the top ten matched keypoints for each pair of the reference image and a transformed image. (2 points)
7. Plot the matching distance for the top 100 matched keypoints. Plot the indices of keypoints on the x-axis and the corresponding matching distance on the yaxis. (2 points)
8. Discuss the trend in the plotted results. What is the effect of increasing the angle of rotation on the matching distance? Reason as to the cause. (1 points)

3 Image Stitching (40 Points)

You are given three different views of the same scene in a folder ‘stitching images’ (Figure:3). Follow these steps in order to stitch given images:

1. Compute the SIFT keypoints and corresponding descriptors for images 1 and 2. (2 points)
2. Find matching keypoints in images 1 and 2 and display the 20 best pairs. (5 points)



Figure 3: Image of a Book

3. Find the homography that best matches the keypoints from image 1 and 2 using the RANSAC method, and apply the resulting transformation to image 1. Image 2 should not be transformed. (5 points)
4. Stitch the transformed image 1 and the original image 2 together using linear image blending. Let us call this image 12. Display this image. (5 points)
5. Compute the SIFT keypoints and corresponding descriptors for images 12 and 3. (2 points)
6. Find the matching keypoints in 12 and 3 images and display the 20 best pairs. (4 points)
7. Compute the homography using the RANSAC method. Apply the transformation to image 3. Image 12 should not be transformed. (5 points)
8. Stitch the transformed image 3 and image 12 together using linear image blending. Display the resulting image. (5 points)
9. Discuss: Note that we could also use multi-band blending in the section 4.8. When should one prefer pyramid blending over linear blending? (5 points)



Figure 4: Three different views of the same scene (a) 1 (b) 2 (c) 3

4 Image Matching (14 points)

Using SIFT features show which (if any) of the three objects shown on the left are present in the image on right? (example is from Lowe 1999)
 Discuss whether the same can be inferred using just the SSD metric on the images and why?



Figure 5: Image of box, Image of a scene