• Wraparound and padding
• Image Correlation
• Image Processing in the frequency domain
• A simple frequency domain filter
• Frequency domain filters
  – High-pass, low-pass
  – Apodization
• Zero-phase filtering
• Frequency domain filters
  – ideal, Butterworth, Gaussian
  – Ringing
Wraparound and Padding

DFT makes f & h periodic

Mirror h to get h(-m)

Now shift h(-m) to get h(x-m)

If f and g are 400 points f★g goes out to 800 points

Wraparound comes from overlap of the periodic functions

This was not here before

We must pad h with 399 points of zero to get the correct result with this periodic function

If f and g are 400 points f★g goes out to 800 points

We must pad h with 399 points of zero to get the correct result with this periodic function

© 2002 R. C. Gonzalez & R. E. Woods
We also have to pad images but in two-dimensions to get the correct convolution results.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure4_38.png}
\caption{Illustration of the need for function padding.}
\item (a) Result of performing 2-D convolution without padding.
\item (b) Proper function padding.
\item (c) Correct convolution result.
\end{figure}
Wraparound and Padding

No blurring at edges — no padding

FIGURE 4.32 (a) A simple image. (b) Result of blurring with a Gaussian lowpass filter without padding. (c) Result of lowpass filtering with padding. Compare the light area of the vertical edges in (b) and (c).

\[
\begin{bmatrix}
\frac{1}{16} & 1 & 2 & 1 \\
1 & 2 & 4 & 2 \\
2 & 4 & 8 & 4 \\
1 & 2 & 1 & 0
\end{bmatrix}
\]
Wraparound and Padding

**FIGURE 4.33** 2-D image periodicity inherent in using the DFT. (a) Periodicity without image padding. (b) Periodicity after padding with 0s (black). The dashed areas in the center correspond to the image in Fig. 4.32(a). (The thin white lines in both images are superimposed for clarity; they are not part of the data.)
Wraparound and Padding

**FIGURE 4.39** Padded lowpass filter in the spatial domain (only the real part is shown).

**FIGURE 4.40** Result of filtering with padding. The image is usually cropped to its original size since there is little valuable information past the image boundaries.
Image Correlation

\[ f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x + m, y + n) \]

**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.
Image Processing in the Frequency Domain

Frequency domain filtering operation

- Pre-processing
  - Input image
  - $f(x, y)$

- Fourier transform
  - $F(u,v)$

- Filter function
  - $H(u,v)$

- Inverse Fourier transform
  - $H(u,v)F(u,v)$

- Post-processing
  - Enhanced image
  - $g(x, y)$

FIGURE 4.5 Basic steps for filtering in the frequency domain.

- Pad original image to $P \times Q$
- Multiply padded image by $(-1)^{x+y}$
- Fourier transform $F(u,v)$
- Construct real, symmetric filter $H(u,v)$ of size $P \times Q$
- Compute $F(u,v)H(u,v)$
- Compute Re$[F^{-1} \{F(u,v)H(u,v)\}]$ and multiply by $(-1)^{x+y}$
- Crop to remove padding in result
Fourier Transform

Due to the short extrusion

**FIGURE 4.29** (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)
Simple Frequency Domain Filter

What if we simply multiply $F(0,0)$ by zero?

This is the minimal high-pass filter

FIGURE 4.29 (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)
Simple Frequency Domain Filtering

Multiplying $F(0,0)$ by zero makes the average image value zero— these are negative image values which cannot be displayed, but get cropped to zero!

FIGURE 4.30  
Result of filtering the image in Fig. 4.29(a) by setting to 0 the term $F(M/2, N/2)$ in the Fourier transform.
Gaussian Filters

Frequency domain filters

Corresponding spatial domain filters

Low-pass filters

High-pass filters

This is a Difference of Gaussians, called a DoG

FIGURE 4.37
(a) A 1-D Gaussian lowpass filter in the frequency domain.
(b) Spatial lowpass filter corresponding to (a).
(c) Gaussian highpass filter in the frequency domain.
(d) Spatial highpass filter corresponding to (c). The small 2-D masks shown are spatial filters we used in Chapter 3.
Gaussian Filters

Low-pass filter

- Blurs image
- Enhances detail; lowers contrast
- Addition of a preserves DC term

High-pass filters

- Low-pass filter

FIGURE 4.31 Top row: frequency domain filters. Bottom row: corresponding filtered images obtained using Eq. (4.7-1). We used $a = 0.85$ in (c) to obtain (f) (the height of the filter itself is 1). Compare (f) with Fig. 4.29(a).
1. Original frequency domain filter $H(f)$

2. Spatial domain filter $h(x) = \mathcal{F}^{-1}\{H(f)\}$

3. Padded spatial domain filter $h_p(x)$

4. Fourier transform of passed spatial domain filter $h_p(x)$

Discontinuities caused by padding

FIGURE 4.34
(a) Original filter specified in the (centered) frequency domain.
(b) Spatial representation obtained by computing the IDFT of (a).
(c) Result of padding (b) to twice its length (note the discontinuities).
(d) Corresponding filter in the frequency domain obtained by computing the IDFT of (c). Note the ringing caused by the discontinuities in (c). (The curves appear continuous because the points were joined to simplify visual analysis.)
Phase Filters

Results of multiplying ONLY the phase of an image by a constant

*FIGURE 4.35*
(a) Image resulting from multiplying by 0.5 the phase angle in Eq. (4.6-15) and then computing the IDFT. (b) The result of multiplying the phase by 0.25. The spectrum was not changed in either of the two cases.

MOST image processing filters operate ONLY on the magnitude and are called zero-phase-shift filters
Frequency Domain Filtering Example

a. Original image
b. Padded image
c. Multiply by \((-1)^{x+y}\)
d. Fourier transform \(F(u,v)\)
e. Centered Gaussian LPF \(H(u,v)\)
f. Compute \(F(u,v)H(u,v)\)
g. Compute \(F^{-1}\{F(u,v)H(u,v)\}\) and multiply by \((-1)^{x+y}\)
h. Crop to remove padding

Padded images often show dark edges

FIGURE 4.36
(a) An \(M \times N\) image, \(f\).
(b) Padded image, \(f_p\) of size \(P \times Q\).
(c) Result of multiplying \(f_p\) by \((-1)^{x+y}\).
(d) Spectrum of \(F_p\).
(e) Centered Gaussian lowpass filter, \(H\), of size \(P \times Q\).
(f) Spectrum of the product \(HF_p\).
(g) \(g_p\), the product of \((-1)^{x+y}\) and the real part of the IDFT of \(HF_p\).
(h) Final result, \(g\), obtained by cropping the first \(M\) rows and \(N\) columns of \(g_p\).
Fourier Transform of an Image

600x600 pixels

F(u,v)

FIGURE 4.38
(a) Image of a building, and
(b) its spectrum.
Sobel spatial/frequency domain filters

**Spatial domain Sobel, 3x3**

<table>
<thead>
<tr>
<th>-1</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Frequency domain Sobel**

Frequency domain filtering where \( h(x,y) \) was padded to 602x602 and centered prior to multiplying by \( F(u,v) \) in the frequency domain.

**FIGURE 4.39**
(a) A spatial mask and perspective plot of its corresponding frequency domain filter. (b) Filter shown as an image. (c) Result of filtering Fig. 4.38(a) in the frequency domain with the filter in (b). (d) Result of filtering the same image with the spatial filter in (a). The results are identical.

**Spatial domain filtering**
Ideal Low-pass Filter

\[ D_0 \text{ is picked based upon the fraction of image power to be removed} \]
If \( M=688 \) and, say, there are 688 pixels/inch each pixel in the Fourier transforms corresponds to \( \Delta u=1/\text{inch} \).
Filtering with Ideal LPF

Filtering the padded test image with an ILPF in the frequency domain

**Figure 4.42** (a) Original image. (b)-(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 90, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 15, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.
Computing the inverse Fourier transform of the ILPF shows ringing in the spatial domain which is clearly shown in Figure 4.42.
Butterworth LPF

For $n=1$ there is no ringing. The ringing will increase as $n$ increases.

$$H(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{2n}}$$
Filtering with Butterworth filter, $n=2$

FIGURE 4.45 (a) Original image. (b)-(f) Results of filtering using BPFs of order 2, with cutoff frequencies at the radii shown in Fig. 4.44. Compare with Fig. 4.42.
Spatial Butterworth filters

**FIGURE 4.46** (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding intensity profiles through the center of the filters (the size in all cases is $1000 \times 1000$ and the cutoff frequency is 5). Observe how ringing increases as a function of filter order.
Gaussian LPF

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

**FIGURE 4.47** (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of \( D_0 \).
Low-pass Filters

**TABLE 4.4**
Lowpass filters. $D_0$ is the cutoff frequency and $n$ is the order of the Butterworth filter.

<table>
<thead>
<tr>
<th>Ideal</th>
<th>Butterworth</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H(u, v) = \begin{cases} 1 &amp; \text{if } D(u, v) \leq D_0 \ 0 &amp; \text{if } D(u, v) &gt; D_0 \end{cases}$</td>
<td>$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$</td>
<td>$H(u, v) = e^{-D(u,v)/2D_0^2}$</td>
</tr>
</tbody>
</table>
Filtering with Gaussian LPF

Figure 4.48 (a) Original image. (b)–(f) Results of filtering using GLPFs with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Figs. 4.42 and 4.45.
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.

No filtering. Broken characters are difficult to recognize.

Filtered with a GLPF with $D_0=80$. Characters are fuller and filled in.

FIGURE 4.49
(a) Sample text of low resolution (note broken characters in magnified view).
(b) Result of filtering with a GLPF (broken character segments were joined).