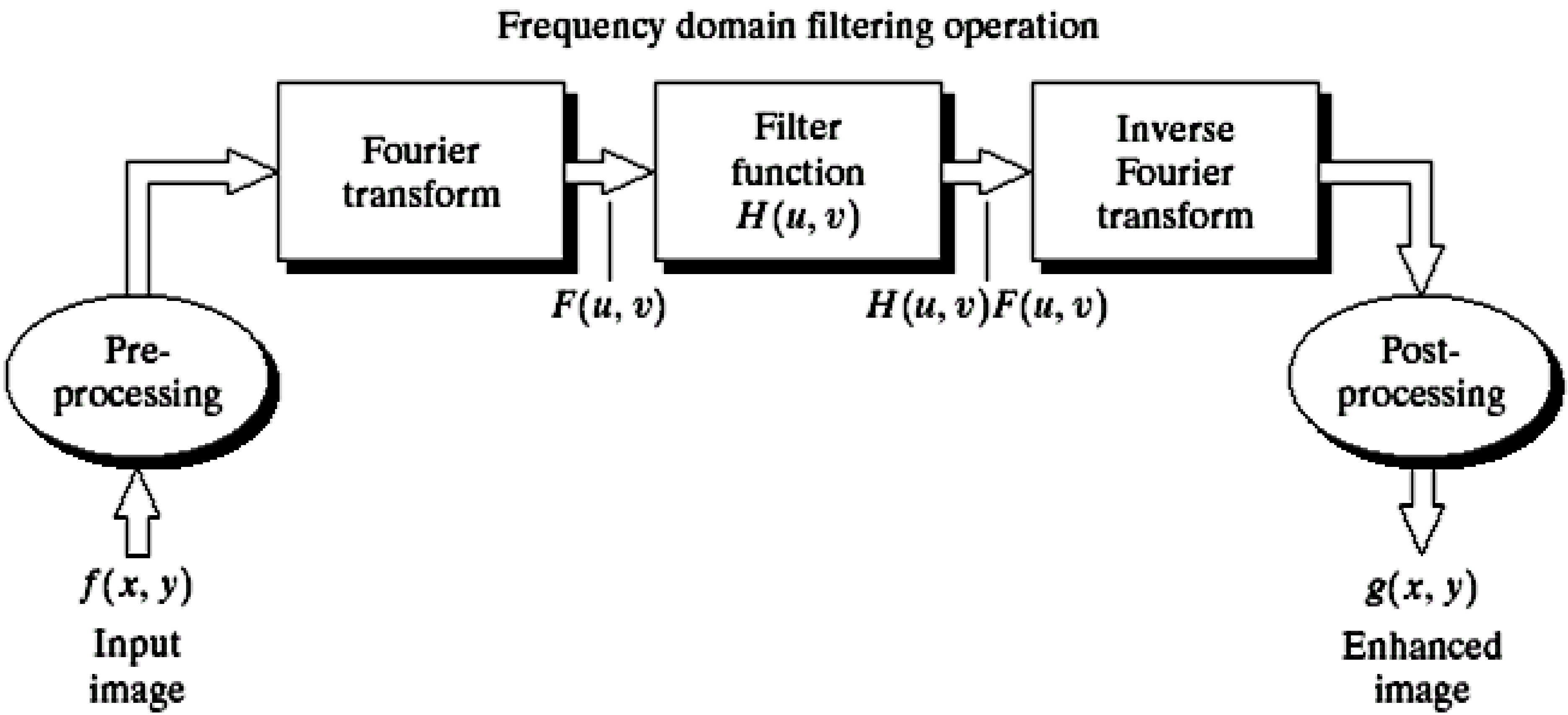
Image Processing
In Frequency Domain

Frequency-Domain Filtering



$$G(u,v) = H(u,v)F(u,v)$$

Filtered image: $\mathcal{F}^{-1}[G(u,v)]$

2-D Convolution (Discrete Form)

2-D discrete form of convolution:

$$f(x,y)*h(x,y)=\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}f(m,n)h(x-m,y-n)$$

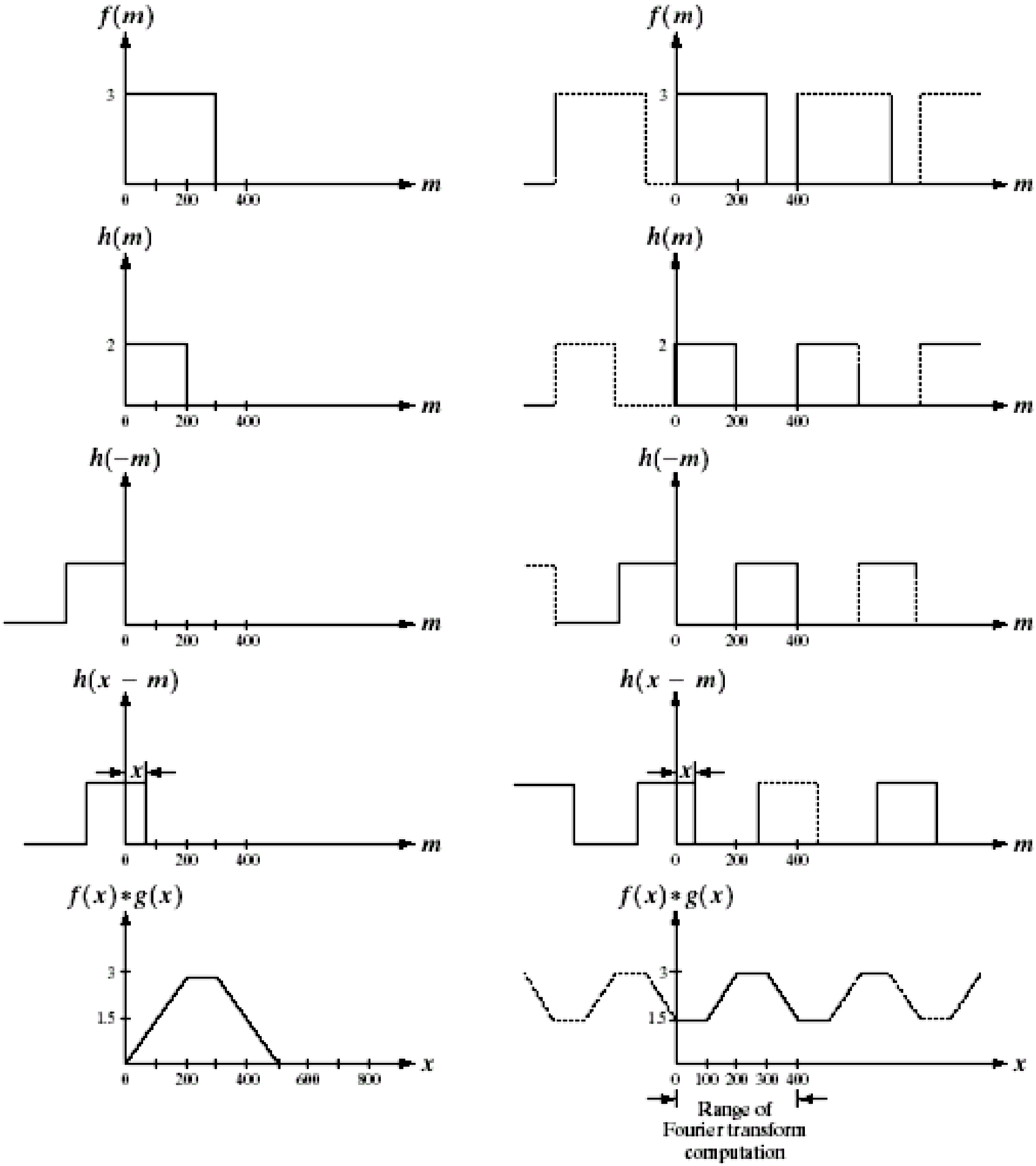
Convolution vs. spatial filtering:

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b}f(x+s,y+t)h(s,t)$$

$$=\sum_{m=x-a}^{x+a}\sum_{n=y-b}^{y+b}f(m,n)h(m-x,n-y)$$

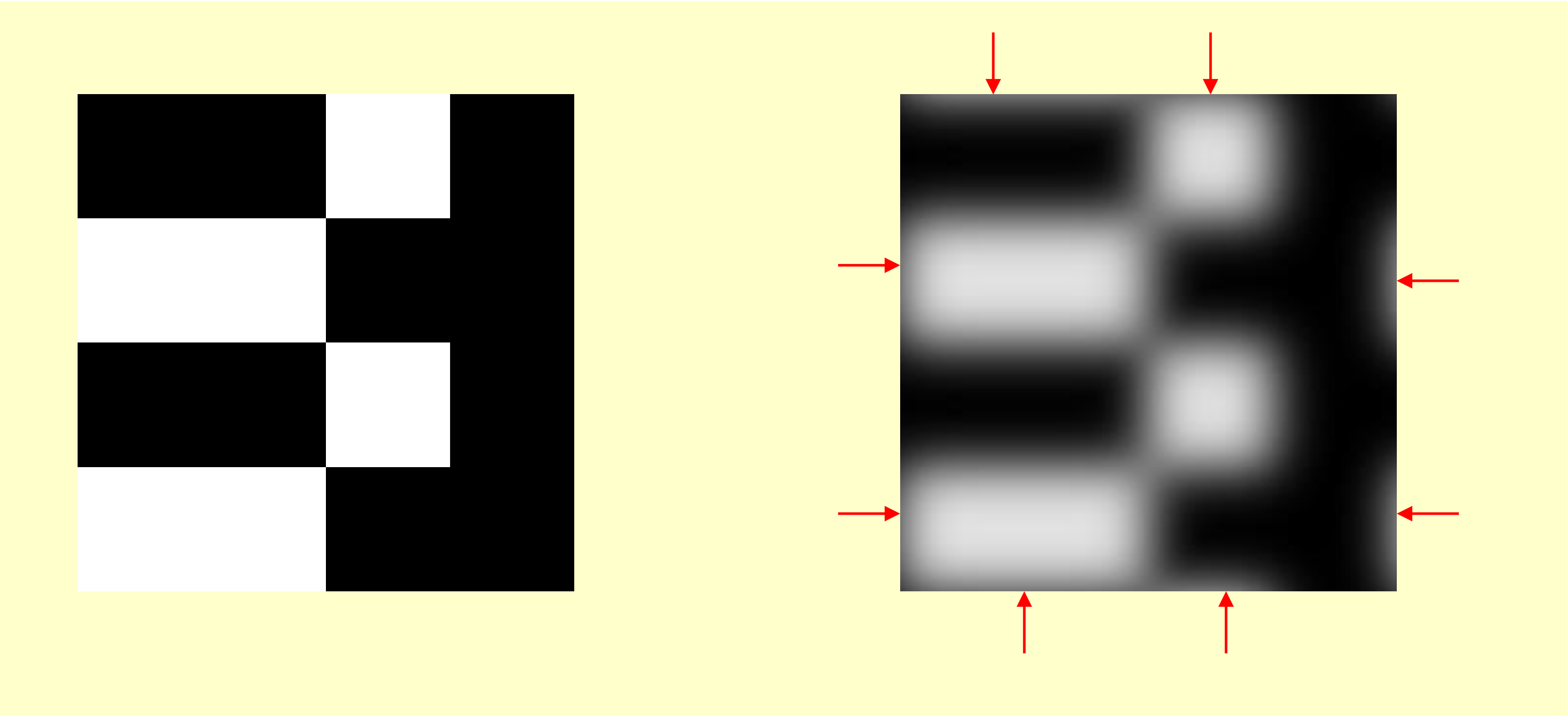
Wraparound Error in Convolution

When using frequency-domain filtering, the inherent periodicity in $f(x)$ becomes a problem:



Wraparound Error in Convolution

When using frequency-domain filtering, the inherent periodicity in $f(x)$ becomes a problem. (The corresponding spatial filter is applied to not a single image, but the mosaic of all the repeated copies of the image.)



Example with Gaussian LPF

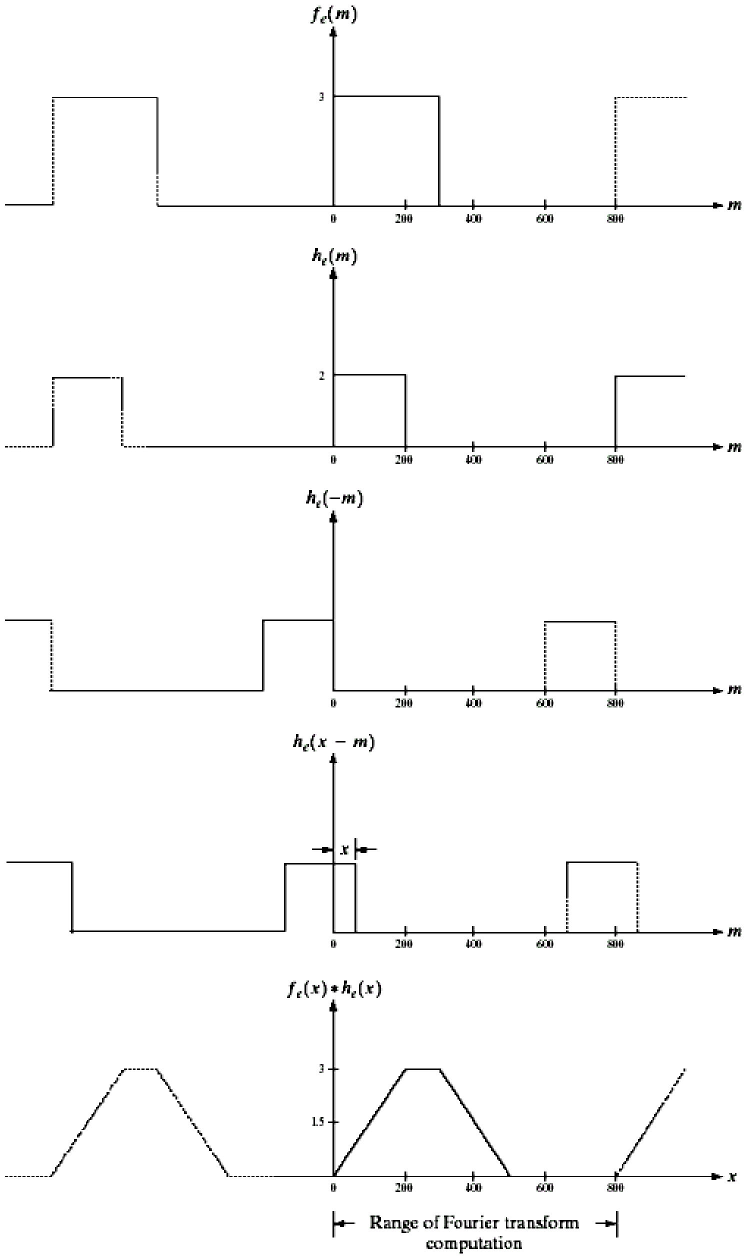
Padding and Convolution

Solution: zero-padding

$$P \geq A + B - 1$$

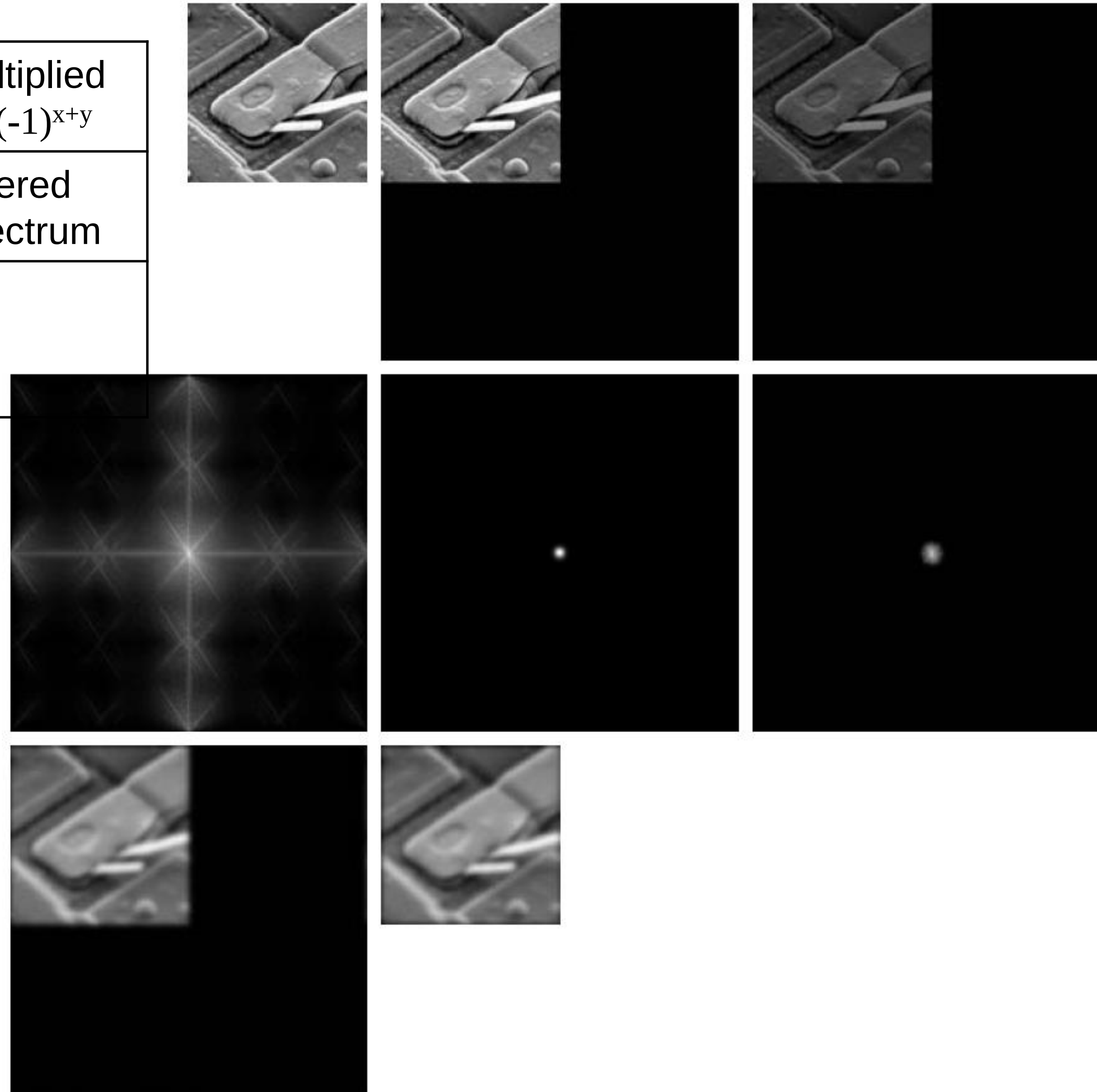
A: # samples in $f(x)$
B: # samples in $h(x)$
P: # samples in both $f(x)$ and $h(x)$ after padding

Typically we just double the size of the image in both directions.

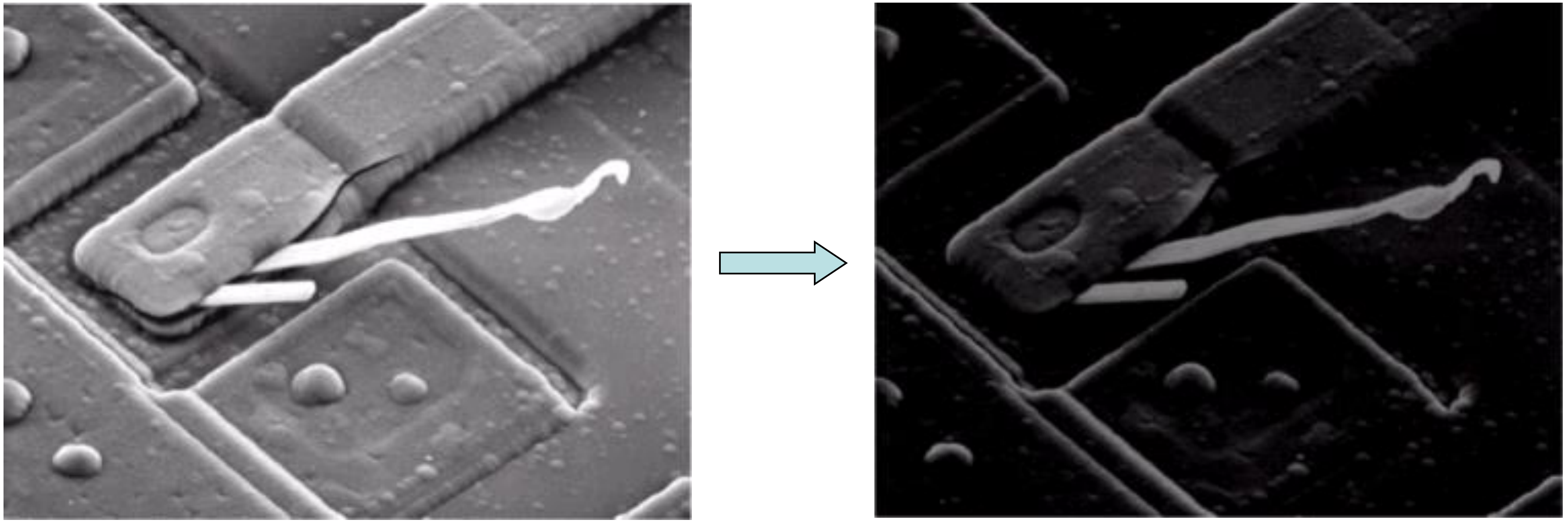


Padding and Convolution

Original	Zero-padded	Multiplied by $(-1)^{x+y}$
Spectrum	LPF	Filtered spectrum
IDFT multiplied by $(-1)^{x+y}$	Final	



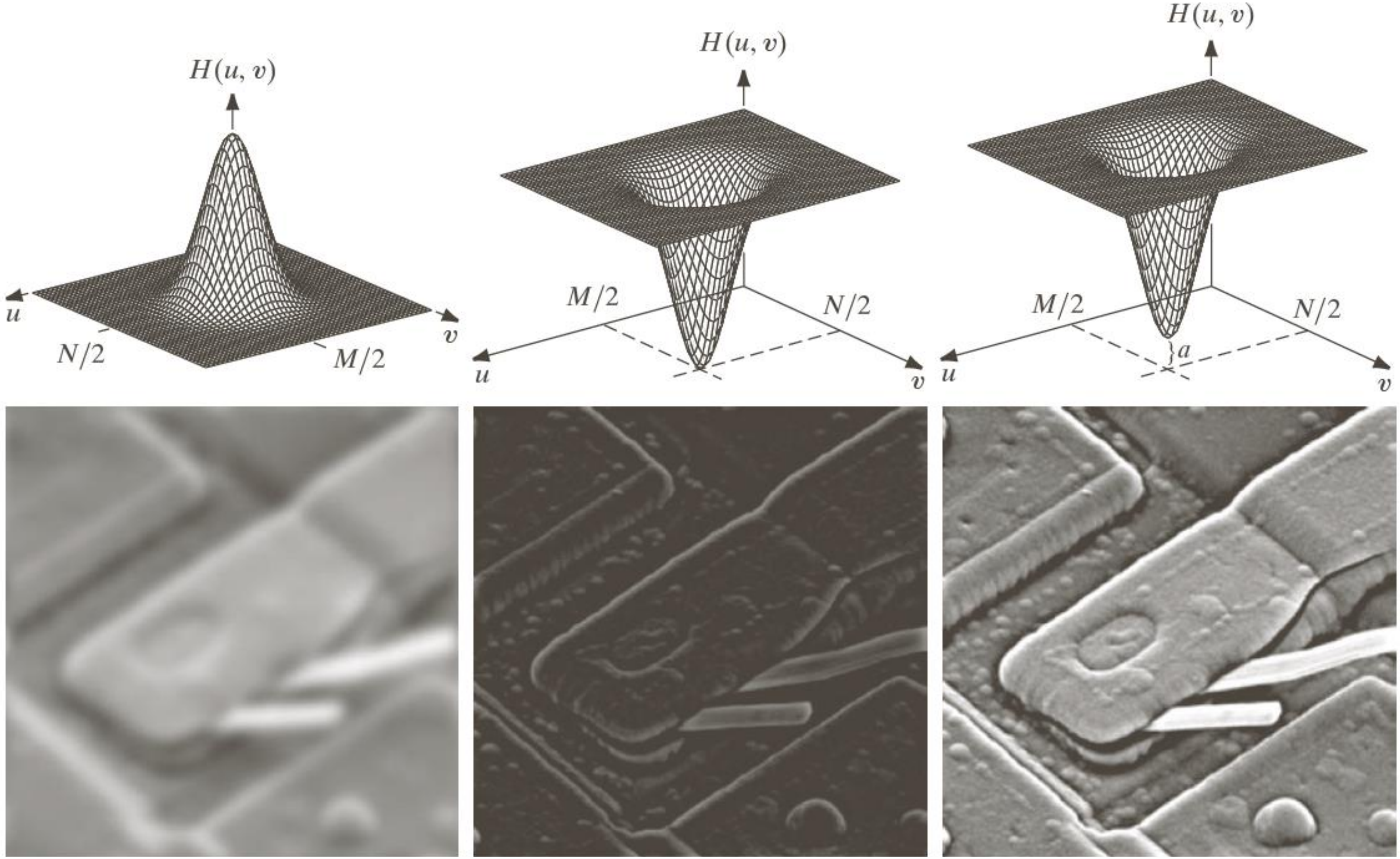
A Basic Filter



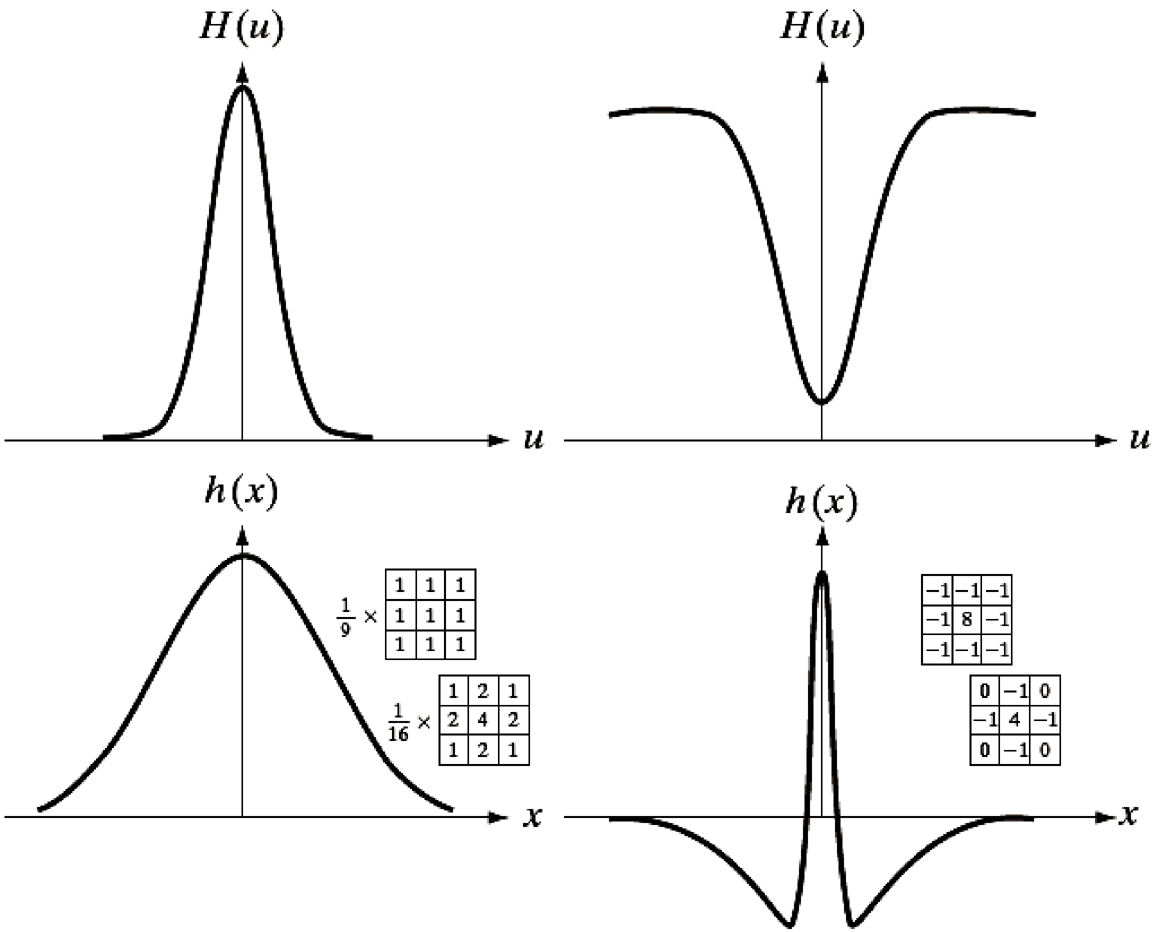
DC term (after centering)

Notch filter: $H(u,v) = \begin{cases} 0, & \text{if } (u,v) = (M/2, N/2) \\ 1, & \text{otherwise} \end{cases}$

High-Pass and Low-Pass Filters



Filter Correspondence



The **impulse response** of a filter is $h(x,y)$. This is the filtered result of a single white pixel in a black ground. A filter can also be characterized by its impulse response.

Choosing the Method

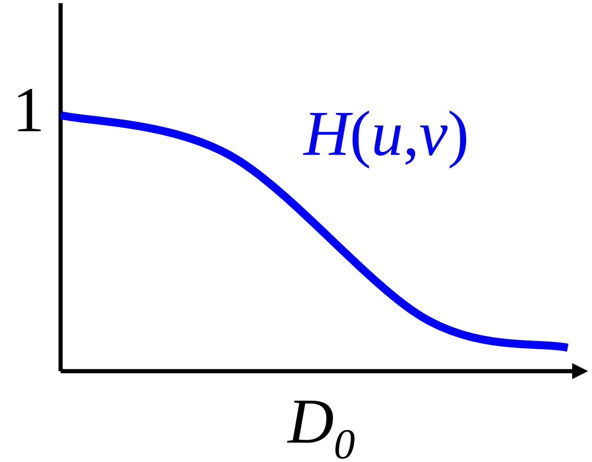
- Large mask – Frequency-domain filtering is usually faster due to the use of FFT
- Small mask – Spatial-domain filtering is usually faster
- Frequency-domain filtering is usually more intuitive. One can do experiments in the frequency domain to obtain the desired result, and then find an approximate corresponding spatial-domain filter for faster execution.

Low-Pass Filters (LPF)

Idea: To reduce high-frequency components
Corresponding to: Smoothing filters

Types of LPF discussed here:

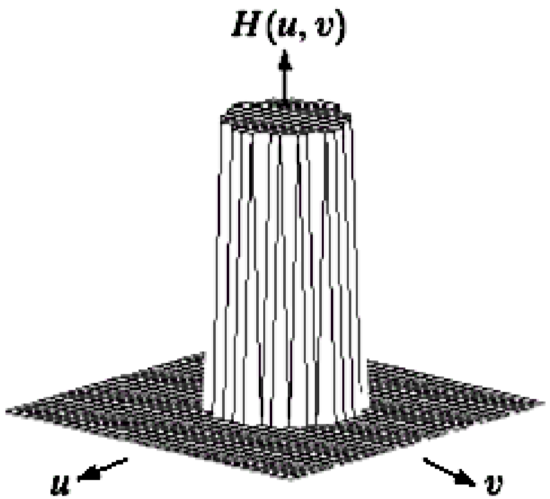
- Ideal LPF
- Butterworth LPF
- Gaussian LPF

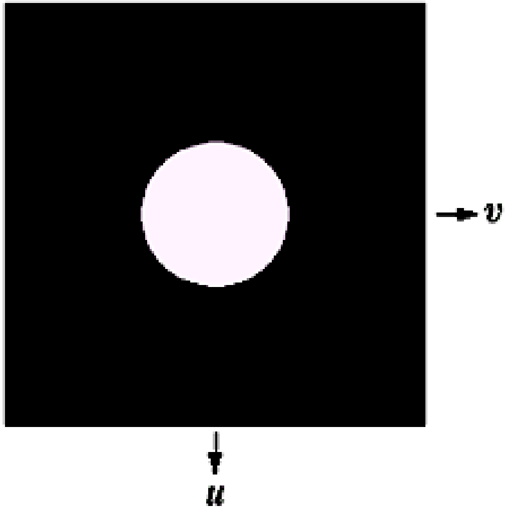


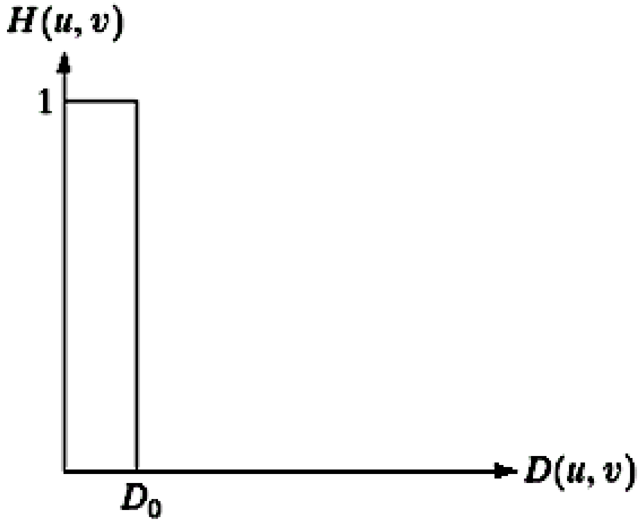
Important properties:

- Cuttoff frequency D_0
- Abruptness/smoothness of transition

Ideal LPF

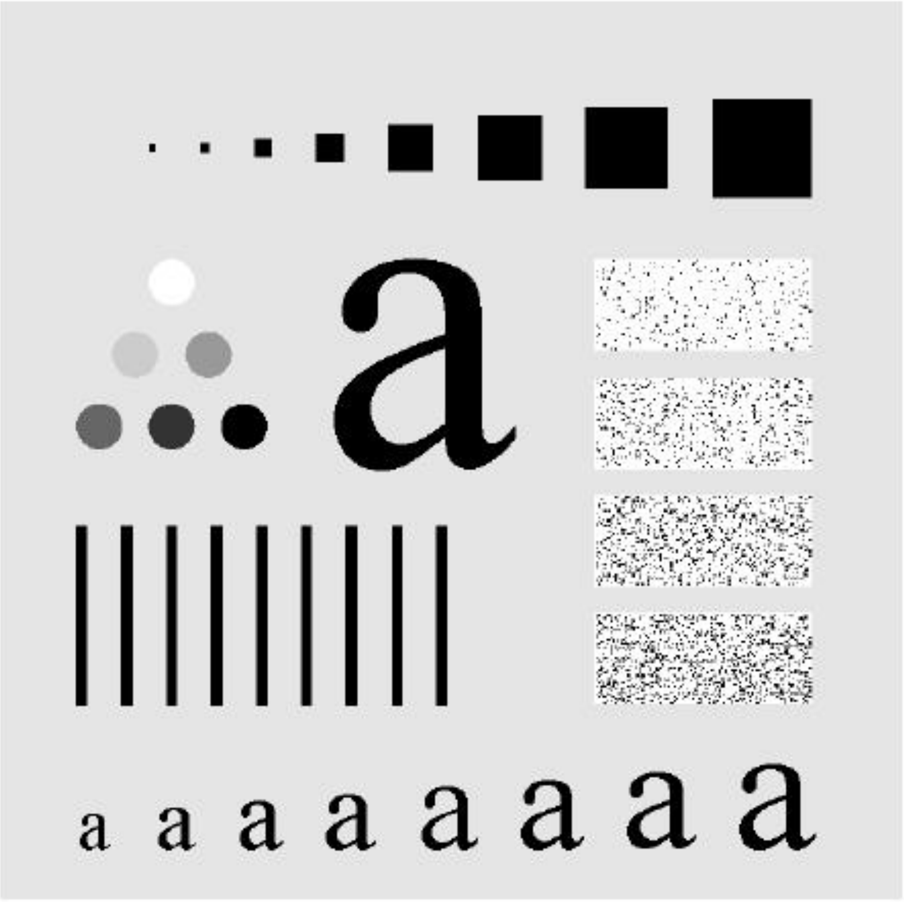


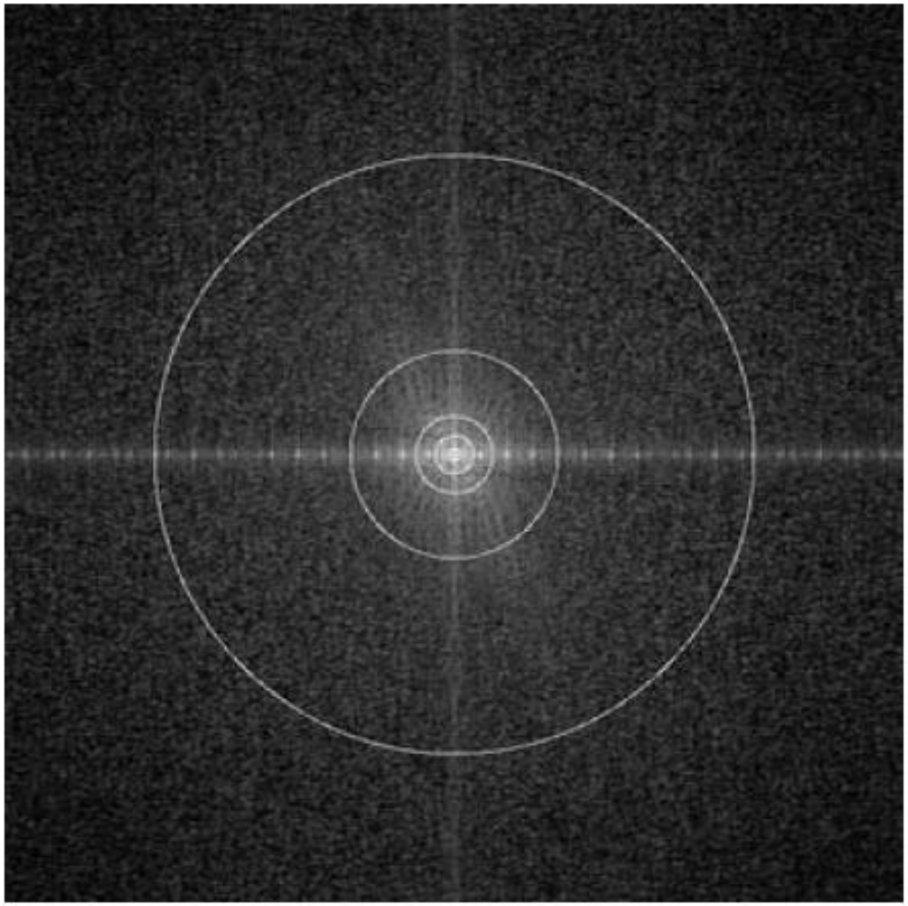




$$H(u,v)=\begin{cases}0, & \text{if } D(u,v)\ge D_0\\ 1, & \text{otherwise}\end{cases}$$

Ideal LPF



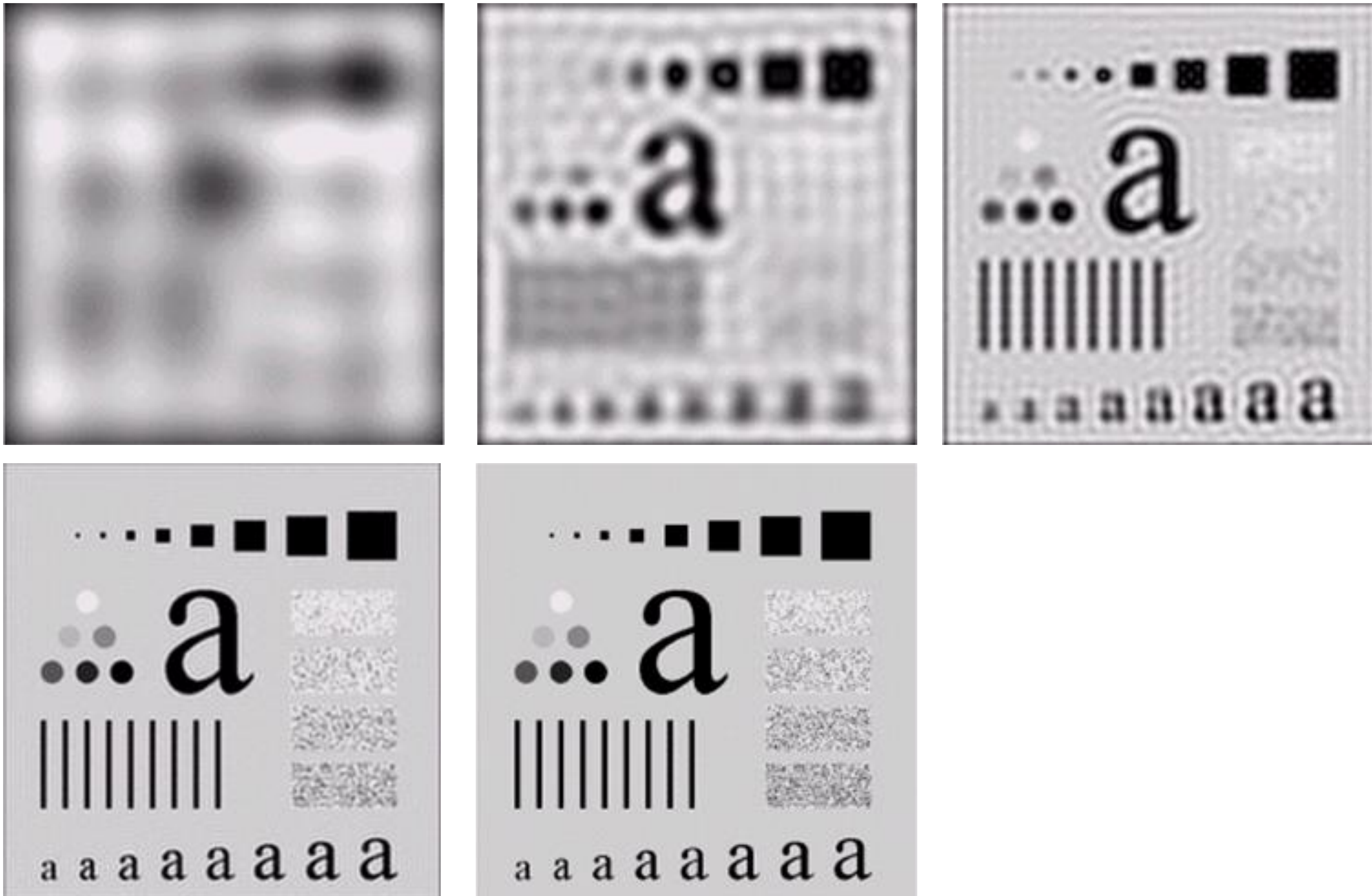


Ratio of image power contained in circles (from inside out):

87%, 93.1%, 95.7%, 97.8%, 99.2%

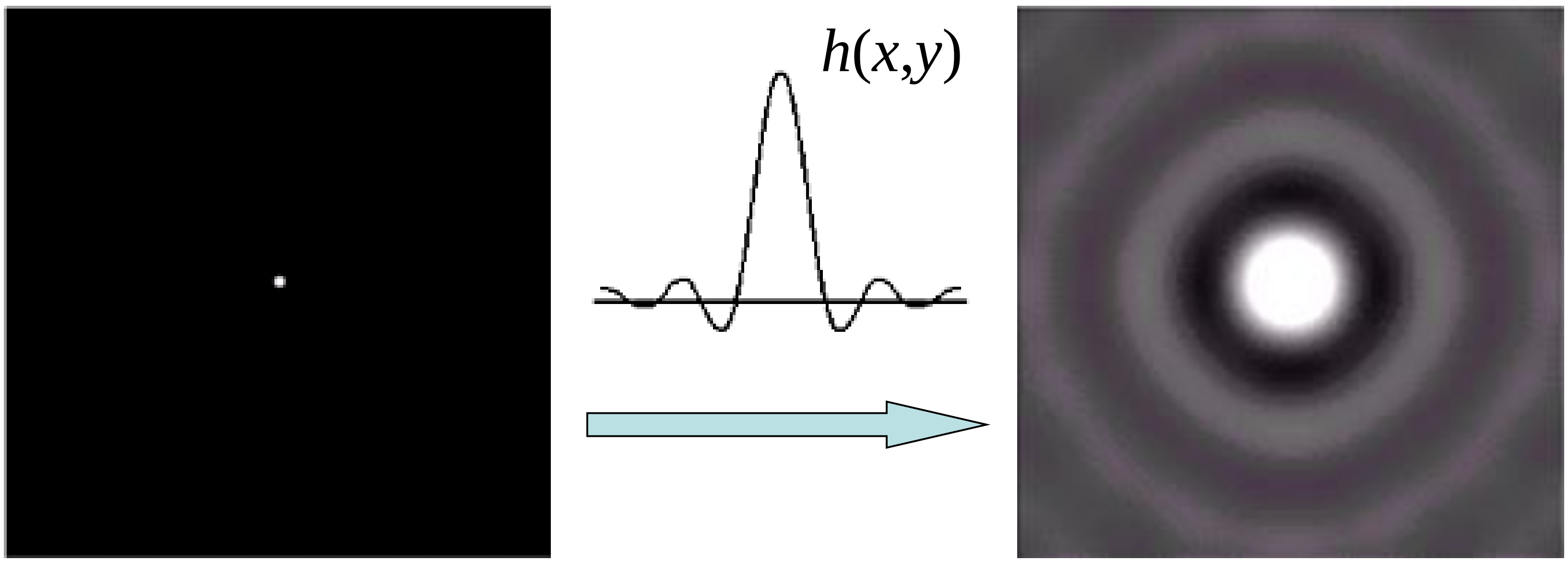
Corresponding D_0: 10, 30, 60, 160, 460

Ideal LPF



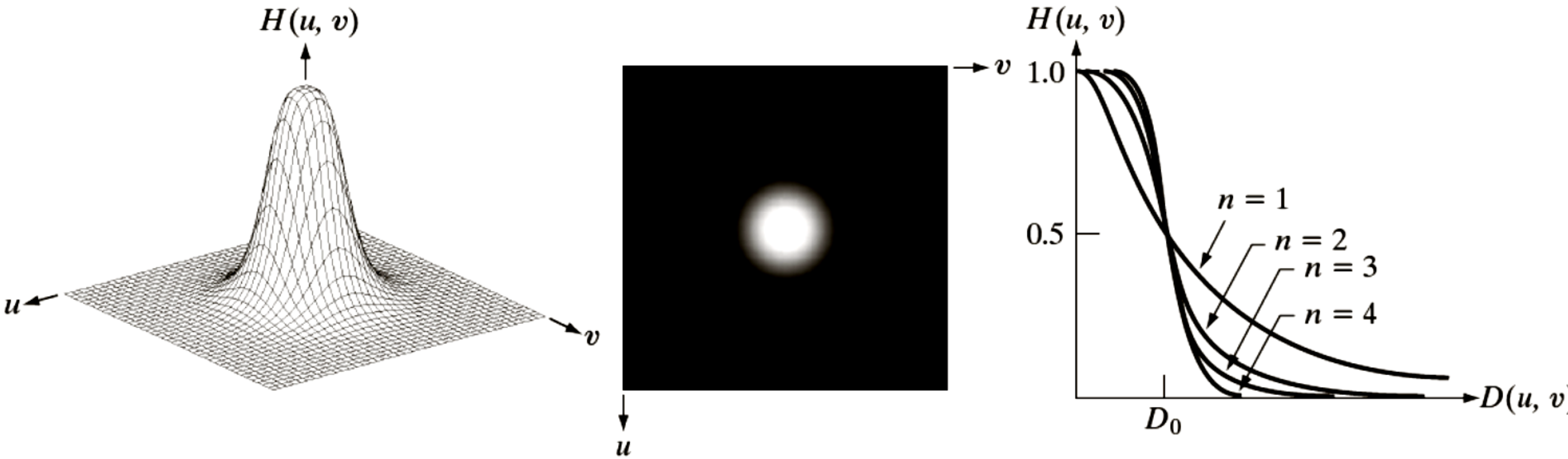
Ideal LPF

The cause of the ringing can be understood with the spatial-domain representation of the filter (the **impulse response**).



Space-frequency reciprocity: The stronger the LPF is (smaller D_0), the wider the peak in $h(x,y)$ is, and hence the more visible ringing after filtering.

Butterworth LPF

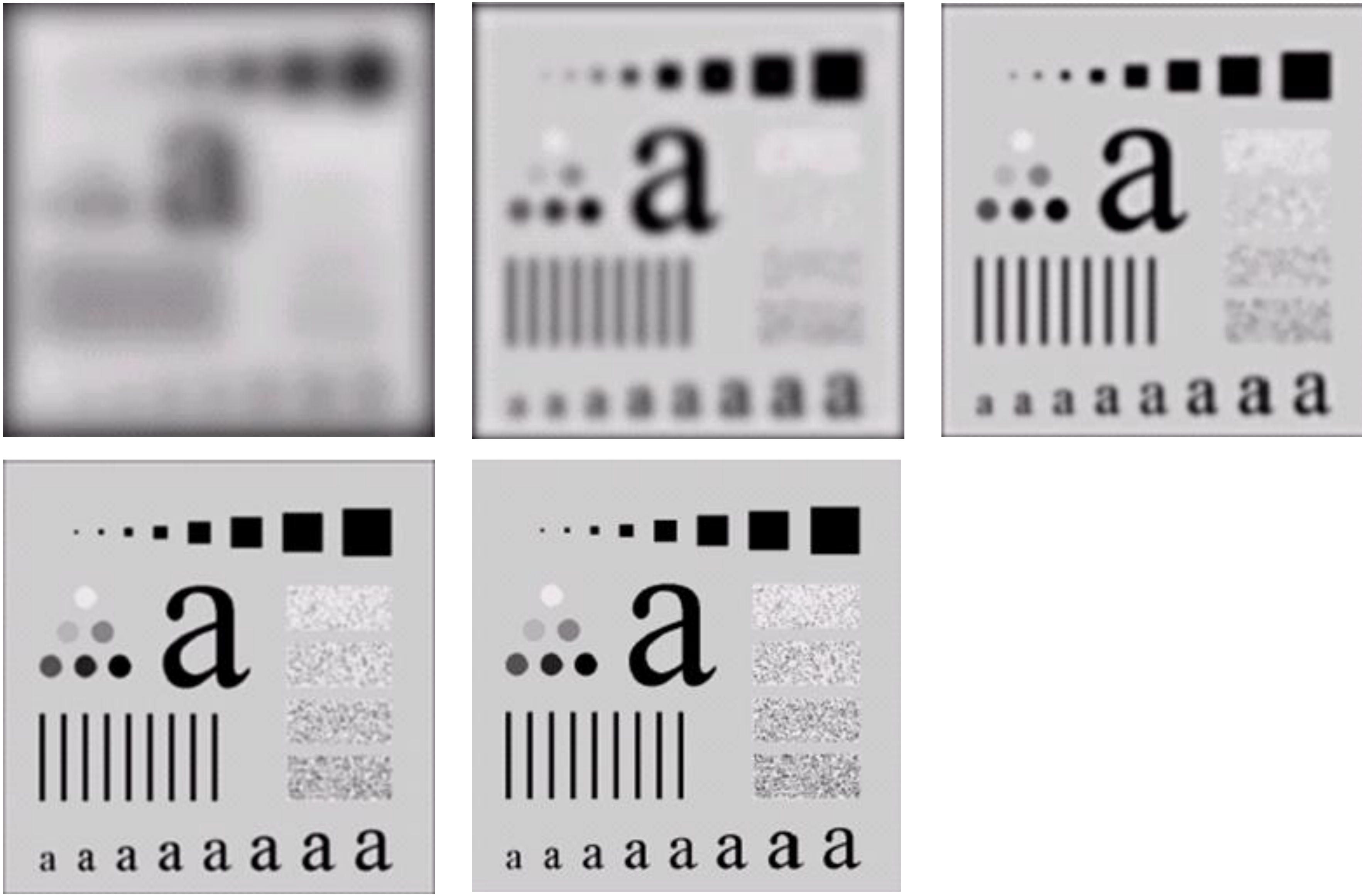


$$H(u,v)=\frac{1}{1+\left[D(u,v)/D_0\right]^{2n}}$$

The order n determines the smoothness of the transition.

Butterworth LPF

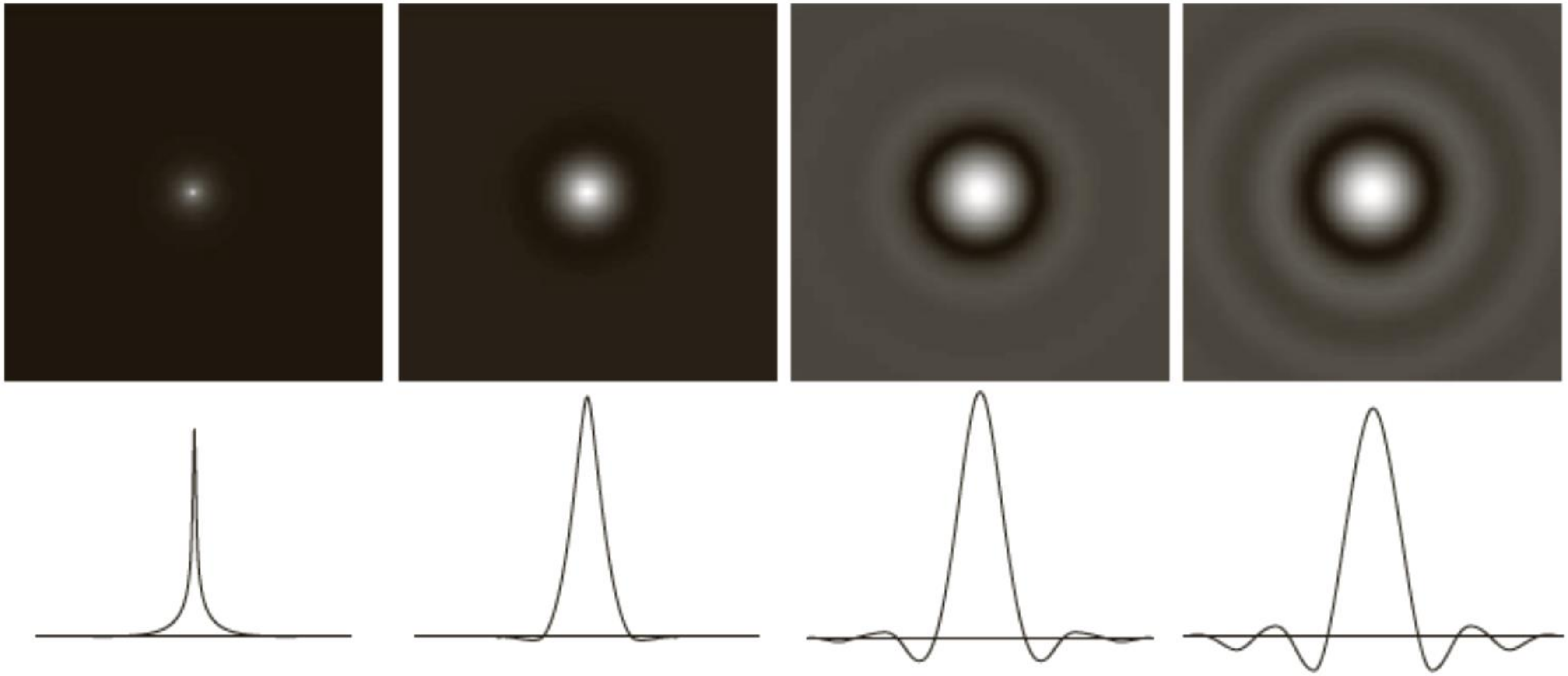
n=2, D_0 = 10, 30, 60, 160, 460, respectively



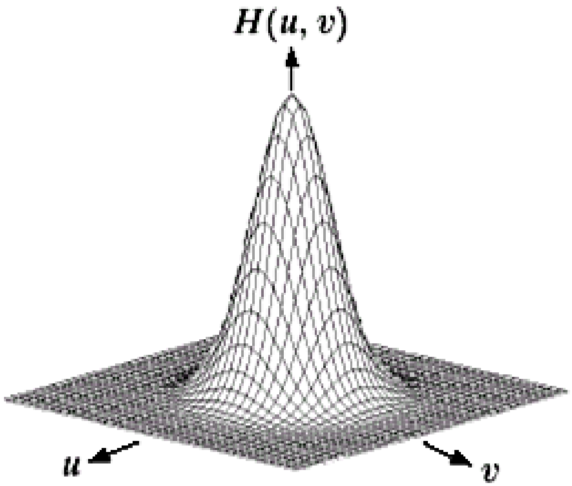
Butterworth LPF

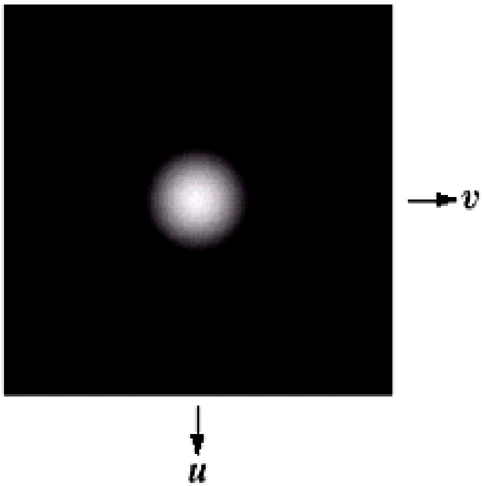
The impulse response vs. n:

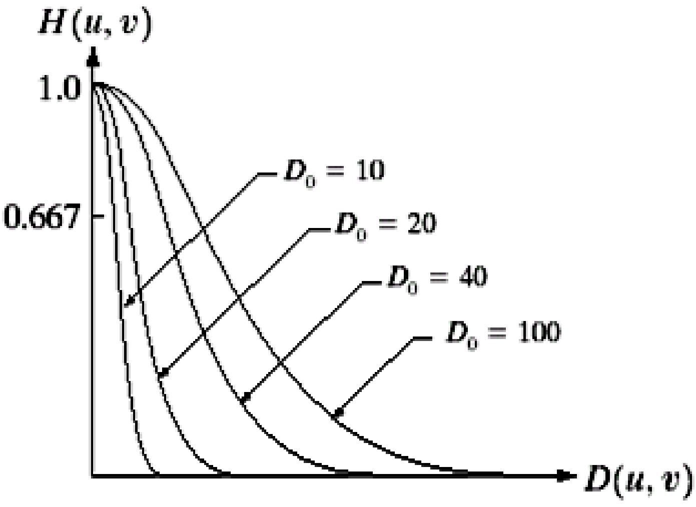
$n = 1$ $n = 2$ $n = 5$ $n = 20$



Gaussian LPF





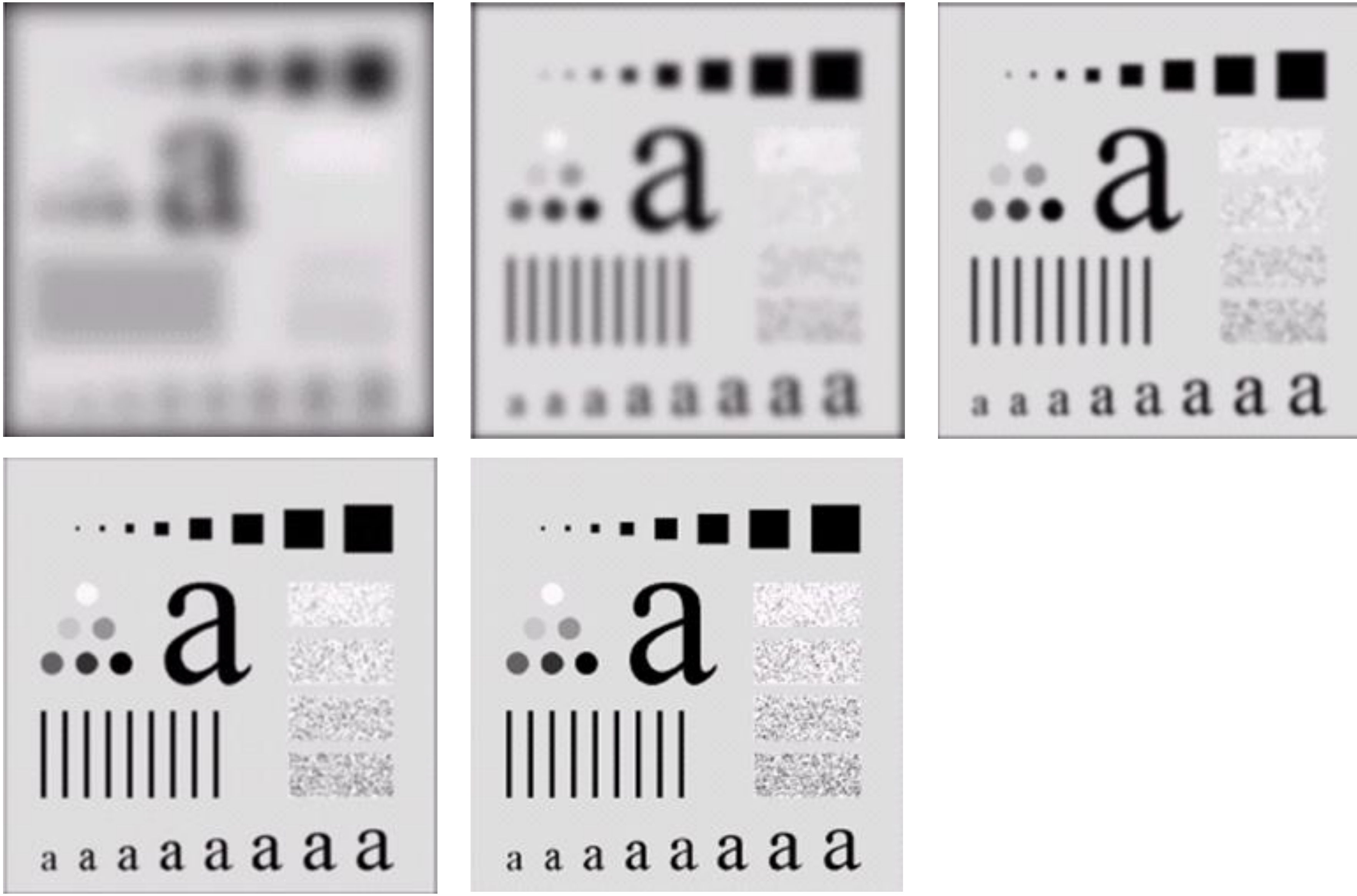


$$H(u,v) = \exp(-D^2(u,v)/2D_0{}^2)$$

The impulse response is also Gaussian (no ringing!) with standard deviation of $(2\pi D_0)^{-1}$.

Gaussian LPF

D_0 = 10, 30, 60, 160, 460, respectively

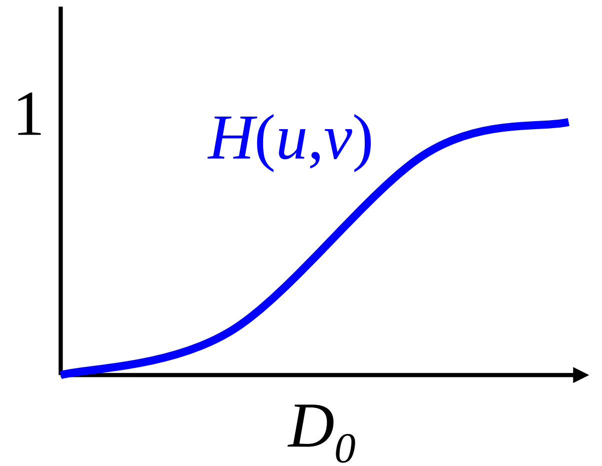


High-Pass Filters (HPF)

Idea: To reduce low-frequency components
Corresponding to: Sharpening filters

Types of HPF discussed here:

- Ideal HPF
- Butterworth HPF
- Gaussian HPF



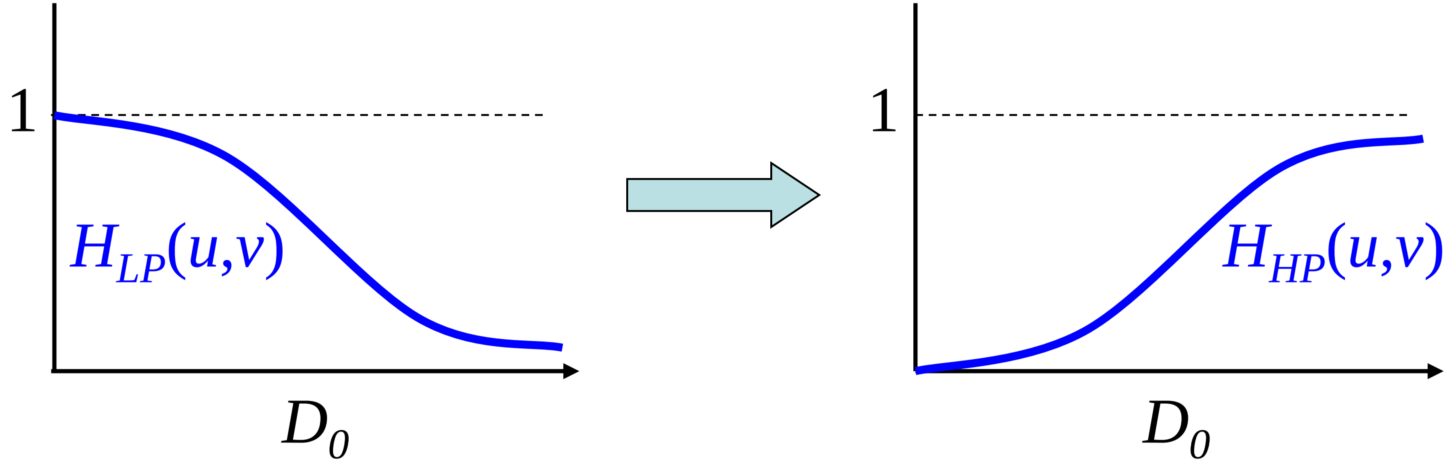
Important properties:

- Cuttoff frequency D_0
- Abruptness/smoothness of transition

Typical HPF

An simple (and useful) approach to get a HPF from a LPF:

$$H_{HP}(u,v) = 1 - H_{LP}(u,v)$$



Typical HPF

$$\boxed{H_{LP}(u,v)} \qquad \boxed{H_{HP}(u,v)}$$

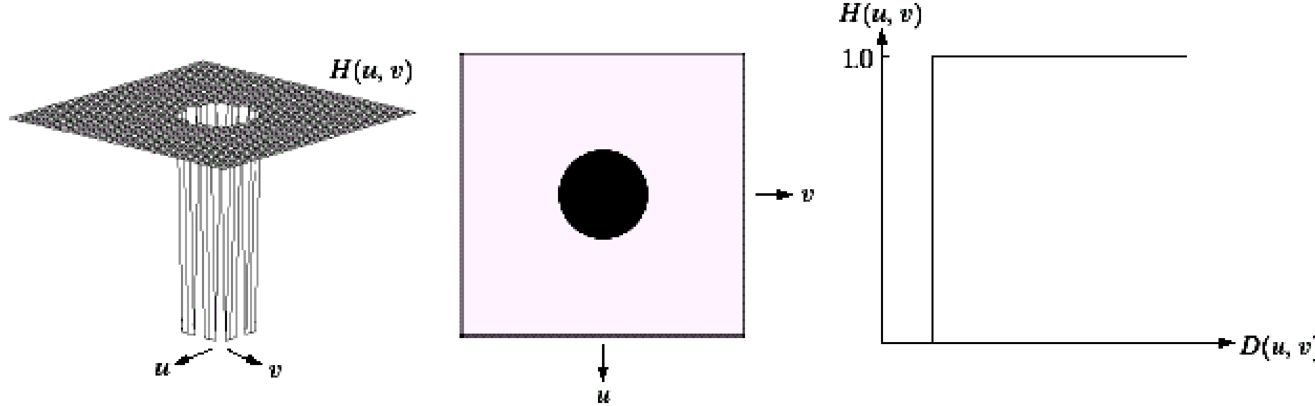
Ideal: $\begin{cases} 0, & \text{if } D(u,v) \geq D_0 \\ 1, & \text{otherwise} \end{cases} \Rightarrow \begin{cases} 1, & \text{if } D(u,v) \geq D_0 \\ 0, & \text{otherwise} \end{cases}$

Butterworth: $\dfrac{1}{1+\left[D(u,v)/D_0\right]^{2n}} \Rightarrow \dfrac{1}{1+\left[D_0/D(u,v)\right]^{2n}}$

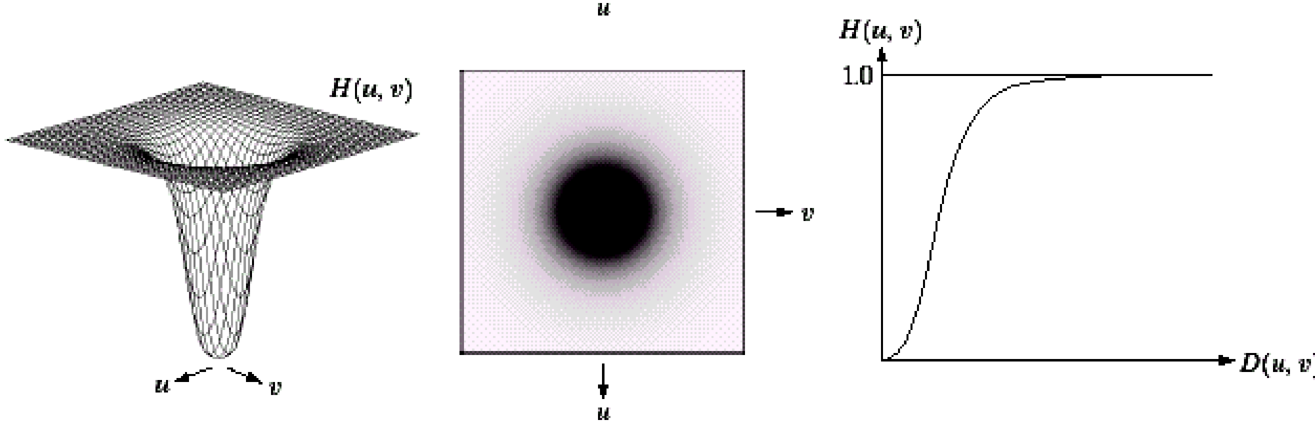
Gaussian: $\exp(-D^2(u,v)/2D_0{}^2) \Rightarrow 1-\exp(-D^2(u,v)/2D_0{}^2)$

Typical HPF

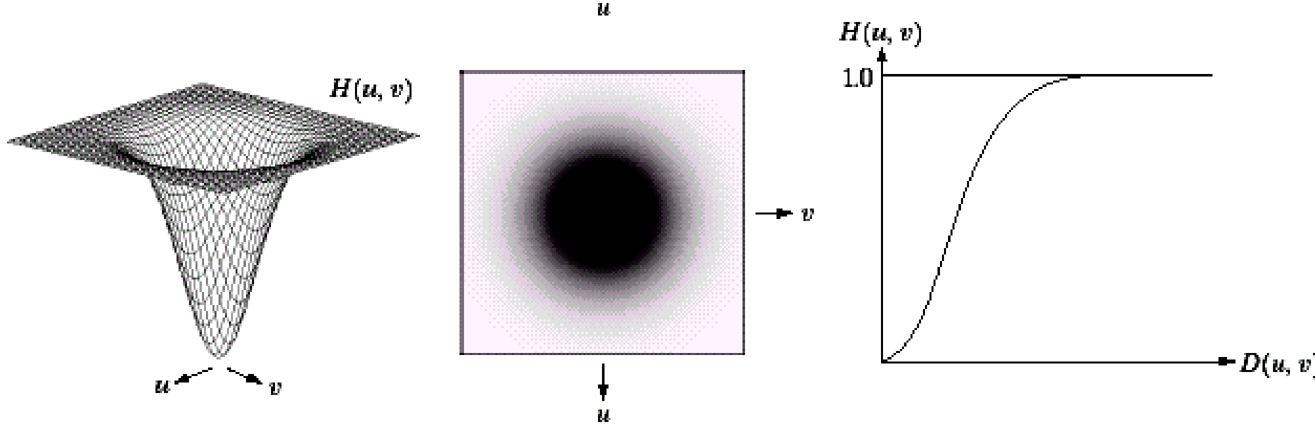
Ideal:



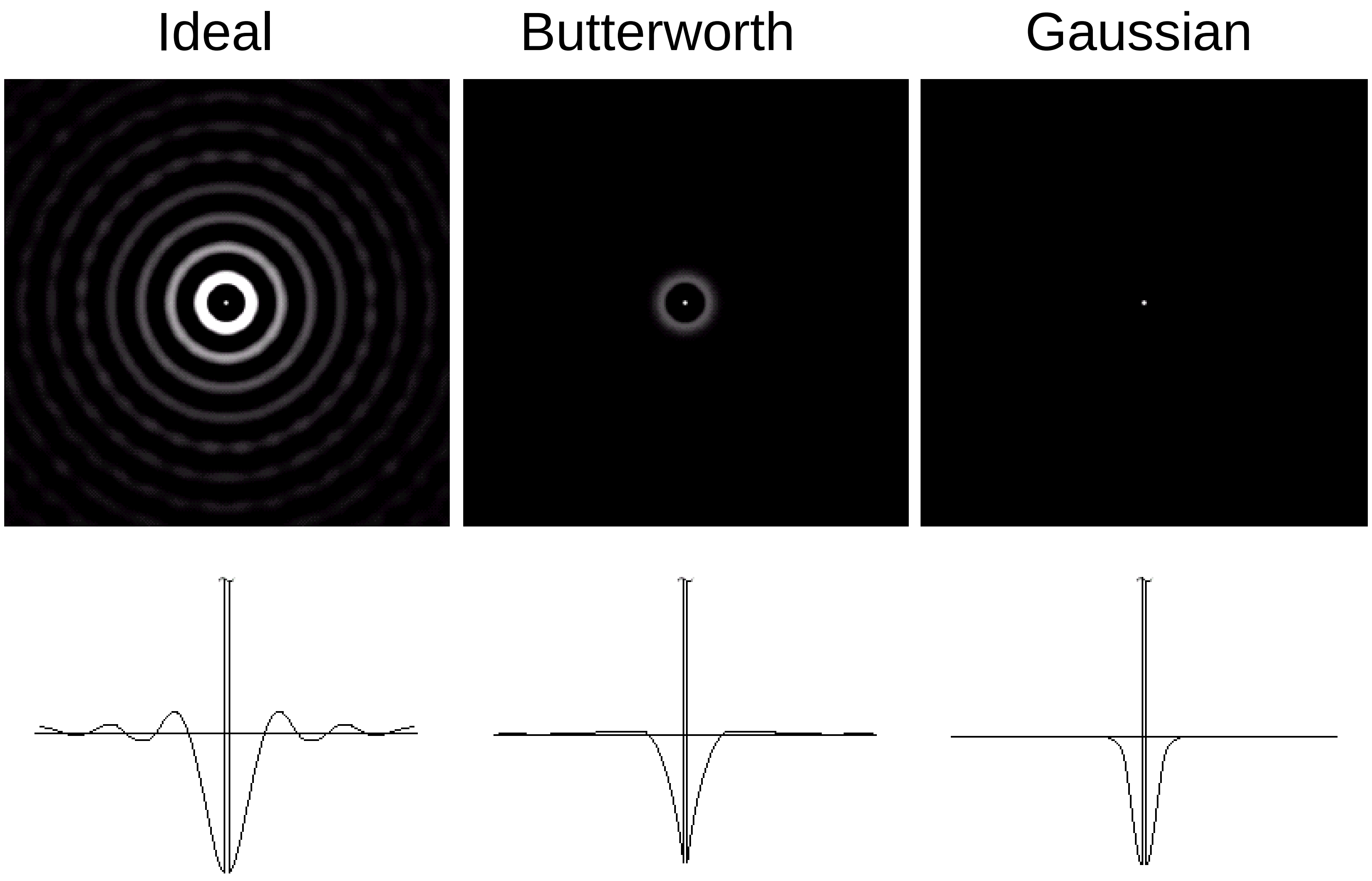
Butterworth:



Gaussian:



Ringing in Ideal HPF



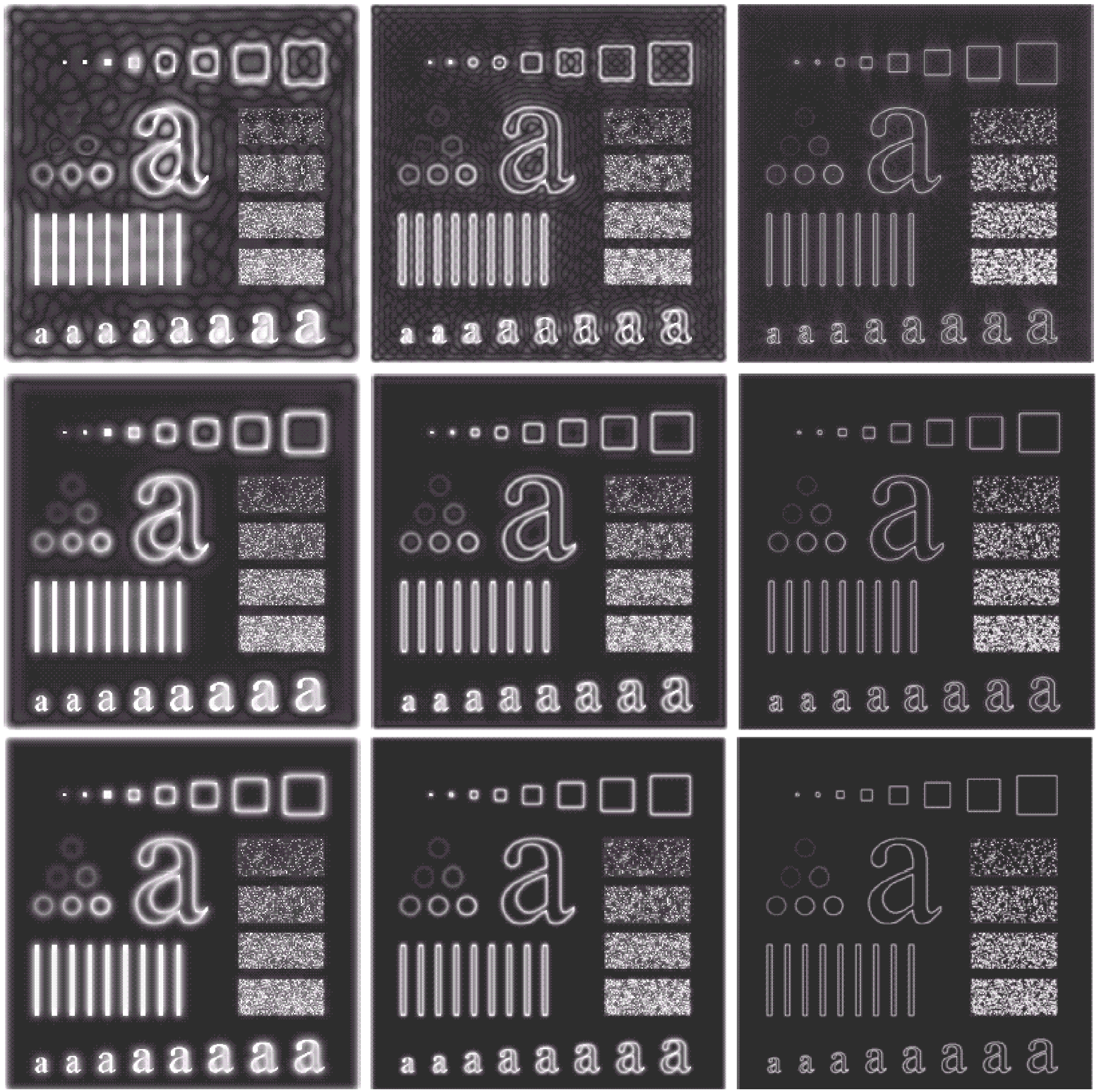
Effect of HPF

$D_0 = 30$ $D_0 = 60$ $D_0 = 160$

Ideal:

Butterworth:

Gaussian:



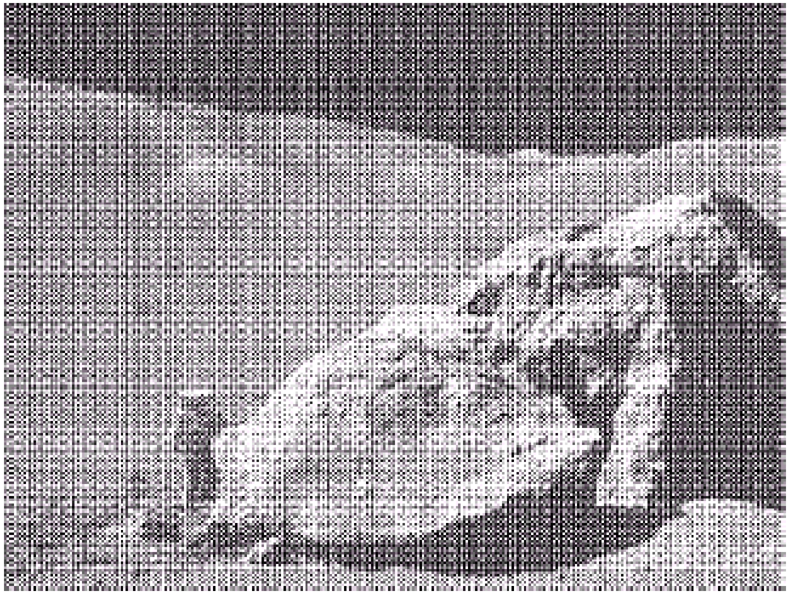
Selective Filtering

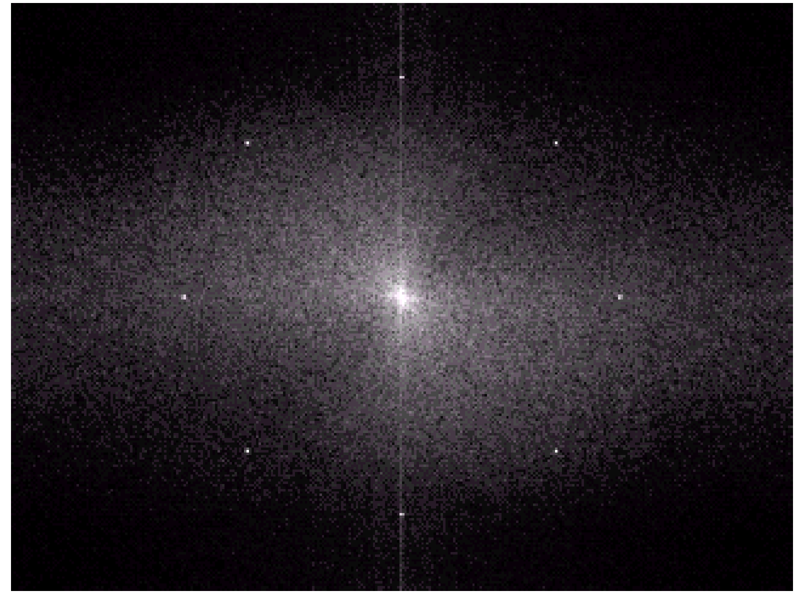
Other than LPF and HPF, frequency domain processing can use frequency domain filters specifically designed to remove noise or other unwanted signal that occur at some particular spatial frequencies.

Such noise can come from the imaging device or interferences.

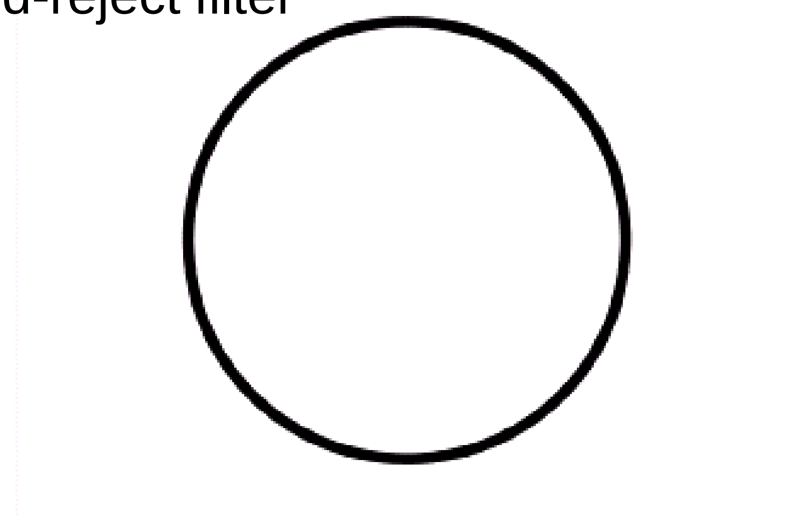
Such filtering is usually done in the frequency domain if possible. It is more difficult (although possible) to find a spatial filter to achieve similar effects.

Selective Filtering: Example

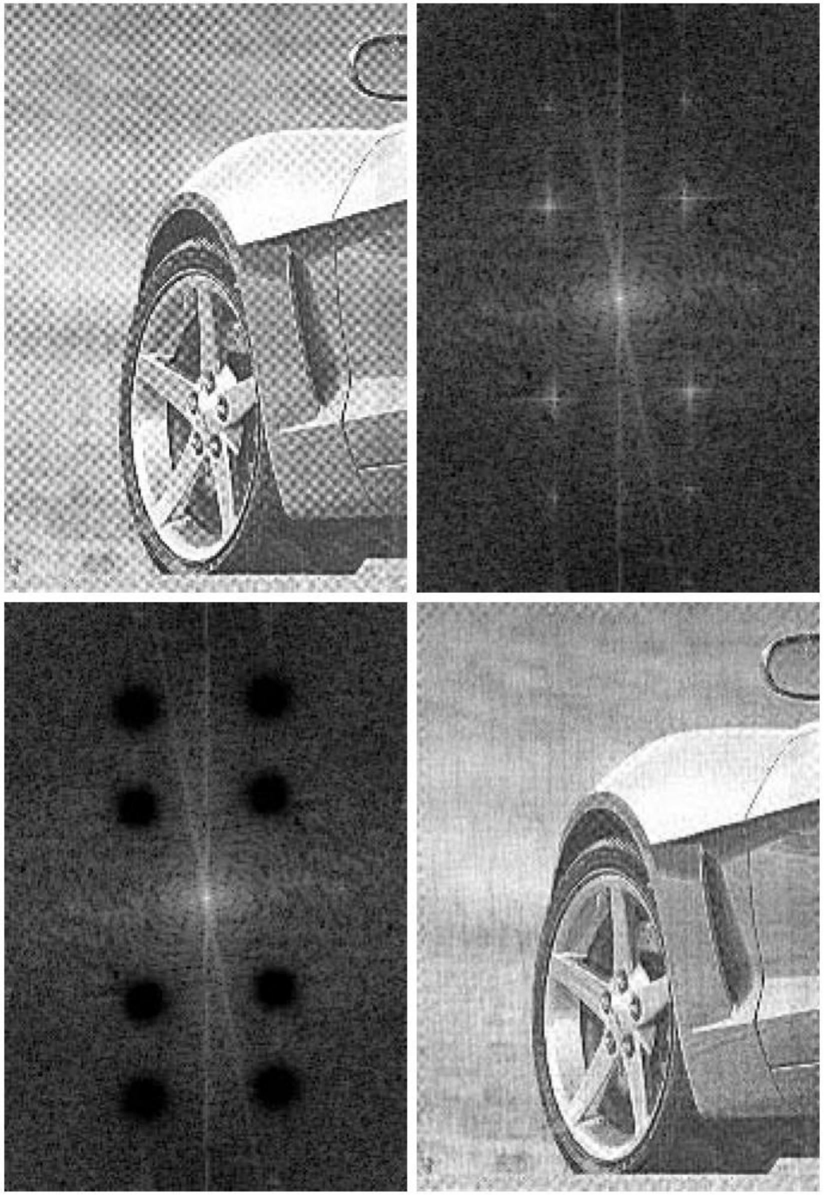




A band-reject filter



Selective Filter: Examples



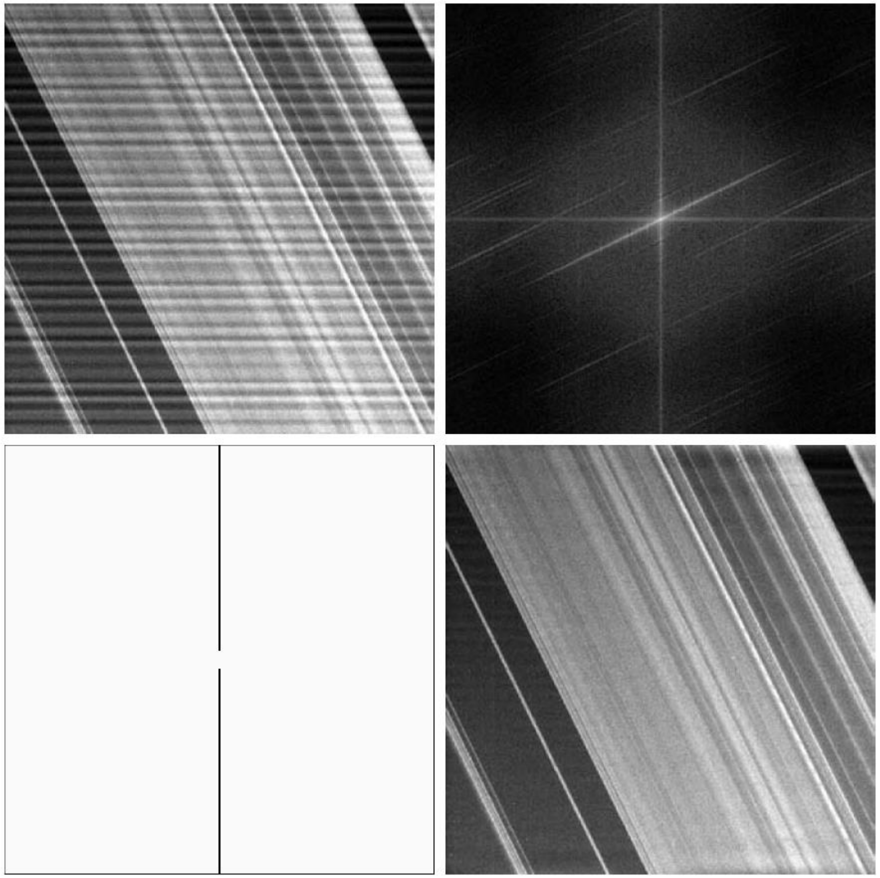
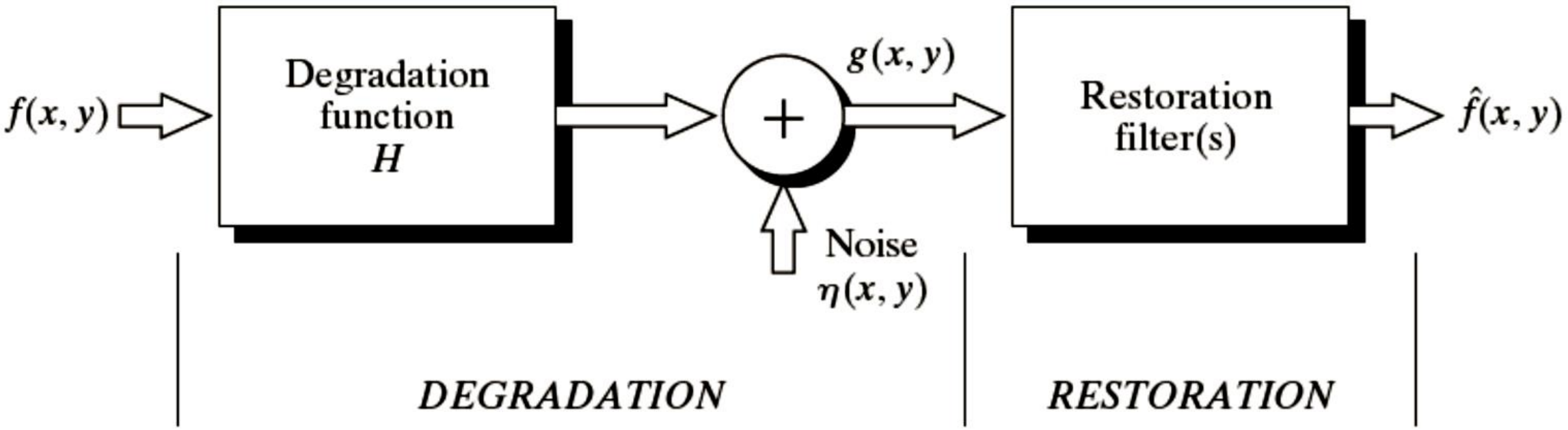


Image Degradation and Restoration

Model of degradation:

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$$

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$



Estimating Degradation Function

- Degradation function is usually unknown.
- Estimation by observation: Trial-and-error using a fairly noise-free patch of the image.
- Estimation by experimentation: If the original image acquisition setting is reproducible, we can obtain the impulse response by imaging a small dot of light.
- Estimation by modeling: A functional form of degradation may be available by assuming a particular source of degradation (such as turbulence for aerial images). Parameters may still require interactive experiments.

Estimating Degradation Function

Motion blur (unidirectional) is a type of degradation with a simple model.

Inverse Filtering

Goal: To undo the effect of degradation function.

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

$$\widehat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

$$\widehat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

If we assume no noise, we can use estimated H to recover the original image.

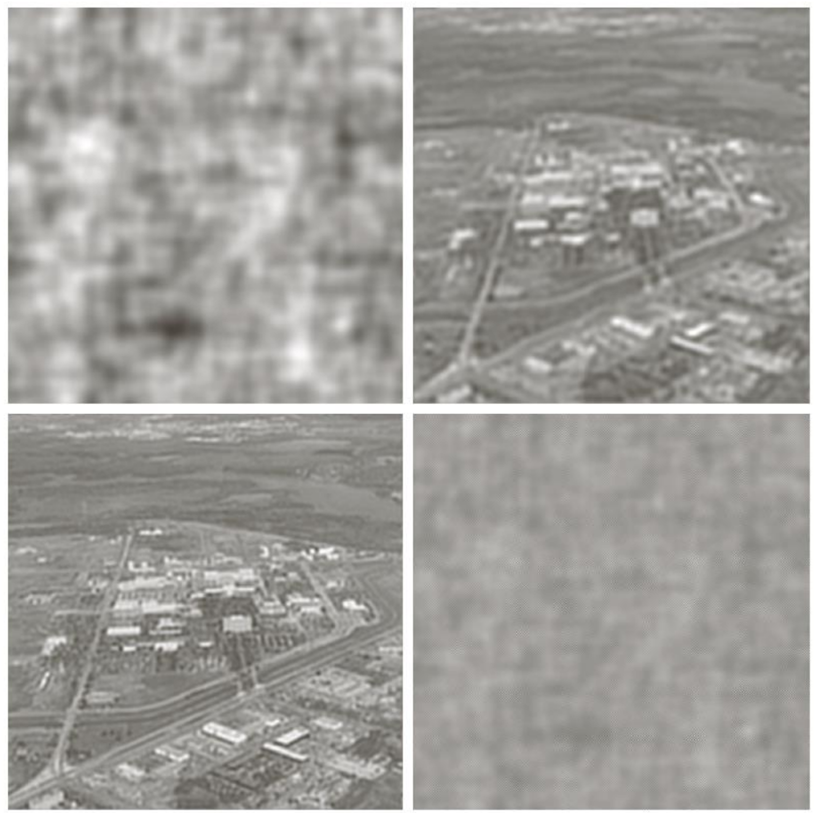
Inverse Filtering

Problem: When H is small (usually at higher frequencies), the actual noise can dominate the result of inverse filtering.

Solution: Band-limiting (ignoring higher frequencies).

Example:

no cutoff



Wiener Filtering

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v)$$

Signal power spectrum

Noise power spectrum

Simplified form:

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} \right] G(u,v)$$

Wiener Filtering Example

Radially-limited inverse filtering

Wiener filtering

Wiener Filtering Example

Original Inverse Wiener

Lots of noise

Some noise

Little noise