Assignment 5: CS 663

Due: 7th November before 11:55 pm

Remember the honor code while submitting this (and every other) assignment. You may discuss broad ideas with other students or ask me for any difficulties, but the code you implement and the answers you write must be your own. We will adopt a zero-tolerance policy against any violation.

Submission instructions: Follow the instructions for the submission format and the naming convention of your files from the submission guidelines file in the homework folder. Please see assignment5_DFT.rar. Upload the file on moodle before 11:55 pm on 7th November. We will not penalize any submissions till 10 am on 8th November. But after that, no submissions will be accepted. Only one student per group needs to upload the assignment. No late assignments will be accepted after this time. Please preserve a copy of all your work until the end of the semester.

1. Read Section 1 of the paper 'An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration' published in the IEEE Transactions on Image Processing in August 1996. A copy of this paper is available in the homework folder.

   (a) Describe the procedure in the paper to determine translation between two given images. What is the time complexity of this procedure to predict translation if the images were of size $N \times N$? How does it compare with the time complexity of pixel-wise image comparison procedure for predicting the translation?

   (b) Also, briefly explain the approach for correcting for rotation between two images, as proposed in this paper in Section II. Write down an equation or two to illustrate your point.

   [10+10=20 points]
   Solution:
   The translation is computed as follows: Let $f_2(x, y) = f_1(x - x_0, y - y_0)$ where (x_0, y_0) is the pixel-space shift. Then, we have $F_2(\mu, \nu) = F_1(\mu, \nu)e^{-j2\pi(ux_0+vy_0)/N}$ where images f_2, f_1 have size $N \times N$. The paper proposes to compute the cross power spectrum of the two images given as:

$$C(\mu, \nu) = \frac{F_2^*(\mu, \nu)F_1(\mu, \nu)}{|F_2(\mu, \nu)||F_1(\mu, \nu)|} = e^{-j2\pi(ux_0+vy_0)/N}. \tag{1}$$

   The inverse fourier transform of C will yield a peak at (x_0, y_0). The student may also only refer to equations in the paper for the answer.
   For the part on rotation, consider equation 4 of the paper and also equation 5 which is obtained by applying the rotation theorem to both sides of equation 4. The value of (x_0, y_0) can be obtained using the cross-power spectrum. To find the angle of rotation, consider equation 6 which is obtained by considering the magnitude of the Fourier transform on both sides of equation 5 and converting Cartesian coordinates to polar coordinates. Rotation in Cartesian coordinates is equivalent to a shift in the angle θ in the polar coordinates for which the cross-power spectrum method can again be used.

   For the part on translation, the time complexity of this procedure is $O(N^2 \log N)$ for an $N \times N$ image. A pixel-wise translation prediction will have time complexity $O(N^2W^2)$ where $W \times W$ is the window size for the range of translations.

   Marking Scheme: Description of procedure to determine translation: 7 points, time complexity for translation: 3 points. Procedure for rotation: 7 points (application of rotation theorem: 5 points, determining (x_0, y_0): 2.5 points, taking magnitudes: 2.5 points, conversion to polar coordinates: 3 points and 2 points for final application of cross-power spectrum.)

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 $g_2 = h_1 * f_1 + f_2$ where h_1 is the blur kernel acting on f_1. 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/unstable about the formula you will derive. What is it? [8+7 = 15 points]
ANSWER:
See model code ReflectionSeparation.m in the solutions folder, though no code was expected from you for this question. Taking Fourier Transforms, we have $G_1(\mu) = F_1(\mu) + H_2(\mu)F_2(\mu)$ and $G_2(\mu) = H_1(\mu)F_1(\mu) + F_2(\mu)$. Solving these equations simultaneously for $F_1(\mu)$ and $F_2(\mu)$, we get $F_2(\mu) = \frac{G_2(\mu)-H_1(\mu)G_1(\mu)}{1-H_1(\mu)H_2(\mu)}$ and $F_1(\mu) = \frac{G_1(\mu)-H_2(\mu)G_2(\mu)}{1-H_1(\mu)H_2(\mu)}$. This solution is well-defined for all values of μ, except if $H_1(\mu)H_2(\mu) = 1$. Now, remember that h_1 and h_2 are low-pass filter kernels, due to the defocussing of the image. These blur kernels will always integrate to 1, i.e. $\int_{-\infty}^{+\infty} h_1(x)dx = 1$ or $\int_a^b h_1(x)dx = 1$ for a blur kernel defined on the interval $[a, b]$. Therefore, we can conclude that $H_1(0) = H_2(0) = 1$ (why? Look at the Fourier transform formula and plug in $\mu = 0$). For all other values of μ, we will have $|H_1(\mu)| \leq 1$ and $|H_2(\mu)| \leq 1$ as this is a low pass filter. This therefore means, that our solution is undefined for those low frequencies, where $H_1(\mu)H_2(\mu) = 1$ (such as $\mu = 0$). Here is a (somewhat rare) situation where reconstruction of higher frequency components is fine, but where lower frequencies (especially the DC component) cannot be recovered robustly. In the case of image denoising, the situation is exactly opposite - the lower frequencies are easy to reconstruct, and hence smooth regions look fine. But the higher frequencies are difficult to reconstruct and hence the loss of finer edges and textures in image denoising. In practice, we would need to add in a small ϵ to the denominators, i.e. we would have $F_2(\mu) = \frac{G_2(\mu)-H_1(\mu)G_1(\mu)}{1-H_1(\mu)H_2(\mu)+\epsilon}$ and $F_1(\mu) = \frac{G_1(\mu)-H_2(\mu)G_2(\mu)}{1-H_1(\mu)H_2(\mu)+\epsilon}$. The resultant image would look somewhat artificial due to incorrect reconstruction of lower frequencies, such as what you see below in Figure 1.

Now, if there is image noise, i.e. $g_1 = f_1 + h_2 * f_2 + \eta_1$ and $g_2 = h_1 * f_1 + f_2 + \eta_2$, we incur errors proportional to $\frac{N_1(\mu)-H2(\mu)N_2(\mu)}{1-H_1(\mu)H_2(\mu)}$ and $\frac{N_2(\mu)-H_1(\mu)N_1(\mu)}{1-H_1(\mu)H_2(\mu)}$. For higher frequencies, the denominator is large (i.e. close to 1) and hence there is no amplification of the noise, unlike the case with the inverse filter under noise. For lower frequencies, the noise will get amplified especially as $H_1(\mu)H_2(\mu)$ gets closer to 1, but the signal strength is also very high in these frequencies, so the relative error will not totally blow off, unlike the case with the inverse low-pass filter under noise. See below a reconstruction result under noise in Figure 2.

Also take note that the blurring of the images was actually useful over here (somewhat counter-intuitively). Without the blur, we would simply have two identical images and the separation would have been impossible.

MARKING SCHEME: 8 points for correct derivation of formula. 7 points for pointing out what is the problem with the solution. While describing the problem with this approach, you must argue that $H_1(0) = H_2(0) = 1$ and hence the lower frequency components (particularly the DC component) are not properly reconstructed. Merely saying that there is a division by 0 when $H_1(\mu) = H_2(\mu) = 1$ will fetch you only 3 points out of 7. No code is expected for this question.

3. This is a fun exercise where you are officially allowed to do a google search and find out a research paper which works on an image restoration problem which is different from all the ones we have seen in class. In your report, you should clearly state the problem, write the title, venue and publication year of the research paper, and mention the cost function that is optimized in the research paper in order to solve the problem. In the cost function, you should mention the meaning of all variables. For your reference, here is a list of restoration problems we have encountered in class: image denoising, image deblurring, image inpainting, reflection removal, stitching together images of torn pieces of paper (you saw this one in the midsem), notch filters for removal of

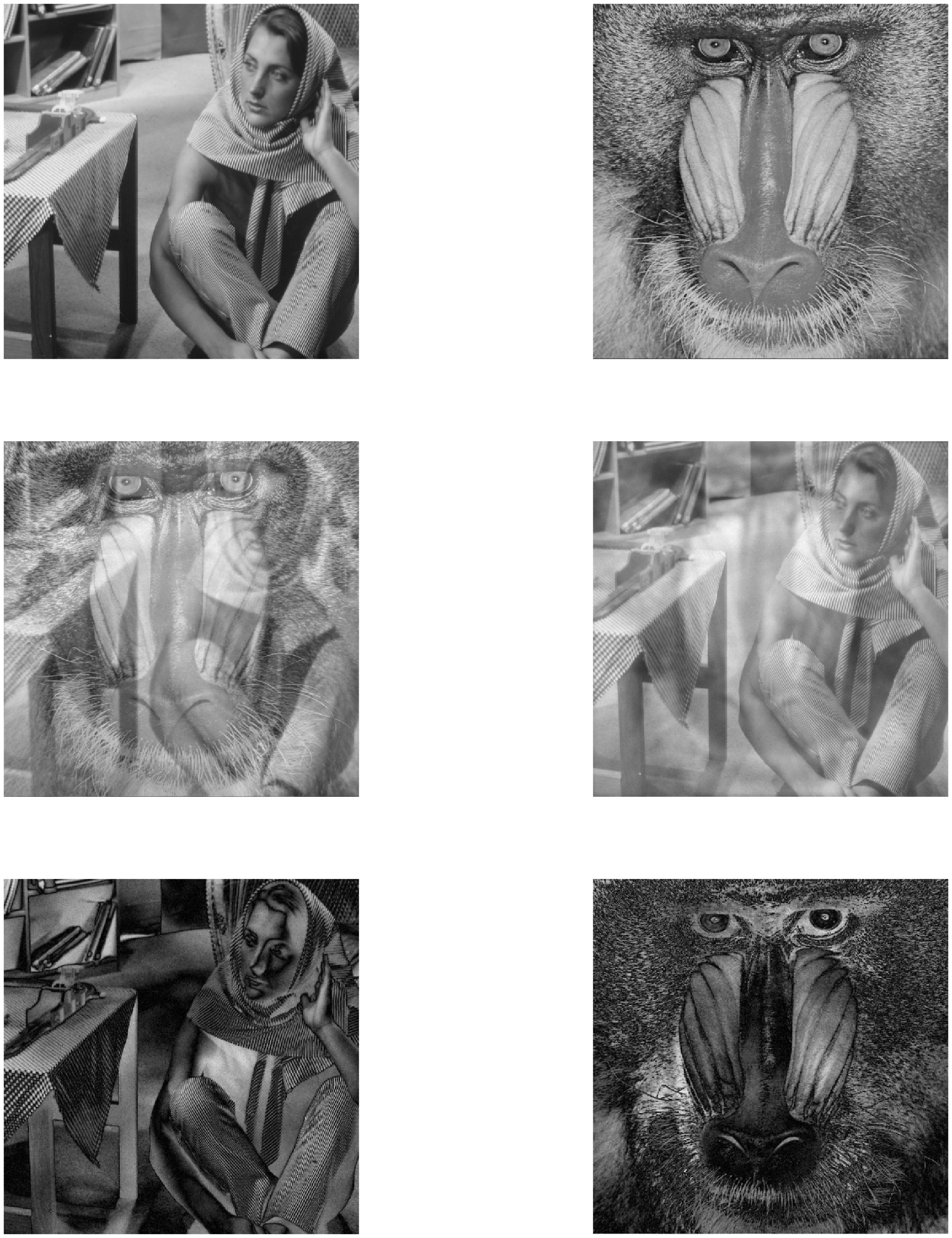

Figure 1: Left to right, top to bottom: barbara, mandrill, mixture 1, mixture 2, reconstructed barbara and reconstructed mandrill. Notice that the features of barbara and mandrill are correctly reconstructed. But the image looks as if it were sent through a high pass filter - which is due to error in the reconstruction of the DC component.

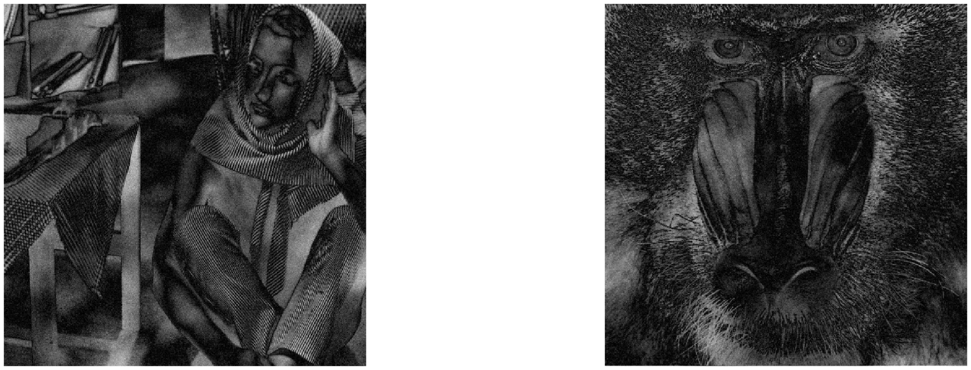

Figure 2: Left to right: reconstructed barbara and reconstructed mandrill when the mixtures had noise. Notice that the features of barbara and mandrill are correctly reconstructed. The reconstructions are not totally different from the earlier results despite the noise. The errors in the DC component, however, remain as before.

periodic interference patterns in images. You are not allowed to mention any of these. [15 points]
Answer: Anyone problem apart from these is allowed! The one I chose is paper decrumpling: Consider a crumpled piece of paper that is spread out to form an approximately rectangular sheet under a flat-bed scanner. The paper is scanned, and the image contains a large number of seam and shadow artifacts. Removing these artifacts is a challenging problem, which remains unsolved to my best knowledge. One approach is do something called 'shape from shading', i.e. derive the 3D shape of the paper from its 2D images (see the wiki article on 'Shape from shading'), and then find a transformation that maps the close-to-rectangular shape in 3D to a flat 2D rectangle. The closest I have seen is the paper 'Intrinsic Decomposition of Document Images In-the-Wild' published in the British Machine Vision Conference in 2020. The paper can be found at `https://www.bmvc2020-conference.com/assets/papers/0906.pdf`. See figure 5 of the paper for some sample results.
Marking scheme: No marks if the paper deals with one of the problems mentioned in the question. 4 marks for the problem statement, 4 marks for the paper title and venue, and 7 marks for the cost function with all symbols explained. No marks for the cost function if most of the symbols are not defined in the writeup.

4. Consider a $n \times n$ image $f(x, y)$ such that only $k \ll n^2$ elements in it are non-zero, where k is known and the locations of the non-zero elements are also known. (a) How will you reconstruct such an image from a set of only m different Discrete Fourier Transform (DFT) coefficients of known frequencies, where $m < n^2$? (b) What is the minimum value of m that your method will allow? (c) Will your method work if k is known, but the locations of the non-zero elements are unknown? Why (not)? [10+5+5 = 20 points]
ANSWER: Part (a): Let $\boldsymbol{f_T}$ be a vector containing the k (unknown) non-zero values of image f, where T is the known support-set (locations of non-zero elements) of f where $|T| = k$. Let $\boldsymbol{y}$ be a vector of m measurements. Then we have $\boldsymbol{y} = \boldsymbol{U}\boldsymbol{f_T}$ where $\boldsymbol{U}$ is a $m \times k$ matrix, where $U_{ij} = e^{-\iota 2\pi(u_i x_j + v_i y_j)/n}$. Here $\iota = \sqrt{-1}$, i is an index for the frequency (u_i, v_i) for the DFT coefficient y_i, and (x_j, y_j) is a spatial index belonging to the set T. Now, you can estimate $\boldsymbol{f_T}$ by pseudo-inverse, provided $m \geq k$. So the minimum value of m must be at least k. This settles part (b).
Now if the support-set is unknown, then one computationally very expensive method is to enumerate all k-size subsets of the $n \times n$ set of indices, and compute a separate pseudo-inverse for each. The trouble is that there is *prima-facie* no guarantee that all these solutions will be equal to each other, and it is not clear which one to pick. If you pointed this out, then I am giving you full points.
It turns out that if $m \geq 2k$, and the matrix $\boldsymbol{U}$ has the property that no k-sparse vector lies in its null-space, then a unique solution is guaranteed. This forms the basis of the theory of something called "compressed sensing" – which is essentially a method of estimating n unknowns from $m < n$ linear equations provided most of the n unknowns are zero (even though you are a priori not told which of the n unknowns are zero).

5. In this exercise, we will study a nice application of the SVD which is widely used in computer vision, computer graphics and image processing. Consider we have a set of points $\boldsymbol{P}_1 \in \mathbb{R}^{2\times N}$ and another set of points

$\boldsymbol{P}_2 \in \mathbb{R}^{2\times N}$ such that $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are related by an orthonormal transformation $\boldsymbol{R}$ such that $\boldsymbol{P}_1 = \boldsymbol{R}\boldsymbol{P}_2 + \boldsymbol{E}$ where $\boldsymbol{E} \in \mathbb{R}^{2\times N}$ is an error (or noise) matrix. The aim is to find $\boldsymbol{R}$ given $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ imposing the constraint that $\boldsymbol{R}$ is orthonormal. Answer the following questions: [30 points = 3 + 3 + 3 + 3 + 3 + (8 + 4 + 3)]

(a) The standard least squares solution given by $\boldsymbol{R} = \boldsymbol{P_1}\boldsymbol{P_2}^T(\boldsymbol{P_2}\boldsymbol{P_2}^T)^{-1}$ will fail. Why? Because it does not enforce the obtained R to be orthonormal.

(b) To solve for $\boldsymbol{R}$ incorporating the important constraints, we seek to minimize the following quantity:

$$
\begin{aligned}
E(\boldsymbol{R}) &= \|\boldsymbol{P_1} - \boldsymbol{R}\boldsymbol{P_2}\|_F^2 && (2)\\
&= \text{trace}((\boldsymbol{P_1} - \boldsymbol{R}\boldsymbol{P_2})^T(\boldsymbol{P_1} - \boldsymbol{R}\boldsymbol{P_2})) && (3)\\
&= \text{trace}(\boldsymbol{P_1^T}\boldsymbol{P_1} + \boldsymbol{P_2^T}\boldsymbol{R^T}\boldsymbol{R}\boldsymbol{P_2} - \boldsymbol{P_2^T}\boldsymbol{R^T}\boldsymbol{P_1} - \boldsymbol{P_1^T}\boldsymbol{R}\boldsymbol{P_2}) && (4)\\
&= \text{trace}(\boldsymbol{P_1^T}\boldsymbol{P_1} + \boldsymbol{P_2^T}\boldsymbol{P_2} - \boldsymbol{P_2^T}\boldsymbol{R^T}\boldsymbol{P_1} - \boldsymbol{P_1^T}\boldsymbol{R}\boldsymbol{P_2}) \text{ justify} && (5)\\
&= \text{trace}(\boldsymbol{P_1^T}\boldsymbol{P_1} + \boldsymbol{P_2^T}\boldsymbol{P_2}) - 2\text{trace}(\boldsymbol{P_1^T}\boldsymbol{R}\boldsymbol{P_2}) \text{ justify} && (6)
\end{aligned}
$$

For the first part: (using the orthonormality of R). For the second part: (using the fact that $trace(AB) = trace(BA)$ for any two matrices A,B of compatible dimensions).

(c) Why is minimizing $E(\boldsymbol{R})$ w.r.t. $\boldsymbol{R}$ is equivalent to maximizing trace$(\boldsymbol{P_1^T}\boldsymbol{R}\boldsymbol{P_2})$ w.r.t. $\boldsymbol{R}$? This is because $E(R) = \boldsymbol{P_1^T}\boldsymbol{P_1} + \boldsymbol{P_2^T}\boldsymbol{P_2}) - 2\text{trace}(\boldsymbol{P_1^T}\boldsymbol{R}\boldsymbol{P_2})$. The first two terms on the RHS do not depend on R and the last term contains a minus sign. Hence minimizing $E(R)$ is equivalent to maximizing trace$(\boldsymbol{P_1^T}\boldsymbol{R}\boldsymbol{P_2})$.

(d) Now, we have

$$
\begin{aligned}
\text{trace}(\boldsymbol{P_1^T}\boldsymbol{R}\boldsymbol{P_2}) &= \text{trace}(\boldsymbol{R}\boldsymbol{P_2}\boldsymbol{P_1^T}) \text{ (justify this step)} && (7)\\
&= \text{trace}(\boldsymbol{R}\boldsymbol{U'}\boldsymbol{S'}\boldsymbol{V'^T}) \text{ using SVD of } \boldsymbol{P_2}\boldsymbol{P_1^T} = \boldsymbol{U'}\boldsymbol{S'}\boldsymbol{V'^T} && (8)\\
&= \text{trace}(\boldsymbol{S'}\boldsymbol{V'^T}\boldsymbol{R}\boldsymbol{U'}) = \text{trace}(\boldsymbol{S'}\boldsymbol{X}) \text{ where } \boldsymbol{X} = \boldsymbol{V'^T}\boldsymbol{R}\boldsymbol{U'} && (9)\\
& && (10)
\end{aligned}
$$

For the first part: (using the fact that $trace(AB) = trace(BA)$ for any two matrices A,B of compatible dimensions).

(e) For what matrix $\boldsymbol{X}$ will the above expression be maximized? (Note that $\boldsymbol{S'}$ is a diagonal matrix.) The way it is defined, X is an orthonormal matrix. As S' is a diagonal matrix of singular values, the diagonal values are all non-negative. Note that $trace(S'X) = S'_{11}X_{11} + S'_{22}X_{22} + S'_{33}X_{33}$. The trace will be maximized when X equals the identity matrix, because then the elements X_{11}, X_{22}, X_{33} will be maximum in value.)

(f) Given this $\boldsymbol{X}$, how will you determine $\boldsymbol{R}$? We have $X = I$ and hence $V'^T RU' = I$, i.e. $R = V'U'^T$.

(g) If you had to impose the constraint that $\boldsymbol{R}$ is specifically a rotation matrix, what additional constraint would you need to impose? You need to impose the constraint that $det(R) = +1$. For an orthonormal R, the determinant is either +1 or -1, but for a rotation matrix, it must be +1.