CS5670: Computer Vision
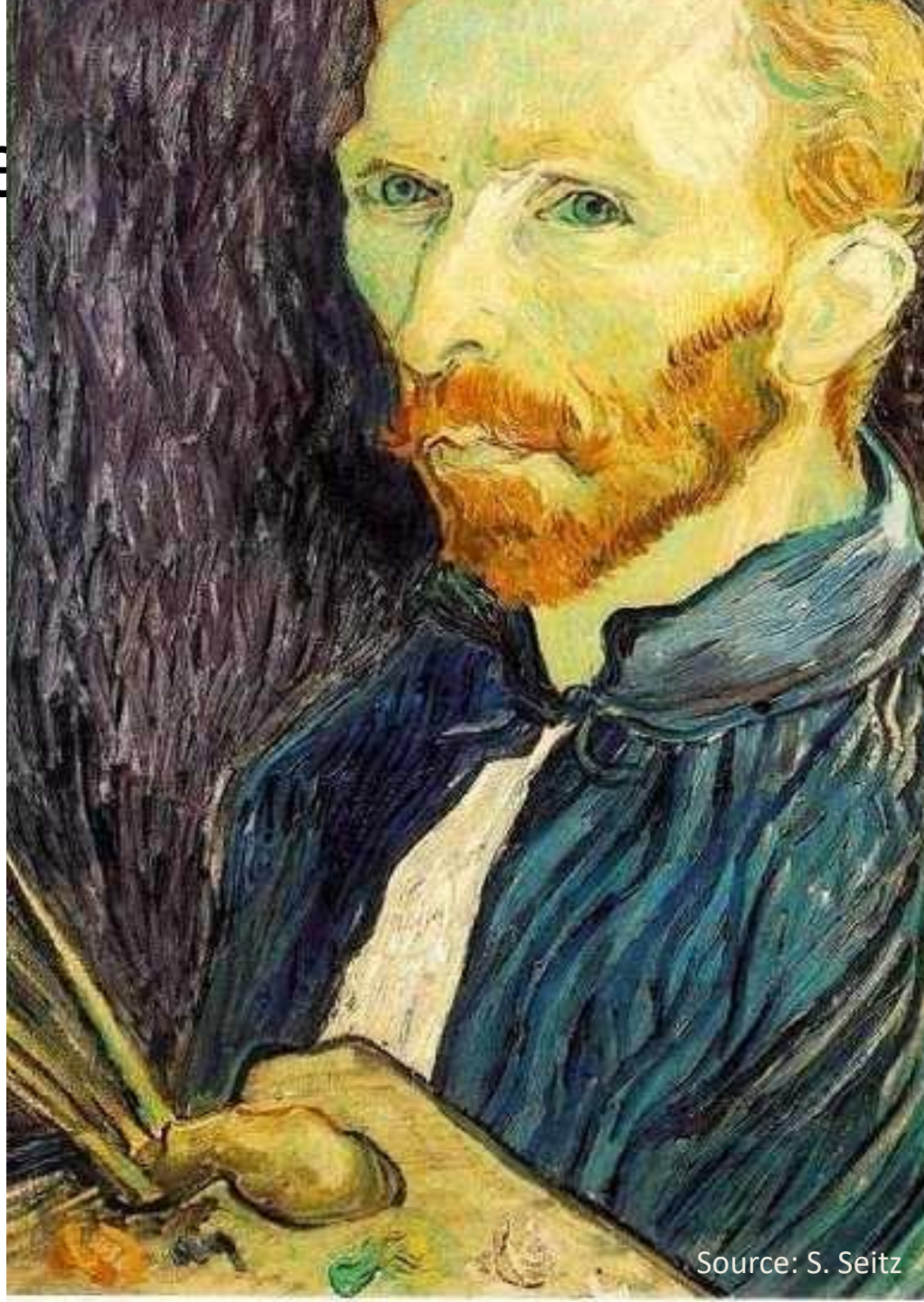## Noah Snavely

Image Resampling & Interpolation

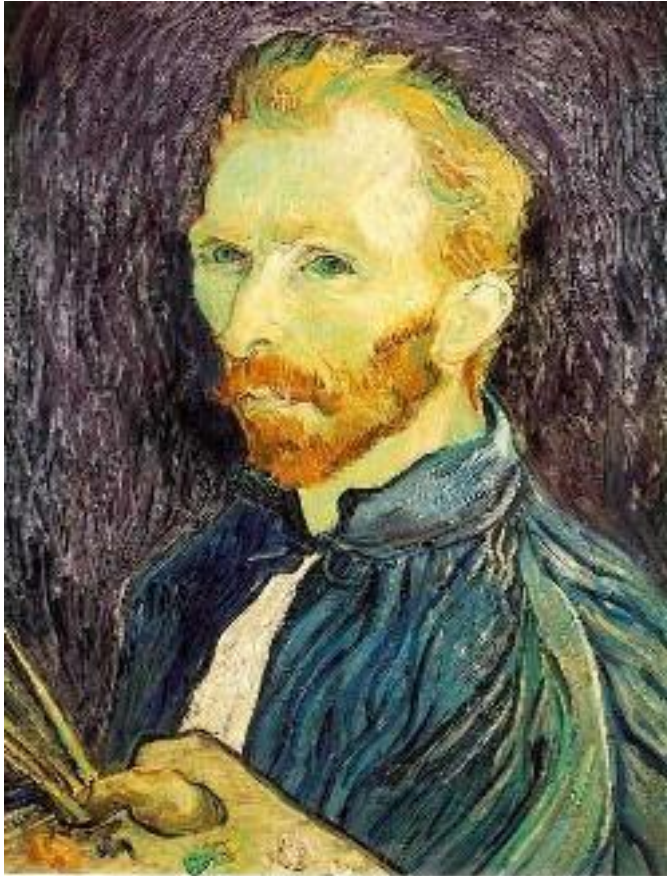# Image

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

Image sub-sampling

1/4

1/8

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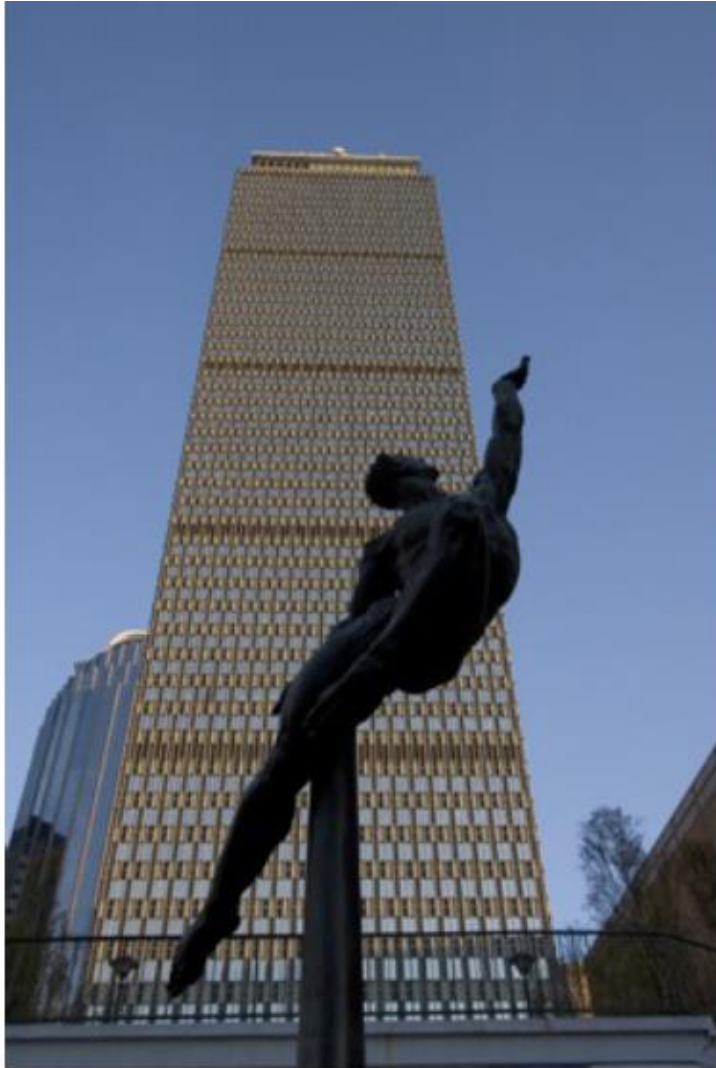- called *image sub-sampling*

Image sub-sampling

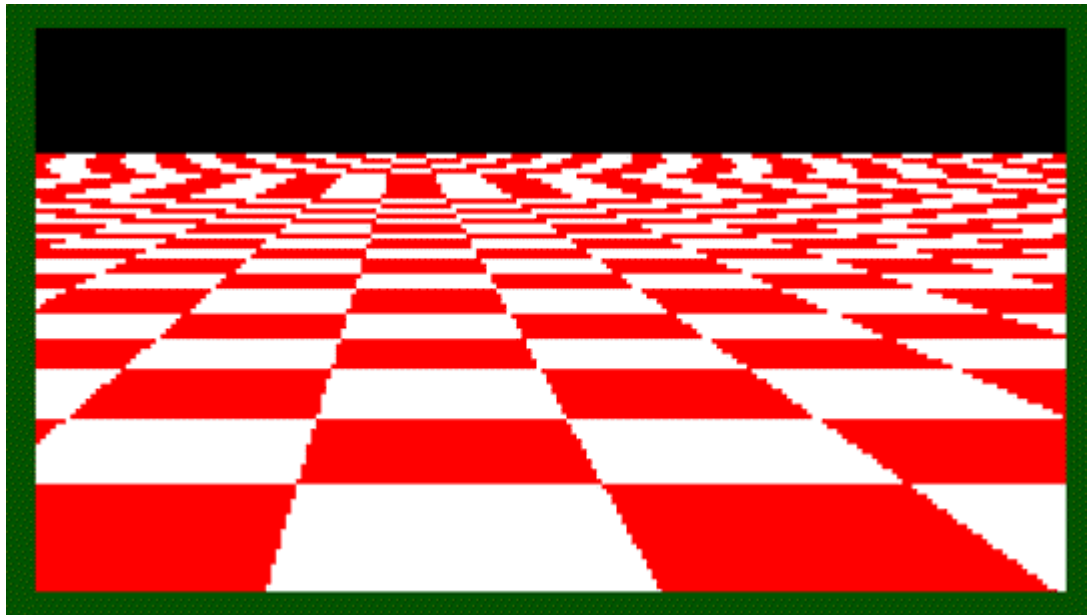

1/2   1/4 (2x zoom)   1/8 (4x zoom)
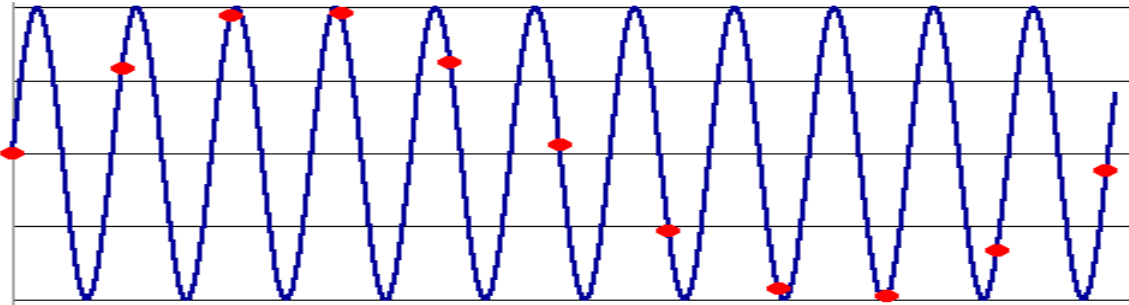
Why does this look so crufty?

Image sub-sampling – another example

Source: F. Durand

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…
    - "But what is the Fourier Transform? A visual introduction." https://www.youtube.com/watch?v=spUNpyF58BY&t=444s
- To avoid aliasing:
    - sampling rate ≥ 2 * max frequency in the image
        - said another way: ≥ two samples per cycle
    - This minimum sampling rate is called the **Nyquist rate**

Source: L. Zhang

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):
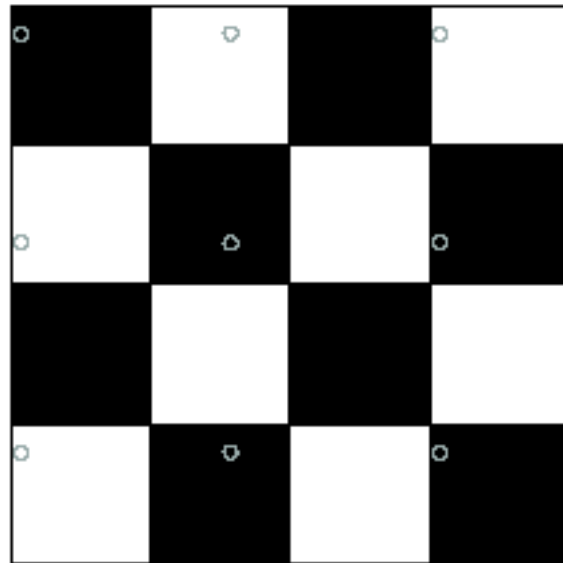
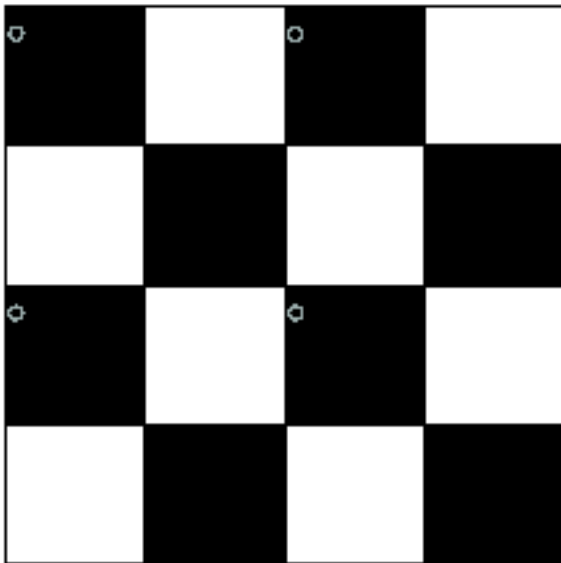frame 0   frame 1   frame 2   frame 3   frame 4

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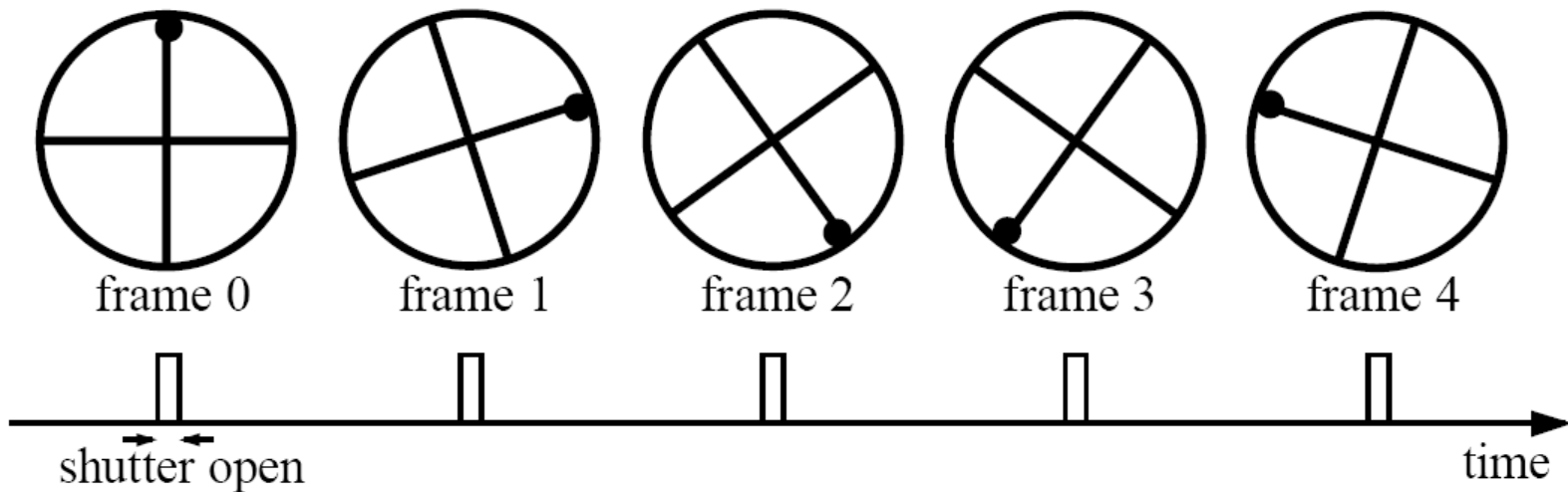
Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

(See http://www.michaelbach.de/ot/mot-wagonWheel/index.html)

Source: L. Zhang

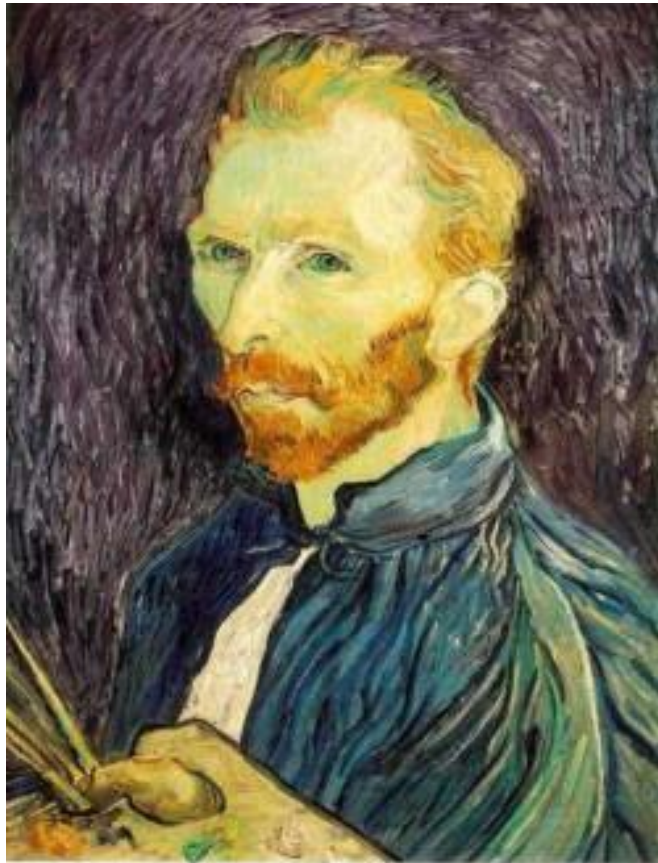# Nyquist limit – 2D example
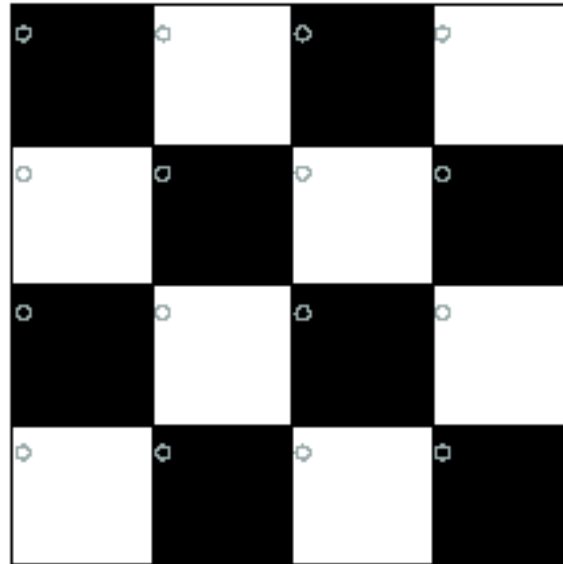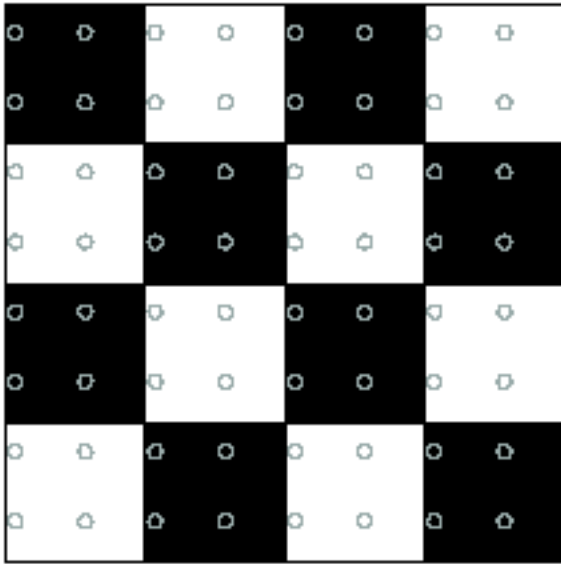
Good sampling

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

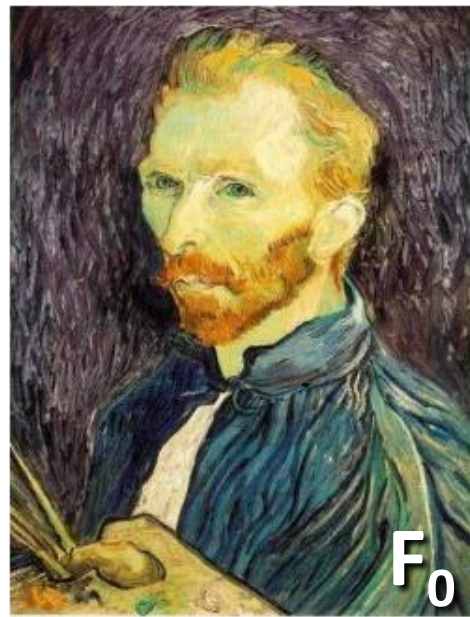- Solution: filter the image, *then* subsample
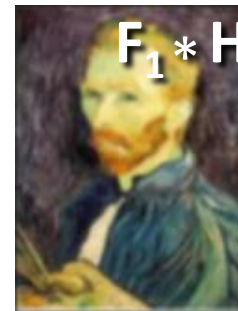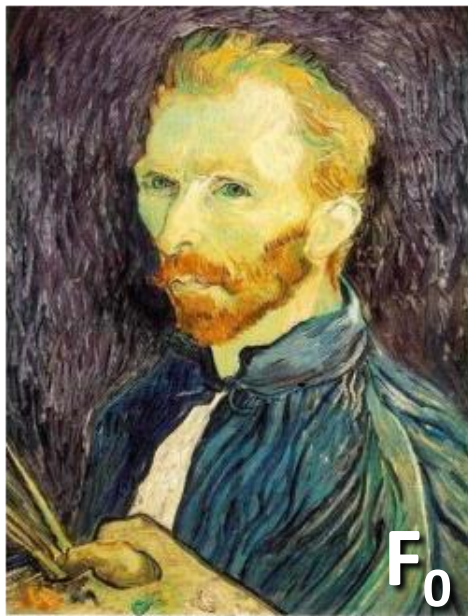
Compare with…

1/2   1/4 (2x zoom)   1/8 (4x zoom)

Source: S. Seitz

Gaussian pre-filtering

- Solution: filter the image, *then* subsample
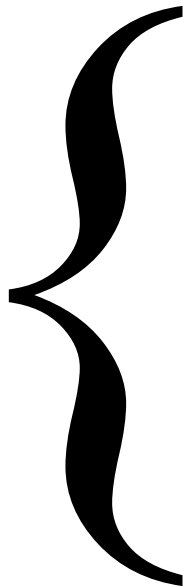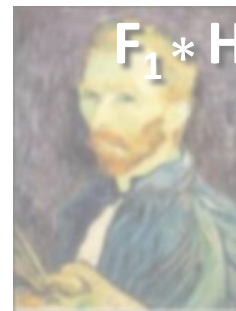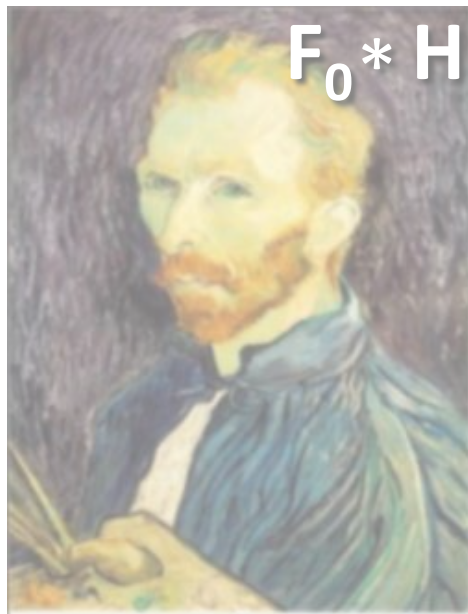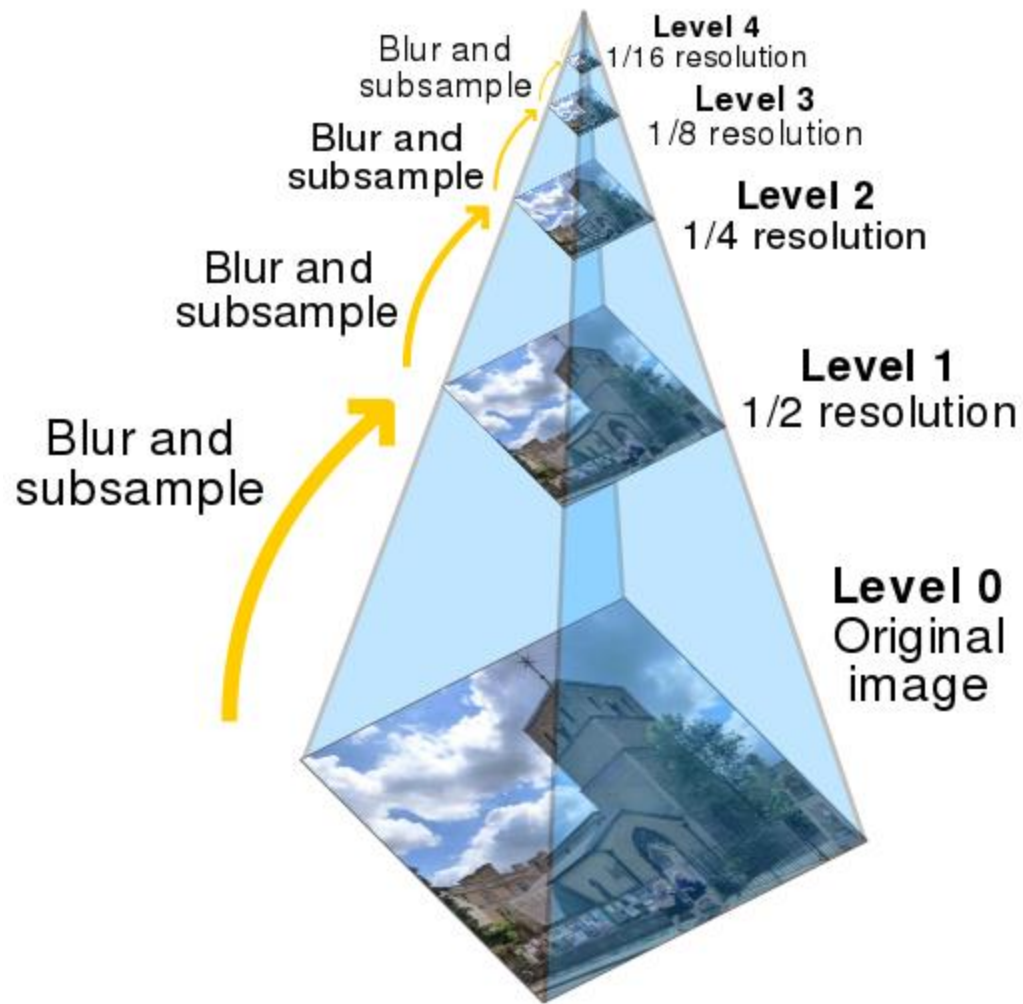
F_0

F_1

F_2

blur   subsample   blur   subsample   · · ·

$F_0 * H$

$F_1 * H$

Gaussian pyramid

F_0

F_1

F_2

blur   subsample   blur   subsample $\cdots$

$F_0 * H$

$F_1 * H$

Blur and subsample

Blur and subsample

Blur and subsample

Blur and subsample

Level 4
1/16 resolution

Level 3
1/8 resolution

Level 2
1/4 resolution

Level 1
1/2 resolution

Level 0
Original image

Gaussian pyramids
[Burt and Adelson, 1983]

Idea: Represent NxN image as a "pyramid" of
1x1, 2x2, 4x4,..., 2^kx2^k images (assuming N=2^k)

level k (= 1 pixel)

level k-1

level k-2

...

level 0 (= original image)
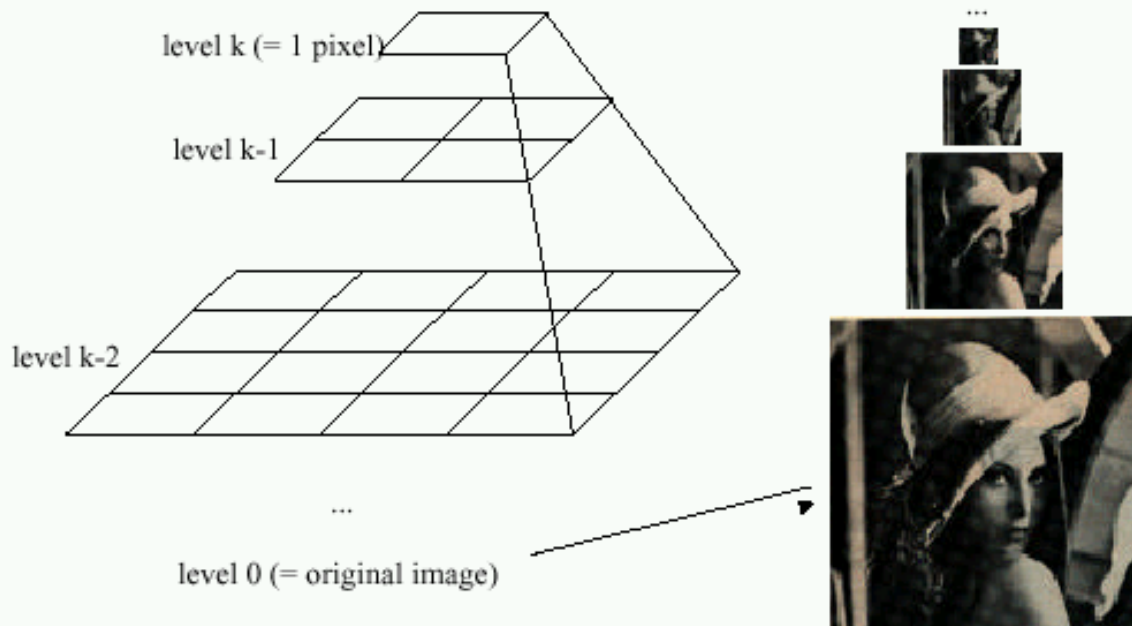
- In computer graphics, a *mip map* [Williams, 1983]
- A precursor to *wavelet transform*
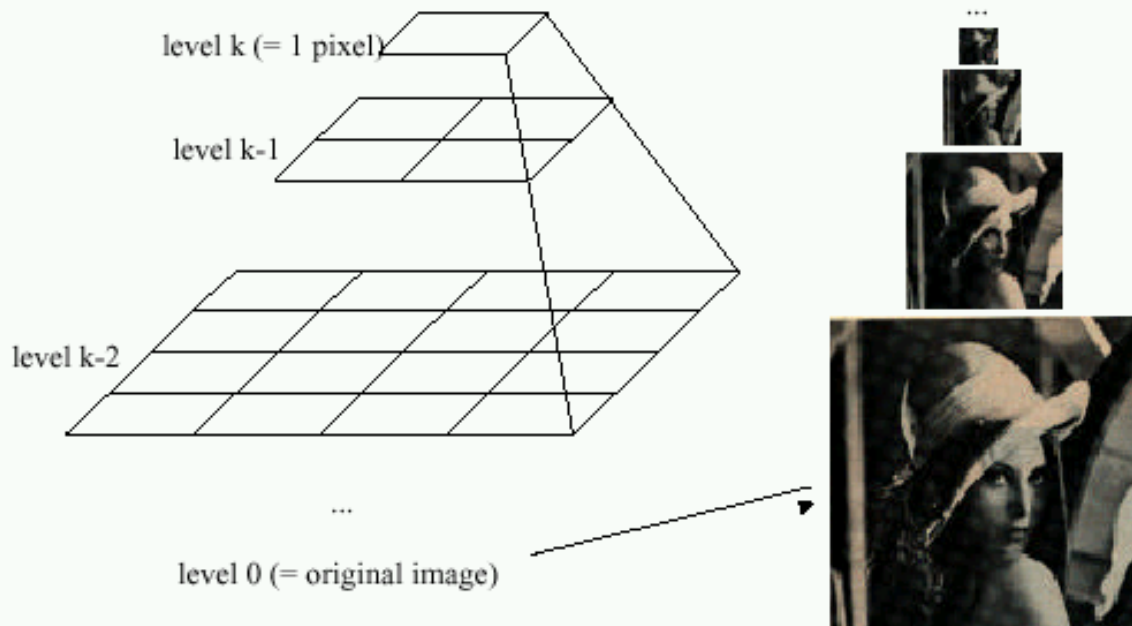
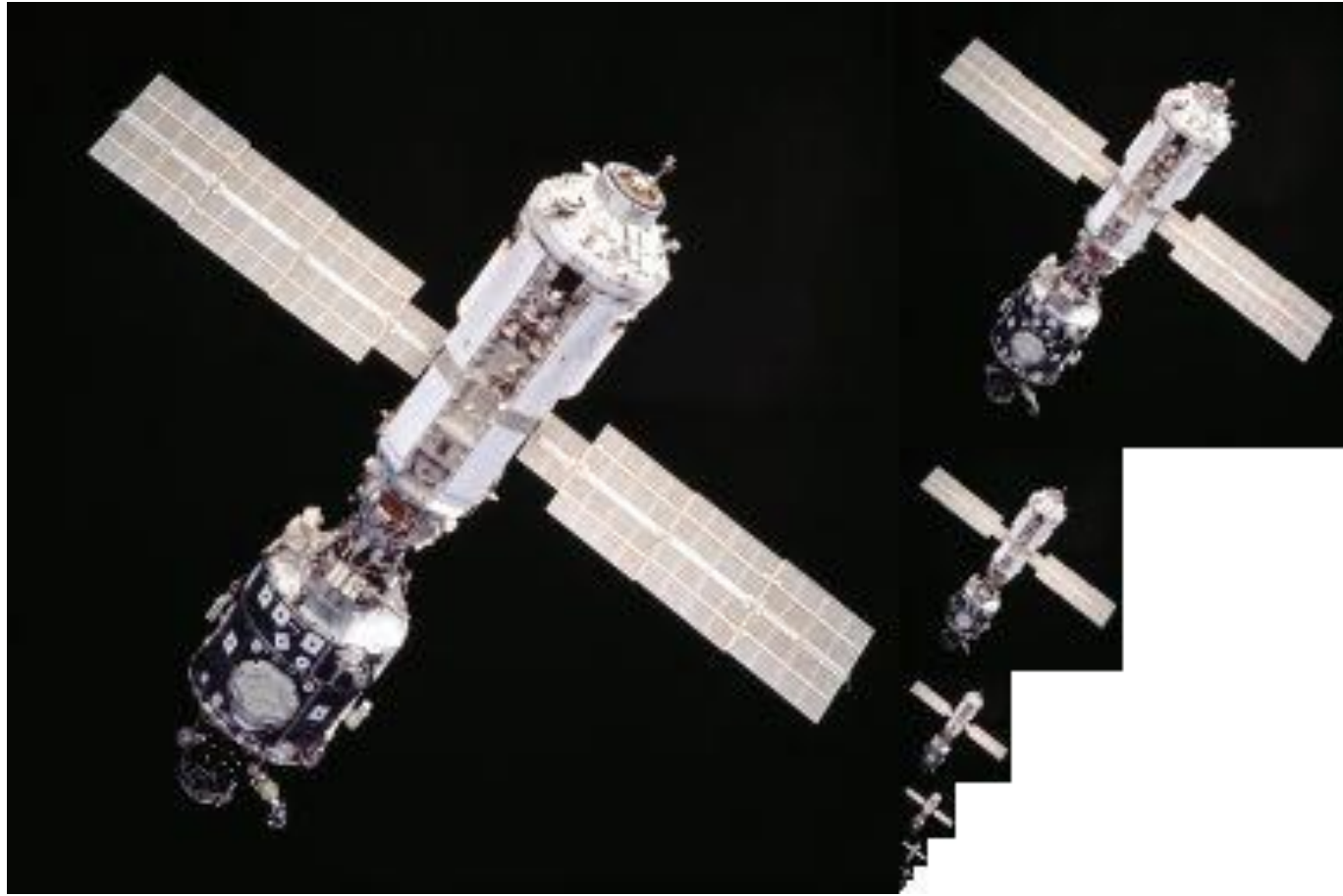Gaussian Pyramids have all sorts of applications in computer vision

Gaussian pyramids
[Burt and Adelson, 1983]

Idea: Represent NxN image as a "pyramid" of
          1x1, 2x2, 4x4,…, 2^kx2^k images (assuming N=2^k)

level k (= 1 pixel)

level k-1

level k-2

…

level 0 (= original image)

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

Gaussian Pyramid

Questions?

Upsampling
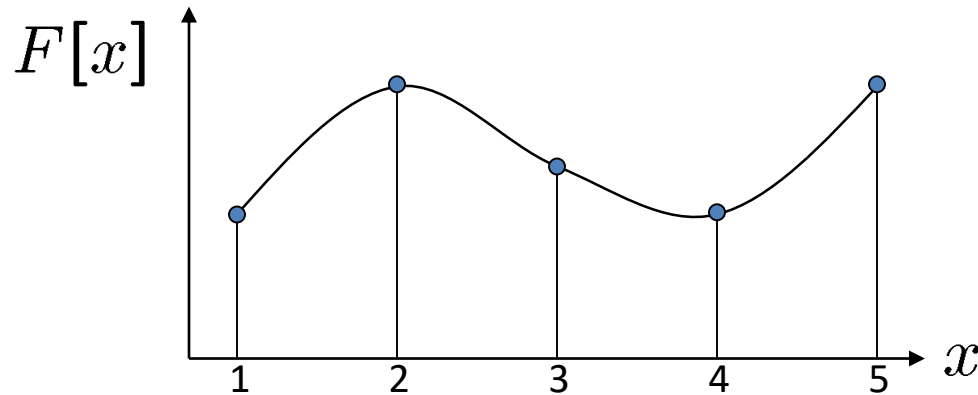
- This image is too small for this screen:

- How can we make it 10 times as big?

- Simplest approach:
  repeat each row
  and column 10 times

- ("Nearest neighbor
  interpolation")

Image interpolation

$F[x]$

d = 1 in this example

Recall that a digital images is formed as follows:

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

- 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
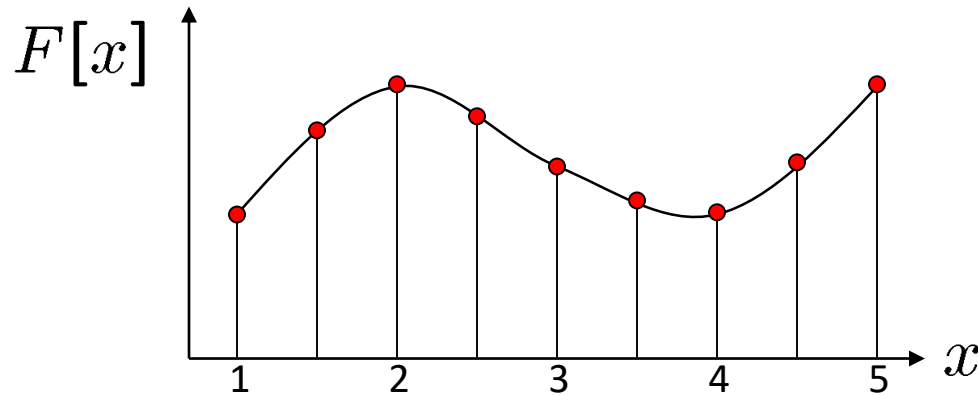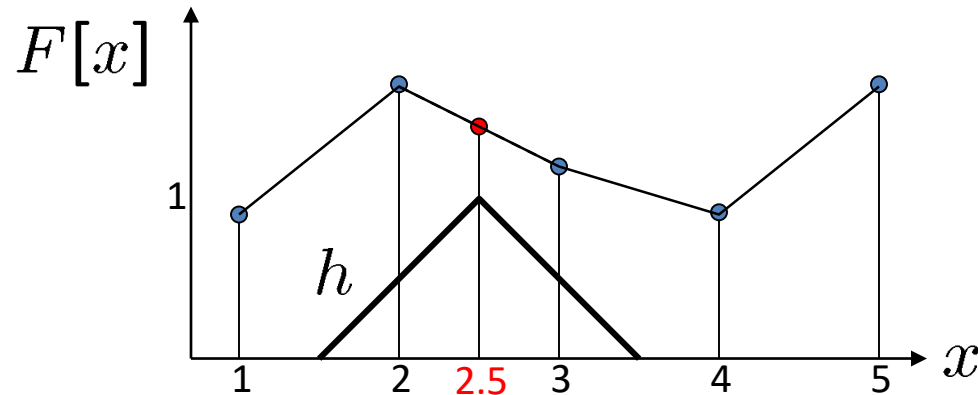
Image interpolation

$F[x]$

d = 1 in this example

Recall that a digital images is formed as follows:

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

- 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 ?
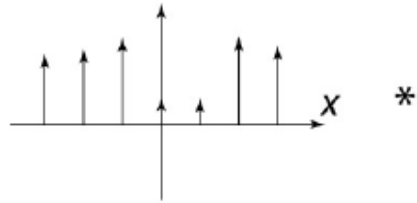 - Guess an approximation: $\tilde{f}$
  - Can be done in a principled way: filtering
  - Convert F to a continuous function:
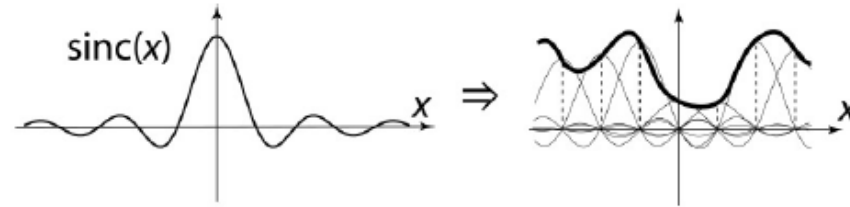 $$f_F(x) = F(\tfrac{x}{d}) \text{ when } \tfrac{x}{d} \text{ is an integer, 0 otherwise}$$
  - Reconstruct by convolution with a *reconstruction filter, h*
    $$\tilde{f} = h * f_F$$
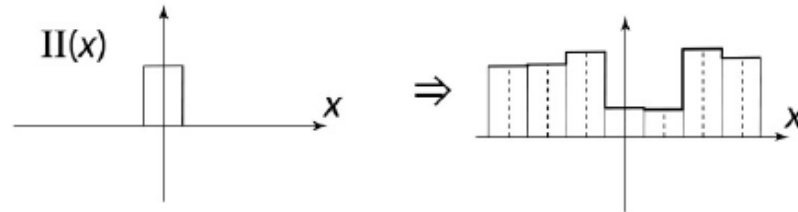
Adapted from: S. Seitz
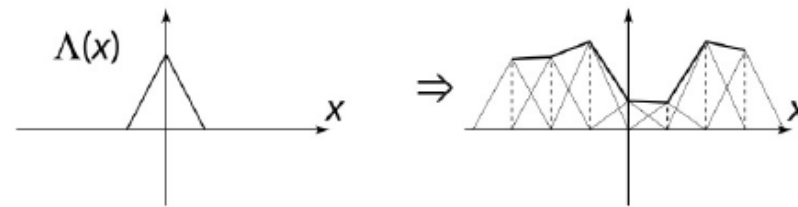
Image interpolation

sinc(x) $\Rightarrow$ "Ideal" reconstruction

II(x) $\Rightarrow$ Nearest-neighbor interpolation

Λ(x) $\Rightarrow$ Linear interpolation
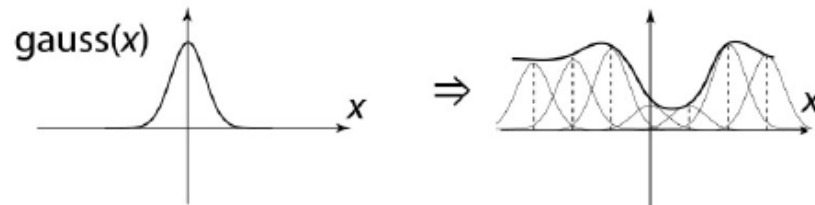
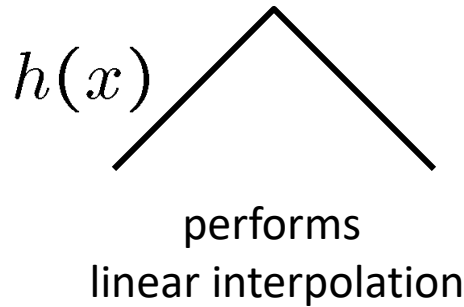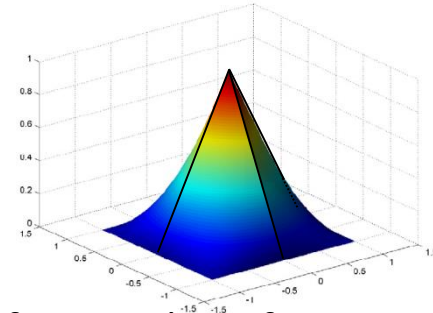gauss(x) $\Rightarrow$ Gaussian reconstruction

Reconstruction filters

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

$h(x)$

performs
linear interpolation

$h(x, y)$

(tent function) performs
bilinear interpolation

Often implemented without cross-correlation

- E.g., http://en.wikipedia.org/wiki/Bilinear_interpolation

Better filters give better resampled images

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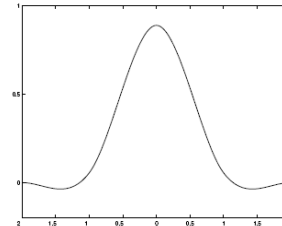
$$r(x) = \frac{1}{6}\begin{cases} (12 - 9B - 6C)|x|^3 + (-18 + 12B + 6C)|x|^2 + (6 - 2B) & |x| < 1 \\ ((-B - 6C)|x|^3 + (6B + 30C)|x|^2 + (-12B - 48C)|x| + (8B + 24C) & 1 \le |x| < 2 \\ 0 & otherwise \end{cases}$$

Cubic reconstruction filter

Image interpolation

Original image: x 10

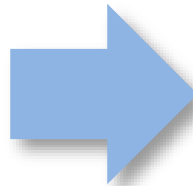Nearest-neighbor interpolation

Bilinear interpolation

Bicubic interpolation

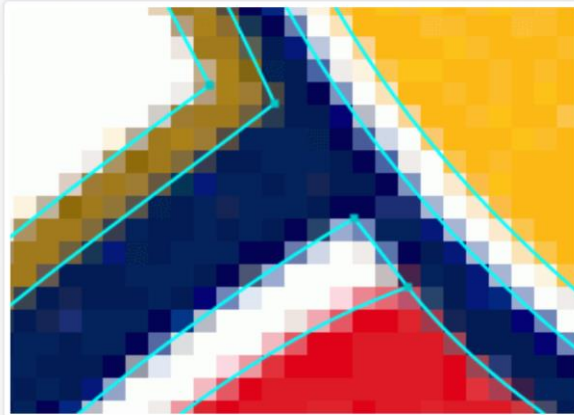# Image interpolation

Also used for *resampling*

Raster-to-vector graphics

Vector Magic

Simply the Best Auto-Tracer in the World

Depixelating Pixel Art

"Yoshi" Input (20 × 30 Pixels) · Our Result · Adobe Live Trace

"Axe Battler" Input (43 × 71 Pixels) · Our Result · Photo Zoom 4

"Bomberman" Input (15 × 23 Pixels) · Our Result · hq4x

"Invaders" Input (11 × 8 Pixels each) · Our Result · Bicubic

"Toad" Input (16 × 27 Pixels) · Our Result · hq4x

"386" Input (25 × 31 Pixels) · Our Result · EPX

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

Questions?