Assignment 4 **COMP 576, spring 2018** **5 March 2018**
Eigenfaces, Gaussian, Sampling **Due Thursday, 22 March. 2018**

1. You are provided a file of 80 grayscale versions of the photos of faces of males without facial hair. Each of those have been centered at the middle of the image, scaled to have about the same facial areas, sampled to a 128 × 128 image, and modified in intensity to have the same distribution of intensities in the region of the face. These are in the files images.zip and images.mat in the Sakai folder *Resources/Homework Resources/HW4 images*. The zip file has all of the individual images. The images.mat file that has all 80 images in a single 80 × 16,384 array, from which you can load (the matlab command is load('images.mat', 'im_array')). You will use a random collection of ¾ (i.e., 60) of these greyscale pictures as training images in this assignment. The rest will be used as test images.

   From this set of images you will create the 60 × 16,384 matrix A of training images. Then compute the 60 ×60 matrix AA^T, and use the MATLAB function *eig* to compute the 60 eigenvalues and 60 corresponding eigenvectors of AA^T. Multiply each of these by A^T to obtain the (non-unit) eigenvectors of A^TA, i.e., a basis spanning the set of training images. Normalize each of the first 40 eigenvectors in decreasing order of eigenvalue to produce unit-length eigenfaces $\underline{u}^1$ through $\underline{u}^{40}$.

   Convert each unit-eigenface to an image, and display it. You can use *reshape* to do each of these conversions.

   Using the dot-product method, for each *training* face compute the 40 coefficients $\underline{b}$ of these unit-length eigenfaces, and by display convince yourself that for each of your first 4 training faces, $\Sigma_{k=1}^{40} b_k \underline{u}^k$ looks rather like the original training image (note that you are replacing 16,384 values by 40, quite a compression). Turn in to the grading system the 4 face images, $\Sigma_{k=1}^{40} b_k \underline{u}^k$. Also, for each of these 4 training mages, turn in the average, over the 4 images and over their pixels, of the square root of the average squared pixel difference.

2. Now pick one of test faces and call it Face(0) with coefficients $\underline{b}(0)$. That is, Face(0)=$\Sigma_{k=1}^{40} b_k(0) \underline{u}^k$. Pick another of the test faces, and call it Face(1) with coefficients $\underline{b}(1)$. That is, Face(1) = $\Sigma_{k=1}^{40} b_k(1) \underline{u}^k$. Turn in to the grader the two images, $\Sigma_{k=1}^{40} b_k(0) \underline{u}^k$ and $\Sigma_{k=1}^{40} b_k(1) \underline{u}^k$.

   What you are now going to do is to produce a morph Face(t), with t running from t=0 to t=1.You will do the morph from Face(0) into Face(1) by linearly interpolating each coefficient to produce $b_k(t)$ from t=0 to t=1: $b_k(t) = b_k(0) + (b_k(1) - b_k(0)) * t$. Using any of the 20 test images as Face(0) into Face(1), evaluate those coefficients at t=0.25, t=0.5, and t=0.75, and from these coefficients reconstruct the morphed images into Face(t) at t=0.25, t=0.5, and t=0.75. Show the morph by displaying the 5 images Face(t) at t=0, t=0.25, t=0.5, t=0.75, and t=1. Turn in to the grader the morphed face at t=0.25

3. Finally, pick 4 of the *test* face images. For each of them, by the dot product method compute its coefficients $\underline{b}$ of the unit-eigenfaces computed in your training from the 4 images. Compute and then display (and for the first of your chosen images turn in to the grader) the image $\Sigma_{k=1}^{40} b_k \underline{u}^k$. Compute the average squared difference between the image and the corresponding "compressed" image, $\Sigma_{k=1}^{40} b_k \underline{u}^k$, and compute and turn in the square root of the average of these image-by-image averages. For your own education (there is nothing more to turn in), compare this average root mean squared pixel difference value to that you found for the training images in problem 1.

4. Compute and turn in to the grader a square image of a sinusoid whose frequency in y is 0 and whose frequency in x is ν=k/(the number of pixels in x) for some integer k < one quarter the number of pixels in each dimension of the image. Now compute and turn in to the grader the same square but where the frequency in x is

$(\nu + 1)$. Compare the two images (there is nothing to turn in for this part). Explain this in terms of aliasing (there is nothing to turn in for this part).

Down-sample by a factor of 4 in each dimension these two images, and display and turn in the results. Do that resampling by dividing the image into non-overlapping groups of 4×4 pixels and averaging each group of 16 pixels to form a single pixel in the result. Compare these to the actual sinusoid images at these down-sampled pixels as well as to each other (there is nothing to turn in for this part). Explain this in terms of aliasing (there is nothing to turn in for this part).

5. This problem has nothing to turn in, but in class I will call on you at random to give the solution. It deals with the resolution of an image that is result of Gaussian convolution by the imaging device. You will solve this problem in 1D, i.e., for an "image" $I(x)$. A scene made from two δ functions (unit impulses) centered at $x=0$ and separated by Δx can be written $\delta(x+\Delta x/2) + \delta(x-\Delta x/2)$. Their image would be $I(x) = (1/(\sigma\sqrt{(2\pi)}))\exp(-½(x+\Delta x/2)^2/\sigma^2) + (1/(\sigma\sqrt{(2\pi)}))\exp(-½(x-\Delta x/2)^2/\sigma^2)$. Plot this function for Δx large enough for the two bumps to be separated and for Δx small enough so that the two bumps have merged into one bump.

   You can see in the plots that when Δx is large enough, that image midway between the impulses (at $x=0$) will be concave upward (have positive 2nd derivative), so the two bumps would be separated (the bumps are resolved). Also, when Δx is too small, that image would be concave downward (have negative 2nd derivative) at $x=0$, so the two bumps would not be separated (would be unresolved). The critical value of Δx, where the two bumps merge to become one bump, is when $I''(0)=0$, and this distance is called the *resolution* of the imaging device..

   Calculate $J''(0)$ for J = a Gaussian. It follows that $I''(x)=$ $(1/(\sigma\sqrt{(2\pi)}))\{[(x+a)^2/\sigma^4-1/\sigma^2)]\exp(-½(x+a)^2/\sigma^2) + [(x-a)^2/\sigma^4-1/\sigma^2)]\exp(-½(x-a)^2/\sigma^2)\}$,where $a=\Delta x/2$. Evaluate this at $x=0$ and find the value of a for which $I''(0)=0$. Plot $I(x)$ for the corresponding value of Δx. Conclude that the critical value of Δx, i.e., the resolution of the camera, is proportional to σ.