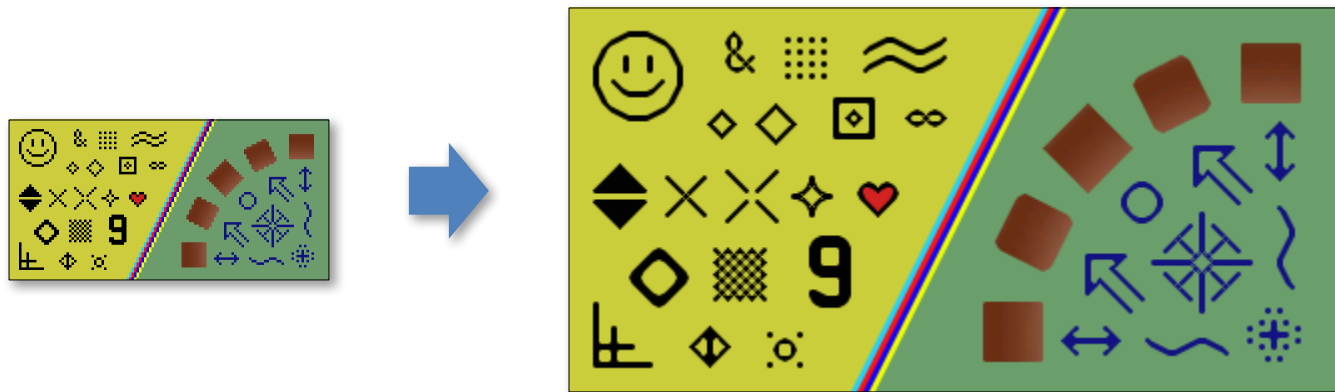


# CS5670: Computer Vision

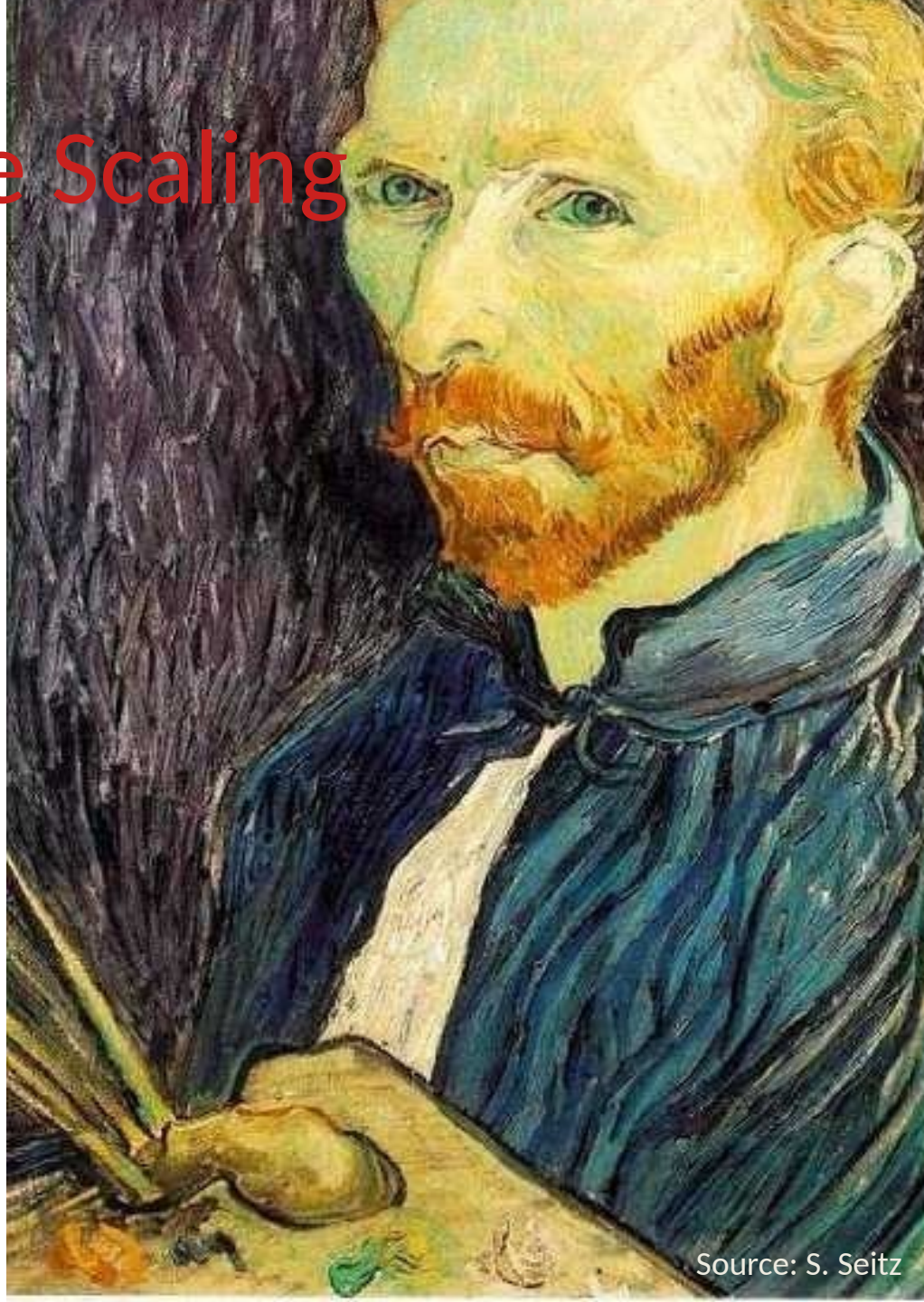
Noah Snavely

## Image Resampling & Interpolation



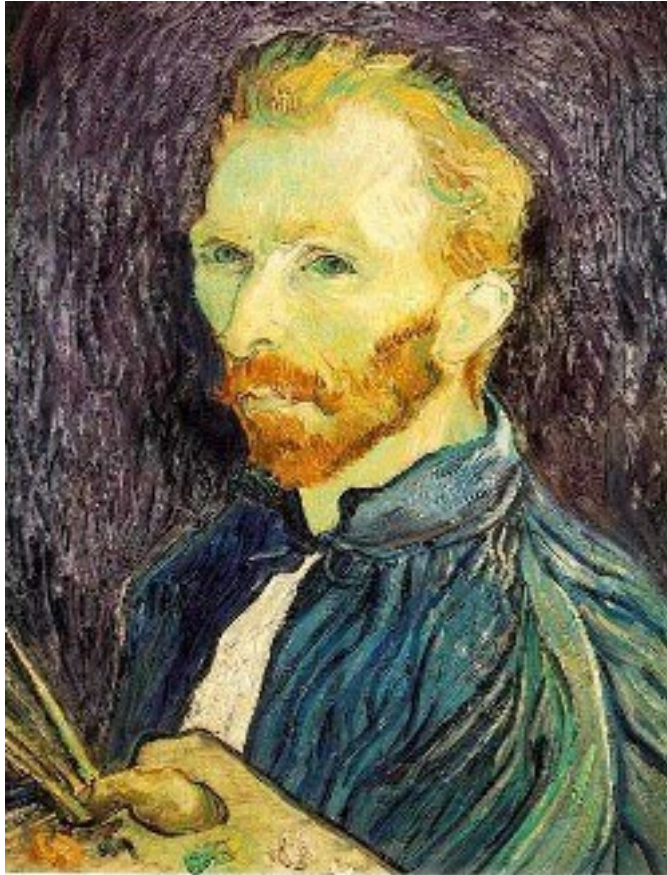
# Image Scaling

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



Source: S. Seitz

# Image sub-sampling



1/4

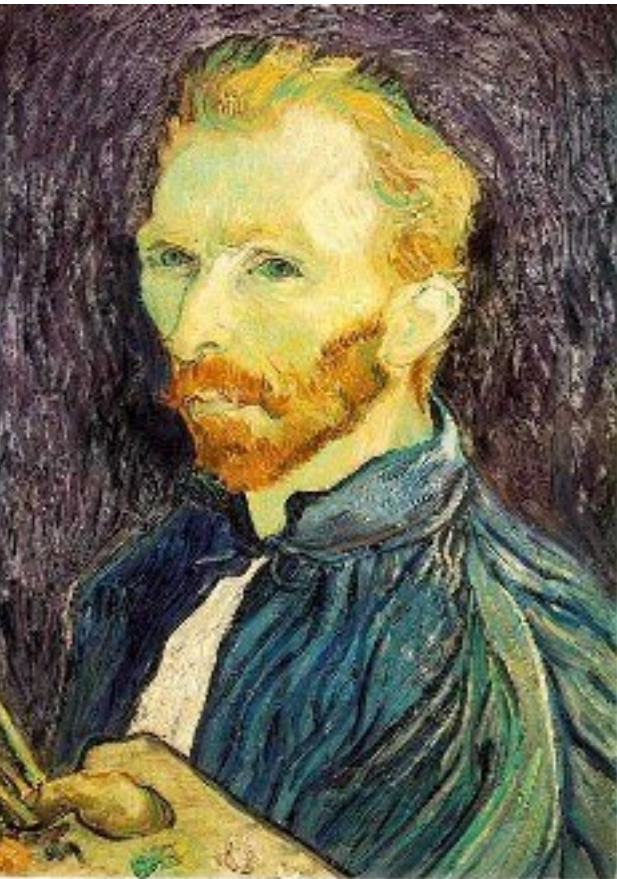


1/8

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



# Image sub-sampling



$1/2$



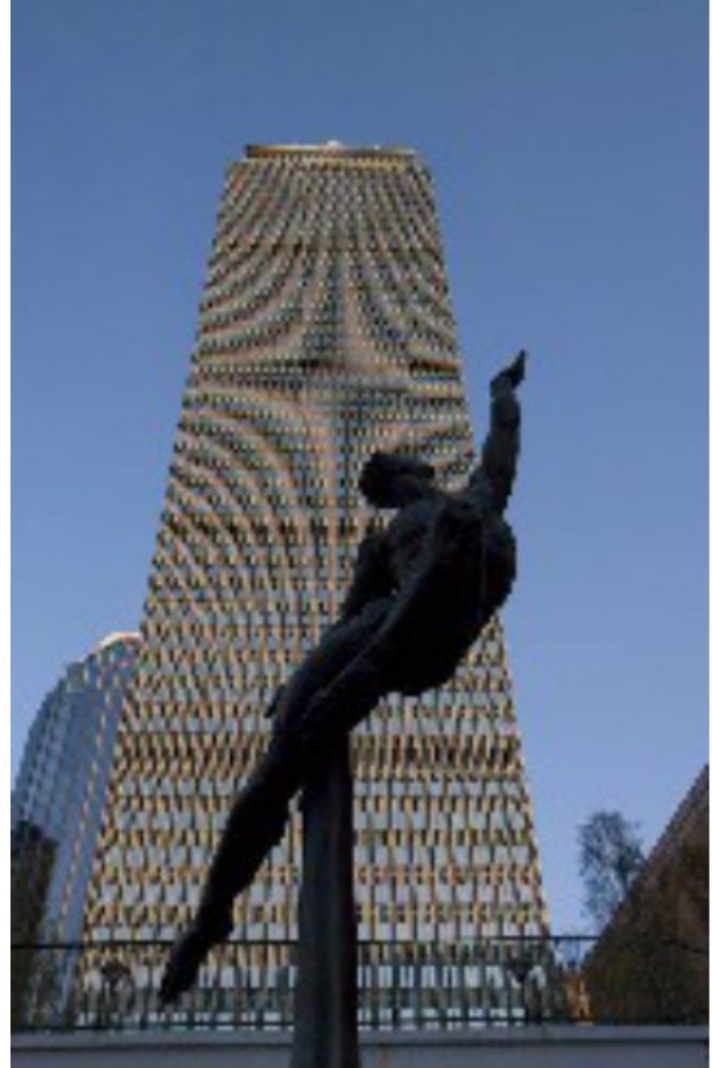
$1/4$  (2x zoom)



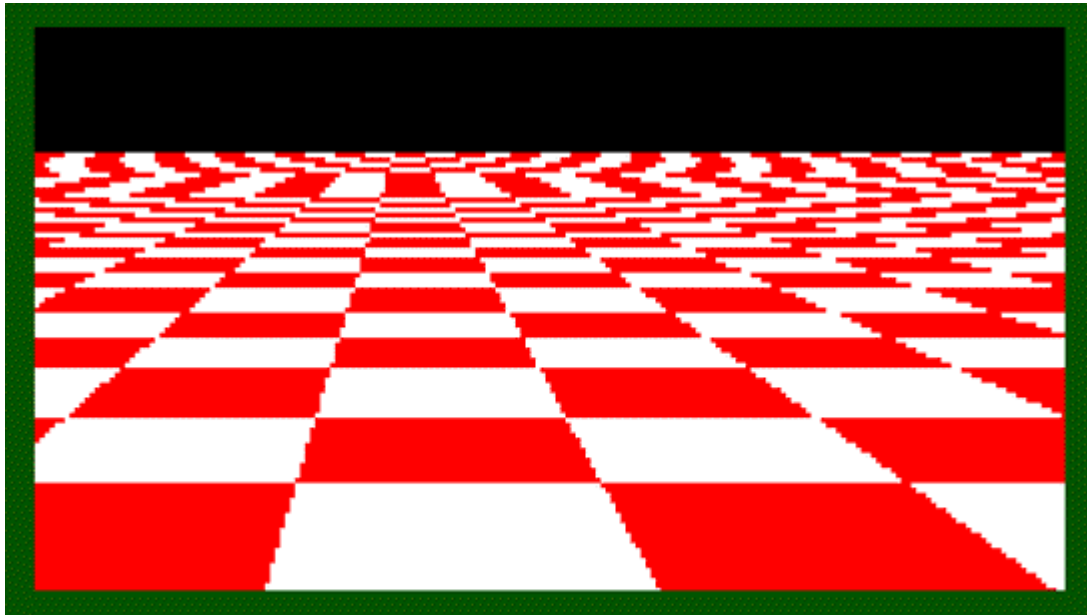
$1/8$  (4x zoom)

Why does this look so cruffy?

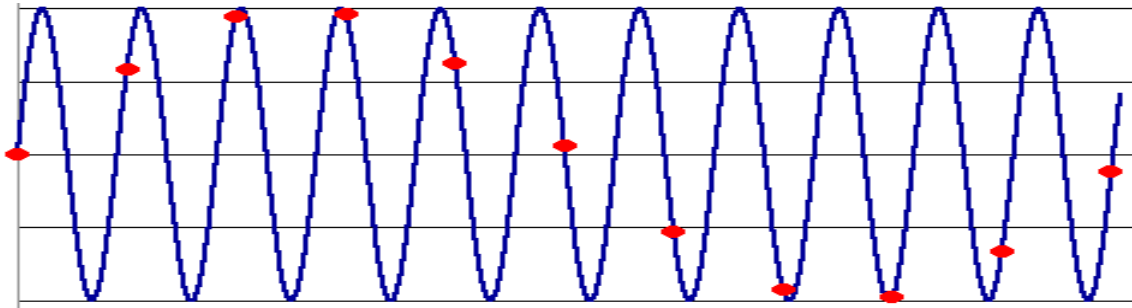
# Image sub-sampling



# Even worse for synthetic images



# Aliasing



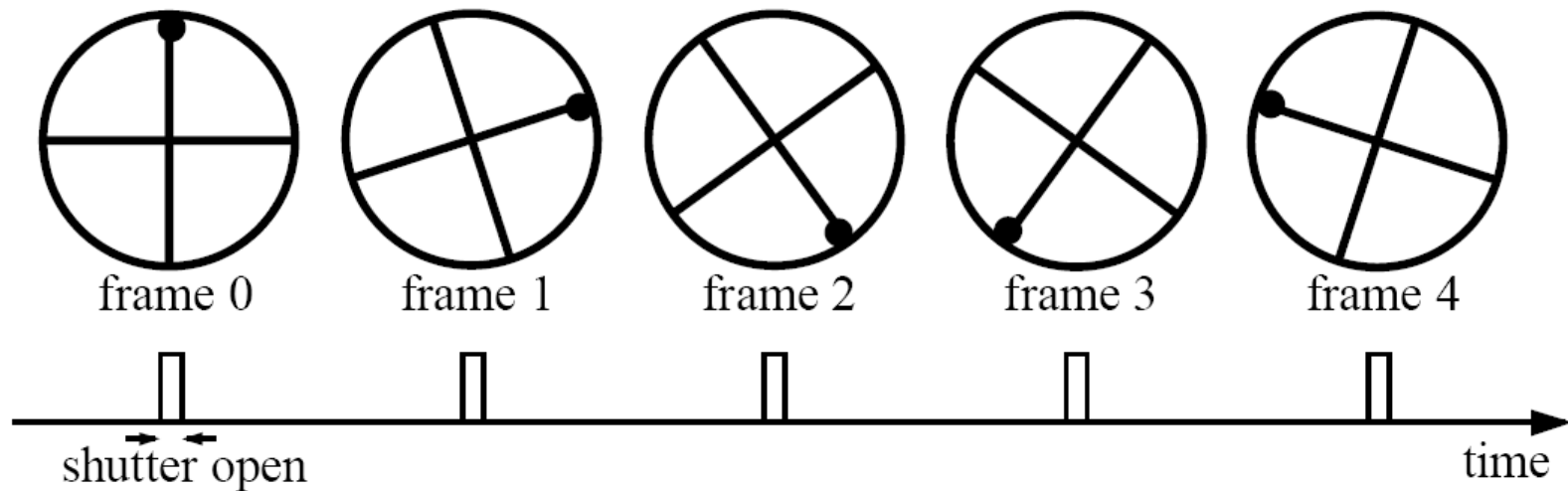
- Occurs when your sampling rate is not high enough to capture the amount of detail in your image
- Can give you the wrong signal/image—an *alias*
- To do sampling right, need to understand the structure of your signal/image
- Enter Monsieur Fourier...
- To avoid aliasing:
  - sampling rate  $\geq 2 \times$  max frequency in the image
    - said another way:  $\geq$  two samples per cycle
  - This minimum sampling rate is called the **Nyquist rate**

# Wagon-wheel effect

Imagine a spoked wheel moving to the right (rotating clockwise).

Mark wheel with dot so we can see what's happening.

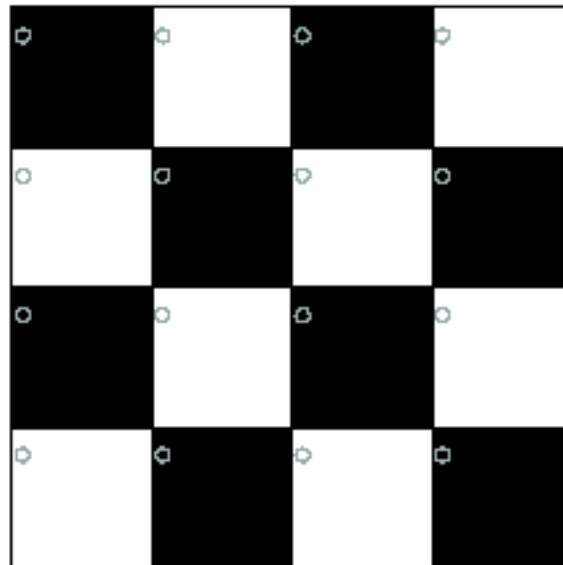
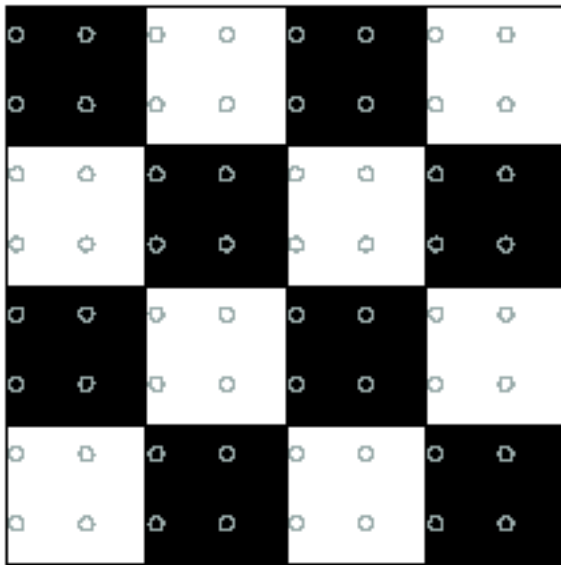
If camera shutter is only open for a fraction of a frame time (frame time =  $1/30$  sec. for video,  $1/24$  sec. for film):



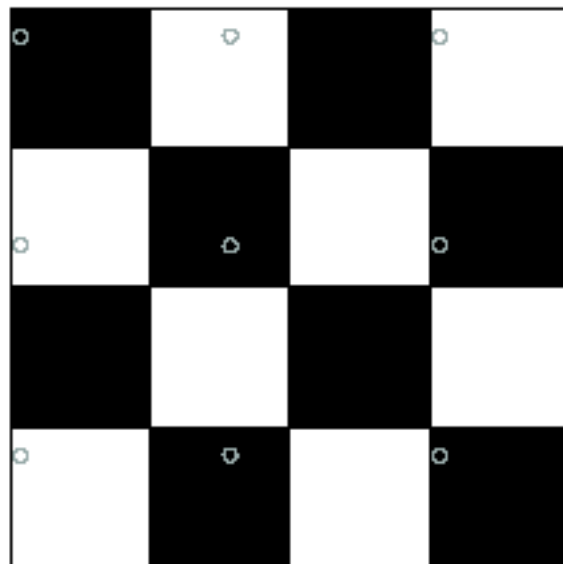
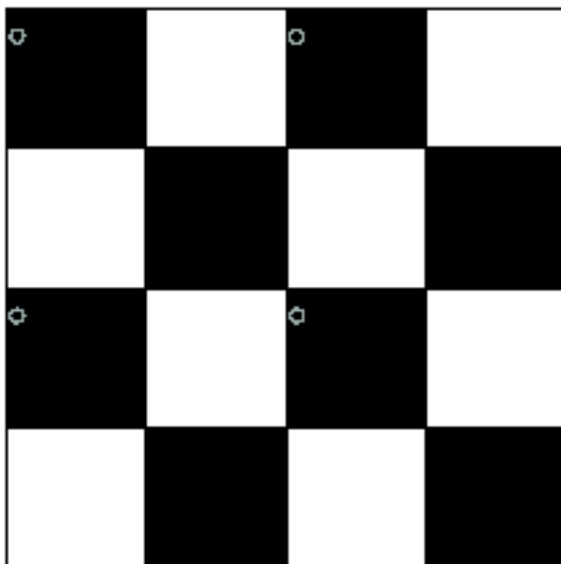
Without dot, wheel appears to be rotating slowly backwards!  
(counterclockwise)



# Nyquist limit – 2D example



Good sampling

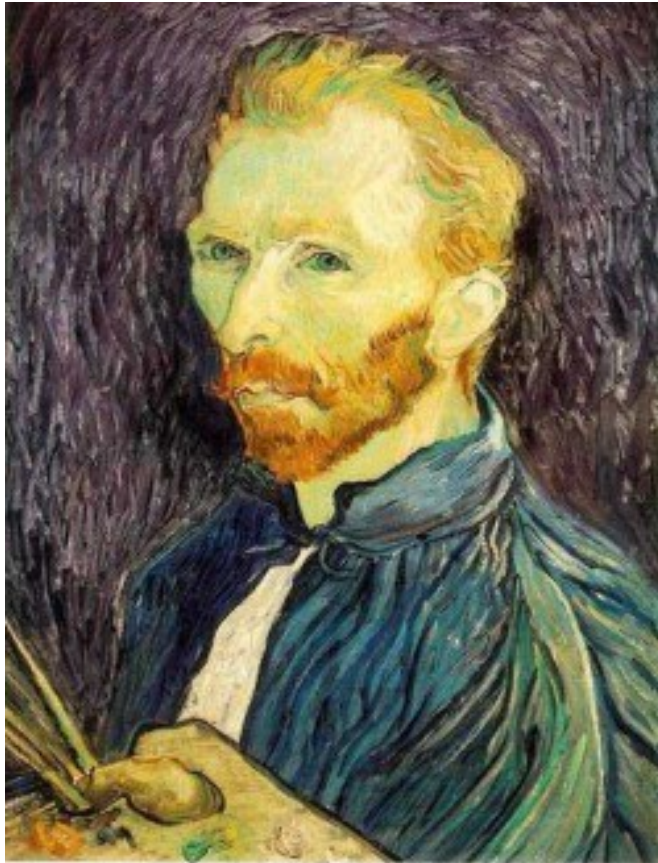


Bad sampling

# 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



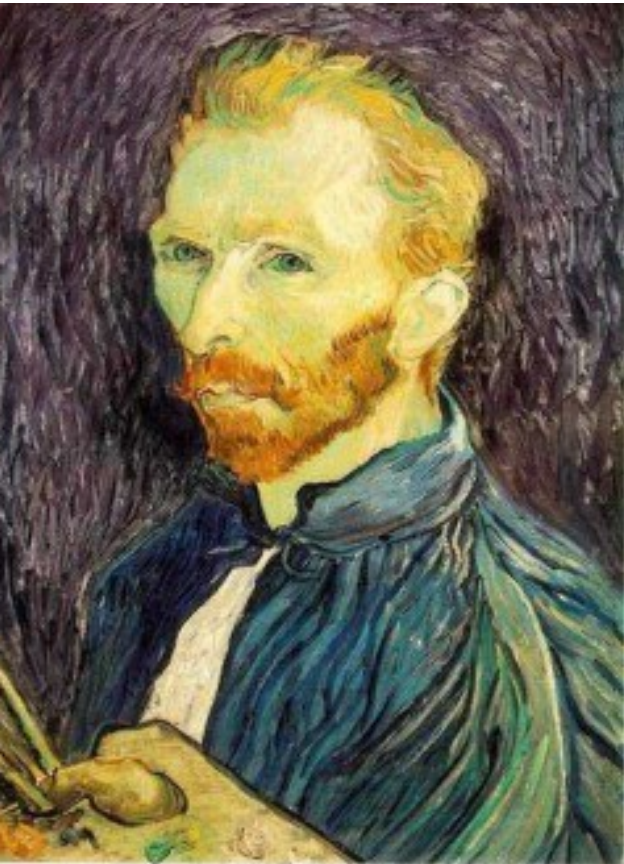
G 1/4



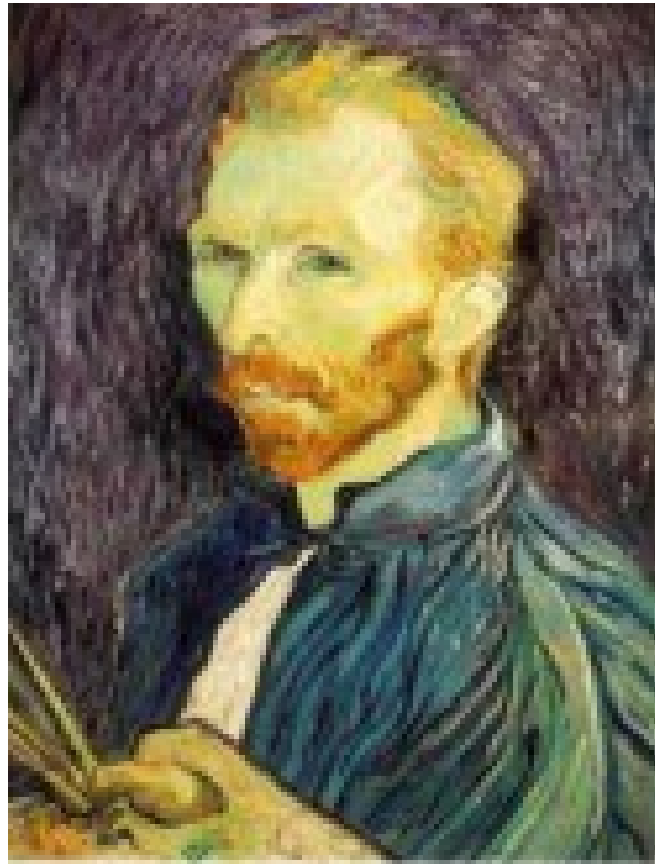
G 1/8

- Solution: filter the image, *then* subsample

# Subsampling with Gaussian pre-filtering



Gaussian 1/2



G 1/4

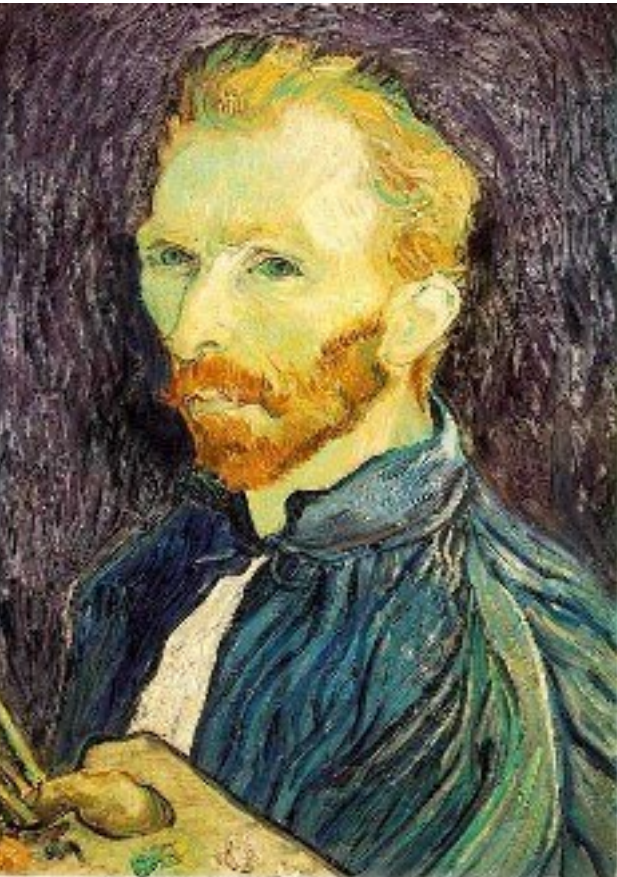


G 1/8

- Solution: filter the image, *then* subsample



# Compare with...



1/2



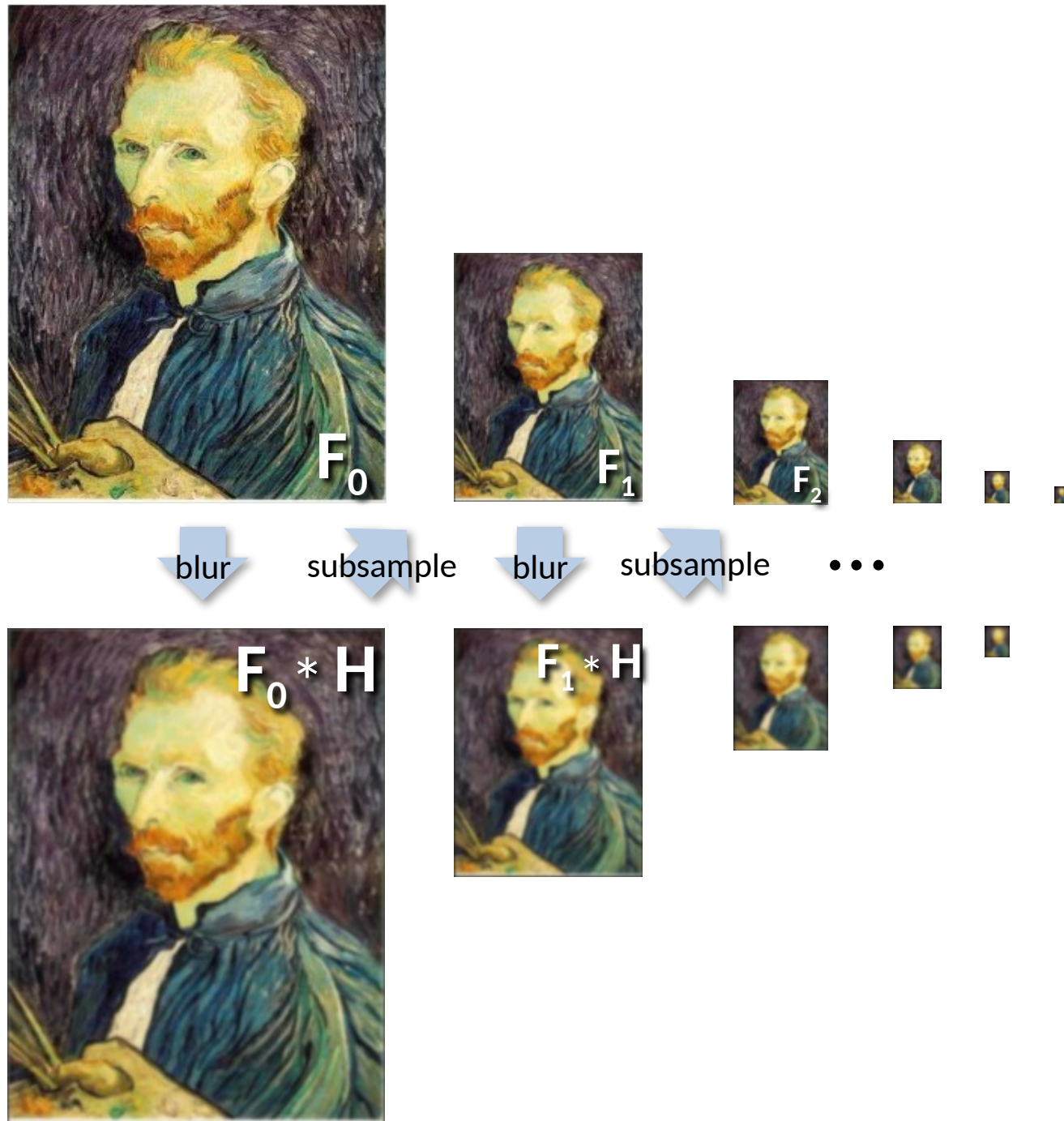
1/4 (2x zoom)



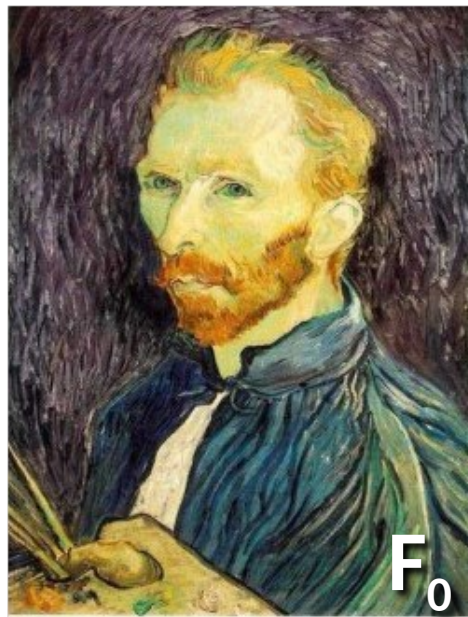
1/8 (4x zoom)

# Gaussian pre-filtering

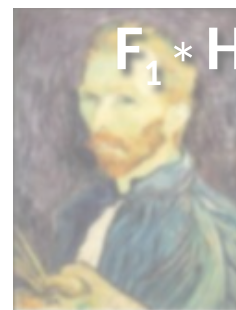
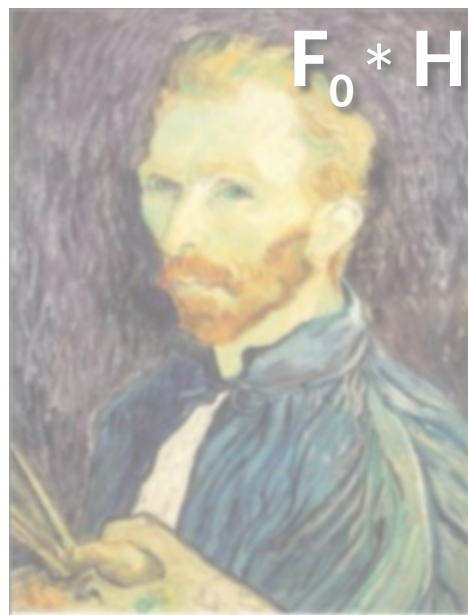
- Solution: filter the image, *then* subsample



*Gaussian  
pyramid*



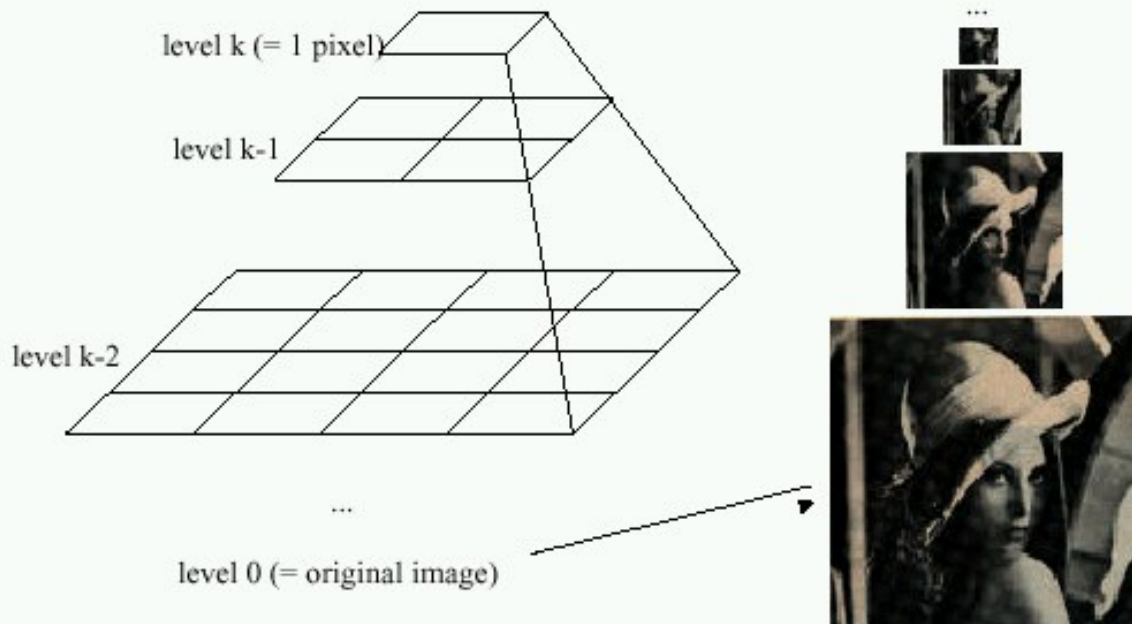
...



# Gaussian pyramids

[Burt and Adelson, 1983]

Idea: Represent  $N \times N$  image as a “pyramid” of  $1 \times 1, 2 \times 2, 4 \times 4, \dots, 2^k \times 2^k$  images (assuming  $N = 2^k$ )



- In computer graphics, a *mip map* [Williams, 1983]
- A precursor to *wavelet transform*

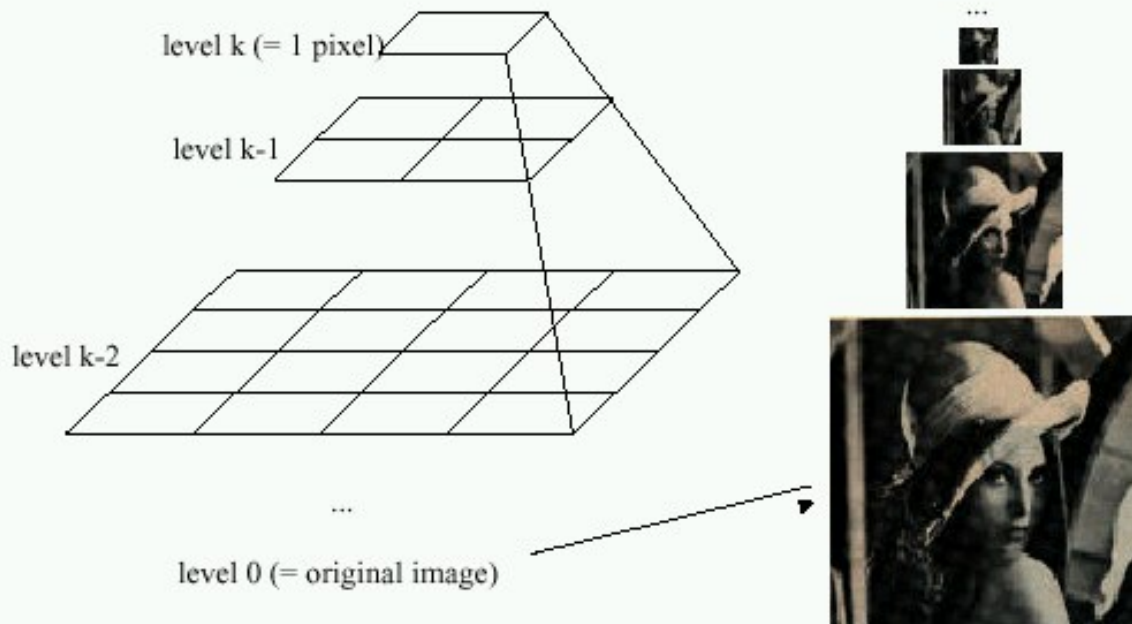
Gaussian Pyramids have all sorts of applications in computer vision



# Gaussian pyramids

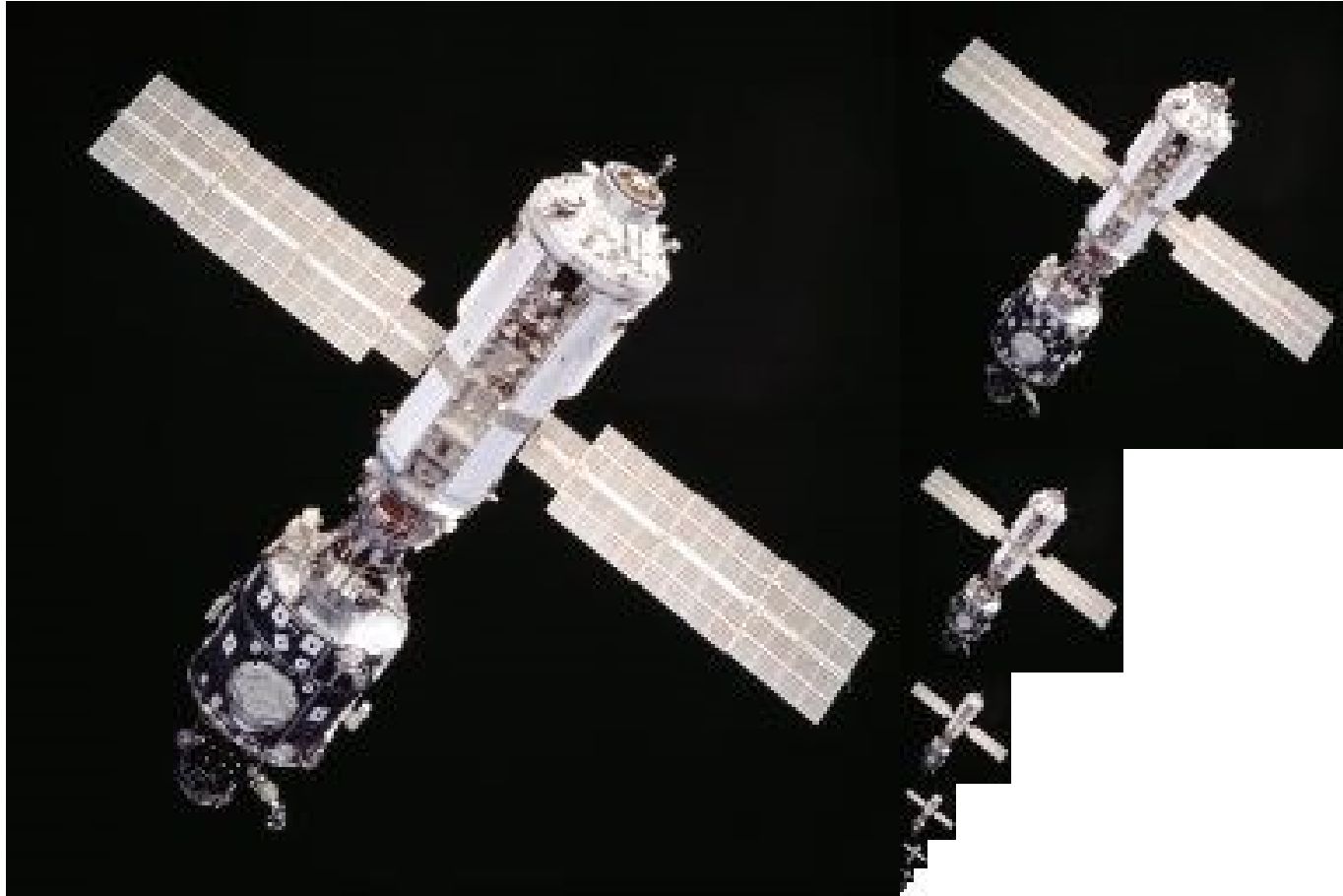
[Burt and Adelson, 1983]

Idea: Represent  $N \times N$  image as a “pyramid” of  $1 \times 1, 2 \times 2, 4 \times 4, \dots, 2^k \times 2^k$  images (assuming  $N = 2^k$ )



- How much space does a Gaussian pyramid take compared to the original image?

# Gaussian Pyramid



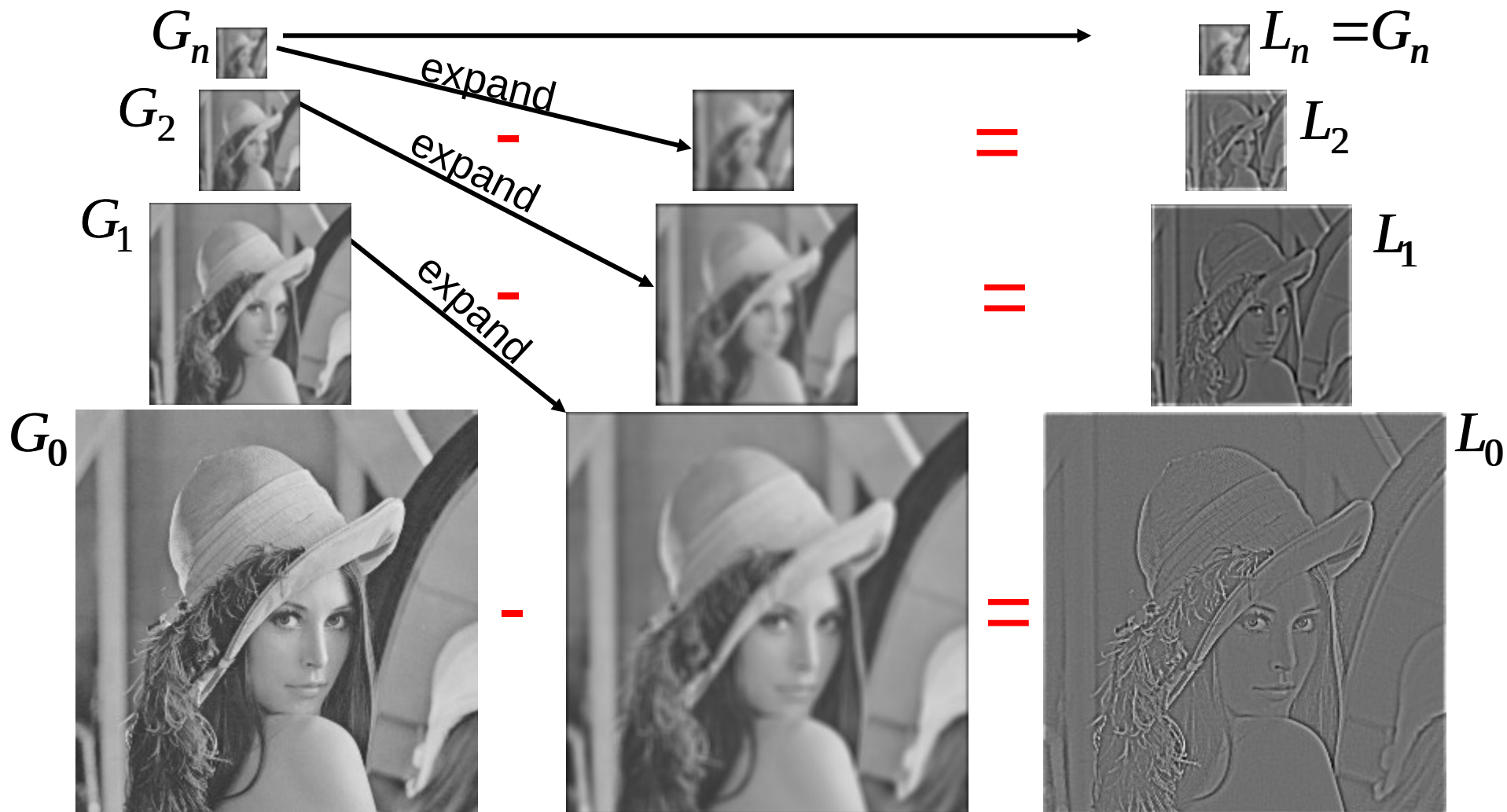
# The Laplacian Pyramid

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


Gaussian Pyramid

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

Laplacian Pyramid



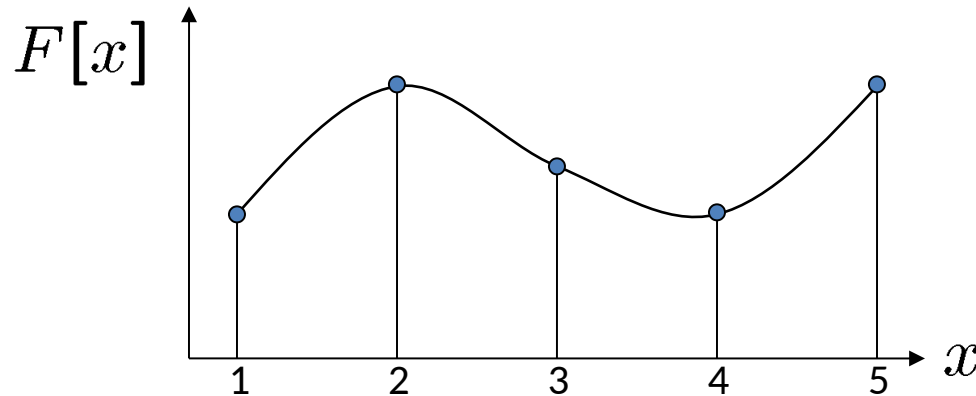
# Upsampling

- This image is too small for this screen: 
- How can we make it 10 times as big?
- Simplest approach:  
    repeat each row  
    and column 10 times
- (“Nearest neighbor interpolation”)





# Image interpolation



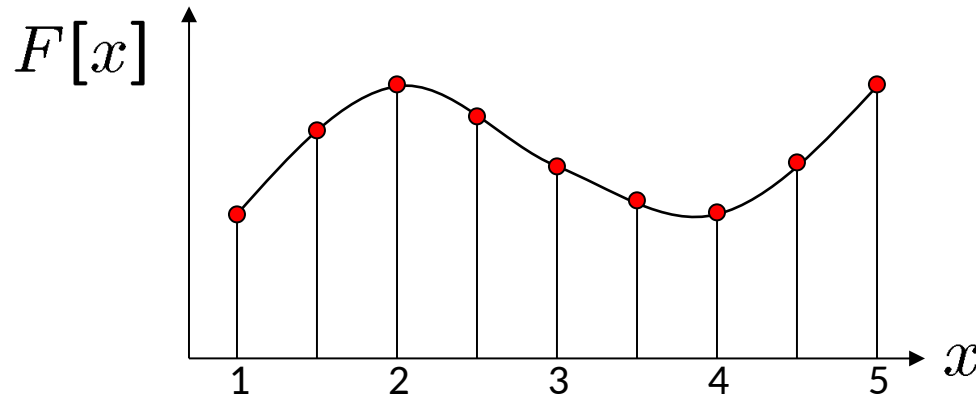
$d = 1$  in this example

Recall how a digital image is formed

$$F[x, y] = \text{quantize}\{f(xd, yd)\}$$

- It is a discrete point-sampling of a continuous function
- If we could somehow reconstruct the original function, any new image could be generated, at any resolution and scale

# Image interpolation



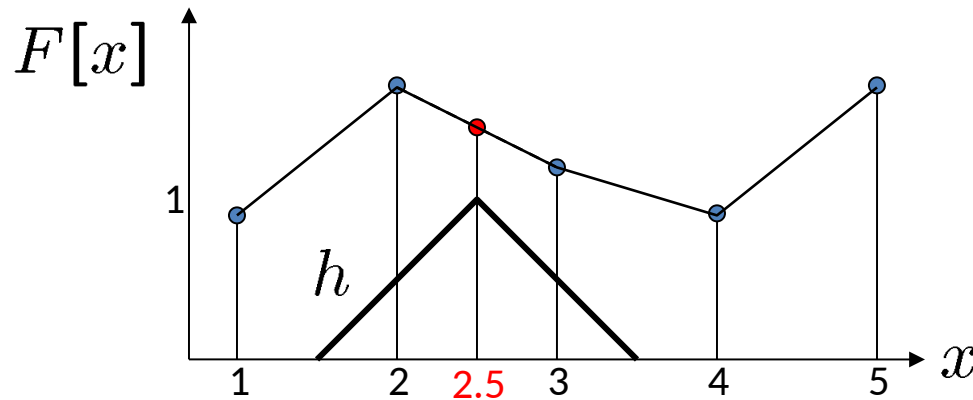
$d = 1$  in this example

Recall how a digital image is formed

$$F[x, y] = \text{quantize}\{f(xd, yd)\}$$

- It is a discrete point-sampling of a continuous function
- If we could somehow reconstruct the original function, any new image could be generated, at any resolution and scale

# Image interpolation



$d = 1$  in this example

- What if we don't know  $f$  ?

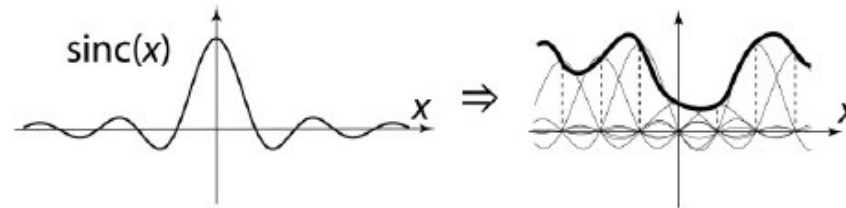
- Guess an approximation:  $\tilde{f}$
- Can be done in a principled way: filtering
- Convert  $F$  to a continuous function:

$$f_F(x) = F\left(\frac{x}{d}\right) \text{ when } \frac{x}{d} \text{ is an integer, } 0 \text{ otherwise}$$

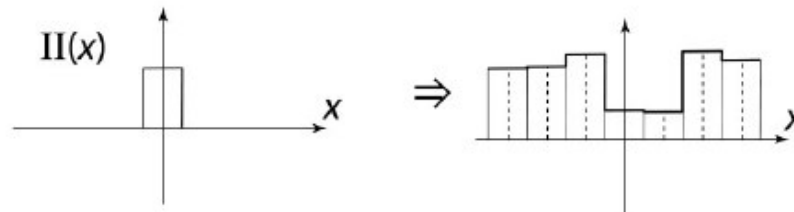
- Reconstruct by convolution with a *reconstruction filter*,  $h$

$$\tilde{f} = h * f_F$$

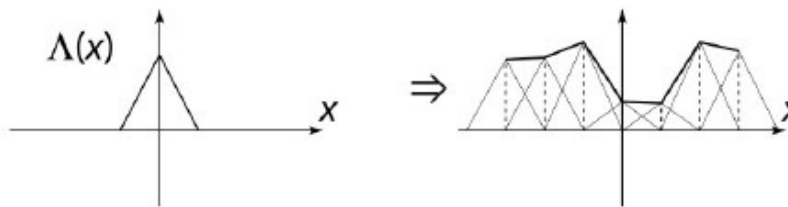
# Image interpolation



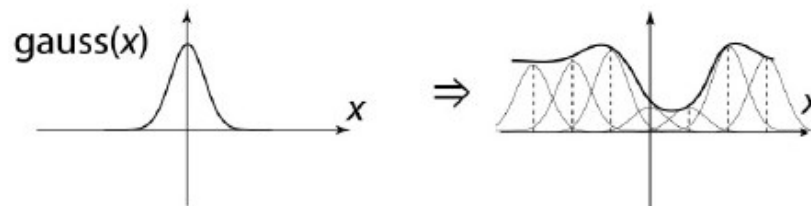
“Ideal” reconstruction



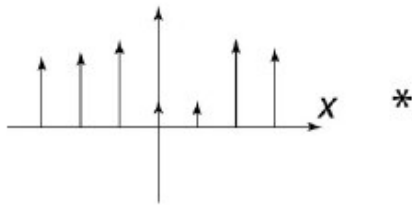
Nearest-neighbor interpolation



Linear interpolation



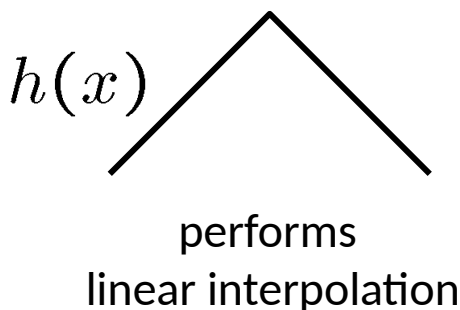
Gaussian reconstruction



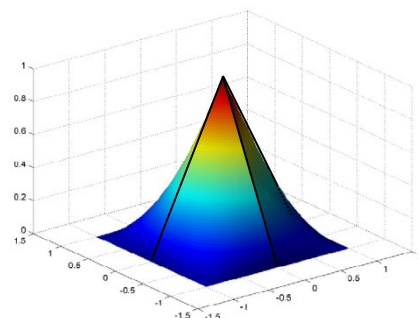


# Reconstruction filters

- What does the 2D version of this hat function look like?



$h(x, y)$



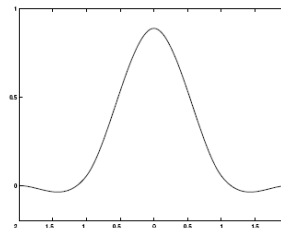
(tent function) performs  
**bilinear interpolation**

Often implemented without cross-correlation

- E.g., [http://en.wikipedia.org/wiki/Bilinear\\_interpolation](http://en.wikipedia.org/wiki/Bilinear_interpolation)

Better filters give better resampled images

- Bicubic** is common choice



Cubic reconstruction filter

$$r(x) = \frac{1}{6} \begin{cases} (12 - 9B - 6C)|x|^3 + (-18 + 12B + 6C)|x|^2 + (6 - 2B) & |x| < 1 \\ ((-B - 6C)|x|^3 + (6B + 30C)|x|^2 + (-12B - 48C)|x| + (8B + 24C)) & 1 \leq |x| < 2 \\ 0 & \text{otherwise} \end{cases}$$

# Image interpolation

Original image:  x 10



Nearest-neighbor interpolation



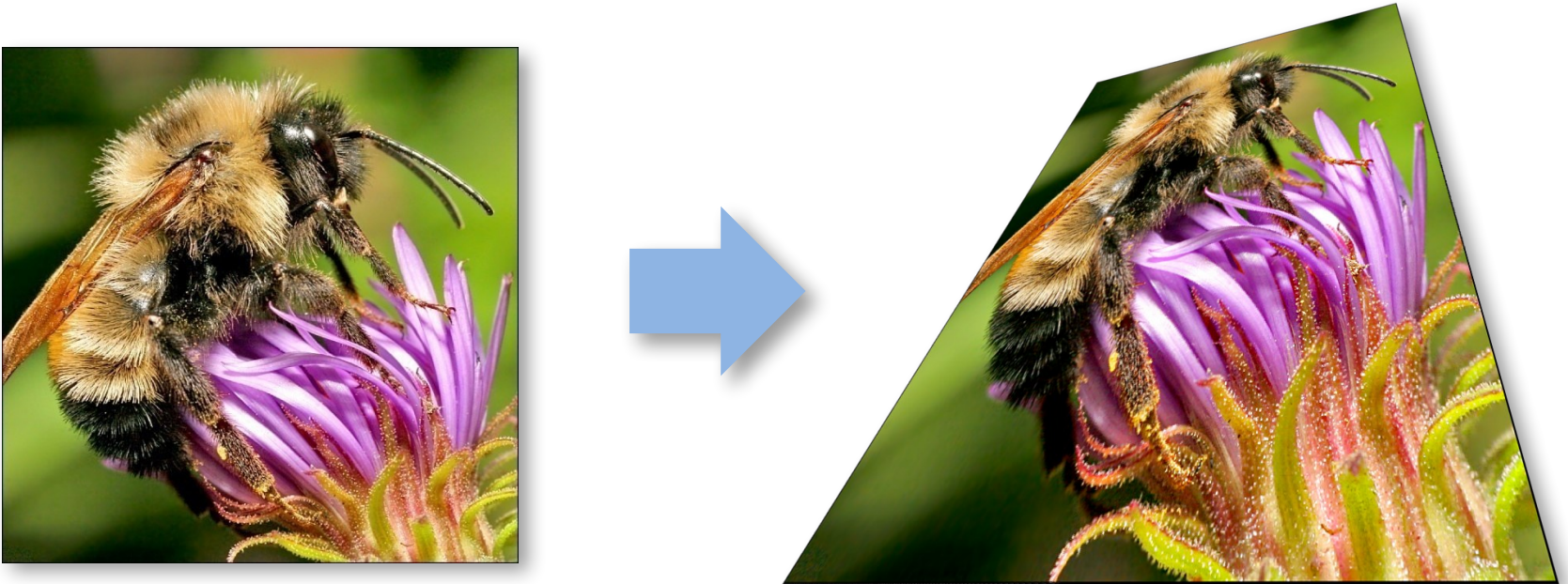
Bilinear interpolation




Bicubic interpolation

# Image interpolation

Also used for *resampling*



# Raster to Vector Graphics

Easily Convert Bitmap Images To Clean Vector Art

[Log In](#) | [Sign Up](#)  
[Forgot Password](#)

[Home](#) | [Online](#) | [Hand-Tracing](#) | [Desktop](#) | [Pricing](#) | [Support](#)

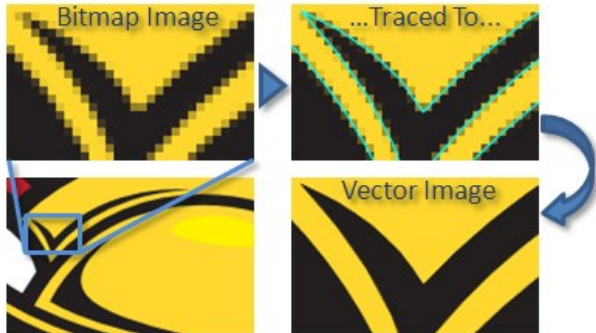
Welcome | [Comparisons](#) | [Examples](#) | [Buzz](#)  [Search](#)


### Easily Convert Bitmap Images To Clean Vector Art

Automatically convert bitmap images like JPEGs, GIFs and PNGs to the crisp, clean, scalable vector art of EPS, SVG, and PDF with the world's best auto-tracing software.

Use the convenient service online by uploading your image below, or start tracing images offline with the [Desktop Edition](#).

Save time and money getting your artwork ready to print, embroider, cut and more!






#### Online

[Subscribe](#) | [Pricing](#)

Upload your bitmap image to convert it to a vector image (EPS, SVG and PDF).

**Free to try, no signup necessary, and results ready right away.** [Learn more](#)

 **Upload Image To Trace...**



#### Desktop

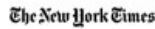

[Buy](#) | [Pricing](#)




Convert images offline to EPS, SVG, PDF, AI, DXF and EMF. Works seamlessly with Illustrator, Corel, and others.

**Learn More**

**Start here!**

**Your product is AMAZING!**  
- Shawn, Reminderband

**Featured in**  
 

**Customers**  
    

**Vector Magic is**

- More accurate
- Easier to use
- Faster

than the alternatives.  
[See comparisons](#)

#### Downloads

 [Desktop Application](#)

Languages: [English](#), [Deutsch](#), [Español](#), and [Français](#)

#### Support

- [FAQ](#)
  - [What Are Vectors?](#)
  - [What Are Bitmaps?](#)
  - [Why Use Vectors?](#)
  - [What Is Vectorization?](#)
- [Uses For Vector Images](#)
- [Useful Tools](#)
- [File Formats](#)

#### Tutorials

- [How To Use Vector Magic](#)
- [Vectorizing Artwork/Logos](#)
- [Vectorizing Photos](#)
- [Vectorizing Scans](#)
- [Tips And Tricks](#)
- [Building a New Font](#)

#### Need help?

- [Online contact form](#)



# Depixelating Pixel Art



# Modern methods



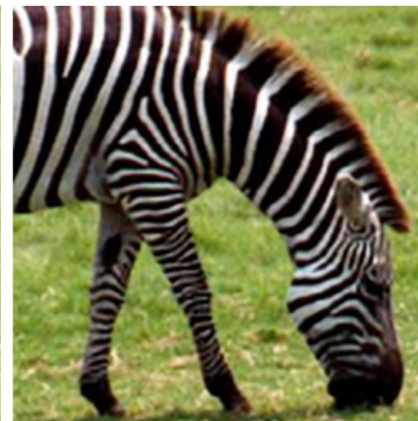
(a) Bicubic



(b) SRCNN



(c) A+



(d) RAISR



(e) Bicubic



(f) SRCNN



(g) A+



(h) RAISR

From Romano, et al: RAISR: Rapid and Accurate Image Super Resolution,  
<https://arxiv.org/abs/1606.01299>