

UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

## Filtering in the frequency domain

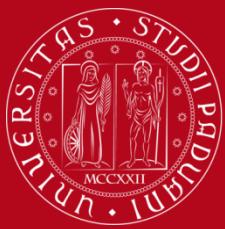
Stefano Ghidoni



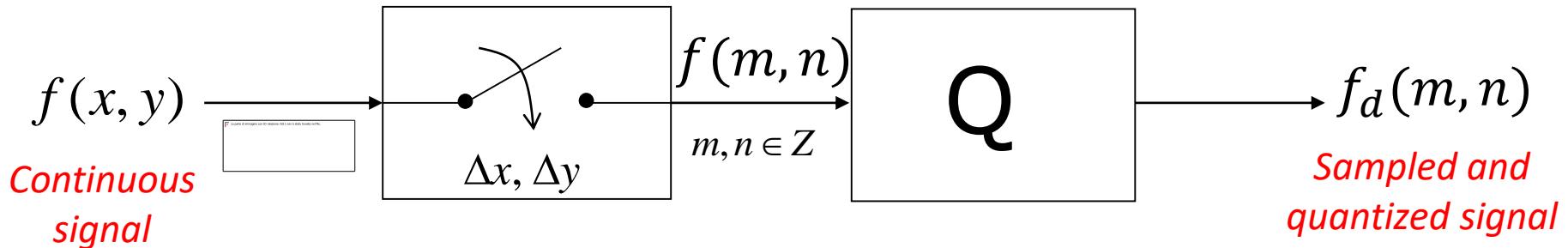
DIPARTIMENTO  
DI INGEGNERIA  
DELL'INFORMAZIONE

INTELLIGENT AUTONOMOUS SYSTEMS LAB





- Image sampling and quantization
- Phase and spectrum
- Image resampling
- Filtering in the frequency domain
- More complex filtering
- Denoising: an example



- Sampling:

$$f(m, n) \triangleq f(m\Delta x, n\Delta y)$$

- $\Delta x$  and  $\Delta y$  sampling period along  $x$  and  $y$  axis

- Quantization:

$$f_d(m, n) = Q[f(m, n)]$$



	<i>t</i>	<i>f</i>	<i>Expression</i>
FT	<i>Continuous aperiodic</i>	<i>Continuous aperiodic</i>	$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt$
Fourier series	<i>Continuous periodic</i>	<i>Discrete aperiodic</i>	$F_n = \frac{1}{2\pi} \int_{-\pi}^{+\pi} f(x)e^{-jnx} dx$
DTFT	<i>Discrete aperiodic</i>	<i>Continuous periodic</i>	$X_T(f) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j2\pi fn} \quad (\text{with } T = 1)$
DFT	<i>Discrete periodic</i>	<i>Discrete periodic</i>	$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi \frac{kn}{N}} \quad (\text{with } n_0 = 0 \text{ and } T = 1)$



- Continuous 2D Fourier transform of a signal  $f(x, y)$

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

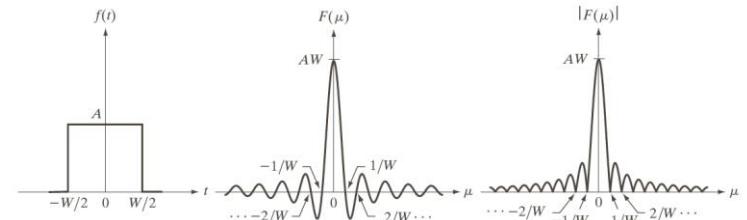
- Discrete 2D Fourier transform (DFT)

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

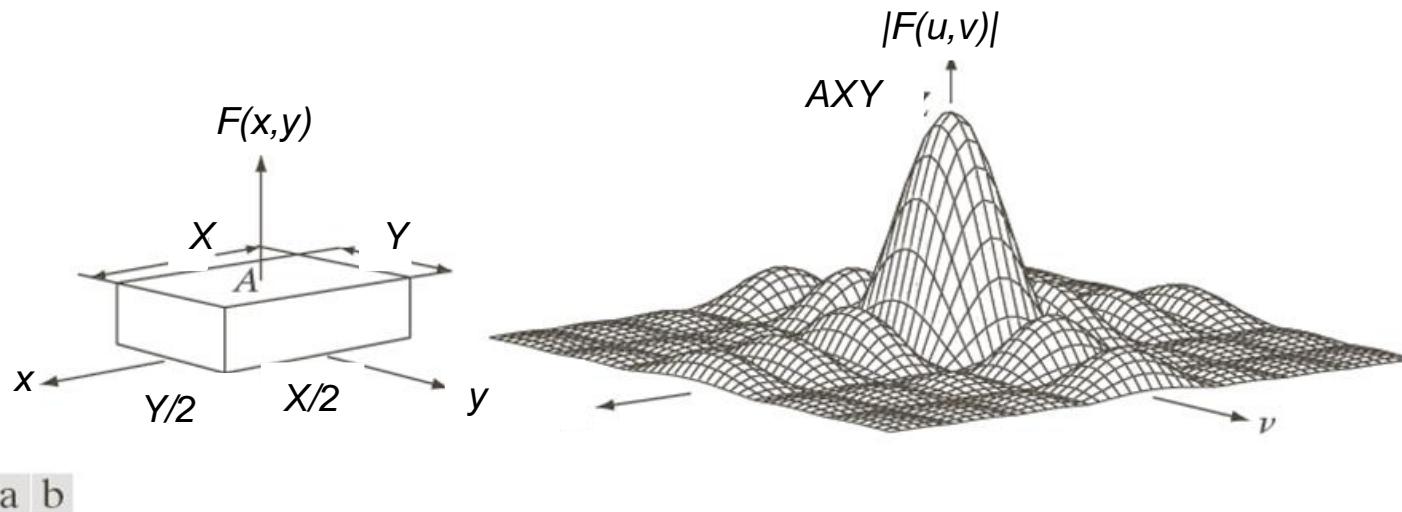
- Separability theorem

$$h(x, y) = f(x)g(y) \rightarrow H(u, v) = F(u)G(v)$$

- Rect-sinc transform  
in 2D



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

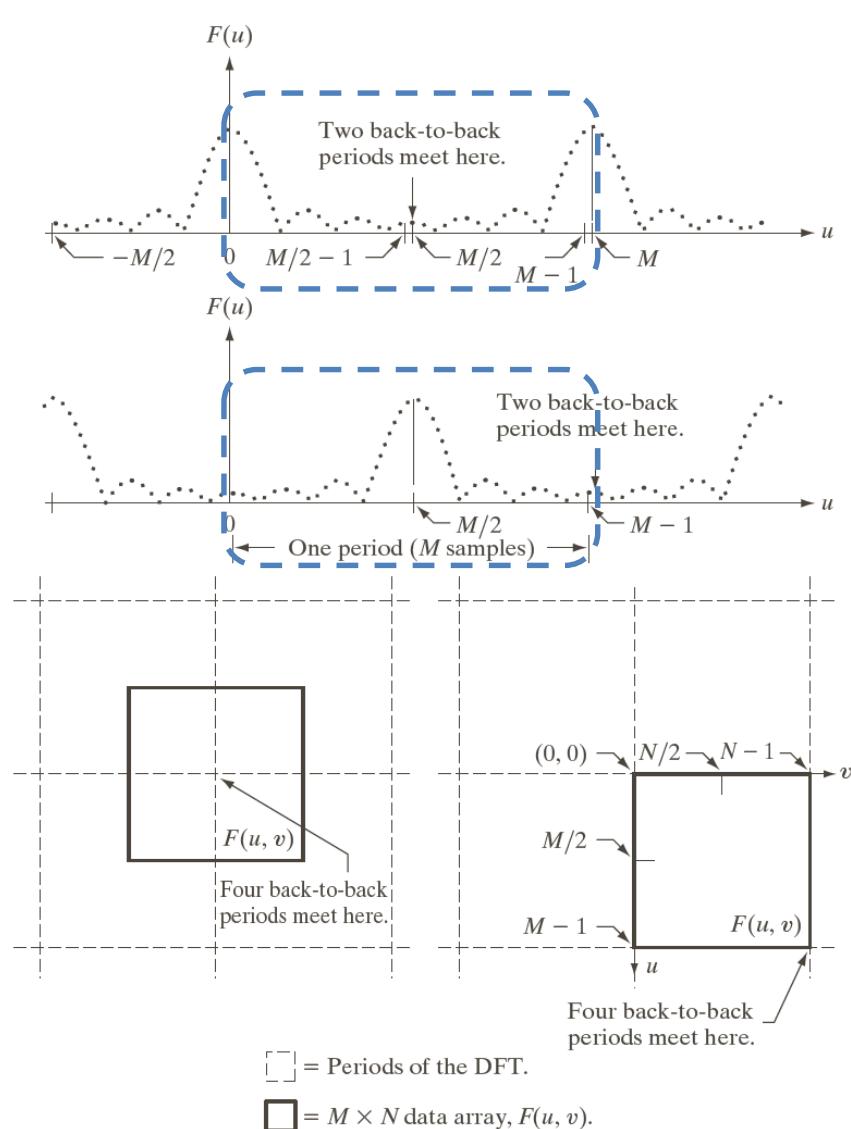


a b

**FIGURE 4.13** (a) A 2-D function, and (b) a section of its spectrum (not to scale). The block is longer along the  $t$ -axis, so the spectrum is more “contracted” along the  $\mu$ -axis. Compare with Fig. 4.4.



# Centering the Fourier spectrum

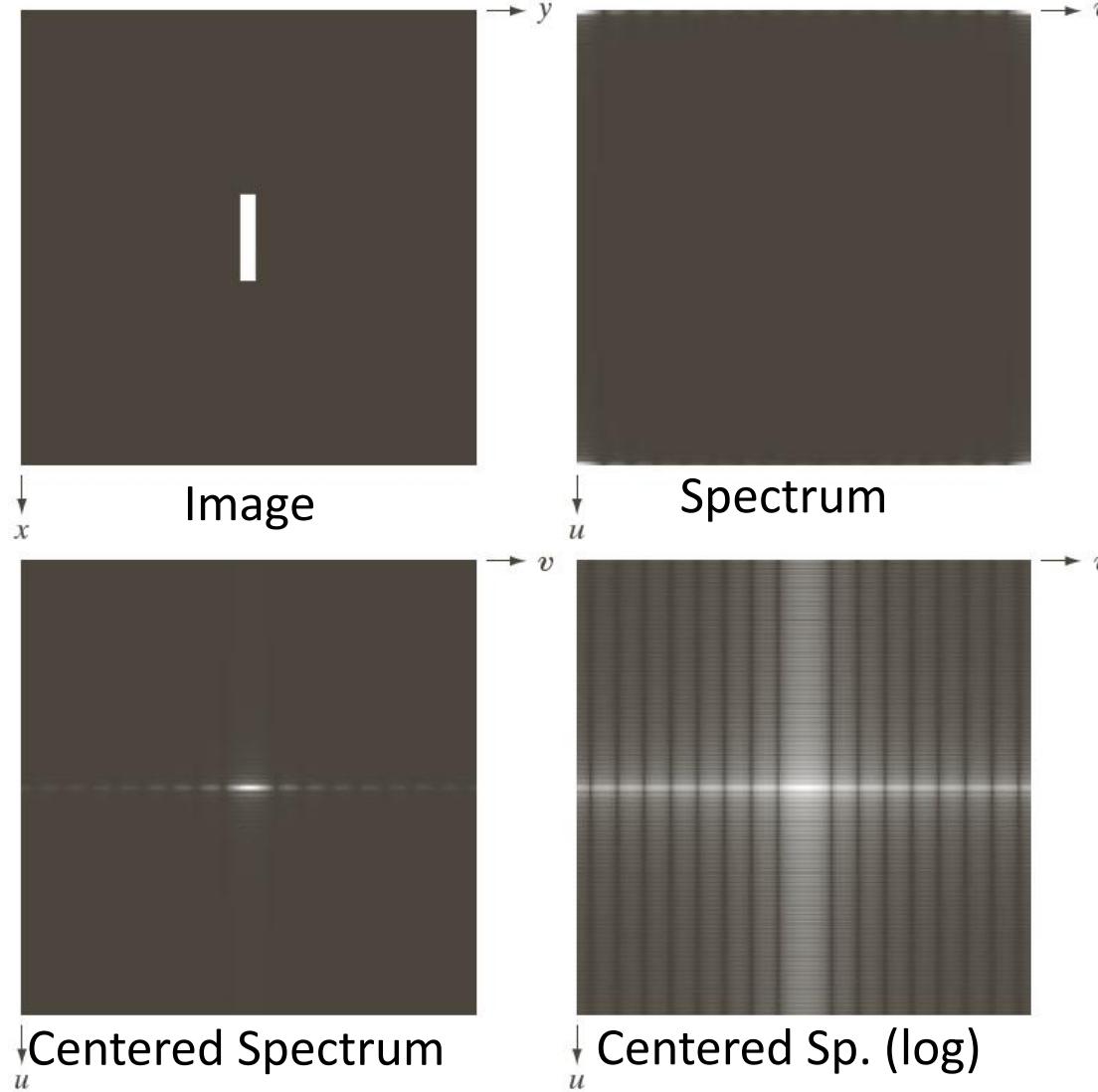


a  
b  
c d

**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).



# Spectrum centering – example

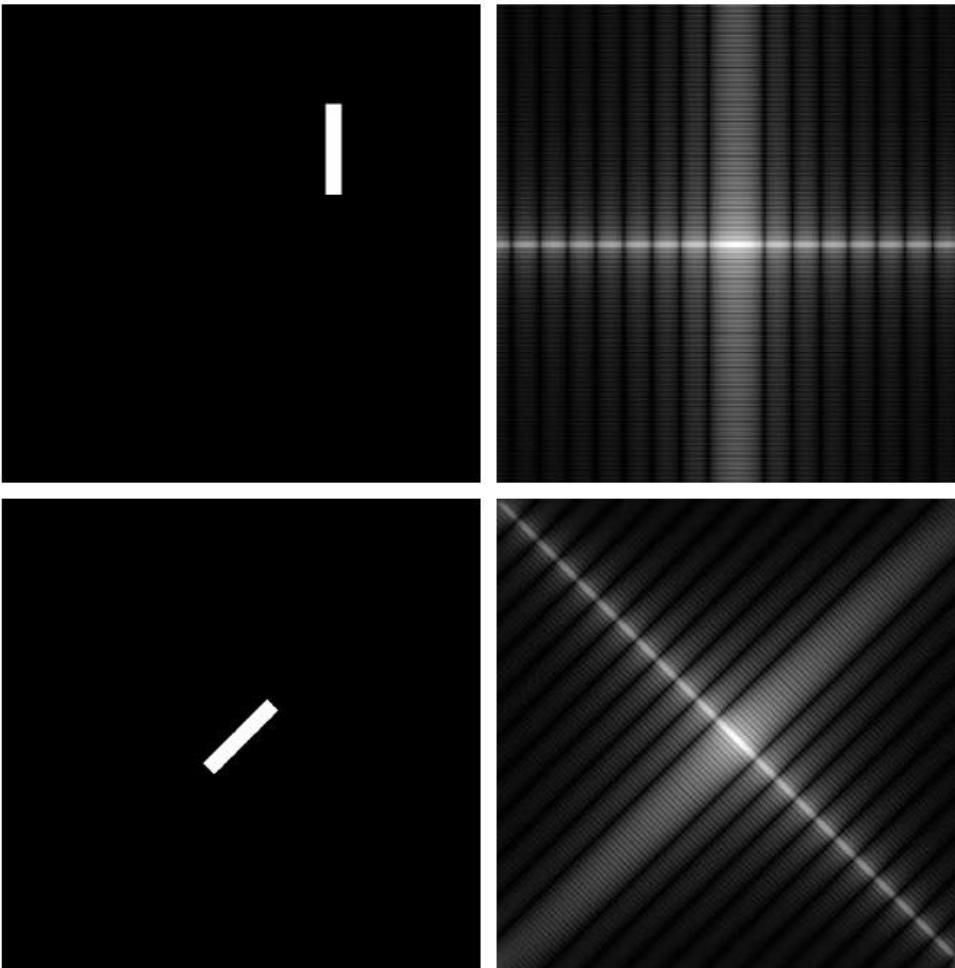


**FIGURE 4.24**

(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.



## Effect of rotation and translation on the spectrum



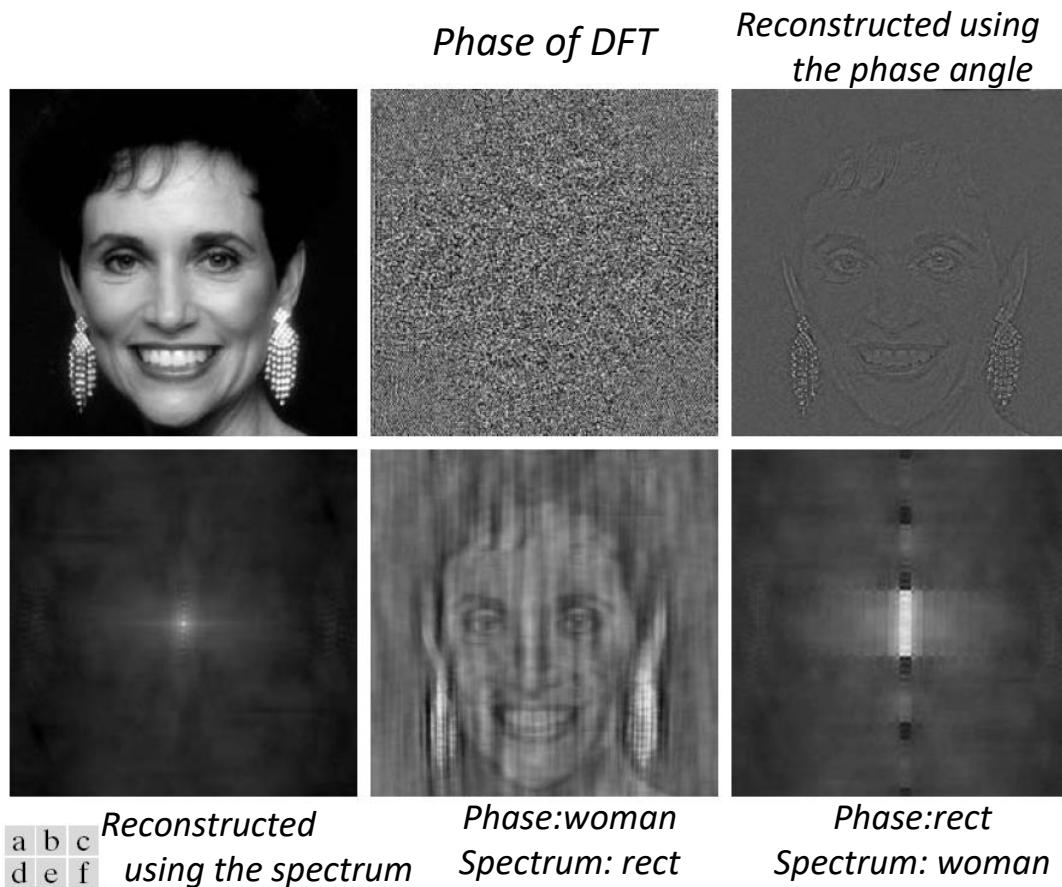
a	b
c	d

**FIGURE 4.25**

(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).

# Phase vs spectrum

IAS-LAB

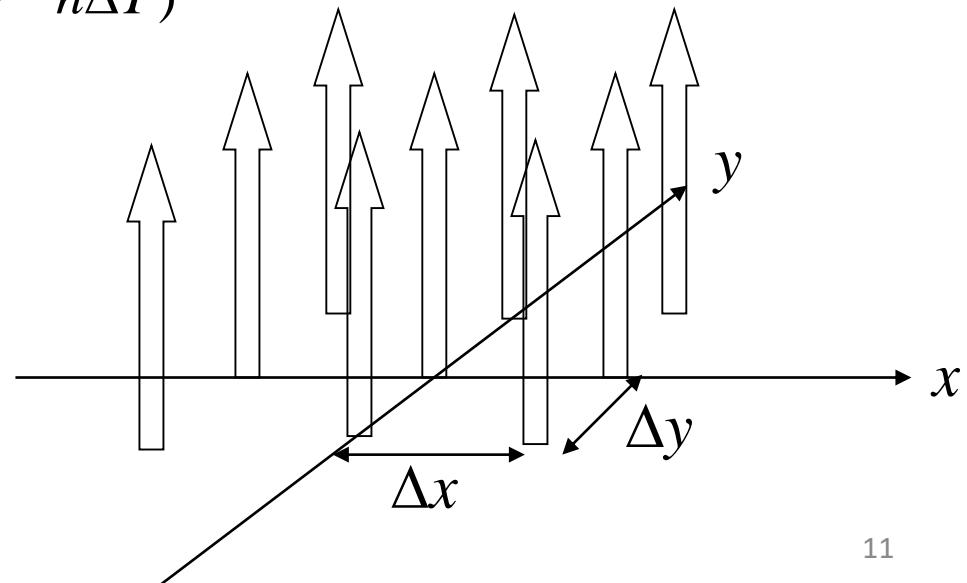


**FIGURE 4.27** (a) Woman. (b) Phase angle. (c) Woman reconstructed using only the phase angle. (d) Woman reconstructed using only the spectrum. (e) Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle in Fig. 4.24(a). (f) Reconstruction using the phase of the rectangle and the spectrum of the woman.

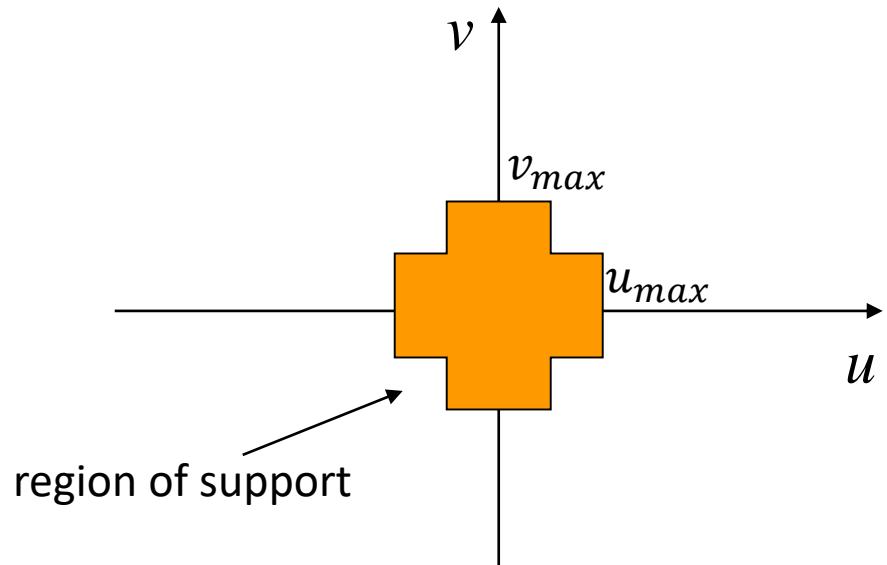
- Image:
  - Discrete domain function
  - Sampling of a continuous function by a 2D impulse train

$$s_{\Delta X, \Delta Y}(x, y) = \sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} \delta(x - m\Delta X, y - n\Delta Y)$$

$$f(m, n) = f(x, y)s_{\Delta X, \Delta Y}(x, y)$$



- An image  $f(x, y)$  is band-limited if its Fourier transform  $F(u, v)$  is s.t.:
  - $F(u, v) = 0 \text{ for } |u| > u_{max}$
  - $F(u, v) = 0 \text{ for } |v| > v_{max}$
- $u_{max}$  and  $v_{max}$  define the bandwidth of the image





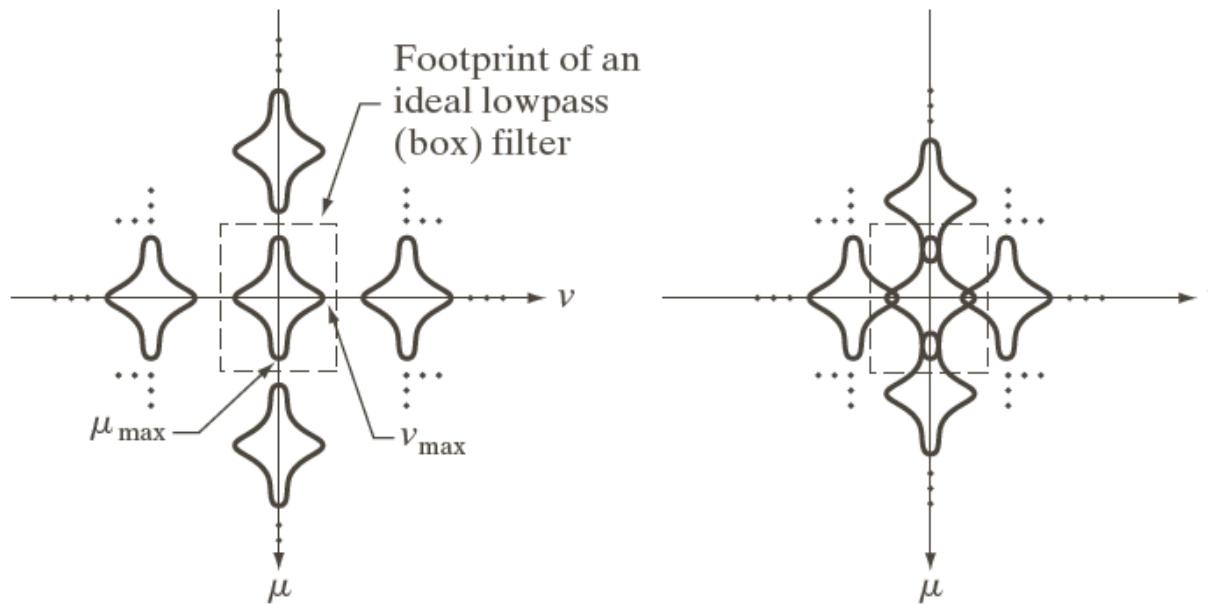
- Consider an image  $f(x, y)$  that is band-limited
- Let  $f(x, y)$  be uniformly sampled over an orthogonal lattice with spacing  $\Delta X$  and  $\Delta Y$
- Then,  $f(x, y)$  can be reconstructed from its sampled values without any loss of information
  - Sampling rate shall be higher than the Nyquist frequency

$$F_X = \frac{1}{\Delta X} > 2u_{max} \text{ and } F_Y = \frac{1}{\Delta Y} > 2v_{max}$$

# 2D space aliasing

IAS-LAB

- What if the previous condition does not hold?
- Periodic replications overlap
- Aliasing
  - Frequencies above sampling freq/2 appear as they were below the sampling rate

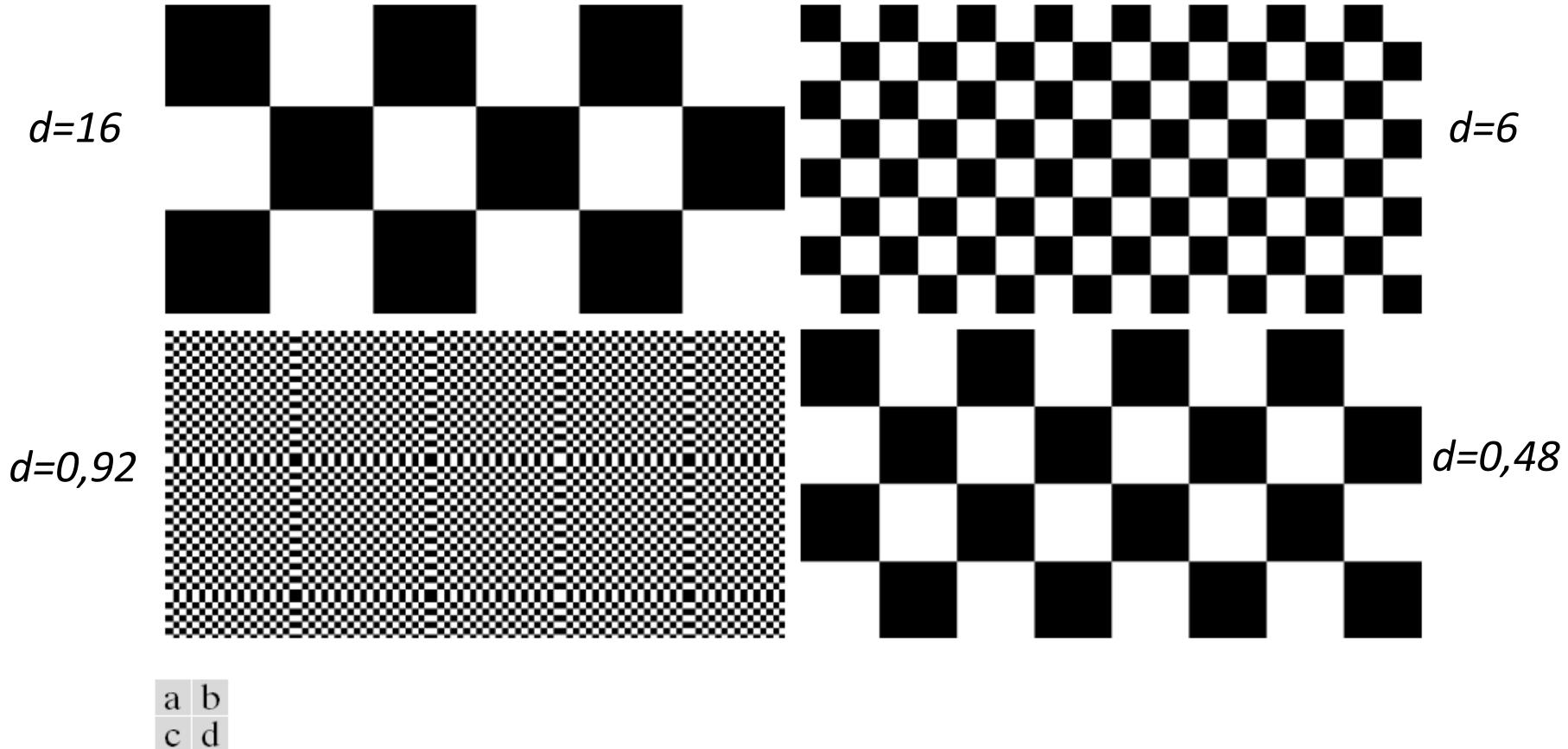


a b

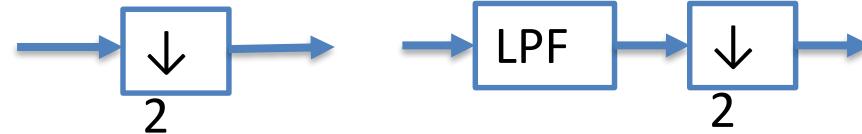
**FIGURE 4.15**  
Two-dimensional Fourier transforms of (a) an over-sampled, and (b) under-sampled band-limited function.

# 2D space aliasing – example

IAS-LAB



**FIGURE 4.16** Aliasing in images. In (a) and (b), the lengths of the sides of the squares are 16 and 6 pixels, respectively, and aliasing is visually negligible. In (c) and (d), the sides of the squares are 0.9174 and 0.4798 pixels, respectively, and the results show significant aliasing. Note that (d) masquerades as a “normal” image.



a b c

**FIGURE 4.17** Illustration of aliasing on resampled images. (a) A digital image with negligible visual aliasing. (b) Result of resizing the image to 50% of its original size by pixel deletion. Aliasing is clearly visible. (c) Result of blurring the image in (a) with a  $3 \times 3$  averaging filter prior to resizing. The image is slightly more blurred than (b), but aliasing is not longer objectionable. (Original image courtesy of the Signal Compression Laboratory, University of California, Santa Barbara.)

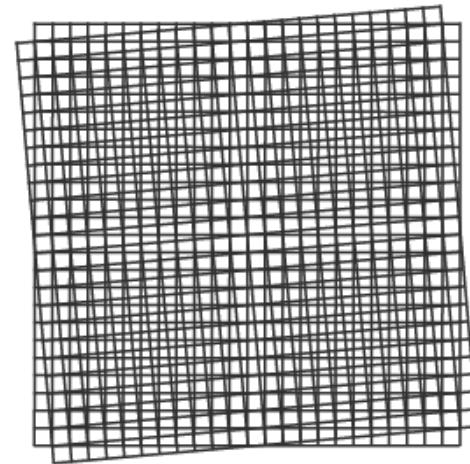


- Wagon wheel effect





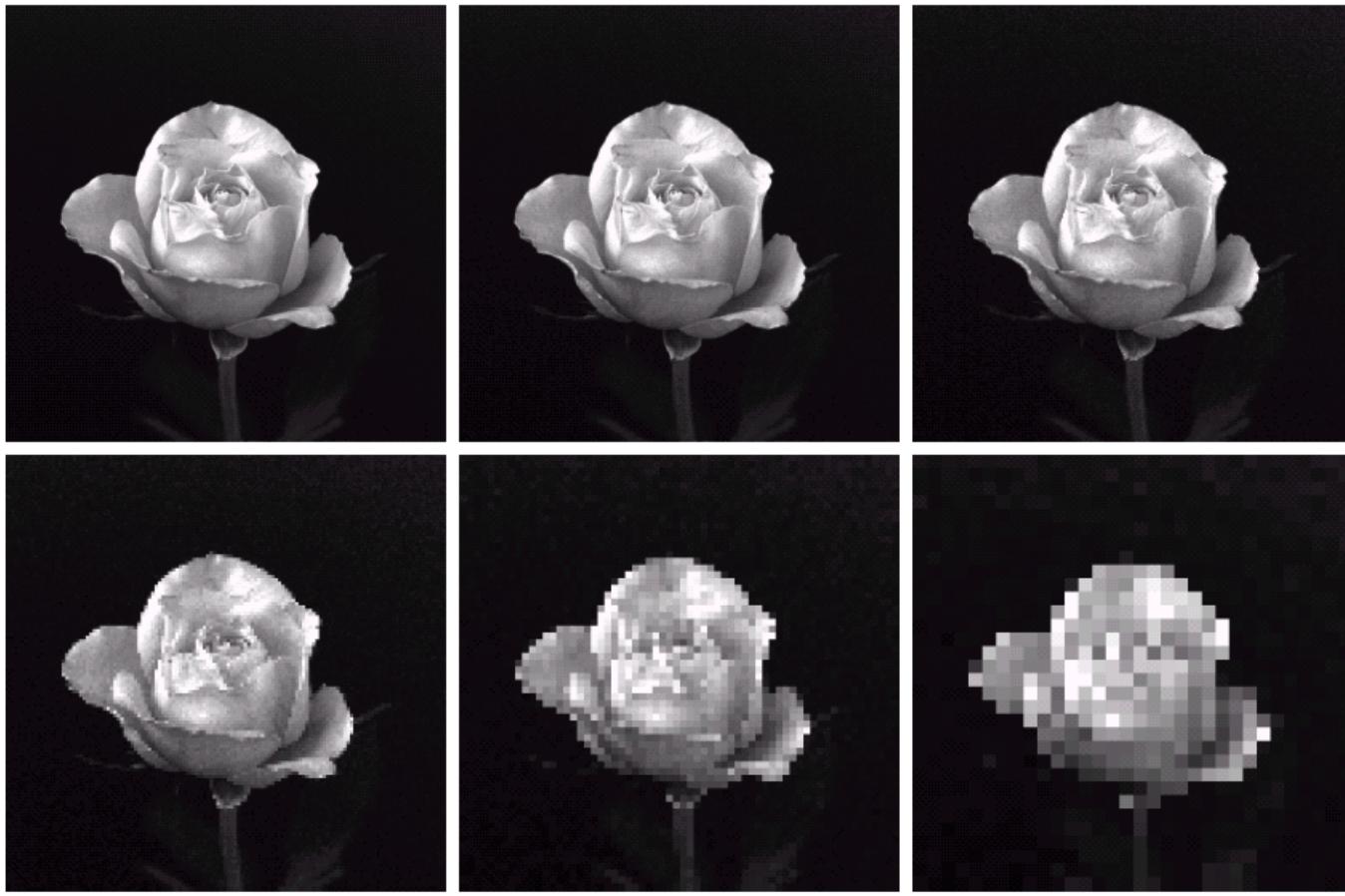
- Caused by beating of similar patterns (grating with similar spacing)
  - No aliasing is involved



# End of lecture 12, video 1

Start of lecture 12, video 2

Image interpolation and  
resampling



a	b	c
d	e	f

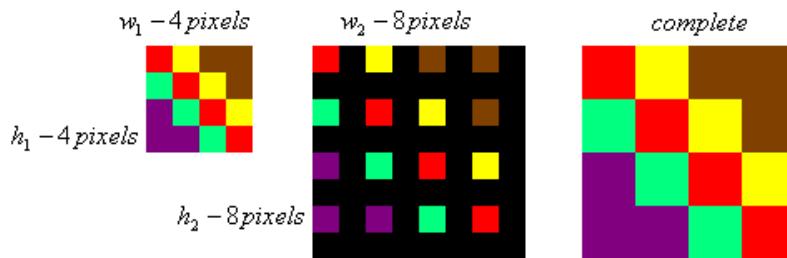
**FIGURE 2.20** (a)  $1024 \times 1024$ , 8-bit image. (b)  $512 \times 512$  image resampled into  $1024 \times 1024$  pixels by row and column duplication. (c) through (f)  $256 \times 256$ ,  $128 \times 128$ ,  $64 \times 64$ , and  $32 \times 32$  images resampled into  $1024 \times 1024$  pixels.



- Change in sampling interval
  - Interpolation: larger image
  - Decimation: smaller image
  - Resampling to a different lattice (e.g., after a geometric transform)
- Image pixels evaluated in locations different from the sampling points!



- Simplest idea
- Pixel color: taken from the closest pixel
  - Fast execution
  - Poor results (pixelization effect)



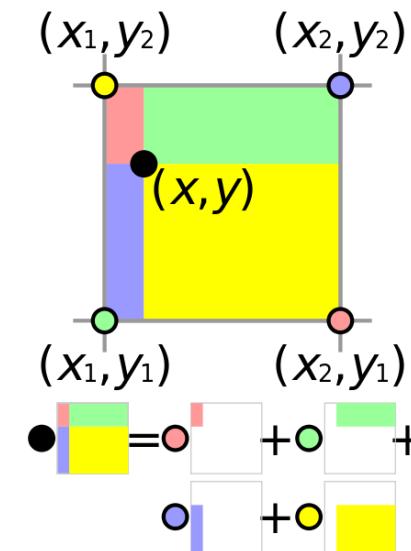
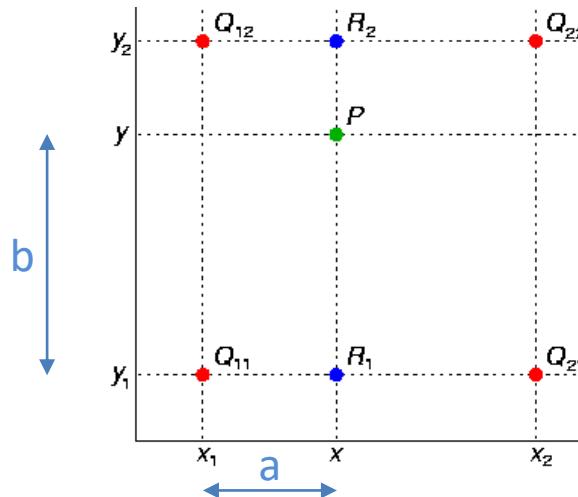
# Bilinear interpolation

- Weighted average of the closest 4 samples
- Coefficients depend on the distance to the samples

$$p(x, y) = (1 - a)(1 - b)I_1 + a(1 - b)I_2 + (1 - a)bI_3 + abI_4$$

– Non linear WRT x and y

- Good balance between speed and accuracy

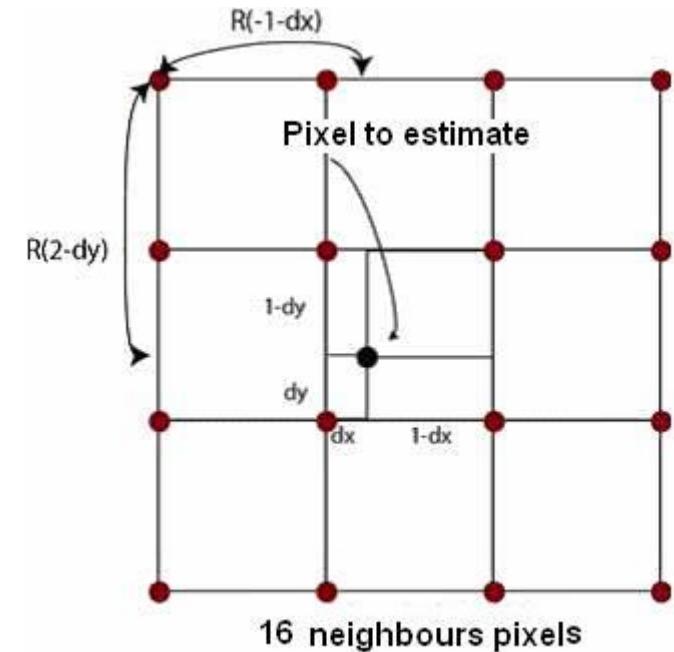


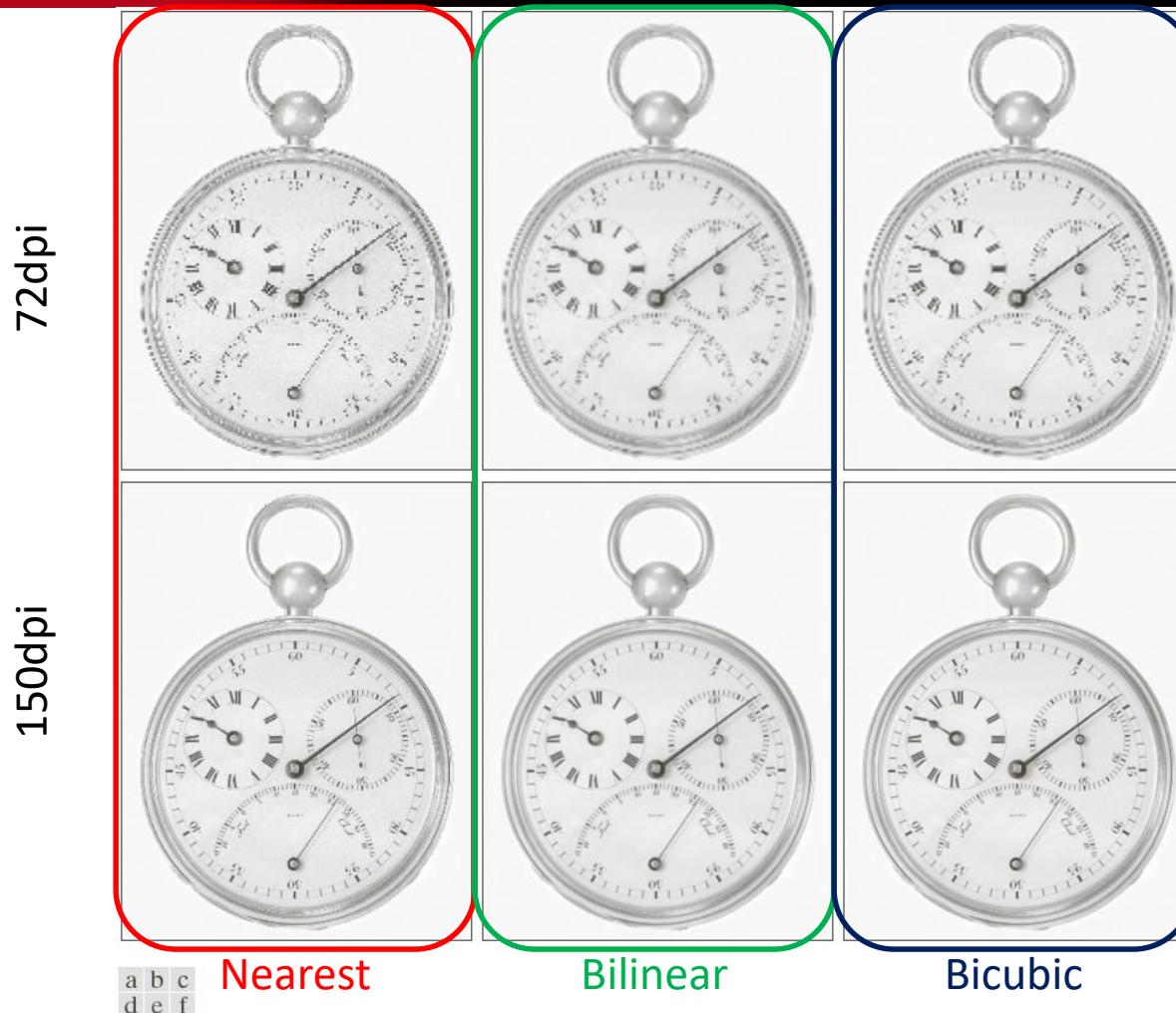
# Bicubic interpolation

- The 16 closest samples are considered

$$p(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{i,j} x^i y^j$$

- 3-degree polynomials
- 16 unknowns
  - 4 constraints on sample locations
  - 8 constraints on 1° order derivatives (4 on x and 4 on y)
  - 4 constraints on x-y cross-derivatives
- Slow, good accuracy





**FIGURE 2.24** (a) Image reduced to 72 dpi and zoomed back to its original size ( $3692 \times 2812$  pixels) using nearest neighbor interpolation. This figure is the same as Fig. 2.20(d). (b) Image shrunk and zoomed using bilinear interpolation. (c) Same as (b) but using bicubic interpolation. (d)–(f) Same sequence, but shrinking down to 150 dpi instead of 72 dpi [Fig. 2.24(d) is the same as Fig. 2.20(c)]. Compare Figs. 2.24(e) and (f), especially the latter, with the original image in Fig. 2.20(a).

# Interpolation – examples

IAS-LAB

Bi-Linear



Lanczos 3



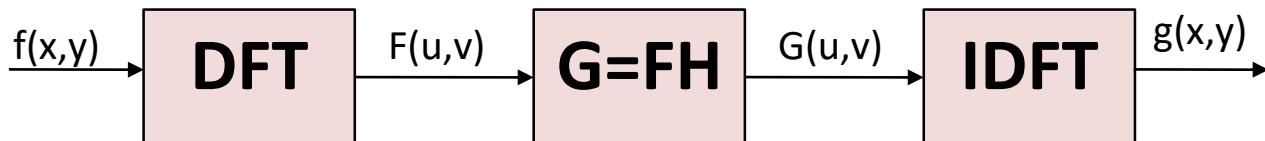
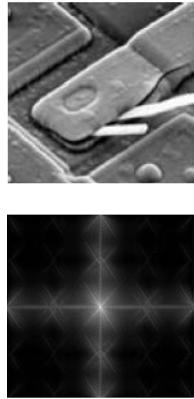
# End of lecture 12, video 2

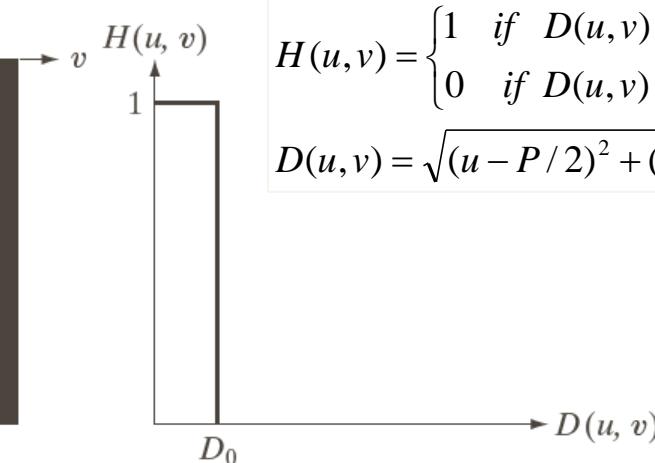
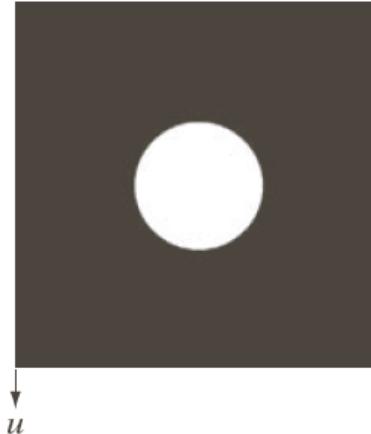
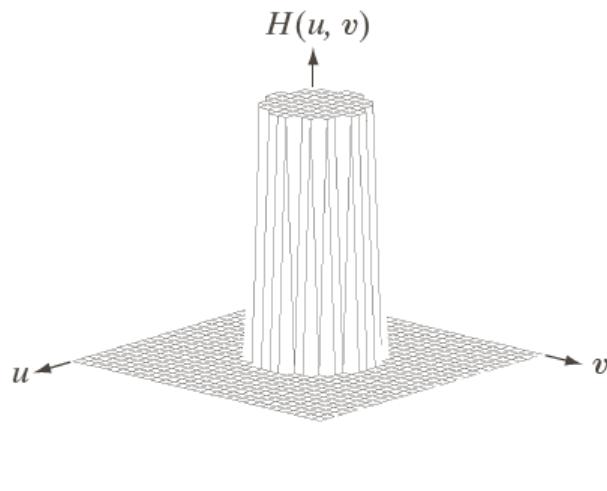
# Start of lecture 12, video 3

## Filtering in the frequency domain



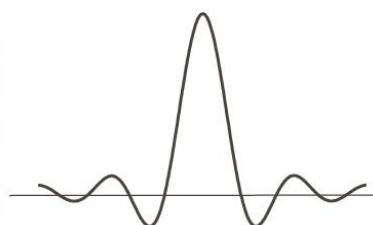
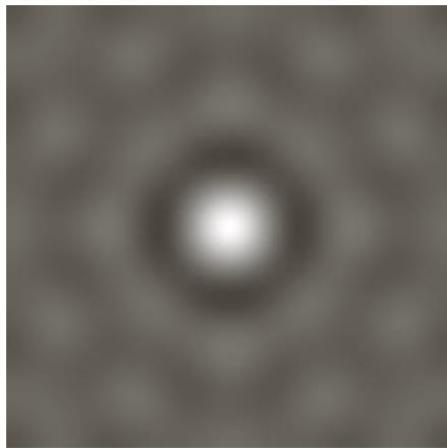
- Frequency domain: implicit periodicity
- $H(u, v)$  commonly specified for spectrum centered in  $(M/2, N/2)$
- Images in the frequency domain with real values





a b c

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

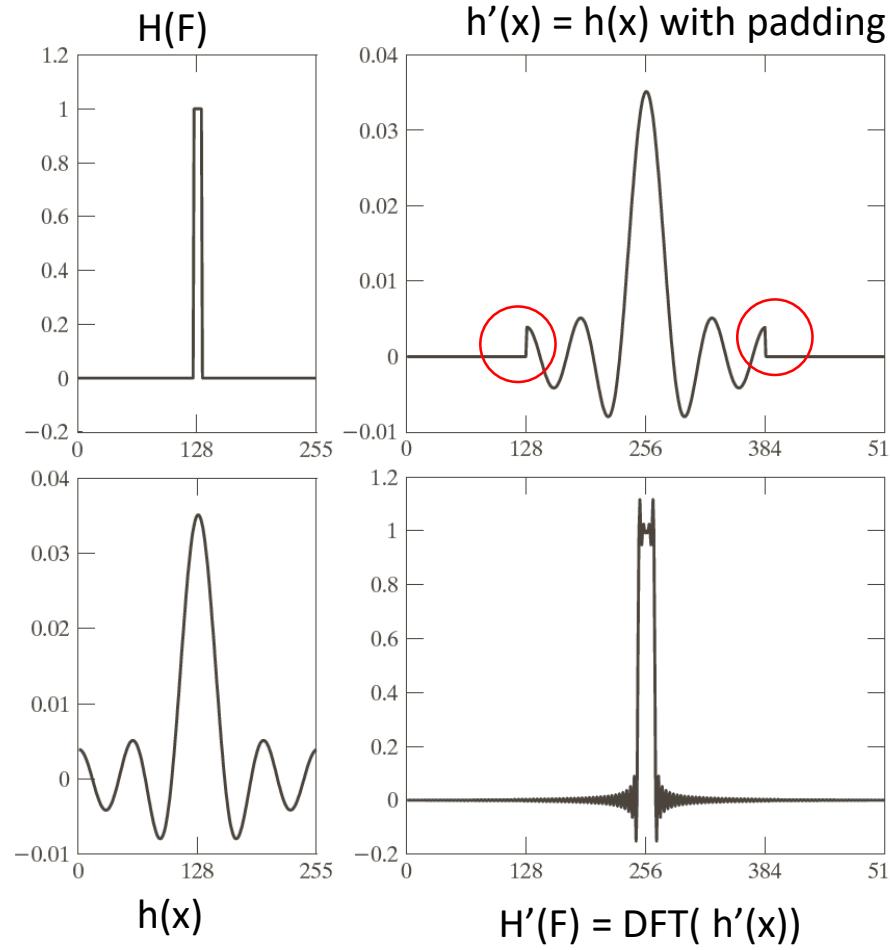


a b

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

# Ringing artifacts

IAS-LAB

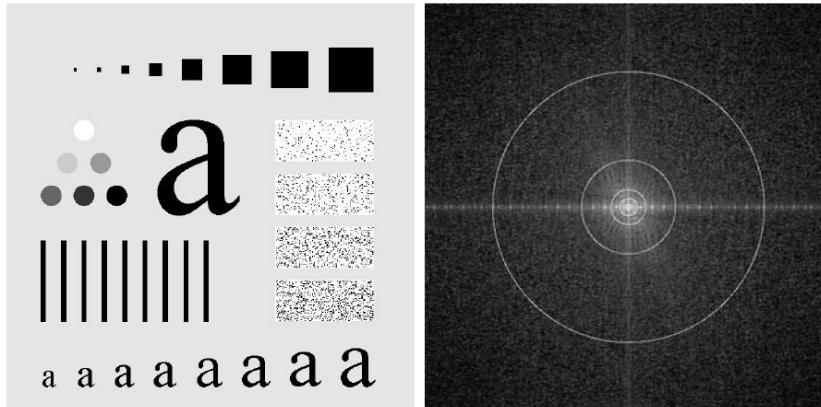


a	c
b	d

**FIGURE 4.34**

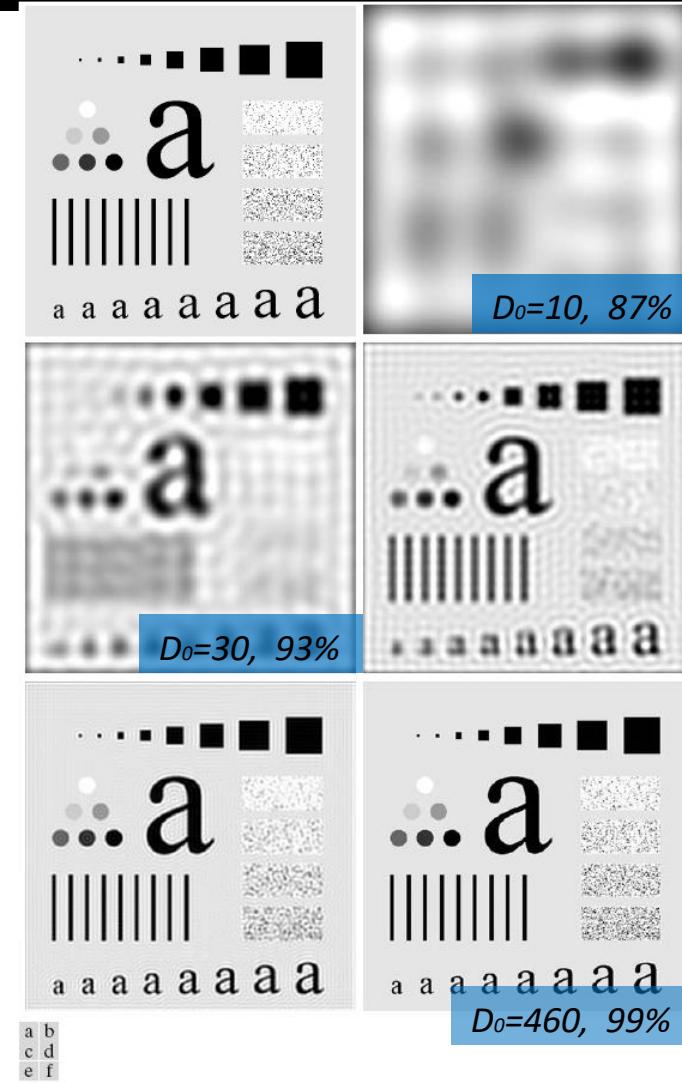
- (a) Original filter specified in the (centered) frequency domain.
- (b) Spatial representation obtained by computing the IDFT of (a).
- (c) Result of padding (b) to twice its length (note the discontinuities).
- (d) Corresponding filter in the frequency domain obtained by computing the DFT of (c). Note the ringing caused by the discontinuities in (c). (The curves appear continuous because the points were joined to simplify visual analysis.)

- The ringing effect is noticeable



a b

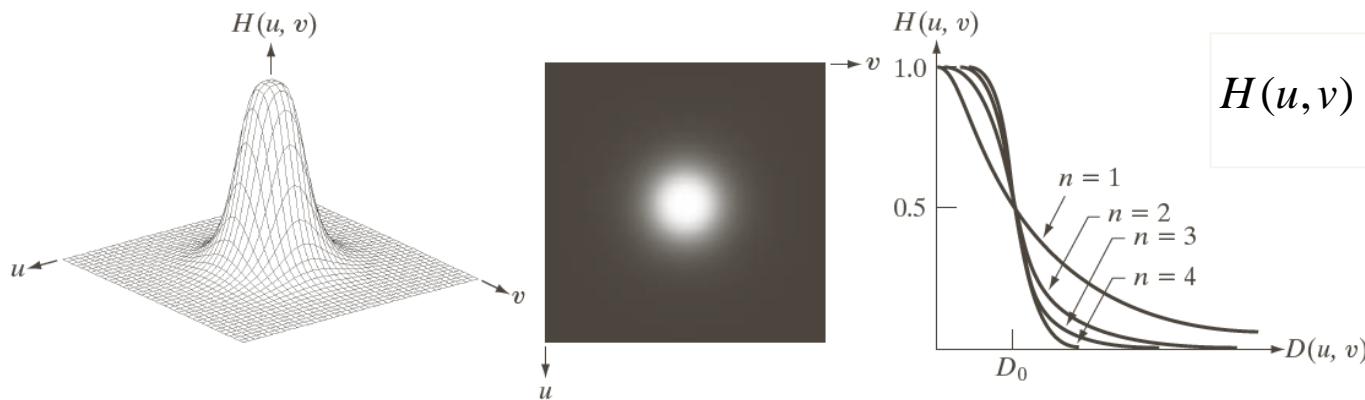
**FIGURE 4.41** (a) Test pattern of size  $688 \times 688$  pixels, and (b) its Fourier spectrum. The spectrum is double the image size due to padding but is shown in half size so that it fits in the page. The superimposed circles have radii equal to 10, 30, 60, 160, and 460 with respect to the full-size 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.

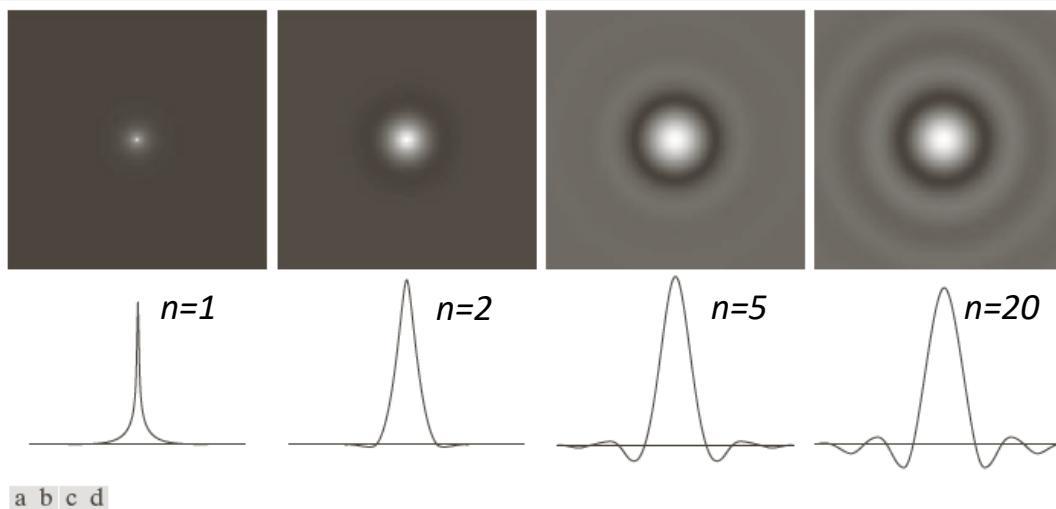
# Butterworth lowpass filter

IAS-LAB



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.



*The ringing effect grows with the order of the filter*

**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  $1000 \times 1000$  and the cutoff frequency is 5). Observe how ringing increases as a function of filter order.



# Butterworth – example

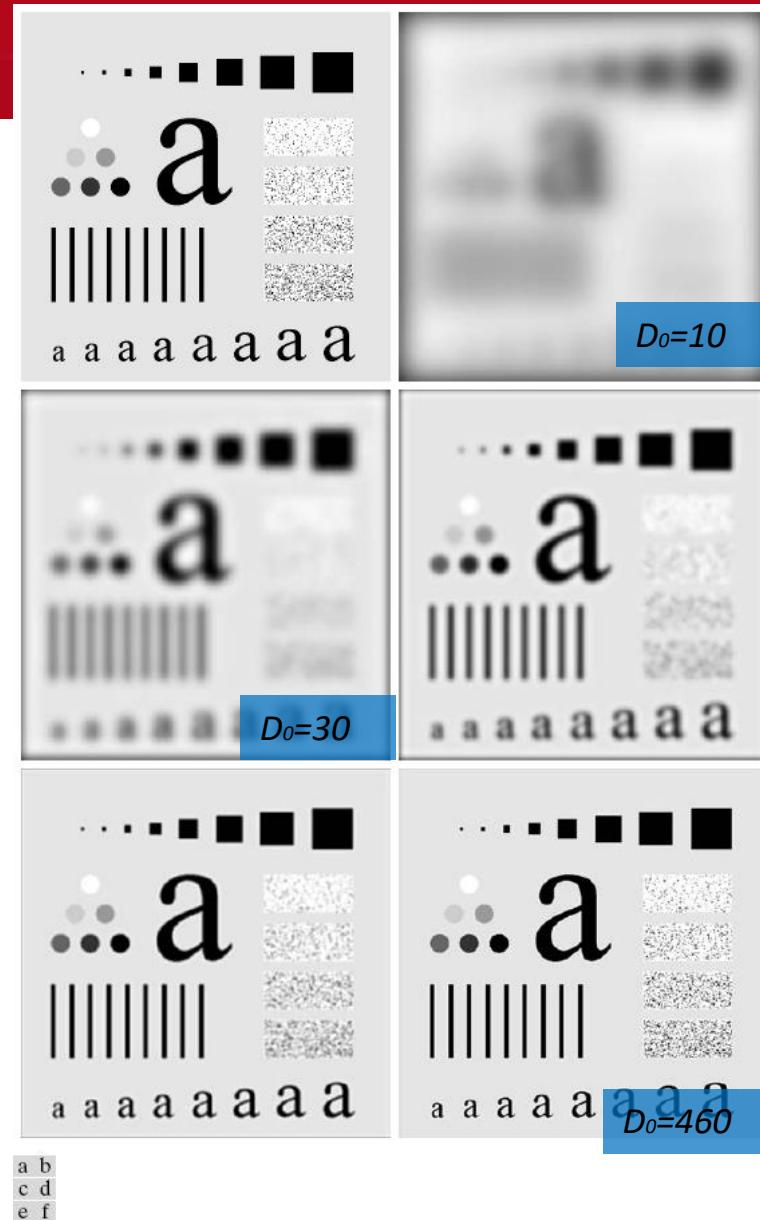
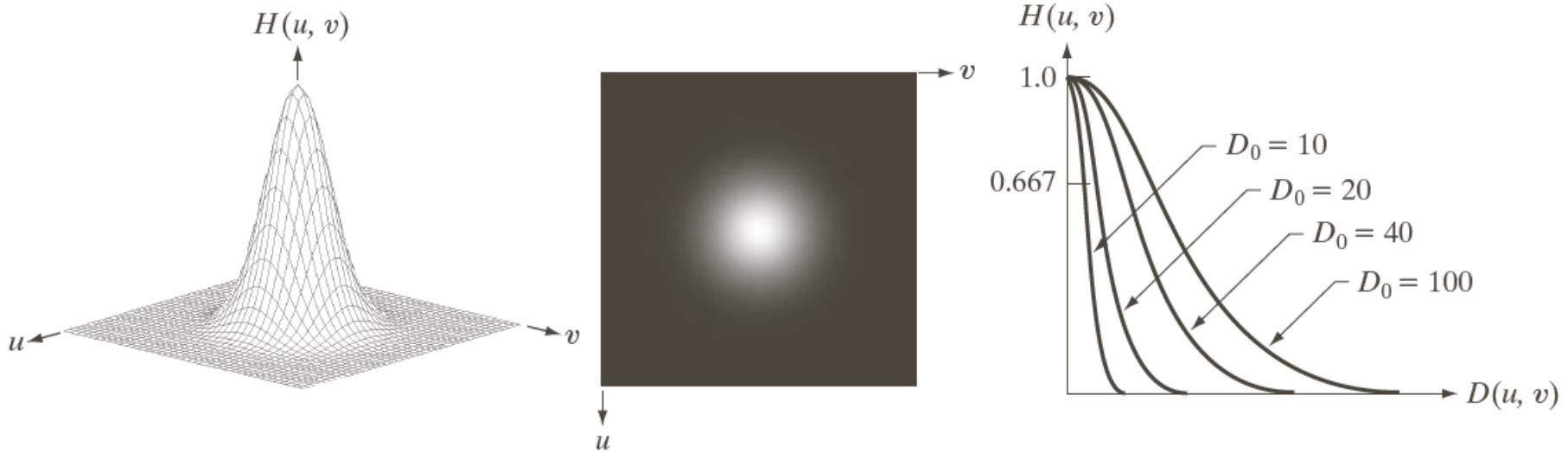


FIGURE 4.45 (a) Original image. (b)–(f) Results of filtering using BLPFs of order 2, with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Fig. 4.42.

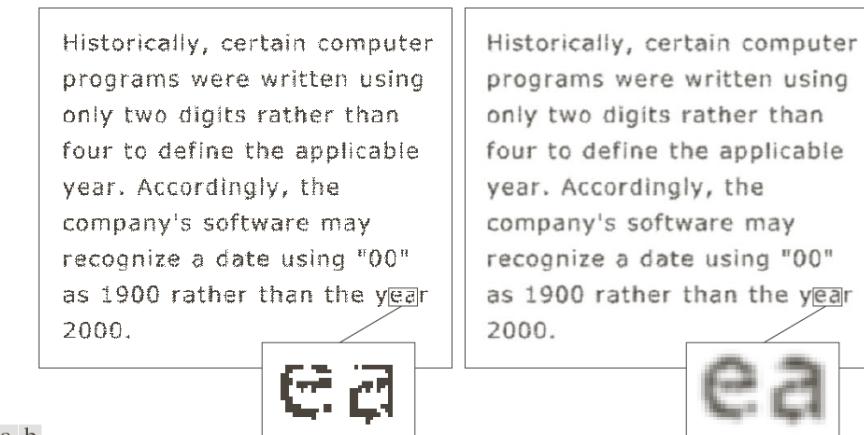
- No ringing

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



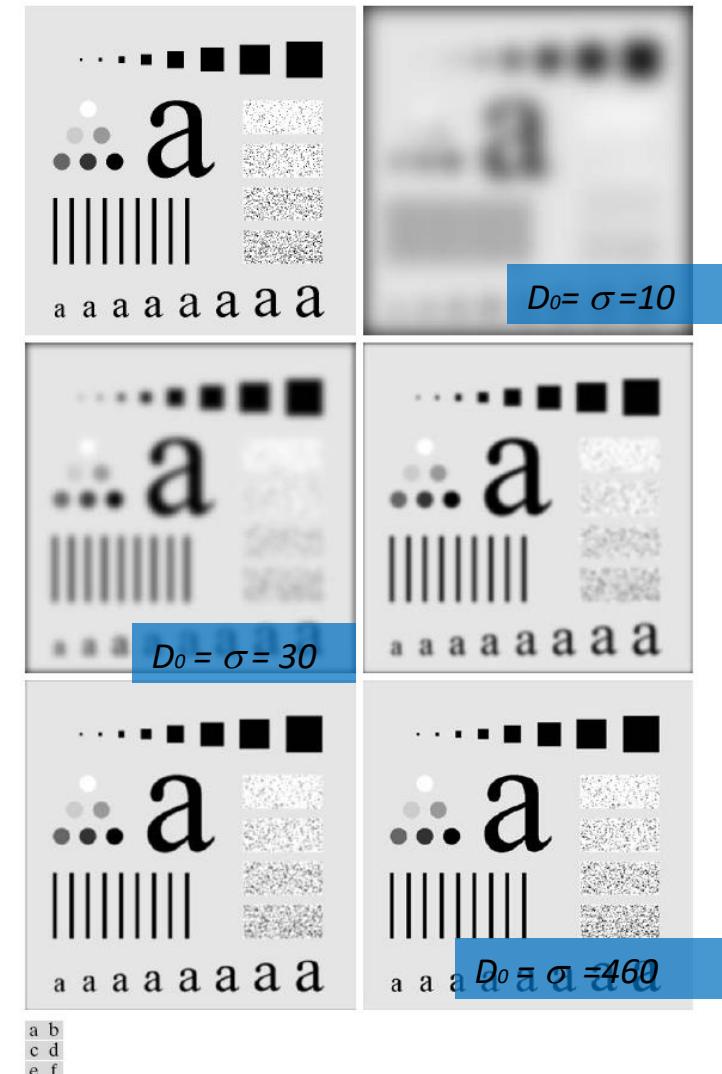
a b c

**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  $D_0$ .



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).



**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.

- Beautification

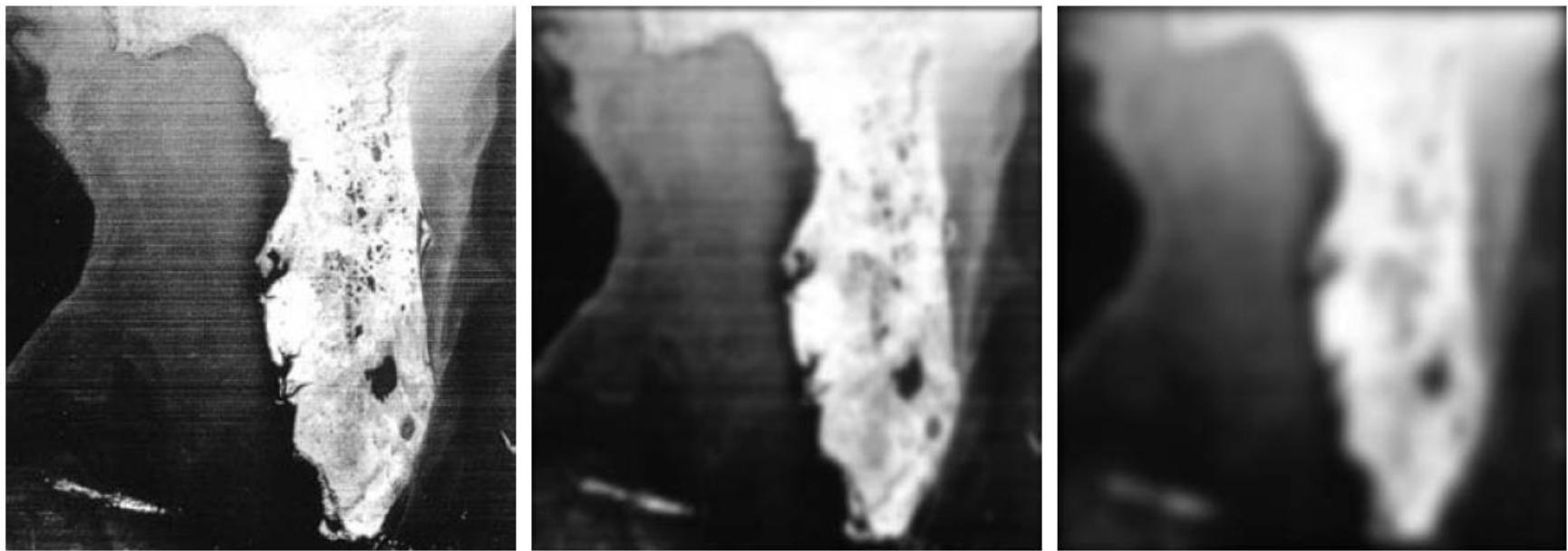


a b c

**FIGURE 4.50** (a) Original image ( $784 \times 732$  pixels). (b) Result of filtering using a GLPF with  $D_0 = 100$ . (c) Result of filtering using a GLPF with  $D_0 = 80$ . Note the reduction in fine skin lines in the magnified sections in (b) and (c).



# Gaussian filter – example



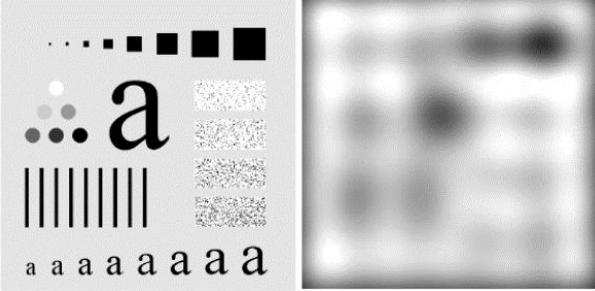
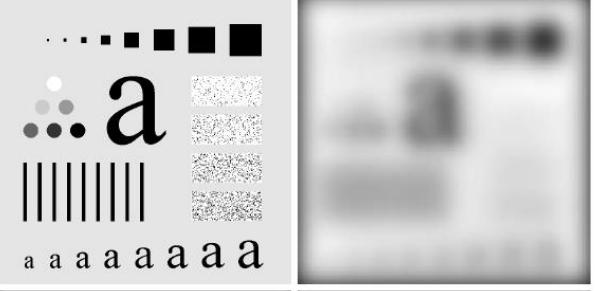
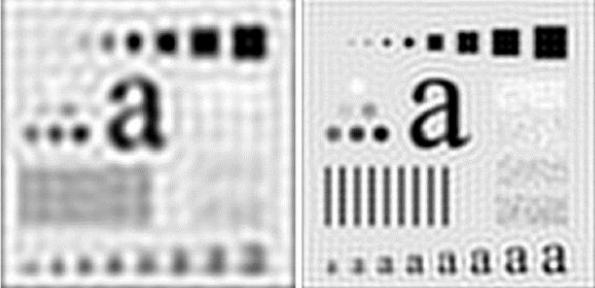
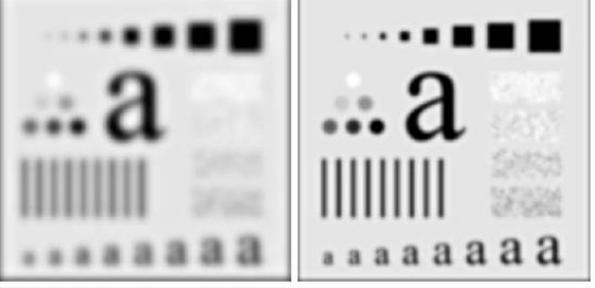
a b c

**FIGURE 4.51** (a) Image showing prominent horizontal scan lines. (b) Result of filtering using a GLPF with  $D_0 = 50$ . (c) Result of using a GLPF with  $D_0 = 20$ . (Original image courtesy of NOAA.)

# Lowpass filter overview

**TABLE 4.4**

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

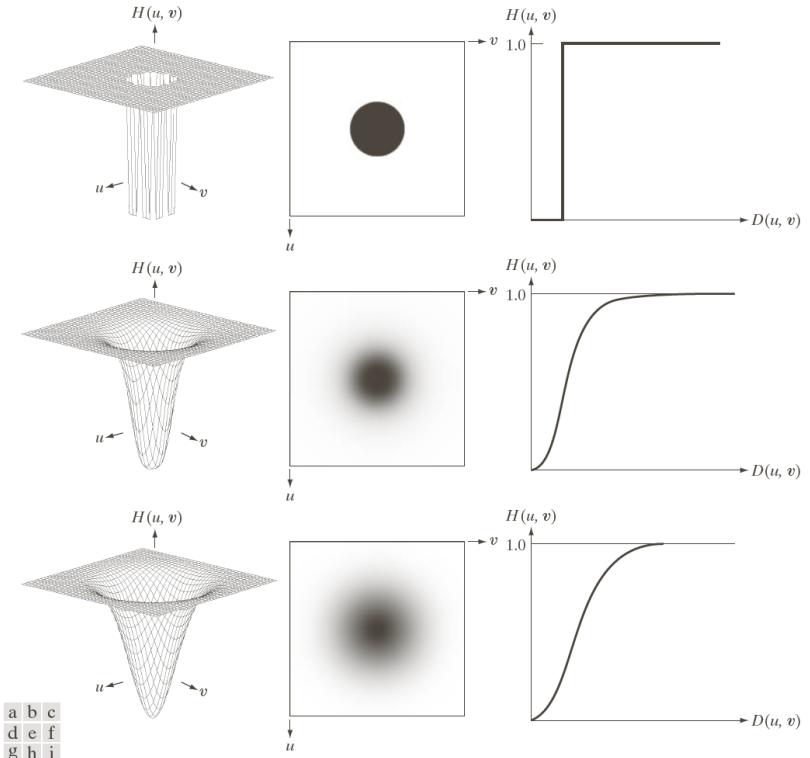
Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$	$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$	$H(u, v) = e^{-D^2(u,v)/2D_0^2}$
		
		

# Highpass filter

IAS-LAB

- Sharpening effect

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$



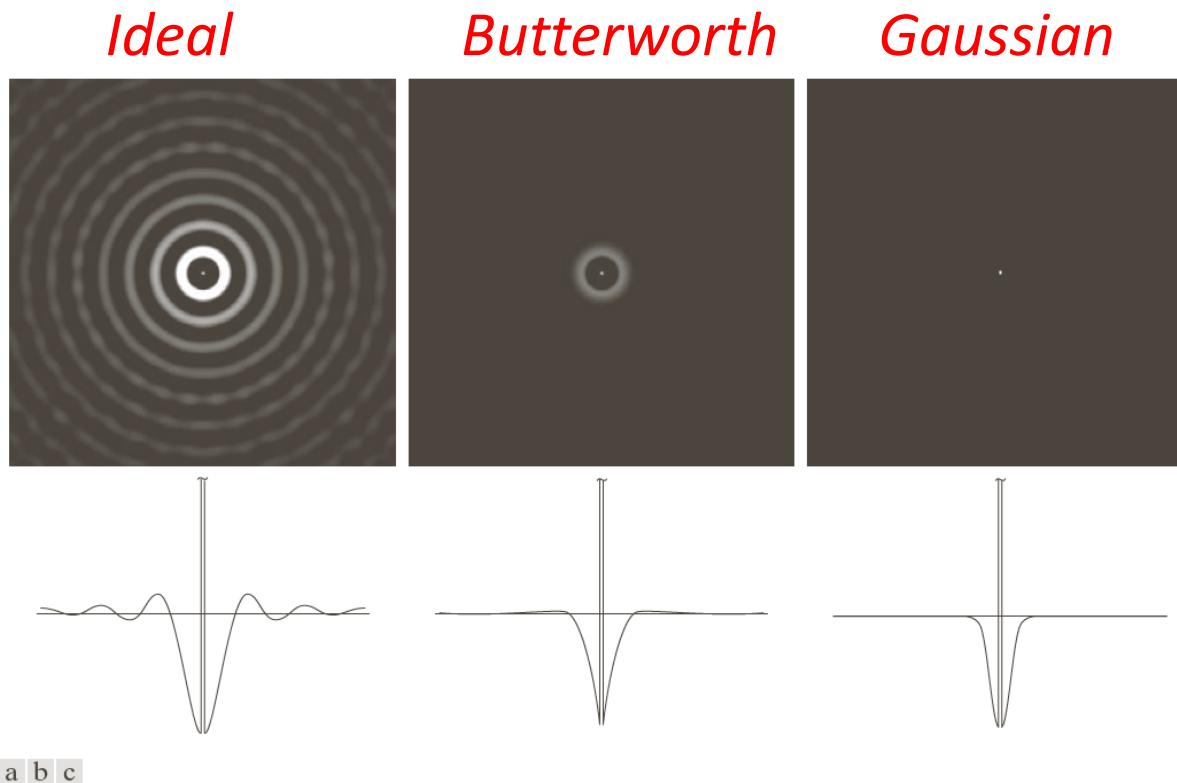
**FIGURE 4.52** Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

**TABLE 4.5**

Highpass filters.  $D_0$  is the cutoff frequency and  $n$  is the order of the Butterworth filter.

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

- Ringing is present as in lowpass filtering



**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.



a b c

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

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



a | b | c

**FIGURE 4.55** Results of highpass filtering the image in Fig. 4.41(a) using a BHPF of order 2 with  $D_0 = 30, 60,$  and  $160,$  corresponding to the circles in Fig. 4.41(b). These results are much smoother than those obtained with an IHPF.

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



a | b | c

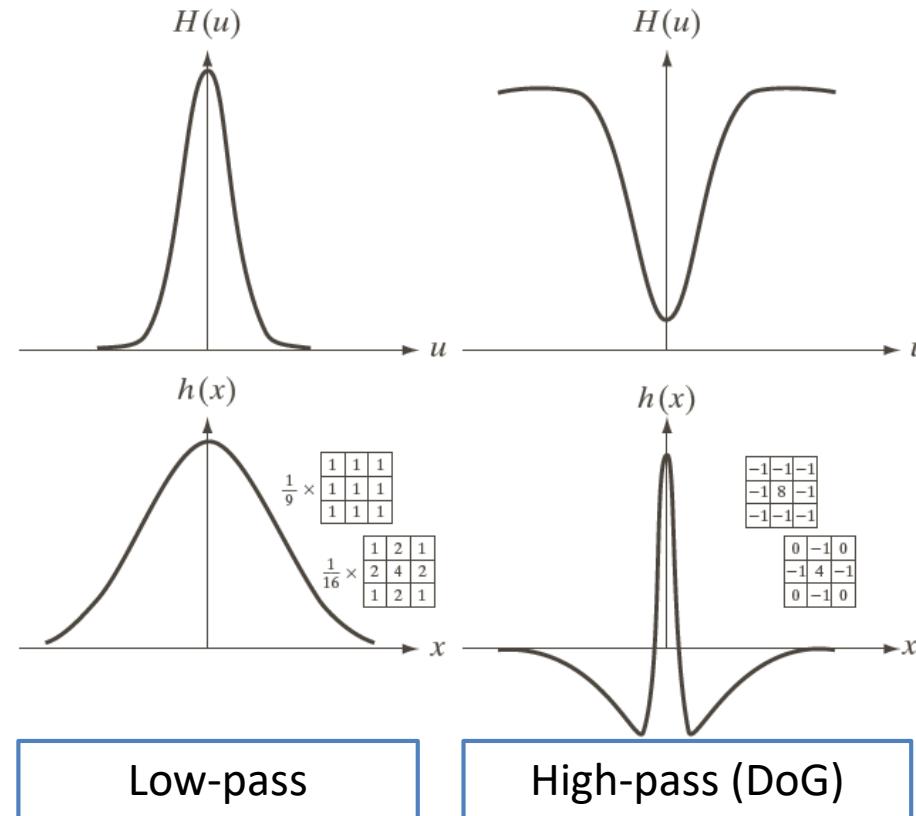
**FIGURE 4.56** Results of highpass filtering the image in Fig. 4.41(a) using a GHPF with  $D_0 = 30, 60$ , and  $160$ , corresponding to the circles in Fig. 4.41(b). Compare with Figs. 4.54 and 4.55.

$$H(u, v) = 1 - e^{-\frac{D^2(u,v)}{2D_0^2}}$$

# Deriving the spatial mask

IAS-LAB

- Fourier inverse transform: guide to derive the spatial mask
- Low-pass: average
- High-pass: sharpening



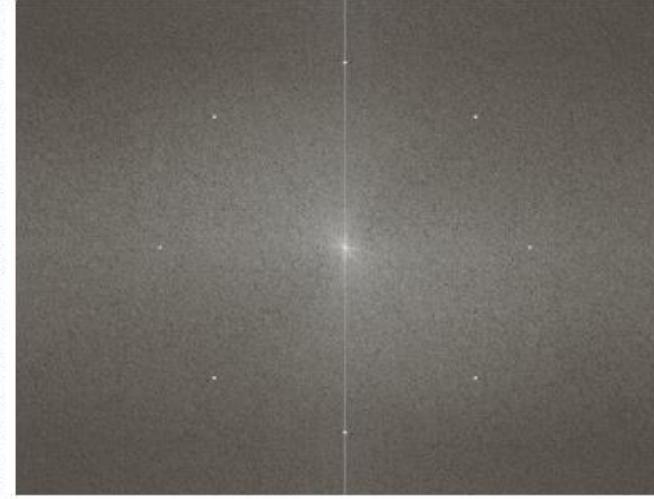
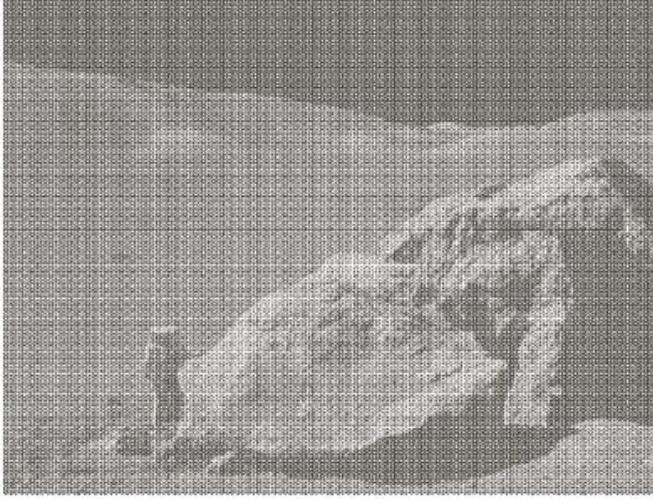
**FIGURE 4.37**  
 (a) A 1-D Gaussian lowpass filter in the frequency domain.  
 (b) Spatial lowpass filter corresponding to (a). (c) Gaussian highpass filter in the frequency domain. (d) Spatial highpass filter corresponding to (c). The small 2-D masks shown are spatial filters we used in Chapter 3.



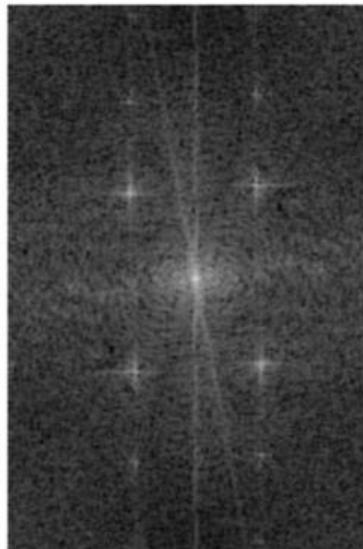
UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

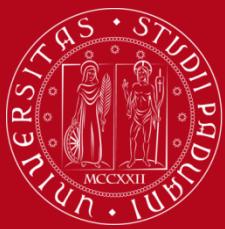
# Noise in frequency domain

IAS-LAB

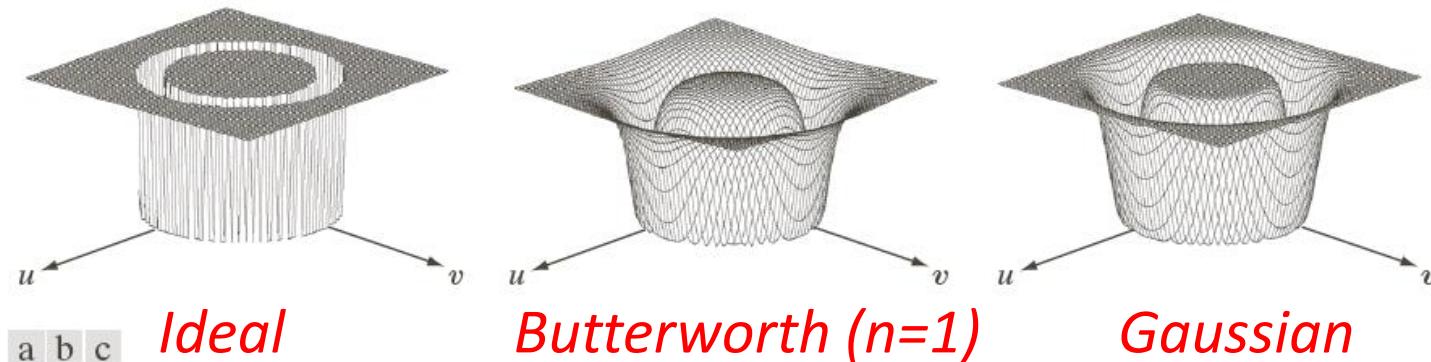


What should  
we do... ?





- Operate on selected frequency bands
  - Band-pass filter
  - Band-reject filter
- Operate on selected frequency rectangles/areas



**FIGURE 5.15** From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

**TABLE 4.6**

Bandreject filters.  $W$  is the width of the band,  $D$  is the distance  $D(u, v)$  from the center of the filter,  $D_0$  is the cutoff frequency, and  $n$  is the order of the Butterworth filter. We show  $D$  instead of  $D(u, v)$  to simplify the notation in the table.

<b>Ideal</b>	<b>Butterworth</b>	<b>Gaussian</b>
$H(u, v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u, v) = \frac{1}{1 + \left[ \frac{DW}{D^2 - D_0^2} \right]^{2n}}$	$H(u, v) = 1 - e^{-\left[ \frac{D^2 - D_0^2}{DW} \right]^2}$

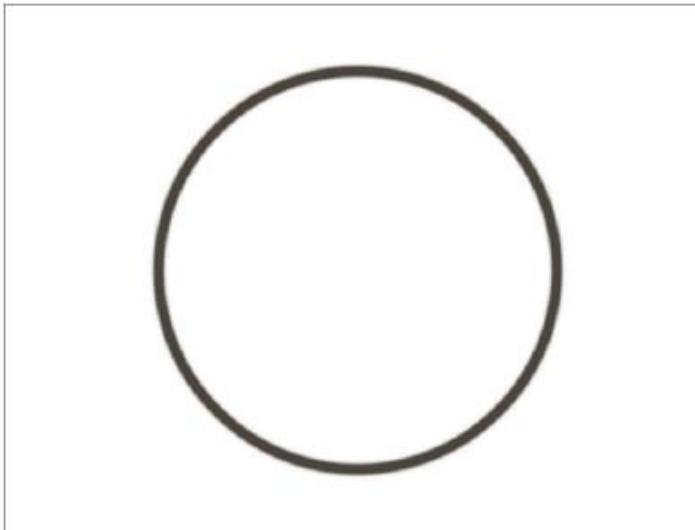
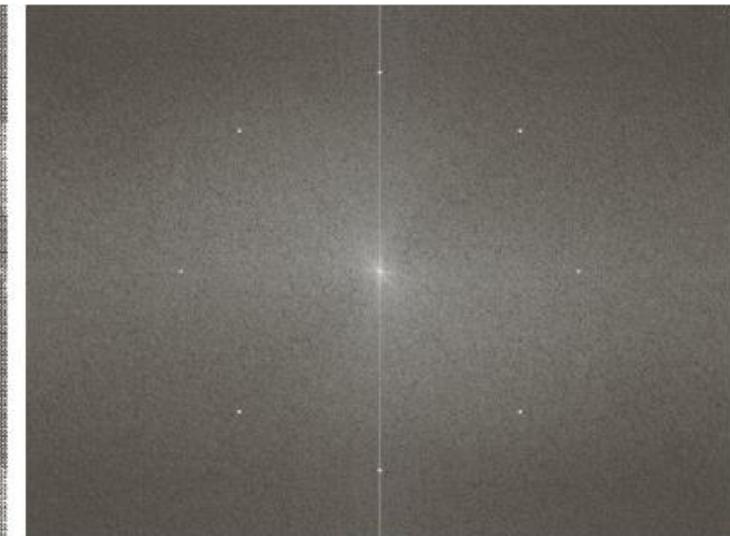
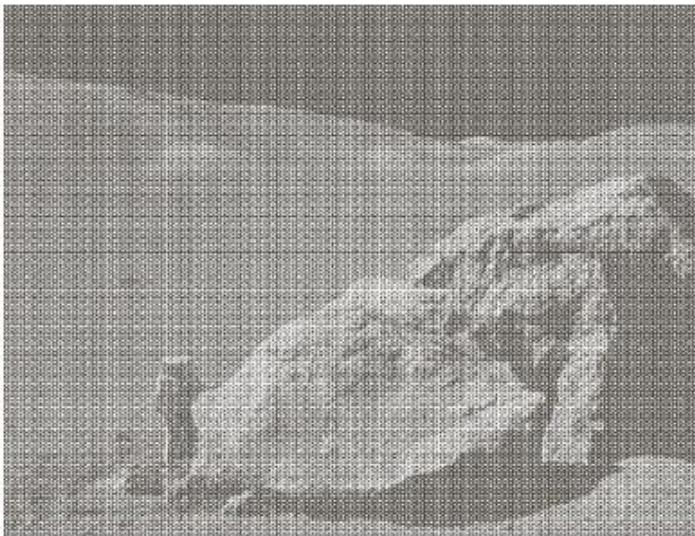
$$H_{bp}(u, v) = 1 - H_{br}(u, v)$$



a b  
c d

**FIGURE 5.16**

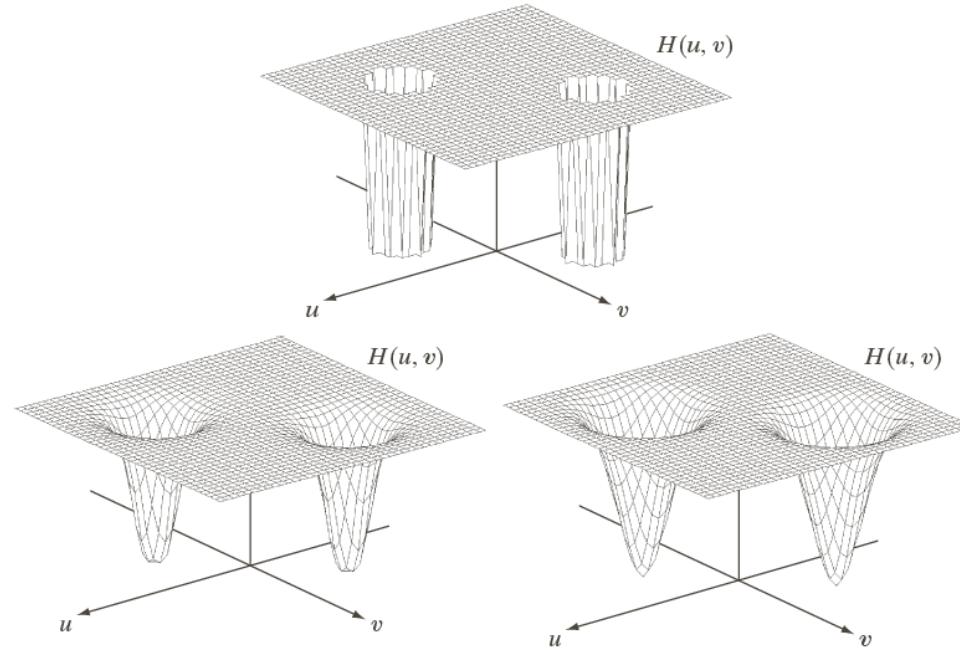
- (a) Image corrupted by sinusoidal noise.  
(b) Spectrum of (a).  
(c) Butterworth bandreject filter (white represents 1). (d) Result of filtering.  
(Original image courtesy of NASA.)

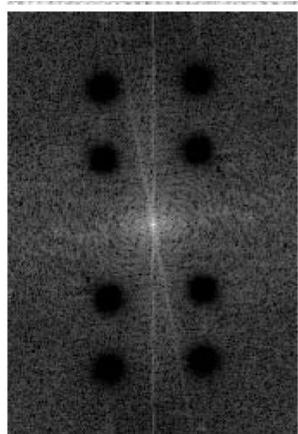
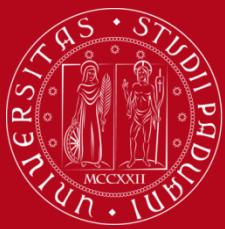




a  
b c

**FIGURE 5.18**  
Perspective plots  
of (a) ideal,  
(b) Butterworth  
(of order 2), and  
(c) Gaussian  
notch (reject)  
filters.





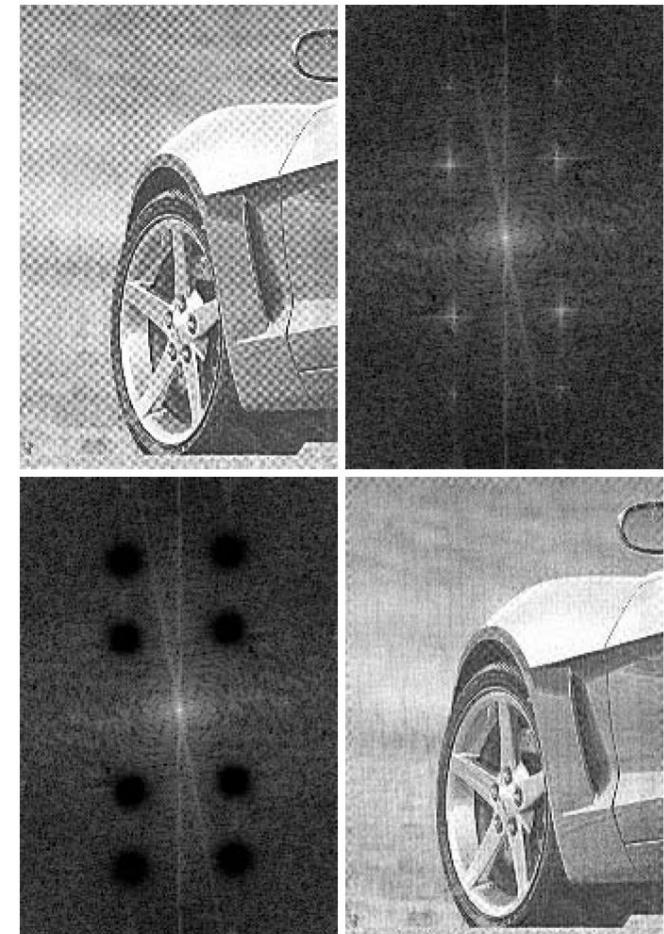
*Notch*

$$H_{NR}(u, v) = \prod_{k=1}^4 \left[ \frac{1}{1 + \left[ \frac{D_{0k}}{D_k(u, v)} \right]^{2n}} \right] \left[ \frac{1}{1 + \left[ \frac{D_{0k}}{D_{-k}(u, v)} \right]^{2n}} \right]$$

a b  
c d

**FIGURE 4.64**

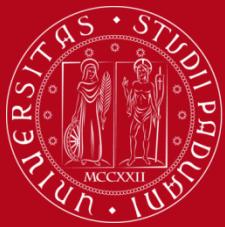
- (a) Sampled newspaper image showing a moiré pattern.
- (b) Spectrum.
- (c) Butterworth notch reject filter multiplied by the Fourier transform.
- (d) Filtered image.



# End of lecture 12, video 3

# Start of lecture 12, video 4

## More complex filters



- Very commonly used
- A spatial filtering!
- Recent method
  - Carlo Tomasi, Roberto Manduchi, “Bilateral Filtering for Gray and Color Images”, Proceedings of the ICCV 1998
- Many applications with high quality results
- Based on the gaussian filter
  - But: edge-preserving filter



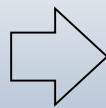
UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

# Gaussian filter example

IAS-LAB



*input*



*smoothed*  
*(structure, large scale)*



*residual*  
*(texture, small scale)*

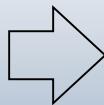
## Gaussian Convolution



# Bilateral filter example



*input*

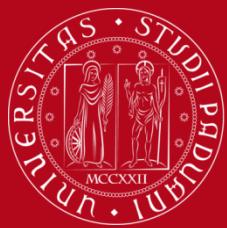


*smoothed*  
*(structure, large scale)*

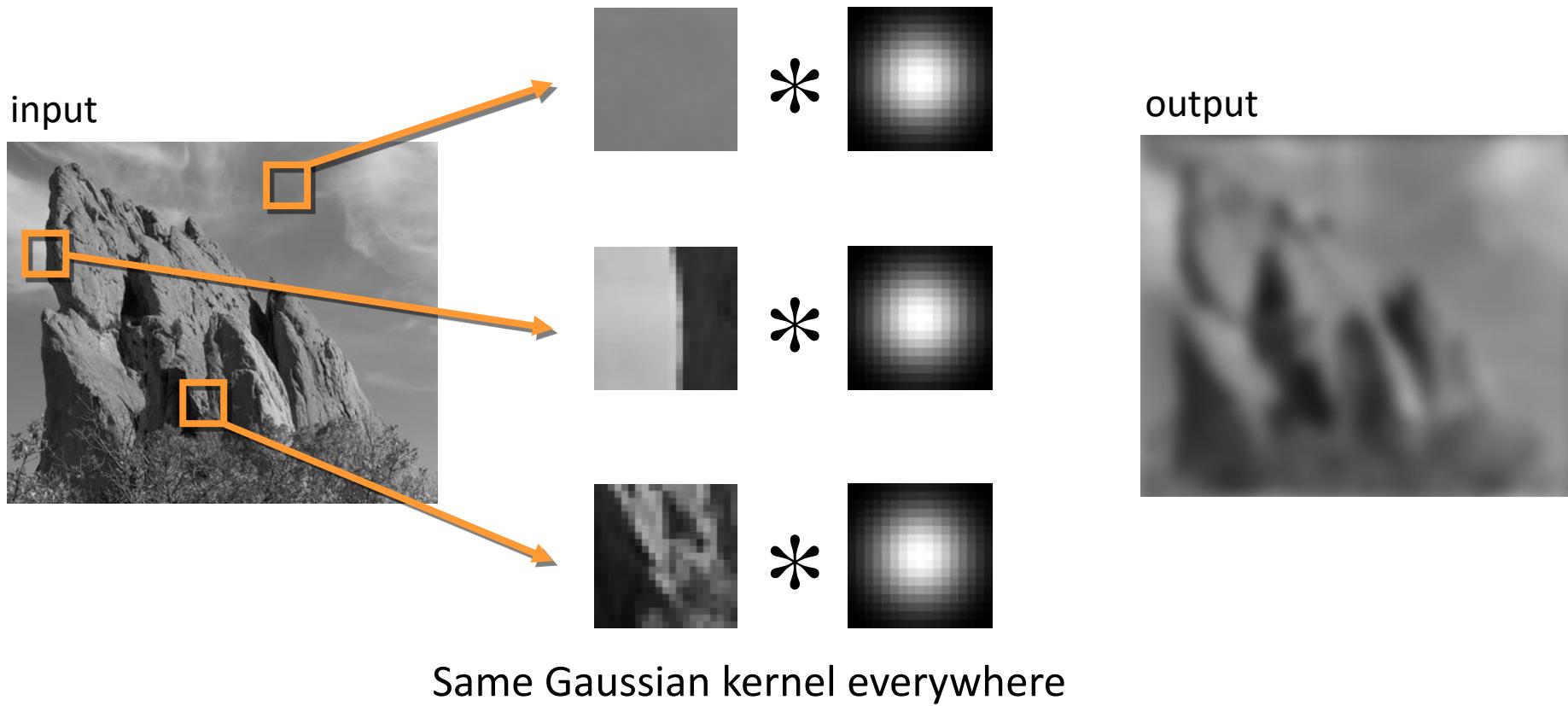


*residual*  
*(texture, small scale)*

**Edge-preserving: Bilateral Filter**

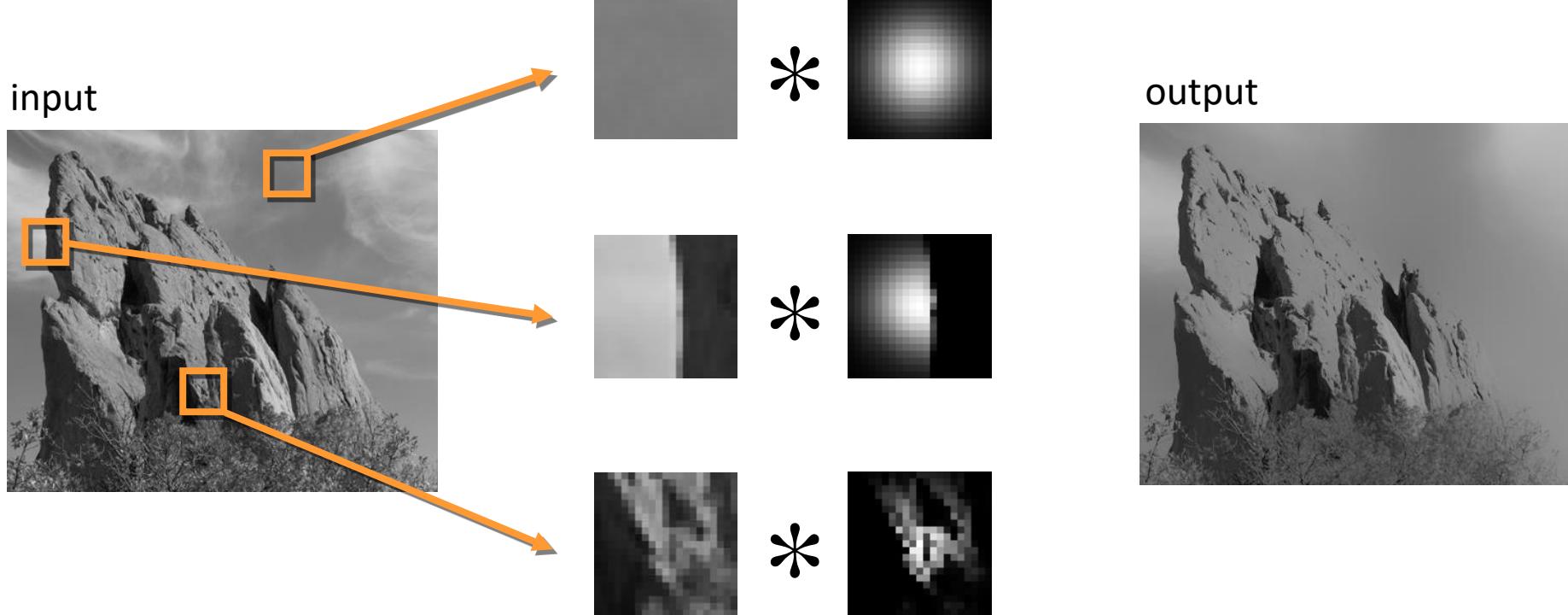


# Gaussian filter action



# Bilateral filter action

IAS-LAB



The kernel shape depends on the image content

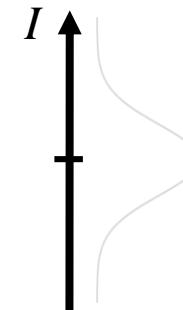
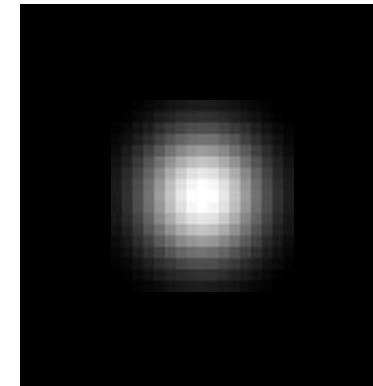
Same idea: **weighted average of pixels.**

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(||p - q||) G_{\sigma_r}(|I_p - I_q|) I_q$$

new                                    not new                            new

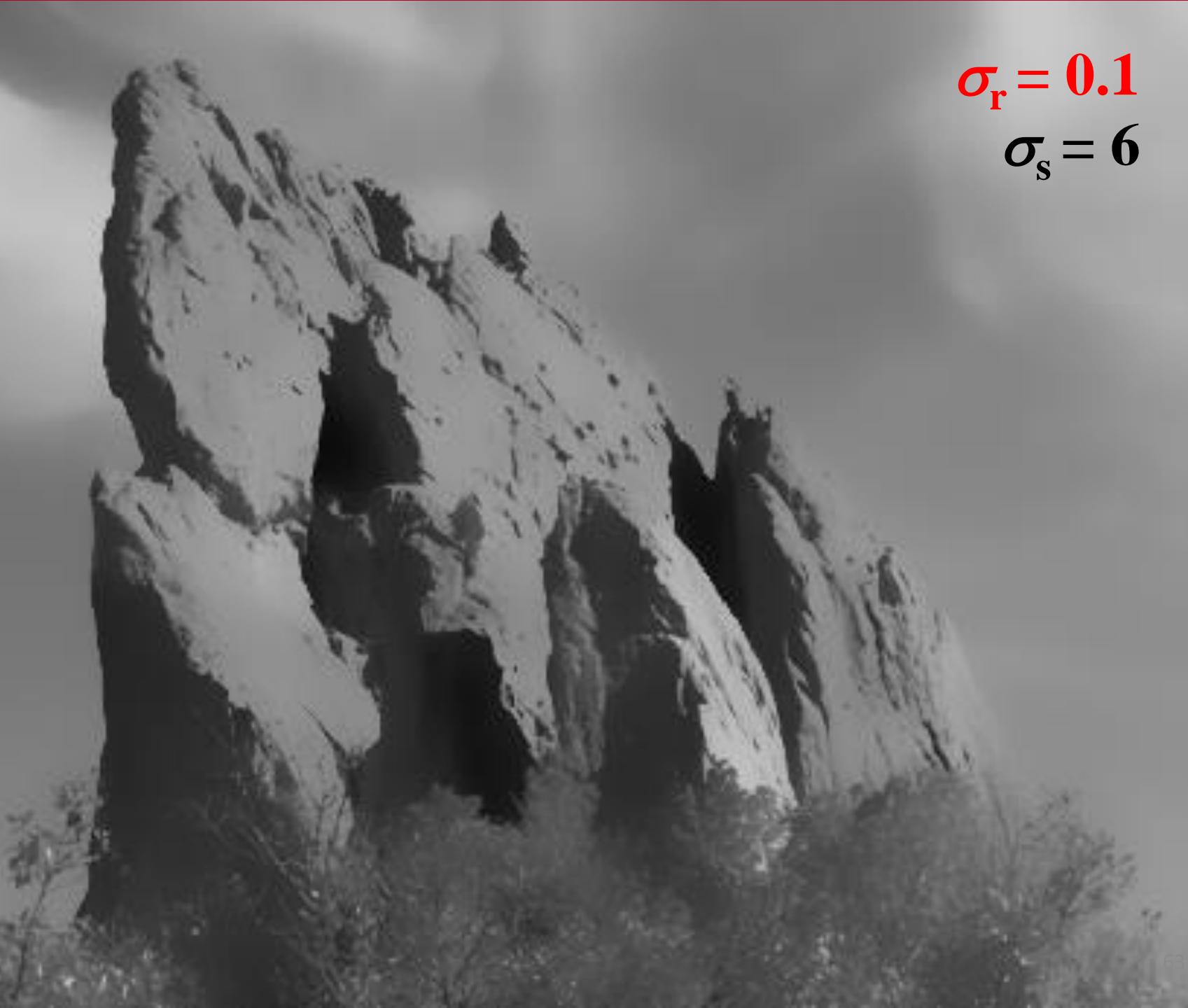
normalization factor                    *space* weight                            *range* weight



# Gaussian filter sequence

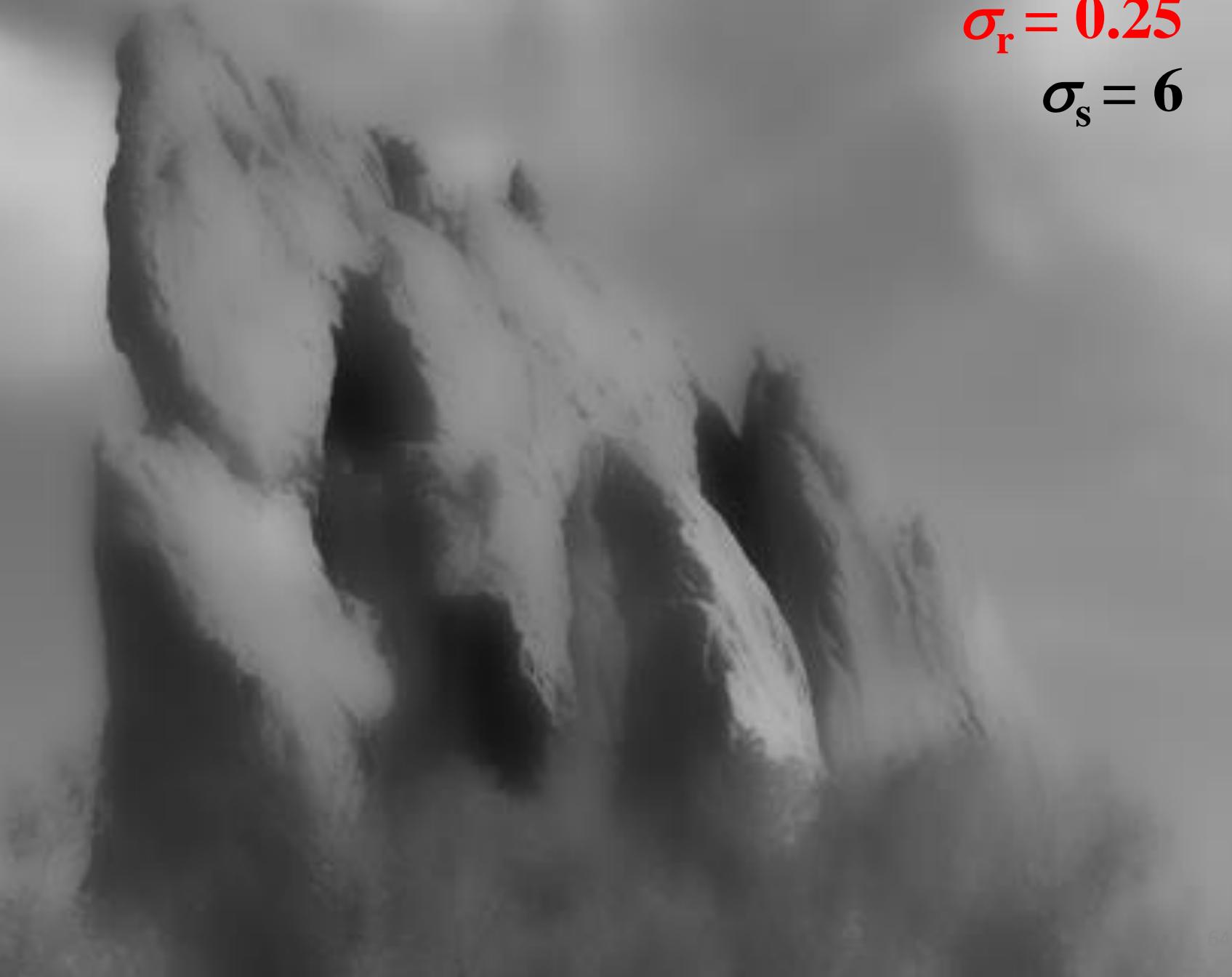
input





$\sigma_r = 0.1$

$\sigma_s = 6$



$\sigma_r = 0.25$

$\sigma_s = 6$

$\sigma_r = \infty$   
**(Gaussian blur)**

$\sigma_s = 6$

# Bilateral filter sequence

input





$\sigma_r = 0.1$

$\sigma_s = 2$



$\sigma_r = 0.1$

$\sigma_s = 6$



$\sigma_r = 0.1$

$\sigma_s = 18$

# Denoising: an example



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

# Basic denoising

IAS-LAB

Noisy input



Bilateral filter 7x7 window





UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

# Basic denoising

IAS-LAB

Bilateral filter



Median 3x3





UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

# Basic denoising

IAS-LAB

Bilateral filter



Median 5x5





UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

# Basic denoising

IAS-LAB

Bilateral filter



Bilateral filter – lower sigma





UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

# Basic denoising

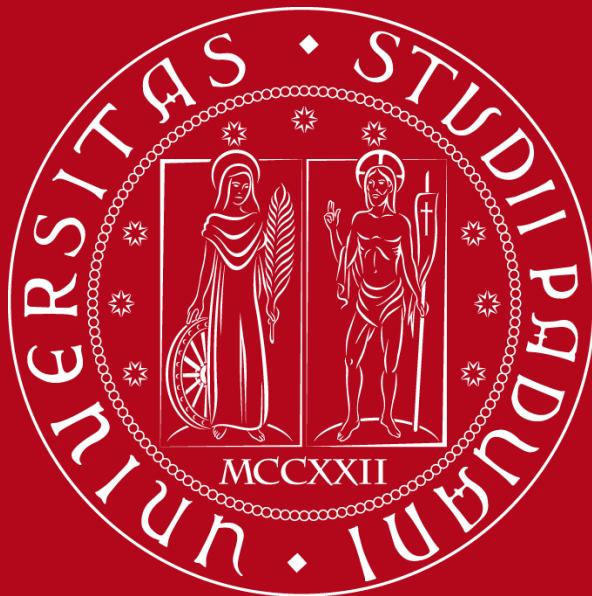
IAS-LAB

Bilateral filter



Bilateral filter – higher sigma





UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

## Filtering in the frequency domain

Stefano Ghidoni



DIPARTIMENTO  
DI INGEGNERIA  
DELL'INFORMAZIONE

INTELLIGENT AUTONOMOUS SYSTEMS LAB

