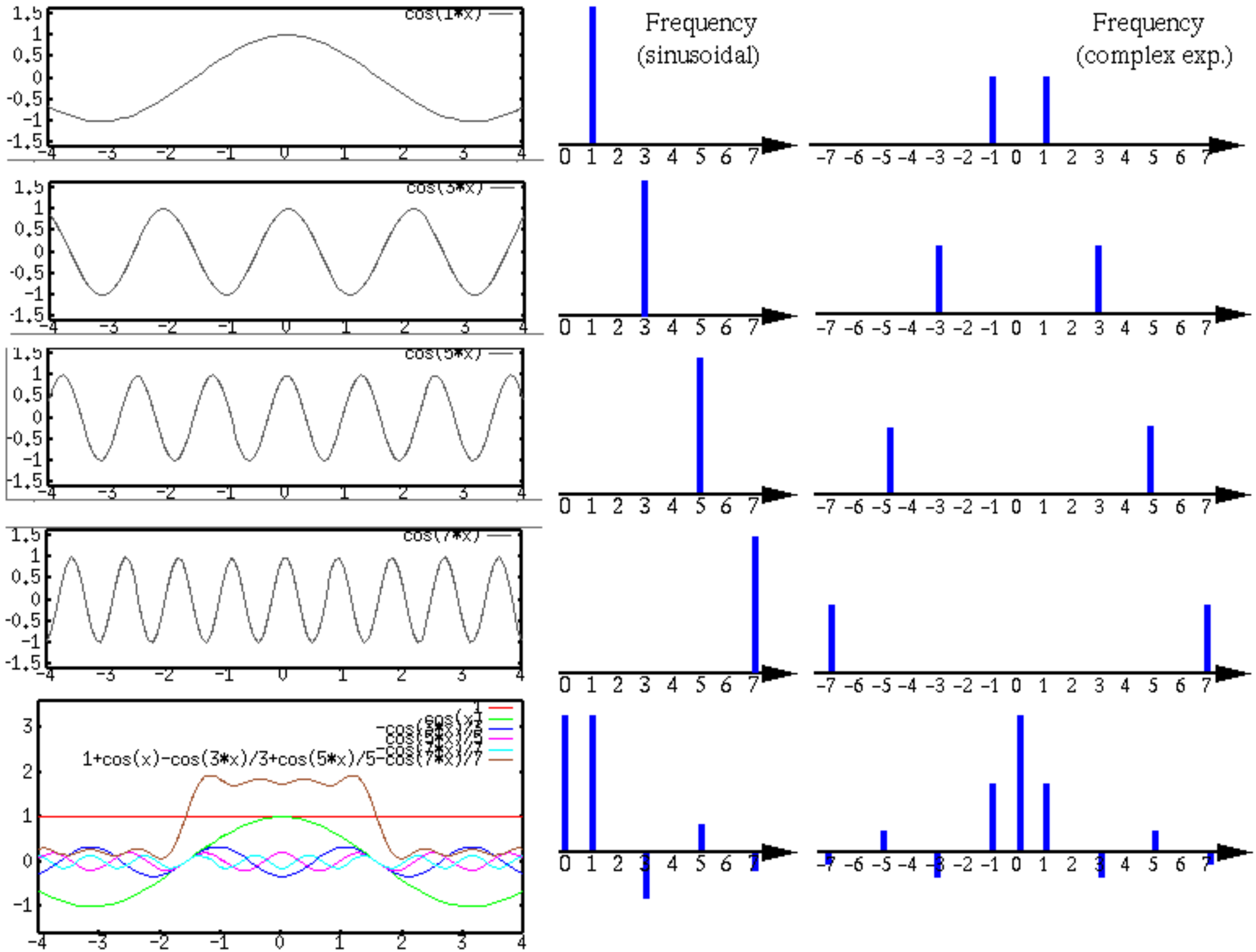
CS4670/5670: Computer Vision

Kavita Bala

Lecture 5: Fourier and Pyramids

Announcements

- PA1-A will be out tonight, due in a week
  - Written part alone
  - Coding part in pairs
- Monday: numPy lecture



Frequency
(sinusoidal)
Frequency
(complex exp.)
cos(1*x)
cos(3*x)
cos(5*x)
cos(7*x)
1+cos(x)-cos(3*x)/3+cos(5*x)/5-cos(7*x)/7

Fourier Transform

- Given a signal compute (co)sine waves that contribute to signal
  - For each frequency k, contribution of that frequency to signal: X[k]

$$x_T(t) = \sum_{k=0}^{\infty} X'[k] cos(k\omega_0 t) = \sum_{k=-\infty}^{\infty} X[k] e^{jk\omega_0 t}$$

$$X[k] = \frac{1}{T} \int_T x_T(t) e^{-jk\omega_0 t} dt = \frac{1}{T} \int_T x_T(t) e^{-j2\pi k f_0 t} dt$$

Fourier Transform

- Fourier transform stores the magnitude and phase at each frequency
  - Magnitude encodes how much signal there is at a particular frequency
  - Phase encodes spatial information (indirectly)
  - For mathematical convenience, this is often notated in terms of real and complex numbers

Amplitude: $A = \pm\sqrt{R(\omega)^2 + I(\omega)^2}$

Phase: $\phi = \tan^{-1}\frac{I(\omega)}{R(\omega)}$

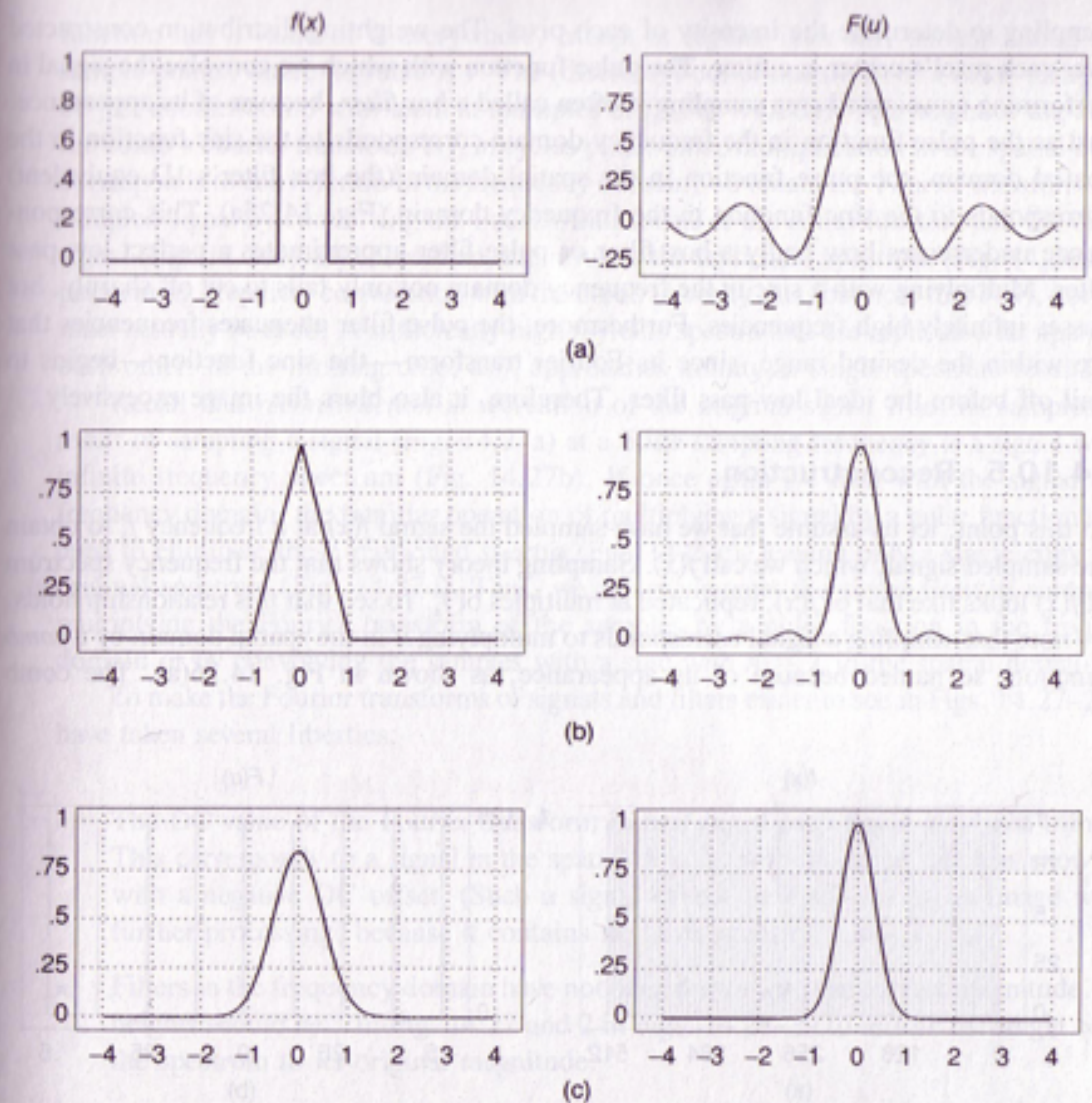


Fig. 14.25 Filters in spatial and frequency domains. (a) Pulse—sinc. (b) Triangle—sinc^2. (c) Gaussian—Gaussian. (Courtesy of George Wolberg, Columbia University.)

Derivation for Gaussian

http://mathworld.wolfram.com/FourierTransformGaussian.html

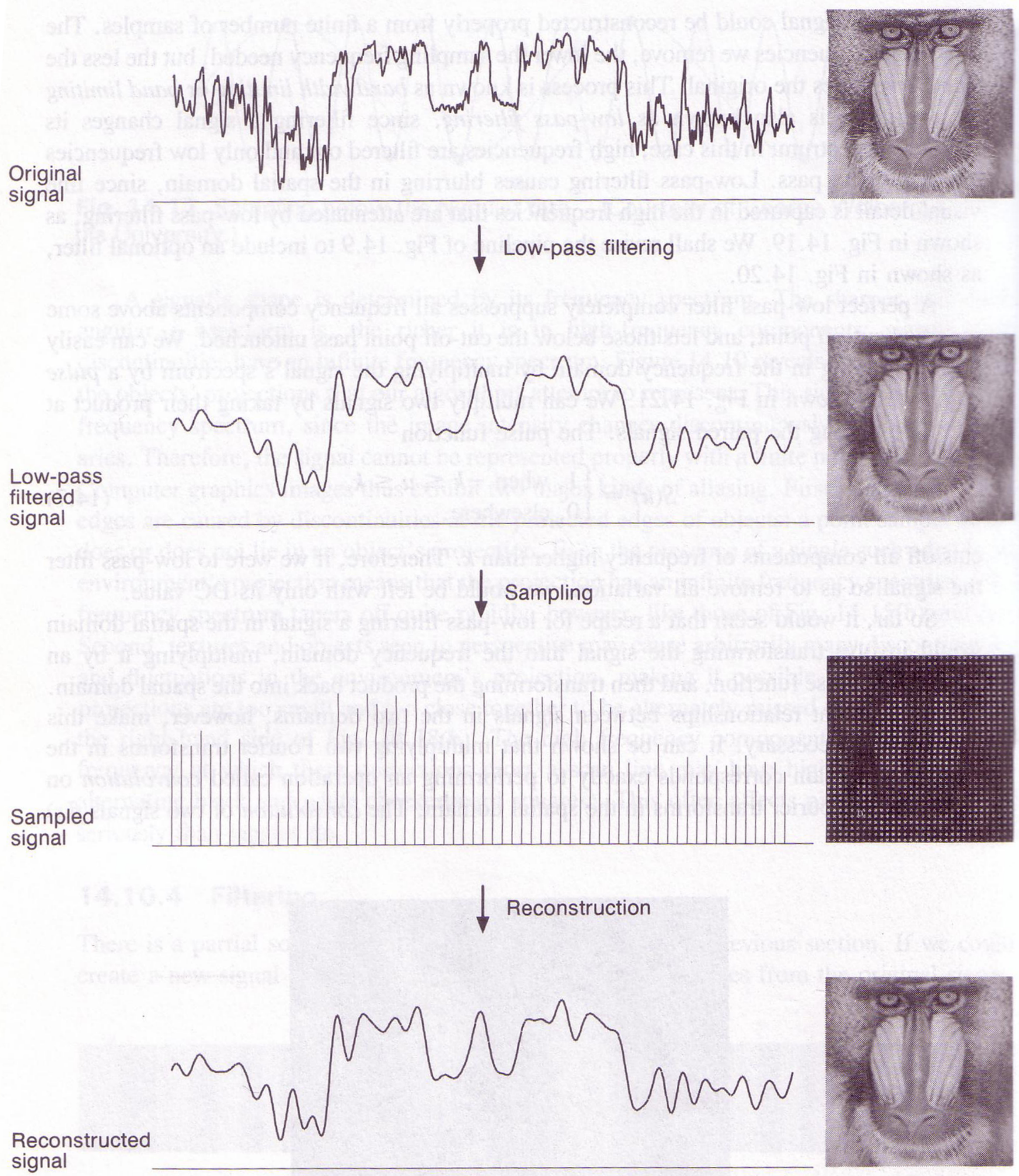
The Fourier transform of a Gaussian function $f(x) \equiv e^{-a x^2}$ is given by

$$\begin{aligned}\mathcal{F}_x\left[e^{-a x^2}\right](k) &= \int_{-\infty}^{\infty} e^{-a x^2}\, e^{-2\pi i k x}\, d x \\ &= \int_{-\infty}^{\infty} e^{-a x^2}\, [\cos(2\pi k x) - i \sin(2\pi k x)]\, d x \\ &= \int_{-\infty}^{\infty} e^{-a x^2} \cos(2\pi k x)\, d x - i \int_{-\infty}^{\infty} e^{-a x^2} \sin(2\pi k x)\, d x.\end{aligned}$$

The second integrand is odd, so integration over a symmetrical range gives 0. The value of the first integral is given by Abramowitz and Stegun (1972, p. 302, equation 7.4.6), so

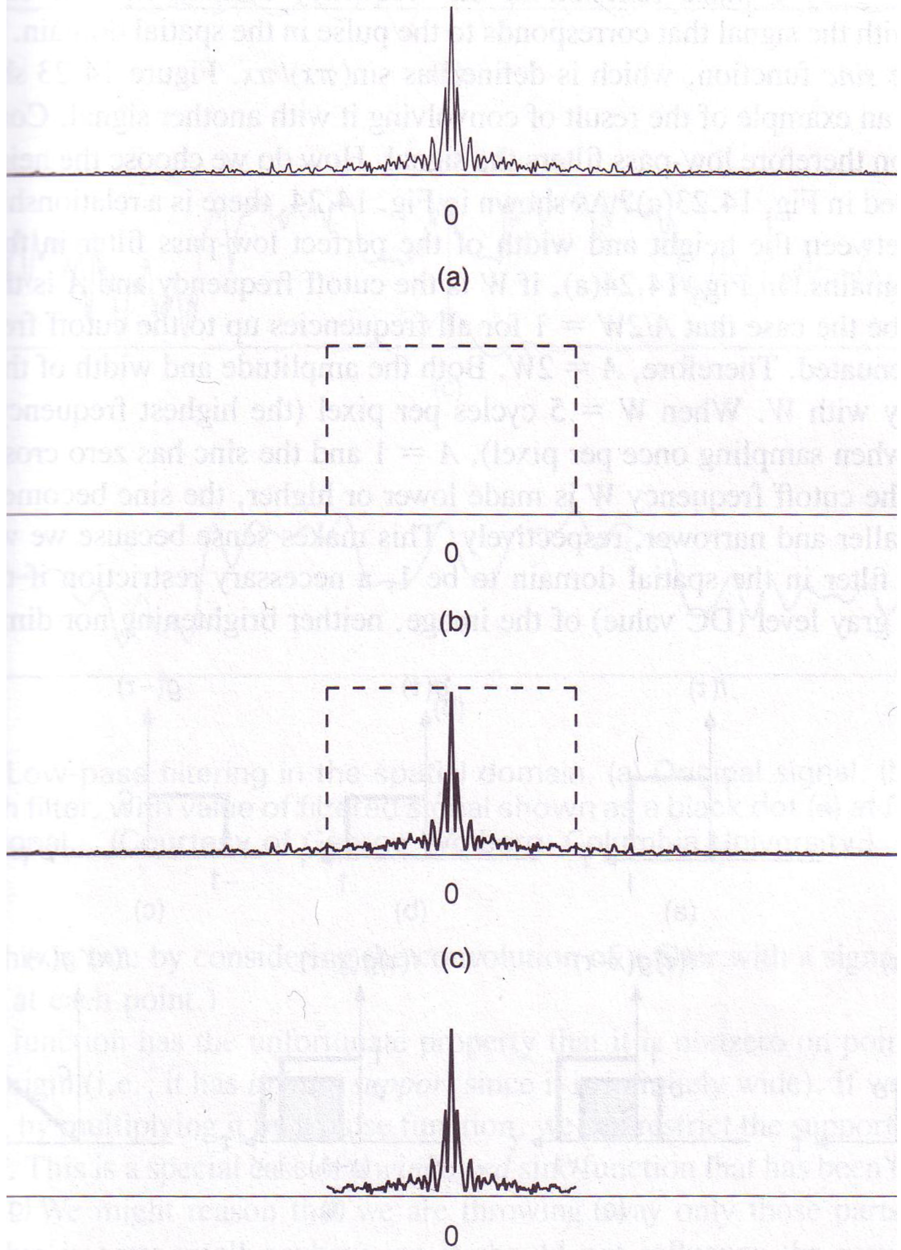
$$\mathcal{F}_x\left[e^{-a x^2}\right](k) = \sqrt{\frac{\pi}{a}}\, e^{-\pi^2 k^2/a},$$

so a Gaussian transforms to another Gaussian.


Original signal
Low-pass filtering
Low-pass filtered signal
Sampling
Sampled signal
Reconstruction
Reconstructed signal

Process

- Convert to frequency domain
- Multiply by low pass filter
  - Eliminate high frequencies
- Sample
- Reconstruct


0
(a)
0
(b)
0
(c)
0

Filtering

- Lost some detail in this process
- But ensure that you eliminate objectionable high frequency artifacts
- Have to transform to frequency domain?
  - Expensive
  - Not necessary because of relationship between spatial and frequency domain

The Convolution Theorem

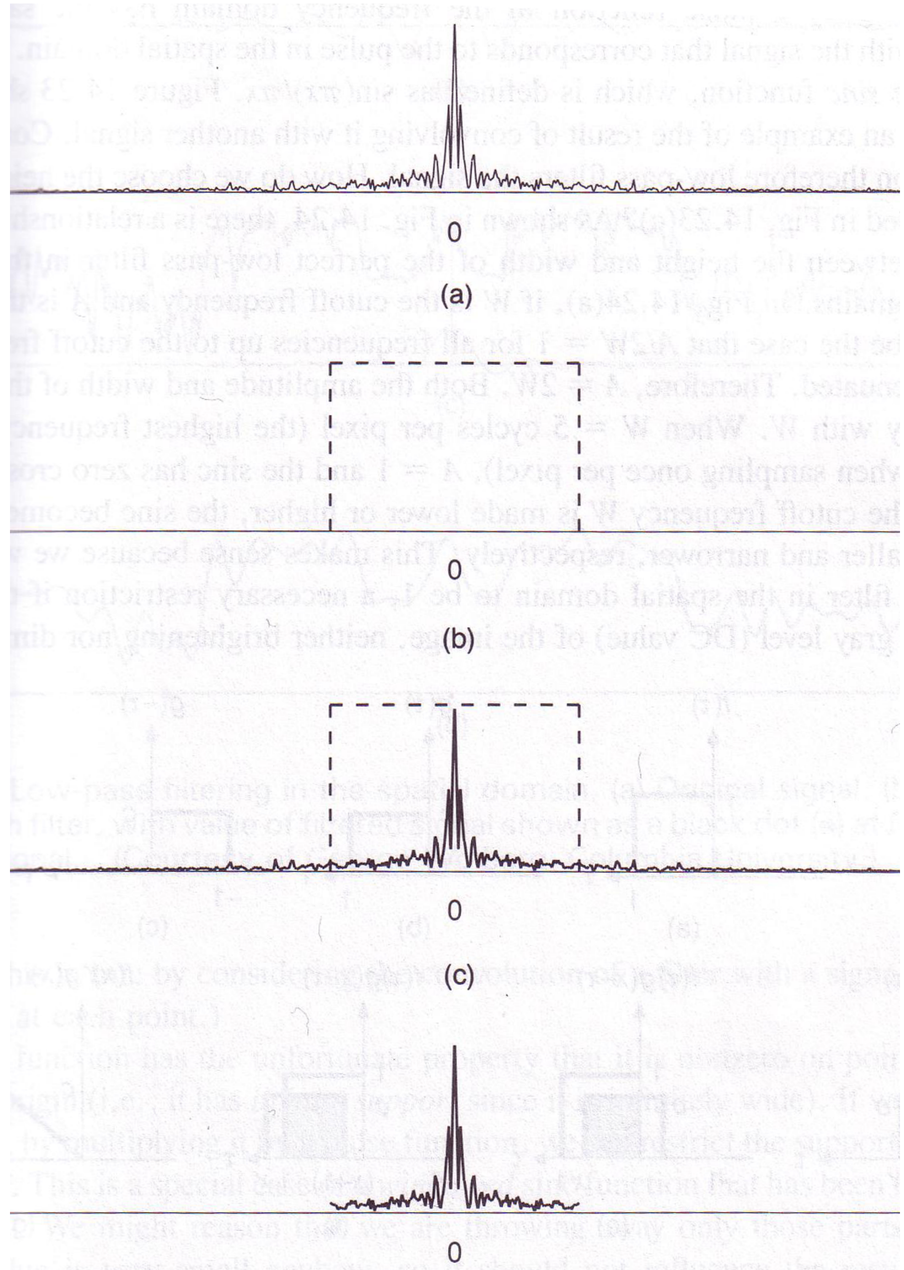
- The Fourier transform of the convolution of two functions is the product of their Fourier transforms

$$\mathrm{F}[g * h] = \mathrm{F}[g]\,\mathrm{F}[h]$$

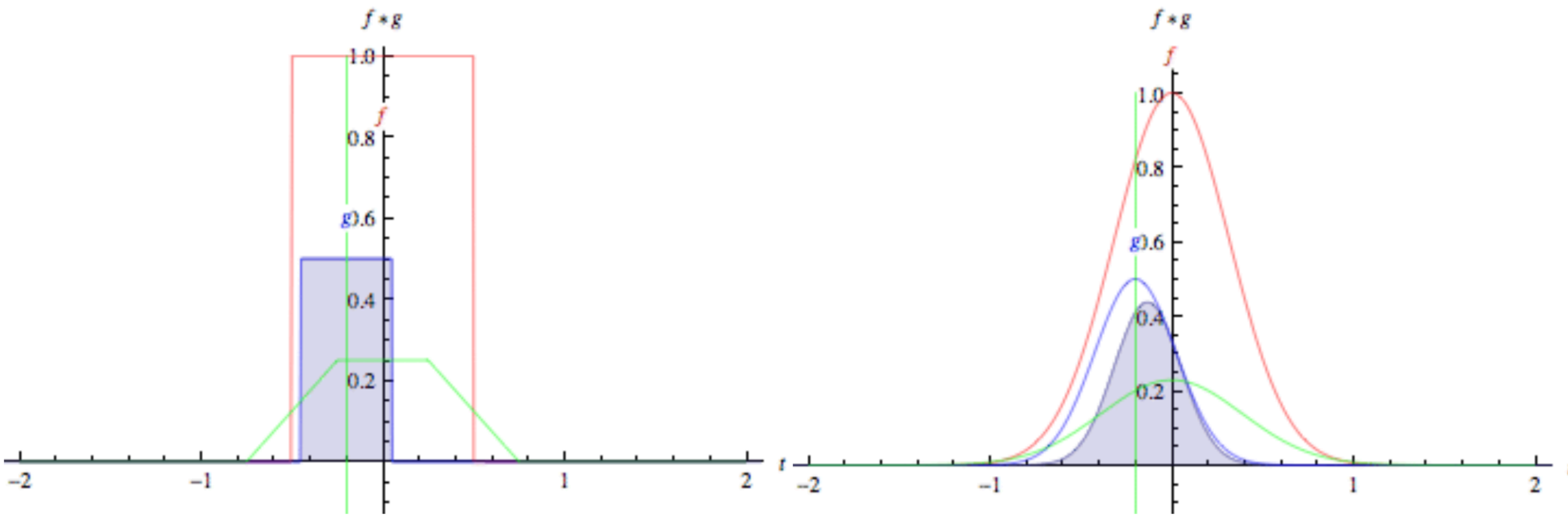
- **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!

$$g * h = \mathrm{F}^{-1}[\mathrm{F}[g]\,\mathrm{F}[h]]$$

- Don’t multiply in frequency domain
- Just convolve in spatial domain



http://mathworld.wolfram.com/Convolution.html



Types of Filters

Spatial

- Sine
- Gaussian
- Box
- Sinc

Frequency

- Impulse
- Gaussian
- Sinc
- Box

Pre-Filtering

- Ideal is box/pulse in frequency space
- Sinc in spatial domain
- But infinite spread

Truncated sinc

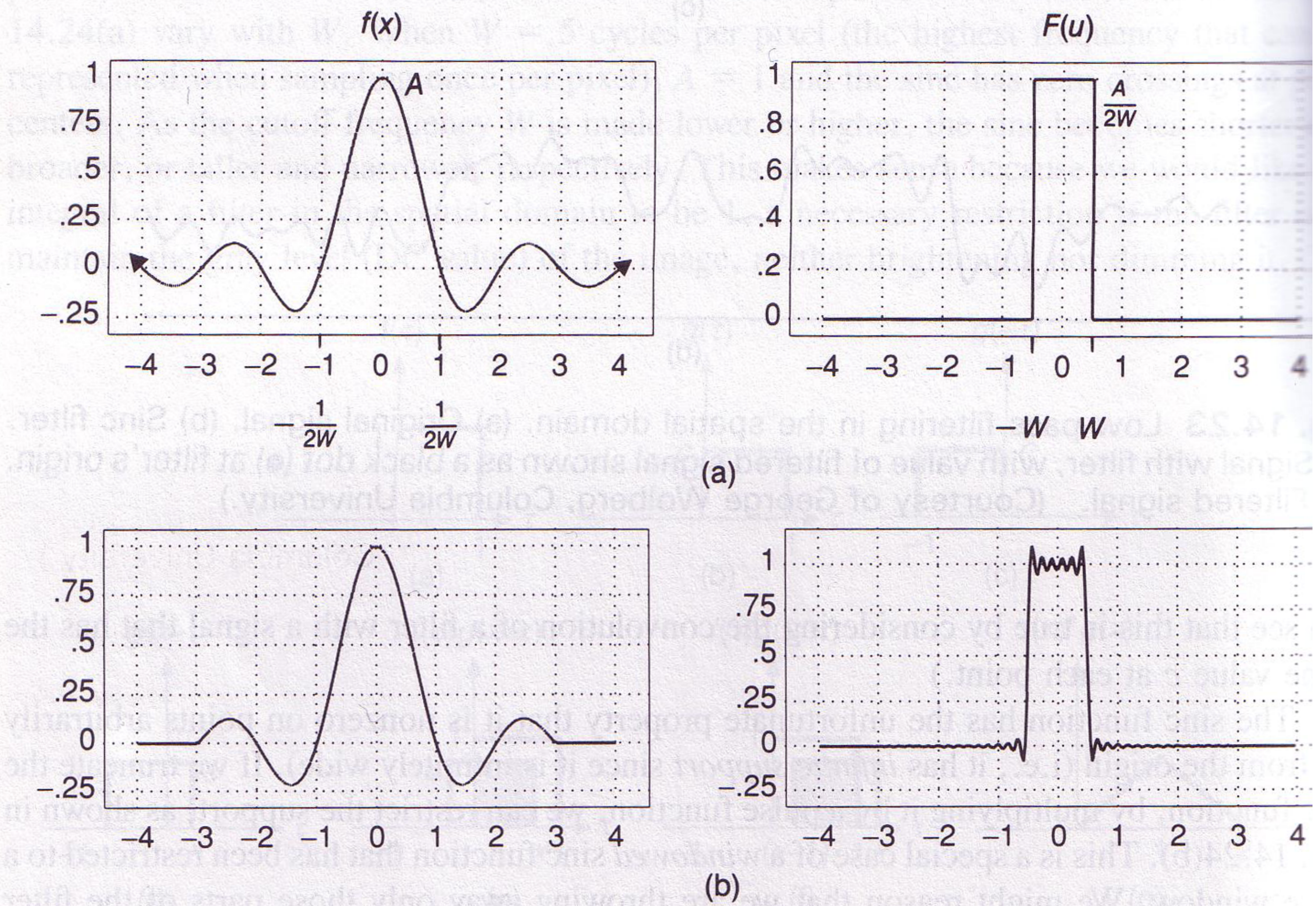


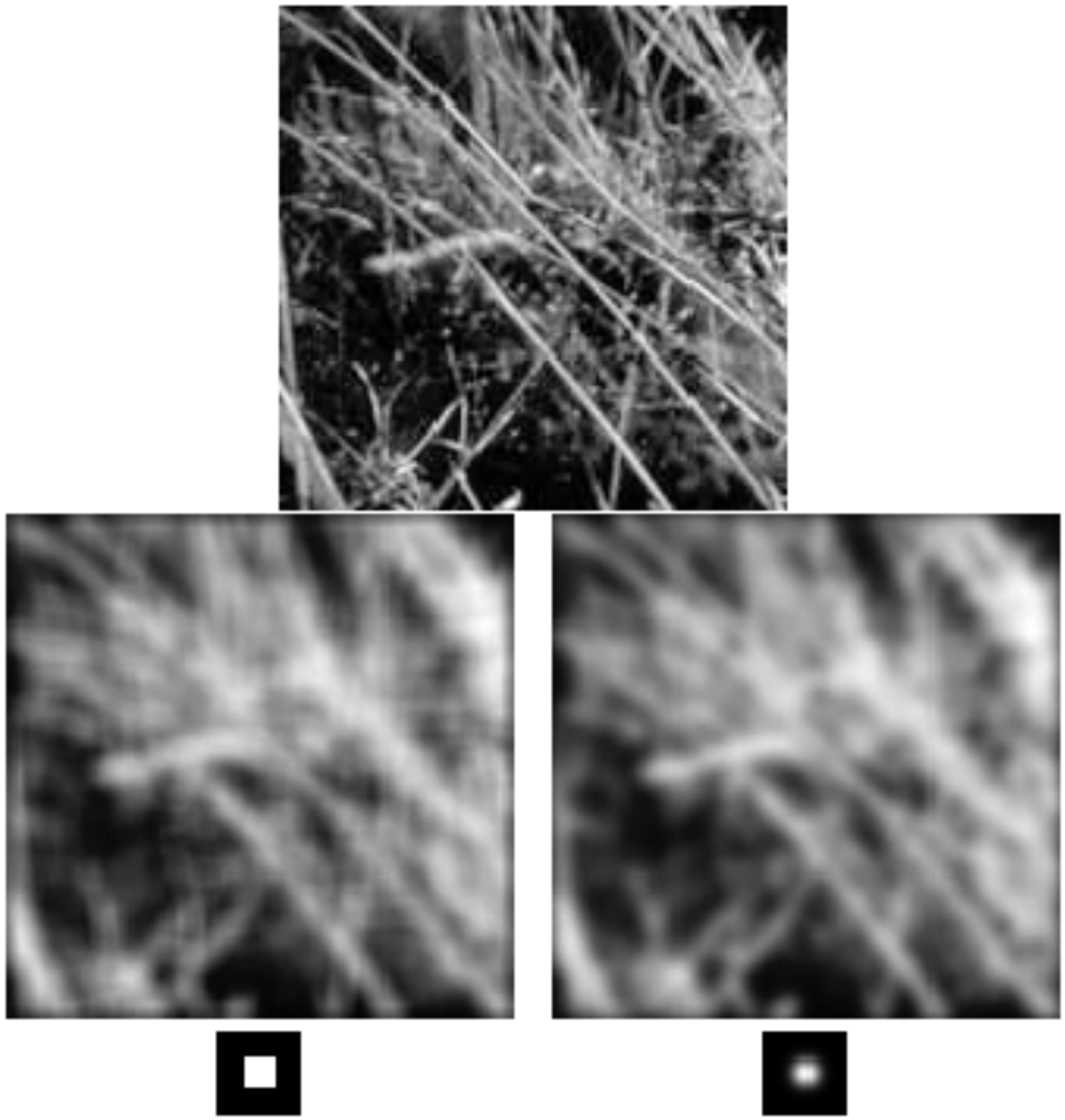
Fig. 14.24 (a) Sinc in spatial domain corresponds to pulse in frequency dom
(b) Truncated sinc in spatial domain corresponds to ringing pulse in frequency dom
(Courtesy of George Wolberg, Columbia University.)

Gaussian Filter

- Gaussian is preferred because it is same in frequency and spatial domain

- Not perfect: attenuates

Mean vs. Gaussian filtering



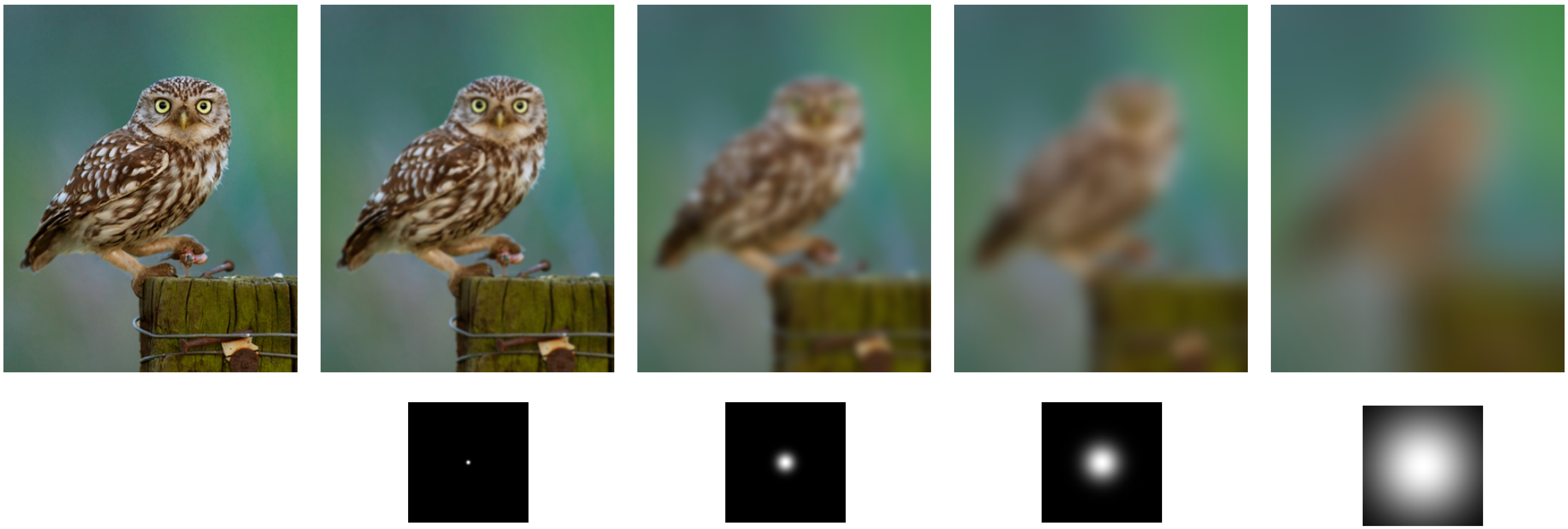
Gaussian filter

- Removes "high-frequency" components from the image (low-pass filter)
- Convolution with self is another Gaussian

  - Convolving twice with Gaussian kernel of width σ = convolving once with kernel of width $\sigma\sqrt{2}$

Source: K. Grauman

Gaussian filters



σ = 1 pixel | σ = 5 pixels | σ = 10 pixels | σ = 30 pixels

Image Resampling

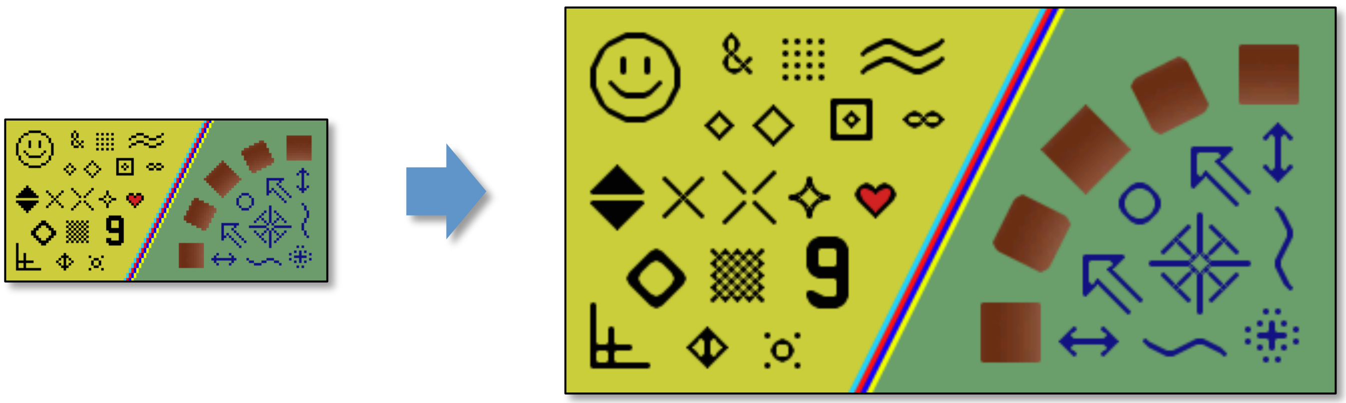


Image Scaling

This image is too big to fit on the screen. How can we generate a half-sized version?

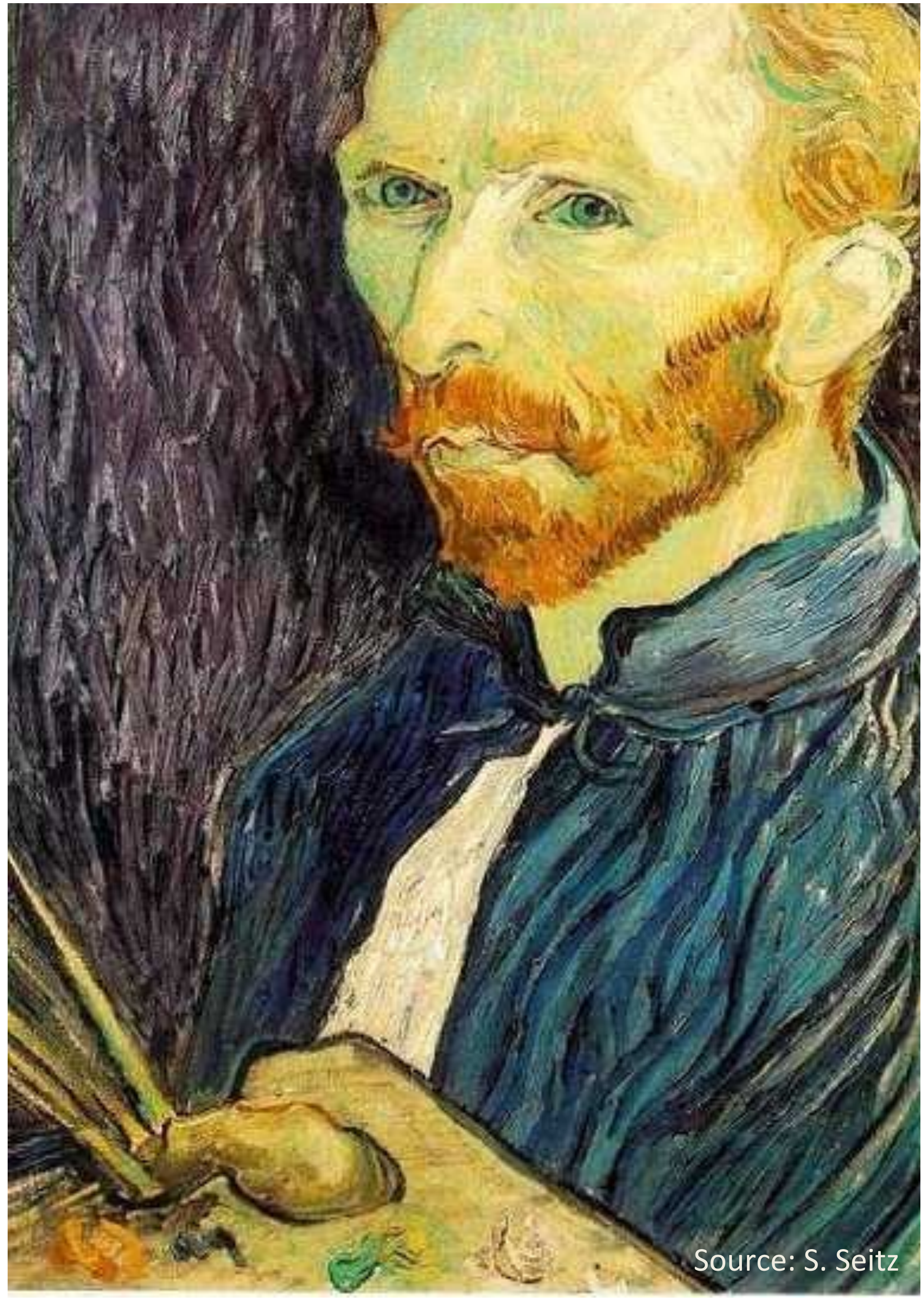
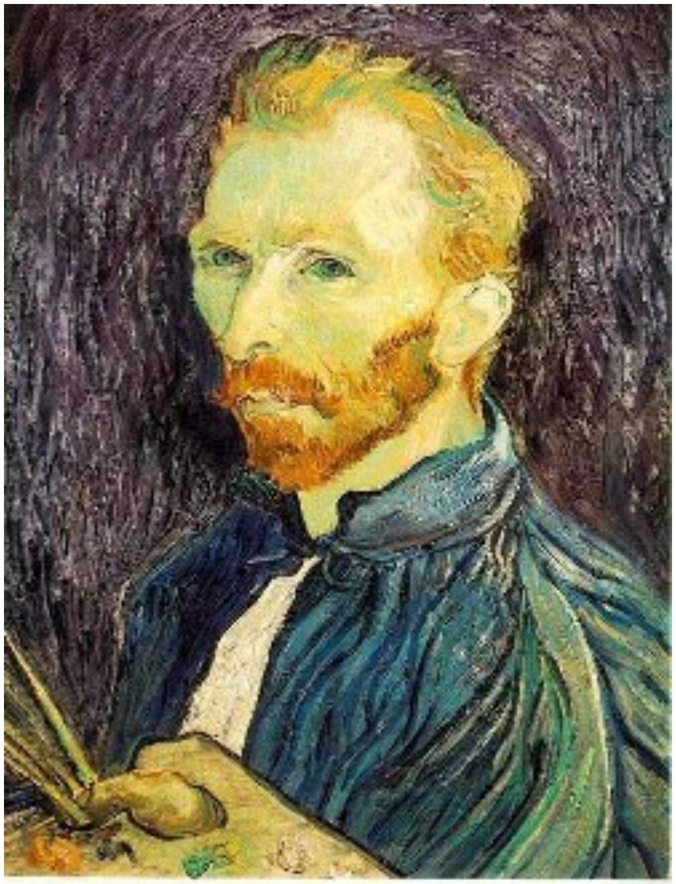


Image sub-sampling



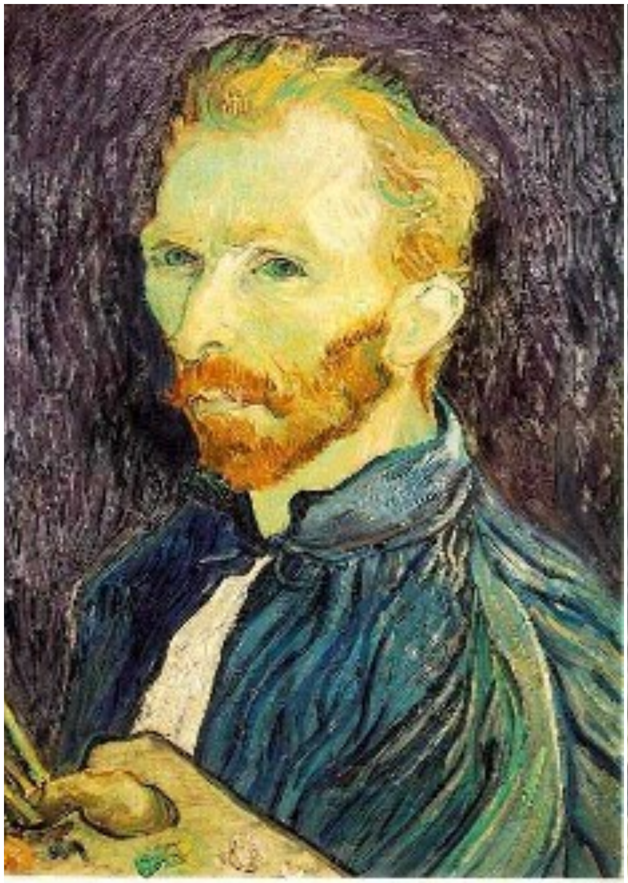
1/4

1/16

Throw away every other row and column to create a 1/2 size image
- called *image sub-sampling*

Source: S. Seitz

Image sub-sampling



1/2

1/4 (2x zoom)

1/16 (4x zoom)

Why does this look so crufty?

Source: S. Seitz

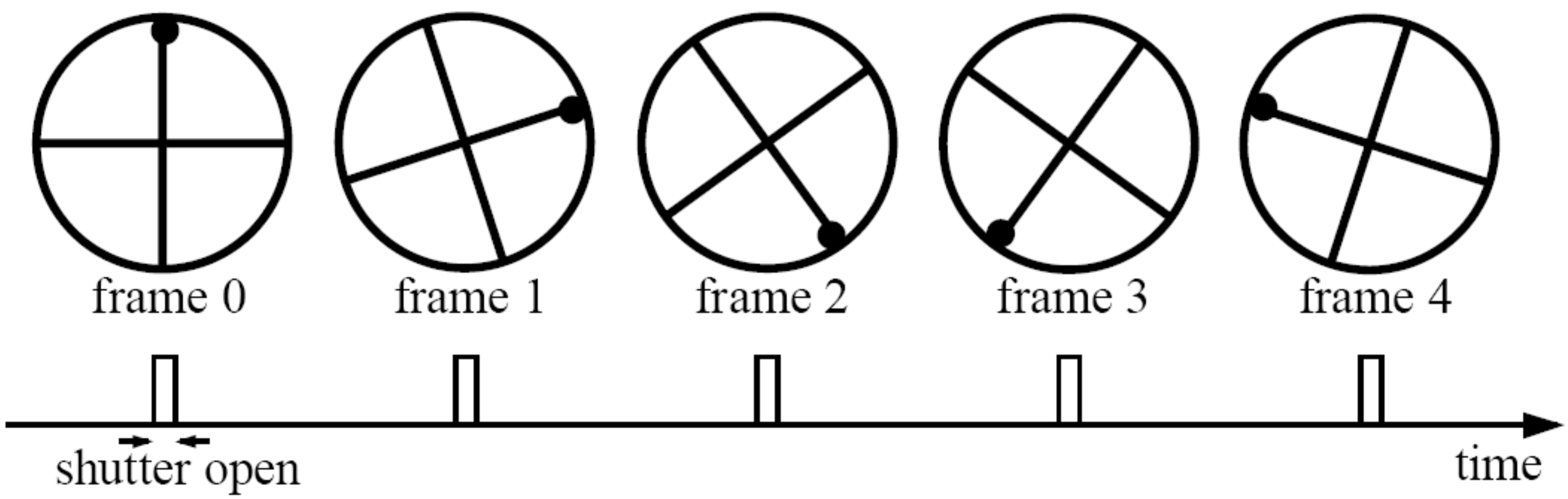
Image sub-sampling

Source: F. Durand

Wagon-wheel effect

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):

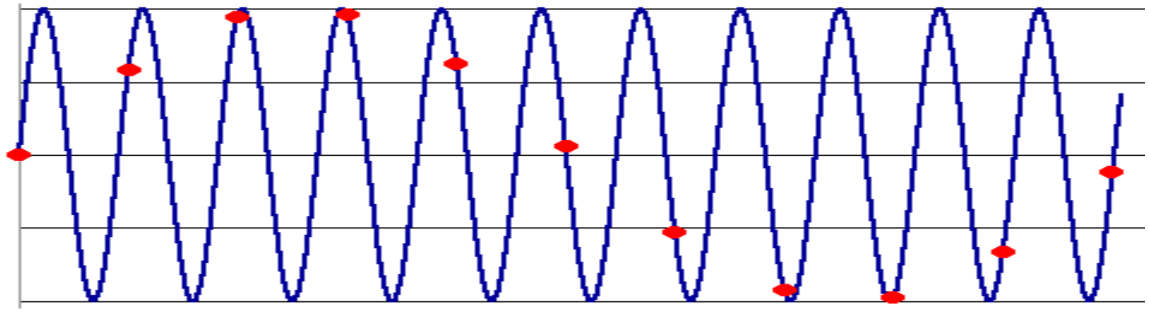


Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

(See http://www.michaelbach.de/ot/mot_wagonWheel/index.html)

Source: L. Zhang

Aliasing

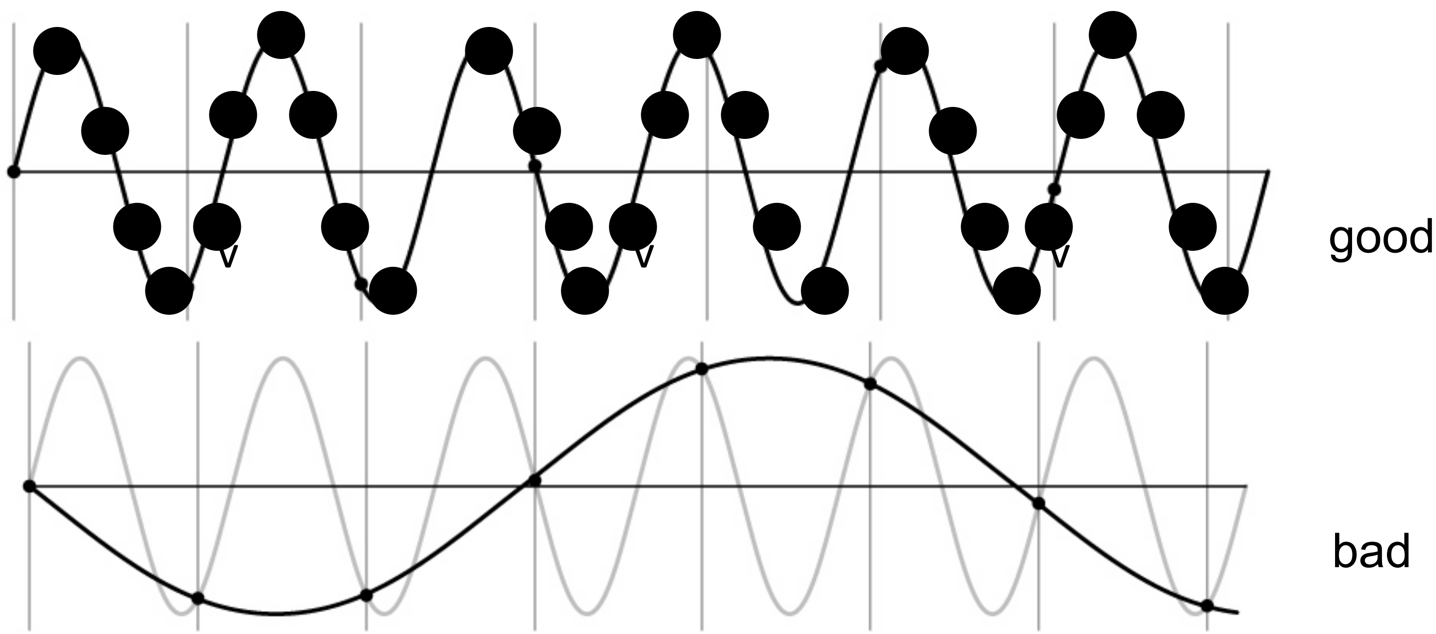


- Occurs when your sampling rate is not high enough to capture the amount of detail in your image
- Can give you the wrong signal/image—an *alias*
- To do sampling right, need to understand the structure of your signal/image
- Enter Monsieur Fourier…

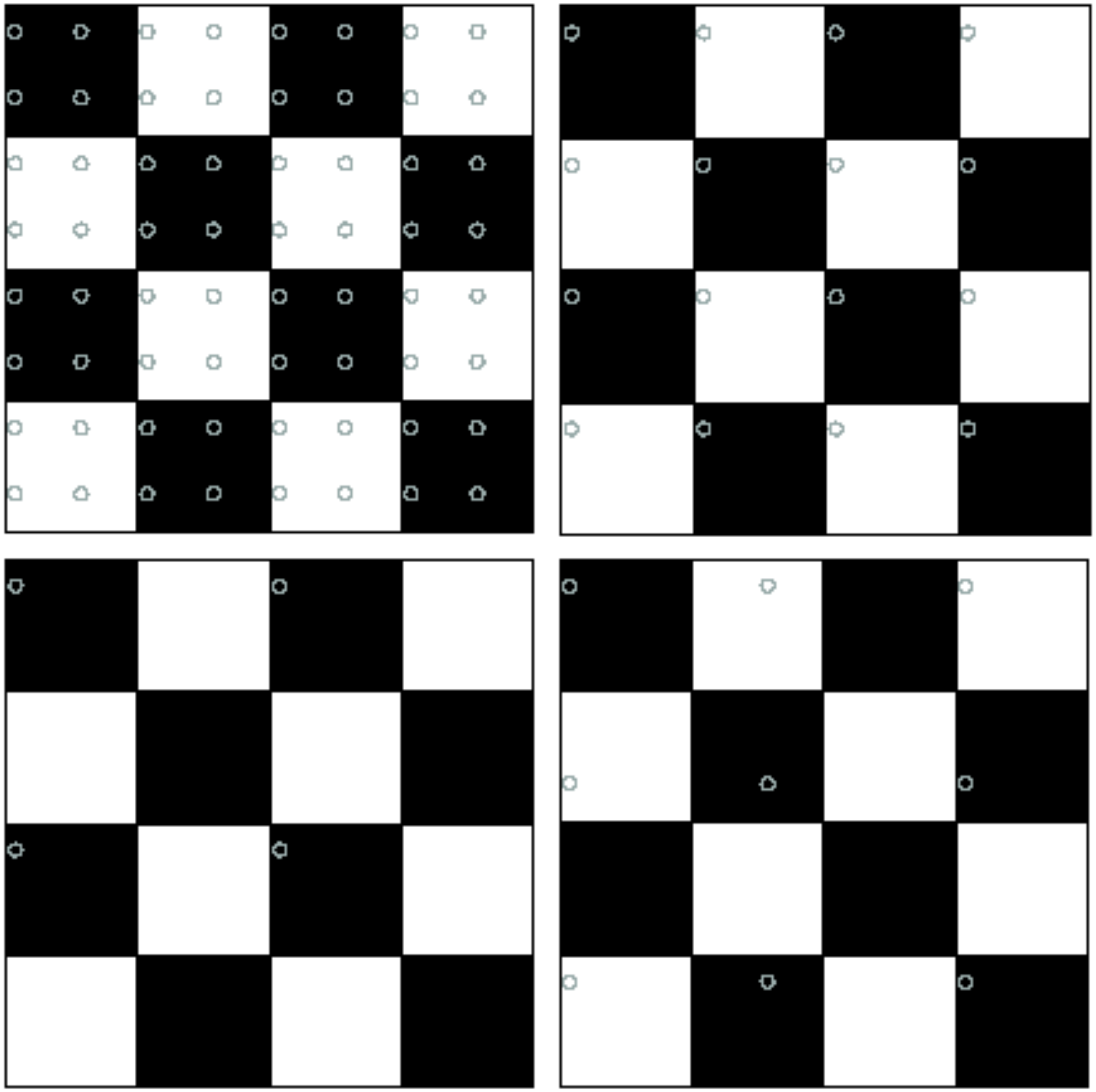
Source: L. Zhang

Nyquist-Shannon Sampling Theorem

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



Nyquist limit – 2D example



Good sampling

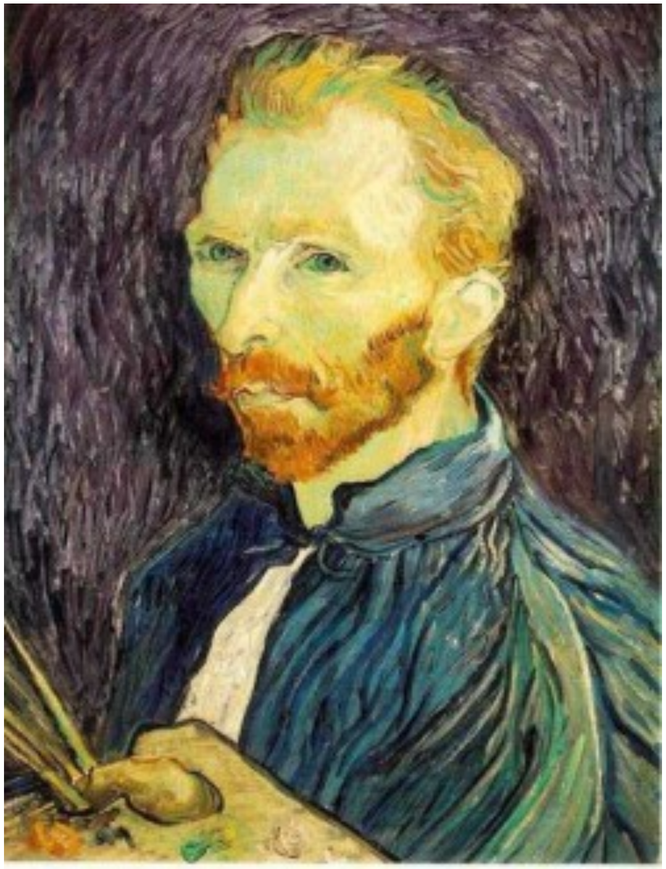
Bad sampling

Aliasing

- When downsampling by a factor of two
  - Original image has frequencies that are too high

- How can we fix this?

Gaussian pre-filtering



Gaussian 1/2

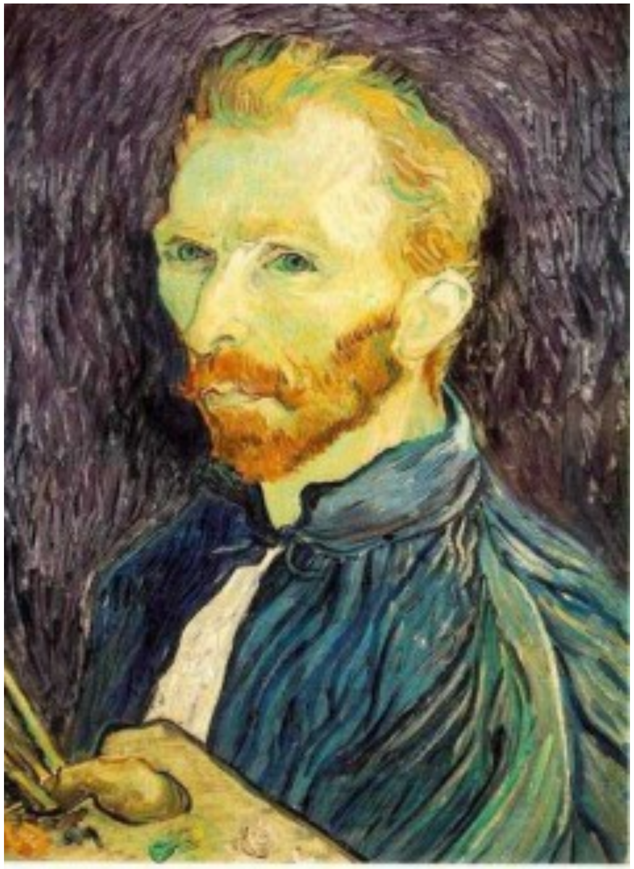
G 1/4

G 1/8

- Solution: filter the image, *then* subsample

Source: S. Seitz

Subsampling with Gaussian pre-filtering



Gaussian 1/2

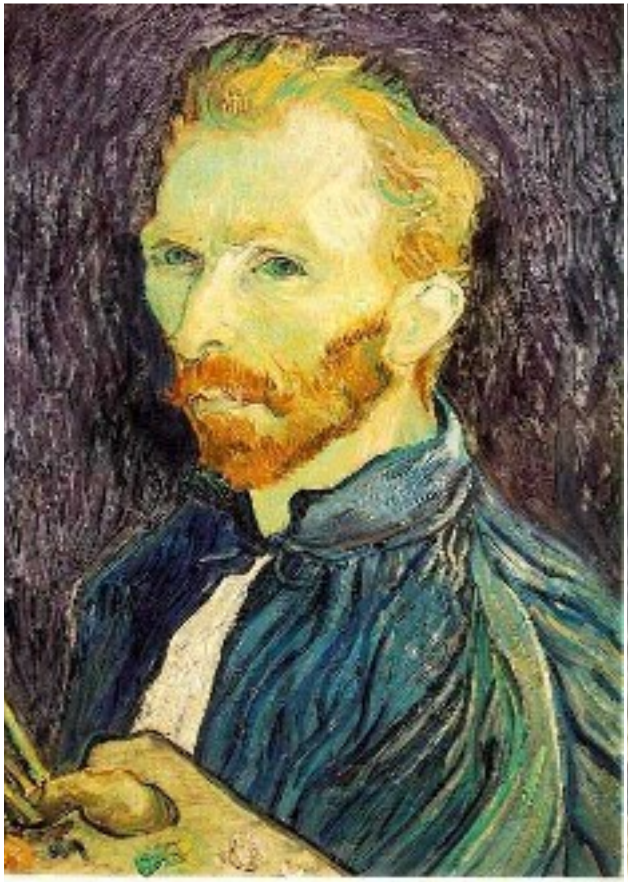
G 1/4

G 1/8

- Solution: filter the image, *then* subsample

Source: S. Seitz

Compare with...



1/2

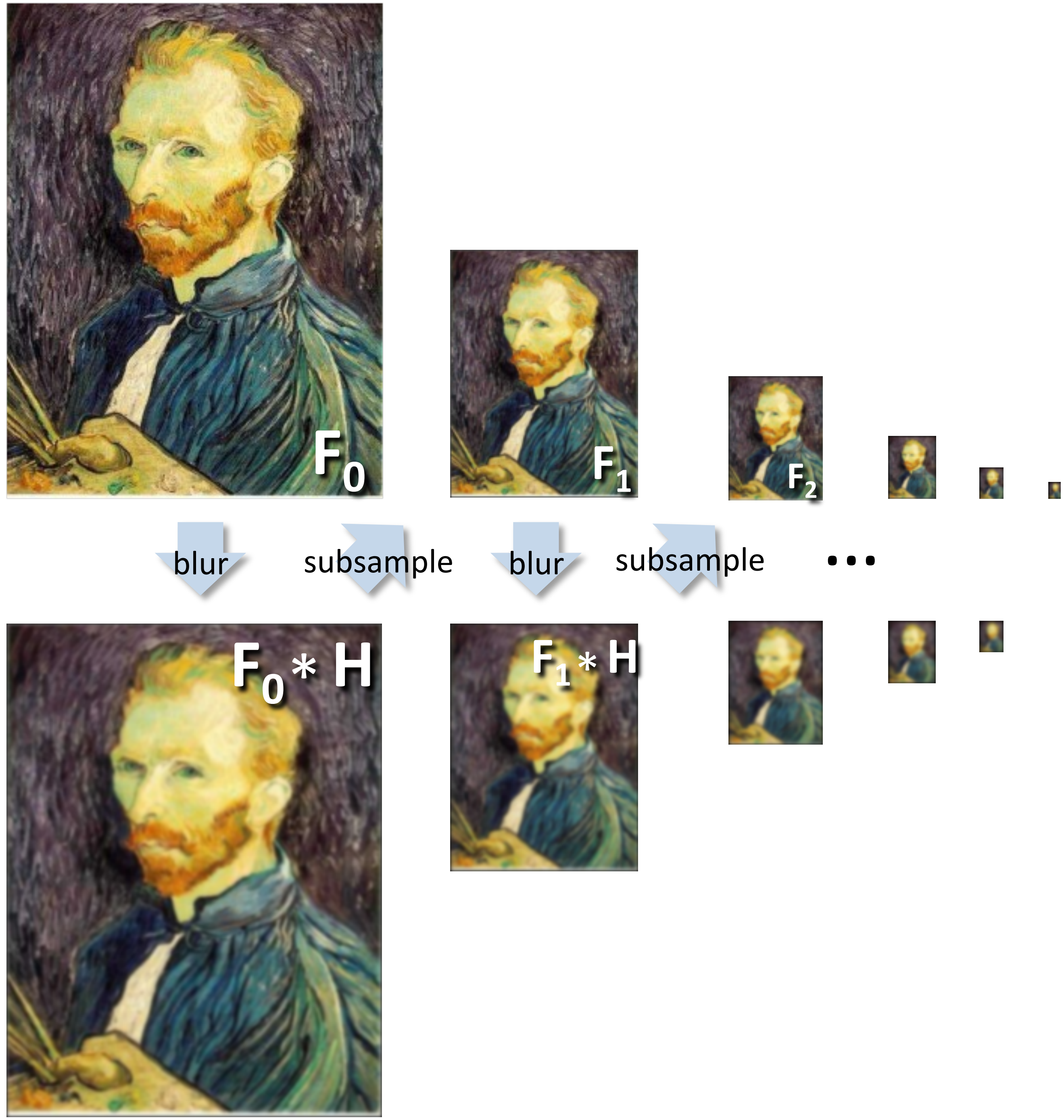
1/4 (2x zoom)

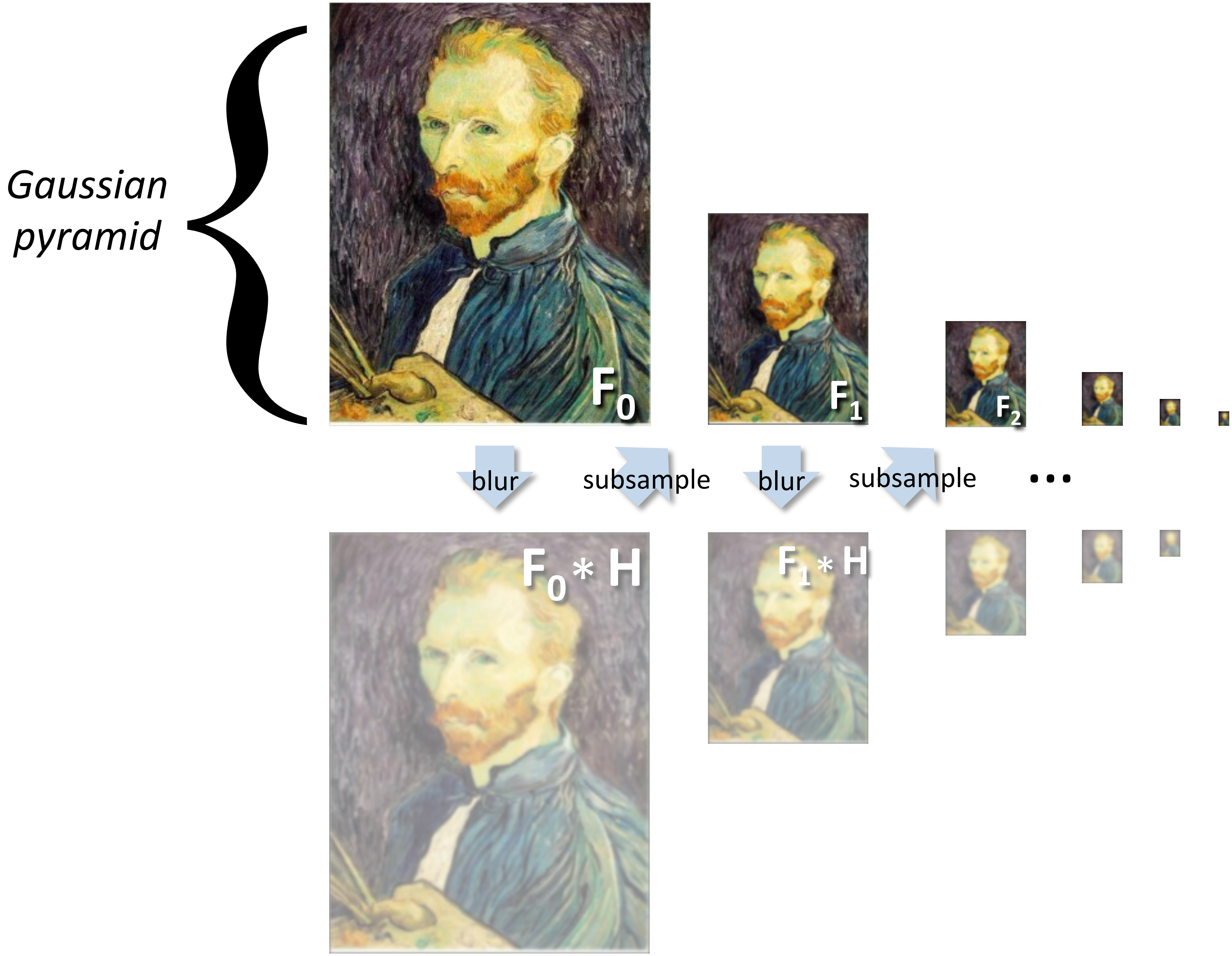
1/8 (4x zoom)

Source: S. Seitz

Gaussian pre-filtering

- Solution: filter the image, *then* subsample

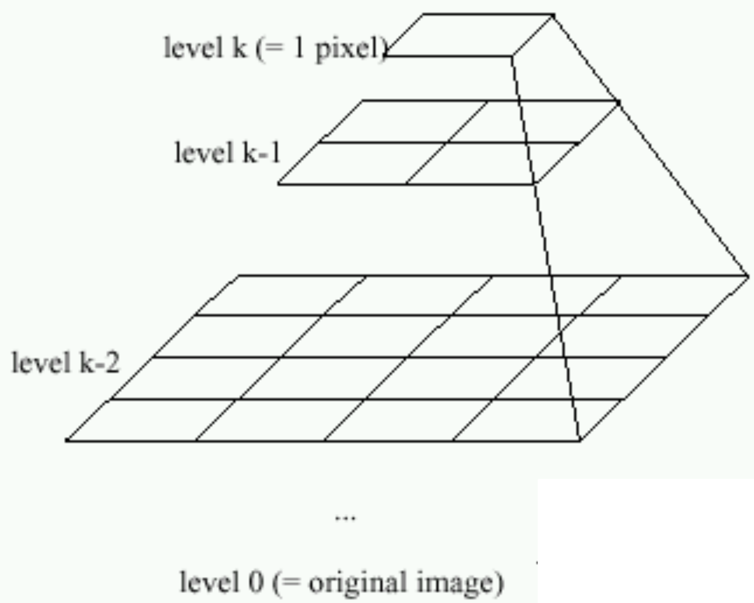




Gaussian pyramid
F0
F1
F2
blur
subsample
blur
subsample
...
F0 * H
F1 * H

Gaussian pyramids
[Burt and Adelson, 1983]

Idea: Represent NxN image as a "pyramid" of 1x1, 2x2, 4x4,…, $2^k x 2^k$ images (assuming $N=2^k$)

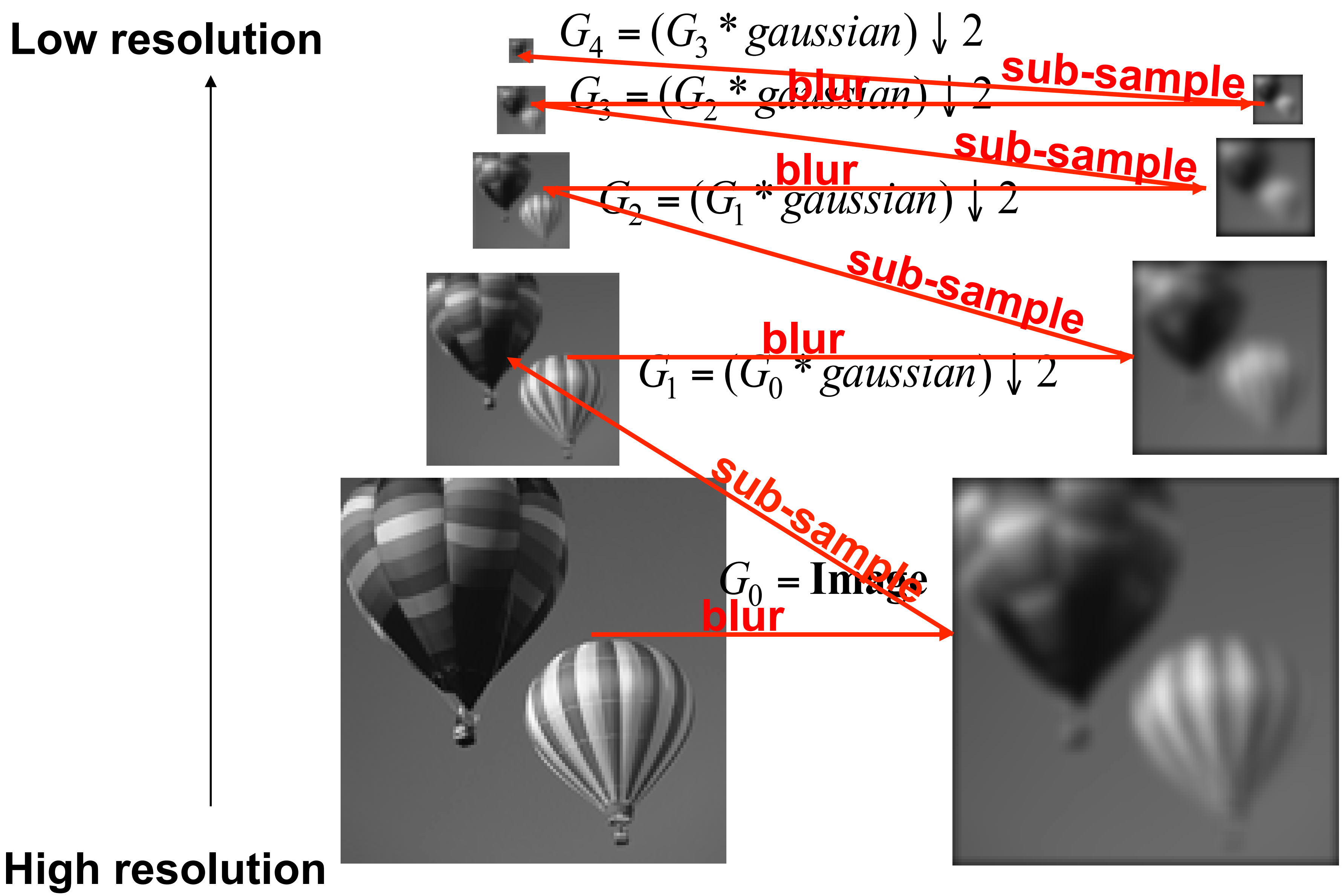


- In computer graphics, a *mip map* [Williams, 1983]

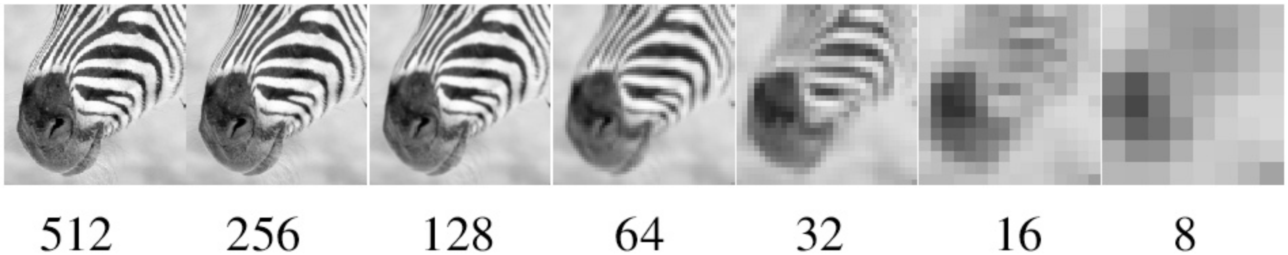
Gaussian Pyramids have all sorts of applications in computer vision

Source: S. Seitz

The Gaussian Pyramid



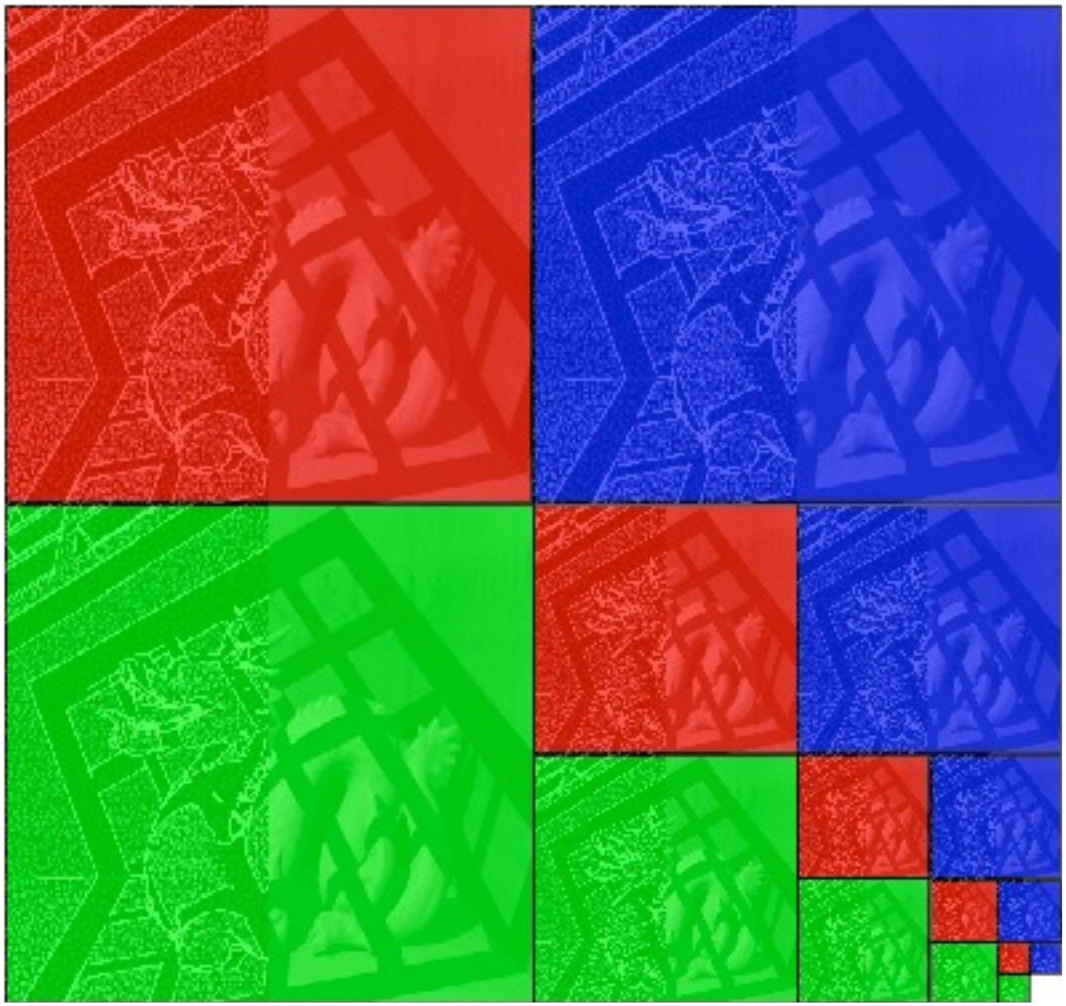
Gaussian pyramid and stack

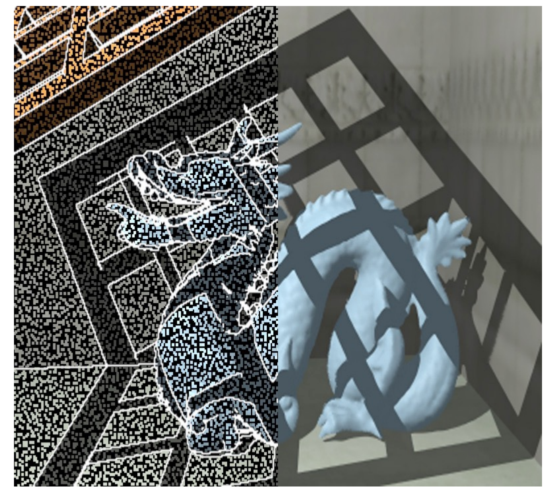


Source: Forsyth

Memory Usage

- What is the size of the pyramid?

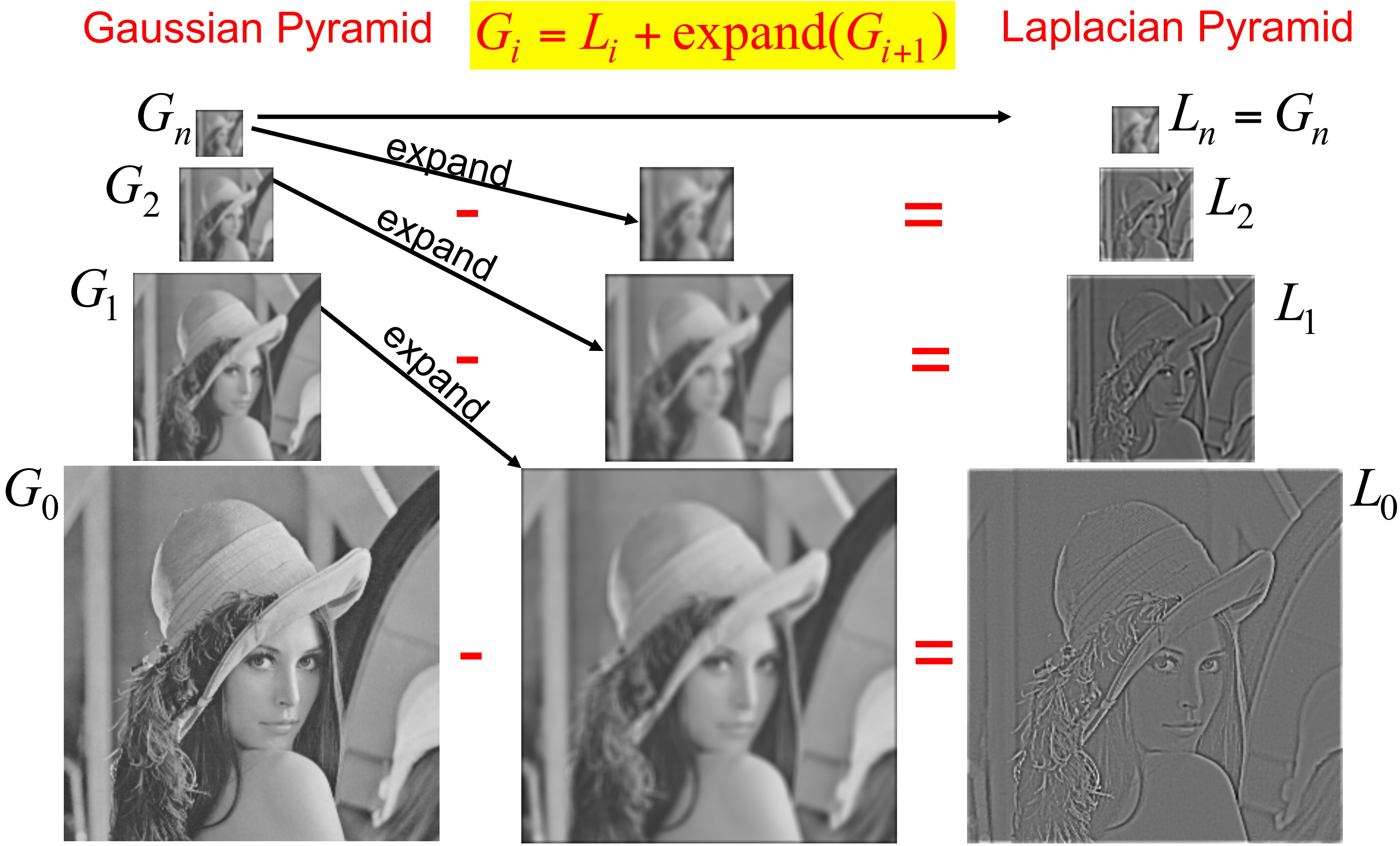




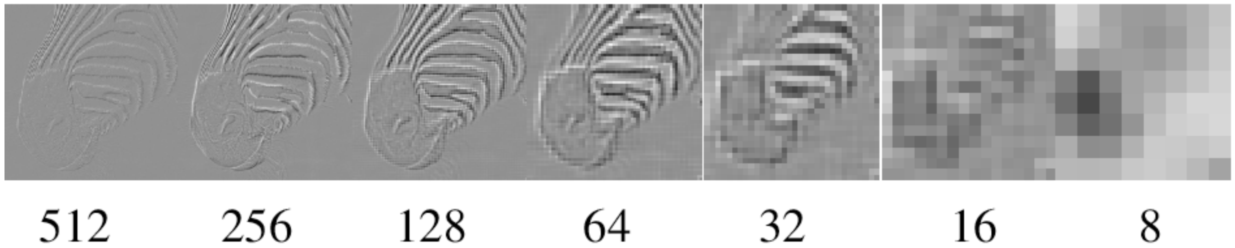
The Laplacian Pyramid

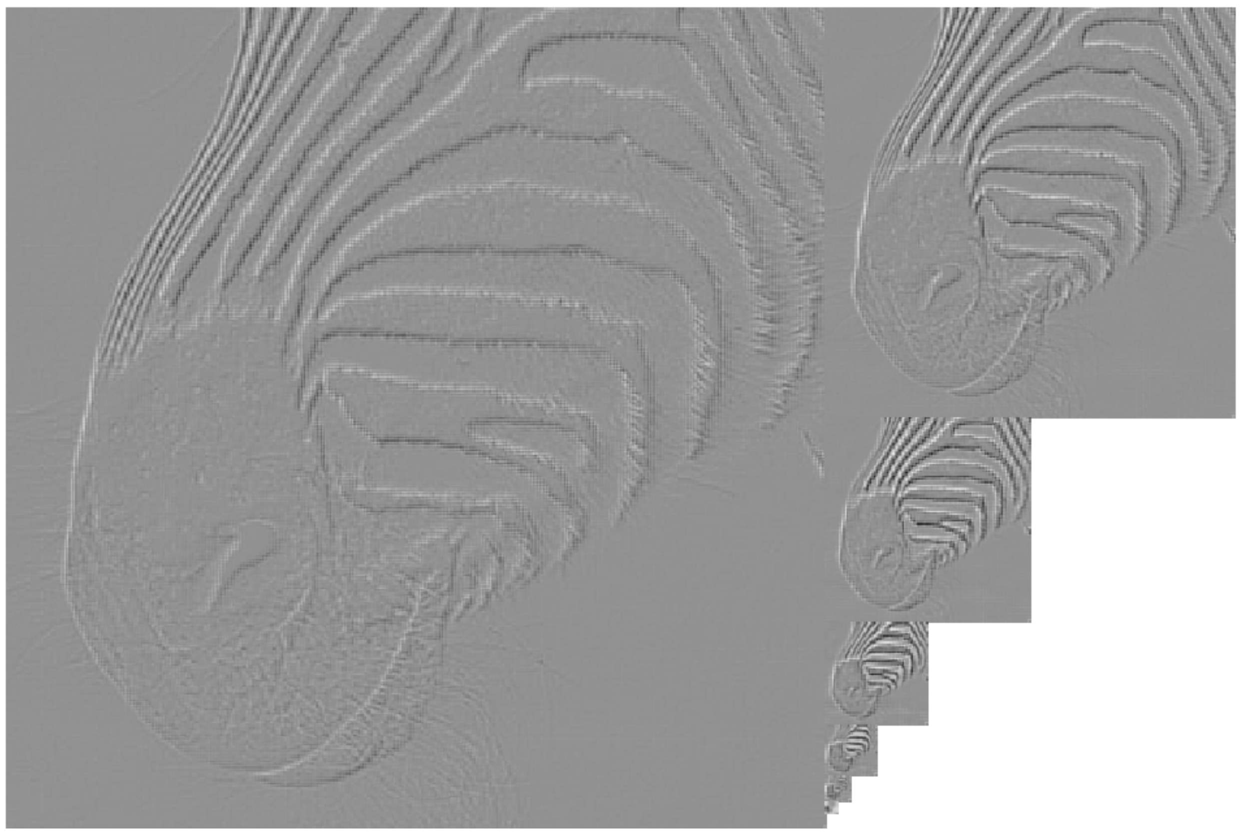
$$L_i = G_i - \text{expand}(G_{i+1})$$

$$G_i = L_i + \text{expand}(G_{i+1})$$



Laplacian pyramid

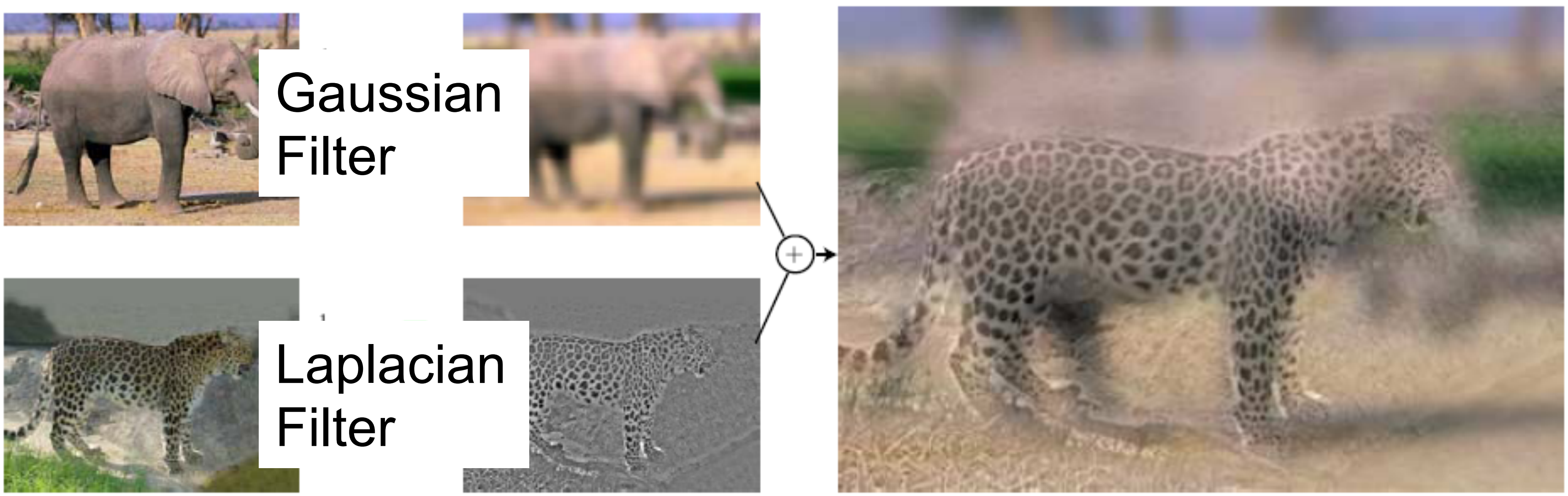




Source: Forsyth

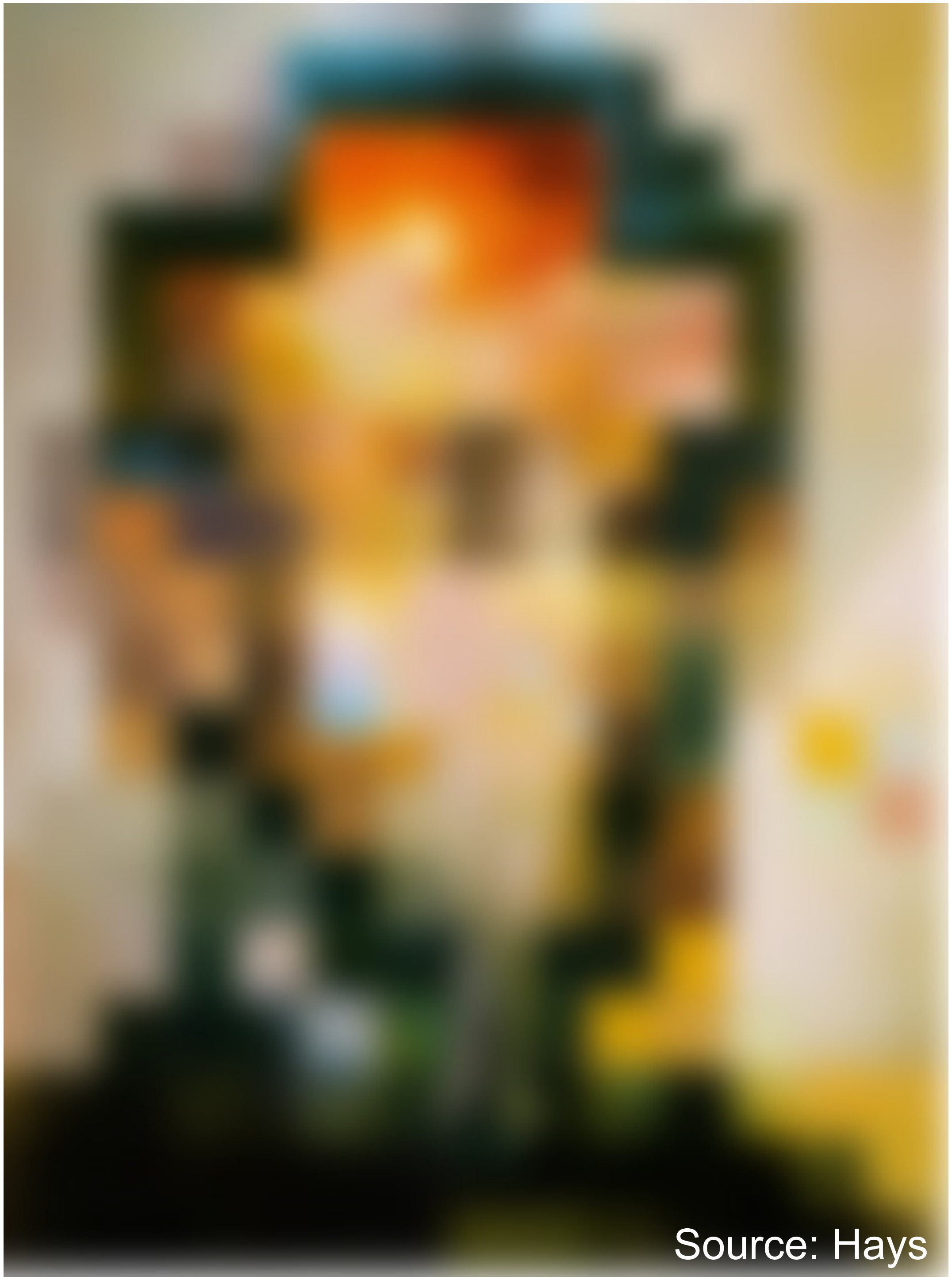
PA1 (A): Hybrid Images

A. Oliva, A. Torralba, P.G. Schyns,
"Hybrid Images," SIGGRAPH 2006



Source: Hays

Salvador Dali invented Hybrid Images?



Salvador Dali
"Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln", 1976

Salvador Dali invented Hybrid Images?

Salvador Dali
"Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln", 1976

Major uses of image pyramids

- Compression
- Object detection
  - Scale search
  - Features
- Registration
  - Course-to-fine