Overview Module 03

* Image Frequency Domain
  – Fourier transform
  – Properties

* Image filtering based on frequency domain
  – Various filter types and their response (Gaussian, Laplace..)
  – Noise reduction by frequency domain filtering
Module 03 – Part 1
Frequency domain representation

Sampling and Fourier transform, 2-D extensions, convolution, aliasing, filtering concept

Image waveform concept of Fourier

Image: essentially a 2-D signal waveform

* Signal: decomposed in a sum of components
  - Fourier found in 1807 that any periodic or ‘limited energy’ waveform: can be a weighted sum of sines and cosines

FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier’s idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.
Fourier series and impulses

Characterization of a video signal $f(t)$

* Assume function $f(t)$ with period $T$, which can be expressed as sum of sines and cosines, with coefficients $(n=0, +/- 1, +/- 2, \ldots)$

$$f(t) = \sum_{n=-\infty}^{+\infty} c_n e^{\frac{j2\pi n t}{T}} \quad c_n = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} f(t) e^{-j2\pi n t/T} dt$$

* 2. Unit impulses

$$\delta(t) = \begin{cases} 0 & (t \neq 0) \\ \delta(0) = \infty \end{cases}; \quad \int_{-\infty}^{+\infty} \delta(t) dt = 1$$

* 3. Sifting process

$$\int_{-\infty}^{+\infty} f(t) \delta(t-t_0) dt = f(t_0)$$

* Exercise: derive discrete versions!

Signal is series of (weighted) impulses

Characterization of a video signal

* Assume signal $s(t)$ with sampling period $T$, which can be expressed as sum impulses, with signal coefficients $(n=0, +/- 1, +/- 2, \ldots)$

$$s_{\Delta T}(t) = \sum_{n=-\infty}^{+\infty} s_n \delta(t - n\Delta T)$$

* If weights are $s_n = 1$, then equal to sampling function!
Fourier transform (one cont. variable)

Fourier transform of a signal $f(t)$
* Assume function $f(t)$ with of time $t$, which
$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j2\pi\omega t} dt$$
* 2. Inverse Fourier transform
$$f(t) = \int_{-\infty}^{+\infty} F(\omega)e^{j2\pi\omega t} d\omega$$

* 3. The equations form a transform pair. Note that with Euler, it also holds that
$$e^{j2\pi\omega t} = \cos(2\pi ft) + j\sin(2\pi ft)$$

Note that $\omega$ is replaced here to frequency $f$ in many books!
If $f(t)$ is real, then the transform is complex!

---

Fourier transform (cont.) - Example

$$F(w) = \int_{-W/2}^{+W/2} A e^{-j2\pi W/2} d\omega = -A \left[ e^{-j2\pi W/2} \right]_{-W/2}^{+W/2} = AW \frac{\sin(\pi W)}{\pi W}$$

Frequency spectrum is magnitude of transform!

µ = f in the drawing!

---

FIGURE 4.4 (a) A simple function; (b) its Fourier transform; and (c) the spectrum. All functions extend to infinity in both directions.
Sampling of a signal – (1)

\[ f_n = f(n\Delta T) = \sum_{n=-\infty}^{+\infty} f(t) \delta(t - n\Delta T) \]

- **Sampling of**
  - **continuous** signal \( f(t) \)
  - **time period** \( \Delta T \) (or \( T_s \))
  - results in
  - **weighted** impulse train

**Figure 4.5**

- A continuous function.
- Train of impulses used to model the sampling process.
- Sampled function formed as the product of (a) and (b).
- Sampled values obtained by integration and using the sifting property of the impulse. (The dashed line in (c) is shown for reference. It is not part of the data.)

Sample speed such that spectra fit!

- **Signal should be bandwidth-limited**

**Figure 4.6**

- Fourier transform of a band-limited function.
- Transforms of the corresponding sampled function under the conditions of over-sampling, critically-sampling, and under-sampling, respectively.

\( \mu = f \) in the drawing!
Sampling of a signal – (3) Theorem

Reconstruction is possible if one spectrum can be extracted, thus

\[ \frac{1}{2\Delta T} > 2f_{\text{max}} \]

Sampling at 2x \( f_{\text{max}} \) is called the Nyquist rate.

Sampling Theorem (& Nyquist rate)

Assume here \( H(f) = \Delta T \)

\[-f_{\text{max}} \leq f \leq +f_{\text{max}}\]

Note that \( \mu \) is frequency in the drawing!
Under-Sampling

Under-sampling gives overlapping frequency areas

- HF becomes LF!

- Overlap is called aliasing

- Aliasing always occurs but can be suppressed with low-pass pre-filters

Discrete Fourier Transform (one variable)

Discrete Fourier Transform of a signal \( f(t) \) 

* Assume function \( f(t) \) sampled at time \( \Delta T \)

\[
F_m = \sum_{n=0}^{M-1} f_n e^{-j2\pi mn/M} \quad m = 0,1,2,...,M-1
\]

* 2. Inverse Discrete Fourier transform

\[
f_n = \sum_{m=0}^{M-1} F_m e^{j2\pi mn/M} \quad n = 0,1,2,...,M-1
\]

* 3. This can be derived with taking \( M \) samples in interval of frequency \( 0 \leq f \leq \frac{1}{\Delta T} \); \( m = 0,1,2,...,M-1 \); \( f = \frac{m}{M\Delta T} \)
1-D Discrete Fourier Transform (mod. Not.)

**Discrete Fourier Transform of an image signal** $I(x, y)$

* Assume function $I(t)$ sampled at discr. times $x \Delta T$ and discr. frequency $u$

$$F(u) = \sum_{x=0}^{M-1} f(x)e^{-j2\pi xu/M} \quad u = 0,1,2,\ldots,M-1$$

* 2. **Inverse Discrete Fourier transform**

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u)e^{j2\pi xu/M} \quad x = 0,1,2,\ldots,M-1$$

* 3. This is the common notation in many books and remainder of the course!

---

**Properties of 1-D DFT**

* 1. Both DFT and IDFT are **infinitely periodic**

$$F(u) = F(u + kM) \quad f(x) = f(x + kM)$$

* 2. Assuming that $f(x)$ is real and $m$ samples are taken, then the DFT gives a set of $m$ complex numbers

* 3. It can be shown that the discrete equivalent of **convolution** is

$$f(x) \otimes h(x) = \sum_{m=0}^{M-1} f(m)h(x-m)$$

* 4. If $f(x)$ consists of $M$ samples of $f(t)$ taken $\Delta T$ units apart, then period duration $T=M\Delta T$ and $\Delta u=1/(M\Delta T)=1/T$
1-D DFT computation example

1. Compute $F(0)$, $F(1)$ etc. of the DFT
   - $F(0)=1+2+4+4=11$, $F(1)=-3+2j$,
   - $F(2)=-(1+0j)$, $F(3)=-(3+2j)$
2. The inverse DFT gives the computation of $f(0)$
   - $f(0)=1/4[ 11-3+2j-1-3-2j ]=1$

Extensions to 2-D DFT – (1)

1. Sampling over 2-D area (image) instead of signal and impulse train becomes 2-D signal
2. DFT: summing over pixels $(x, y)$, and 2 param’s $m$, $n$
3. DFT periodic in 2-D frequency domain, param’s $(u, v)$
4. 2-D sampling theorem: 2 constraints for band-limitations

$\frac{1}{\Delta T} > 2 f_{h\text{max}}$; $\frac{1}{\Delta Z} > 2 f_{v\text{max}}$
Extensions to 2-D DFT – (2)

* 1. Ideal footprint of 2-D LPF becomes rectangular
* 2. Aliasing becomes 2-D pattern (e.g. Moire)

\[ \mu = f_h \quad \text{and} \quad \nu = f_v \]

in the drawings!

Extensions to 2-D DFT – (3) / Moire

FIGURE 4.15
Two-dimensional Fourier transforms of (a) an over-sampled, and
(b) under-sampled band-limited function.

FIGURE 4.20
Examples of the moiré effect. These are ink drawings, not
digitized patterns. Superimposing one pattern on
the other is equivalent mathematically to multiplying the
patterns.
Extensions to 2-D DFT – (4) / Moire

2-D Discrete Fourier Transform

※ 2-D Discrete Fourier Transform of image signal \( f(x, y) \) for \( 0 \leq u \leq M-1 \) and \( 0 \leq v \leq N-1 \) is

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

※ 2. 2-D Inverse Discrete Fourier Transform

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

※ 3. The above equations hold for an image of size \( M \times N \) pixels and together they form a transform pair
Properties of 2-D DFT – (1)

* **2-D Spatial and frequency intervals** using image signal \( f(x, y) \) of size \( M \times N \)

\[
\Delta u = \frac{1}{M \Delta u}; \quad \Delta v = \frac{1}{N \Delta v}
\]

* **Translation and rotation**

\[
f(x, y)e^{+j2\pi(u_0/M + v_0/N)} \iff F(u - u_0, v - v_0)
\]

\[
f(x - x_0, y - y_0) \iff F(u, v)e^{-j2\pi(u_0/M + v_0/N)}
\]

* Likewise, a rotation over an angle \( \theta \) \( f(x, y) \) gives an equal rotation of \( F(u, v) \) (this can be proven with polar coordinates with \( x = r \cos(\theta) \) etc., and \( u = r \cos(\varphi) \), etc.)

Properties of 2-D DFT – (2)

* **2-D Periodicity of the DFT spectra**

\[
F(u, v) = F(u + k, v) = F(u, v + k) = etc.
\]

\[
f(x, y) = f(x + k, y) = f(x, y + k) = etc
\]

* As a special case (use Euler rule for this with \( u_0 = M/2 \))

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

* As a result, the spectrum shifts such that the \( F(0,0) \) is now at the origin!
Properties of 2-D DFT – (3)

Centering

Multiplication with \((-1)^{x+y}\) relocates \(F(0,0)\) to the origin.

\[ \mu=f \text{ in the drawings!} \]

Properties of 2-D DFT – (4)

Symmetry

TABLE 4.1 Some symmetry properties of the 2-D DFT and its inverse. \(R(u, v)\) and \(I(u, v)\) are the real and imaginary parts of \(F(u, v)\), respectively. The term \(complex\) indicates that a function has nonzero real and imaginary parts.

<table>
<thead>
<tr>
<th>Spatial Domain (\mathbb{R})</th>
<th>Frequency Domain (\mathbb{C})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) (f(x, y)) real (\iff)</td>
<td>(F(u, v) = F(-u, -v))</td>
</tr>
<tr>
<td>2) (f(x, y)) imaginary (\iff)</td>
<td>(F(-u, -v) = -F(u, v))</td>
</tr>
<tr>
<td>3) (f(x, y)) real (\iff)</td>
<td>(R(u, v)) even; (I(u, v)) odd</td>
</tr>
<tr>
<td>4) (f(x, y)) imaginary (\iff)</td>
<td>(R(u, v)) odd; (I(u, v)) even</td>
</tr>
<tr>
<td>5) (f(-x, -y)) real (\iff)</td>
<td>(F^*(u, v)) complex</td>
</tr>
<tr>
<td>6) (f(-x, -y)) complex (\iff)</td>
<td>(F(-u, -v)) complex</td>
</tr>
<tr>
<td>7) (f^*(x, y)) complex (\iff)</td>
<td>(F^*(-u, -v)) complex</td>
</tr>
<tr>
<td>8) (f(x, y)) real and even (\iff)</td>
<td>(F(u, v)) real and even</td>
</tr>
<tr>
<td>9) (f(x, y)) imaginary and odd (\iff)</td>
<td>(F(u, v)) imaginary and odd</td>
</tr>
<tr>
<td>10) (f(x, y)) imaginary and even (\iff)</td>
<td>(F(u, v)) imaginary and even</td>
</tr>
<tr>
<td>11) (f(x, y)) imaginary and odd (\iff)</td>
<td>(F(u, v)) real and odd</td>
</tr>
<tr>
<td>12) (f(x, y)) complex and even (\iff)</td>
<td>(F(u, v)) complex and even</td>
</tr>
<tr>
<td>13) (f(x, y)) complex and odd (\iff)</td>
<td>(F(u, v)) complex and odd</td>
</tr>
</tbody>
</table>

\(^1\)Recall that \(x, y, u, \) and \(v\) are discrete (integer) variables, with \(x, y\) in the range \([0, M-1]\), and \(u, v\) in the range \([0, N-1]\). To say that a complex function is even means that its real and imaginary parts are even, and similarly for an odd complex function.
Fourier spectrum and phase 2-D DFT

* 2-D DFT is complex, it can be expressed in polar form

\[ F(u, v) = |F(u, v)| e^{i\phi(u, v)} \]

* Where the magnitude is the Fourier spectrum and \( \phi \) the phase angle

\[ |F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)} \]

\[ \phi(u, v) = \arctan \left( \frac{I(u, v)}{R(u, v)} \right) \]

* Finally, the power spectrum is

\[ P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v) \]

Fourier spectrum and phase – Examples

* Typically, \( f(x, y) \) is real, which implies that 2-D DFT spectrum is even symmetric about the origin

\[ |F(u, v)| = |F(-u, -v)| \]

* and \( \phi \) the phase angle is odd symmetric about the origin

\[ \phi(u, v) = \phi(-u, -v) \]

* Finally, note that

\[ |F(0,0)| = |MN \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)| = MN |f(x, y)| \]
**Fourier spectrum – Example 1**

(a) Image. 
(b) Spectrum showing bright spots in the four corners. 
(c) Centered spectrum. 
(d) Result showing increased detail after a log transformation. The zero crossings of the spectrum are closer in the vertical direction because the rectangle in (a) is longer in that direction. The coordinate convention used throughout the book places the origin of the spatial and frequency domains at the top left.

---

**Fourier spectrum – Example 2a**

(a) The rectangle in Fig. 4.24(a) translated, and (b) the corresponding spectrum. 
(c) Rotated rectangle, and (d) the corresponding spectrum. The spectrum corresponding to the translated rectangle is identical to the spectrum corresponding to the original image in Fig. 4.24(a).
Fourier spectrum – Example 2b Phase

**FIGURE 4.26** Phase angle array corresponding (a) to the image of the centered rectangle in Fig. 4.24(a), (b) to the translated image in Fig. 4.25(a), and (c) to the rotated image in Fig. 4.25(c).

---

**TABLE 4.2** Summary of DFT definitions and corresponding expressions.

<table>
<thead>
<tr>
<th>Name</th>
<th>Expression(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Discrete Fourier transform (DFT) of ( f(x, y) )</td>
<td>( F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(xu/M+yv/N)} )</td>
</tr>
<tr>
<td>2) Inverse discrete Fourier transform (IDFT) of ( F(u, v) )</td>
<td>( f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(xu/M+yv/N)} )</td>
</tr>
<tr>
<td>3) Polar representation</td>
<td>( F(u, v) =</td>
</tr>
</tbody>
</table>
| 4) Spectrum | \( |F(u, v)| = \left[ R^2(u, v) + I^2(u, v) \right]^{1/2} \)  
| | \( R = \text{Real}(F); \quad I = \text{Imag}(F) \) |
| 5) Phase angle | \( \phi(u, v) = \tan^{-1} \left[ \frac{I(u, v)}{R(u, v)} \right] \) |
| 6) Power spectrum | \( P(u, v) = |F(u, v)|^2 \) |
| 7) Average value | \( \overline{f}(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = \frac{1}{MN} \overline{F}(0, 0) \) |

(Continued)
Summary 2-D DFT Properties – (3)

<table>
<thead>
<tr>
<th>Name</th>
<th>DFT Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Symmetry properties</td>
<td>See Table 4.1</td>
</tr>
<tr>
<td>2) Lincarity</td>
<td>$af(x, y) + bf(x, y) \iff aF(u, v) + bF(u, v)$</td>
</tr>
<tr>
<td>3) Translation (general)</td>
<td>$f(x, y)e^{j2\pi m(Mx + ny/M)} \iff F(u - um, v - yn)$</td>
</tr>
<tr>
<td>4) Translation to center of the frequency rectangle, $(M/2, N/2)$</td>
<td>$f(x - x_0, y - y_0) \iff F(u, v)e^{-j2\pi (um + yn/N)}$</td>
</tr>
<tr>
<td>5) Rotation</td>
<td>$f(r, \theta + \theta_0) \iff F(\omega, \varphi) \iff F(\omega \cos \theta - \varphi \sin \theta, \omega \sin \theta + \varphi \cos \theta)$</td>
</tr>
<tr>
<td>6) Convolution theorem$^1$</td>
<td>$f(x, y) \ast h(x, y) \iff F(u, v)H(u, v)$</td>
</tr>
</tbody>
</table>

(Continued)
Summary 2-D DFT Properties – (4)

* And remember that due to Euler’s formula and the cosine/sine rules:

\[
\sin(2\pi u_0 x + 2\pi v_0 y) \iff \frac{j}{2} \left[ \delta(u + Mu_0, v + Nu_0) - \delta(u - Mu_0, v - Nu_0) \right]
\]

* And

\[
\cos(2\pi u_0 x + 2\pi v_0 y) \iff \frac{1}{2} \left[ \delta(u + Mu_0, v + Nu_0) + \delta(u - Mu_0, v - Nu_0) \right]
\]

* Finally, note that

\[
\delta(x, y) \iff 1
\]

Summary 2-D DFT Properties – (5)

* Exercise: derive one of those properties in a step by step fashion
Module 03 – Part 2
Image Filtering based on DFT

Filtering concept using DFT, filter types, special cases

Frequency-domain filtering fundament

* Filtering techniques in the frequency domain are based on modifying the Fourier transform

* Note that each value of $F(u,v)$ contains information of all image samples!

* If a filter has a filter transfer function, we exploit that

$$g(x, y) = IDFT[H(u, v)F(u, v)]$$

Where $F$, $H$ and $g$ are all matrices of $M \times N$. Assume that $H$ and $F$ are centered, requiring pre-processing!
Necessity of padding with zeros – (1)

* Without any measures, the use of the DFT creates a wraparound error when periodicity is created

* The solution is to enlarge the the window with extra samples, such as **padding with zeros** (creates a border)

* **Padding can be used to decrease ringing** in the frequency domain, employing the extended filter impulse response in the time domain.

---

Necessity of padding zeros – (2)

**Example**

---

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 DFT of (c). Note the ringing caused by the discontinuities in (c). (The curves appear continuous because the points were joined to simplify visual analysis.)
### Summary of steps for DFT-based filters

1. Determine padding parameters, typically \( P = 2M \) and \( Q = 2N \).
2. Form padded image \( f_p(x, y) \) of size \( P \times Q \) appending zeros.
3. Multiply padded image with \((-1)^{x+y}\) for centering spectrum.
4. Compute the DFT, \( F(u, v) \), of the centered padded image.
5. Generate a real, symmetric filter, \( H(u, v) \), of size \( P \times Q \) and center in the middle, and compute \( G(u, v) = H(u, v)F(u, v) \).
6. Obtain the processed image with the IDFT of \( G(u, v) \) and take the real part, and re-center it.
7. Extracting the \( g(x, y) \) result by selecting the \( M \times N \) region from the top-left quadrant.

### Example of DFT-based filters / Gaussian

\[
h(x) = \sqrt{2\pi}\sigma\exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad \leftrightarrow \quad H(u) = \exp\left(-\frac{u^2 + v^2}{2\sigma^2}\right)
\]

Both real functions!

---

\( h(x) \) is 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.
Example of DFT-based filters / Gaussian

Sobel filter as DFT-based filter example
**Ideal 2-D LPF filter – (1) Concept**

- Passes all frequencies within circle of radius $D_0$ : $H(u,v)=1$
- Rejects all frequencies outside circle with radius $D_0$ : $H(u,v)=0$

$$D(u,v) = \left[(u - P / 2)^2 + (v - Q / 2)^2\right]^{1/2}$$

**Ideal 2-D LPF filter – (2)**

Performance comparison shows

- this is not very practical, as
- most detail is removed in only 13% of the power
- and ringing occurs at already 2% removal
Ideal 2-D LPF filter – (3) Explanation

1. Blur comes from the main lobe in impulse response
2. Ringing comes from the side lobes in impulse response
3. Spread of sinc function is inversely proportional to pass-band of $H(u, v)$. (Discuss the trade-off)

Smooth 2-D LPF filter – (1) Gaussian

1. Establish smooth roll-off in transfer function $H(u, v)$
2. Ringing comes from the side lobes in impulse response

$$H(u, v) = e^{-D^2(u, v) / 2D_0^2} \quad (note: \sigma = D_0)$$
1. Blur can be regulated by variance
2. Ringing is absent
3. Spread of sinc function is smooth
4. Good visual results

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.

2-D HPF filters – various forms

\[ H_{HP}(u, v) = 1 - H_{LP}(u, v) \]

\[ H_{HP}(u, v) = 1 \quad : D(u, v) > D_0 \]
\[ H_{HP}(u, v) = 0 \quad : D(u, v) \leq D_0 \]

HPF is opposite of LPF filter!

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.

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.
Smooth 2-D HPF filter – cf. Gaussian

\[ H_{HP}(u, v) = 1 - e^{-D^2(u, v) / 2D_0^2} \quad (\text{note: } \sigma = D_0) \]

2-D HPF filters – perform.

Ideal HPF vs. Gaussian HPF performance

**FIGURE 4.53** Spatial representation of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding intensity profiles through their centers.

**FIGURE 4.54** Results of highpass filtering the image in Fig. 4.41(a) using an IHPF with \( D_0 = 30, 60, \) and 160.

**FIGURE 4.56** Results of highpass filtering the image in Fig. 4.41(a) using a GHPF with \( D_0 = 30, 60, \) and 160, corresponding to the circles in Fig. 4.41(b). Compare with Figs. 4.54 and 4.55.
Laplacian filtering - Concept

1. Laplacian can be implemented in frequency domain with
   \[ H(u,v) = -4\pi^2(u^2 + v^2) \]

2. Or, with respect to center of frequency rectangle
   \[ H(u,v) = -4\pi^2((u-P/2)^2 + (v-Q/2)^2) = -4\pi^2D^2(u,v) \]

3. Furthermore, it holds that for the filter in the freq. domain
   \[ \nabla^2 f(x,y) = IDFT\{H(u,v)F(u,v)\} \]

4. Enhancement with subjective sharpness is obtained when adding this term (note the scaling), so that
   \[ g(x,y) = f(x,y) + c\nabla^2 f(x,y) \]
   \[ g(x,y) = IDFT\{F(u,v) - H(u,v)F(u,v)\} = IDFT\{[1 - H(u,v)]F(u,v)\} = IDFT\{[1 + 4\pi^2D^2(u,v)]F(u,v)\} \]

Laplacian filtering - performance

**Figure 4.58**
(a) Original, blurry image.
(b) Image enhanced using the Laplacian in the frequency domain. Compare with Fig. 3.38(e).
Unsharp masking & highboost filtering

1. Concept: sharpen images by **subtracting an unsharp image** from the original. The difference is **the mask**.

2. Steps are:
   1. Blur the original image
   2. Subtract the blurred image from original (difference is the mask)
   3. Add mask to the original

3. More formally, the specification is
   
   \[ g_{\text{mask}}(x, y) = f(x, y) - \bar{f}(x, y) \]
   
   \[ g(x, y) = f(x, y) + k \cdot g_{\text{mask}}(x, y) \]

4. If \( k=1 \), it is unsharp masking, if \( k>1 \), we have highboost filtering, with if \( k<1 \), we have high-freq. emphasis.

High-frequency emphasis filtering

1. For the unsharp mask, we can take the freq.-domain approach

   \[ f_{LP}(x, y) = IDFT[H_{LP}(u, v)F(u, v)] \]

2. Function \( f_{LP}(x, y) \) is a low-pass smoothed image from the filtering with \( H_{LP}(u, v) \).

3. The unsharp masking equation from the previous slide becomes now **in the frequency domain**

   \[ g(x, y) = IDFT\{[1 + k \cdot (1 - H_{LP}(u, v))]F(u, v)\} \]

4. This result can be respecified as a **high-pass filter result**

   \[ g(x, y) = IDFT\{[1 + k \cdot H_{HP}(u, v)]F(u, v)\} \]

\( \text{High-frequency emphasis filter} \)
High-freq. emphasis filtering / Example

1. For more general form, take coefficients \( k_1, k_2 \) such that

\[
g(x, y) = IDFT\{[k_1 + k_2 * H_{hp}(u, v)]F(u, v)\}
\]

2. Where \( k_1 > 0 \) gives controls of the offset of the origin and \( k_2 > 0 \) the high-frequency contribution

3. In medical applications, ringing artifacts from filters are not accepted. Because spatial and freq.-domain Gaussian filters are transform pairs, these filters give smooth response and avoid ringing

4. In the following example, HF emphasis filtering helps

Gaussian filter with \( D_0 = 40 \)

HF emphasis & histogram equaliz.