Assignment 4: CS 663, Digital Image Processing

Due: 21st October before 11:55 pm

Instructions for submission are at `www.cse.iitb.ac.in/~suyash/cs663/submissionStyle.pdf`
5 points are reserved for submission in the described format.

1. In this part, we will apply the PCA technique for the task of image denoising. Take the barbara256.png image from the folder for homework 4 - this image has gray-levels in the range from 0 to 255. Add zero mean Gaussian noise of $\sigma = 20$ to it using the MATLAB code 'im1 = im + randn(size(im))*20'. Note that this noise is image-independent. If during the course of your implementation, your program takes too long, you can instead work with the file barbara256-part.png which has size 128 by 128 instead of 256 by 256.

   (a) In the first part, you will divide the entire noisy image im1 into overlapping patches of size 7 by 7, and create a matrix $\mathbf{P}$ of size $49 \times N$ where N is the total number of image patches. Each column of $\mathbf{P}$ is a single patch reshaped to form a vector. Compute eigenvectors of the matrix $\mathbf{P}\mathbf{P}^T$, and the eigen-coefficients of each noisy patch. Let us denote the j^{th} eigen-coefficient of the i^{th} (noisy) patch (i.e. $\mathbf{P}_i$) by α_{ij}. Define $\bar{\alpha}_j^2 = \max(0, \frac{1}{N}[\sum_{i=1}^{N} \alpha_{ij}^2] - \sigma^2)$, which is basically an estimate of the average squared eigen-coefficients of the 'original (clean) patches'. Now, your task is to manipulate the noisy coefficients $\{\alpha_{ij}\}$ using the following rule, which is along the lines of the Wiener filter update that we studied in class: $\alpha_{ij}^{\text{denoised}} = \dfrac{\alpha_{ij}}{1 + \frac{\sigma^2}{\bar{\alpha}_j^2}}$. Here, $\alpha_{ij}^{\text{denoised}}$ stands for the j^{th} eigencoefficient of the i^{th} denoised patch. Note that $\frac{\sigma^2}{\bar{\alpha}_j^2}$ is an estimate of the ISNR, which we absolutely need for any practical implementation of a Wiener filter update. After updating the coefficients by the Wiener filter rule, you should reconstruct the denoised patches and re-assemble them to produce the final denoised image which we will call 'im2'. Since you chose overlapping patches, there will be multiple values that appear at any pixel. You take care of this situation using simple averaging. Display the final image 'im2' in your report and state its mean squared error with respect to 'im'.

(b) In the second part, you will modify this technique. Given any patch $\mathbf{P}_i$ in the noisy image, you should collect $K = 200$ most similar patches (in a mean-squared error sense) from within a 31×31 neighborhood centered at the top left corner of $\mathbf{P}_i$. We will call this set of similar patches as Q_i (this set will of course include $\mathbf{P}_i$). Build an eigen-space given Q_i and denoise the eigen-coefficients corresponding to **only** P_i using the Wiener update mentioned earlier. The only change will be that $\bar{\alpha}_j^2$ will now be defined using only the patches from Q_i (as opposed to patches from all over the image). Reconstruct the denoised version of P_i. Repeat this for every patch from the noisy image (i.e. create a fresh eigen-space each time). At any pixel, there will be multiple values due to overlapping patches - simply average them. Reconstruct the final denoised image, display it in your report and state the mean squared error value.

(c) Now run your bilateral filter code from Homework 2 on the noisy version of the barbara image. Compare the denoised result with the result of the previous two steps. What differences do you observe? What, in your opinion, is the most important difference between this PCA based approach and the bilateral filter? [20 + 20 + 10 = 50 points]

2. Suppose you are standing in a well-illuminated room with a large window, and you take a picture of the scene outside. The window undesirably acts as a semi-reflecting surface, and hence the picture will contain a reflection of the scene inside the room, besides the scene outside. While solutions exist for separating the two components from a single picture, here you will look at a simpler-to-solve version of this problem where you would take two pictures. The first picture g_1 is taken by adjusting your camera lens so that the scene outside (f_1) is in focus (we will assume that the scene outside has negligible depth variation when compared to the distance from the camera, and so it makes sense to say that the entire scene outside is in focus), and the reflection off the window surface (f_2) will now be defocussed or blurred. This can be written as $g_1 = f_1 + h_2 * f_2$ where h_2 stands for the blur kernel that acted on f_2. The second picture g_2 is taken by focusing the camera onto the surface of the window, with the scene outside being defocussed. This can be written as $g_2 = h_1 * f_1 + f_2$. Given g_1 and g_2, and assuming h_1 and h_2 are known, your task is to derive a formula to determine f_1 and f_2. Note that we are making the simplifying assumption that there was no relative motion between the camera and the scene outside while the two pictures were being acquired, and that there were no changes whatsoever to the scene outside or inside. Even with all these assumptions, you will notice something inherently problematic about the formula you will derive. What is it? What do you think the effect of noise (in g_1 and g_2) will be on the accuracy of your solution (be careful)? [10 + 10 + 5 = 25 points]

3. Consider a matrix $\mathbf{A}$ of size $m \times n$. Define $\mathbf{P} = \mathbf{A}^T\mathbf{A}$ and $\mathbf{Q} = \mathbf{A}\mathbf{A}^T$.

(a) Prove that for any vector $\mathbf{y}$ with appropriate number of elements, we have $\mathbf{y}^t\mathbf{P}\mathbf{y} \geq 0$. Similarly show that $\mathbf{z}^t\mathbf{Q}\mathbf{z} \geq 0$ for a vector $\mathbf{z}$ with appropriate number of elements. Why are the eigenvalues of $\mathbf{P}$ and $\mathbf{Q}$ non-negative?

(b) If $\mathbf{u}$ is an eigenvector of $\mathbf{P}$ with eigenvalue λ, show that $\mathbf{A}\mathbf{u}$ is an eigenvector of $\mathbf{Q}$ with eigenvalue λ. If $\mathbf{v}$ is an eigenvector of $\mathbf{Q}$ with eigenvalue μ, show that $\mathbf{A}^T\mathbf{v}$ is an eigenvector of $\mathbf{P}$ with eigenvalue μ. What will be the number of elements in $\mathbf{u}$ and $\mathbf{v}$?

(c) If $\mathbf{v}_i$ is an eigenvector of $\mathbf{Q}$ and we define $\mathbf{u}_i = \dfrac{\mathbf{A}^T\mathbf{v}_i}{\|\mathbf{A}^T\mathbf{v}_i\|}$. Then prove that there will exist some real, non-negative γ_i such that $\mathbf{A}\mathbf{u}_i = \gamma_i\mathbf{v}_i$.

(d) It can be shown that $\mathbf{u}_i^T\mathbf{u}_j = 0$ for $i \neq j$ and likewise $\mathbf{v}_i^T\mathbf{v}_j = 0$ for $i \neq j$ for correspondingly distinct eigenvalues.[1]. Now, define $\mathbf{U} = [\mathbf{v}_1|\mathbf{v}_2|\mathbf{v}_3|...|\mathbf{v}_m]$ and $\mathbf{V} = [\mathbf{u}_1|\mathbf{u}_2|\mathbf{u}_3|...|\mathbf{u}_n]$. Now show that $\mathbf{A} = \mathbf{U}\mathbf{\Gamma}\mathbf{V}^T$ where $\mathbf{\Gamma}$ is a diagonal matrix containing the non-negative values $\gamma_1, \gamma_2, ..., \gamma_n$. With this, you have just established the existence of the singular value decomposition of any matrix $\mathbf{A}$. This is a key result in linear algebra and it is widely used in image processing, computer vision, computer graphics, statistics, machine learning, numerical analysis, natural language processing and data mining. [5 + 5 + 5 + 5 = 20 points]

[1]This follows because $\mathbf{P}$ and $\mathbf{Q}$ are symmetric matrices. Consider $\mathbf{P}\mathbf{u}_1 = \lambda_1\mathbf{u}_1$ and $\mathbf{P}\mathbf{u}_2 = \lambda_2\mathbf{u}_2$. Then $\mathbf{u}_2^T\mathbf{P}\mathbf{u}_1 = \lambda_1\mathbf{u}_2^T\mathbf{u}_1$. But $\mathbf{u}_2^T\mathbf{P}\mathbf{u}_1$ also equal to $(\mathbf{P}^T\mathbf{u}_2)^T\mathbf{u}_1 = (\mathbf{P}\mathbf{u}_2)^T\mathbf{u}_1 = (\lambda_2\mathbf{u}_2)^T\mathbf{u}_1 = \lambda_2\mathbf{u}_2^T\mathbf{u}_1$. Hence $\lambda_2\mathbf{u}_2^T\mathbf{u}_1 = \lambda_1\mathbf{u}_2^T\mathbf{u}_1$. Since $\lambda_2 \neq \lambda_1$, we must have $\mathbf{u}_2^T\mathbf{u}_1 = 0$.