Digital Image Processing

Linear image processing: frequency domain filtering

Erchan Aptoula

Spring 2016-2017

Contents

- Principles of frequency domain filtering
- Low-pass filters
- High-pass filters
- Image noise and and how to deal with it
- Subsampling and (anti-)aliasing

Main **concept**: frequency - the number of times a periodic function repeats the same sequence of values during a unit variation of an independent variable.

Main **tool**: Fourier Transform

Main **idea**: convert images into a different representation, filter them, and return to the initial representation

Fourier Series

Any periodic function can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by different coefficients:

$$f(x) = \tfrac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \sin(nx),$$

$$\begin{aligned} a_0 &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\,dx \\ a_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(nx)\,dx \\ b_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(nx)\,dx \end{aligned}$$

Taylor Series: $f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \ldots.$

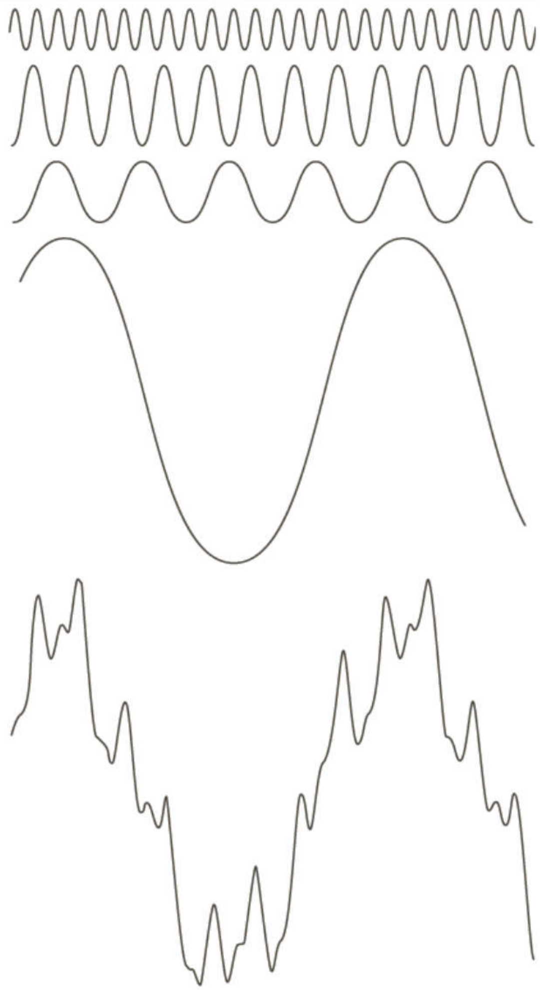
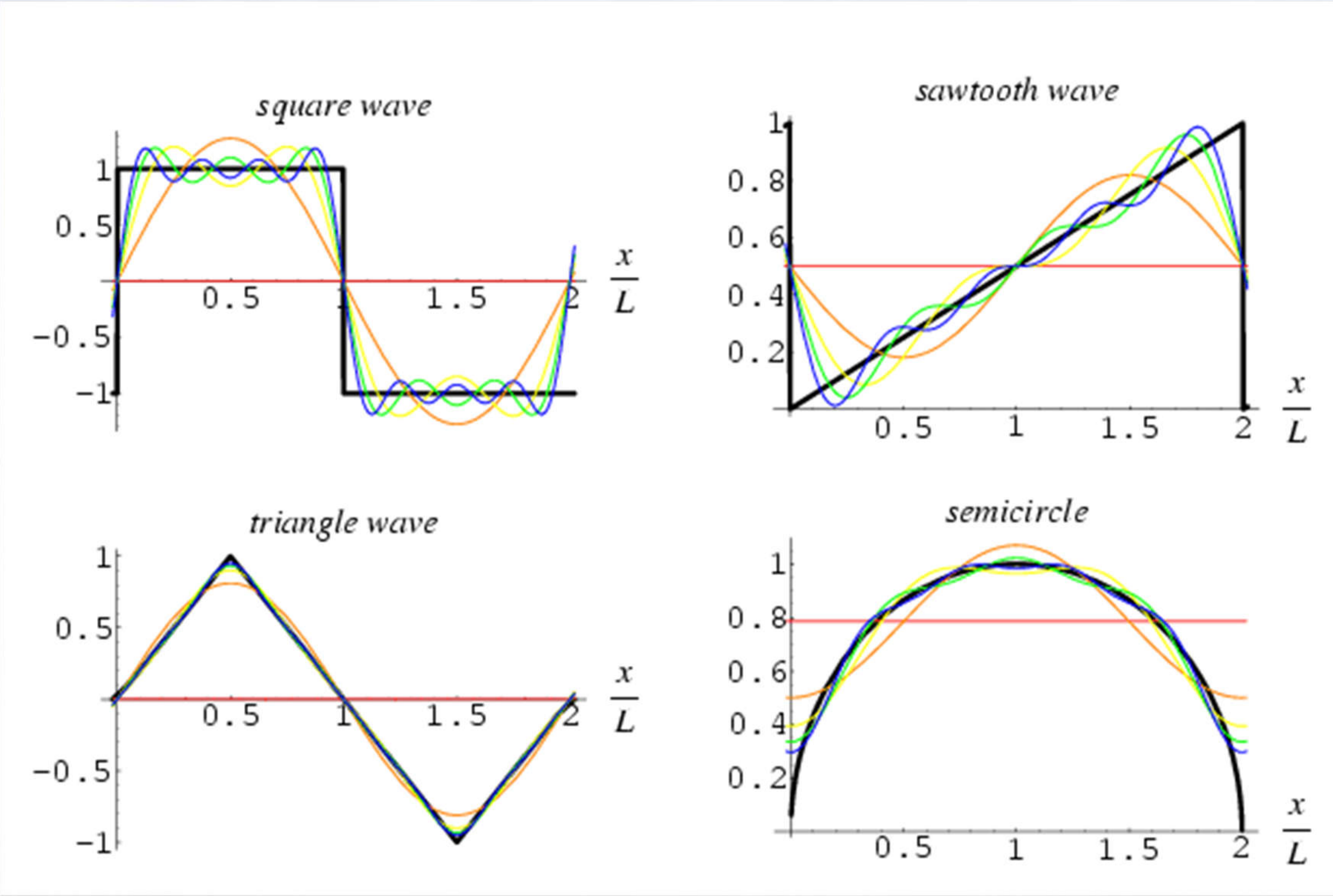


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Linear image processing



Fourier Transform (FT) (= generalization of the complex Fourier series)

Any function that **is not periodic** can be expressed as the integral of sines and/or cosines multiplied by a weighing function

The 1D FT $F(u)$ of a continuous single variable function $f(x)$:

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

Given $F(u)$, $f(x)$ can be obtained through the inverse FT:

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

In the 2D continuous case:

$$F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}\,dx\,dy$$

$$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{j2\pi(ux+vy)}\,du\,dv$$

And now the discrete case

The 1D Discrete FT (**DFT**) $F(u)$ of a discrete single variable function $f(x)$, $x = 0, 1, 2, \ldots M - 1$:

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

Given $F(u)$, $f(x)$ can be obtained through the inverse DFT (**IDFT**):

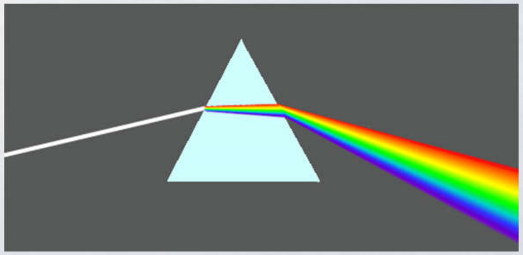
$$f(x) = \sum_{u=0}^{M-1} F(u)e^{j2\pi ux/M} \text{ for } x = 0,1,2,\ldots,M-1$$

And the 2D DFT and IDFT for a discrete function $f(x,y), x = 0,1,2,\dots,M-1;$
$y = 0,1,2,\dots,N-1$

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

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

- Every term of $F(u,v)$ is composed of ALL the terms of $f(x,y)$
- The domain over which $F(u,v)$ ranges is called the frequency domain



The Fourier Transform can be expressed in polar coordinates as well:

$$F(u) = |F(u)| e^{-j\phi(u)}$$

The magnitude or spectrum: $|F(u)| = \left[R^2(u) + I^2(u)\right]^{1/2}$

The phase spectrum: $\phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right]$

Where $R(u)$ and $I(u)$ represent respectively the real and imaginary parts of $F(u)$

Power spectrum is the square of the spectrum.

Applications of the FT and Fourier series:

- Signal processing: JPEG,MP3,MPEG encoding, filtering, …
- Telecommunications: antenna design,…
- Solving differential equations
- Interferometers, GPS,…
- Synthetic aperture radar,…
- Side scan sonar technology
- Geology: seismic research,…
- Astronomy: radar imaging
- Optics…

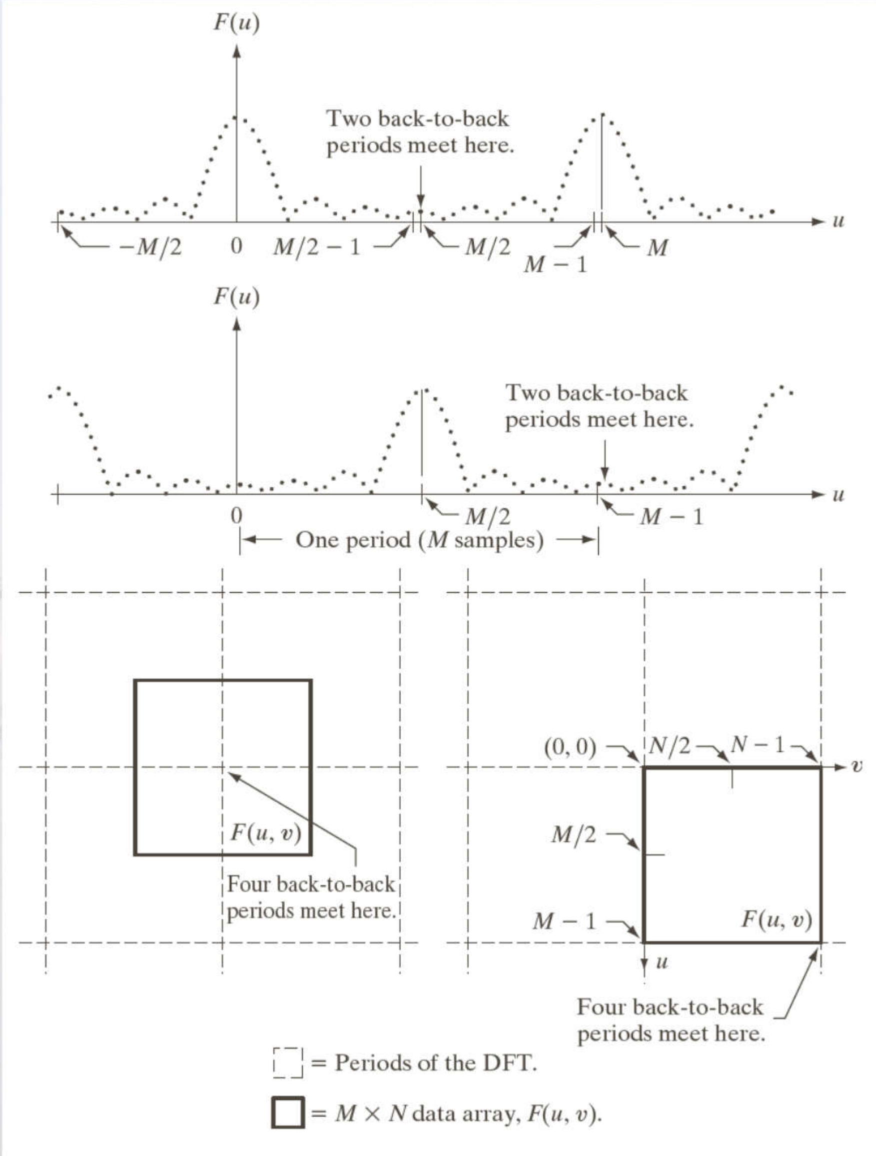
Notes

- The value at $F(0,0)$ is the mean of $f(x, y)$
- The spectrum is not an image, but a display of power in the original image vs frequency components
- Rotating in the spatial domain, leads to a rotation in the frequency domain
- Translation invariant
- FT has a complexity of $O((MN)^2)$; Fast Fourier Transform: $O(MNlog(MN))$,

works only with images that have dimensions equal to a power of 2.

- FFT is considered among the greatest algorithms of the 20th century (1965).
- FT is conjugate symmetric $\left(F(u, v) = F^*(-u, -v)\right)$ and its spectrum is symmetric around the origin ($|F(u, v)| = |F(-u, -v)|$) (when $f(x, y)$ is real).
- The input image is multiplied with $(-1)^{x+y}$ prior to computing the FT, so that the origin of the FT is located at $(u = \frac{M}{2}, v = \frac{N}{2})$. BUT WHY?

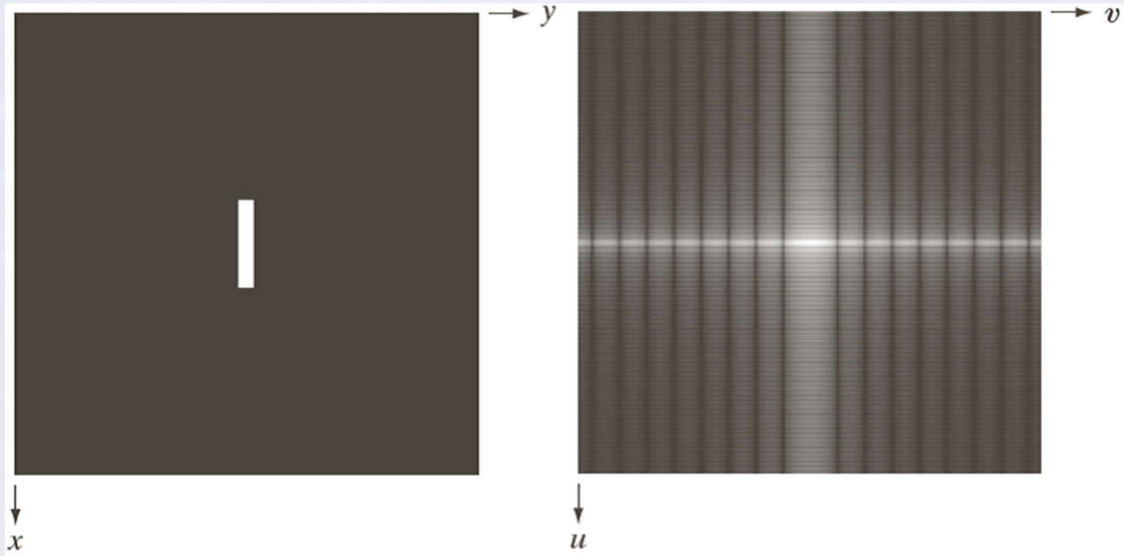
Because:

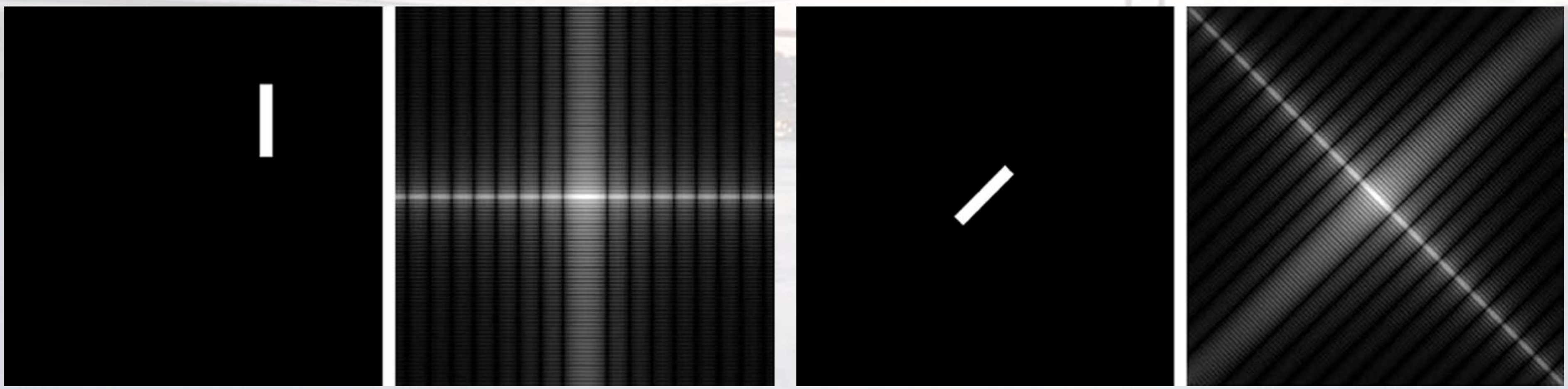


a
b
c d

FIGURE 4.23 Centering the Fourier transform. (a) A 1-D DFT showing an infinite number of periods. (b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^x$ before computing $F(u)$. (c) A 2-D DFT showing an infinite number of periods. The solid area is the $M \times N$ data array, $F(u, v)$, obtained with Eq. (4.5-15). This array consists of four quarter periods. (d) A Shifted DFT obtained by multiplying $f(x, y)$ by $(-1)^{x+y}$ before computing $F(u, v)$. The data now contains one complete, centered period, as in (b).

Linear image processing



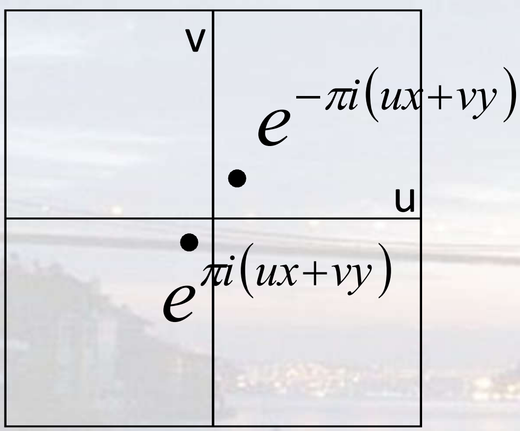


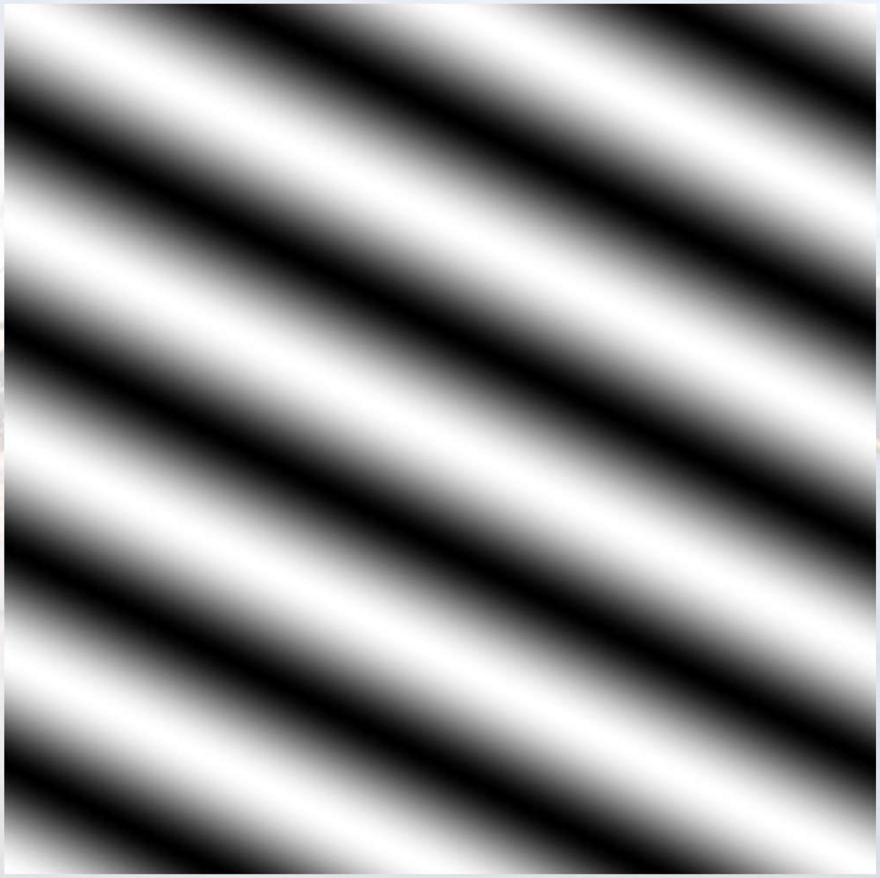
Linear image processing

Here we have an image consisting of a single frequency.

The magnitude of vector (u, v): frequency

The direction of vector (u, v): orientation of the sinusoid and it is constant in the direction perpendicular to it.

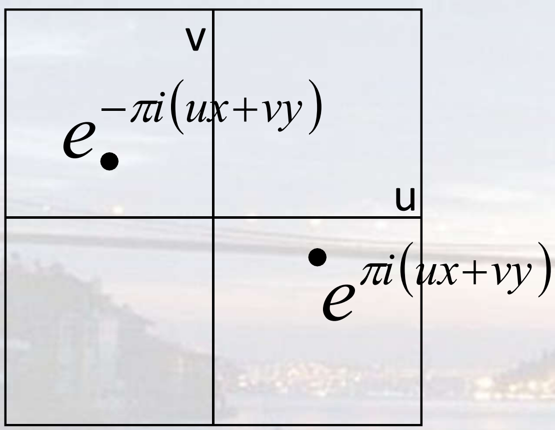




The DC is missing..don't mind it.

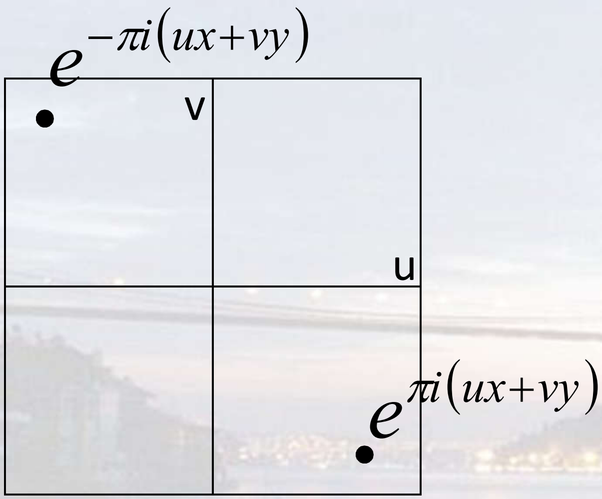
Adapted from Selim Aksoy, Bilkent U.

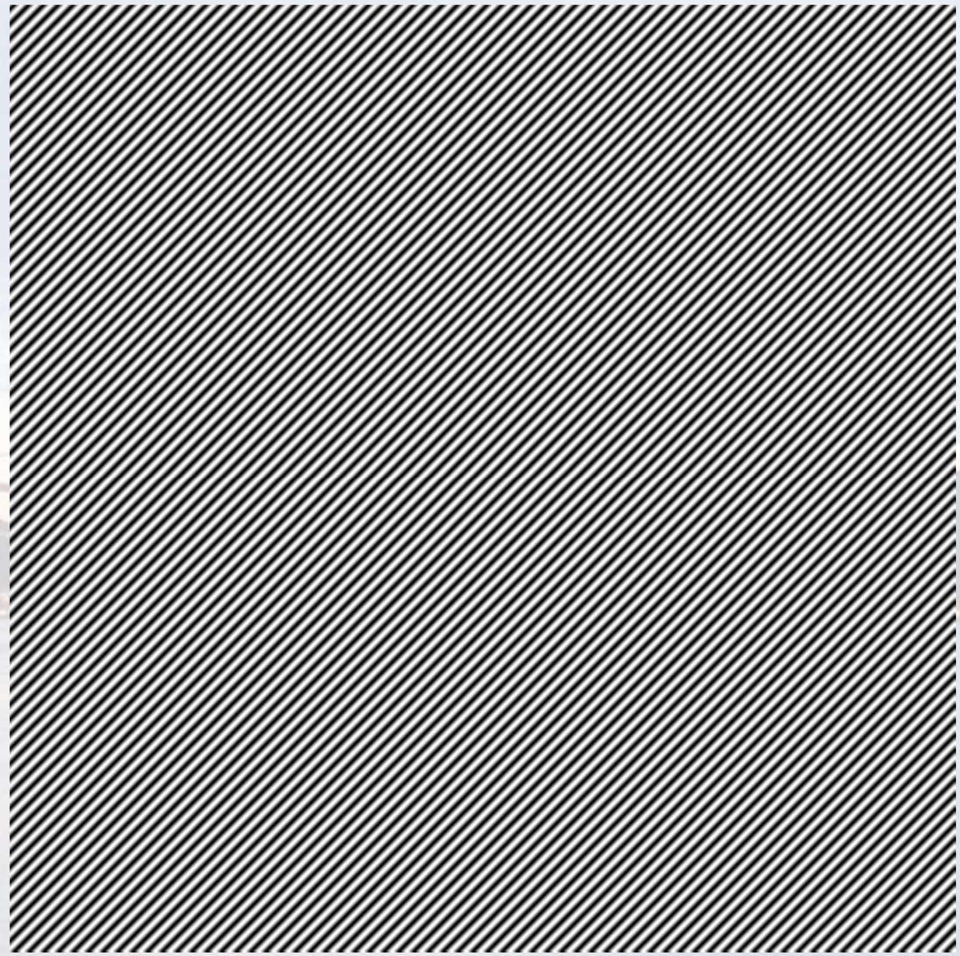
Another example with a single sinusoid of higher frequency and different direction



Linear image processing

Another example with a single sinusoid of even higher frequency and different direction

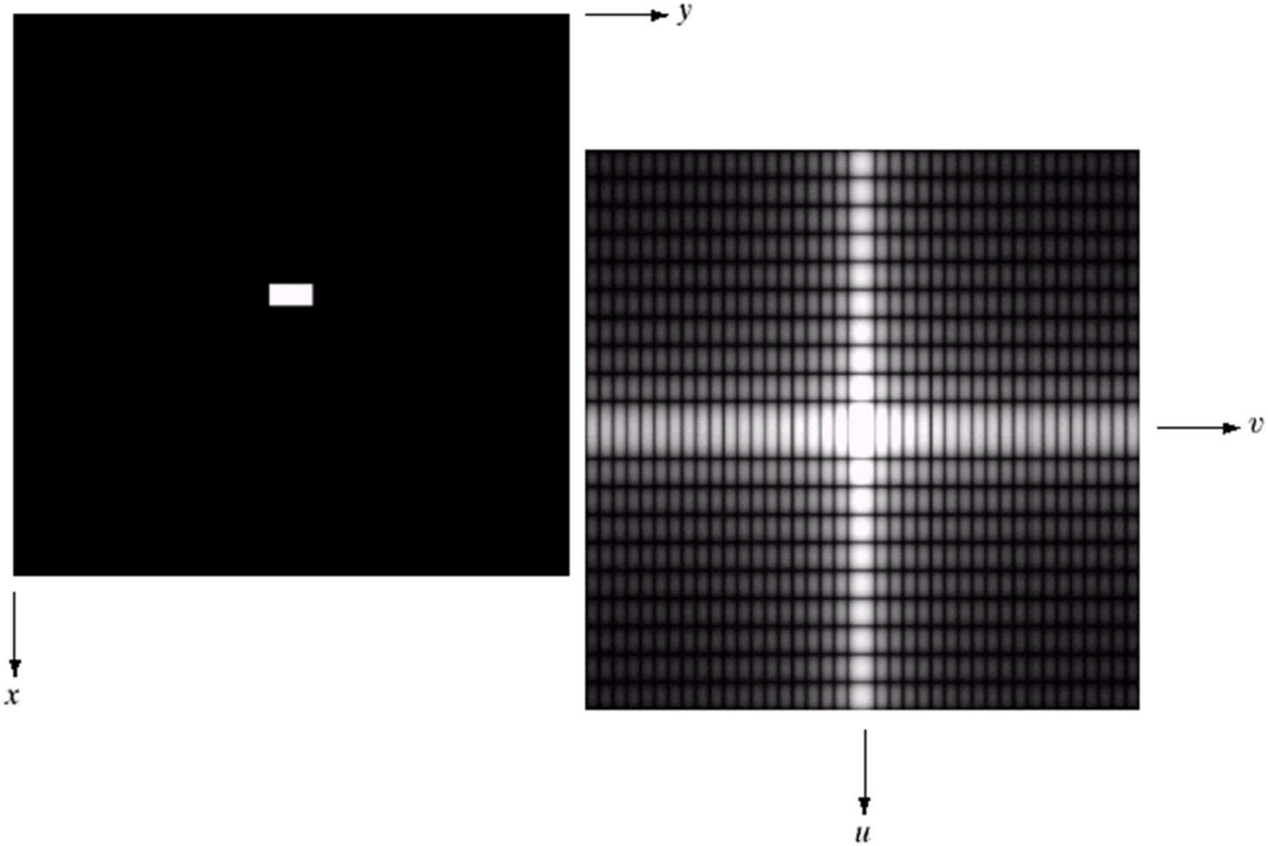




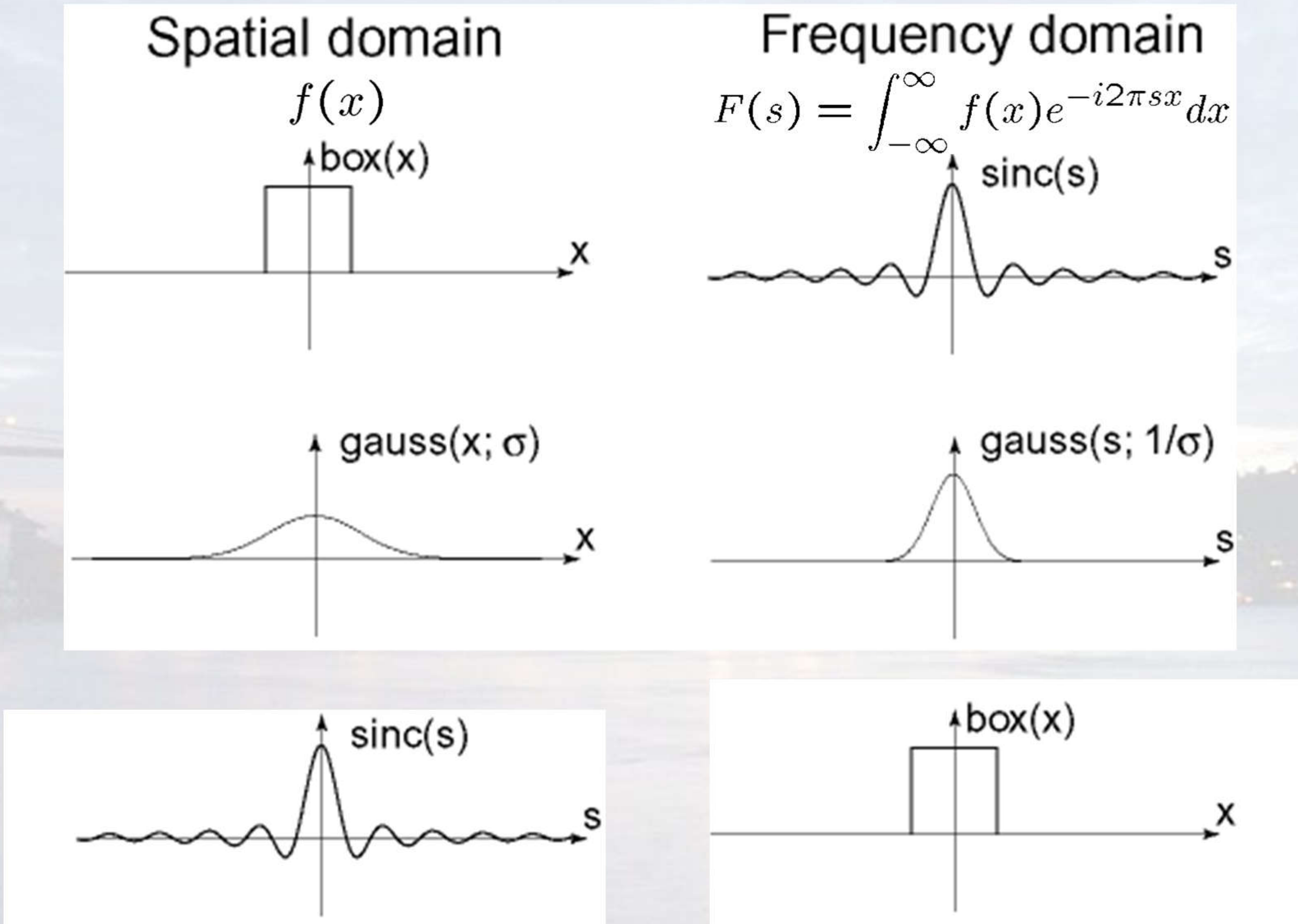
Example

a b

FIGURE 4.3
(a) Image of a 20×40 white rectangle on a black background of size 512×512 pixels.
(b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.



More examples

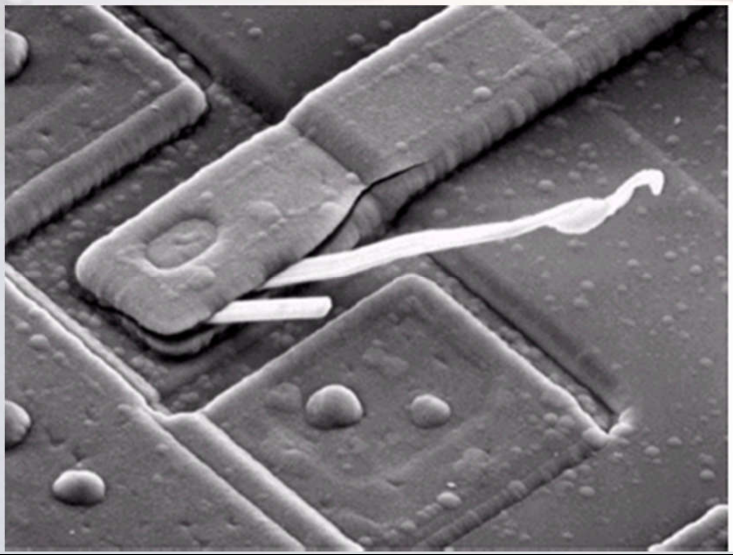


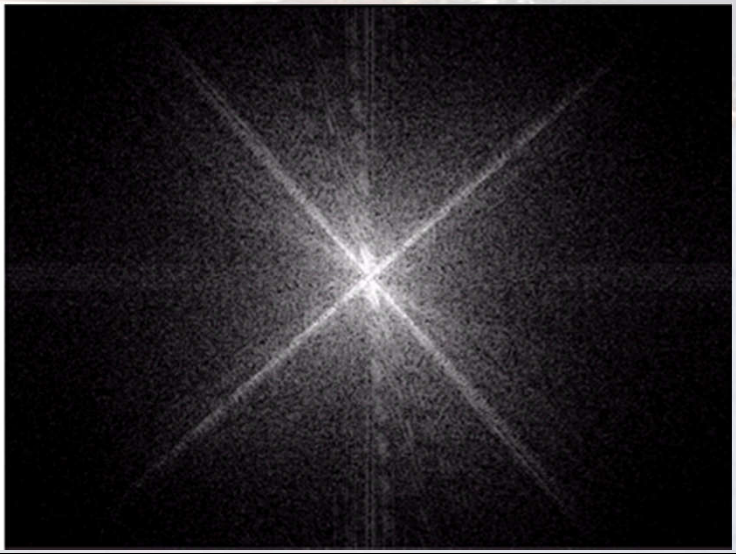
Adapted from Alexei Efros, CMU

The “meaning” of FT: Since frequency is related directly to rate of change, frequencies in the Fourier transform can be associated with the **patterns of intensity variations** in an image.

Low frequencies correspond to the slowly varying components of the image.

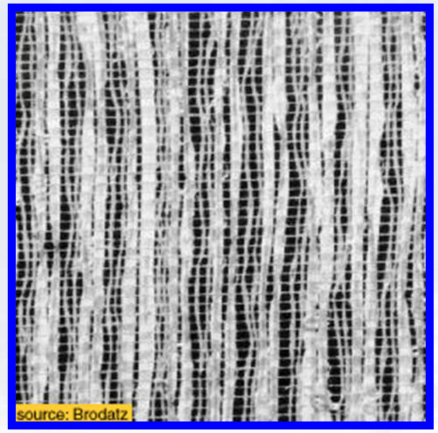
High frequencies correspond to fastly varying components of the image such as edges and noise.

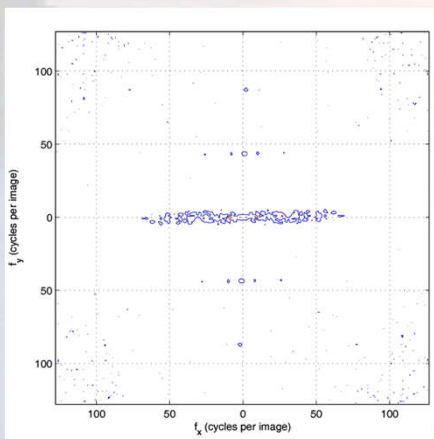


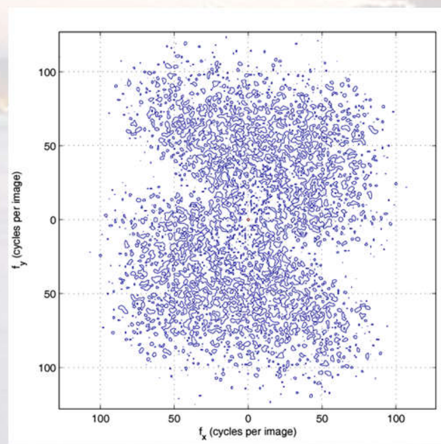


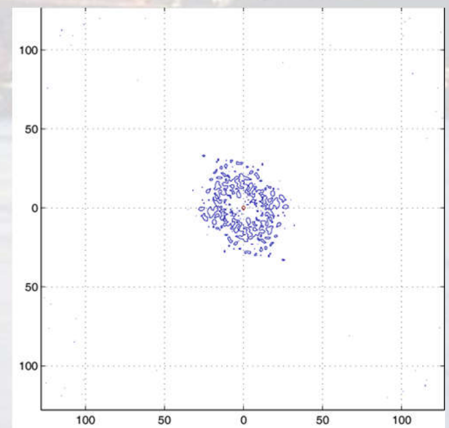
Linear image processing

Can you match the images with the spectra?







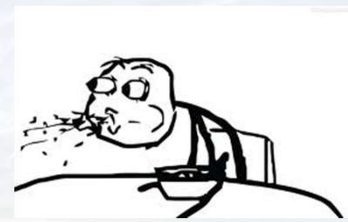


Adapted from Selim Aksoy, Bilkent U.

Filtering the frequency domain relies on the **Convolution Theorem**:

$$f(x,y)*h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

$$f(x,y)h(x,y) \Leftrightarrow F(u,v)*H(u,v)$$



Convolving in the spatial domain, is the same as multiplying in the frequency domain and vice versa. What does this mean for us?

It means:

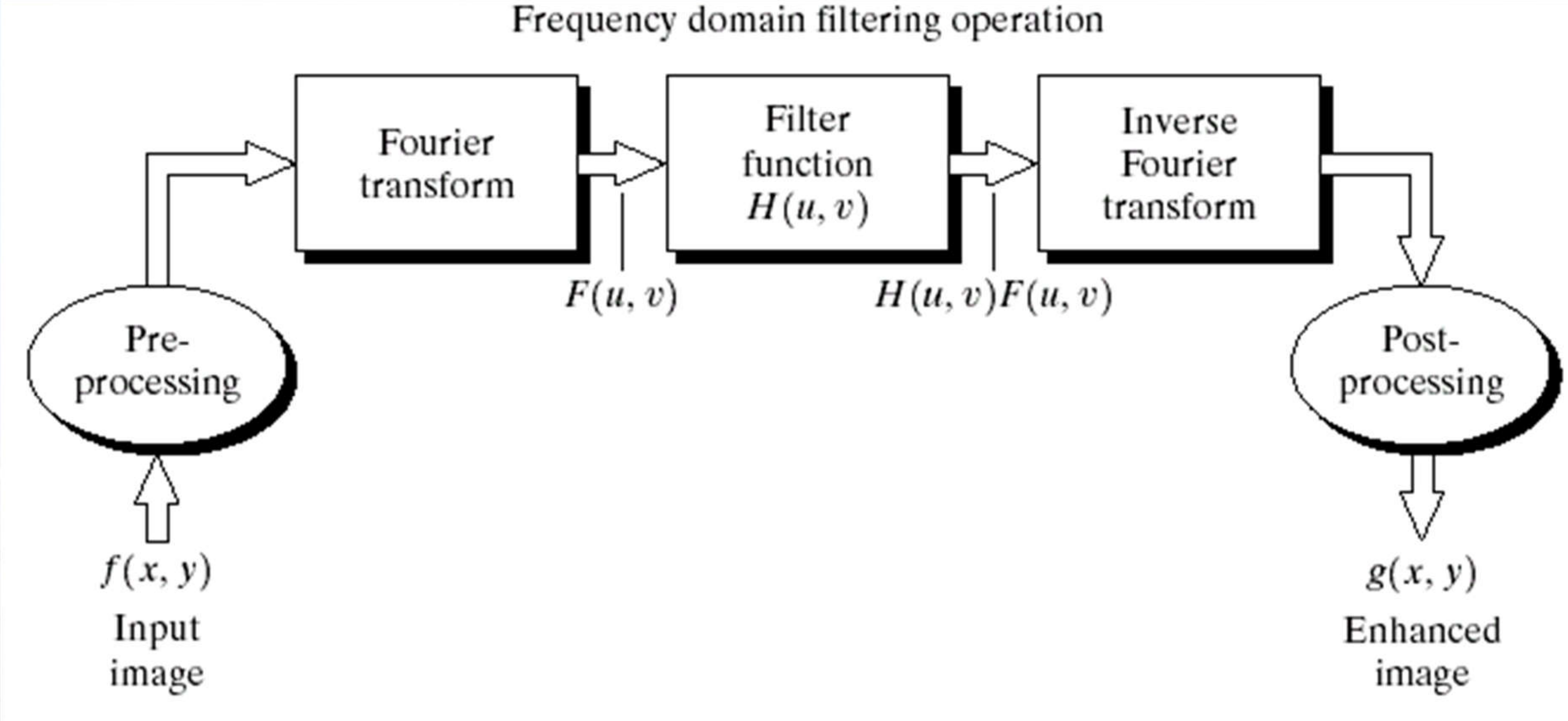
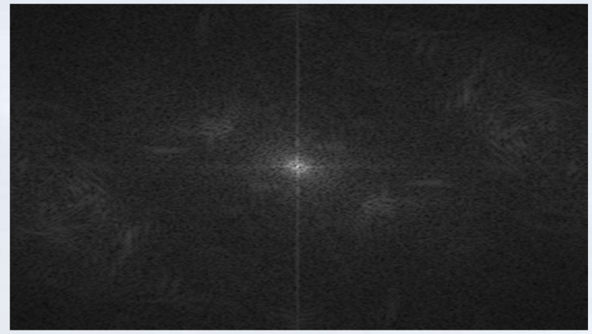


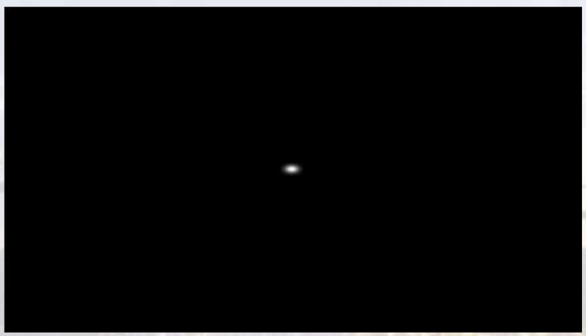
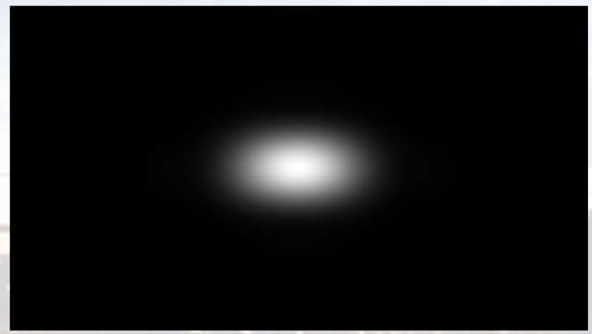
FIGURE 4.5 Basic steps for filtering in the frequency domain.

1. Compute the FT $F(u, v)$ of the input image $f(x, y)$
2. Compute the FT $H(u, v)$ of the filter
3. Compute $G(u, v,) = H(u, v,)F(u, v)$
4. Compute the inverse FT of $G(u, v)$ to get the output $g(x, y)$

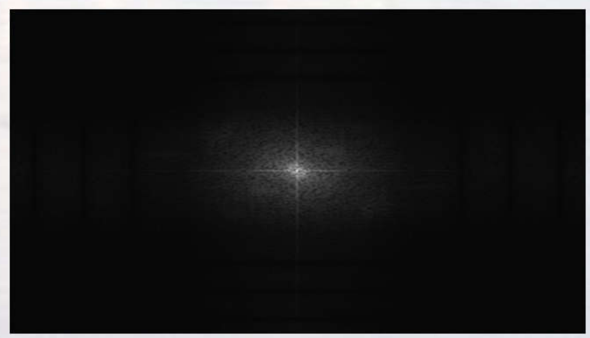
Linear image processing

$f(x,y)$   $|F(u,v)|$

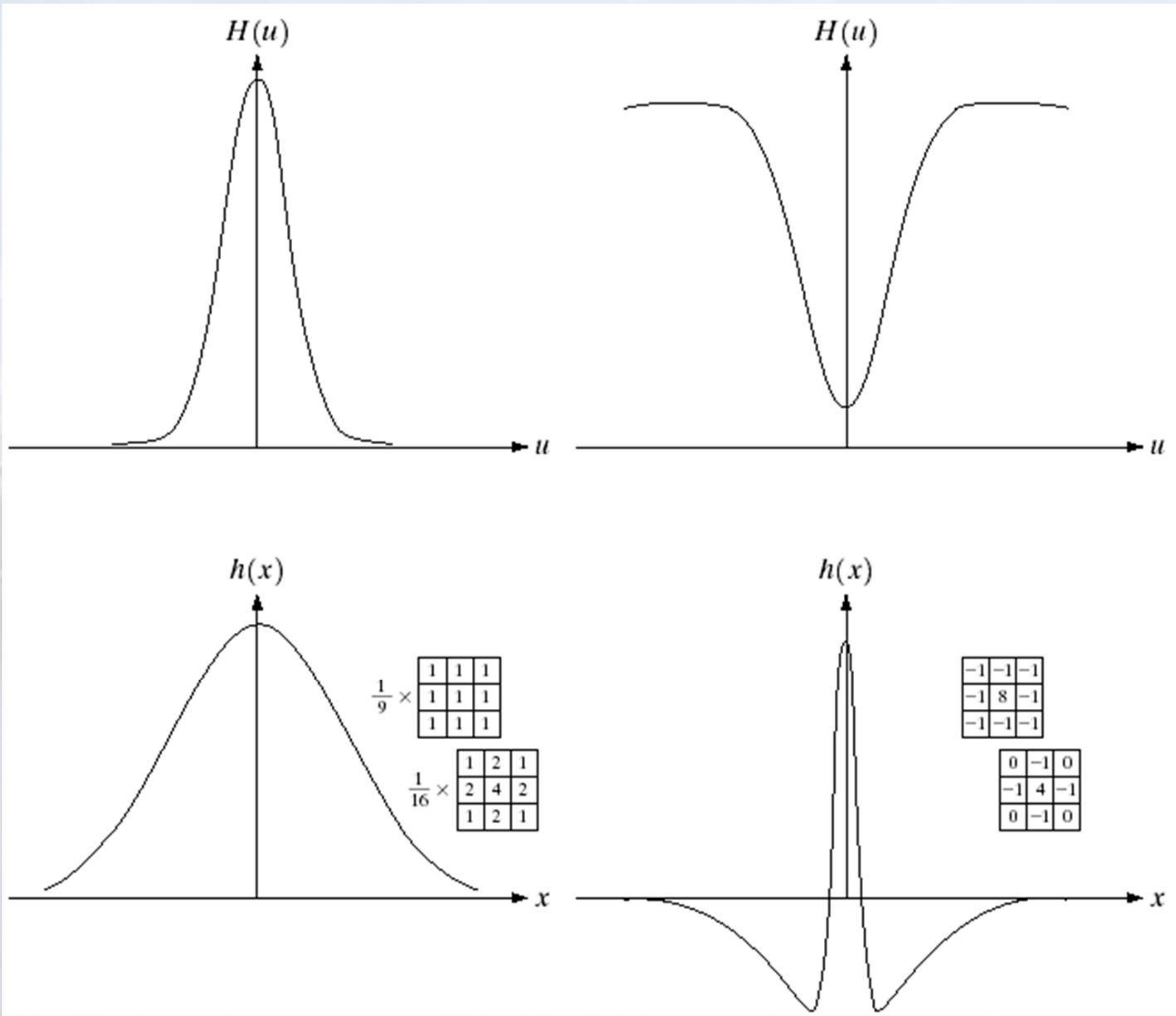
$*$ $\times$

$h(x,y)$   $|H(u,v)|$

$\Downarrow$ $\Downarrow$

$g(x,y)$   $|G(u,v)|$

Some familiar filters: **lowpass** and **highpass**



a b
c d

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.

Application result of lowpass and highpass filters

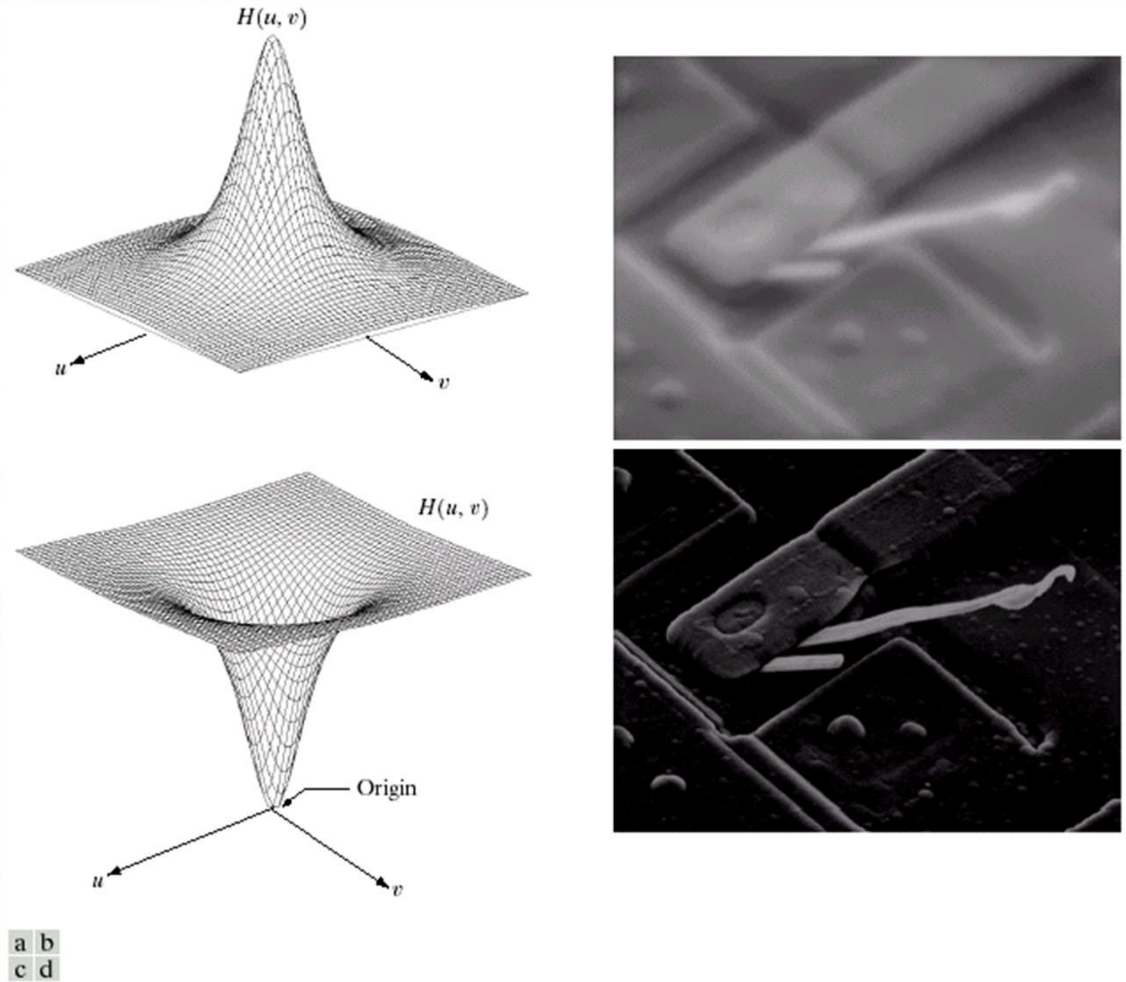
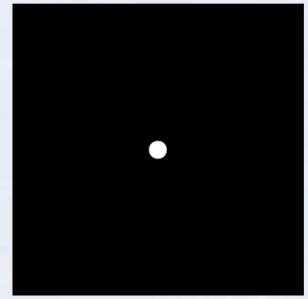
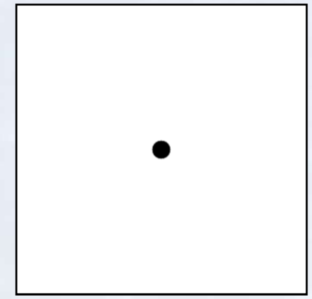
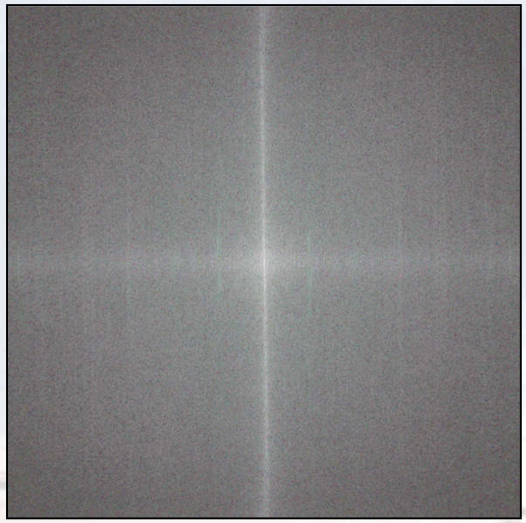


FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

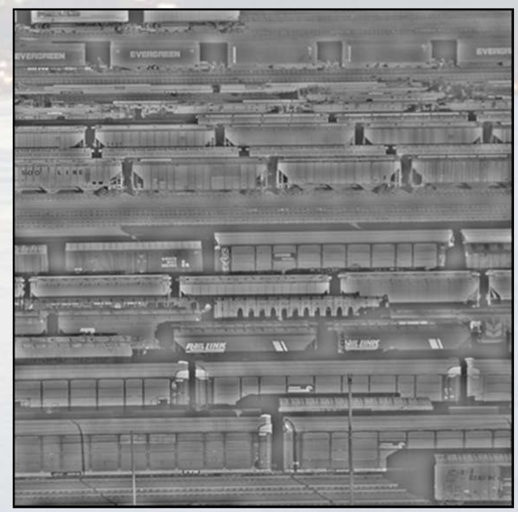
Ideal filters however **do not** produce ideal results!

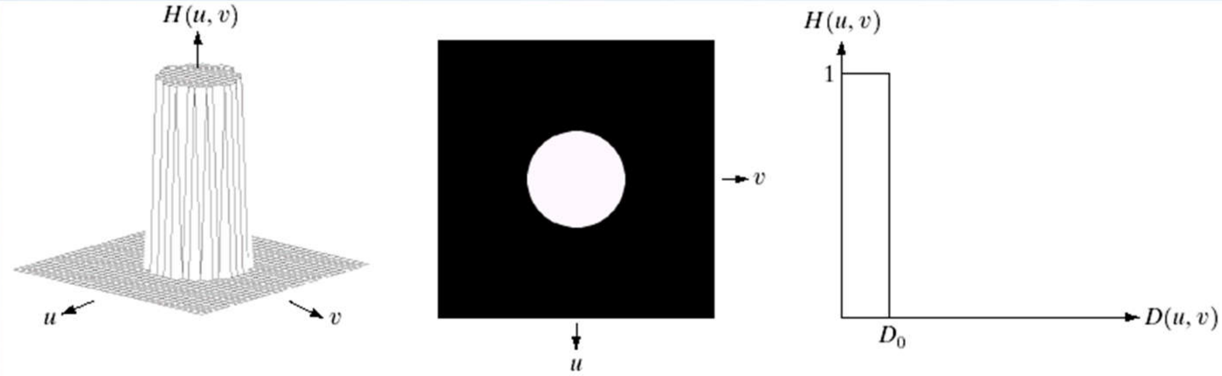






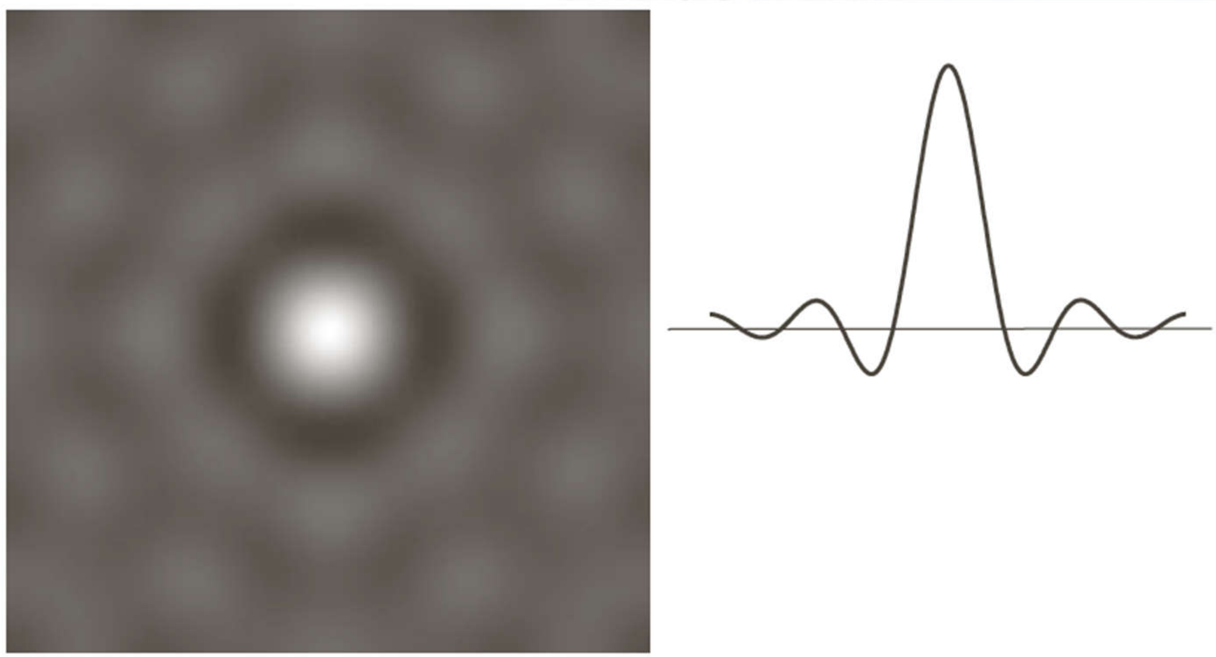
There is a “ringing” effect!





a b c

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

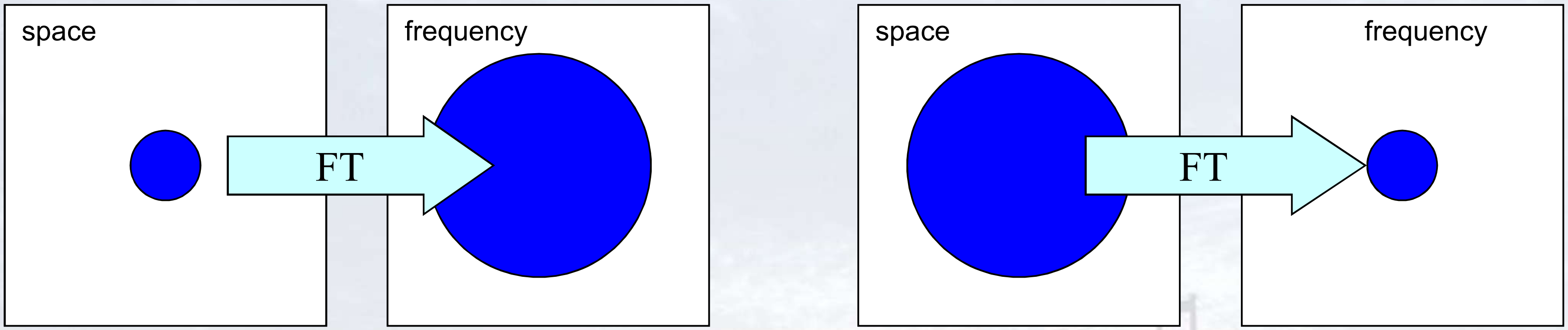


a b

FIGURE 4.43 (a) Representation in the spatial domain of an ILPF of radius 5 and size 1000×1000. (b) Intensity profile of a horizontal line passing through the center of the image.

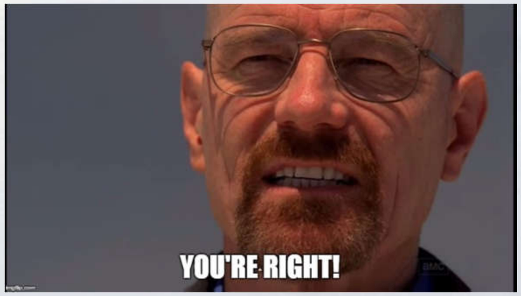
Linear image processing

The reason is related to the **uncertainty principle**.

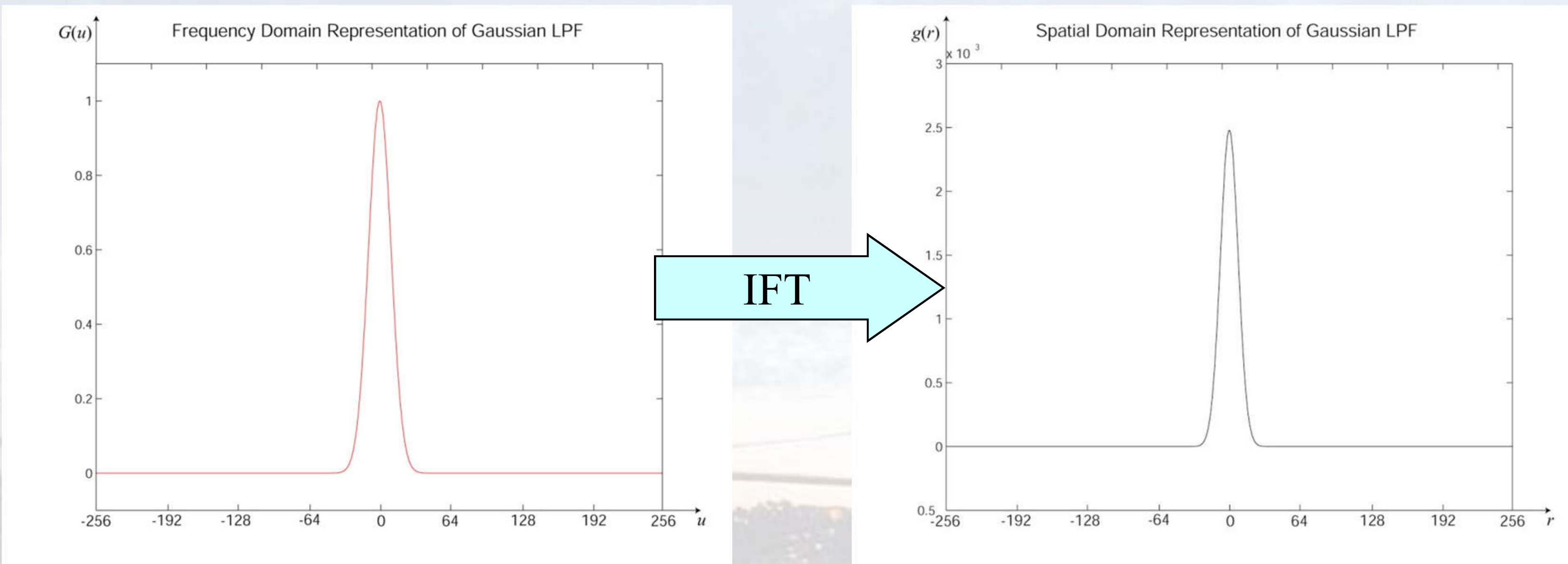


If $\Delta x \Delta y$ is the extent of the object in space and if $\Delta u \Delta v$ is the extent in frequency then, $\Delta x \Delta y \Delta u \Delta v \geq \frac{1}{16\pi^2}$

A small object in space has a large frequency extent and vice-versa.



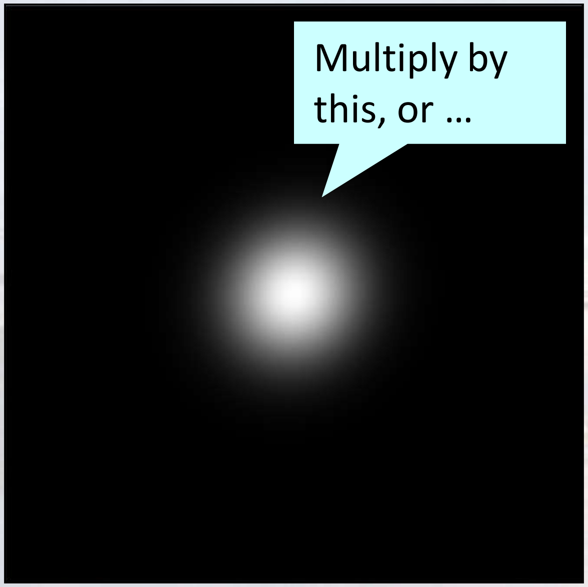
- One more reason to like the Gaussian filter

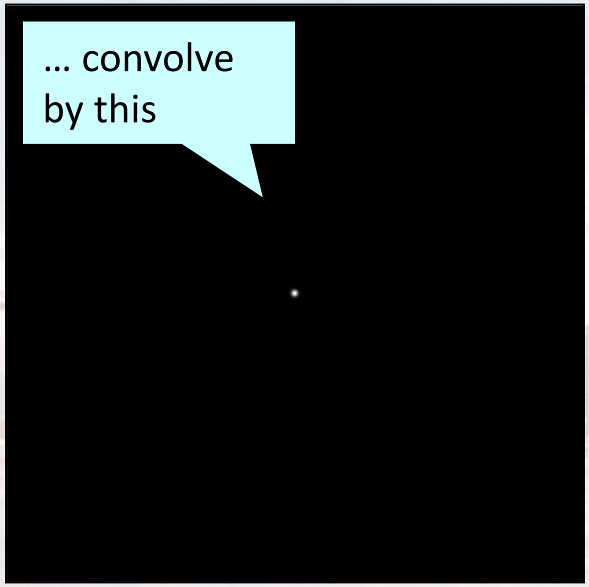


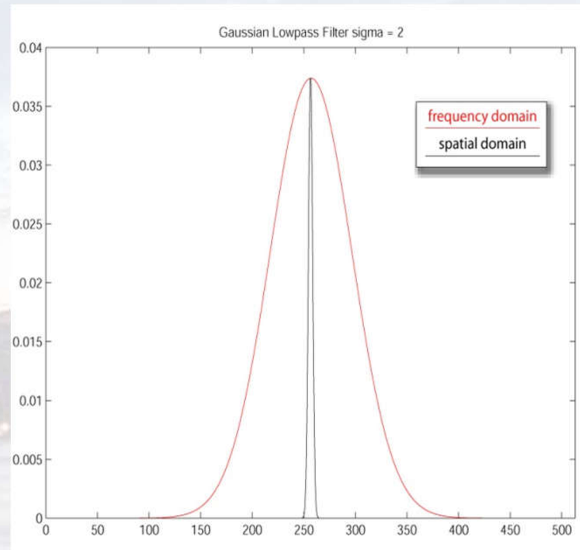
The Gaussian filter optimizes the uncertainty relation.
It provides the sharpest cutoff with the least ringing.

Gaussian low-pass filter

Image size: 512x512
SD filter sigma = 2





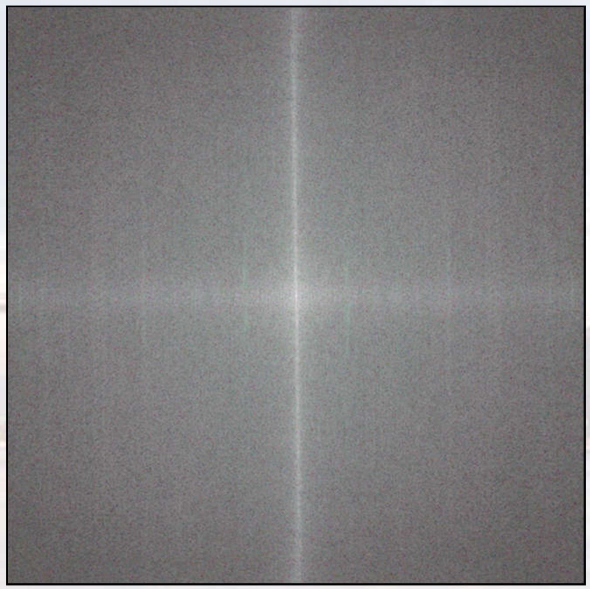


Fourier Domain Rep. | Spatial Representation | Central Profile

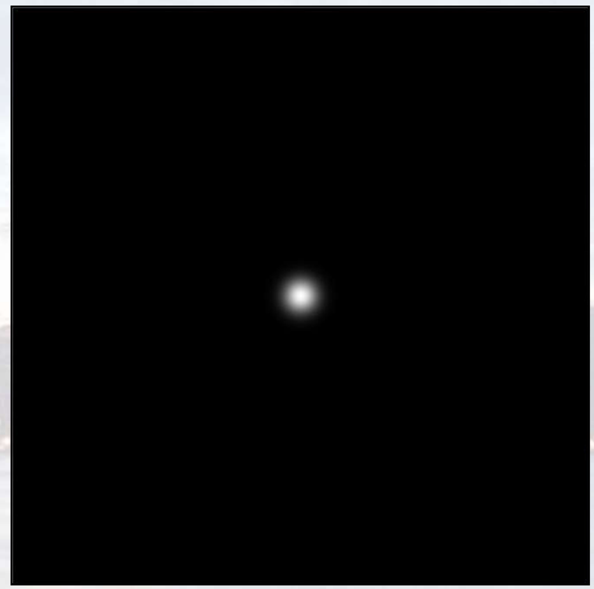
Gaussian low-pass filter

Image size: 512x512
SD filter sigma = 8

Original Image



Power Spectrum

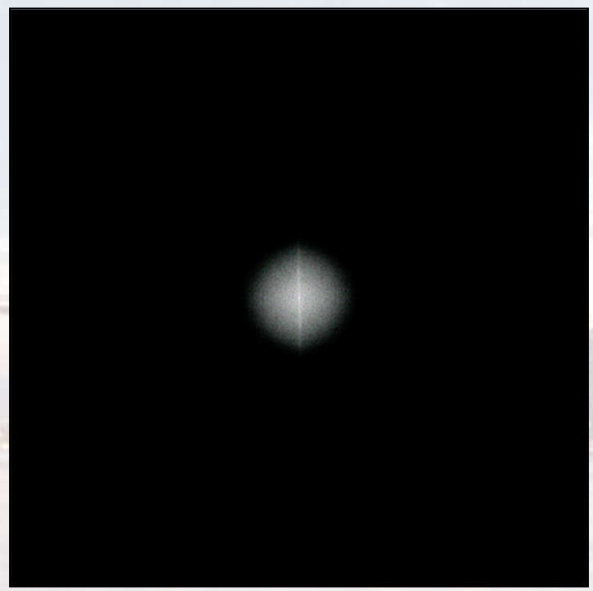


Gaussian LPF in FD

Gaussian low-pass filter

Image size: 512x512
SD filter sigma = 8

Filtered Image

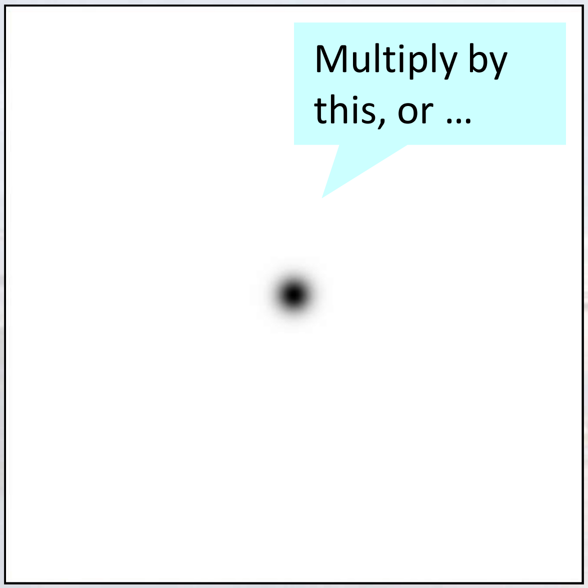


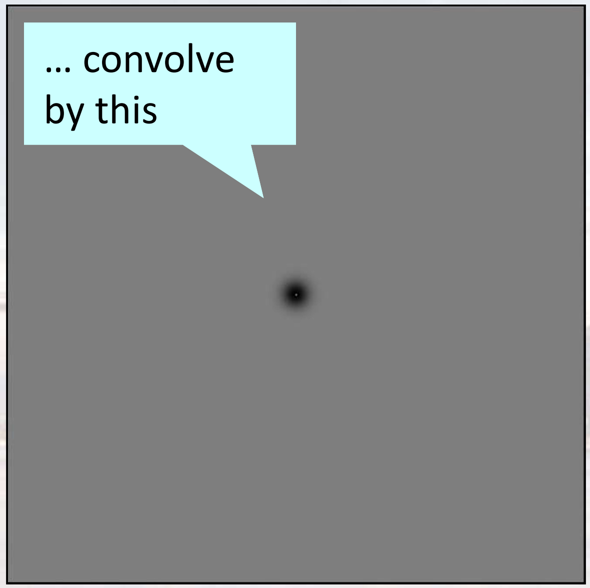
Filtered Power Spectrum

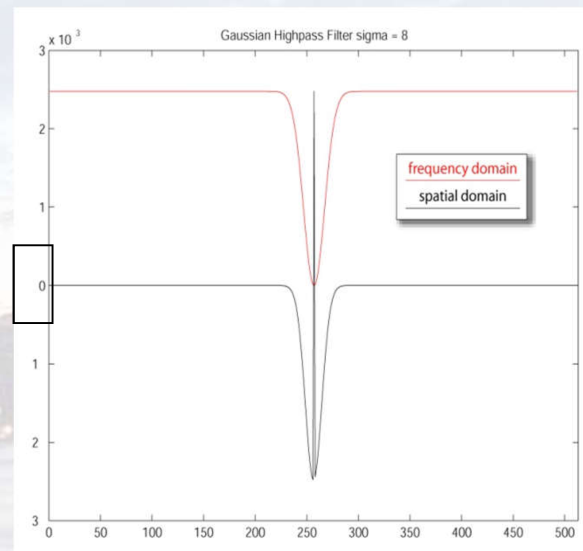
Original Image

Gaussian highpass filter

Image size: 512x512
FD notch sigma = 8







Fourier Domain Rep.

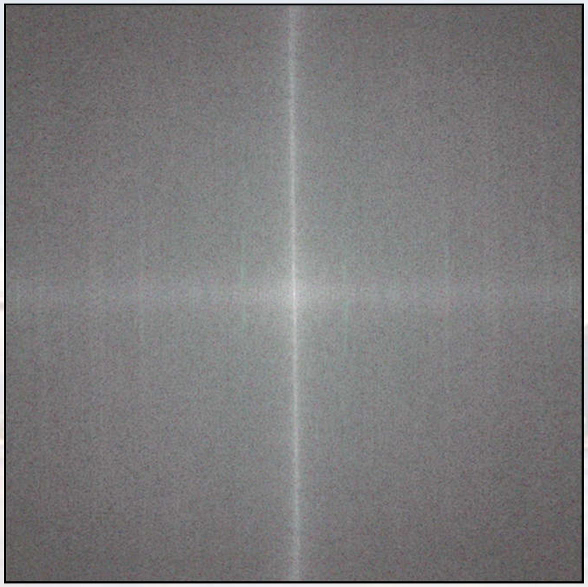
Spatial Representation

Central Profile

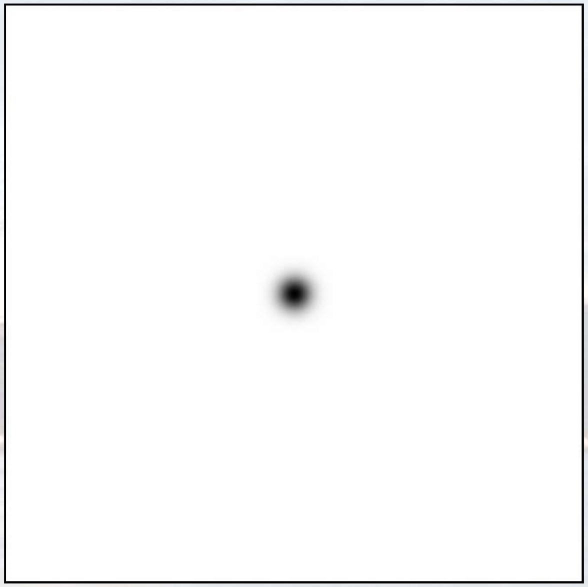
Gaussian highpass filter

Image size: 512x512
FD notch sigma = 8

Original Image



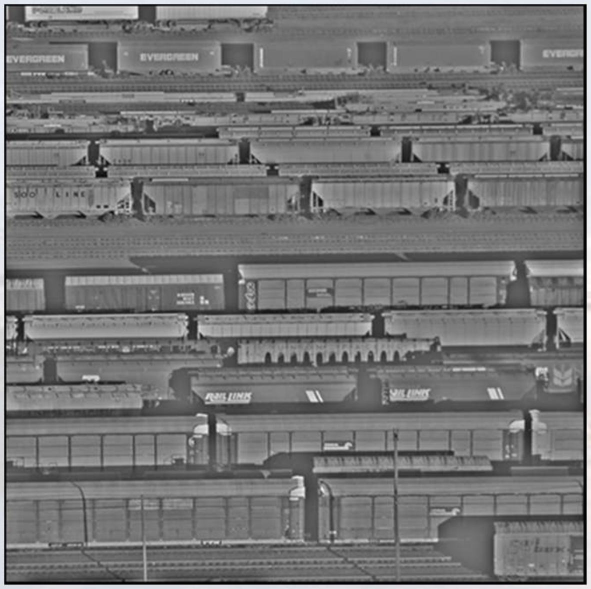
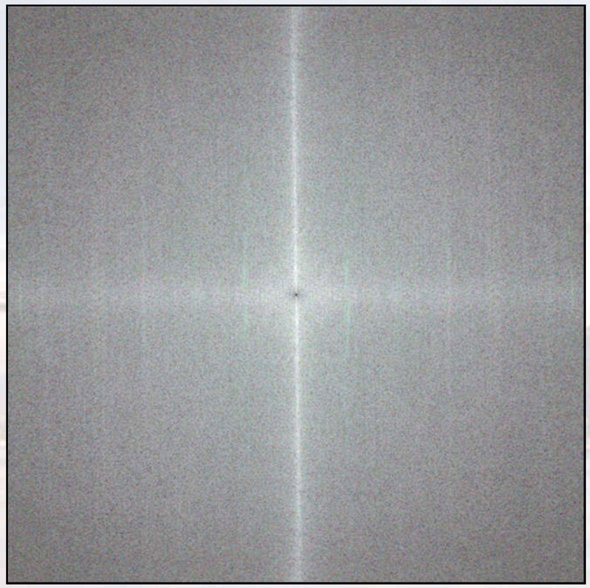
Power Spectrum



Gaussian HPF in FD

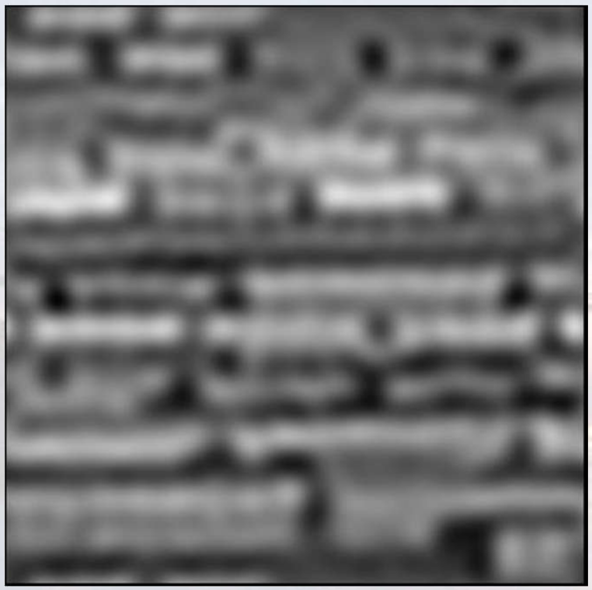
Gaussian highpass filter

Image size: 512x512
FD notch sigma = 8

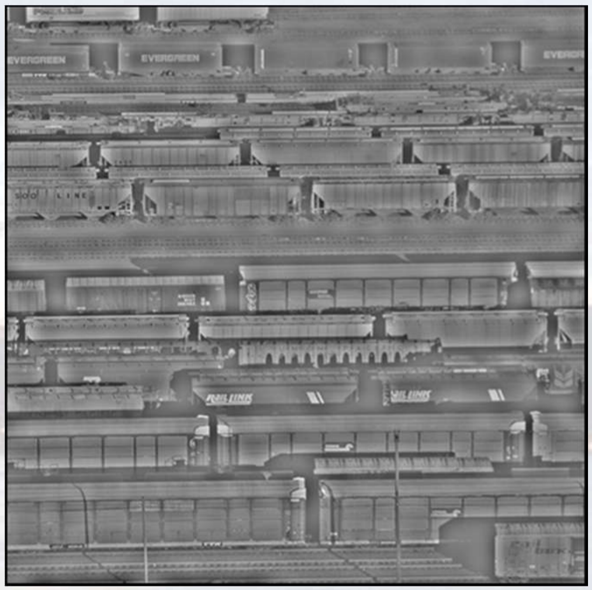

Filtered Image | Filtered Power Spectrum | Original Image

Comparison of ideal and Gaussian filters



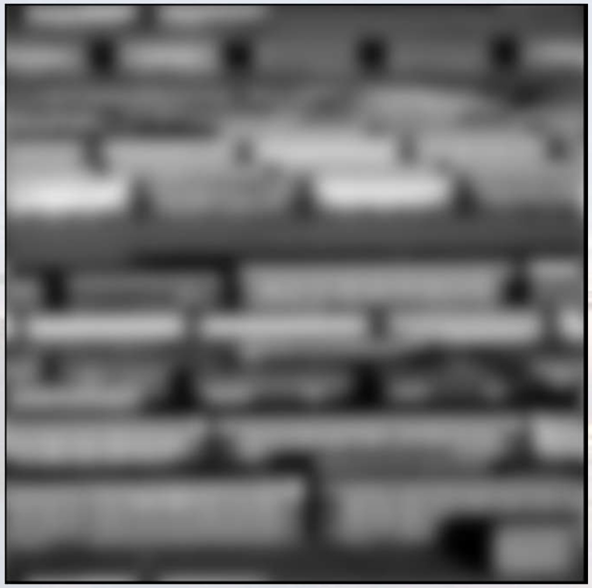
Ideal LPF

Original Image



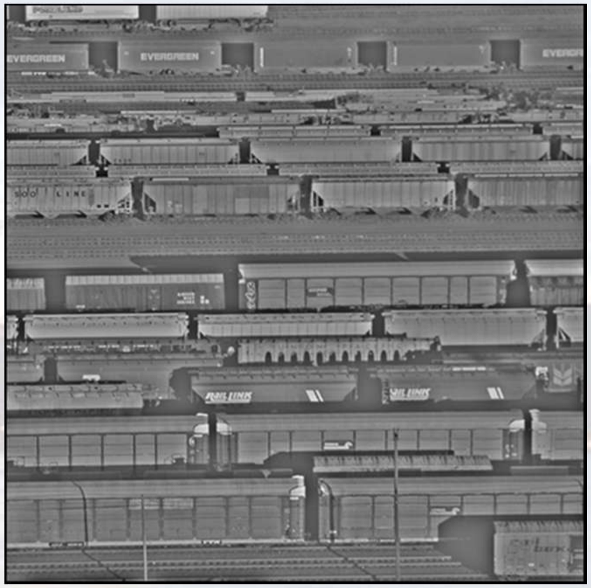
Ideal HPF

Comparison of ideal and Gaussian filters



Gaussian LPF

Original Image

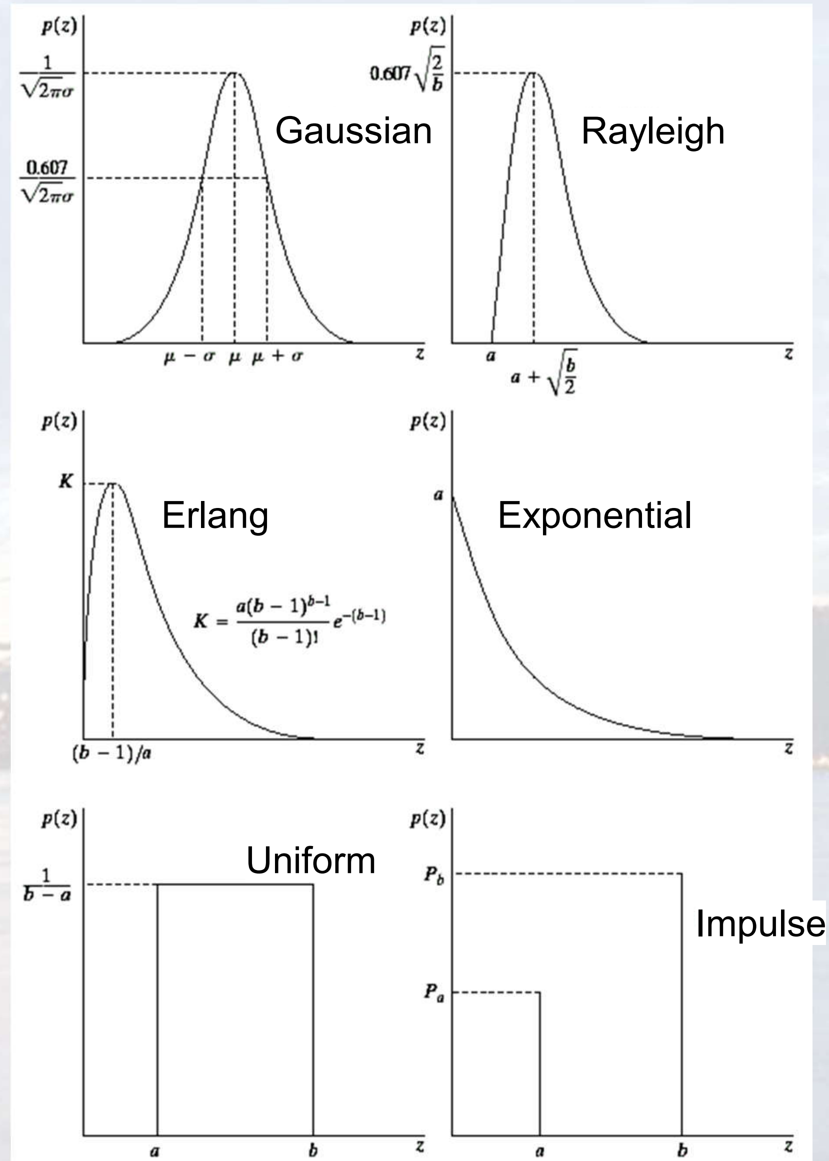


Gaussian HPF

Linear image processing: about noise

Cameras are imperfect; **noise** is part of the deal.

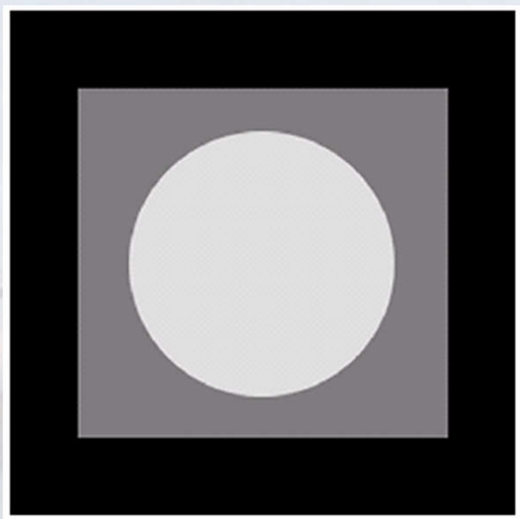
There are many **types** of image noise, e.g. Gaussian, Rayleigh, Erlang, Exponential, Uniform, Impulse, etc.

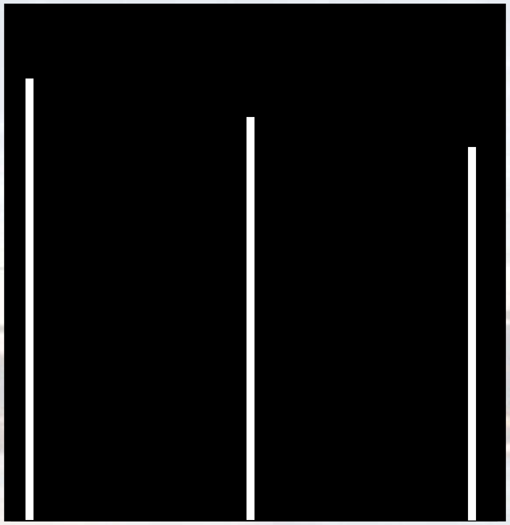


Linear image processing: about noise

Let’s take a closer look to noise types

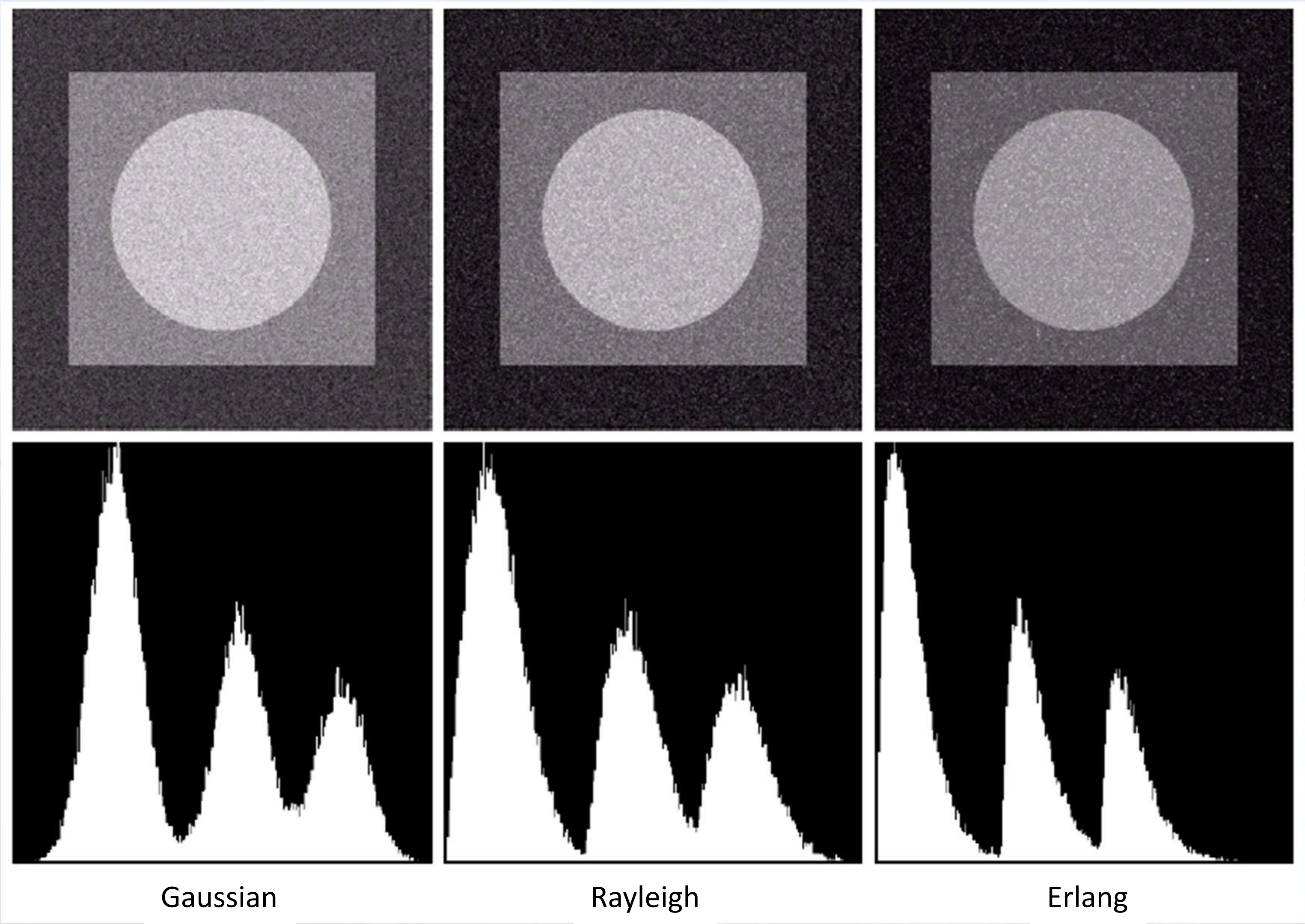
Noise free





Histogram

Linear image processing: about noise

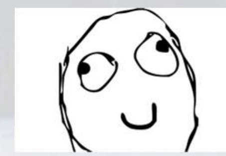


Linear image processing: about noise

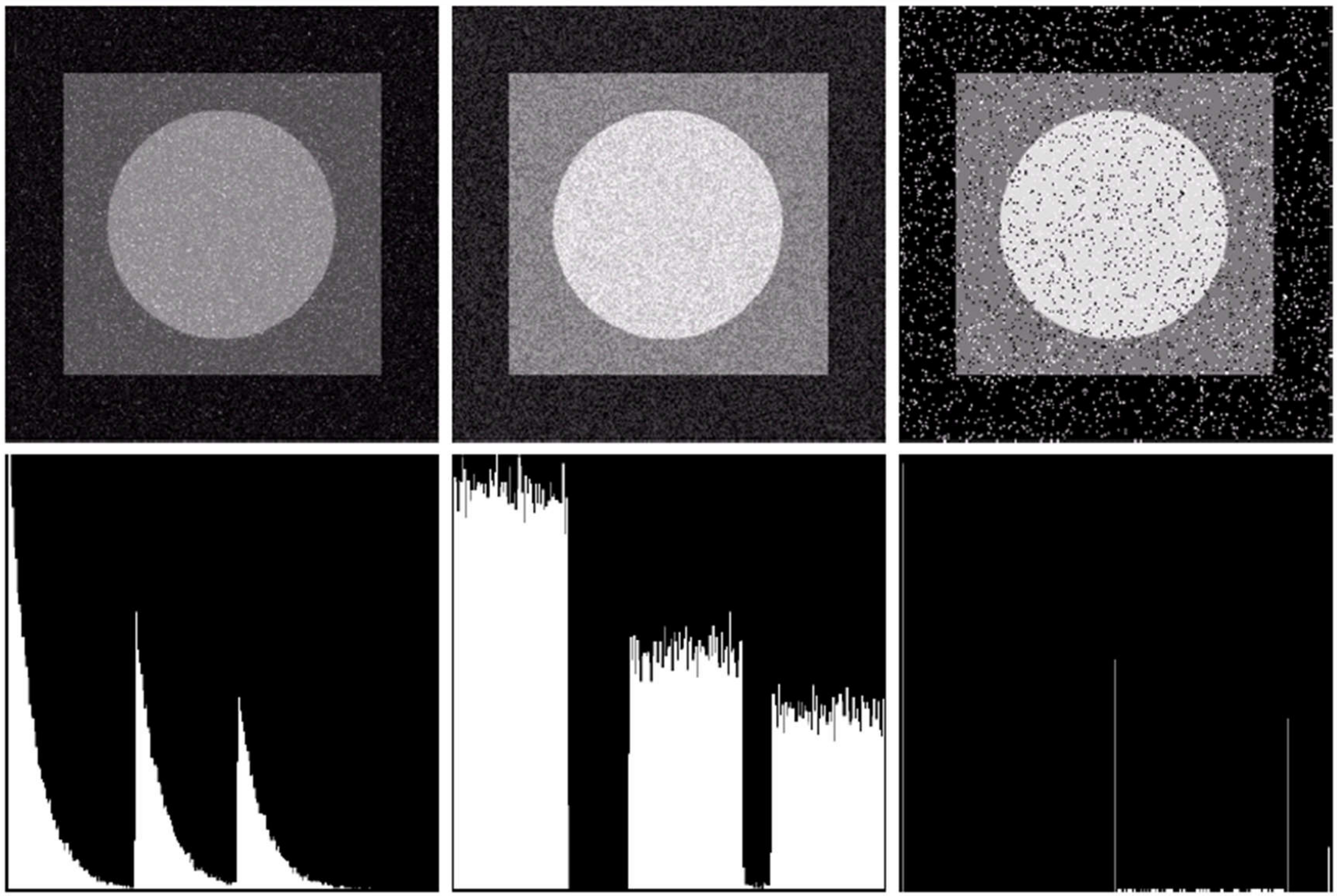
Gaussian noise

intensity variations that follow a Gaussian distribution, caused mostly by thermal interference.

can be removed with Gaussian filters



Linear image processing: about noise

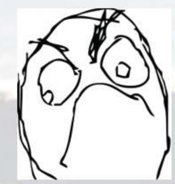


Exponential Uniform [1] Impulse

Salt and pepper noise

Random white/dark pixels, due to transmission errors, analog to digital conversion errors, etc.

Gaussian filtering in this case is blending the image and noise together

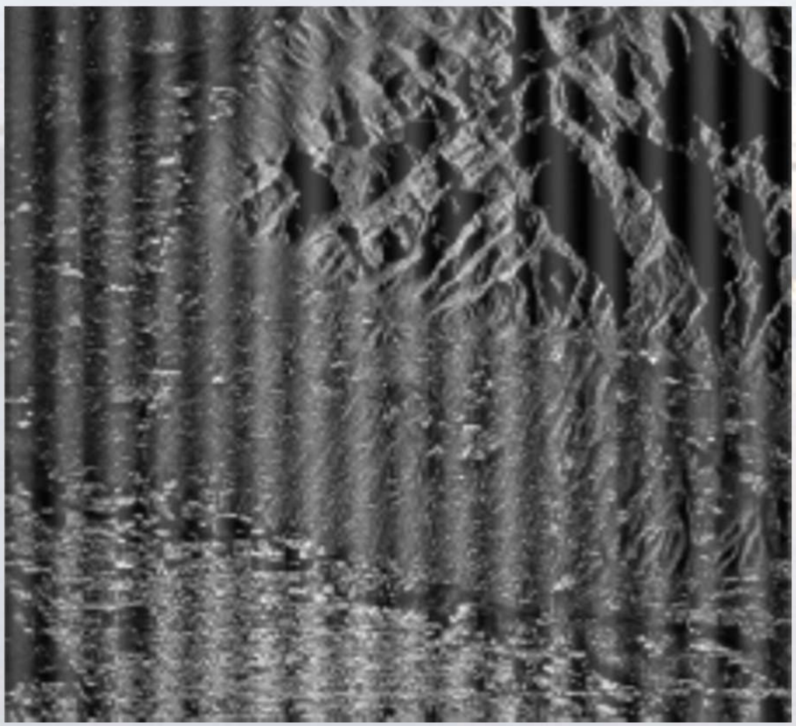


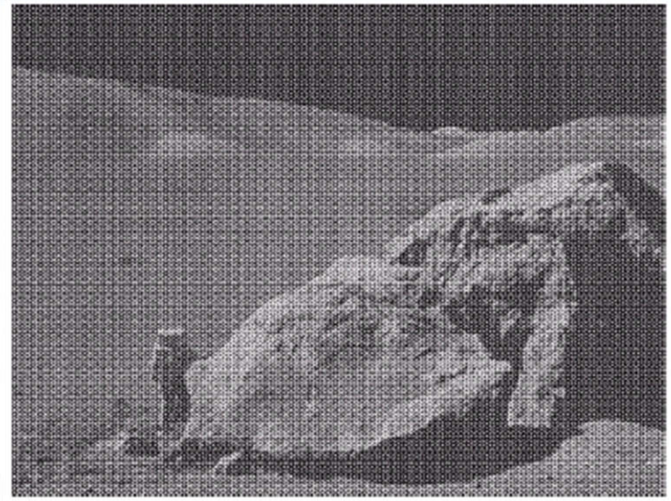
p = 10%

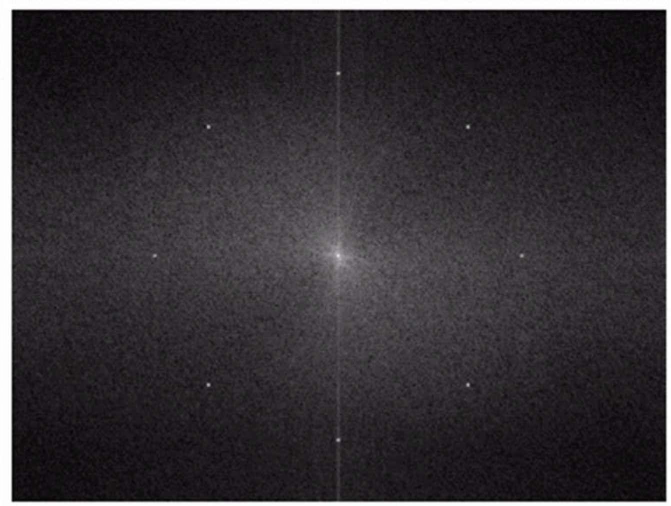
for increasing σ

- Structured/periodic noise: has a fixed amplitude, frequency and phase; caused by electronic interference

Frequency domain techniques are most effective!

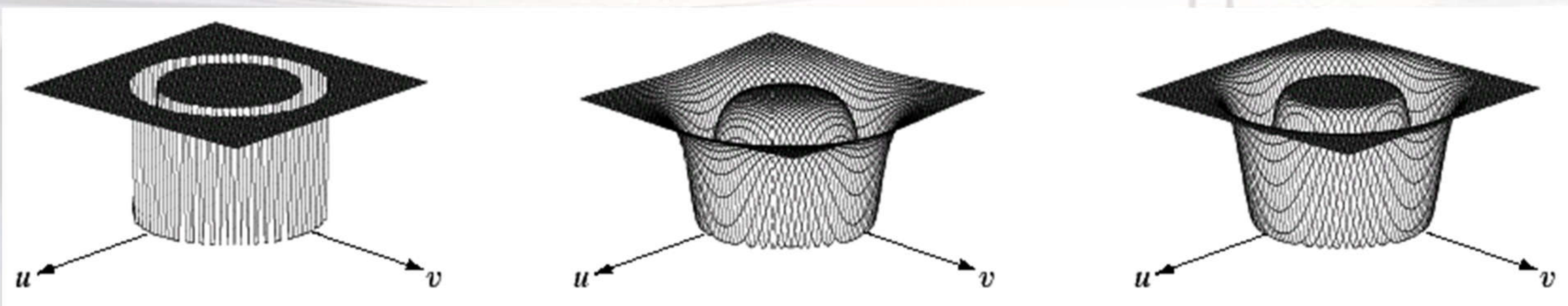






If you can identify the frequencies responsible for the noise, then just remove them with a **band reject** filter. An ideal band reject filter looks like this (and has ringing problems):

$$H(u,v)=\begin{cases}1 & if\ \ D(u,v)<D_0-\dfrac{W}{2}\\ 0 & if\ \ D_0-\dfrac{W}{2}\le D(u,v)\le D_0+\dfrac{W}{2}\\ 1 & if\ \ D(u,v)>D_0+\dfrac{W}{2}\end{cases}$$



Ideal Band Reject Filter

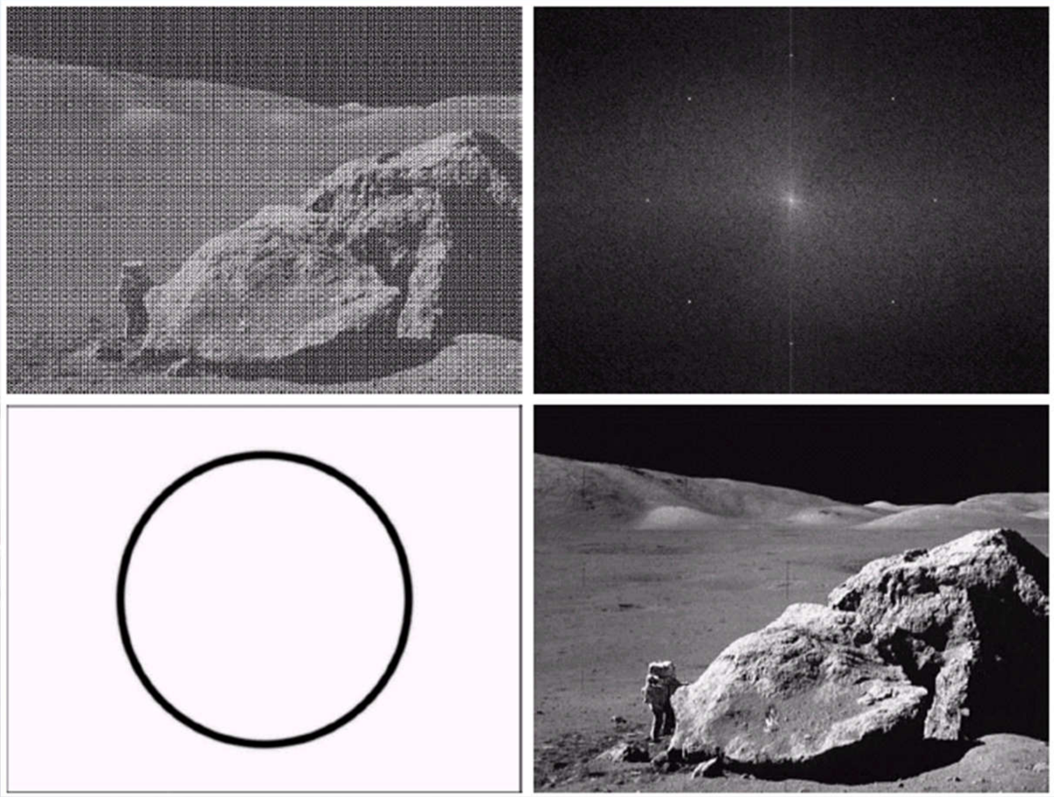
Butterworth Band Reject Filter (of order 1)

Gaussian Band Reject Filter

Linear image processing: about noise

Image corrupted by sinusoidal noise

Fourier spectrum of corrupted image



Butterworth band reject filter

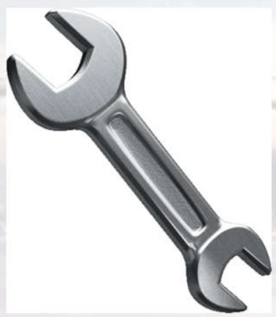
Filtered image

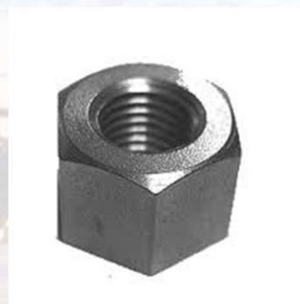
Linear image processing: about noise

Gaussian filter

Gaussian noise

???????????



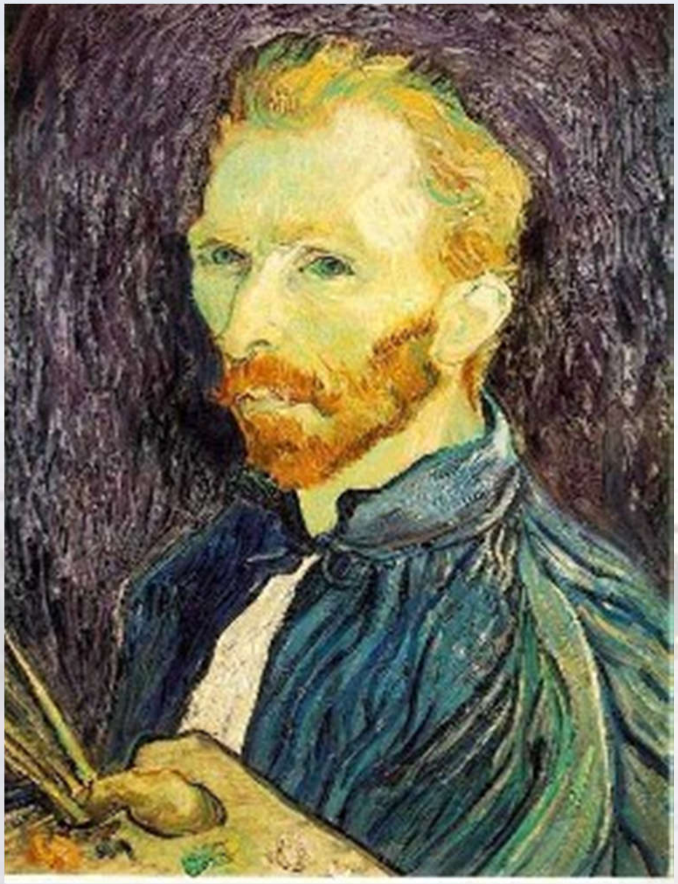


Salt-pepper / impulse noise

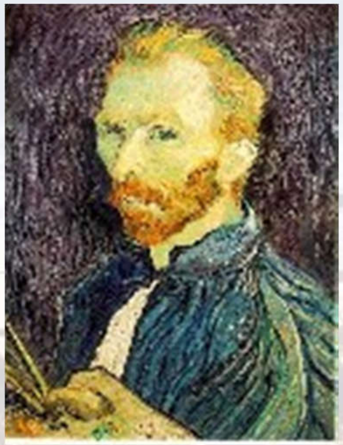
Frequency techniques

Periodic noise

Now that we know about Gaussian smoothing and FFT, let's revisit **subsampling**!

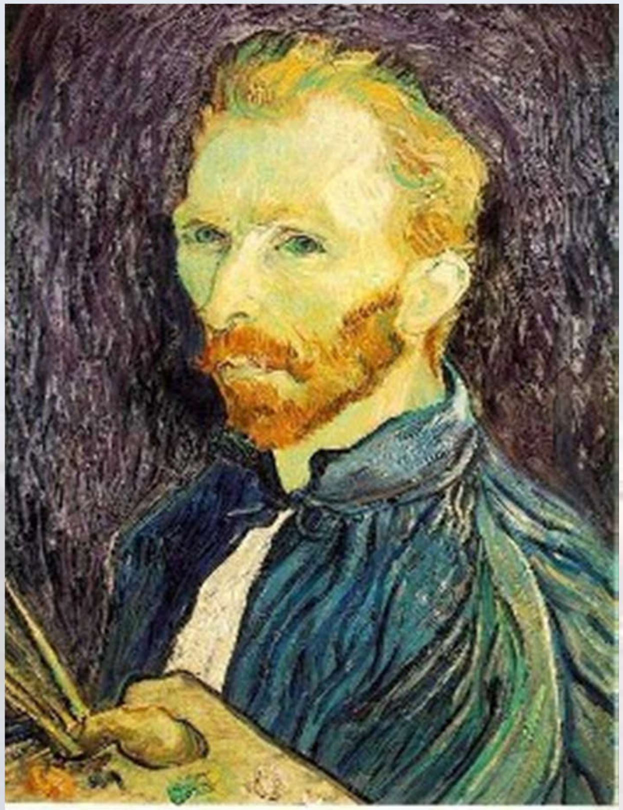


Throw away every other row and column to create a 1/2 size image

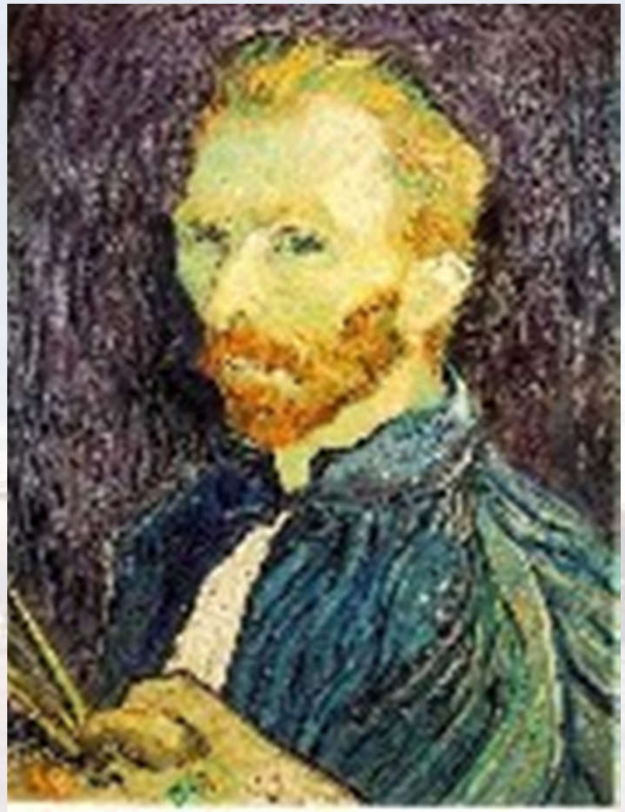


1/4

1/8



1/2



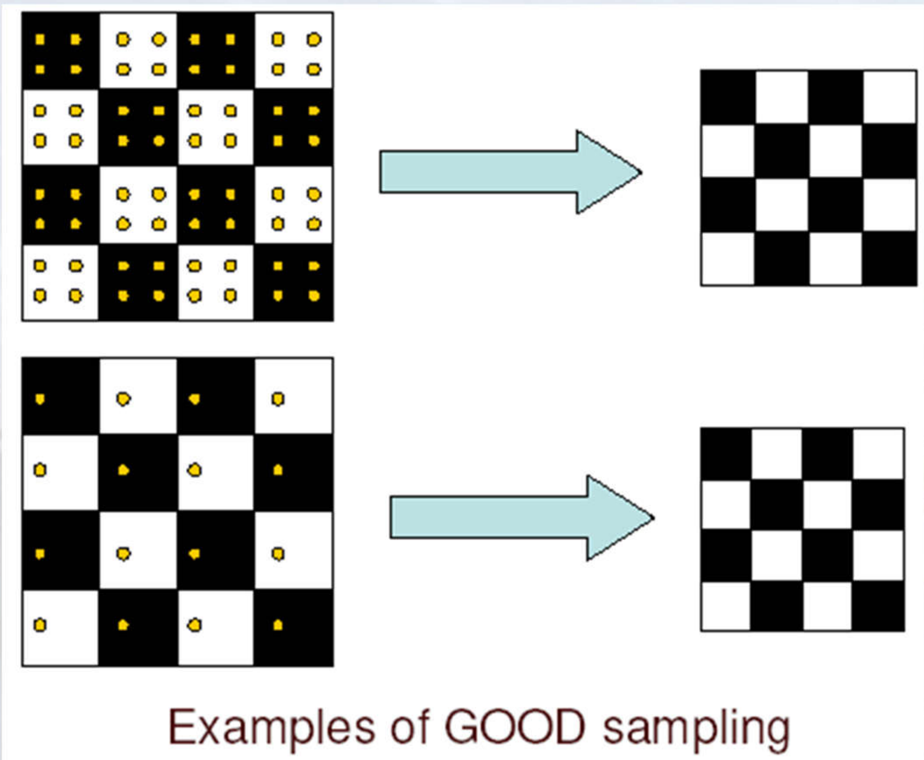
1/4 (2x zoom)

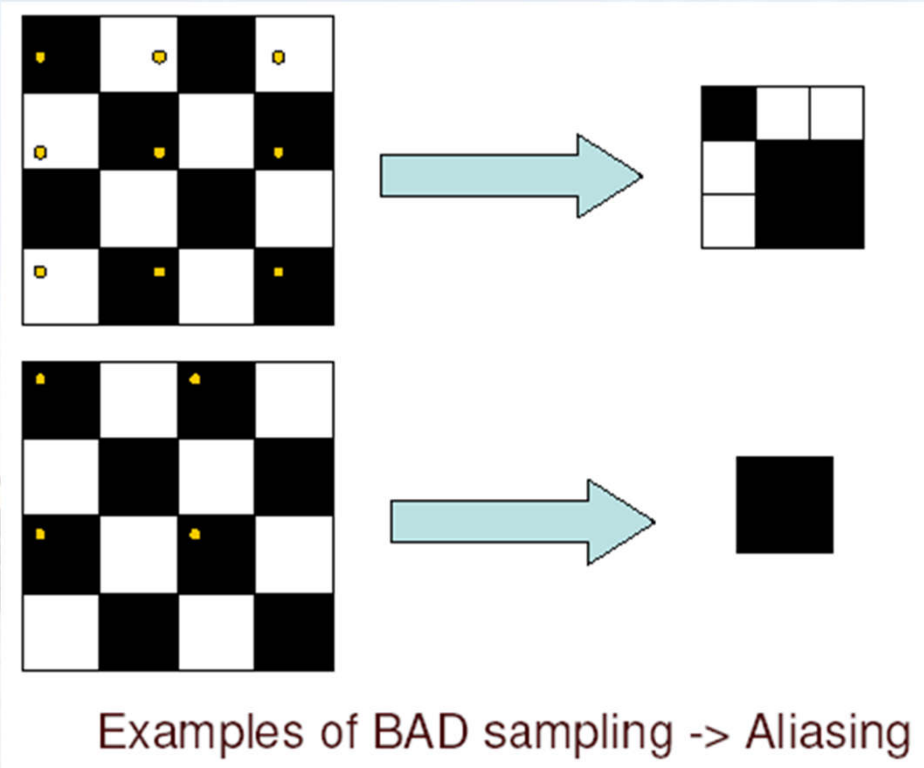
1/8 (4x zoom)

Why does this look so crufty?

Source: S. Seitz

Because of bad sampling!





- High spatial frequency components appear as low frequency
- Wagon wheels roll backwards
- Striped shirts look weird

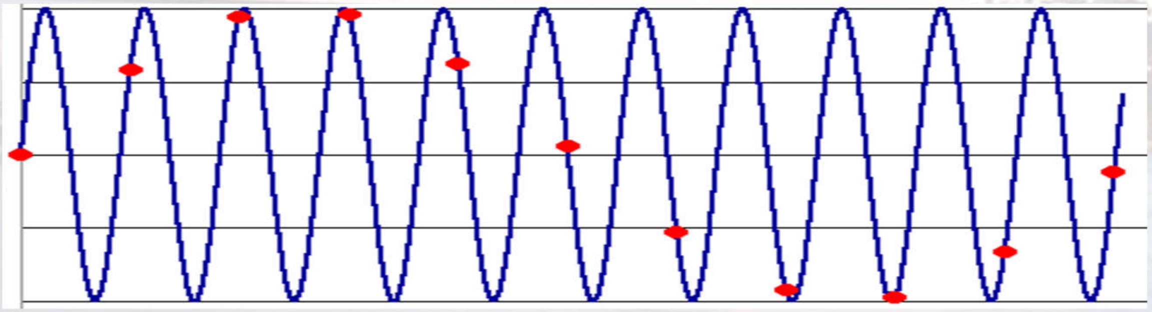
Another sampling gone wrong

Another sampling gone wrong (experienced newscasters never wear these…)

Occurs when your sampling rate is not high enough to capture the amount of detail in your image. It can give you the wrong signal/image—an *alias*

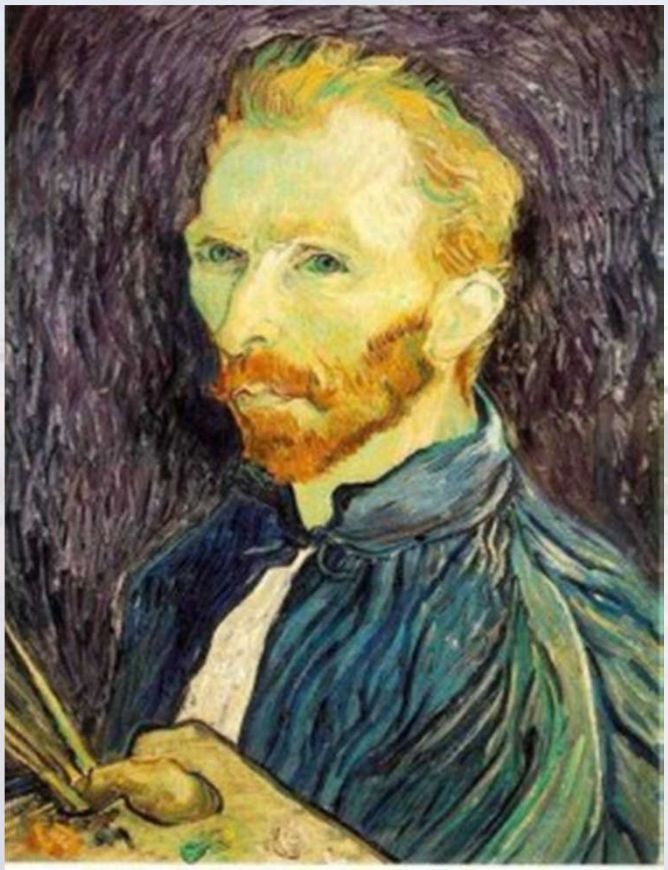
To avoid aliasing:

– sampling rate > 2 * max frequency in the image (Nyquist-Shannon Sampling Theorem)

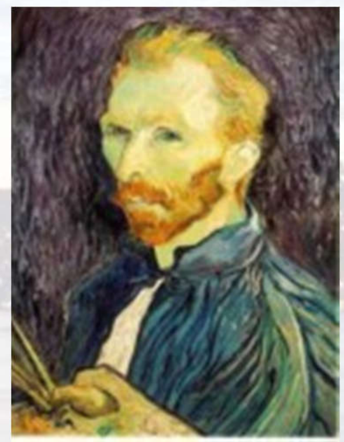


Aliasing

- When subsampling by a factor of two the original image has frequencies that are too high, so we smooth out (**anti-aliasing filtering**) those high frequencies/fine details!

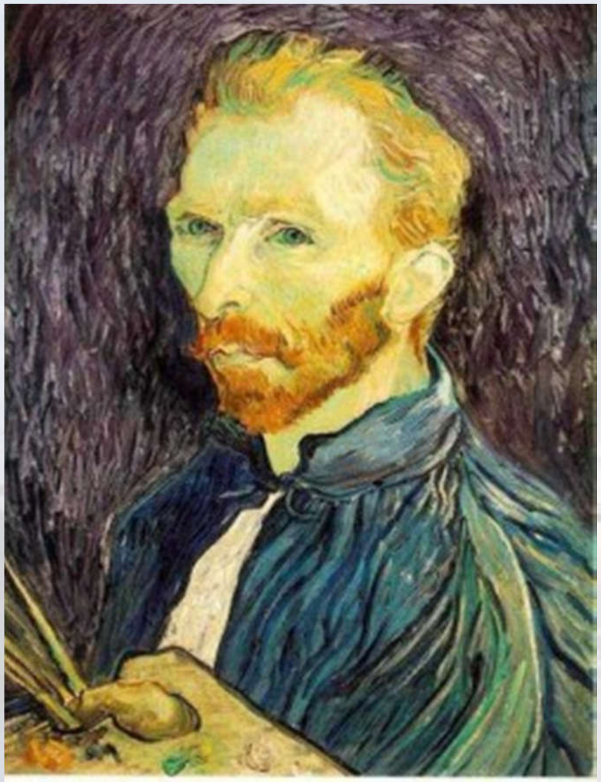


Gaussian 1/2

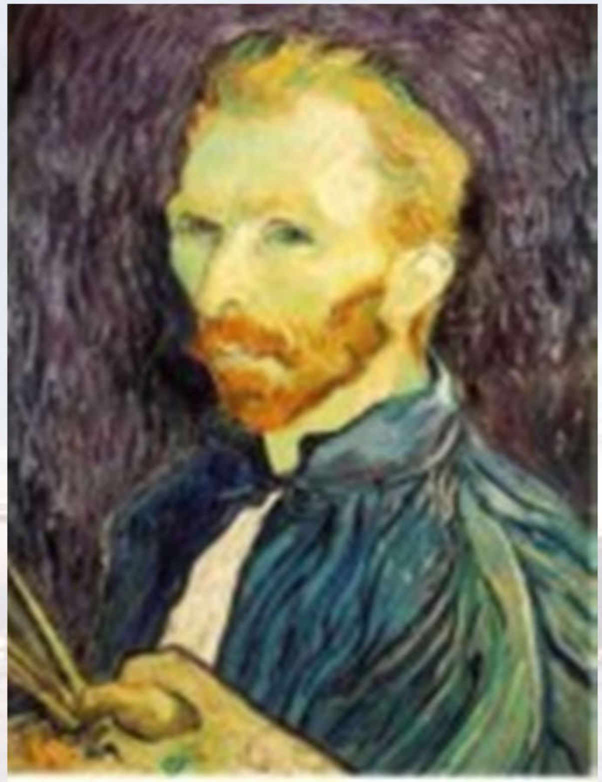


G 1/4

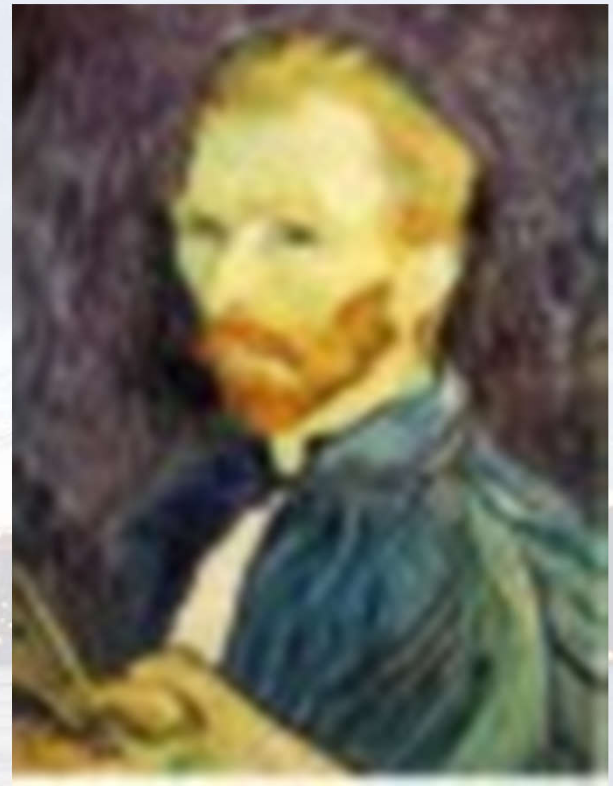
G 1/8



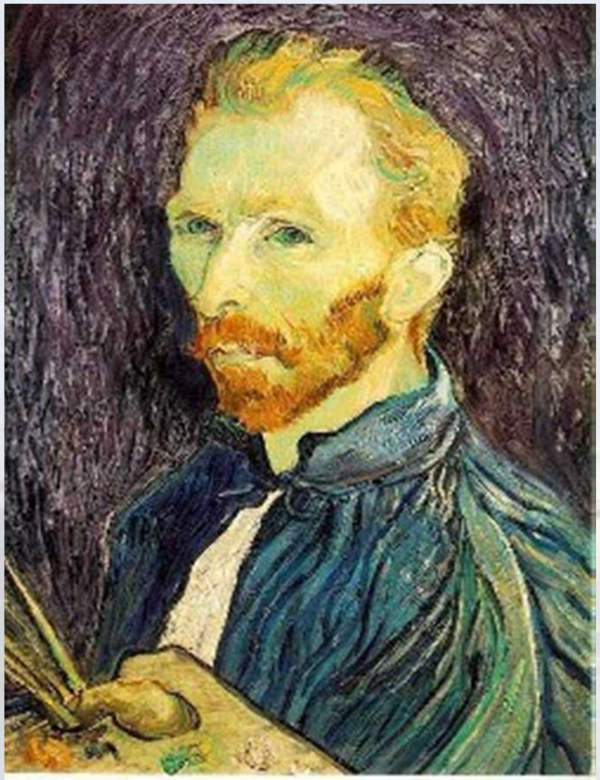
Gaussian 1/2



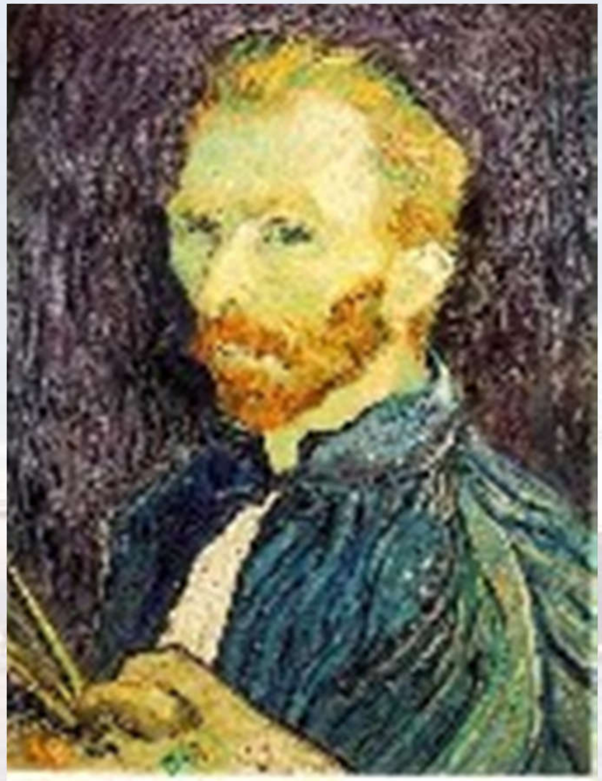
G 1/4



G 1/8



1/2



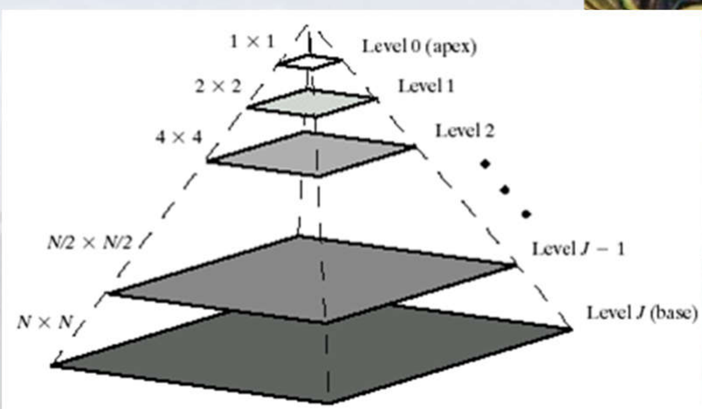
1/4 (2x zoom)

1/8 (4x zoom)

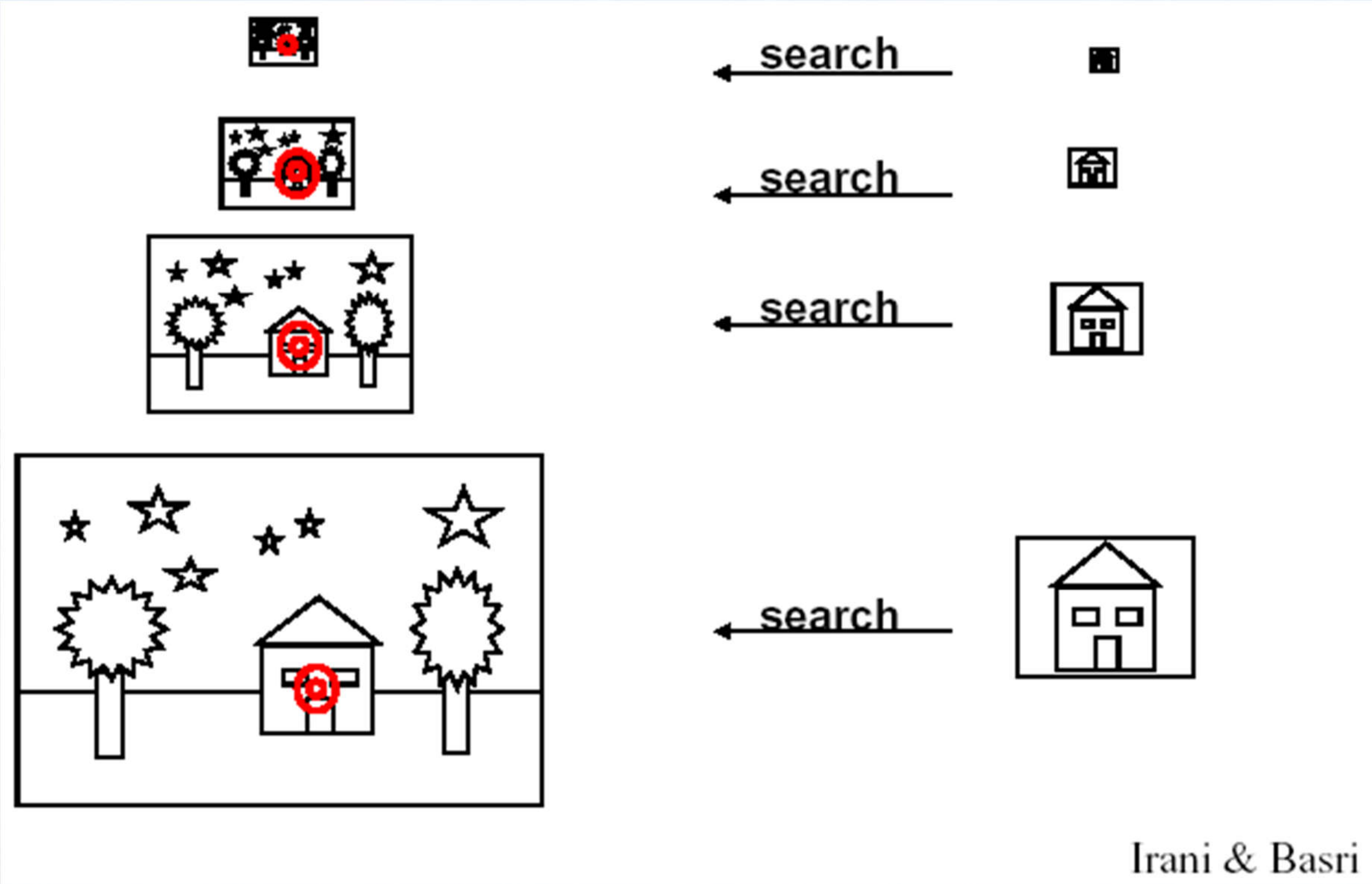
Compare with the other one…

Source: S. Seitz

Gaussian pyramid construction…



Pyramids have helped a lot to advance scale invariant object detection!



Adapted from Michael Black, Brown University

Your cameras are nowadays equipped with anti-aliasing filters

Moire patterns in real-world images. Here are comparison images by Dave Etchells of Imaging Resource using the Canon D60 (with an antialias filter) and the Sigma SD-9 (which has no antialias filter). The bands below the fur in the image at right are the kinds of artifacts that appear in images when no antialias filter is used. Sigma chose to eliminate the filter to get more sharpness, but the resulting apparent detail may or may not reflect features in the image.

Adapted from
Selim Aksoy
Bilkent U.

Modern computer games use anti-aliasing intensively