Assignment Solutions

AI Assistant

December 2, 2025

1 Problem 1

Directional filtering is used to emphasize or suppress frequency components along specific orientations in the frequency domain. For an $M \times N$ centered Discrete Fourier Transform (DFT) spectrum, the angle of each frequency component relative to the center of the spectrum is given by:

$$\theta(u,v) = \tan^{-1}\left(\frac{v - \frac{N}{2}}{u - \frac{M}{2}}\right)$$

A directional filter $H(u,v)$ can then be defined as:

$$H(u,v;\theta_{\min},\theta_{\max}) = \begin{cases} 1, & \text{if } \theta_{\min} \le \theta(u,v) \le \theta_{\max} \\ 0, & \text{otherwise} \end{cases}$$

Here, $\theta_{\min}$ and $\theta_{\max}$ specify the angular range of frequencies to be retained.

1.1 $Generateanimagex$ of size $M \times M$ $(M = 256)$ using three sinusoidal components:

$$x_1(m,n) = \sin\left(\frac{2\pi \cdot 12\,m}{M}\right),$$

$$x_2(m,n) = \sin\left(\frac{2\pi \cdot 8\,n}{M}\right),$$

$$x_3(m,n) = \sin\left(\frac{2\pi(6\,m + 10\,n)}{M}\right),$$

$$x(m,n) = \frac{x_1(m,n) + x_2(m,n) + x_3(m,n)}{3}.$$

Compute the centered 2D DFT of $x(m,n)$. Plot the results in a single figure showing:

1.1.1 The image of the sinusoidal component $x_1(m,n)$

Solution:

Figure 1: The image of the sinusoidal component $x_1(m,n)$

1.1.2 The image of the sinusoidal component $x_2(m,n)$

Solution:

Figure 2: The image of the sinusoidal component $x_2(m,n)$

1.1.3 The image of the sinusoidal component $x_3(m,n)$

Solution:

Figure 3: The image of the sinusoidal component $x_3(m,n)$

1.1.4 The combined image $x(m, n)$

Solution:

Figure 4: The combined image $x(m, n)$

1.1.5 The magnitude of the 2D DFT of the combined image

Solution:

Figure 5: The magnitude of the 2D DFT of the combined image

1.2 $Designdirectionalfiltersofsize M \times M$ $(M = 256)$ for given angular ranges using the expression above. Apply each filter to the DFT of the combined image $(X(u, v))$ reconstruct the filtered images using the inverse DFT, visualize the results and comment on the observations. For each directional filter, display the following in a single figure:

1.2.1 The original image.

Solution:

Figure 6: The original image.

1.2.2 The original image magnitude spectrum.

Solution:

Figure 7: The original image magnitude spectrum.

1.2.3 The directional filter magnitude spectrum.

Solution:

Figure 8: The directional filter magnitude spectrum.

1.2.4 The filtered magnitude spectrum.

Solution:

Figure 9: The filtered magnitude spectrum.

1.2.5 The reconstructed filtered image.

Solution:

Figure 10: The reconstructed filtered image.

1.3 Compute the Mean Squared Error (MSE) between the original and each filtered output and comment on their values.

Solution:

The Mean Squared Error (MSE) is a measure of the average squared difference between the original image x and each reconstructed image x_{recon}. It is calculated as $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_{\text{recon},}\ i)^2$, where n is the total number of pixels in the image.

In the given code, the MSE is computed using the function $\text{mse}(img1, img2) = \frac{1}{n}\sum_{i=1}^{n}(img1_i - img2_i)^2$, which takes two images as input, subtracts them element-wise, squares the result, and returns the mean of the squared differences.

The MSE values between the original image and each reconstruction are stored in a dictionary mse_results, where the keys are the names of the reconstructions and the values are the corresponding MSE values. These values are then printed out, representing the difference between the original and reconstructed images.

The MSE values can be interpreted as follows: a lower MSE value indicates that the reconstructed image is closer to the original image, while a higher MSE value indicates a greater difference between the two.

2 Problem 1

Gaussian Blurring and Inverse Filtering: Gaussian blurring is a common technique to smooth an image and attenuate high-frequency components. A Gaussian kernel of size $k \times k$ and standard deviation σ is defined as:

$$G(x, y) = K \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad \text{with } Kchosensuchthat\text{with } Kchosensuchthat\text{with } Kchosensuchthat\text{with } Kchosensuchthat \textstyle\sum_{x,y} G$$

2.1 $Readtheimagebuildings.jpg.DesignaGaussiankernelofsize 13\times13$ with standard deviation $\sigma = 2.5$. Apply the blur in the frequency domain and reconstruct the blurred image using the inverse DFT.

Solution:

Figure 11: Read the image buildings.jpg. Design a Gaussian kernel of size 13 × 13 with standard devi...

2.2 For the Gaussian kernel compute and plot in a single figure and comment on your observations:

2.2.1 Its centered 2D DFT magnitude spectrum (13×13-point DFT).

Solution:

Figure 12: Its centered 2D DFT magnitude spectrum (13×13-point DFT).

2.2.2 $Theinversecenteredmagnitudespectrum1/(|H(u,v)|+\epsilon)$, with $\epsilon = 10^{-3}$.

Solution:

Figure 13: The inverse centered magnitude spectrum $1/(|H(u,v)|+\epsilon)$, with $\epsilon = 10^{-3}$.

2.2.3 The centered 2D DFT magnitude spectrum (1036×1036-point DFT).

Solution:

Figure 14: The centered 2D DFT magnitude spectrum (1036×1036-point DFT).

2.2.4 The inverse centered magnitude spectrum $1/(|H(u,v)|+\epsilon)$.

Solution:

Figure 15: The inverse centered magnitude spectrum $1/(|H(u,v)|+\epsilon)$.

2.3 Gaussian frequency response fit:

2.3.1 $The frequency-domain representation of a Gaussian kernel can be modeled as: H_{\text{cont}}(u,v) = \exp\Big(-k\,(U^2+V^2)\Big), \quad U = u - \frac{M-1}{2},\ V = v - \frac{N-1}{2}$

Solution:

The frequency-domain representation of a Gaussian kernel, $H_{\text{cont}}(u,v)$, is modeled as $\exp\Big(-k\,(U^2+V^2)\Big)$, where $U = u - \frac{M-1}{2}$ and $V = v - \frac{N-1}{2}$.

In the provided Python code, the ‘create$_{g}aussian_{k}ernel‘functiongeneratesa2DGaussiankernelusingNumPy,which$

The ‘get$_{p}added_{d}ft‘functionpadstheGaussiankernelwithzerosandcomputesits2DDiscreteFourierTransform(DFT$
$0) for the FFT. The uncentered and centered versions of the DFT are computed using ‘np.fft.fft2‘ and ‘np.fft.fftshift‘, res$

2.3.2 Sweep k over the range 10^{-6} to 10^{-3} and find the value that minimizes the error:

$$\min_k \sum_{u,v} \Big| H_{\text{cont}}(u,v) - |H_{\text{DFT}}(u,v)| \Big|^2$$

Solution:

The goal is to find the optimal value of k that minimizes the sum of squared errors between the continuous Gaussian function $H_{\text{cont}}(u,v)$ and the magnitude of the discrete Fourier transform $|H_{\text{DFT}}(u,v)|$.

We define the continuous Gaussian function as $H_{\text{cont}}(u,v) = \exp(-k(u^2+v^2))$. The sum of squared errors is calculated as $\sum_{u,v} |H_{\text{cont}}(u,v) - |H_{\text{DFT}}(u,v)||^2$.

To find the optimal k, we sweep over the range 10^{-6} to 10^{-3} and compute the sum of squared errors for each k value. The optimal k is the one that minimizes this error.

The Python code implements this approach by creating centered pixel indices for U and V coordinate grids using NumPy's meshgrid function. It then defines a function ‘sse$_{e}rror‘tocomputethesumofsquarederrorsforagiven$

2.3.3 Report the optimized k_{opt} obtained from the previous sweep and plot in a single figure the magnitude spectrum of the Gaussian fit $|H_{\text{cont}}(u,v)|$ along with its inverse $1/(|H_{\text{cont}}(u,v)|+\epsilon)$.

Solution:

Figure 16: Log-scaled magnitude spectra of the Gaussian fit and its inverse

Figure 17: Report the optimized k_{opt} obtained from the previous sweep and plot in a single figure...

optimized k_{opt} is obtained by minimizing the sum of squared errors (SSE) between the target magnitude spectrum H_{target} and the generated Gaussian function H_{cont}. This is achieved by sweeping over a range of k values and selecting the one that results in the smallest SSE.

The magnitude spectrum of the Gaussian fit $|H_{\text{cont}}(u, v)|$ is calculated as $\exp(-k_{\text{opt}}(u^2 + v^2))$, where u and v are the centered pixel indices of the frequency coordinate grids.

The inverse of the Gaussian fit is calculated as $1/(|H_{\text{cont}}(u, v)| + \epsilon)$, where ϵ is a small value added to prevent division by zero.

The resulting magnitude spectrum and its inverse are plotted in a single figure using a log scale, as shown below:

2.4 Restore the original image by applying the previously computed inverse responses (both from the direct kernel DFT and the Gaussian fit) and reconstruct the images using the inverse DFT. Plot in a single figure and compare:

2.4.1 The original image.

Solution:

Figure 18: Magnitude spectrum of the Gaussian fit $|H_{\text{cont}}(u,v)|$ and its inverse $1/(|H_{\text{cont}}(u,v)|+\epsilon)$

Figure 19: The original image.

2.4.2 The restored image using the direct kernel inverse response.

Solution:

Figure 20: The restored image using the direct kernel inverse response.

2.4.3 The restored image using the Gaussian fit inverse response.

Solution:

Figure 21: The restored image using the Gaussian fit inverse response.

2.4.4 Compute and report the Mean Squared Error (MSE) between the original image and each reconstructed image and comment on which method gives better restoration and why.

Solution:

The Mean Squared Error (MSE) between the original image x and each reconstructed image x_{recon} is calculated using the function $\text{mse}(x, x_{\text{recon}}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_{\text{recon},i})^2$, where n is the number of pixels in the image.

The MSE is a measure of the average squared difference between the original and reconstructed images. A lower MSE value indicates better restoration, as it means the reconstructed image is closer to the original.

The script computes the MSE for each reconstructed image in the dictionary reconstructed_images and stores the results in the dictionary mse_results. The MSE values are then printed out for each reconstruction, allowing for comparison of the restoration quality.

Note that the MSE is sensitive to outliers, so a single pixel with a large difference can greatly affect the MSE value. However, it is a simple and widely used metric for image restoration quality.