Digital Image Processing
Chapter 4:
Image Enhancement in the Frequency Domain
Background: Fourier Series

Fourier series:
Any periodic signals can be viewed as weighted sum of sinusoidal signals with different frequencies.

Frequency Domain:
view frequency as an independent variable.
Fourier Tr. and Frequency Domain

Time, spatial Domain Signals

Fourier Tr.

Frequency Domain Signals

Inv Fourier Tr.

1-D, Continuous case

Fourier Tr.: \[ F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux} \, dx \]

Inv. Fourier Tr.: \[ f(x) = \int_{-\infty}^{\infty} F(u)e^{j2\pi ux} \, du \]
1-D, Discrete case

Fourier Tr.: \[ F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M} \quad u = 0, \ldots, M-1 \]

Inv. Fourier Tr.: \[ f(x) = \sum_{u=0}^{M-1} F(u)e^{j2\pi ux/M} \quad x = 0, \ldots, M-1 \]

\( F(u) \) can be written as

\[ F(u) = R(u) + jI(u) \quad \text{or} \quad F(u) = |F(u)|e^{-j\phi(u)} \]

where

\[ |F(u)| = \sqrt{R(u)^2 + I(u)^2} \quad \phi(u) = \tan^{-1}\left(\frac{I(u)}{R(u)}\right) \]
Example of 1-D Fourier Transforms

Notice that the longer the time domain signal, the shorter its Fourier transform.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Relation Between $\Delta x$ and $\Delta u$

For a signal $f(x)$ with $M$ points, let spatial resolution $\Delta x$ be space between samples in $f(x)$ and let frequency resolution $\Delta u$ be space between frequencies components in $F(u)$, we have

$$\Delta u = \frac{1}{M \Delta x}$$

Example: for a signal $f(x)$ with sampling period 0.5 sec, 100 point, we will get frequency resolution equal to

$$\Delta u = \frac{1}{100 \times 0.5} = 0.02 \text{ Hz}$$

This means that in $F(u)$ we can distinguish 2 frequencies that are apart by 0.02 Hertz or more.
2-Dimensional Discrete Fourier Transform

For an image of size MxN pixels

2-D DFT

\[
F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi \left(\frac{ux}{M} + \frac{vy}{N}\right)}
\]

\(u = \text{frequency in } x \text{ direction, } u = 0 ,\ldots, M-1\)
\(v = \text{frequency in } y \text{ direction, } v = 0 ,\ldots, N-1\)

2-D IDFT

\[
f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi \left(\frac{ux}{M} + \frac{vy}{N}\right)}
\]

\(x = 0 ,\ldots, M-1\)
\(y = 0 ,\ldots, N-1\)
2-Dimensional Discrete Fourier Transform (cont.)

\( F(u, v) \) can be written as

\[
F(u, v) = R(u, v) + jI(u, v) \quad \text{or} \quad F(u, v) = |F(u, v)|e^{-j\phi(u,v)}
\]

where

\[
|F(u, v)| = \sqrt{R(u, v)^2 + I(u, v)^2} \quad \phi(u, v) = \tan^{-1}\left(\frac{I(u, v)}{R(u, v)}\right)
\]

For the purpose of viewing, we usually display only the Magnitude part of \( F(u,v) \)
# 2-D DFT Properties

## TABLE 4.1
Summary of some important properties of the 2-D Fourier transform.

<table>
<thead>
<tr>
<th>Property</th>
<th>Expression(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fourier transform</td>
<td>[ F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)} ]</td>
</tr>
<tr>
<td>Inverse Fourier transform</td>
<td>[ f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)} ]</td>
</tr>
<tr>
<td>Polar representation</td>
<td>[ F(u, v) =</td>
</tr>
<tr>
<td>Spectrum</td>
<td>[</td>
</tr>
<tr>
<td>Phase angle</td>
<td>[ \phi(u, v) = \tan^{-1} \left( \frac{I(u, v)}{R(u, v)} \right) ]</td>
</tr>
<tr>
<td>Power spectrum</td>
<td>[ P(u, v) =</td>
</tr>
<tr>
<td>Average value</td>
<td>[ \bar{f}(x, y) = F(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) ]</td>
</tr>
<tr>
<td>Translation</td>
<td>[ f(x, y) e^{j2\pi(ux_0/M + vy_0/N)} \Leftrightarrow F(u - u_0, v - v_0) ] [ f(x - x_0, y - y_0) \Leftrightarrow F(u, v) e^{-j2\pi ux_0/M + vy_0/N} ]</td>
</tr>
</tbody>
</table>

When \( x_0 = u_0 = M/2 \) and \( y_0 = v_0 = N/2 \), then

\[ f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2) \]

\[ f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{u-v} \]
2-D DFT Properties (cont.)

<table>
<thead>
<tr>
<th>Property</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conjugate symmetry</td>
<td>$F(u, v) = F^*(u, -v)$</td>
</tr>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td>Differentiation</td>
<td>$\frac{\partial^n f(x, y)}{\partial x^n} \Leftrightarrow (ju)^n F(u, v)$</td>
</tr>
<tr>
<td></td>
<td>$(-jx)^n f(x, y) \Leftrightarrow \frac{\partial^n F(u, v)}{\partial u^n}$</td>
</tr>
<tr>
<td>Laplacian</td>
<td>$\nabla^2 f(x, y) \Leftrightarrow -(u^2 + v^2) F(u, v)$</td>
</tr>
<tr>
<td>Distributivity</td>
<td>$\mathbb{F}[f_1(x, y) + f_2(x, y)] = \mathbb{F}[f_1(x, y)] + \mathbb{F}[f_2(x, y)]$</td>
</tr>
<tr>
<td></td>
<td>$\mathbb{F}[f_1(x, y) \cdot f_2(x, y)] \neq \mathbb{F}[f_1(x, y)] \cdot \mathbb{F}[f_2(x, y)]$</td>
</tr>
<tr>
<td>Scaling</td>
<td>$af(x, y) \Leftrightarrow aF(u, v), \ f(ax, by) \Leftrightarrow \frac{1}{</td>
</tr>
<tr>
<td>Rotation</td>
<td>$x = r \cos \theta, \ y = r \sin \theta, \ u = \omega \cos \varphi, \ v = \omega \sin \varphi$</td>
</tr>
<tr>
<td></td>
<td>$f(r, \theta + \theta_0) \Leftrightarrow F(\omega \cdot \varphi + \theta_0)$</td>
</tr>
<tr>
<td>Periodicity</td>
<td>$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$</td>
</tr>
<tr>
<td></td>
<td>$f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N)$</td>
</tr>
<tr>
<td>Separability</td>
<td>See Eqs. (4.6-14) and (4.6-15). Separability implies that we can compute the 2-D transform of an image by first computing 1-D transforms along each row of the image, and then computing a 1-D transform along each column of this intermediate result. The reverse, columns and then rows, yields the same result.</td>
</tr>
</tbody>
</table>

(Images from Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, 2nd Edition.)
### 2-D DFT Properties (cont.)

<table>
<thead>
<tr>
<th>Property</th>
<th>Expression(s)</th>
</tr>
</thead>
</table>
| Computation of the inverse Fourier transform  | \[
\frac{1}{MN} f^*(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j2\pi (ux/M + vy/N)}
\]  
This equation indicates that inputting the function \(F^*(u, v)\) into an algorithm designed to compute the forward transform (right side of the preceding equation) yields \(f^*(x, y)/MN\). Taking the complex conjugate and multiplying this result by \(MN\) gives the desired inverse. |
| Convolution†                                   | \[
f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)
\]                                                                                   |
| Correlation†                                   | \[
f(x, y) \circ h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) h(x + m, y + n)
\]                                                                                   |
| Convolution theorem†                           | \[
\begin{align*}
f(x, y) * h(x, y) & \iff F(u, v) H(u, v); \\
f(x, y) h(x, y) & \iff F(u, v) * H(u, v)
\end{align*}
\]                                                                                   |
| Correlation theorem†                           | \[
\begin{align*}
f(x, y) \circ h(x, y) & \iff F^*(u, v) H(u, v); \\
F^*(x, y) h(x, y) & \iff F(u, v) \circ H(u, v)
\end{align*}
\]                                                                                   |

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
### 2-D DFT Properties (cont.)

Some useful FT pairs:

- **Impulse** \( \delta(x, y) \leftrightarrow 1 \)
- **Gaussian** 
  \[ A \sqrt{2\pi\sigma} e^{-2\pi^2\sigma^2(x^2 + y^2)} \leftrightarrow Ae^{-(u^2 + v^2)/2\sigma^2} \]
- **Rectangle** 
  \[ \text{rect}[a, b] \leftrightarrow ab \frac{\sin(\pi u a)}{\pi u a} \frac{\sin(\pi v b)}{\pi v b} e^{-j\pi(u a + v b)} \]
- **Cosine** 
  \[ \cos(2\pi u_0 x + 2\pi v_0 y) \leftrightarrow \frac{1}{2} \left[ \delta(u + u_0, v + v_0) + \delta(u - u_0, v - v_0) \right] \]
- **Sine** 
  \[ \sin(2\pi u_0 x + 2\pi v_0 y) \leftrightarrow j \frac{1}{2} \left[ \delta(u + u_0, v + v_0) - \delta(u - u_0, v - v_0) \right] \]

* Assumes that functions have been extended by zero padding.

---

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Computational Advantage of FFT Compared to DFT

**FIGURE 4.42**
Computational advantage of the FFT over a direct implementation of the 1-D DFT. Note that the advantage increases rapidly as a function of $n$.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Relation Between Spatial and Frequency Resolutions

\[
\Delta u = \frac{1}{M \Delta x}
\]

\[
\Delta v = \frac{1}{N \Delta y}
\]

where

\( \Delta x = \) spatial resolution in \( x \) direction

\( \Delta y = \) spatial resolution in \( y \) direction

( \( \Delta x \) and \( \Delta y \) are pixel width and height. )

\( \Delta u = \) frequency resolution in \( x \) direction

\( \Delta v = \) frequency resolution in \( y \) direction

\( N, M = \) image width and height
How to Perform 2-D DFT by Using 1-D DFT

$f(x,y)$

1-D DFT by row

$F(u,y)$

1-D DFT by column

$F(u,v)$
Alternative method

\[ f(x, y) \]

\[ F(x, v) \]  
1-D DFT by column

\[ F(u, v) \]  
1-D DFT by row
**Periodicity of 1-D DFT**

From DFT:

\[
F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux / M}
\]

DFT repeats itself every \( N \) points (Period = \( N \)) but we usually display it for \( n = 0, \ldots, N-1 \).
Conventional Display for 1-D DFT

The graph $F(u)$ is not easy to understand!
Conventional Display for DFT: FFT Shift

FFT Shift: Shift center of the graph $F(u)$ to 0 to get better display which is easier to understand.

- High frequency area
- Low frequency area
Periodicity of 2-D DFT

2-D DFT: \[ F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)} \]

For an image of size \( N \times M \) pixels, its 2-D DFT repeats itself every \( N \) points in \( x \)-direction and every \( M \) points in \( y \)-direction.

We display only in this range

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Conventional Display for 2-D DFT

$F(u,v)$ has low frequency areas at corners of the image while high frequency areas are at the center of the image which is inconvenient to interpret.

High frequency area

Low frequency area

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
2-D FFT Shift: Better Display of 2-D DFT

2-D FFT Shift is a MATLAB function: Shift the zero frequency of $F(u,v)$ to the center of an image.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
2-D FFT Shift (cont.) : How it works

Display of 2D DFT
After FFT Shift

Original display of 2D DFT

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Notice that the longer the time domain signal, the shorter its Fourier transform.
Example of 2-D DFT

Notice that direction of an object in spatial image and its Fourier transform are orthogonal to each other.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Example of 2-D DFT

Original image

2D DFT

2D FFT Shift
Example of 2-D DFT

Original image

2D DFT

2D FFT Shift
From Fourier Transform Property:

\[ g(x, y) = f(x, y) * h(x, y) \iff F(u, v) \cdot H(u, v) = G(u, v) \]

We can perform filtering process by using multiplication in the frequency domain, which is easier than convolution in the spatial domain.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Filtering in the Frequency Domain with FFT shift

In this case, $F(u,v)$ and $H(u,v)$ must have the same size and have the zero frequency at the center.
Multiplication of DFTs of 2 signals is equivalent to perform circular convolution in the spatial domain.

“Wrap around” effect
Multiplication in Freq. Domain = Circular Convolution

$H(u,v)$
Gaussian Lowpass Filter with $D_0 = 5$

Original image

Filtered image (obtained using circular convolution)

Incorrect areas at image rims
Linear Convolution by using Circular Convolution and Zero Padding

\[ f(x) \rightarrow \text{Zero padding} \rightarrow \text{DFT} \rightarrow F(u) \]

\[ h(x) \rightarrow \text{Zero padding} \rightarrow \text{DFT} \rightarrow H(u) \]

\[ G(u) = F(u)H(u) \]

\[ \text{IDFT} \rightarrow \text{Concatenation} \]

\[ g(x) \]

Before DFT

Padding zeros

Keep only this part
Linear Convolution by using Circular Convolution and Zero Padding

**FIGURE 4.38**
Illustration of the need for function padding.
(a) Result of performing 2-D convolution without padding.
(b) Proper function padding.
(c) Correct convolution result.

Linear Convolution by using Circular Convolution and Zero Padding

Zero padding area in the spatial Domain of the mask image (the ideal lowpass filter)

Filtered image

Only this area is kept.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Filtering in the Frequency Domain: Example

In this example, we set \( F(0,0) \) to zero which means that the zero frequency component is removed.

Note: Zero frequency = average intensity of an image

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Filtering in the Frequency Domain: Example

Lowpass Filter

Highpass Filter

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)

FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).
Filtering in the Frequency Domain: Example (cont.)

FIGURE 4.8
Result of highpass filtering the image in Fig. 4.4(a) with the filter in Fig. 4.7(c), modified by adding a constant of one-half the filter height to the filter function. Compare with Fig. 4.4(a).

Result of Sharpening Filter

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Filter Masks and Their Fourier Transforms

**FIGURE 4.9**
(a) Gaussian frequency domain lowpass filter.
(b) Gaussian frequency domain highpass filter.
(c) Corresponding lowpass spatial filter.
(d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Ideal Lowpass Filter

Ideal LPF Filter Transfer function

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

where \( D(u,v) = \text{Distance from (u,v) to the center of the mask.} \)

FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.
Examples of Ideal Lowpass Filters

FIGURE 4.11 (a) An image of size $500 \times 500$ pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

The smaller $D_0$, the more high frequency components are removed.
Results of Ideal Lowpass Filters

Ringing effect can be obviously seen!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
How ringing effect happens

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

Ideal Lowpass Filter with \( D_0 = 5 \)

Abrupt change in the amplitude
How ringing effect happens (cont.)

Spatial Response of Ideal Lowpass Filter with $D_0 = 5$

Ripples that cause ringing effect
How ringing effect happens (cont.)

**FIGURE 4.13** (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Butterworth Lowpass Filter

Transfer function

\[
H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0}\right]^{2N}}
\]

Where \(D_0 = \text{Cut off frequency}, N = \text{filter order}\.}

\(H(u, v)\)

\(u\)

\(v\)

\(D_0\)

\(D(u, v)\)

**FIGURE 4.14** (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Results of Butterworth Lowpass Filters

There is less ringing effect compared to those of ideal lowpass filters!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Spatial Masks of the Butterworth Lowpass Filters

Some ripples can be seen.

**FIGURE 4.16** (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Gaussian Lowpass Filter

Transfer function

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

Where \( D_0 \) = spread factor.

Note: the Gaussian filter is the only filter that has no ripple and hence no ringing effect.
Gaussian Lowpass Filter (cont.)

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

Gaussian lowpass filter with \( D_0 = 5 \)

Spatial responses of the Gaussian lowpass filter with \( D_0 = 5 \)

Gaussian shape
Results of Gaussian Lowpass Filters

No ringing effect!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Application of Gaussian Lowpass Filters

**FIGURE 4.19**
(a) Sample text of poor resolution (note broken characters in magnified view).
(b) Result of filtering with a GLPF (broken character segments were joined).

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

The GLPF can be used to remove jagged edges and "repair" broken characters.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Application of Gaussian Lowpass Filters (cont.)

Remove wrinkles

Original image

Softer-Looking

**FIGURE 4.20** (a) Original image (1028 × 732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Application of Gaussian Lowpass Filters (cont.)

Original image: The gulf of Mexico and Florida from NOAA satellite.

Filtered image


Remove artifact lines: this is a simple but crude way to do it!
Highpass Filters

\[ H_{hp} = 1 - H_{lp} \]

**Figure 4.22** Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Ideal Highpass Filters

Ideal LPF Filter Transfer function

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

where \(D(u,v) = \text{Distance from } (u,v) \text{ to the center of the mask.}\)

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Butterworth Highpass Filters

Transfer function

\[ H(u, v) = \frac{1}{1 + \left[ \frac{D_0}{D(u, v)} \right]^{2N}} \]

Where \( D_0 \) = Cut off frequency, \( N \) = filter order.
Gaussian Highpass Filters

Transfer function

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

Where \( D_0 \) = spread factor.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Gaussian Highpass Filters (cont.)

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

Gaussian highpass filter with \( D_0 = 5 \)

Spatial responses of the Gaussian highpass filter with \( D_0 = 5 \)
Spatial Responses of Highpass Filters

**FIGURE 4.23** Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Results of Ideal Highpass Filters

Ringing effect can be obviously seen!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
**FIGURE 4.25** Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15, 30,$ and $80$, respectively. These results are much smoother than those obtained with an ILPF.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Results of Gaussian Highpass Filters

FIGURE 4.26  Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15, 30, \text{ and } 80$, respectively. Compare with Figs. 4.24 and 4.25.
**Laplacian Filter in the Frequency Domain**

From Fourier Tr. Property:

\[
\frac{d^n f(x)}{dx^n} \leftrightarrow (ju)^n F(u)
\]

Then for Laplacian operator

\[
\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \leftrightarrow -(u^2 + v^2)F(u, v)
\]

We get

\[
\nabla^2 \leftrightarrow -(u^2 + v^2)
\]

Image of \(-(u^2+v^2)\)

Surface plot

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Laplacian Filter in the Frequency Domain (cont.)

Spatial response of \(-(u^2+v^2)\)

Cross section

Laplacian mask in Chapter 3
Sharpening Filtering in the Frequency Domain

Spatial Domain

\[ f_{hp}(x, y) = f(x, y) - f_{lp}(x, y) \]

\[ f_{hb}(x, y) = Af(x, y) - f_{lp}(x, y) \]

\[ f_{hb}(x, y) = (A - 1)f(x, y) + f(x, y) - f_{lp}(x, y) \]

\[ f_{hb}(x, y) = (A - 1)f(x, y) + f_{hp}(x, y) \]

Frequency Domain Filter

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

\[ H_{hb}(u, v) = (A - 1) + H_{hp}(u, v) \]

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Sharpening Filtering in the Frequency Domain (cont.)

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Sharpening Filtering in the Frequency Domain (cont.)

\[ f_{hb}(x, y) = (A - 1) f(x, y) + f_{hp}(x, y) \]

\[ f_{hp} = \nabla^2 P \]

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
High Frequency Emphasis Filtering

$$H_{hfe}(u,v) = a + bH_{hp}(u,v)$$

Original

Butterworth highpass filtered image

High freq. emphasis filtered image

$$a = 0.5, \quad b = 2$$

After Hist Eq.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
**Homomorphic Filtering**

An image can be expressed as

\[ f(x, y) = i(x, y)r(x, y) \]

- \( i(x,y) \) = illumination component
- \( r(x,y) \) = reflectance component

We need to suppress the effect of illumination that causes image intensity to change slowly.

\[ f(x, y) \xrightarrow{\text{ln}} \xrightarrow{\text{DFT}} \xrightarrow{H(u, v)} \xrightarrow{(\text{DFT})^{-1}} \xrightarrow{\text{exp}} g(x, y) \]

**FIGURE 4.31**
Homomorphic filtering approach for image enhancement.
Homomorphic Filtering

\[ f(x, y) \rightarrow \ln \rightarrow \text{DFT} \rightarrow H(u, v) \rightarrow (\text{DFT})^{-1} \rightarrow \exp \rightarrow g(x, y) \]

**FIGURE 4.31**
Homomorphic filtering approach for image enhancement.

**FIGURE 4.32**
Cross section of a circularly symmetric filter function. \( D(u, v) \) is the distance from the origin of the centered transform.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Homomorphic Filtering

FIGURE 4.33
(a) Original image. (b) Image processed by homomorphic filtering (note details inside shelter). (Stockham.)

More details in the room can be seen!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)
Correlation Application: Object Detection

**Figure 4.41**
(a) Image.
(b) Template.
(c) and (d) Padded images.
(e) Correlation function displayed as an image.
(f) Horizontal profile line through the highest value in (e), showing the point at which the best match took place.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)