Assignment 3
Old But Gold: Fourier Transform Applications in Image Processing

Homeworks Guidelines and Policies

- ***What you must hand in.*** It is expected that the students submit an assignment report (HW3_[student_id].pdf) as well as required source codes (.m or .py) into an archive file (HW3_[student_id].zip).
- ***Pay attention to problem types.*** Some problems are required to be solved *by hand* (shown by the icon), and some need to be implemented (shown by the icon). Please don't use implementation tools when it is asked to solve the problem by hand, otherwise you'll be penalized and lose some points.
- ***Don't bother typing!*** You are free to solve by-hand problems on a paper and include picture of them in your report. Here, cleanness and readability are of high importance. Images should also have appropriate quality.
- ***Reports are critical.*** Your work will be evaluated mostly by the quality of your report. Don't forget to explain what you have done, and provide enough discussions when it's needed.
- ***Appearance matters!*** In each homework, 5 points (out of a possible 100) belongs to compactness, expressiveness and neatness of your report and codes.
- ***Python is also allowable.*** By default, we assume you implement your codes in MATLAB. If you're using Python, you have to use equivalent functions when it is asked to use specific MATLAB functions.
- ***Be neat and tidy!*** Your codes must be separated for each question, and for each part. For example, you have to create a separate .m file for part b. of question 3. Please name it like p3b.m.
- ***Use bonus points to improve your score.*** Problems with bonus points are marked by the icon. These problems usually include uncovered related topics or those that are only mentioned briefly in the class.
- ***Moodle access is essential.*** Make sure you have access to Moodle because that's where all assignments as well as course announcements are posted on. Homework submissions are also done through Moodle.

- ***Assignment Deadline.*** Please submit your work **before the end of June 8th**.
- ***Delay policy.*** During the semester, students are given 7 free late days which they can use them in their own ways. Afterwards there will be a 25% penalty for every late day, and no more than three late days will be accepted.
- ***Collaboration policy.*** We encourage students to work together, share their findings and utilize all the resources available. However you are not allowed to share codes/answers or use works from the past semesters. Violators will receive a zero for that particular problem.
- ***Any questions?*** If there is any question, please don't hesitate to contact me through the following email address: **ali.the.special@gmail.com**.

1. Fundamentals of 1-D Fourier Transform (12 Pts.)

Keywords: *Fourier Transform, Inverse Fourier Transform, Duality/Linearity/Time Shift/Convolution Property of Fourier Transform, Dirac Delta Function*

The **Fourier Transform** is arguably one of the greatest insights ever made in the history of science. Theoretically, everything in the world can be represented via a waveform, i.e. a function of time, space or some other variables. The Fourier transform gives us a unique and powerful way of viewing these waveforms, by breaking them into an alternate representation characterized by sine and cosines.

The goal of this problem is for you to practice the theoretical background of the Fourier transform, before facing more challenging image-related applications in this area.

First, calculate and determine the Fourier transform of the following functions using properties of the Fourier transform and the definitions related to it:

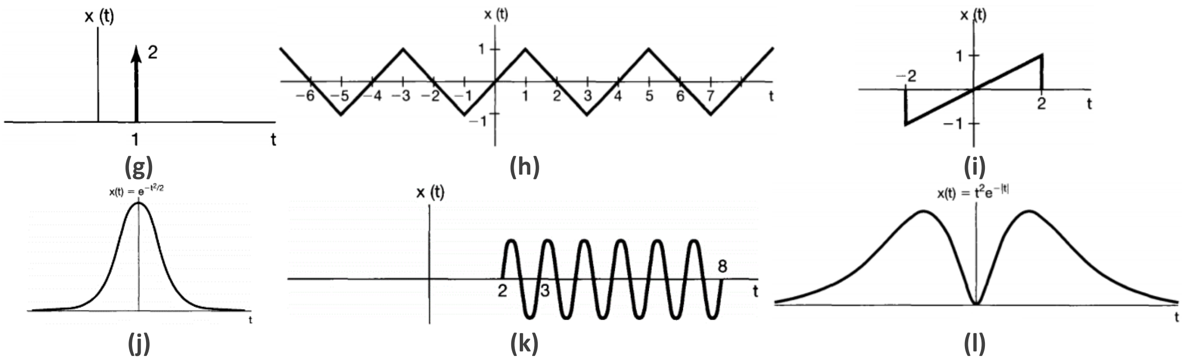
a. $g(t) = \sin(t-2)$ b. $g(t) = \dfrac{2t^2+1}{t^2+1}$ c. $g(t) = \delta(t-1) + \delta(t+2)$

d. $g(t) = \mathcal{F}^{-1}\left\{\dfrac{1}{f^2+1}\right\}$ e. $g(t) = \cos t e^{-|t|}$ f. $g(t) = \cos(4\pi t)\cos(6\pi t)$

Next, specify which – if any – of the real signals shown in the figure have Fourier transforms that meet the following conditions:

g. $\mathfrak{Re}\left\{X(j\omega)\right\} = 0$ h. $\mathfrak{Im}\left\{X(j\omega)\right\} = 0$ i. $\int_{-\infty}^{\infty} X(j\omega) d\omega = 0$

j. $\int_{-\infty}^{\infty} \omega X(j\omega) d\omega = 0$ k. $X(j\omega)$ is periodic. l. There is a real α such that $e^{j\alpha\omega}X(j\omega)$ is real.



(g) (h) (i) (j) (k) (l)

Finally, let $f = [1, 3, -1, 2, 0, -3]$ and $h = [-1, 3, -2]$.

m. Pad both f and h with zeros to a length of 12 and find the convolution $f * g$.
n. Find the convolution by calculating a product of discrete Fourier transforms.
o. Plot $F_p(u)$, $G_p(u)$ and $H_p(u)$ as points in the complex plane for $0 \le u \le 11$.
p. Calculate and interpret $\sum g_p^2(n)$ and $\sum \left|G_p(u)\right|^2$.

2. The Mysterious Moiré Pattern (12 Pts.)

Keywords: *Frequency Domain, Image Filtering, Notch Filter, Halftone Image, Moiré Effect, Color Spaces*

If you happen to be in the town of Abertillery in Wales, make sure to pay a visit to *The Guardian* sculpture. It's a 20m tall statue made from over 20,000 horizontal strips of a special steel. These pieces are connected in a specific fashion so that the sculpture appears almost transparent from a distance, while from a closer range it is seen as a solid structure. The unique style of the sculpture creates a *Moiré effect*, where the steel sections appear to move or shimmer in the light.

(a) **(b)**

Figure 1 The Guardian sculpture is a special structure with unique properties. It is often used as a well-known example of Moiré pattern (a) View from a distance (b) Moiré pattern can be seen on hand and forearm

In the world of imaging, Moiré effect appears as an undesirable artefact when two or more *halftones* conflict. Halftone is referred to a printed image of dots, varying in size, shape or spacing, which from a certain distance appears smooth and forms shapes and objects. That's why some documents extracted from old books and newspapers look dotted.

We now want to use our knowledge in the frequency domain image filtering to treat with these two undesirable effects.

I. Moiré pattern. In the area of medical imaging, *Moiré pattern removal* is an active research topic since this artefact is quite prevalent in the images. Here, two images of this type are given.

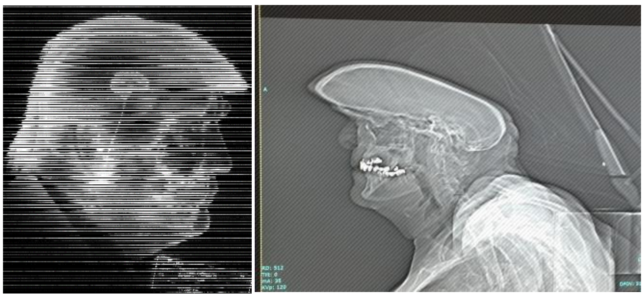


Figure 2 Due to the limitations of the medical imaging instruments, many radiographic images suffer from Moiré pattern

a. Display the images spectrum in the frequency domain. Give your interpretations of the resultant spectrum images.
b. Design appropriate filters and apply them on the images to reduce the effect of Moiré pattern. Display the results as well as the filters used.

II. Halftone effect. Because of the advances made in printing devices, halftone effect is no more a worrying artefact in the area of imaging. However, old printed images which can often be found in newspapers and books may still look dotted, and reducing this effect must be taken into consideration when digitizing these photos. You are provided with two examples of these images, and the goal is to restore the original color images without dots.

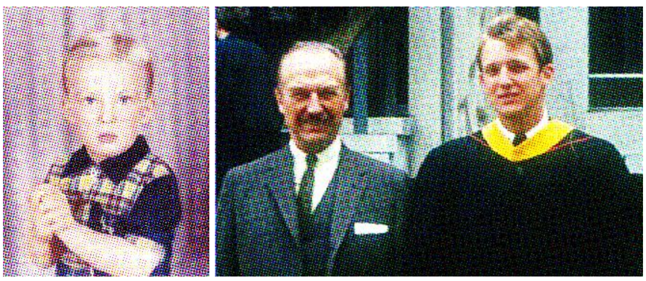


Figure 3 These color images are formed by dotted points, which may not be clearly seen without zooming

c. Implement a method in spatial domain to reduce the halftone effect.
d. Now convert the given RGB images to different color spaces, namely *HSV*, *YCbCr* and *Lab*. Display the Fourier transform of each channel (R, G, B, H, S, V, Y, Cb, Cr, L, a and b) and determine which color space the Fourier transform is more informative in.
e. Use the color space determined in the previous part, and design suitable filters for each channel separately to reduce the halftone effect. Combine the filtered channels to obtain filtered color results. Display the resultant images as well as the filters.
f. Compare the results obtained in part (e) with part (c).

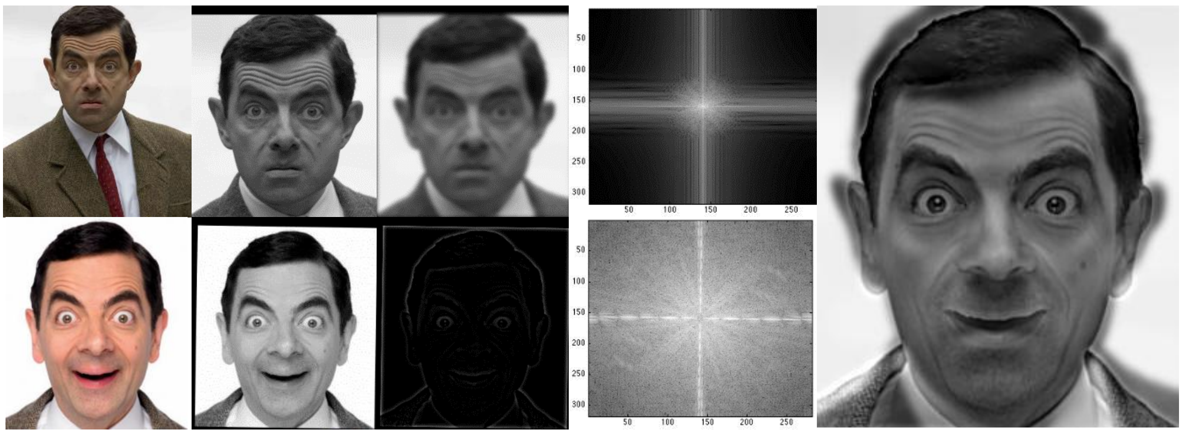
3. Implementing the *Bittersweet emotion* in Photos! (15 Pts.)

Keywords: *Image Filtering, Frequency Domain, Bandpass Filters, Hybrid Images, Image Alignment*

Have you ever felt happy and sad at the same time? This, in fact, is a normal feeling known as the *Bittersweet emotion*, which is experienced quite often in life. Just remember the last time you accidentally came across your childhood toys and started feeling nostalgic about your old days.

Figure 4 The woman appears to be sad in the large image, while her smile gradually appears as the size grows smaller. You can also blur your vision if you wish to see her mood changes.

We aim to bring this special feeling into images by using fundamentals of **Image Filtering** in **Fourier Domain**. As you know, high frequency details of an image tend to be more visible from close range, whereas low frequency signal is more noticeable from a distance. We intuitively make use of this fact by merging high frequency content of one image with the low frequency content of another to obtain a combined image whose interpretation depends on the distance where it is being seen. More details can be found in [this paper](#).



(a) **(b)** **(c)** **(d)** **(e)**

Figure 5 The process of creating a hybrid image (a) Initial face images (b) Images are aligned so that they match visually (c) The result of applying band-pass filters on the images (d) Fourier spectrum of the filtered images. As can be seen, only low frequencies are preserved in the upper spectrum, while the opposite is true in the lower one (e) Resultant hybrid image

You are given several frames taken from Donald Trump and Joe Biden in their final presidential debate. For each of these guys, choose two suitable images according to your preference. You are required to perform the following steps on each set of images.

a. **Image alignment.** A function (`align_imgs.m` for MATLAB and `align_imgs.py` for Python) is provided for you, which takes two images and two pairs of points, and align them so that the two pairs of points will have approximately equal coordinates. Use this function to align the input images. Display the resultant aligned images and their amplitudes of the Fourier transform.
Note: The Python function is not tested, hence it is not guaranteed to work flawlessly.
b. **Image filtering.** Apply low-pass filter on the first image using a standard Gaussian filter, and high-pass filter on the other by subtracting the image filtered with Gaussian filter from the original one. Choose proper values for cut-off frequencies. Display the results as well as their logarithmic amplitude of the Fourier transform.
c. **Merge Images.** Merge the images you obtained in the previous part, and display the final image and the corresponding amplitude of the Fourier transform.
d. **Visualisation.** Apply a Gaussian filter with five increasing cut-off values on the resultant images in order to illustrate the process of transformation of one expression into another.

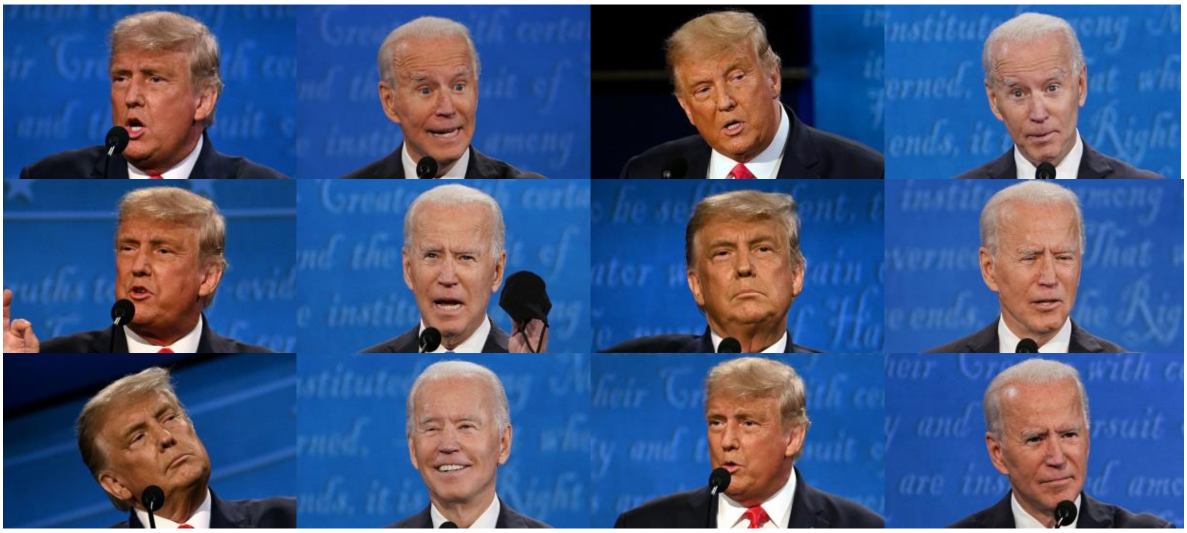


Figure 6 Different facial expressions were given by the candidates during the final debate of 2020 United States presidential election. Two images for each person must be selected as the input images of the algorithm.

4. Who is Who: Face Recognition in the Frequency Domain (15 Pts.)

Keywords: *Frequency Domain, Fourier Transform, Inverse Fourier Transform, Image Filtering, Face Recognition, Template Matching, Minimum Average Correlation Energy (MACE) filter, Cross Correlation*

Those who own an iPhone X (or a newer version) may perceive a **Face Recognition** system as being extremely sophisticated, complicated, high-tech technology which requires a great deal of skills and knowledge to be implemented. While in the case of *robust* face recognition systems (as is the case with Apple Face ID) it is more or less true, here we are going to demonstrate only basic knowledges of images in the **Frequency Domain** is enough for us to implement a sufficiently capable face recognition system.

More precisely, the idea is to use **Template Matching** to design a specific filter, called *minimum average correlation energy* (MACE) filter, and perform a **Cross Correlation** between the test image and this filter.

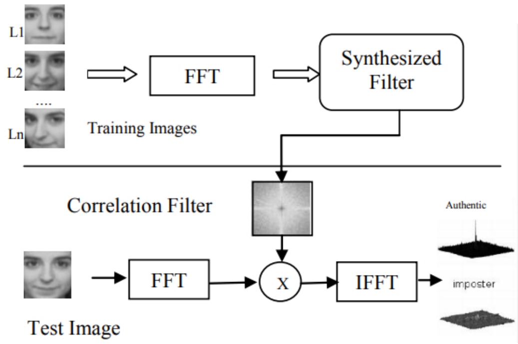


Figure 7 Block diagram of the algorithm. Training images are used to obtain a synthesized filter in the frequency domain. In the testing phase, the filtering output will give high peak value at the origin of the correlation plane if the test face belongs to true class.

The intuition behind designing a MACE filter is to create a filter using a set of training images that would produce correlation output. Through the MACE filter, the average correlation energy over the given training faces will be minimized, hence it produces sharp peaks at the origin of the correlation plane, while producing lower values in the rest of the plane, Figure 8.

Assuming N images of the size n x m with d (= n x m) pixels in the training set, the MACE filter H is given by:

$$H = D^{-1}X\left(X^{*}D^{-1}X\right)^{-1}u$$

where D is a d x d diagonal matrix which contains the average correlation energies of the training images in its diagonal elements, X is a N x d matrix and X* denotes the complex conjugate of X. The i^{th} column of X represents the Discrete Fourier coefficients of the i^{th} training image. The column vector u is also an N x 1 vector which contains the correlation peak values for a series of training images. It has N elements, each corresponding to desired values at the origin of the correlation plane of the training images. These constraint values are often set to one for all training images from the authentic class.

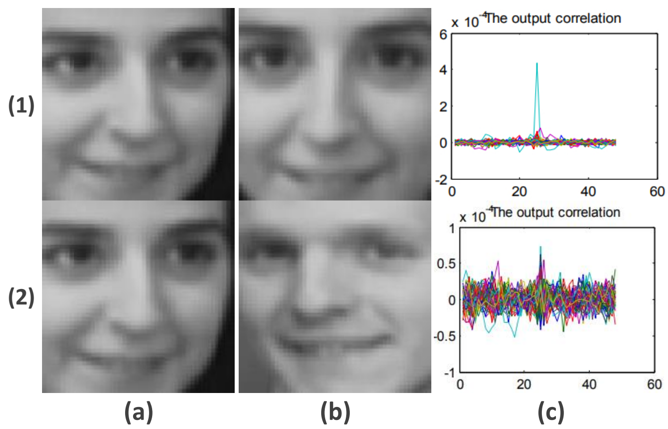


Figure 8 Two test cases, one a 'matched' and the other a 'non-matched' state. As can be seen, correlation coefficients shows a peak when the classes are the same (a) Test image (b) training images (c) The correlation coefficients

The MACE filter optimizes a criterion to produce a desired correlation output plane. Here, we use maximum peak value for measuring correlation plane. The correlation plane with sharp peak value indicates that the test image is matched, while the imposter output denotes that the test image is not matched, Figure 8. Please refer to [this paper](#) for more details about how the algorithm works.

Download the cropped version of [AR face database](#). This subset of the database contains 2600 images of 100 individuals, each with 26 images with different facial expressions, illumination conditions and occlusions, Figure 9. The images must be converted to grayscale before being used in the following parts.

Figure 9 Samples in AR database are varied in their facial expressions, illumination conditions and occlusions

a. Ignore the images with occlusions, and keep the images with postfix 01 to 07 as the training images, and those with postfix 14 to 20 as the test images. Pick a random test image, and perform face recognition using the above algorithm. Display the MACE filter, output correlation and the recognition result.
b. Calculate the recognition rate by dividing the number of true identifications to total number of test images.
c. Now consider the whole dataset, i.e. images with postfix 01 to 13 as the training images, and those with postfix 14 to 26 as the test images. Pick a random test image, and perform face recognition using the above algorithm. Display the MACE filter, output correlation and the recognition result.
d. Calculate the recognition rate, and compare the result with part (b).

5. Stop Moving: Dealing with the Motion Blur (18 Pts.)

Keywords: *Image Deconvolution, Motion Blur, Image Deblurring, Inverse Problem, Ill-posed Problem, Least Squares Solution, Tikhonov Regularization, Closed-form Solution, Laplacian Filter*

Take a look at images in Figure 10. You are now able to convert (a) to (b) and add **Motion Blur** to images using the concept of **Image Convolution** in the **Frequency Domain**. But do you think this process is invertible? If yes, under what circumstances?

(a) **(b)**

Figure 10 A clean image and its corresponding blurry version

The process of converting (b) to (a) is called **Image Deconvolution**, which can be *blind* (when the blurring kernel is unknown) or *non-blind* (If the blurring kernel parameters are known). As expected, in non-blind deconvolution a higher accuracy will be achieved. However, in most real-world applications the blurring kernel is not given, means the problem must be addressed using assumptions about the image contents, known as *a priori* knowledge. So, in summary, if the blurring kernel is available, converting (b) to (a) with an acceptable level of precision is always possible. Let's see this in action.

The degradation process of an image from a high-resolution version x to a blurry version y can be modelled as

$$y = k * x + \varepsilon \qquad (1)$$

where h is the PSF (blur kernel), ε is the additive noise and $*$ denotes the convolution operator. In the frequency domain, it forms a system of linear equations such that

$$\mathbf{y} = \mathbf{H}\mathbf{x} \qquad (2)$$

which is said to be an *ill-posed problem*. Given $\mathbf{y}$ and $\mathbf{H}$, one can seek a solution $\mathbf{x}$ by minimizing the error

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\| \mathbf{y} - \mathbf{H}\mathbf{x} \right\|_2^2 \qquad (3)$$

which can be shown to have a closed-form solution, known as the *least squares solution*

$$\hat{\mathbf{x}}_{LS} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{y} \qquad (4)$$

However, since the solution space in (3) is highly varying, we add a regularization term to make it constrained. The new optimization problem, known as *Tikhonov-regularized least squares*, contains a prior information about the solution and can be written as

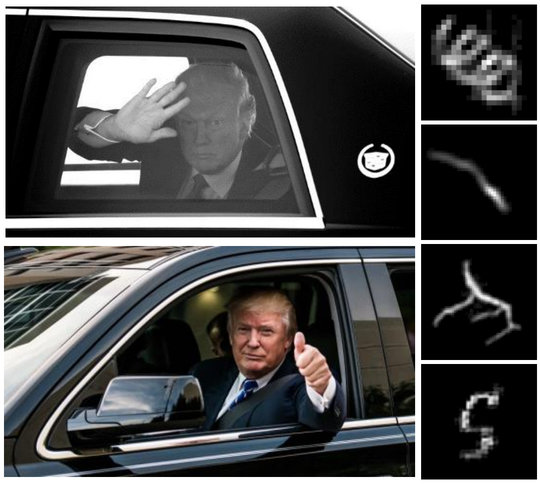
$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda \|\mathbf{A}\mathbf{x}\|_2^2 \qquad (5)$$

with a closed-form solution given by

$$\hat{\mathbf{x}}_{TRLS} = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{H}^T\mathbf{y} \qquad (6)$$

When it comes to image deconvolution, there are various options for the additional term in (5). The simplest way is to just assume $\mathbf{A} = \mathbf{I}$ and define a new optimization problem, known as *ridge regression* in machine learning, in which the prior term dictates $\mathbf{x}$ to have lower values. Another more intuitive option is to consider $\mathbf{A} = \mathbf{D}$, where $\mathbf{D}$ is a Laplacian filter which calculates the second derivative of the image. The idea is to control smoothness/sharpness of edges in the solution using the regularization parameter λ, which often leads to more accurate solutions.

Here, two input images (one in grayscale and the other in color) as well as a set of PSF kernels with different sizes are provided. Please perform the following tasks using the above definitions.



(a) **(b)**

Figure 11 images provided for this problem (a) Input images (grayscale and color) (b) PSF kernels

a. Use the PSF kernels and blur the input images by performing image convolution in the frequency domain. Display the results (8 in total).
b. Use the least-square problem (Equation (4)) and recover the input images using blurred images and their corresponding PSF kernels.
c. Use the least-square problem (Equation (6)) with $\mathbf{A} = \mathbf{I}$, and recover the input images using blurred images and their corresponding PSF kernels. Evaluate the deconvolution result using quality measurements such as PSNR and SSIM.
d. Use the least-square problem (Equation (6)) with $\mathbf{A} = \mathbf{D}$, where $\mathbf{D}$ is a Laplacian operator, and recover the input images using blurred images and their corresponding PSF kernels. Evaluate the deconvolution result using quality measurements such as PSNR and SSIM.
e. Rewrite a new optimization problem similar to (5) to obtain unknown PSF kernel given a clean image as well as its corresponding blurry version. Write down its closed-form solution, and use it to obtain PSF kernels when the original input images are given. Evaluate the results using quality measurements such as RMSE.

Note 1: The regularization parameter λ must be chosen by trial and error.

Note 2: The blurring/deblurring results of the colored input image must also be colored.

Recommended MATLAB Functions: `psf2otf(),otf2psf(),psnr()`

6. Photoshop?! Gaussian Filter is There for You! (18 Pts.)

Keywords: *Image Filtering, Frequency Domain, Fourier Transform, Bandpass Filters, Lowpass Filter, Highpass Filter, Gaussian Filter, Laplacian Filter, Image Blending*

Mixing two images and creating a blended image has always been a popular operation in the area of images. This operation – which is usually done using image editing softwares like Photoshop – generally leads to produce hilarious images which amuse everyone. In this problem, we wish to get familiar with a Fourier transform-based method which blend images even better than Photoshop!

Here's the idea. Imagine we want to mix specific regions of an image – defined by a mask – with another image, so that the result looks as natural as possible. One way to do so, is to simply crop those regions using the provided mask and place them in the destination image. But as expected, it leads to a stark result (Figure 12). To obtain a seamless and gradual result in the boundaries, one can compute a gentle seam between the two images separately at each band of image frequencies, and combine them eventually to get the final result.

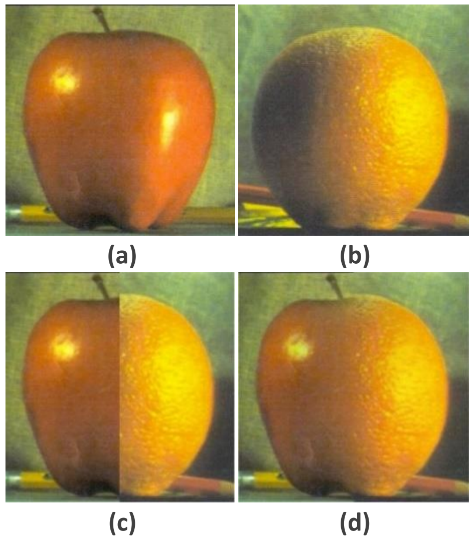


Figure 12 Images blended by the proposed method look very realistic and natural (a) Left pair (b) Right pair (c) Result of the copy-and-paste method (d) Result of the proposed method

More specifically, first both images are convolved with a highpass filter at different levels, each with increasing amount of sharpness. At the same time, mask of desired regions is also convolved with a lowpass filter at different levels, each with increasing amount of smoothness. Finally, at each level, highpass filtered images are combined using lowpass filtered mask as weights. The final result is obtained by adding the blended results at each level.

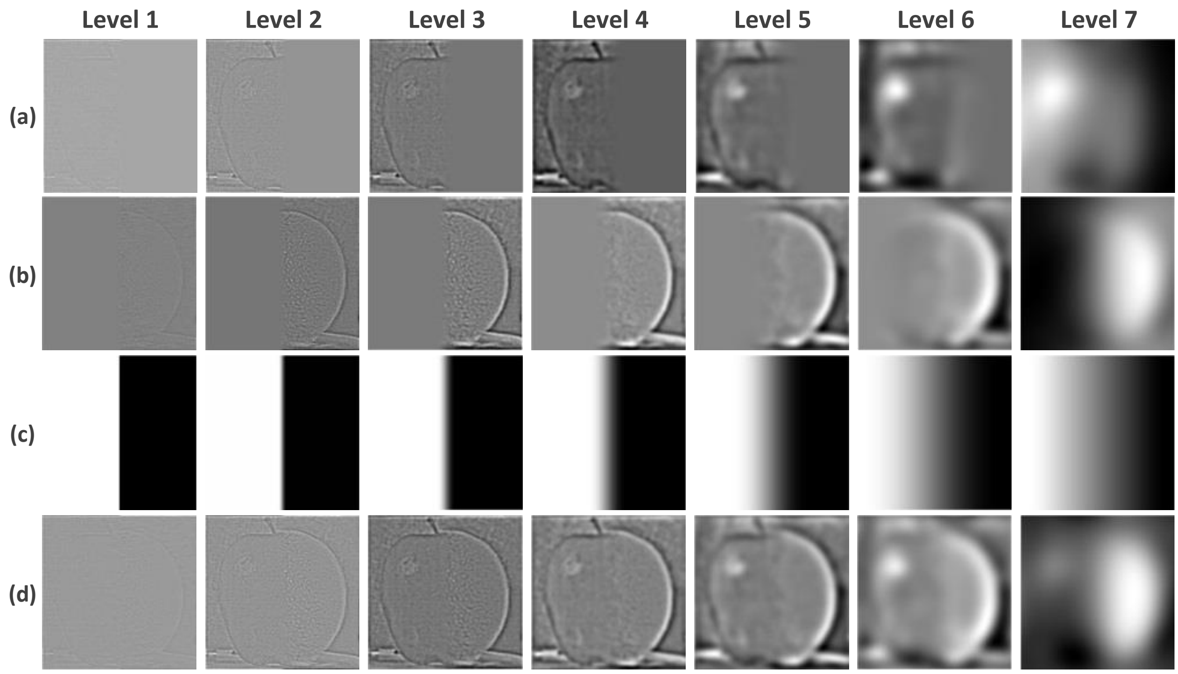


Figure 13 The procedure of the proposed method for mixing two images in different levels (a) The result of applying Laplacian filter on the left part (b) The result of applying Laplacian filter on the right part (c) The result of applying Gaussian filter on the mask (d) The result of combining two filtered images

Now let's see this in action. You are provided with three separate tasks, each with different considerations. The first one is asked to be blended in grayscale, while the other two must be done in color space. Also, the mask of desired regions is given in the first two tasks, while you have to design a mask of your own in the third.

You must apply the following procedure on each one of the given input pairs.

i. Build a set of Laplacian filtered versions of the images A and B using the equation $L_A^l = G_A^l - G_A^{l+1}$ (or $L_B^l = G_B^l - G_B^{l+1}$), where L_A^l (or L_B^l) and G_A^l (or G_B^l) are the Laplacian and Gaussian filtered versions of the image A (or B) at the level l, respectively. Set the number of levels $n = 6$, and the sigma value of the Gaussian filter at each level $\sigma = 2^{l-1}$.

ii. Build a set of Gaussian filtered versions of the mask image M. Consider $n = 6$, and the sigma value of the Gaussian filter at each level $\sigma = 2^{l-1}$.

iii. Compute the combined images C^l of each level l using

$$C^l = G_M^l L_A^l + (1 - G_M^l) L_B^l$$

iv. Obtain the final result C by summing C^l's through the levels $1, 2, \ldots, n$.

Note 1: Display all the intermediate results and also include them in your report.

Note 2: Your results must look as natural as possible. Try to find the most convenient parameters (mainly the number of levels, n) to accomplish this task.

(a) (b) (c) (d) (e)

Figure 14 Two examples of mixing images with the proposed method, one without alignment (top) and the other with alignment (bottom) (a) First image (base) (b) Second image (c) Mask of desired region (d) The result of simple copy-and-paste method (e) The result obtained by the proposed method

I. Joint product of Audi and Saipa!

The first task is a simple one; it only contains mixing images in grayscale.

a. Convert the images to grayscale and mix them using the above mentioned method. Your results must be in grayscale. Set different values for the number of levels *n*, and report the best result you obtained.

(a) **(b)** **(c)**

Figure 15 Creating a new model using a mask and a Gaussian filter! (a) 2020 Audi Q7 (b) SAIPA 151 (c) Mask which keeps half of each cars

II. Messi who? We have Donald in our team!

Now a more complicated case. The goal here is to merge images in color space. The above-mentioned method must first be applied to each channels separately, and then the blended results in each channel must be combined together to construct a final image.

b. Apply the method to different channels of the first image and obtain a final colored result. Set different values for the number of levels *n*, and report the best result you obtained.

(a) **(b)** **(c)**

Figure 16 Bringing Donald Trump into the football field! (a) Image of Lionel Messi (b) Aligned and edited image of Donald Trump (c) Mask separating Trump's head

III. Just a reminder: Wash your hands!

In the third and final task, the goal is to mix images in color space, while the mask is not given.

c. Define a zero matrix of the size of the t-shirt image, and then place the note image in the centre of it. Use this matrix to create a mask (set the nonzero values to 1) and a modified image of the note (set the zero values to the background color) to use in the above-mentioned algorithm.

d. Apply the method to different channels of the first image, and then obtain a final colored result. Set different values for the number of levels *n*, and report the best result you obtained.

(a) **(b)**

Figure 17 Avoid Coronavirus with regularly washing your hands! (a) T-shirt image (b) Note image

7. Some Explanatory Questions (5 Pts.)

Please answer the following questions as clear as possible:

a. Mathematically explain how rotation and scaling affects phase plot of an image in the Fourier domain.
b. How and under what circumstances the discrete Fourier transform relates to the Fourier transform?
c. In practice, even if Nyquist criterion is satisfied, aliasing cannot be avoided in general. Explain why.
d. The shifted 2D DFT usually forms the shape of plus sign, as it has larger values along the horizontal and vertical axes. Explain why.
e. Although performing convolution in 3D space, for example in video processing, and considering three dimensions of space-space-time, is possible, but it's a poor approach which has almost never been used in video processing. Explain why.

Good Luck!
Ali Abbasi