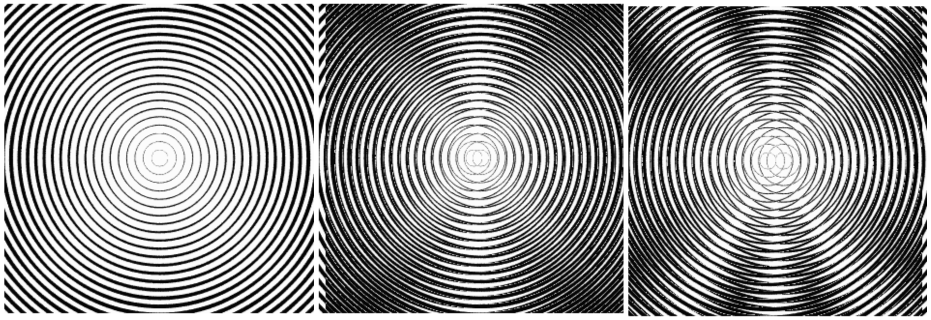
Image Sampling



Moire patterns
- http://www.sandlotscience.com/Moire/Circular_3_Moire.htm

Image Scaling

This image is too big to fit on the screen. How can we reduce it?

How to generate a half-sized version?

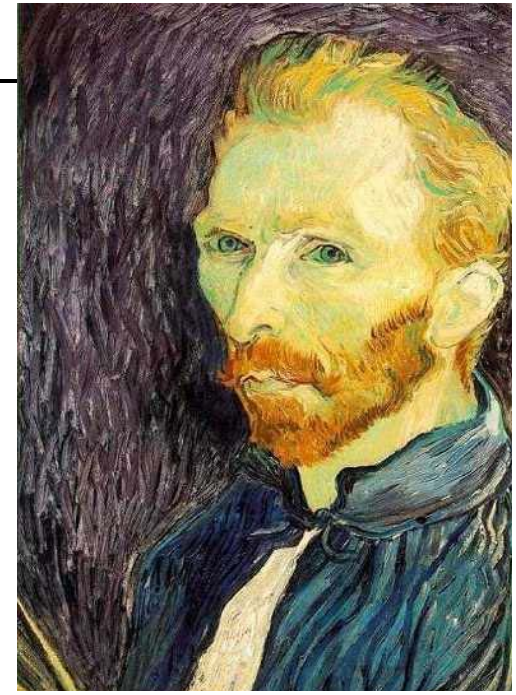
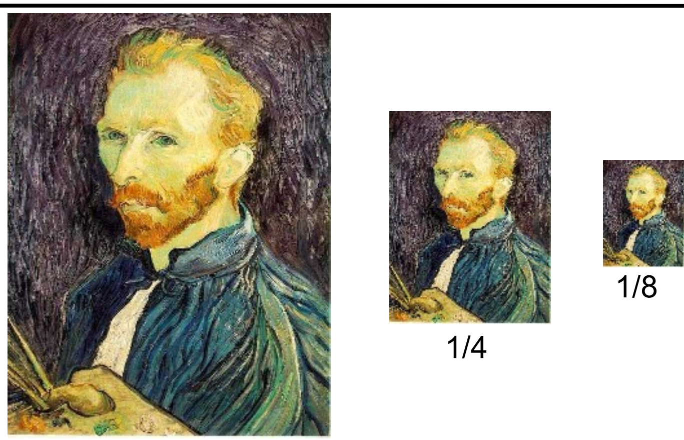


Image sub-sampling

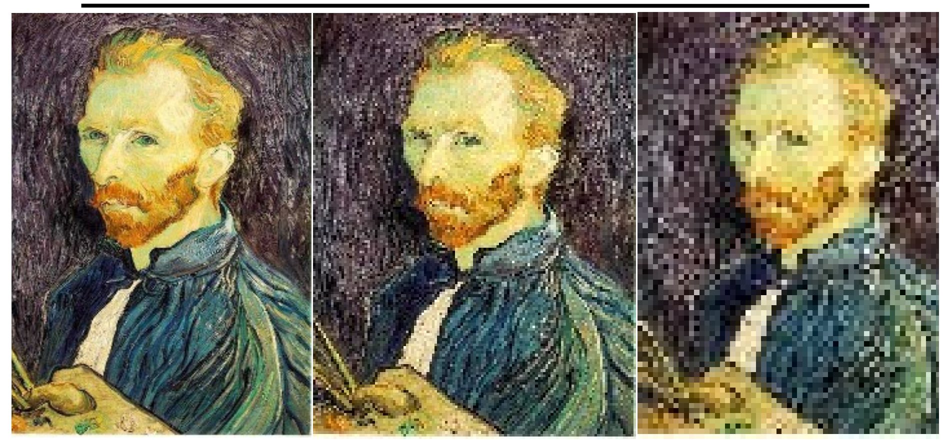


1/4

1/8

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

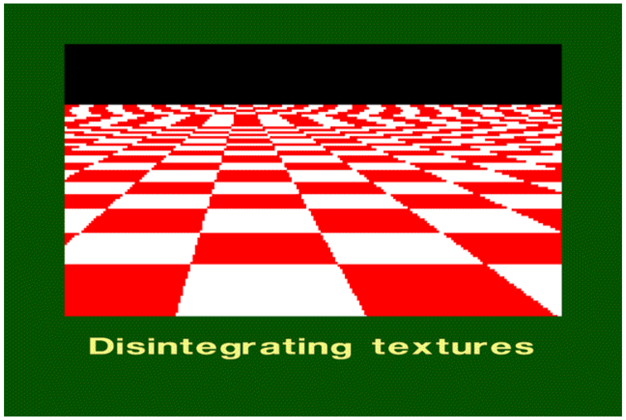
Image sub-sampling



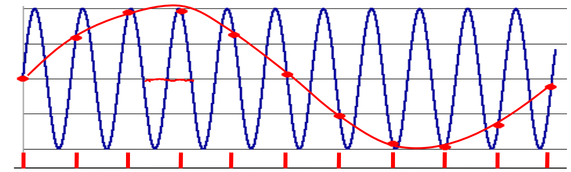
1/2 1/4 (2x zoom) 1/8 (4x zoom)

Why does this look so crufty?

Even worse for synthetic images



Sampling and the Nyquist rate



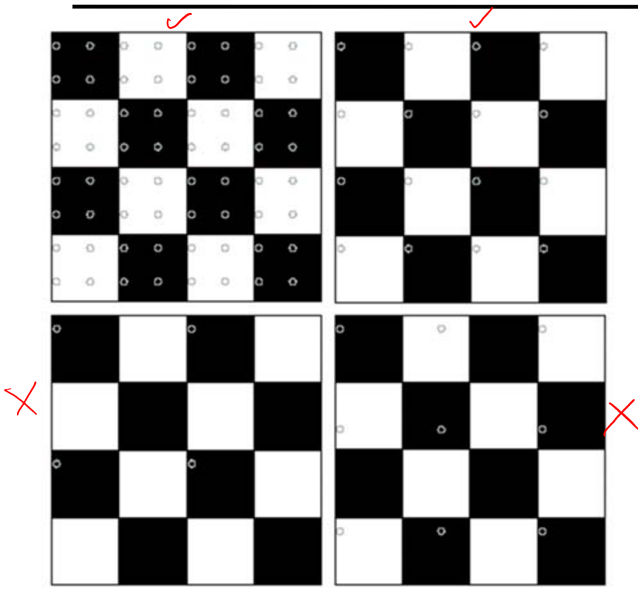
Aliasing can arise when you sample a continuous signal or image

- 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*
- formally, the image contains structure at different scales
  - called “frequencies” in the Fourier domain
- the sampling rate must be high enough to capture the highest frequency in the image

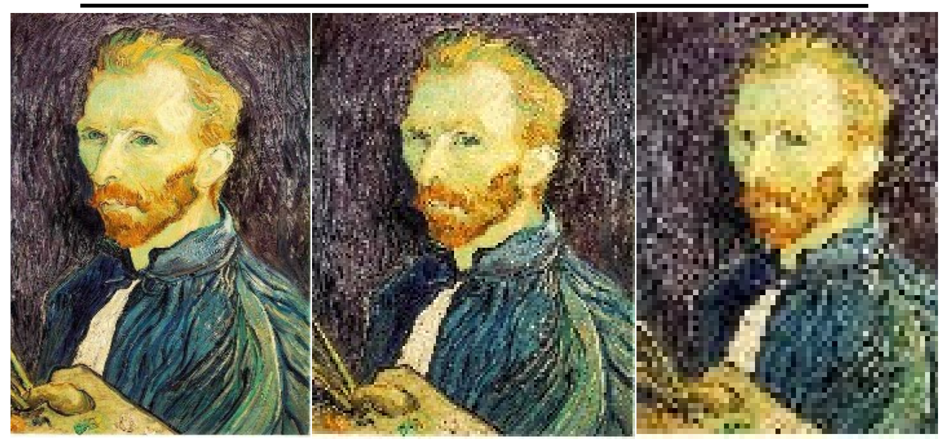
To avoid aliasing:

- sampling rate ≥ 2 * max frequency in the image
  - said another way: ≥ two samples per cycle
- This minimum sampling rate is called the **Nyquist rate**

2D example

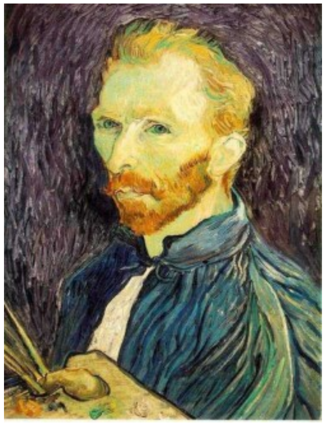


How to fix this?



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

Subsampling with Gaussian pre-filtering



Gaussian 1/2

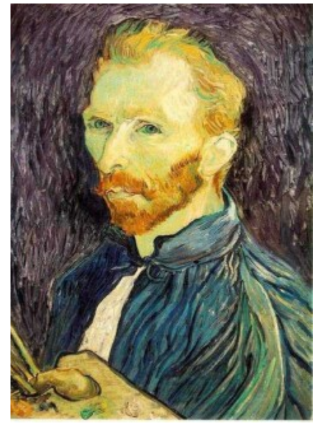
G 1/4

G 1/8

Solution: filter the image, *then* subsample

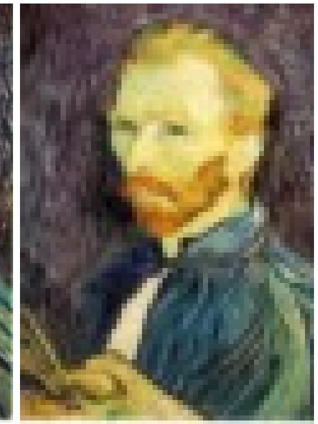
- Filter size should double for each ½ size reduction. Why?

Subsampling with Gaussian pre-filtering



Gaussian 1/2

G 1/4

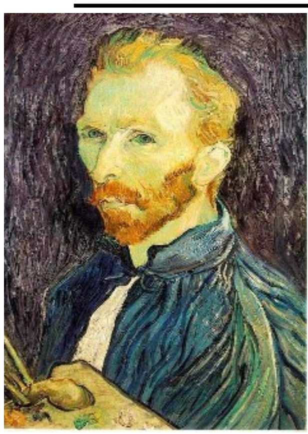


G 1/8

Solution: filter the image, *then* subsample

- Why does this work?

Compare with...



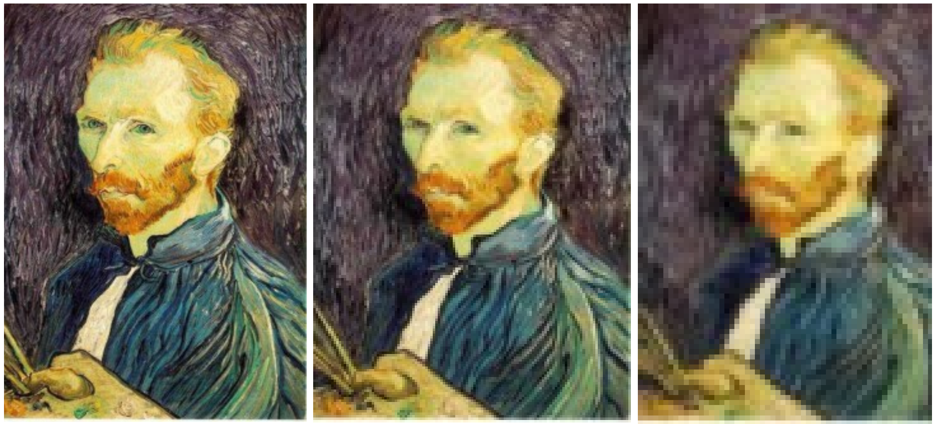
1/2

1/4 (2x zoom)

1/8 (4x zoom)

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.

Subsampling with Gaussian pre-filtering

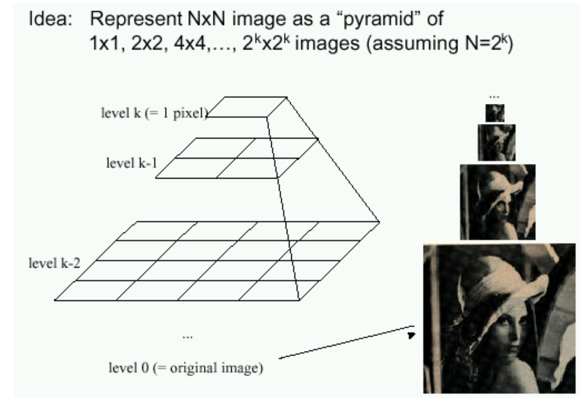


Gaussian 1/2 G 1/4 G 1/8

Solution: filter the image, *then* subsample

- Filter size should double for each ½ size reduction. Why?
- How can we speed this up?

Some times we want many resolutions



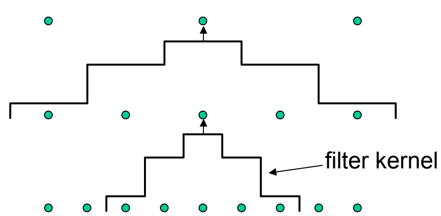
Known as a **Gaussian Pyramid** [Burt and Adelson, 1983]

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

Gaussian Pyramids have all sorts of applications in computer vision

- We'll talk about these later in the course

Gaussian pyramid construction



Repeat

- Filter
- Subsample

Until minimum resolution reached

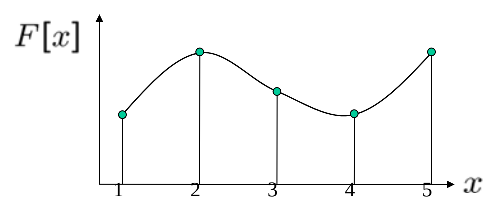
- can specify desired number of levels (e.g., 3-level pyramid)

The whole pyramid is only 4/3 the size of the original image!

Image resampling

So far, we considered only power-of-two subsampling

- What about arbitrary scale reduction?
- How can we increase the size of the image?



d = 1 in this example

Recall how a digital image is formed

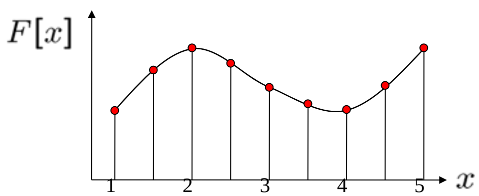
$$F[x,y] = \text{quantize}\{f(xd, yd)\}$$

- 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 resampling

So far, we considered only power-of-two subsampling

- What about arbitrary scale reduction?
- How can we increase the size of the image?



d = 1 in this example

Recall how a digital image is formed

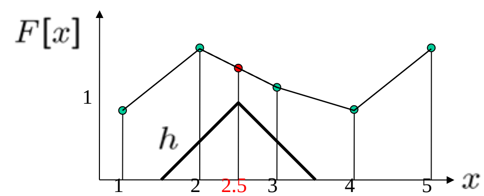
$$F[x, y] = \text{quantize}\{f(xd, yd)\}$$

- 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 resampling

So what to do if we don't know f

- Answer: guess an approximation $\tilde{f}$
- Can be done in a principled way: filtering



d = 1 in this example

Image reconstruction

- Convert F to a continuous function

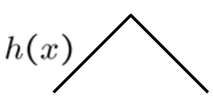
$$f_F(x) = F(\tfrac{x}{d}) \text{ when } \tfrac{x}{d} \text{ is an integer, 0 otherwise}$$

- Reconstruct by cross-correlation:

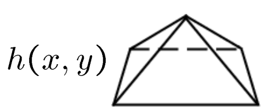
$$\tilde{f} = h \otimes f_F$$

Resampling filters

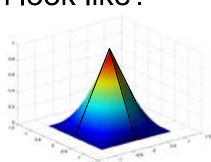
What does the 2D version of this hat function look like?



performs
linear interpolation



(tent function) performs
bilinear interpolation



Simpler implementation of bilinear interpolation

Better filters give better resampled images

- Bicubic is common choice
- fit 3rd degree polynomial surface to pixels in neighborhood
- can also be implemented by a convolution