Introduction to Computer Vision (ECSE 415) Assignment 2: Image Matching and Face Detection

Due date: 11:59PM, October 19th, 2020

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 **85 points**.

You can use OpenCV and Numpy library functions for all parts of the assignment unless specified 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 48 hours late will be penalized by 30%. Assignments received more than 48 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. Answers to reasoning questions should be comprehensive but concise.
8. Submissions that do not follow the format will be penalized 10%.

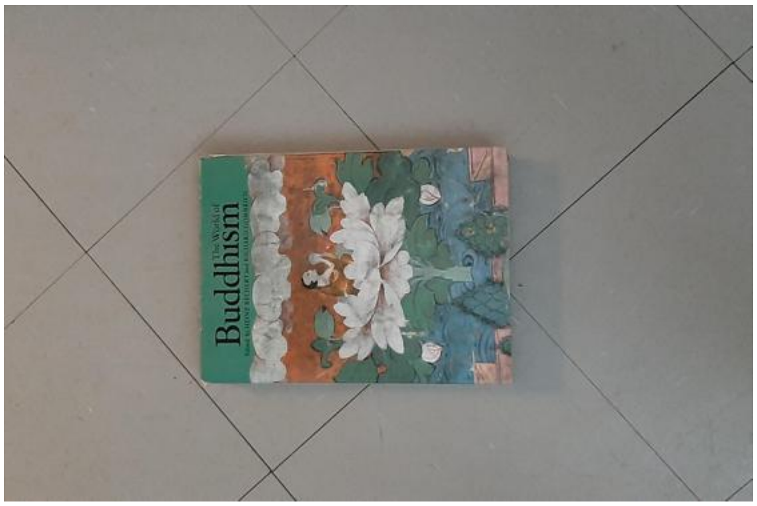


Figure 1: Reference image of a book.

1 Invariance of SIFT Features (34 Points)

You are given a reference image of a book as shown in Figure 1. Verify the invariance of SIFT features under changes in image scale and rotation.

1.1 Invariance Under Changes in Scale

1. Compute SIFT keypoints for the reference image. **(2 points)**

~~2. Scale reference image using scaling factors of (0.2, 0.5, 0.8, 1.25, 2, 5). **(2 points)**~~

~~3. Compute SIFT keypoints for the 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 matching keypoints according to the matching distance. **(2 points)**~~

~~6. Display top ten matched keypoints for each pair of reference image and a transformed image. **(2 points)**~~

~~7. Plot the matching distance for top 100 matched keypoints. Plot indices of keypoints on x-axis and corresponding matching distance on y-axis. **(2 points)**~~

8. ~~Discuss the trend in the plotted results. What is the effect of increasing the scale on the matching distance? Reason the cause. **(3 points)**~~

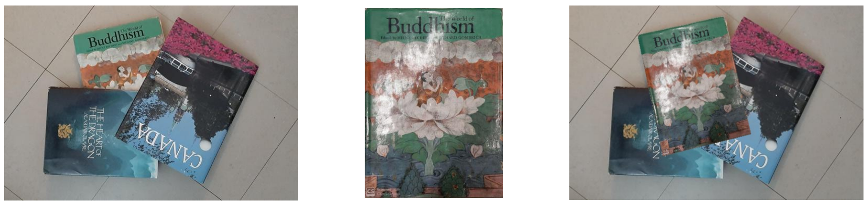
1.2 Invariance Under Rotation

1. ~~Compute SIFT keypoints for the reference image. **(2 points)**~~
2. ~~Rotate reference image at the angle of (10, 30, 90, 150, 170, 180). **(2 points)**~~
3. ~~Compute SIFT keypoints for the 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 matching keypoints according to the matching distance. **(2 points)**~~
6. ~~Display top ten matched keypoints for each pair of reference image and a transformed image.~~ **(2 points)**
7. ~~Plot the matching distance for top 100 matched keypoints. Plot indices of keypoints on x-axis and corresponding matching distance on y-axis. **(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 the cause. **(3 points)**~~

2 Matching using SIFT - Book Reveal (16 Points)

You are given an image of the reference book taken under different acquisition conditions: (a) under occlusions (Figure 2(a)) and (2) under different lighting conditions (Figure 2(b)). The task is to transform the image of the book in Figure 2(a) and align and merge it with the occluded view in (Figure 2(b) to generate an image with an unoccluded view of the book (See Figure 2(c)). To achieve this objective, please perform following steps:

~~1. Find SIFT keypoints in given input images.~~ **(2 points)**



(a) (b) (c)

Figure 2: Input and desired output for image manipulation task (a) image of an occluded book (book occlusion.jpg) (b) reference image of a book in different lighting condition (book crop.jpg) (c) desired output.

Figure 3: Examples of CelebA dataset images.

2. ~~Match keypoints of reference image to the keypoints of the occluded image using brute-force method. **(2 points)**~~
3. ~~Sort matching keypoints according to the matching distance. **(2 points)**~~
4. ~~Display top ten matching keypoints. **(2 points)**~~
5. ~~Compute a homography to align the images using RANSAC method and apply the transformation on the reference image. **(6 points)**~~
6. ~~Paste transformed reference image on the occluded view to generate un-occluded view as shown in Figure 2(c). **(2 points)**~~

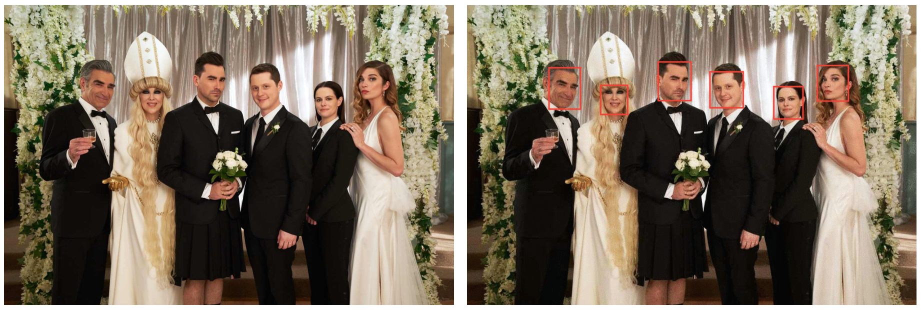
3 Face detection (35 Points)

In this question, you will work on the task of Face detection. For this purpose, you will explore two different algorithms for this purpose: (1) EigenFaces and (2) Viola-Jones detector.

We will use a publicly available celebA face dataset (Figure 3). A subset of 1000 images is given with the assignment. From these images, choose any random 100 images as your training dataset.

3.1 Eigenface Representation

Produce eigenface representation for your training data through PCA. Please note that you are **not** allowed to use the in-built PCA function in OpenCV/Scikit-Learn. You should implement Snapshot method for PCA (covered in class -



(a) (b)

Figure 4: Face Detection: (a) Group Image and (b) example of detected faces with bounding boxes around the detected faces.

Lecture 8 - Slide 55) **from scratch using numpy (15 points)** [1]. Display first 6 eigenfaces **(3 points)**

3.2 Face Detection

You will now detect all the faces in a group image (Figure 4 (a)). Use a sliding window to detect the faces. We will follow PCA base detection algorithm covered in class (Lecture 8 - Slide 63). Set a threshold on the distance in eigenspace between the window contents and your training data. Try different values of thresholds and use the one which gives you good results. Display your image with bounding boxes around detected faces for your best threshold (ex. Figure 4 (b)) [2] **(10 points)**. How well does the method work? How many false positive face detections do you get? **(2 points)**.

Use an existing implementation of the Viola-Jones face detector, and compare the results with your detector (e.g. how many false positives do you obtain?). Under what conditions would you expect the Viola-Jones detector to work when PCA does not? **(5 points)**

[1] You are allowed to use numpy.linalg for computing eigenvalues and eigenvectors

[2] You can use any online available code for bounding box generation. Please cite the source for this in your report.