Exercise 8 for MA-INF 2201 Computer Vision WS2022/2023
Submission deadline: 5.01.2023

1. **Statistical Shape Modeling.** Build a PCA based statistical shape model $\mathcal{M}$ using the data in *hands_aligned_train.txt*. The data contains a set of 56 corresponding landmark points on hand-contours from 39 instances, which have already been aligned using Procrustes Analysis. Refer to the *readme* file for details about the data organization. The model $\mathcal{M}$ is formulated by defining the subspace model as:

$$w_i \approx \mu + \sum_{k=1}^{N} \phi_k h_{ik} \tag{1}$$

$$\mathcal{M} = \{\mu, \phi_1, \phi_2, \ldots, \phi_N\}, \tag{2}$$

where N is the minimum number of principal components preserving 90% of the energy. Visualize μ and the effect of the weights $W = (-0.4, -0.2, 0.0, 0.2, 0.4)$ of each ϕ_k.

Remarks: Please implement PCA yourself. You may use *np.linalg.eig* and *np.linalg.svd*.

(5 Points)

2. **Inference:** Given the test shape in *hand_aligned_test.txt* and the generated model $\mathcal{M}$, estimate the values of h_{ik} using the ICP approach. Also, visualize the reconstructed shapes and calculate the RMS error between the reconstructed shape and the original shape for all of the iterations.

(3 Points)

3. **Eigenfaces**. In this exercise, we will perform face detection and face recognition using the Eigenfaces method. Here, you are allowed to use the *scikit-learn* library.

   (a) Use PCA to learn the k principle component of face images in the LFW dataset. Split the data randomly into 80% for training and 20% for testing. Display the first 10 Eigen Faces. Find the average reconstruction error of the images residing in `data/task3/detect/face` using the k principal components. You can set $k = 100$ or any other reasonable value.
   (3 Points)

   (b) Using the learned PCA, detect the face images in `data/task3/detect`. The images are organized into two directories `face` and `notFace` which indicate whether an image is of a face or not. Calculate the accuracy of your method.
   (2 Points)

   (c) Using the learned PCA, recognize the images in the test set split, i.e. given a test face image, find out to whom this face belongs. To do that, apply the Nearest Neighbor classifier with respect to the training set. Calculate the accuracy of your recognition method.
   (1 Point)

4. **Corner Detectors**: Impelementing Harris corner detector and Förstner corner detector

(a) Implement the structural tensor $M = \sum w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$, where I_d is the image gradient in the d direction and w is a convolutional kernel.
(2 Points)

(b) Implement Harris corner detector, then display the response function and the detected corners of the image `data/task4/palace.jpeg`.
(2 Points)

(c) Implement Förstner corner detector and display $w = \frac{det(M)}{Tr(M)}$ and $q = \frac{4det(M)}{Tr(M)^2}$ after thresholding, as well as the detected corners of the palace image. Do you observe differences to the results obtained by Harris corner detector?
(2 Points)

5. **Keypoint Matching**:

(a) Read the image `data/task5/castle.jpeg` then use SIFT to extract its keypoints and the corresponding descriptors. Create several transformed images of castle.jpeg by applying one of the transformations: scale, rotation, translation, and affine for each instance. You may cv2 SIFT class which can be created using`cv2.SIFT_create()`, as well as cv2 transformation functions.
(2 Points)

(b) Implement the best match ratio test with the threshold 0.5 (or any other reasonable value). Implement an additional matching criteria that looks at the best 2 matches. Please implement the matching yourself. Display the matched keypoints with both criterias using the function `cv2.drawMatches()`. Point out a case where having a second best match criteria makes a difference.
(2 Points)

6. **RANSAC for homography.** In this exercise we will implement RANSAC algorithm for estimating homography to create a mosaic.

(a) Implement the RANSAC algorithm (yourself) and apply it to stitch the two images that reside in `data/task6`.
(5 Points)