IMAGE PROCESSING
Filtering in frequency domain

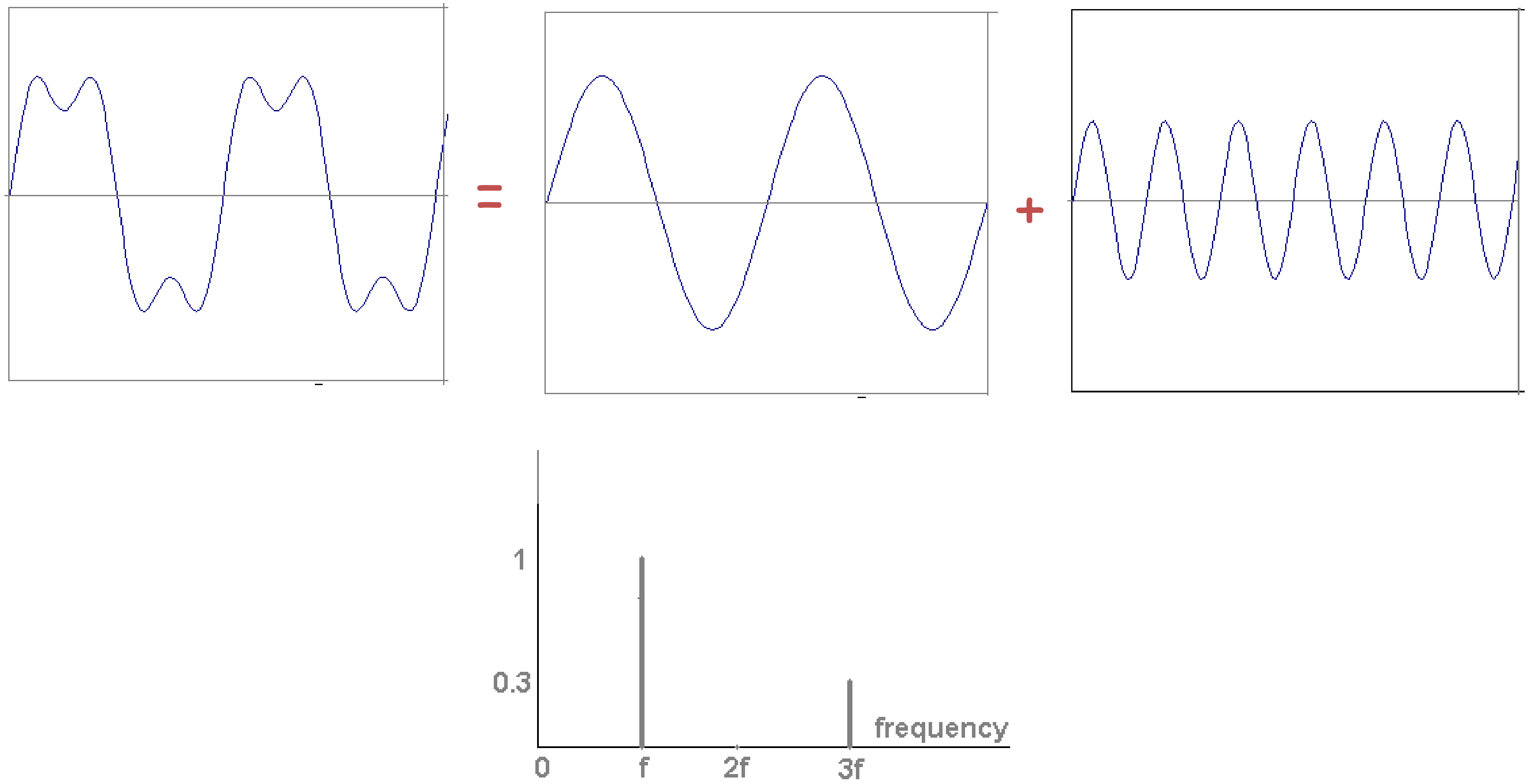
Le Thanh Ha, Ph.D
Assoc. Prof. at University of Engineering and Technology,
Vietnam National University
lthavnu@vnu.edu.vn; lthavnu@gmail.com; 0983 692 592

- Discrete Fourier transform
- Filtering in frequency domain
- Aliasing
- Hybrid image
- Common Frequency filters

THINKING IN FREQUENCY

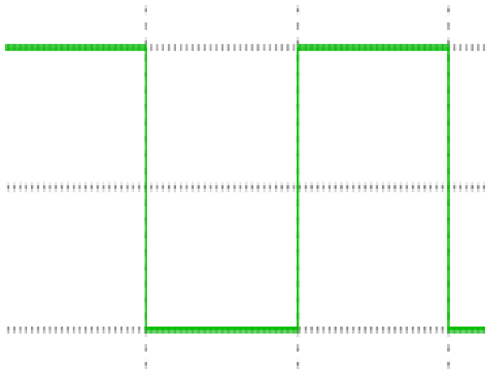
Frequency Spectra

- example : $g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi(3f)\, t)$

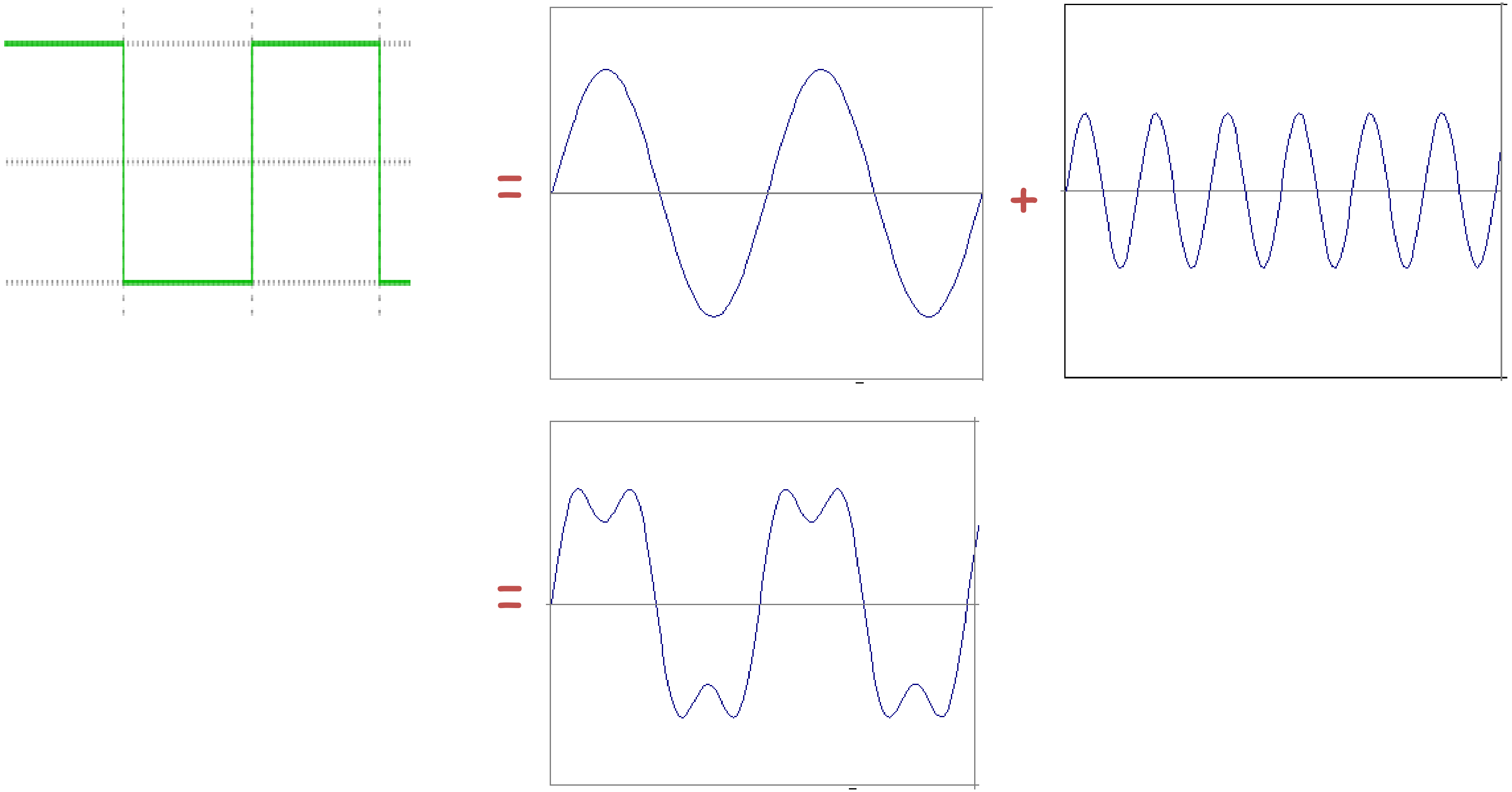


Slides: Efros

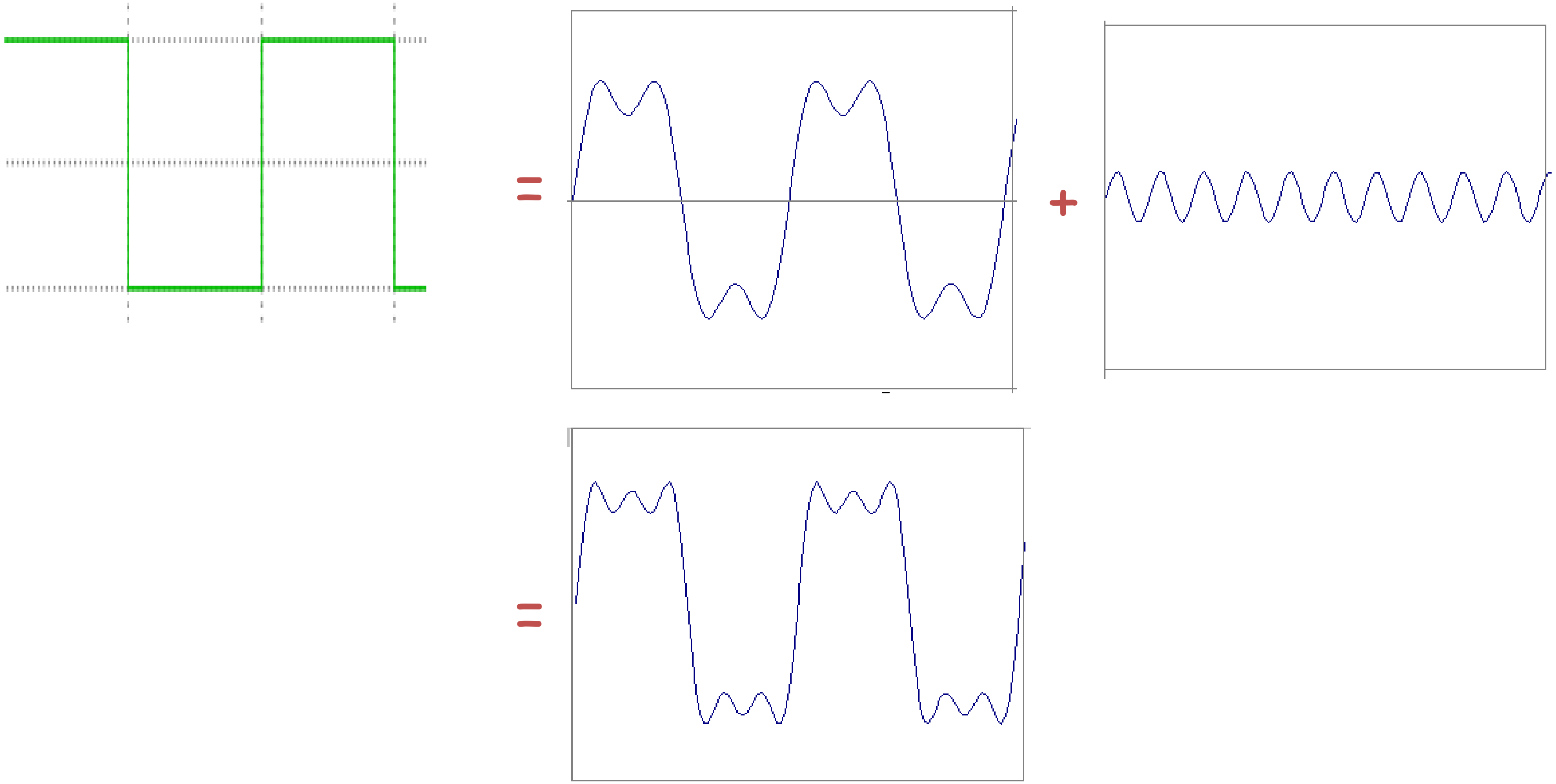
Frequency Spectra



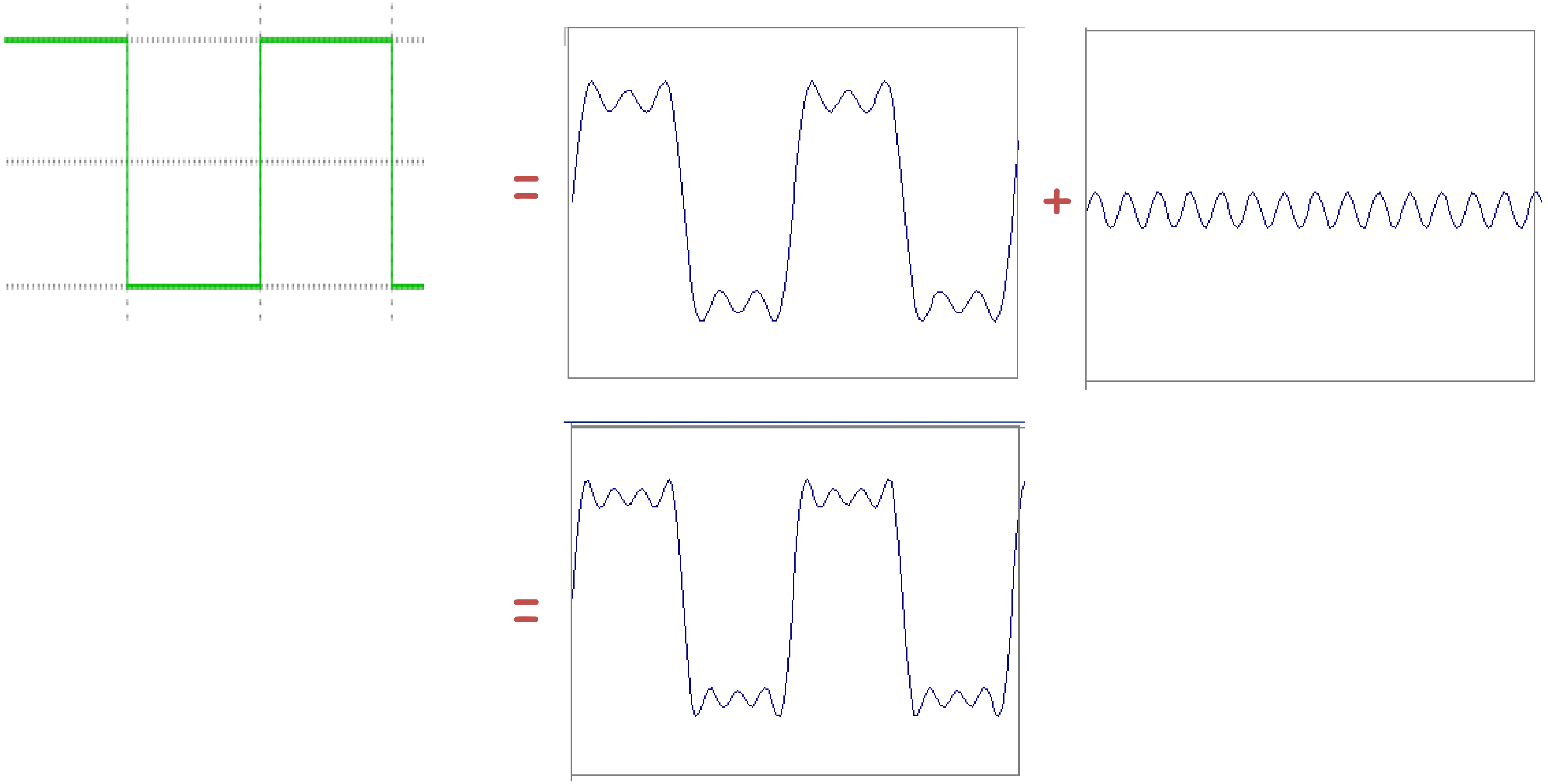
Frequency Spectra



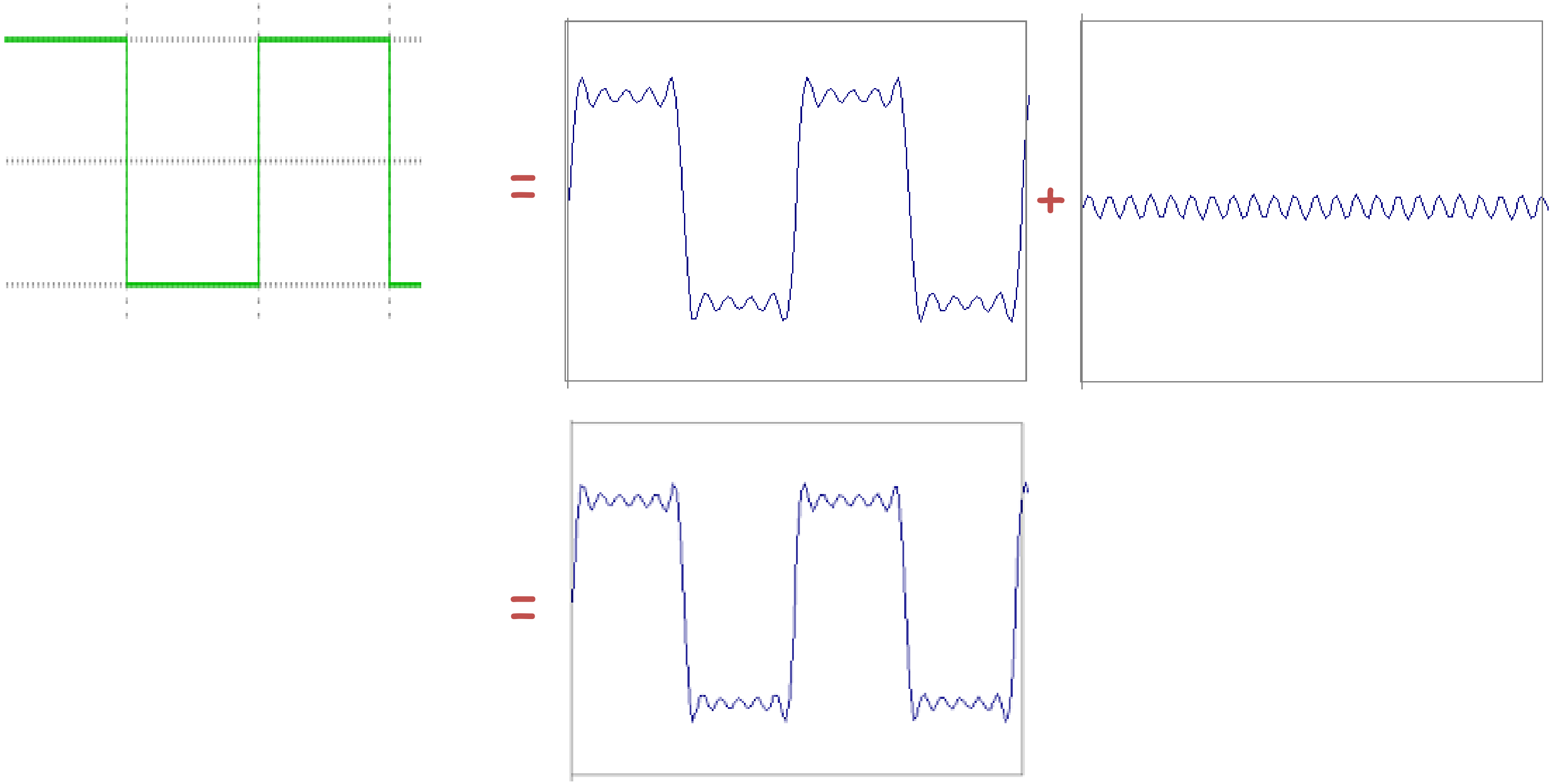
Frequency Spectra



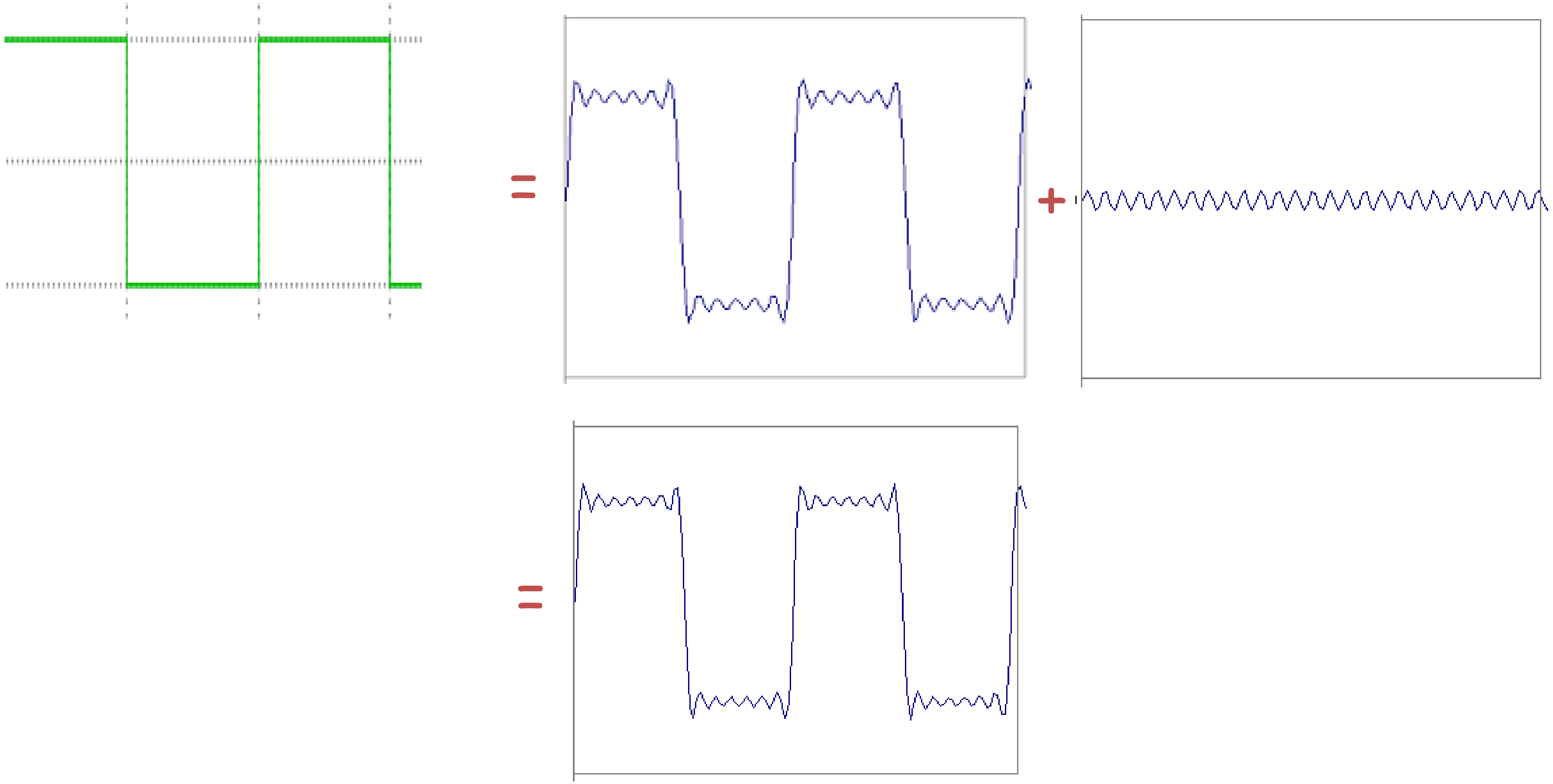
Frequency Spectra



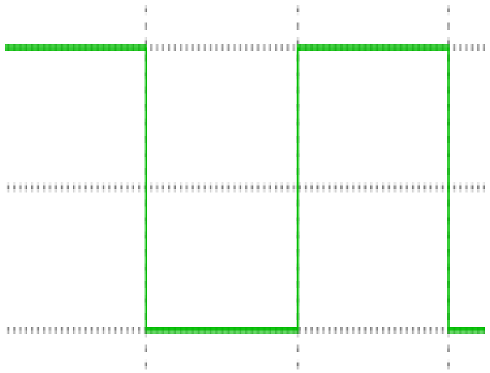
Frequency Spectra



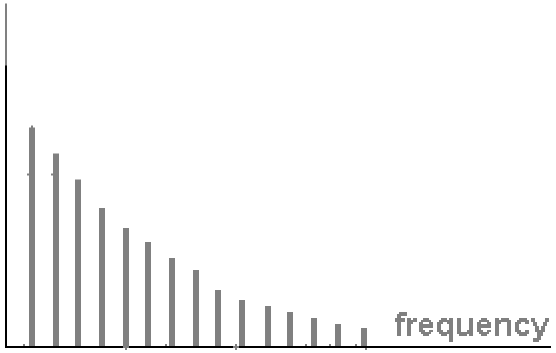
Frequency Spectra



Frequency Spectra



$$= \boxed{A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi kt)}$$



1D Discrete Fourier transform

- Complex series: X_k, x_n

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} \qquad k = 0, \ldots, N-1$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N}kn} \qquad n = 0, \ldots, N-1$$

2D Discrete Fourier transform

- Discrete domain:

$$F(w_x, w_y) = \sum_{x=1}^{M} \sum_{y=1}^{N} f(x, y) e^{-i(w_x x + w_y y)}$$

$$f(x, y) = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} F(w_x, w_y) e^{i(w_x x + w_y y)}$$

HMI
HUMAN MACHINE INTERACTION LABORATORY

2D Fourier transform

- f – image, matrix of real numbers
- F – frequency image, matrix of complex numbers.

MAGNITUDE(F) = SQRT(real(F)^2 +imag(F)^2) – Energy of frequency

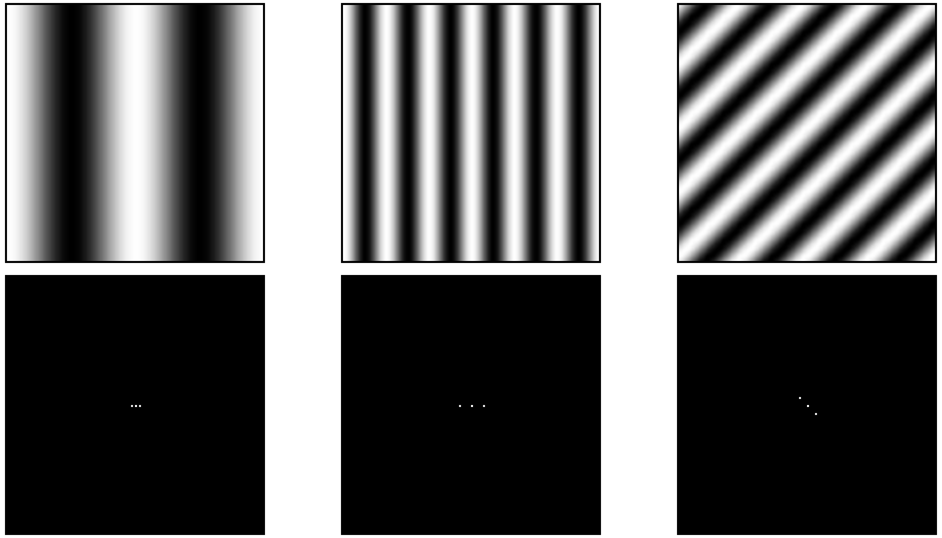
PHASE(F) = ATAN(imag(F)/real(F))

Which one is more informative, magnitude or phase?

Fourier analysis in images

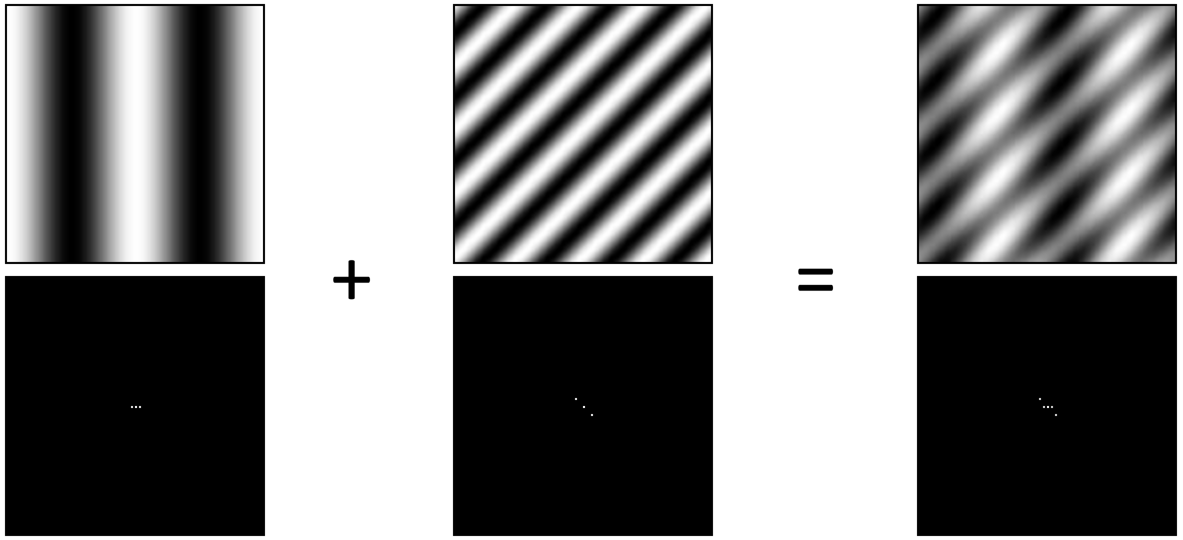
Intensity Image

Fourier Image

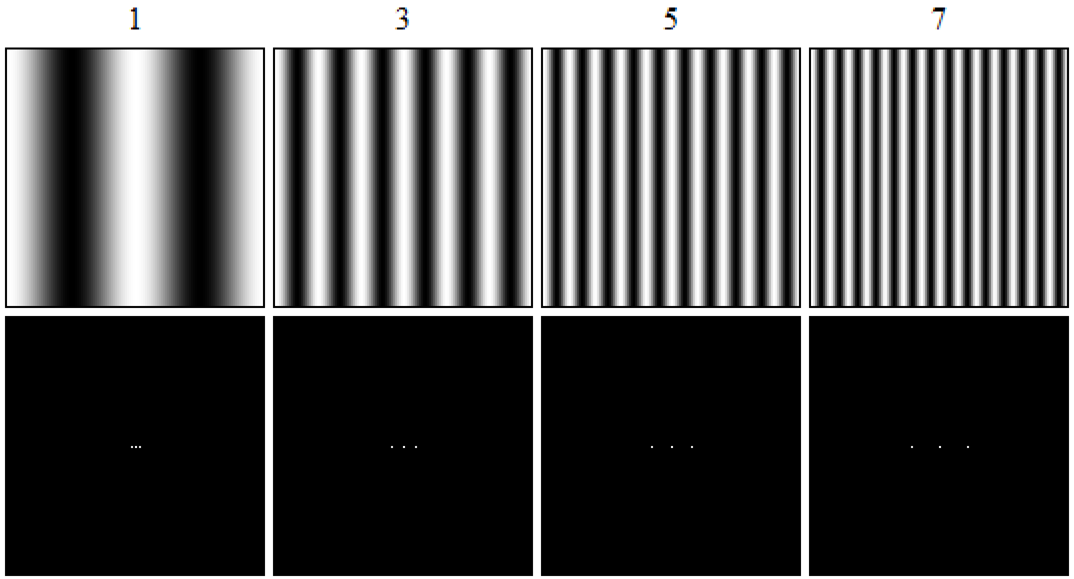


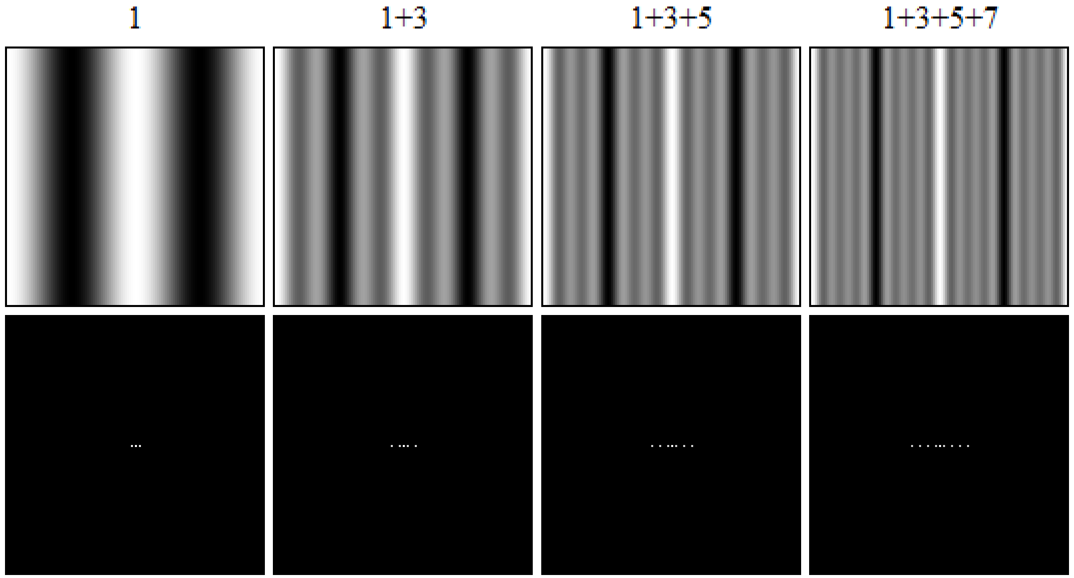
http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

Signals can be composed

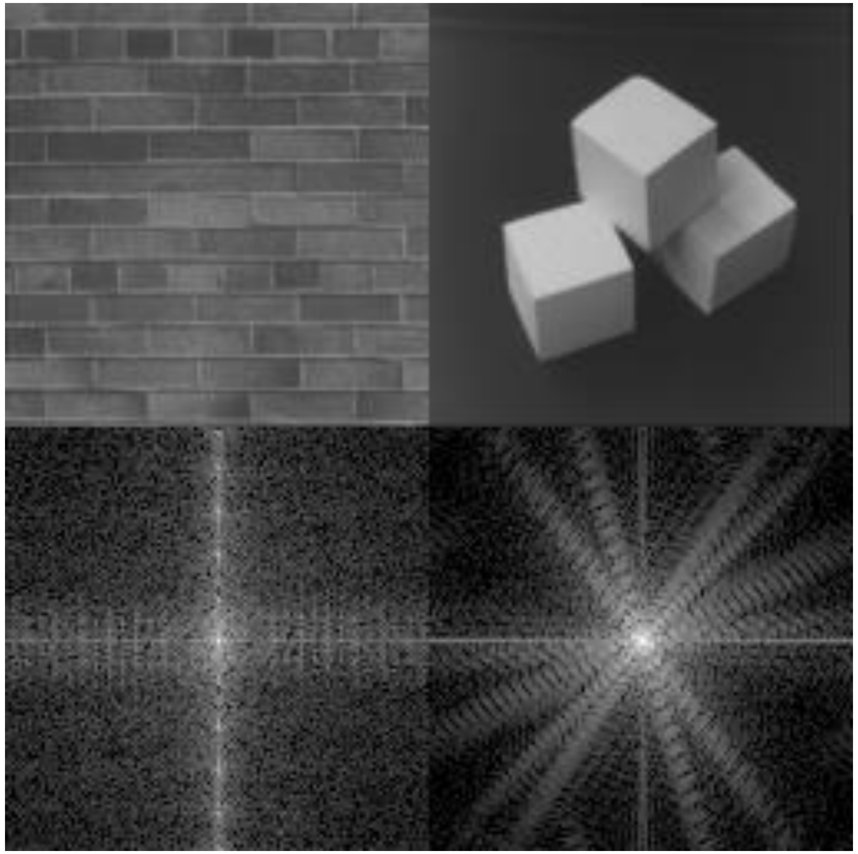


http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering
More: http://www.cs.unm.edu/~brayer/vision/fourier.html


1
3
5
7


1
1+3
1+3+5
1+3+5+7

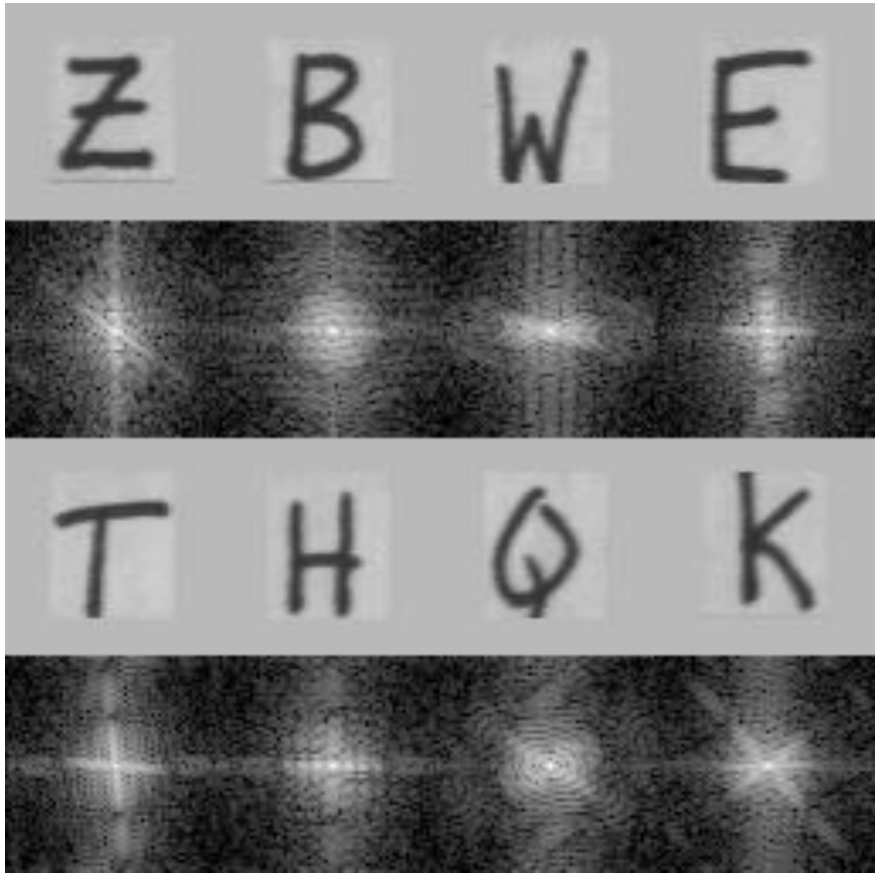
2D Fourier transform



Left: shows the vertical and horizontal frequencies as the edges of bricks.

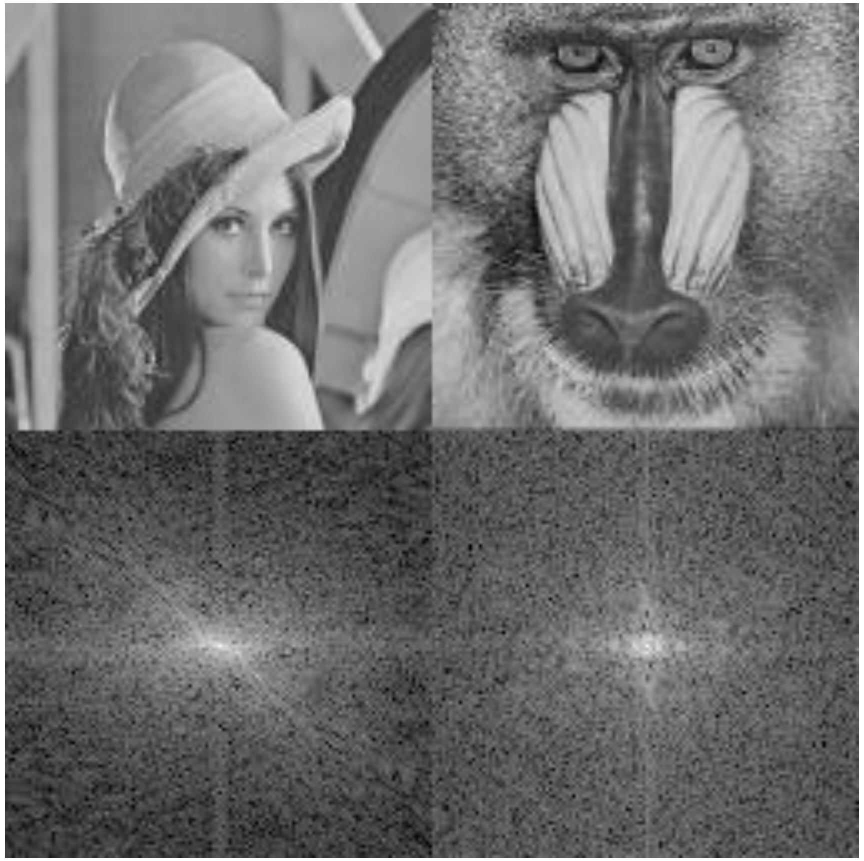
Right: shows that the edges generate high frequencies with high energy.

2D Fourier transform



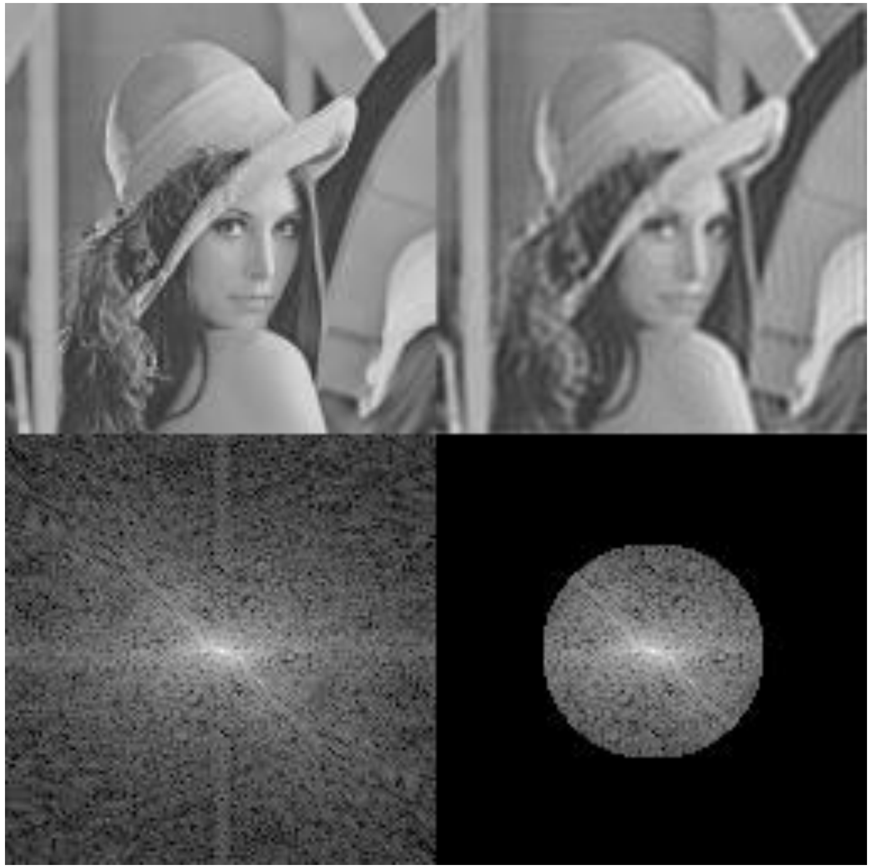
FT of each character symbol is different from others, especially at the low frequencies.

2D Fourier transform



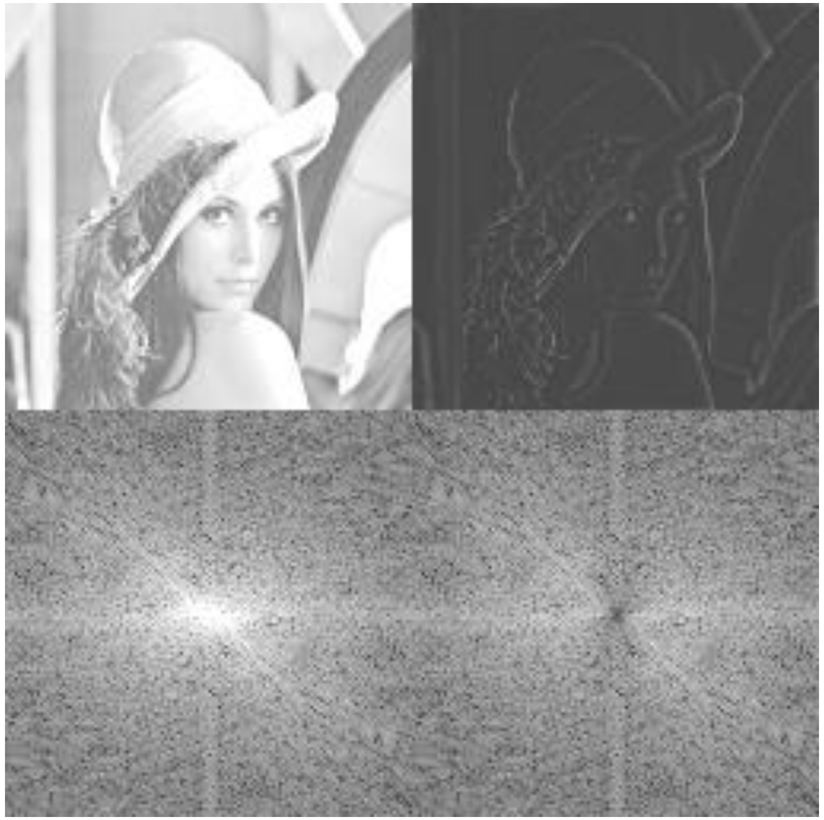
- 2 real images
- FTs show that:
  - The high frequencies of the left image is less than those of the right image.

2D Fourier transform



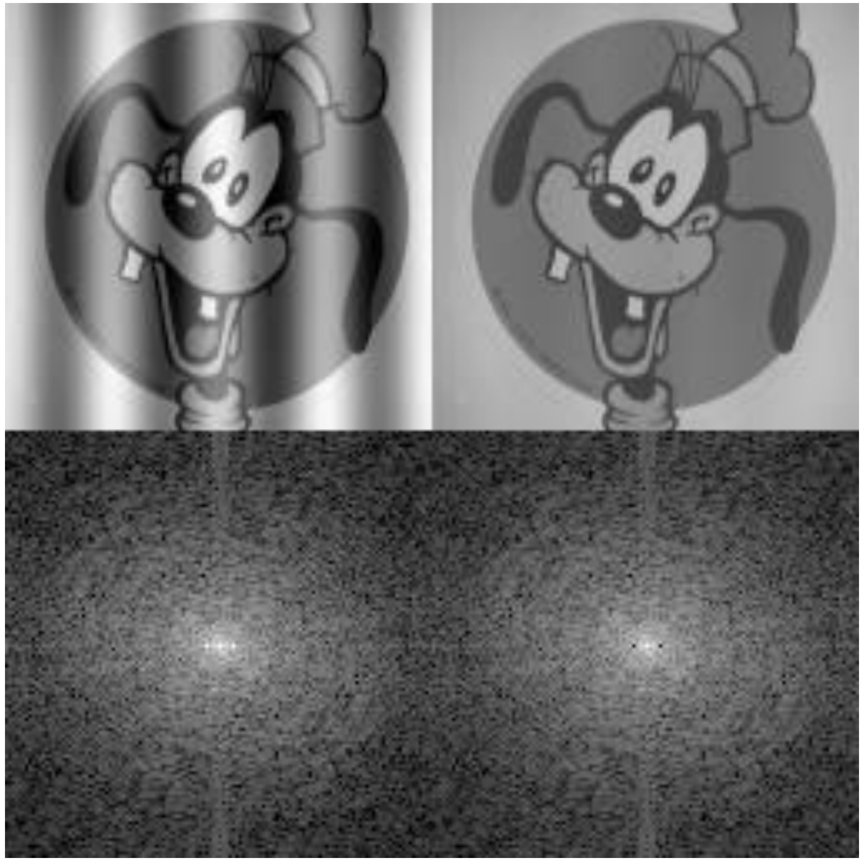
- Ideal low-pass filter:
  - Completely remove the high frequencies of the left image.
  - The right image is obtained by applying the inverse FT (IFT)

2D Fourier transform



- High pass filter
  - Remove allmost all low frequencies
  - The resulted image is too dard because high energy is condensed in low frequencies which are removed.
  - There edges left (high frequencies)

2D Fourier transform - IP



- The left image is added with horizontal wave
- There are very bright dots in frequency domain.
- The image looks good after remove the noise frequency.

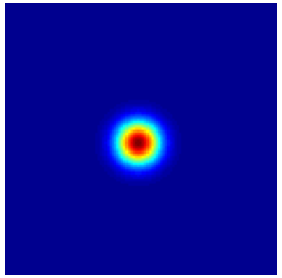
Demonstration using MATLAB

- Operator order:
  - Read an Image
  - Transform using fft2
  - Shift frequency coefs to center using fftshift
  - Create filters (ideal, gaussian, …)
  - Filter in frequency domain
  - Shift frequency coefs back using ifftshift
  - Invert transform using ifft2

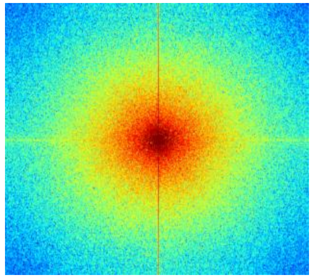
Practice question

Match the spatial domain image to the Fourier magnitude image

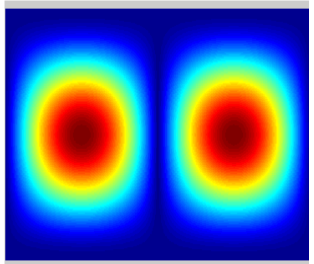
1



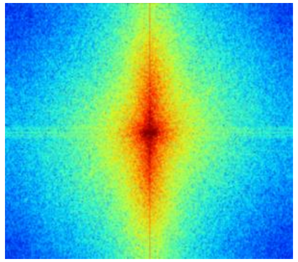
2



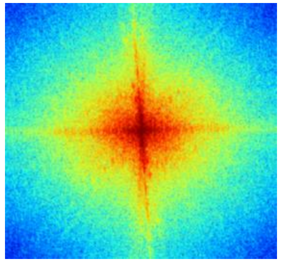
3



4



5

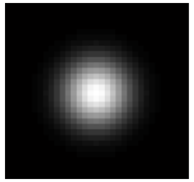


A

B

C

D



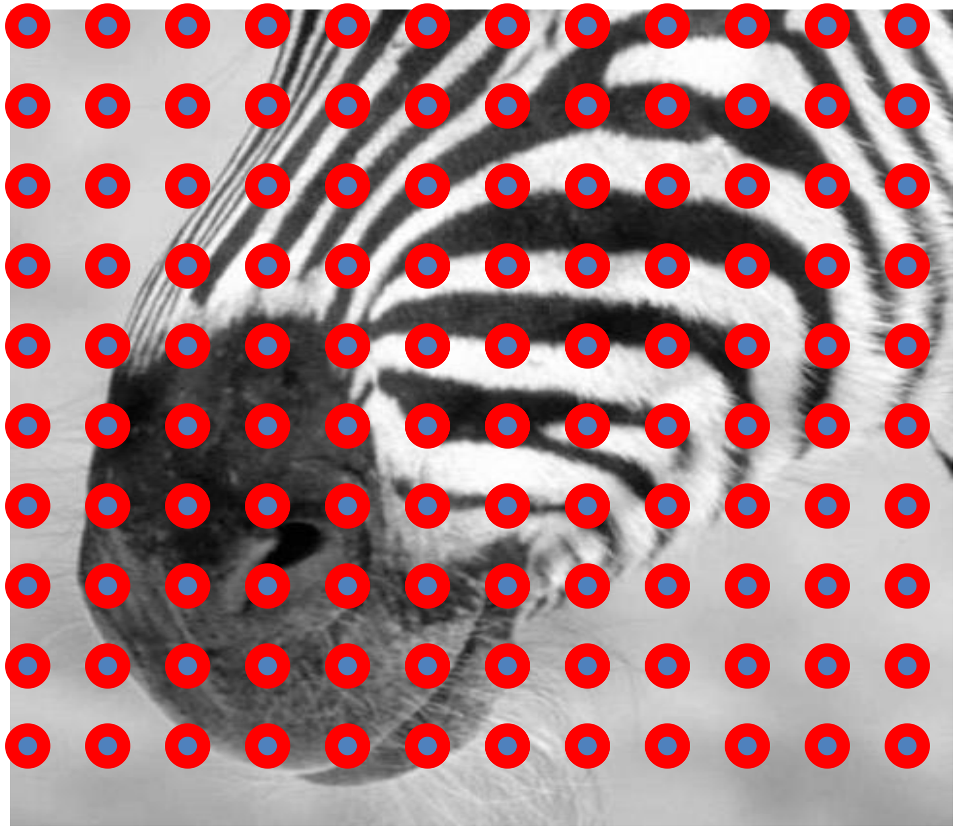
E

Sampling

Why does a lower resolution image still make sense to us? What do we lose?

Image: http://www.flickr.com/photos/igorms/136916757/

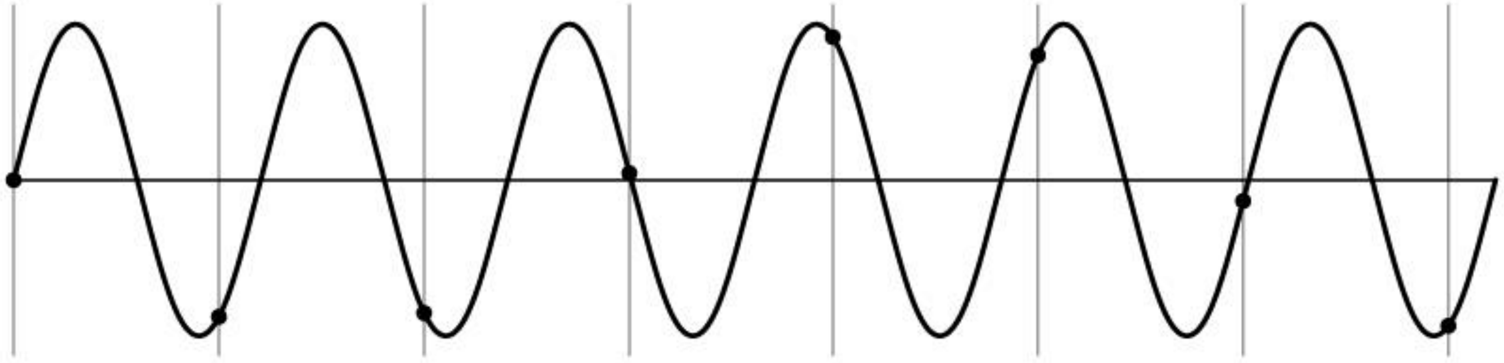
Subsampling by a factor of 2



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

Aliasing problem

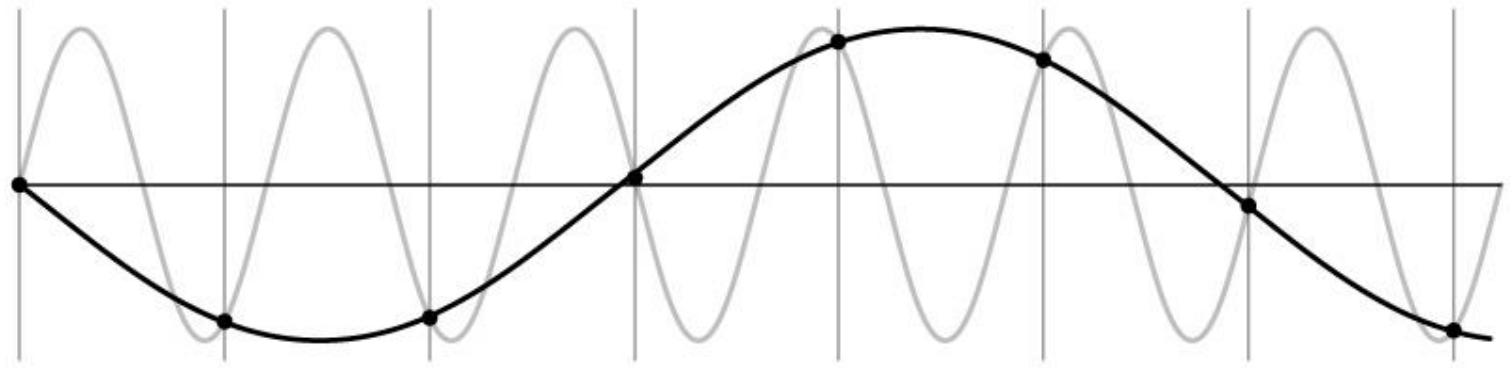
- 1D example (sinewave):



Source: S. Marschner

Aliasing problem

- 1D example (sinewave):



Source: S. Marschner

Aliasing problem

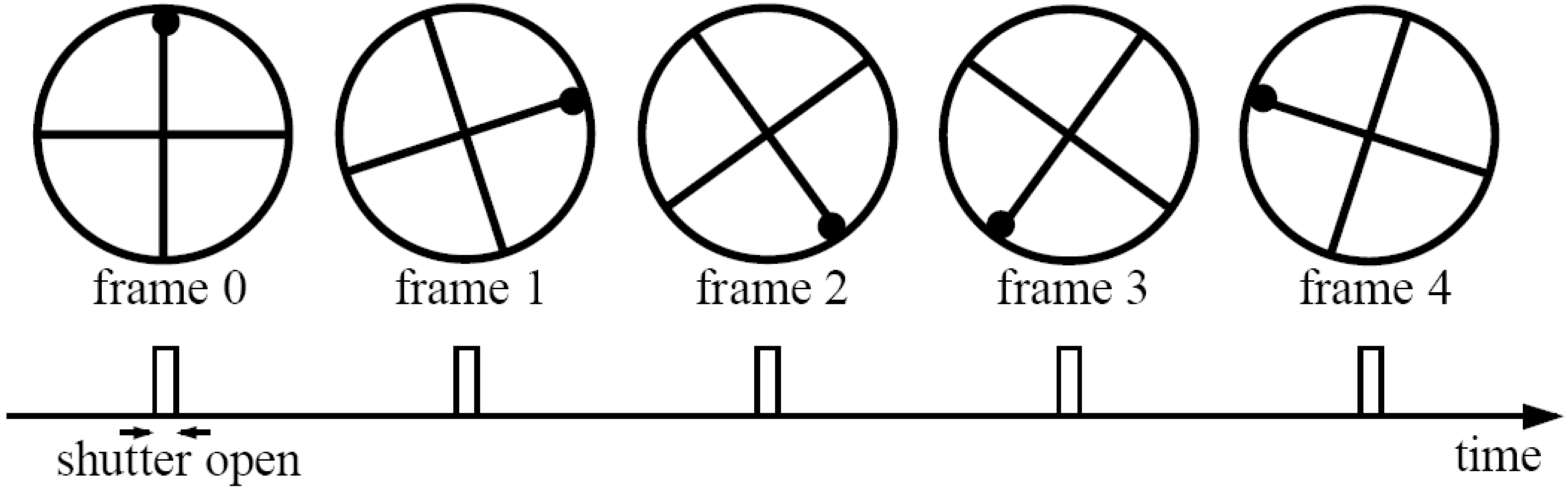
- Sub-sampling may be dangerous….
- Characteristic errors may appear:
  - “Wagon wheels rolling the wrong way in movies”
  - “Checkerboards disintegrate in ray tracing”
  - “Striped shirts look funny on color television”

Source: D. Forsyth

Aliasing in video

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what’s happening.

If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



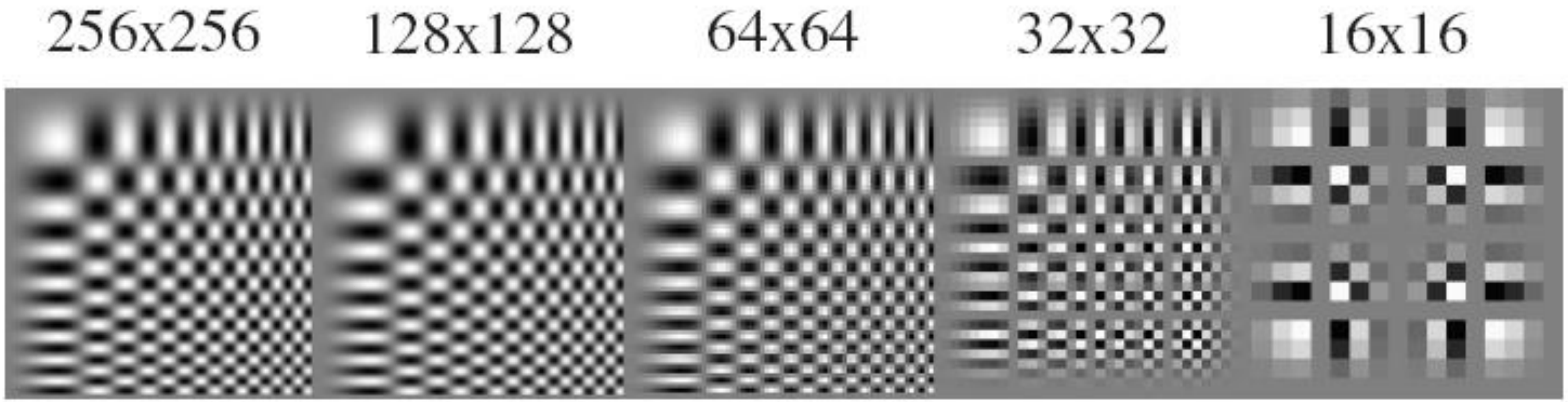
Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

Slide by Steve Seitz

Aliasing in graphics

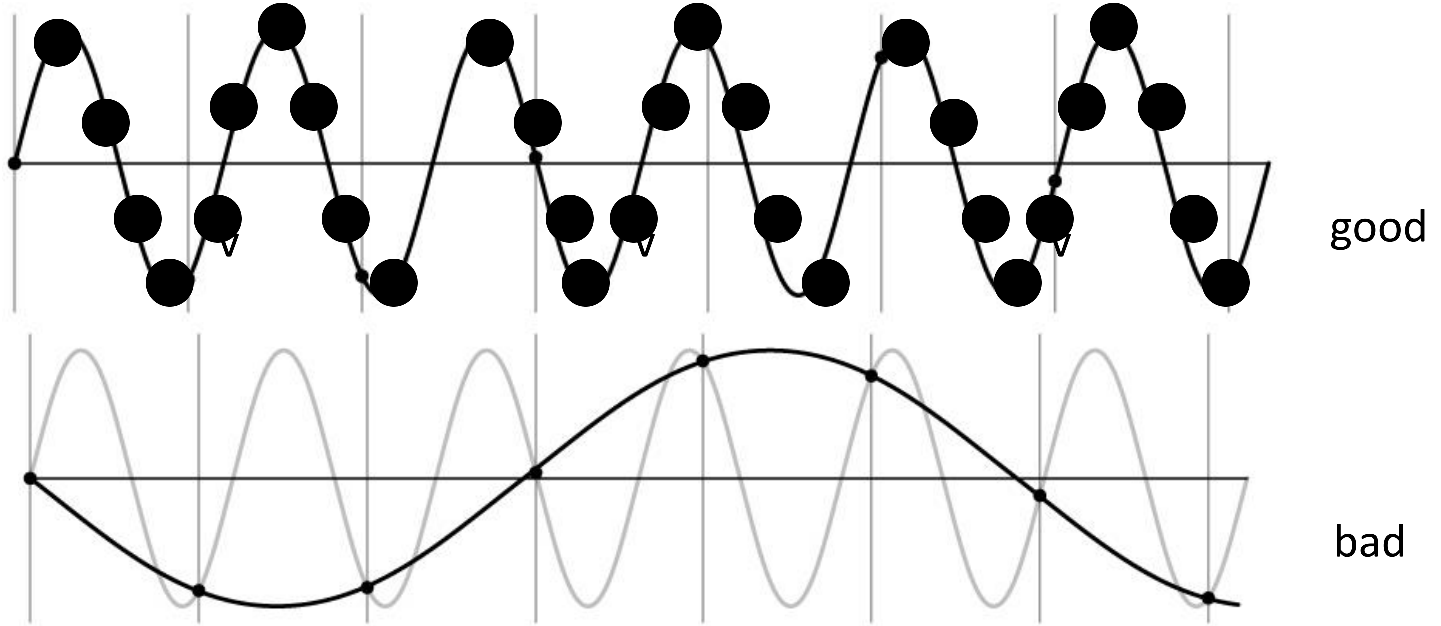
Source: A. Efros

Sampling and aliasing



Nyquist-Shannon Sampling Theorem

- When sampling a signal at discrete intervals, the sampling frequency must be $\geq 2 \times f_{max}$
- f_{max} = max frequency of the input signal
- This will allows to reconstruct the original perfectly from the sampled version



Anti-aliasing

Solutions:

- Sample more often

- Get rid of all frequencies that are greater than half the new sampling frequency
  - Will lose information
  - But it’s better than aliasing
  - Apply a smoothing filter

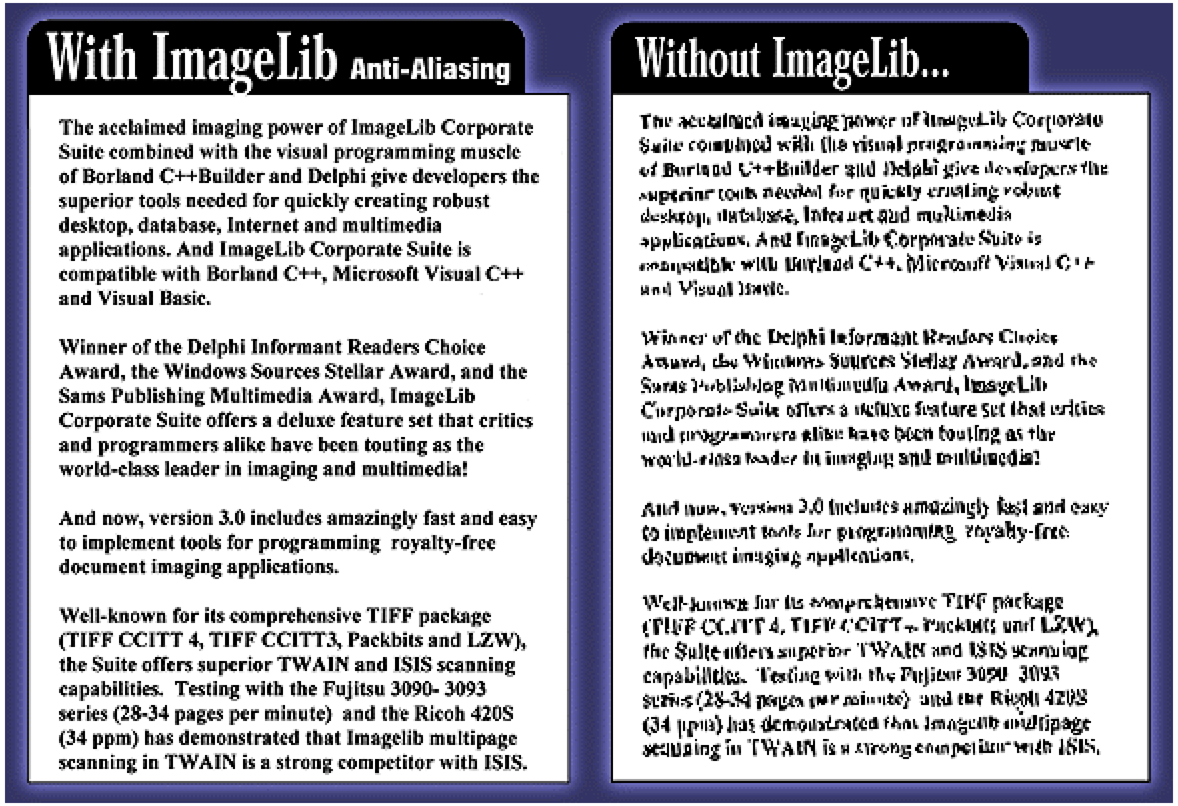
Algorithm for downsampling by factor of 2

1. Start with image(h, w)
2. Apply low-pass filter

   im_blur = imfilter(image, fspecial('gaussian', 7, 1))
3. Sample every other pixel

   im_small = im_blur(1:2:end, 1:2:end);

With and without anti-aliasing

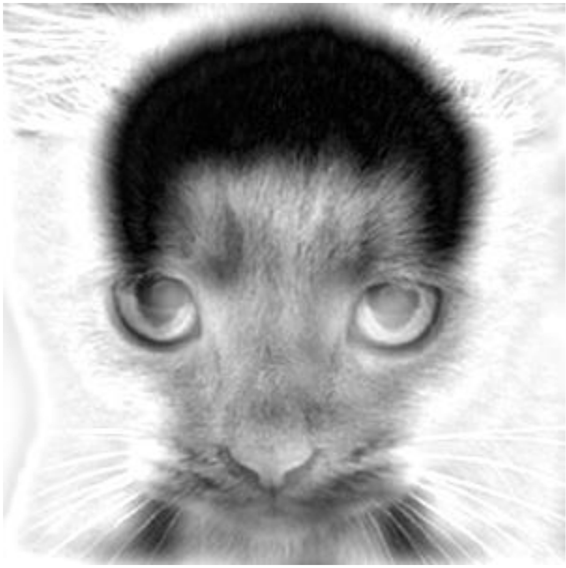


Why does a lower resolution image still make sense to us? What do we lose?

Image: http://www.flickr.com/photos/igorms/136916757/

What do you see?

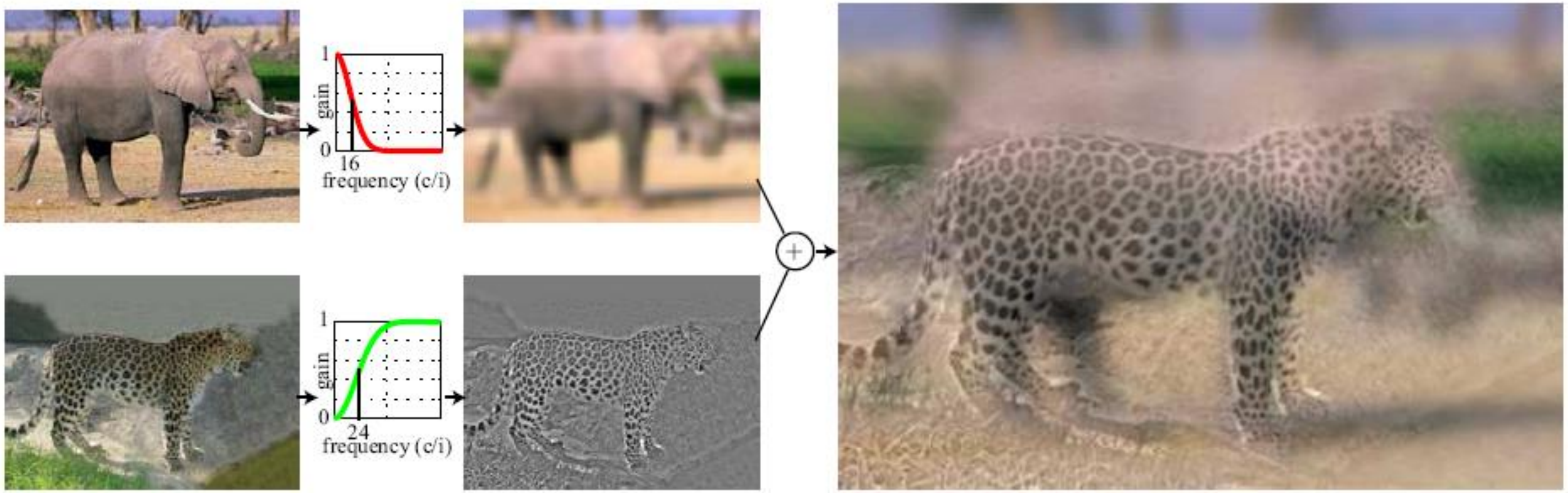
Why do we get different, distance-dependent interpretations of hybrid images?



Come closer

Go further

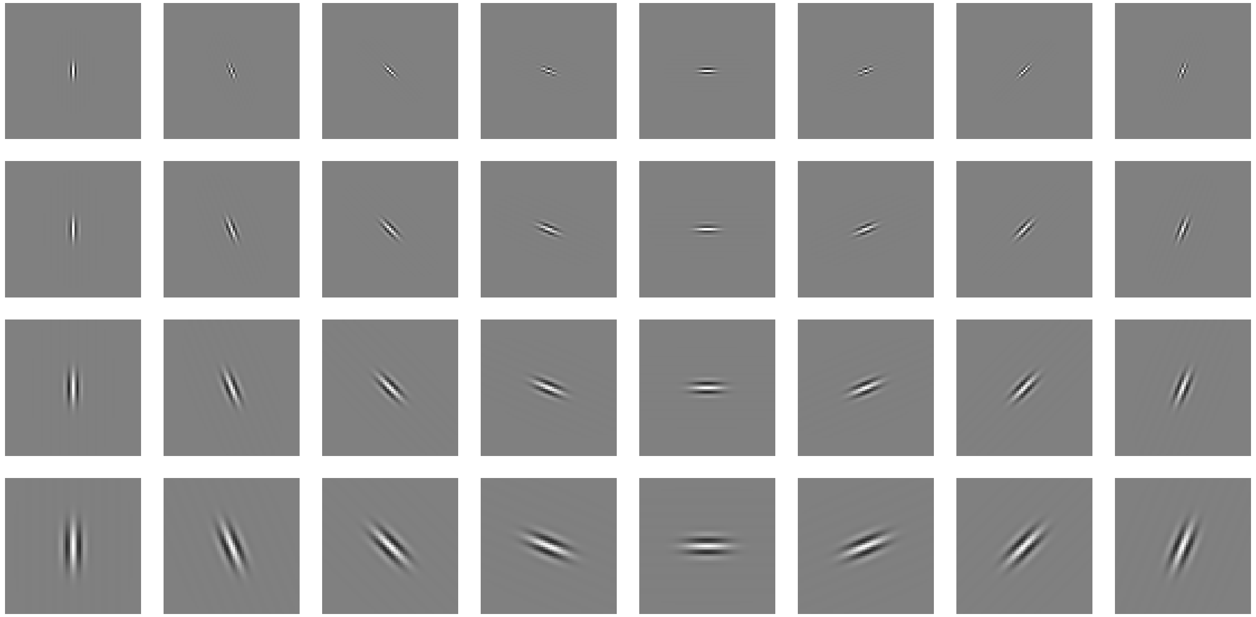
Hybrid Images



- A. Oliva, A. Torralba, P.G. Schyns, [“Hybrid Images,”]() SIGGRAPH 2006

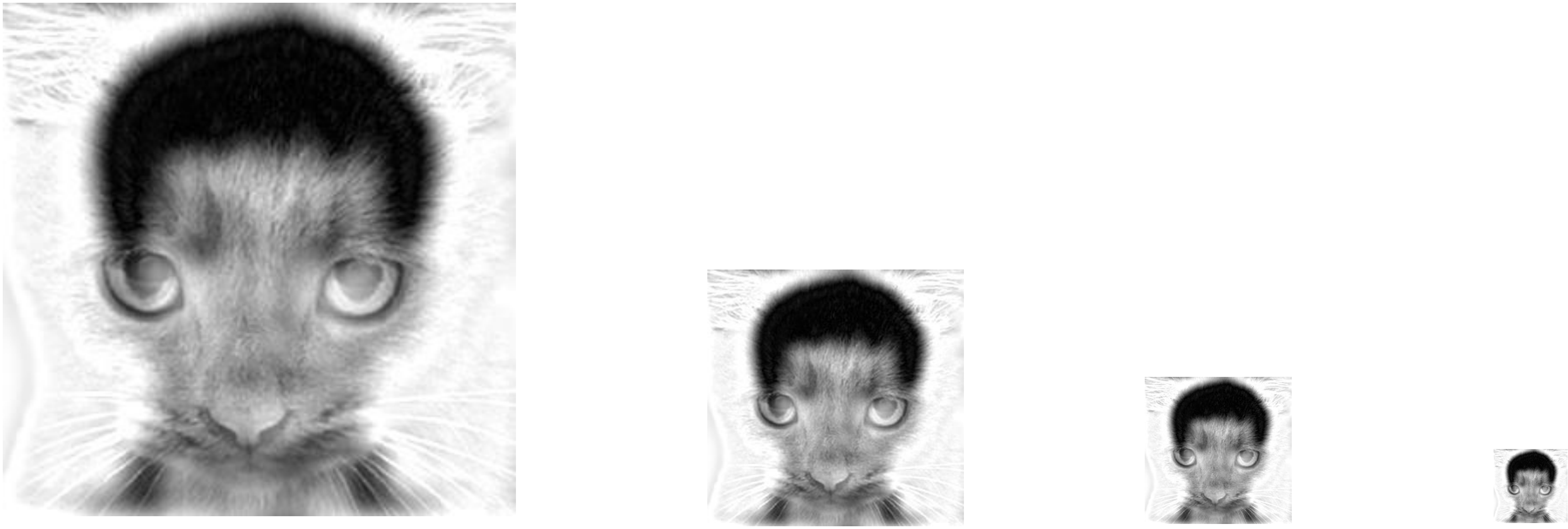
Clues from Human Perception

- Early processing in humans filters for various orientations and scales of frequency
- Perceptual cues in the mid-high frequencies dominate perception
- When we see an image from far away, we are effectively subsampling it

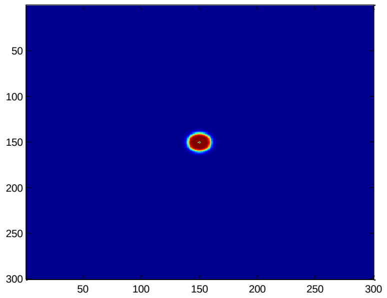


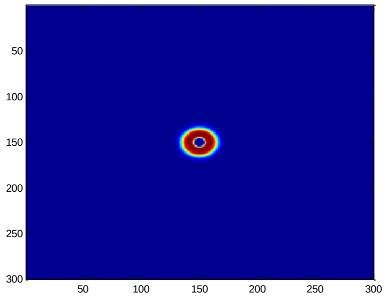
Early Visual Processing: Multi-scale edge and blob filters

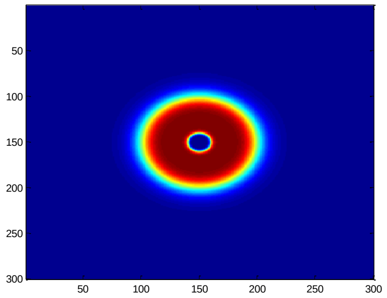
Scaling



Subband filter







Ideal Lowpass filter

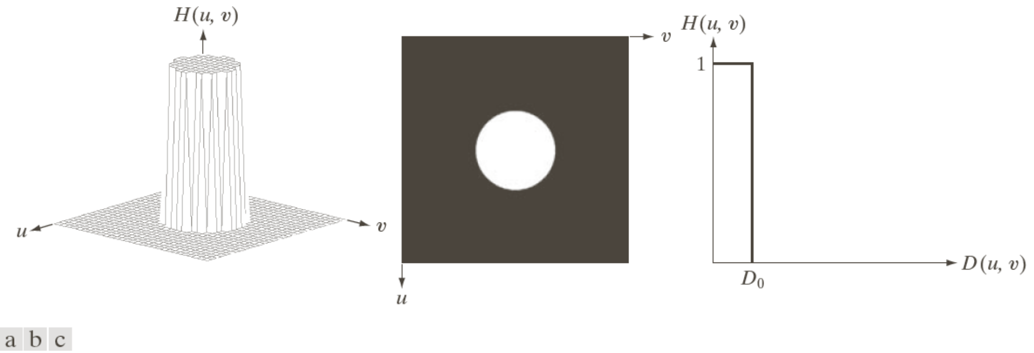


FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

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

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

- Cut off high frequencies specified by a distance d_0.
  - Cannot be realized by electronic component $\rightarrow$ not practical
  - Causes ringing effect $\rightarrow$ How to remove

Ideal Lowpass filter

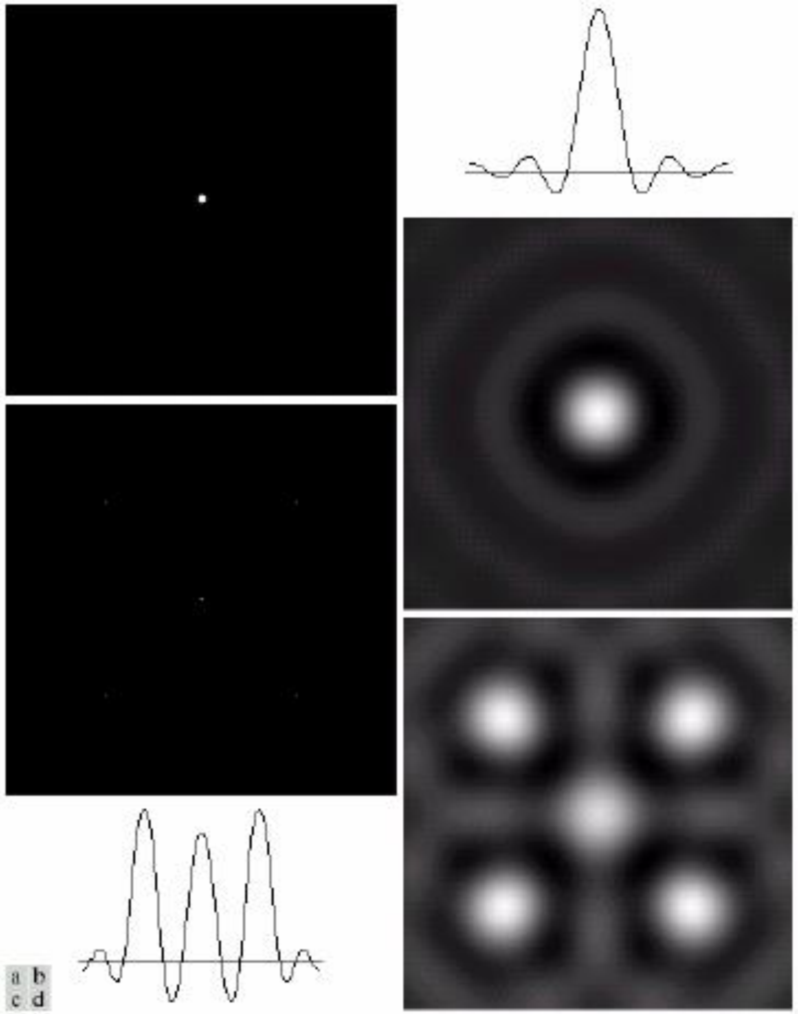


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

Ideal Lowpass filter

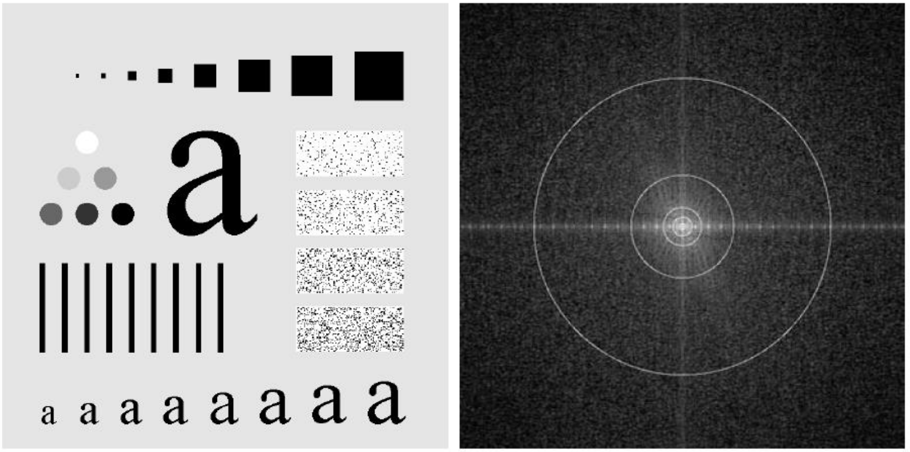


FIGURE 4.41 (a) Test pattern of size 688 × 688 pixels, and (b) its Fourier spectrum. The spectrum is double the image size due to padding but is shown in half size so that it fits in the page. The superimposed circles have radii equal to 10, 30, 60, 160, and 460 with respect to the full-size spectrum image. These radii enclose 87.0, 93.1, 95.7, 97.8, and 99.2% of the padded image power, respectively.

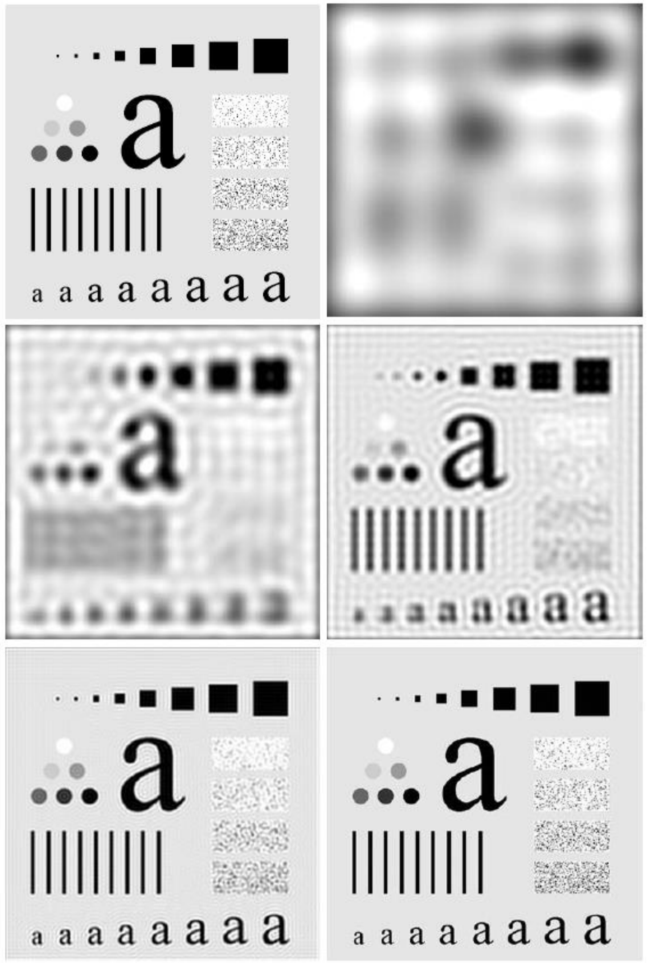
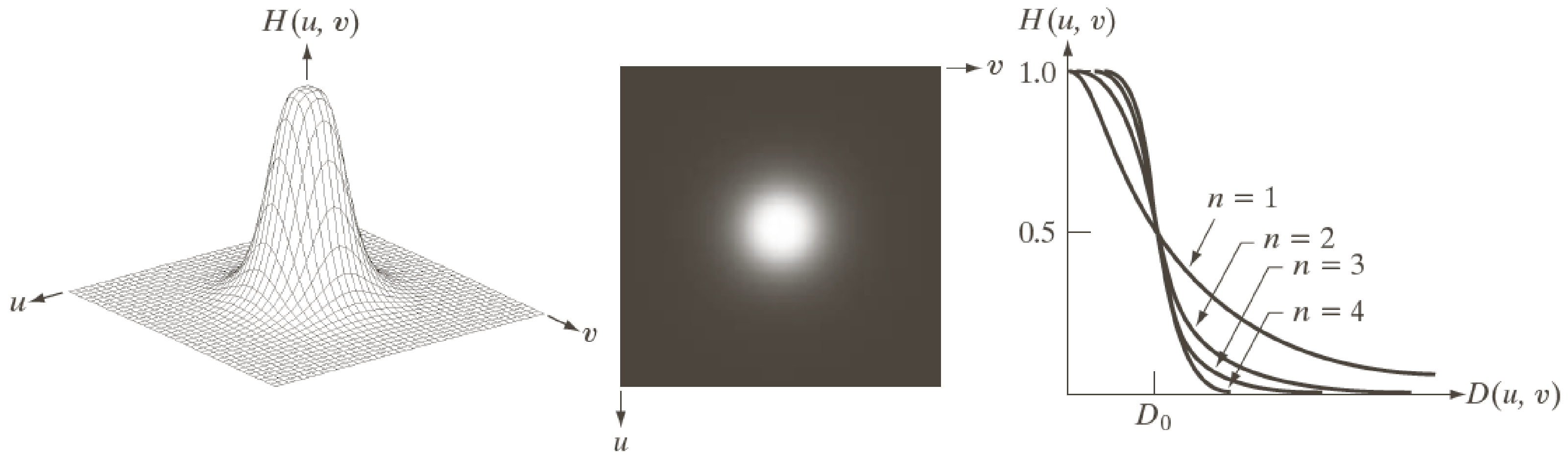


FIGURE 4.42 (a) Original image. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.

Butterworth Filter

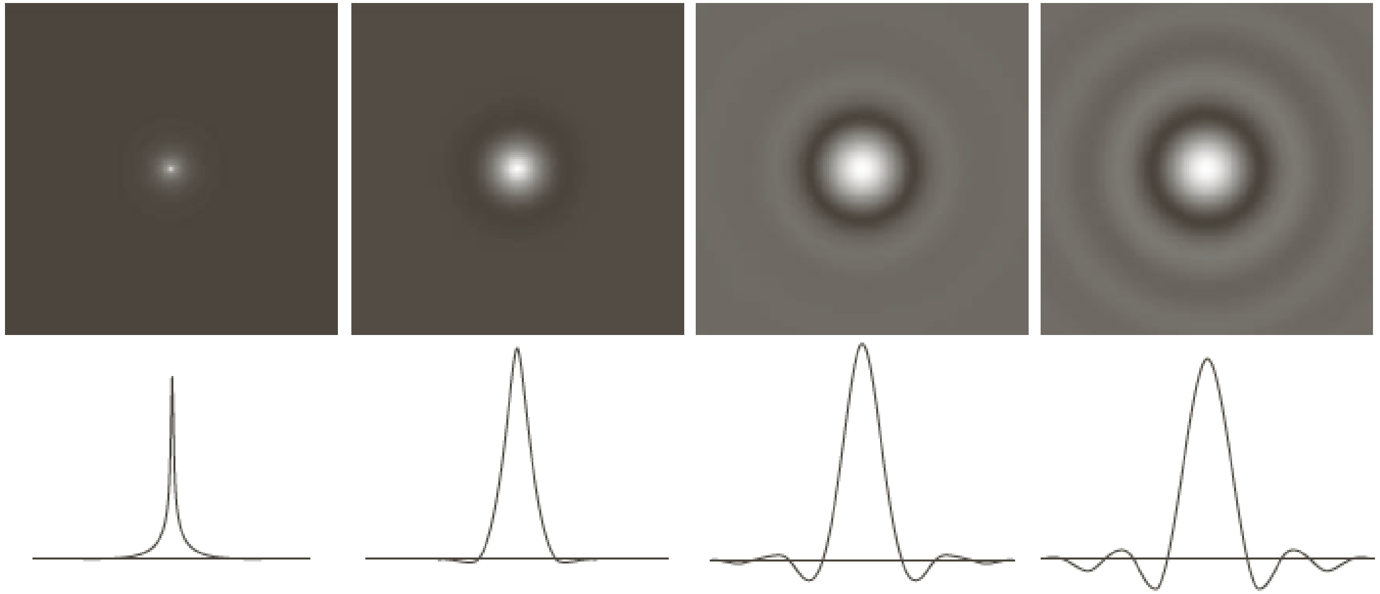


a b c

FIGURE 4.44 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

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

Butterworth Filter



a b c d

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 × 1000 and the cutoff frequency is 5). Observe how ringing increases as a function of filter order.

Butterworth Filter

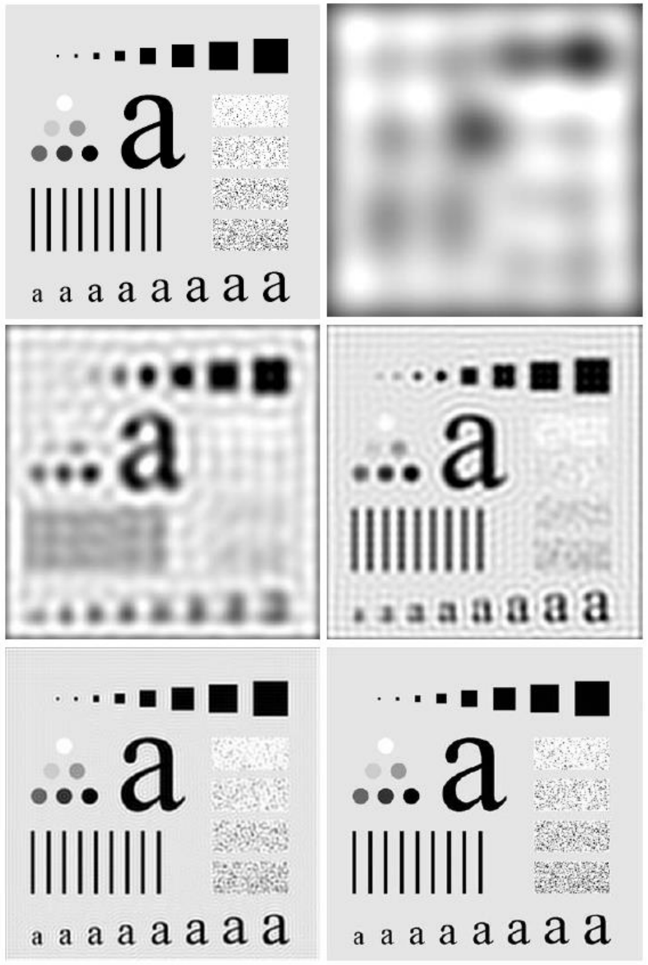


FIGURE 4.42 (a) Original image. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.

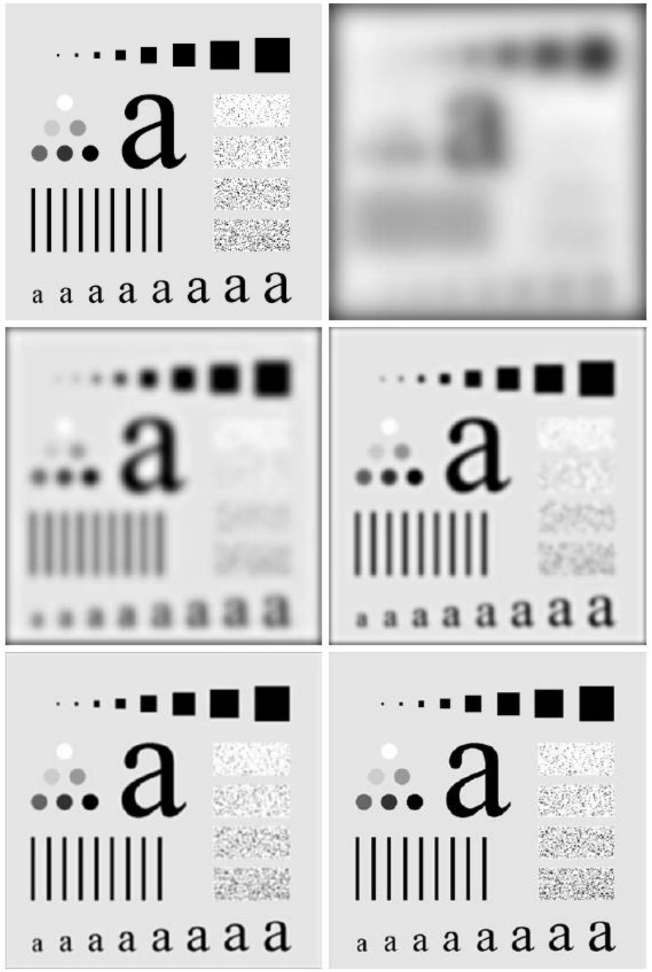
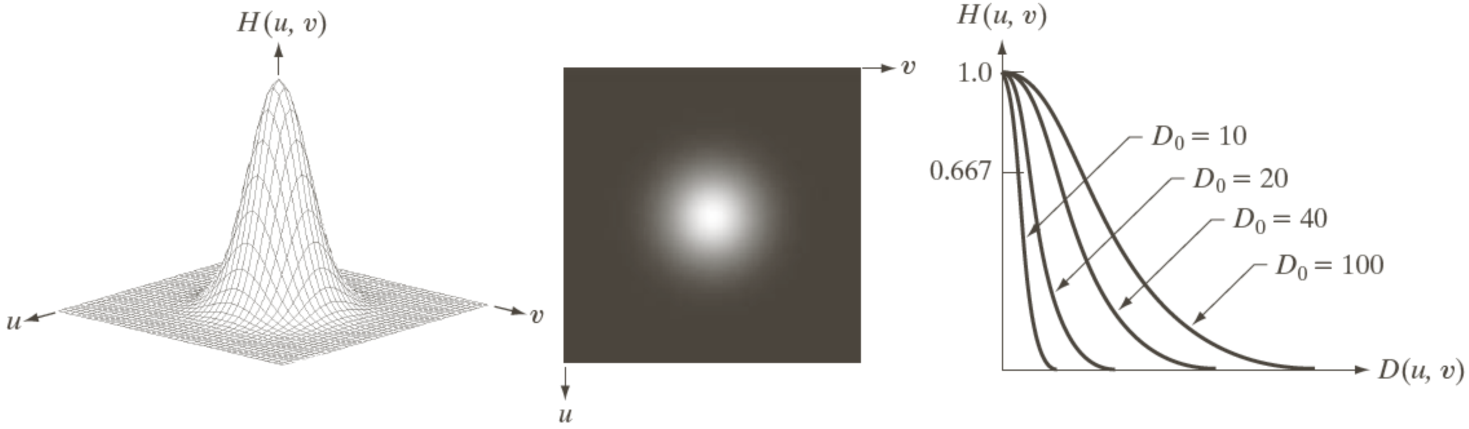


FIGURE 4.45 (a) Original image. (b)–(f) Results of filtering using BLPFs of order 2, with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Fig. 4.42.

Gaussian Filter

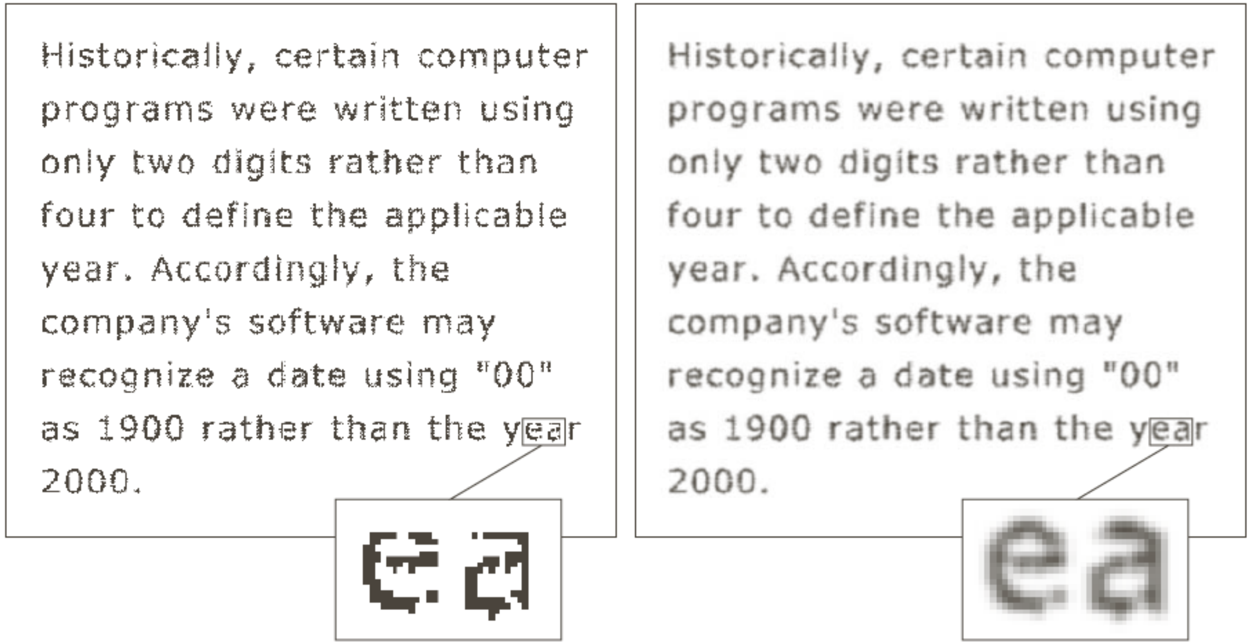


a b c

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.

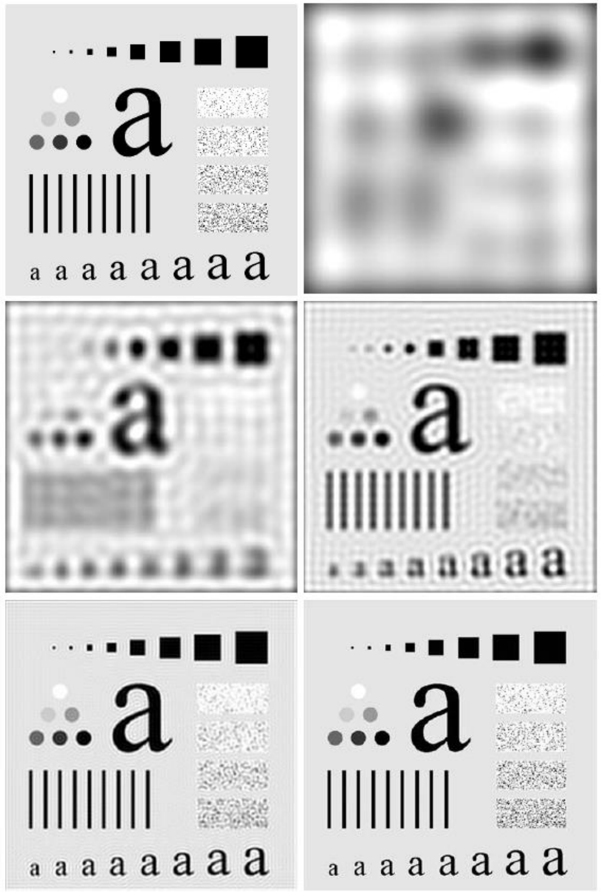
$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

Gaussian Filter



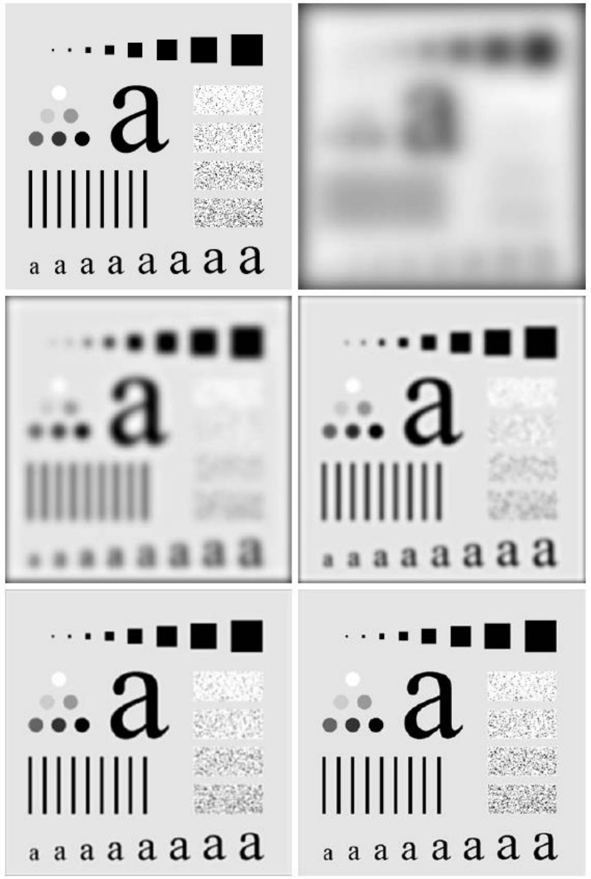
a b

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).



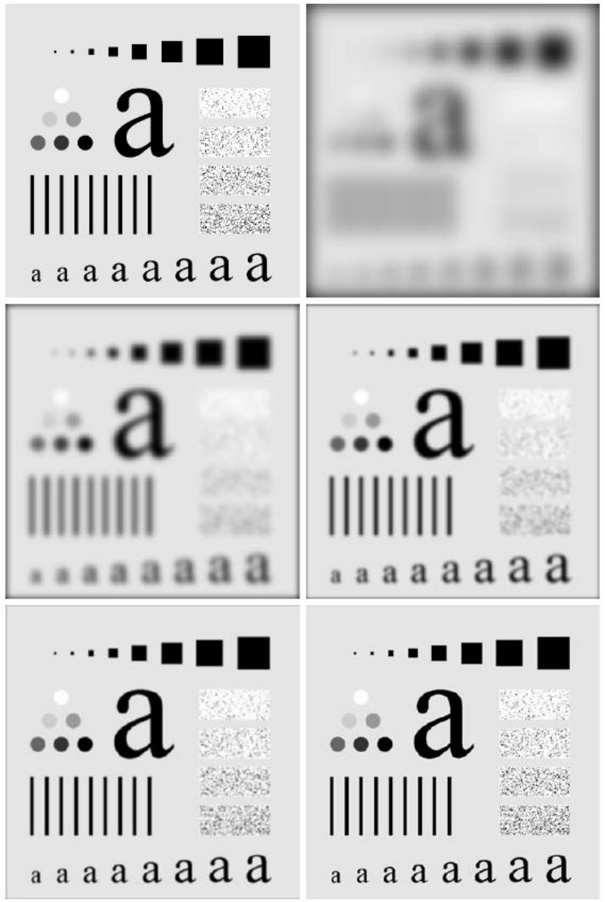
a b
c d
e f

FIGURE 4.42 (a) Original image. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.



a b
c d
e f

FIGURE 4.45 (a) Original image. (b)–(f) Results of filtering using BLPFs of order 2, with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Fig. 4.42.



a b
c d
e f

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.

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

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D(u,v)/D_0]^{2n}}$	$H(u,v) = e^{-D^2(u,v)/2D_0^2}$

Demonstration using MATLAB

- Operator order:
  - Read an Image
  - Transform using fft2
  - Shift frequency coefs to center using fftshift
  - Create filters (ideal, butterworth, …)
  - Filter in frequency domain
  - Shift frequency coefs back using ifftshift
  - Invert transform using ifft2