ENGN4528/6528 Computer Vision - 2018
Computer Lab 3 (CLab-3)

Objective:

Goals of CLab-3 are to help you to

(1) practise Eigen-Face based technique for face representation, face detection and recognition.

(2) familiar with DLT algorithm for two-view homography estimation.

Before doing this lab, you need make sure you have studied (and understood) the related textbook materials and lecture notes on these topics.

The expected workload for completing CLab-3 is 6~7 hours (excluding textbook reading/study time).

C-Lab-3 Tasks:

Task-1: Face Recognition using eigenface technique (8 marks)

1. In this task, you are given a "training_set" dataset of totally 135 face images captured from 15 individuals (i.e. available in "Yale-FaceA.zip"). Each individual has 9 images in the "training" dataset. You are also given 10 test images in the "Test_set" directory.

(1) Open the training images, and have a look at them, and get an idea about what the images look like.

(2) Please use all the 135 images to train Eigen-faces.

Specifically, at a minimum your face recognition system should do the following:

2. Read all the 135 face images, represent each of the images as a single high dimensional vector, and collect all the vectors into a big data matrix.
3. Perform PCA on the data matrix.
4. Determine the top *k* principal components, display the top-k eigenfaces on your screen (and also include them in your Lab Report with proper caption). Try both k=5 and k=10 cases.

5. For each of the 10 test images, read in the image, determine its projection onto the basis spanned by the top *k* eigenfaces. Use this projection, do a nearest-neighbour search over all the 135 faces, find out which three images are the most similar faces. Display these top 3 faces next to the input test image on screen (and in your Lab report). What is the recognition rate of your method?

Hints for doing PCA-Eigen Face computation:

(1) Before doing anything, you must make sure that all face images must be geometrically aligned. Specifically, you need to take into account of the differences in position of the face in each image. A simple way is, before you do any training or testing, you should manually crop the face region out, define a standard window size, resize the face image, make sure the face region are all aligned -- e.g. eyes, noses, mouths are roughly at the same positions in an image --save the results into disk, so you don't have to do the above pre-processing more than once.

(2) In doing eigen-decomposition, always remember to subtract the mean face and, when reconstructing images based on the first k principal components, add the mean face back in at the end.

(3) You can Matlab's eigs() functions to implement PCA. Other than this, you should not use Matlab's inbuilt eigen-face function if there is one. If you encounter some difficulty in solving the eigenvalues/eigenvectors for inner-product matrix (A*A'), think about whether or not you can instead solve the eigen problem for matrix (A'*A). Explain (in your Lab Report) why this is possible ?

(4) You may choose to use Python to complete this task.

Task-2: DLT for 2-view homography estimation

(7 marks)

A transformation from the projective space $\mathbb{P}^3$ to itself is called homography. An homography is represented by a 3×3 matrix with 8 degrees of freedom (scale as usual does not matter):

$$\begin{bmatrix} x^C \\ y^C \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x^R \\ y^R \\ 1 \end{bmatrix} \qquad (1)$$

The goal of this task is to the DLT algorithm to estimate a 3x3 homography matrix.

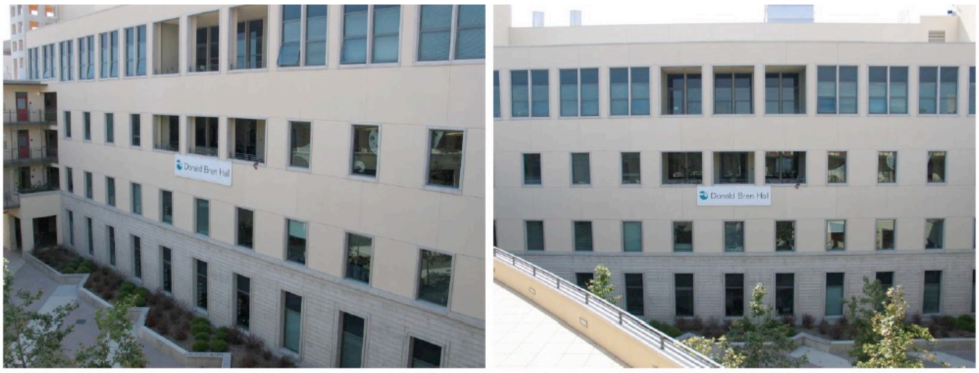

Fig. 1. The left images shows one of the almost planar facades of the Bren School with a large perspective distortion. The goal is to construct a projective transformation that rectifies the image producing a result similar to the right image.

Manually pick any six corresponding coplanar points in the images `left.jpg` and `right.jpg` and get their image coordinates.

In doing this step you may find it useful to check the `Matlab` function `ginput`. This is however only optional.

Calculate the 3x3 homography matrix between the two images, from the above 6 pairs of corresponding points, using DLT algorithm. You are required to implement your function in the following syntax.

H=DLT(u2Trans, v2Trans, uBase, vBase)

```
% Computes the homography H applying the Direct Linear Transformation
% The transformation is such that
% p = H p' , i.e.,:
% (uBase, vBase, 1)'=H*(u2Trans , v2Trans, 1)'
%
% INPUTS:
% u2Trans, v2Trans - vectors with coordinates u and v of the transformed image point (p')
% uBase, vBase - vectors with coordinates u and v of the original base image point p
%
% OUTPUT
% H - a 3x3 Homography matrix
%
% your name, date
```

In doing this lab task, you should include the followings in your Lab-Report PDF file:

1. List your source code for homography estimation in your PDF file.

2. Display the two images in your PDF file, as well as the location of the six pairs of corresponding points.

3. List the 3x3 homography matrix H that you have calculated. Explain how many points are minimally required in order to apply DLT to estimate a Homography.

======= End of CLab-3 =====