**Frequency-Domain Filter**

**Outline**
- Convolution theorems:
  - Steps for filtering in frequency domain
- Frequency-domain filters:
  - Notch filter
  - Low-pass filters
  - High-pass filters

---

**Convolution Theorems**

\[ f(x,y) * h(x,y) = \frac{1}{MN} \sum_{m=-M}^{M} \sum_{n=-N}^{N} f(m,n) h(x-m, y-n) \]

- The convolution in spatial domain reduces to multiplication in the frequency domain
- Element-by-element complex number multiplication
- The convolution in frequency domain reduces to multiplication in the spatial domain

---

**Filtering in Frequency Domain**

Pre-processing step:
- Multiply image with \((-1)^{x+y}\) to center the transform
- Compute the DFT
- Multiply with frequency filter function \(H(u,v)\)
- Compute the inverse DFT
  - In general, the result is complex number
  - When both input image & filter function are real, the imaginary components should all be zero

Post-processing step:
- Multiply with \((-1)^{x+y}\)

---

**Notch Filter**

- \(F(0,0)\) keeps the average of the input image
- Changing this term to zero then taking the inverse transform will change the average intensity to zero
- The corresponding filter function is a constant function with a hole at the origin
- Called notch filter

\[ H(u,v) = \begin{cases} 0 & \text{if } (u,v) = \left(\frac{M}{2}, \frac{N}{2}\right) \\ 1 & \text{otherwise} \end{cases} \]
**Frequency Signals**

- **Low frequency:**
  - Areas close to the center in the Fourier transform
  - Responsible for the general gray-level appearance of an image over smooth areas
- **Low-pass filters:**
  - Preserve low frequencies & suppress high frequencies
  - Smoothing filters

- **High frequency:**
  - Areas far away from the center in the Fourier transform
  - Responsible for detail, such as edges & noise
- **High-pass filters:**
  - Preserve high frequencies & suppress low frequencies
  - Sharpening filters

**Ideal Low-pass Filter (ILPF)**

- Simply cut off all high frequency components of the Fourier transform
- \( D_0 \) is cutoff frequency
- The corresponding spatial filter has ringing behavior

\[
H(u,v) = \begin{cases} 
1 & \text{if } D(u,v) \leq D_0 \\
0 & \text{Otherwise}
\end{cases}
\]

\[
D(u,v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}
\]

**Ideal High-pass Filter (IHPF)**

- Simply cut off all low frequency components of the Fourier transform
- Notch filter can be considered as a special case
- The corresponding spatial filter also has ringing behavior

\[
H(u,v) = \begin{cases} 
0 & \text{if } D(u,v) \leq D_0 \\
1 & \text{Otherwise}
\end{cases}
\]
Fourier Transform of Gaussian

\[ h(u) = \int e^{-u^2} e^{2\pi j u x} du \]
\[ H(u) = e^{-u^2} \]
\[ = \int e^{-u^2} \sin(2\pi mx + j \sin(2\pi mx)) du \]
\[ = \int e^{-u^2} \cos(2\pi mx) du + j \int e^{-u^2} \sin(2\pi mx) du \]
\[ = \int e^{-u^2} \cos(2\pi mx) du = \frac{\pi}{\sqrt{u}} e^{-\pi^2 u} \]
\[ \int e^{-u^2} \sin(2\pi mx) du = 0 \]

Gaussian Low-pass Filter (GLPF)

- Multiply the Fourier transform with a 2D Gaussian function
- \( D_0 \) is the cutoff frequency
- The value of the filter is 0.607 when \( D(u,v)=D_0 \)
- The corresponding spatial filter is also a Gaussian
- Has no ringing behavior

\[ H(u,v) = e^{-\frac{D(u,v)}{2D_0}} \]

Example of GLPF

Gaussian High-pass Filter (GHPF)

- Multiply the Fourier transform with a flipped 2D Gaussian function
- The corresponding spatial filter has no ringing behavior either
- Once the curve passes \( y=0 \), it stays at negative

\[ H(u,v) = 1 - e^{-\frac{D(u,v)}{2D_0}} \]

Example of GHPF

Laplacian in Frequency Domain

\[ \frac{d^2 f(x)}{dx^2} \]
\[ \nabla^2 f(x,y) = (ju)^2 F(u,v) + (jv)^2 F(u,v) = -(u^2 + v^2)F(u,v) \]

Laplacian can be implemented in the frequency domain using:

\[ H(u,v) = \left[ u - \frac{M^2}{2} \right] + \left( v - \frac{N^2}{2} \right) \]
<table>
<thead>
<tr>
<th>Understanding in spatial domain:</th>
<th>Understanding in frequency domain:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate a sharp image by subtracting from an image a blurred version of itself</td>
<td>Obtain a high-pass filtered image by subtracting from the image a low-pass filtered version of itself</td>
</tr>
</tbody>
</table>

\[ f_{ls}(x, y) = f(x, y) - \Delta f(x, y) \]

\[ H_{hp}(u, v) = 1 - H_{lp}(u, v) \]