Basic General Filtering Algorithm

- Multiply image \( f(x,y) \) by \((-1)^{x+y}\)
- Compute DFT, \( F(u,v) \)
- Multiply by filter function \( G(u,v) = F(u,v) H(u,v) \)
- Compute inverse DFT, \( g(x,y) \)
- Retain only real part of \( g(x,y) \)
- Multiply image \( g(x,y) \) by \((-1)^{x+y}\)
- Might need to do some shifting or scaling of pixel values

"Ideal" Low Pass Filter

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

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

"Ideal" Low Pass Filter

frequency domain  |  spatial domain

original  |  low pass  |  high pass
Butterworth Low Pass Filter

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

Butterworth Low Pass Filter

frequency domain  
spatial domain

Butterworth Low Pass Filter

original  
low pass  
high pass

Gaussian Low Pass Filter

\[ H(u, v) = \exp\left(-\frac{D^2(u, v)}{2D_0^2}\right) \]
Gaussian Low Pass Filter

- Frequency domain
- Spatial domain

Gaussian Low Pass Filter

- Original
- Low pass
- High pass

Sharpening Filters

The Laplacian in the Frequency Domain
Homomorphic Filtering

- We want to reduce the effects of uneven illumination while leaving reflectance unchanged.
- In general, the illumination function contains lower spatial frequencies, and the reflectance function contains higher spatial frequencies.
- We have a problem, because the image is the \textit{product} of the illumination and reflectance functions.

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

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

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

We hope to be left with this:

\[
F[\log g(x, y)] \approx F[\log r(x, y)]
\]

\[
g(x, y) \approx r(x, y)
\]
Spectrum of a Sinusoid

Spectrum of Two Superimposed Square Waves