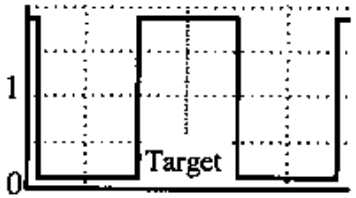
Filtering – Part II

Sedat OZER
Department of Computer Engineering
Bilkent University
sedat@cs.bilkent.edu.tr

Fourier theory

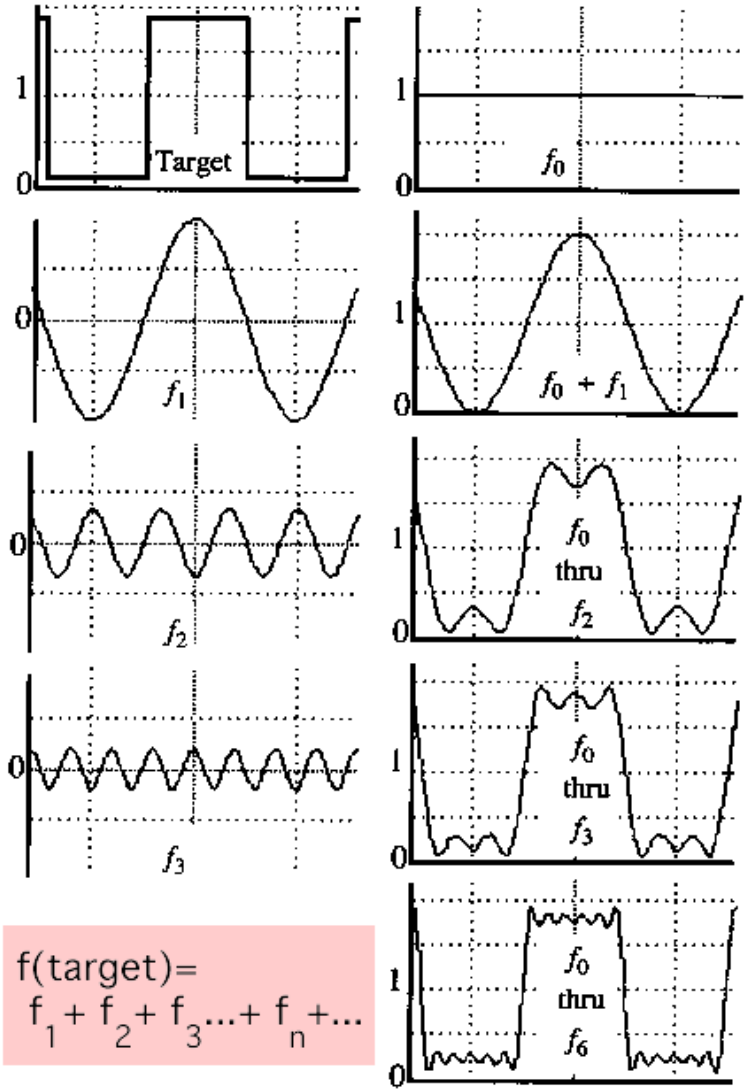
- The Fourier theory shows how most real functions can be represented in terms of a basis of sinusoids.
- The building block:
  - $A \sin(\omega x + \Phi)$
- Add enough of them to get any signal you want.



Adapted from Alexei Efros, CMU

Fourier theory

- The Fourier theory shows how most real functions can be represented in terms of a basis of sinusoids.
- The building block:
  - A sin(ωx + Φ)
- Add enough of them to get any signal you want.



Adapted from Alexei Efros, CMU

Fourier transform

- The *Fourier transform*, $F(u)$, of a single variable, continuous function, $f(x)$, is defined by

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

- Given $F(u)$, we can obtain $f(x)$ using the *inverse Fourier transform*

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

Fourier transform

- The *discrete Fourier transform (DFT)*, $F(u)$, of a discrete function of one variable, $f(x)$, $x = 0, 1, 2, \ldots, M-1$, is defined by

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

for $u = 0, 1, 2, \ldots, M-1$.

- Given $F(u)$, we can obtain the original function back using the *inverse DFT*

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

for $x = 0, 1, 2, \ldots, M-1$.

Fourier transform

- These formulas can be extended for functions of two variables.
- Fourier transform:

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

- Inverse Fourier transform:

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

Fourier transform

- Discrete Fourier transform:

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

for $u = 0, 1, 2, \ldots, M-1,\ v = 0, 1, 2, \ldots, N-1.$

- Inverse discrete Fourier transform:

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

for $x = 0, 1, 2, \ldots, M-1,\ y = 0, 1, 2, \ldots, N-1.$

Fourier transform

- $F(u,v)$ can also be expressed in polar coordinates as

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

where

$$|F(u,v)| = \left(\Re^2\{F(u,v)\} + \Im^2\{F(u,v)\}\right)^{1/2}$$

is called the *magnitude* or *spectrum* of the Fourier transform, and

$$\phi(u,v) = \tan^{-1}\left(\frac{\Im\{F(u,v)\}}{\Re\{F(u,v)\}}\right)$$

is called the *phase angle* or *phase spectrum*.

- $\Re\{F(u,v)\}$ and $\Im\{F(u,v)\}$ are the real and imaginary parts of $F(u,v)$, respectively.

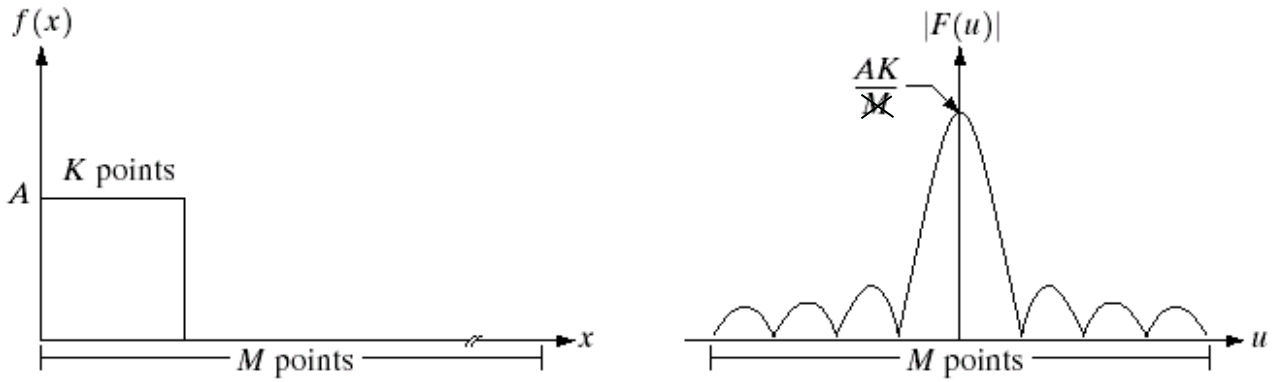
Fourier transform

- The spectrum need not be interpreted as an image, but rather as a 2D display of the power in the original image versus the frequency components u and v.
- The value $F(0,0)$ is the average of $f(x,y)$.
- Fourier transform is conjugate symmetric ($F(u,v) = F^*(-u,-v)$) and its spectrum is symmetric about the origin ($|F(u,v)| = |F(-u,-v)|$) (when $f(x,y)$ is real).
- Usually the input image function is multiplied by $(-1)^{x+y}$ prior to computing the Fourier transform so that

$$\mathfrak{F}[f(x,y)\ (-1)^{x+y}] = F(u - M/2, v - N/2).$$

The origin of the Fourier transform is located at $u = M/2$ and $v = N/2$.

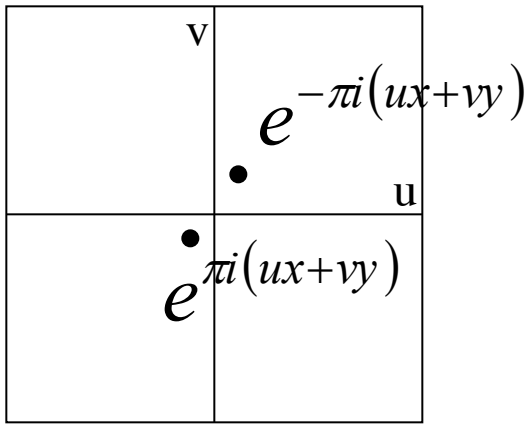
Fourier transform - matlab

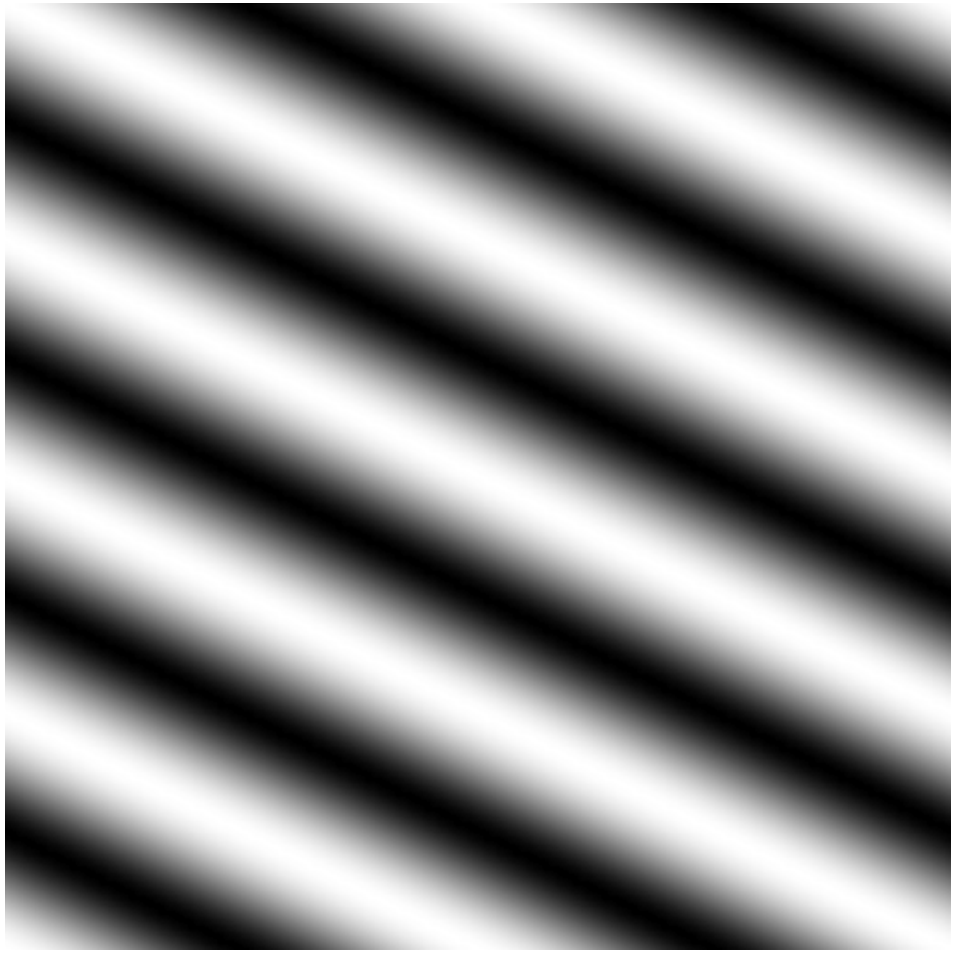



```
A=1;  K=10; M=100;
t=[ones(1,K)*A  zeros(1,M-K)];
subplot(3,1,1);  bar(t);  ylim([0  2*A])
subplot(3,1,2); bar(abs(fftshift(fft(t)))); ylim([0 A*K+1])
% Matlab uses DFT formulation without normalization by M.
subplot(3,1,3); bar(real(fftshift(fft(t)))); ylim([-A*K+1 A*K+1])
```

Fourier transform

To get some sense of what basis elements look like, we plot a basis element --- or rather, its real part --- as a function of x,y for some fixed u, v. We get a function that is constant when (ux+vy) is constant. The magnitude of the vector (u, v) gives a frequency, and its direction gives an orientation. The function is a sinusoid with this frequency along the direction, and constant perpendicular to the direction.

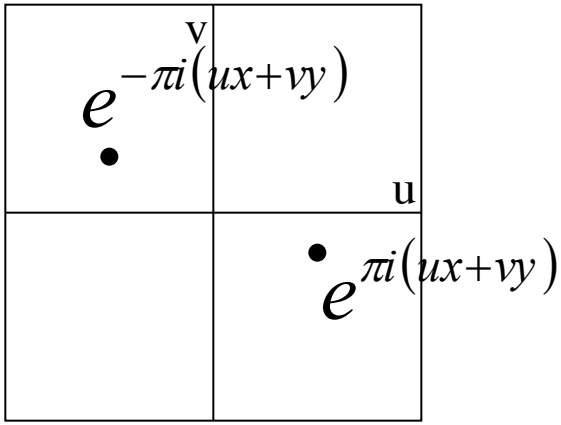




Adapted from Antonio Torralba

Fourier transform

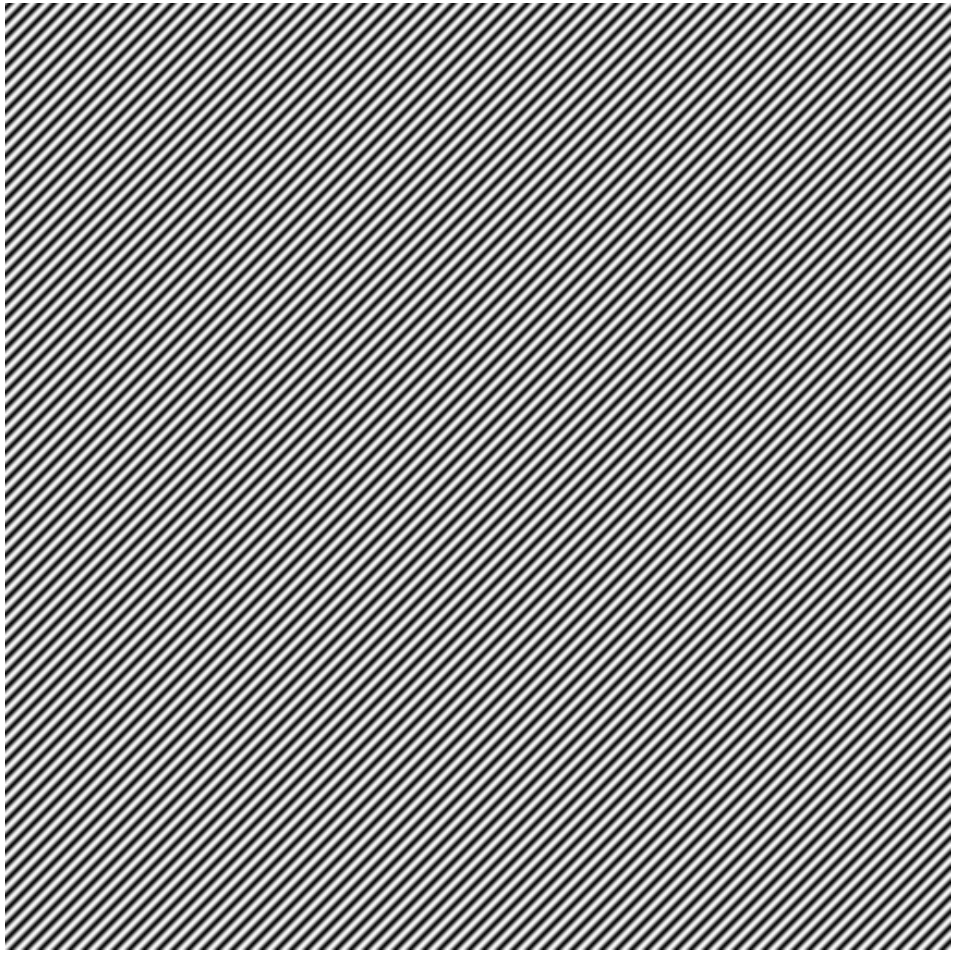
Here u and v are larger than in the previous slide.

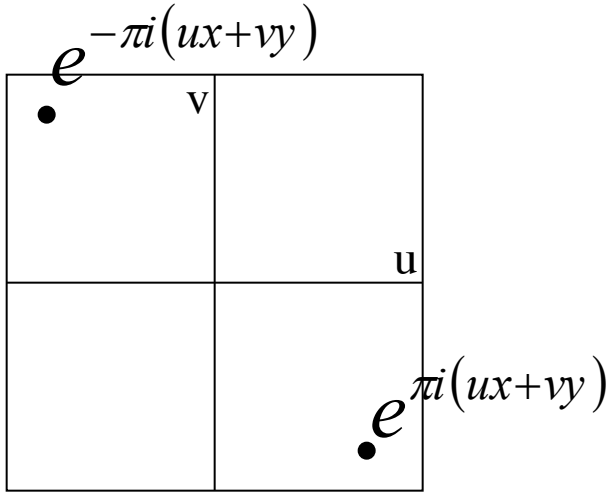


Adapted from Antonio Torralba

Fourier transform

And larger still...





Adapted from Antonio Torralba

Fourier transform

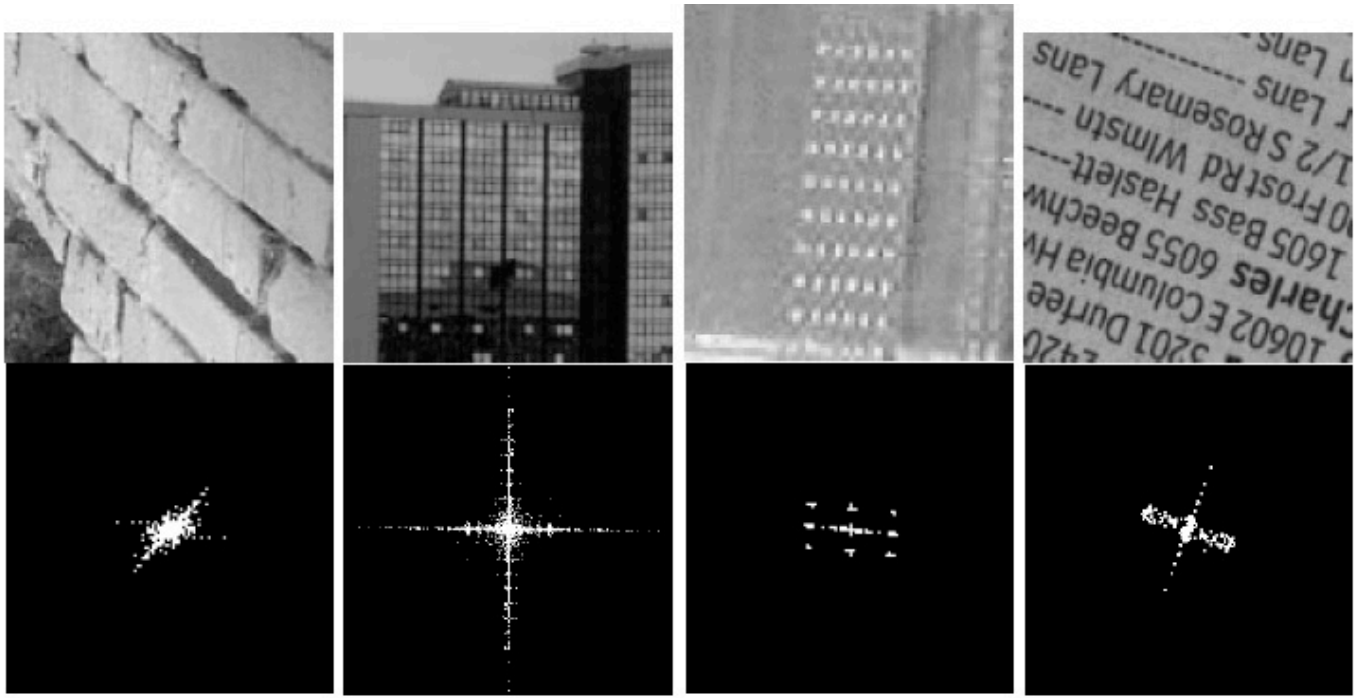


Figure 5.42: Four images (above) and their power spectrums (below). The power spectrum of the brick texture shows energy in many sinusoids of many frequencies, but the dominant direction is perpendicular to the 6 dark seams running about 45 degrees with the X-axis. There is noticable energy at 0 degrees with the X axis, due to the several short vertical seams. The power spectrum of the building shows high frequency energy in waves along the X-direction and the Y-direction. The third image is an aerial image of an orchard: the power spectrum shows the rows and columns of the orchard and also the "diagnonal rows". The far right image, taken from a phone book, shows high frequency power at about 60^o with the X-axis, which represents the texture in the lines of text. Energy is spread more broadly in the perpendicular direction also in order to model the characters and their spacing.

Adapted from Shapiro and Stockman

Convolution theorem

- The discrete *convolution* of two functions $f(x, y)$ and $h(x, y)$ of size $M \times N$ is defined as

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

- This is equivalent to the *correlation* of $f(x, y)$ with $h(x, y)$ flipped about the origin.
- Convolution theorem:

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

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

where "$\Leftrightarrow$" indicates a Fourier transform pair.

Frequency domain filtering

Filter image $f(x, y)$ with mask $h(x, y)$

(1) Fourier transform the image $f(x, y)$ to obtain its frequency rep. $F(u, v)$.
(2) Fourier transform the mask $h(x, y)$ to obtain its frequency rep. $H(u, v)$
(3) multiply $F(u, v)$ and $H(u, v)$ pointwise to obtain $F'(u, v)$
(4) apply the inverse Fourier transform to $F'(u, v)$ to obtain the filtered image $f'(x, y)$.

Algorithm 3: Filtering image $f(x, y)$ with mask $h(x, y)$ using the Fourier transform

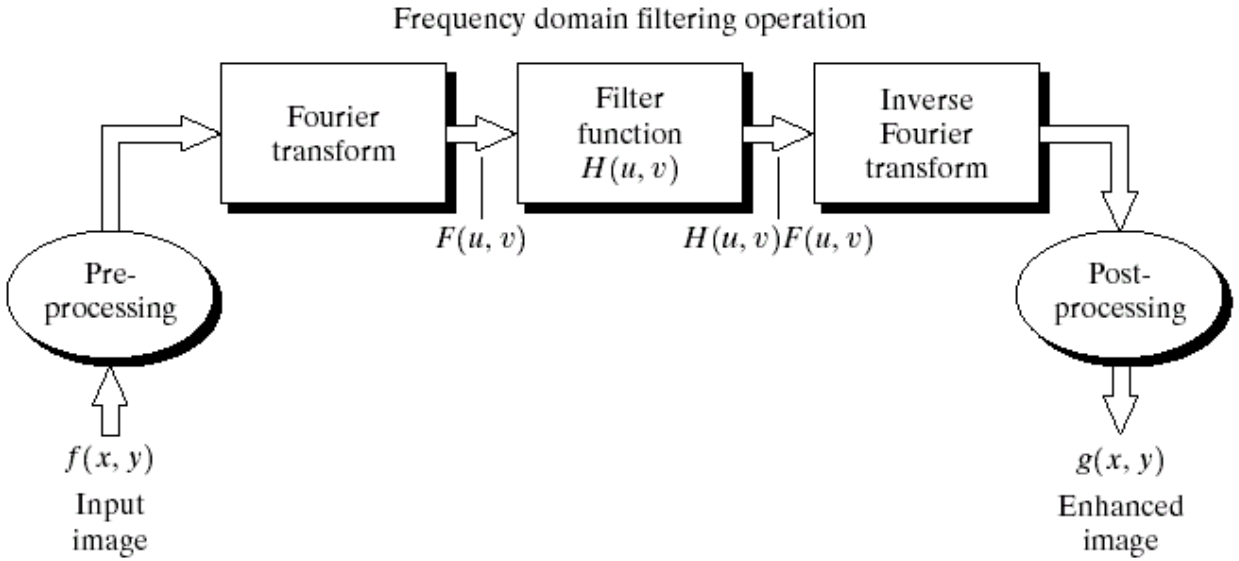
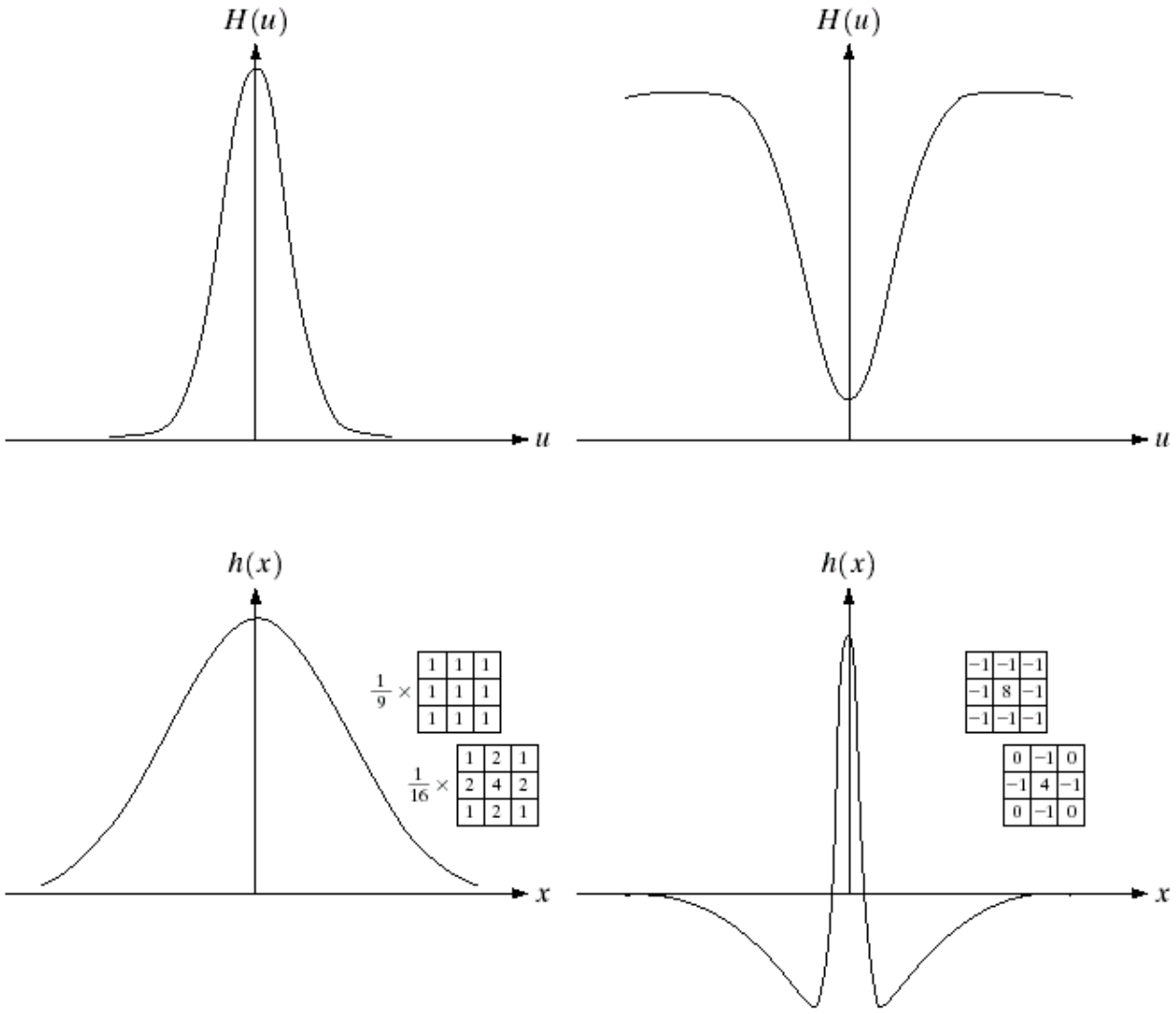


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Adapted from Shapiro and Stockman, and Gonzales and Woods

Frequency domain filtering



a b
c d

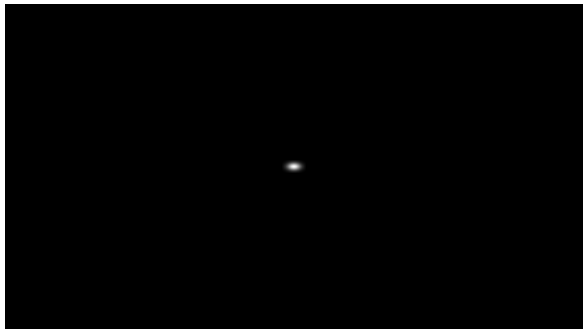
FIGURE 4.9
(a) Gaussian frequency domain lowpass filter.
(b) Gaussian frequency domain highpass filter.
(c) Corresponding lowpass spatial filter.
(d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

Frequency domain filtering

f(x,y)

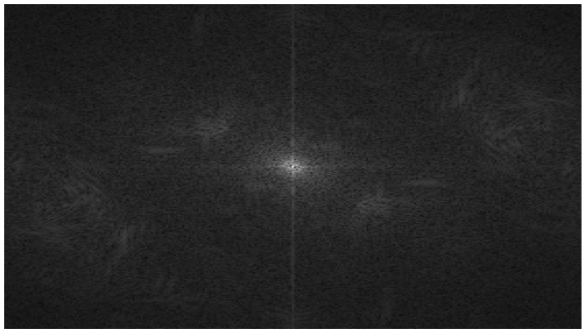
★

h(x,y)



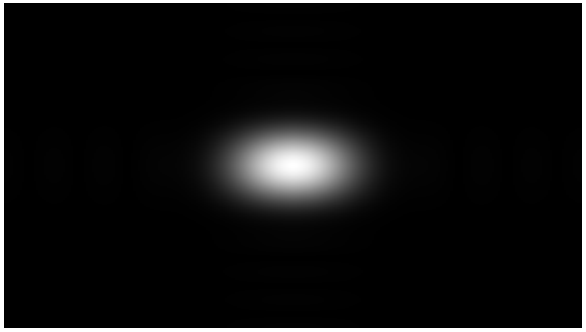
⇓

g(x,y)



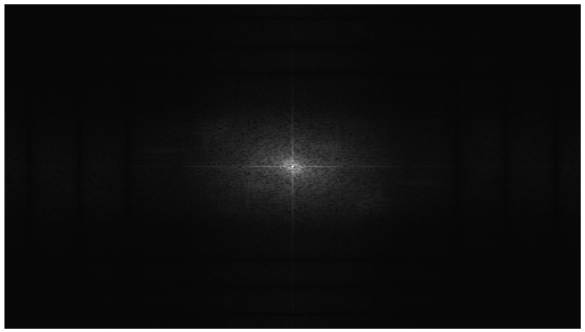
|F(u,v)|

×



|H(u,v)|

⇓



|G(u,v)|

Adapted from Alexei Efros, CMU

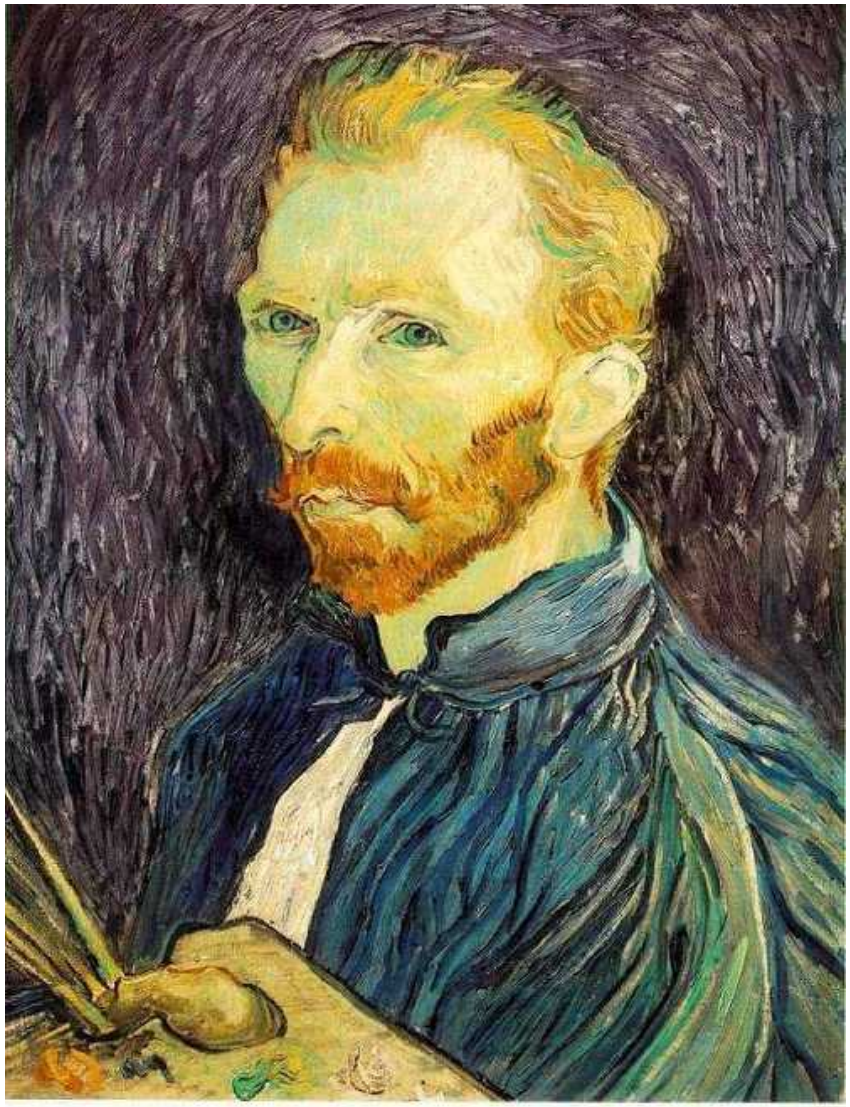
Template matching

- Correlation can also be used for matching.
- If we want to determine whether an image f contains a particular object, we let h be that object (also called a template) and compute the correlation between f and h.
- If there is a match, the correlation will be maximum at the location where h finds a correspondence in f.
- Preprocessing such as scaling and alignment is necessary in most practical applications.

Template matching

Face detection using template matching: detected faces.

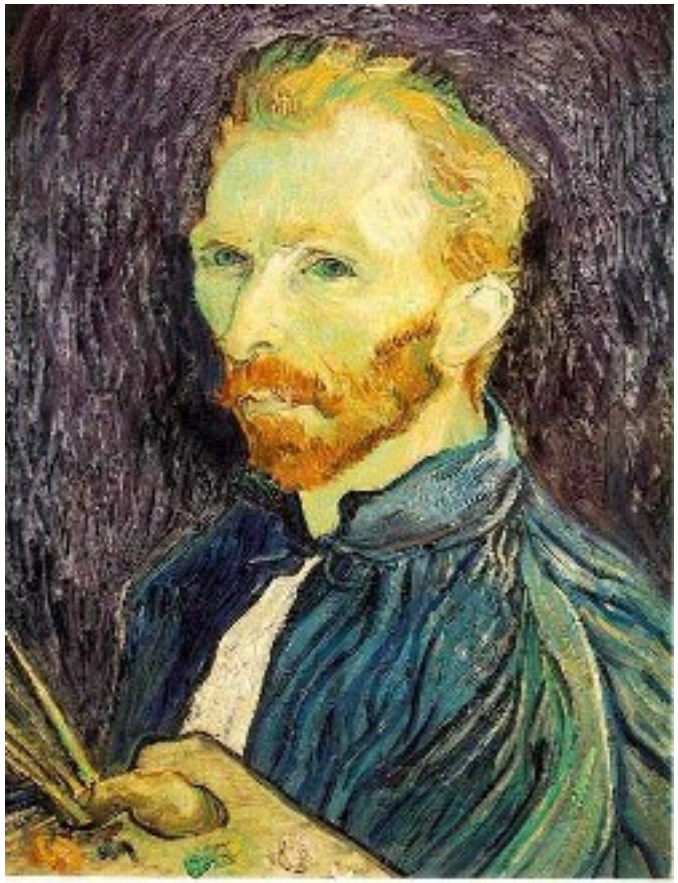
Resizing images



How can we generate a half-sized version of a large image?

Adapted from Steve Seitz, U of Washington

Resizing images



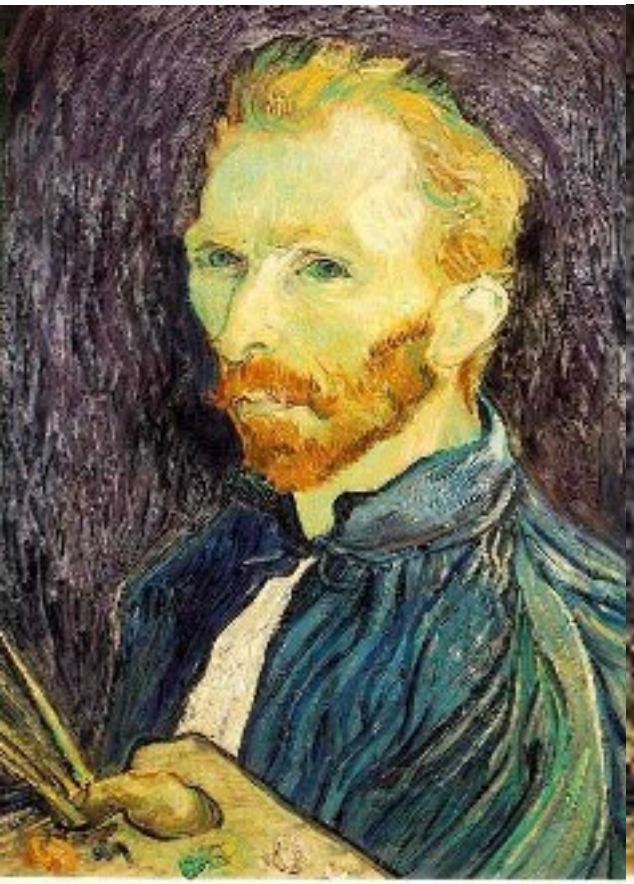
1/4

1/8

Throw away every other row and column to create a 1/2 size image (also called sub-sampling).

Adapted from Steve Seitz, U of Washington

Resizing images



1/2

1/4 (2x zoom)

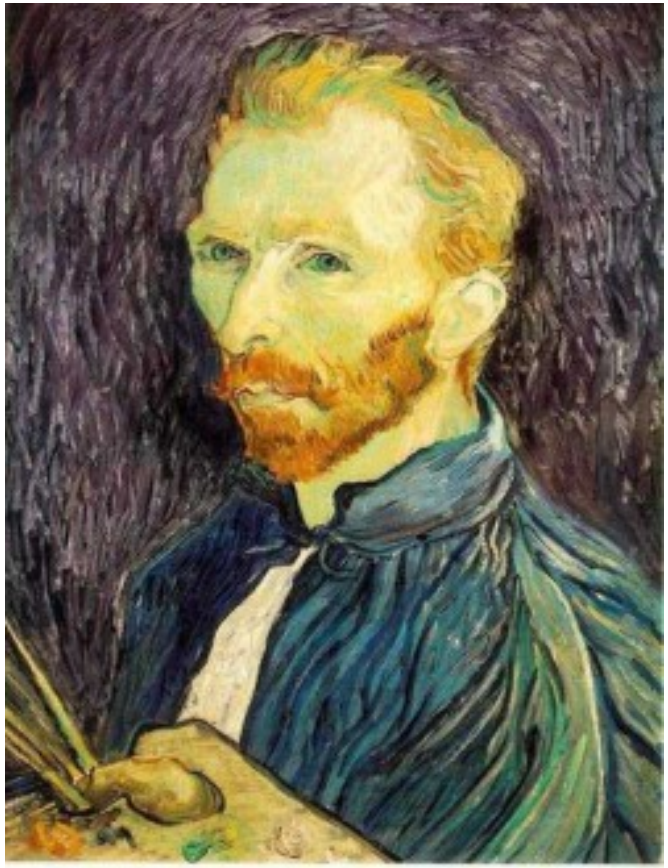
1/8 (4x zoom)

Does this look nice?

Adapted from Steve Seitz, U of Washington

Resizing images

- We cannot shrink an image by simply taking every k'th pixel.
- Solution: smooth the image, then sub-sample.



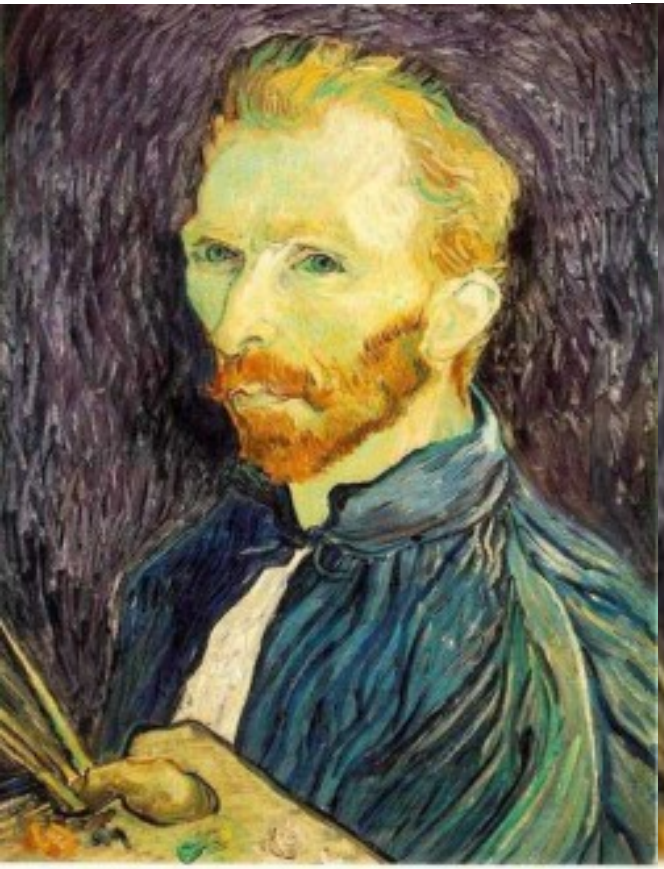
Gaussian 1/2

Gaussian 1/4

Gaussian 1/8

Adapted from Steve Seitz, U of Washington

Resizing images



Gaussian 1/2

Gaussian 1/4
(2x zoom)

Gaussian 1/8
(4x zoom)

Adapted from Steve Seitz, U of Washington

Sampling and aliasing

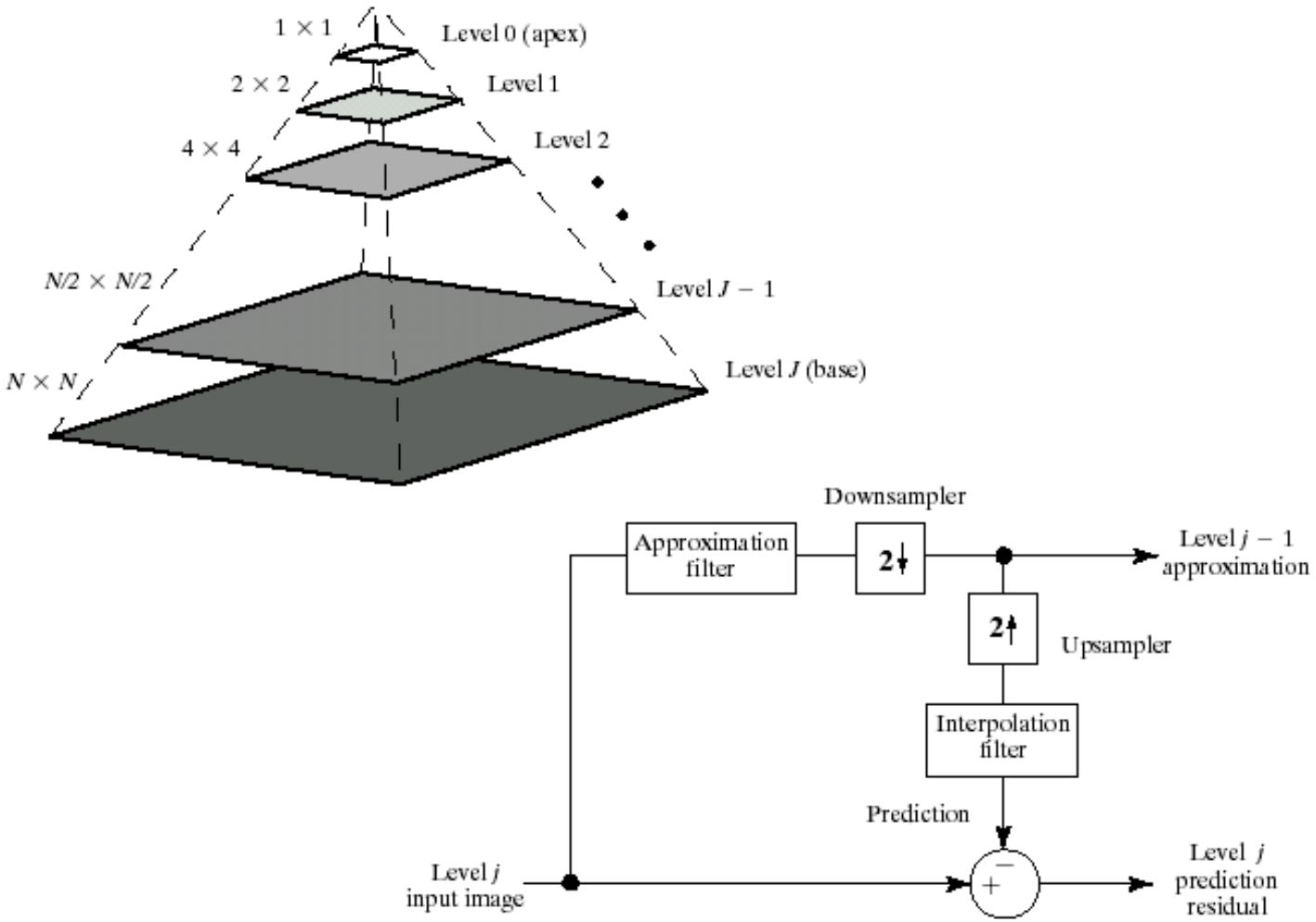
- Errors appear if we do not sample properly.
- Common phenomenon:
  - High spatial frequency components of the image appear as low spatial frequency components.
- Examples:
  - Wagon wheels rolling the wrong way in movies.
  - Checkerboards misrepresented in ray tracing.
  - Striped shirts look funny on color television.

Sampling and aliasing

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 Ali Farhadi

Gaussian pyramids

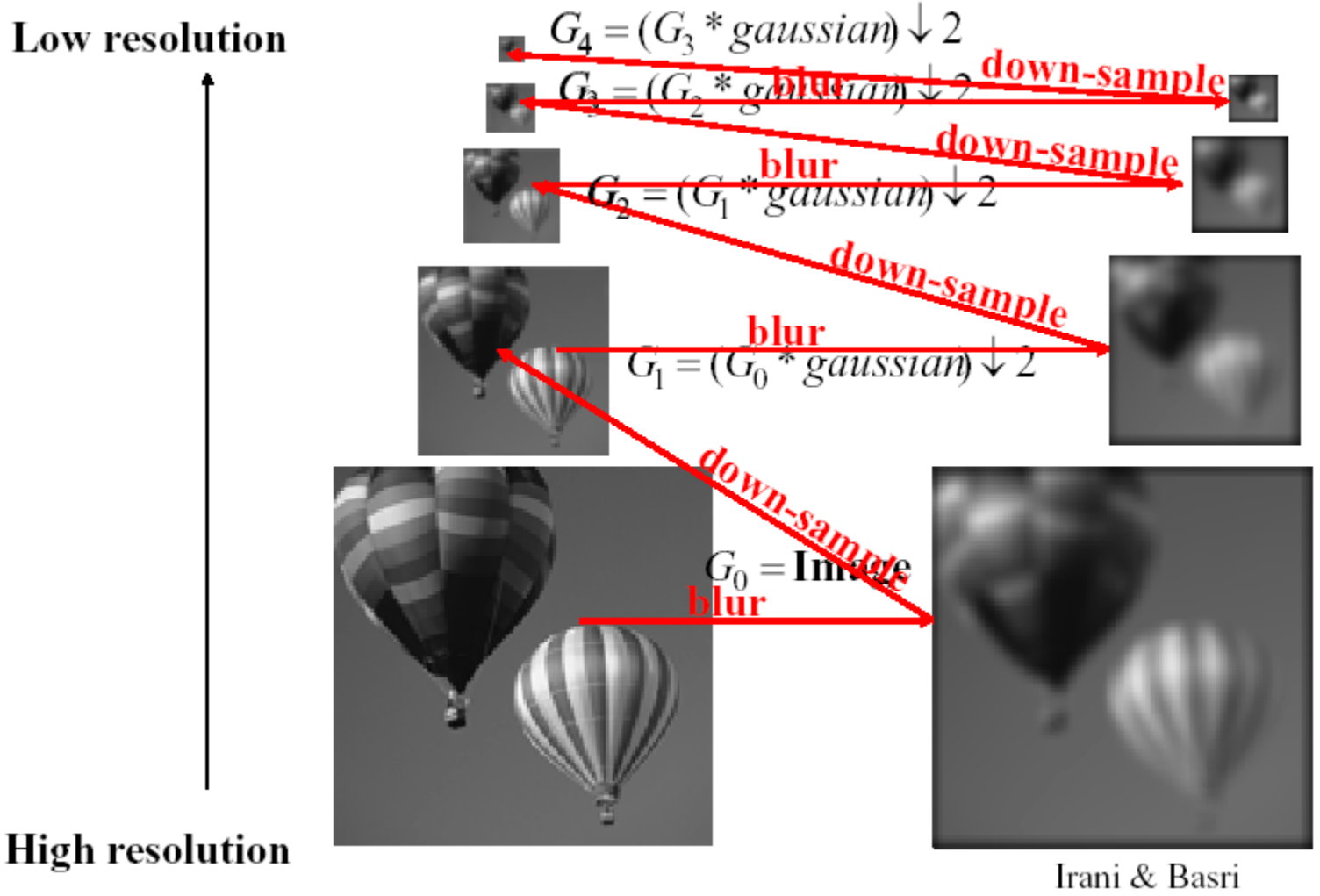


a
b

FIGURE 7.2 (a) A pyramidal image structure and (b) system block diagram for creating it.

Adapted from Gonzales and Woods

Gaussian pyramids



Adapted from Michael Black, Brown University