ECE 4973/5973: Lecture 6
Resampling

Samuel Cheng

Slide credits: Noah Snavely

Image Scaling

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

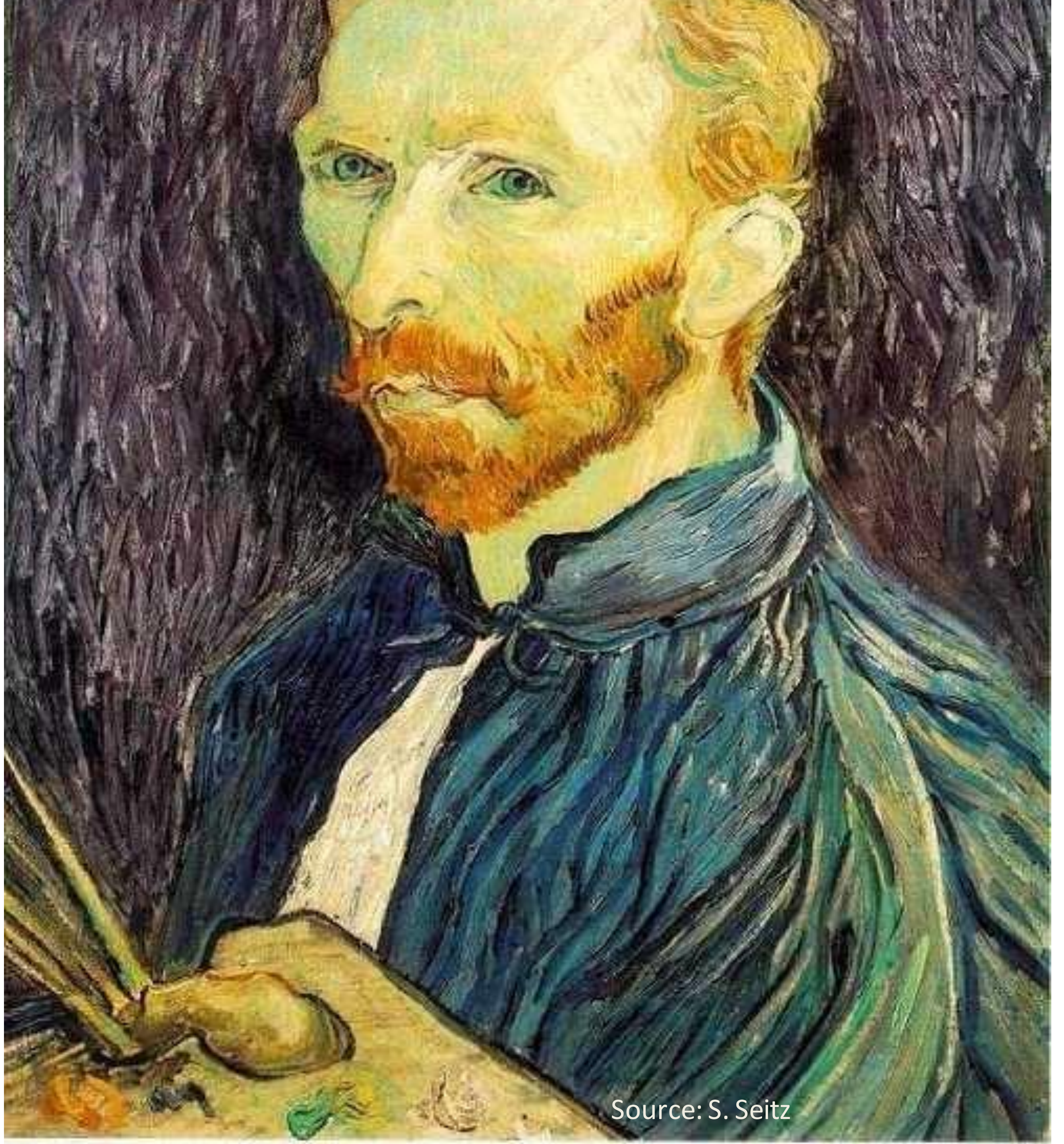
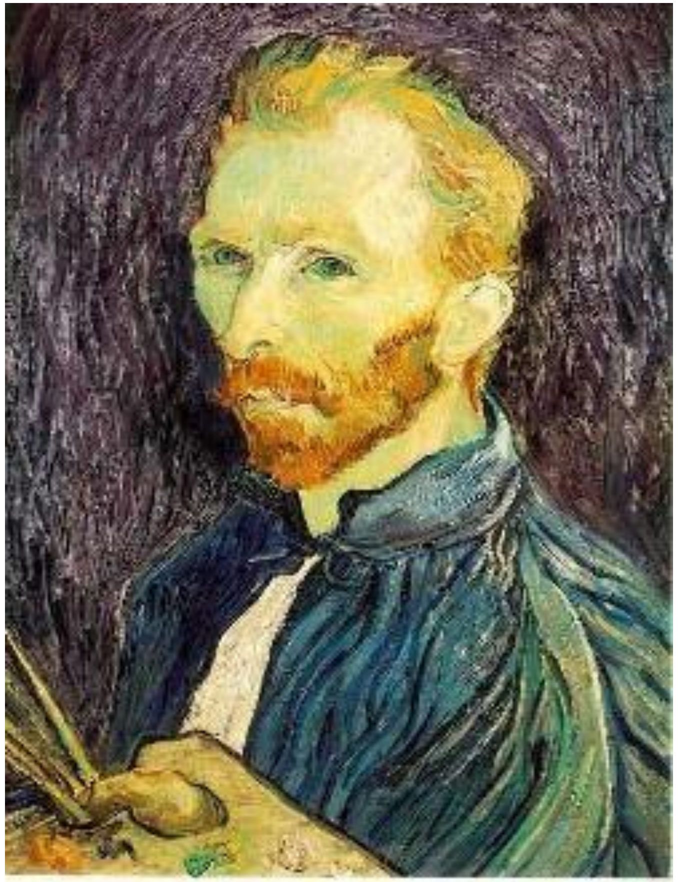


Image sub-sampling



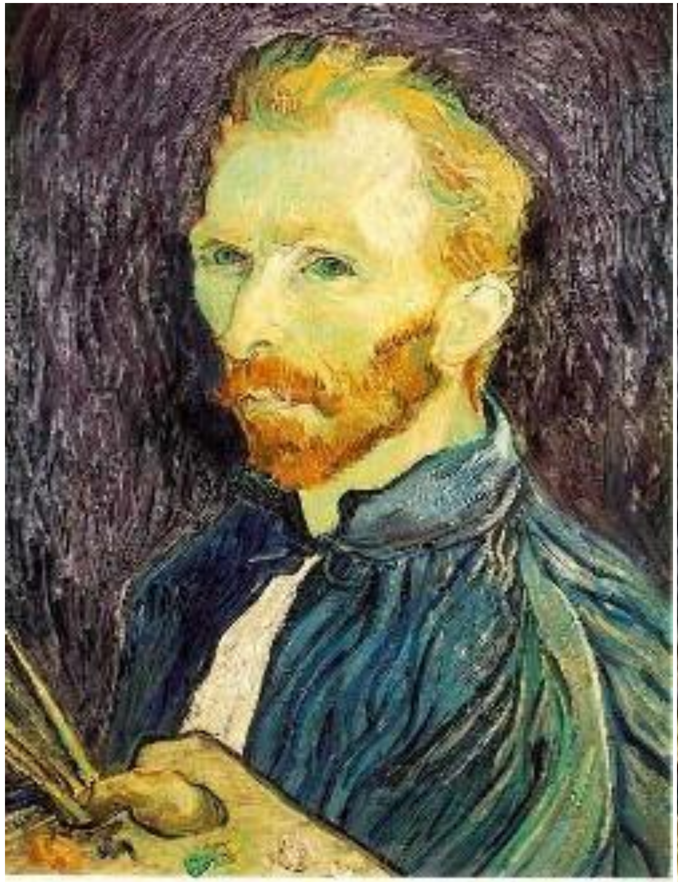
Throw away every other row and column to create a 1/2 size image
- called *image sub-sampling*

1/4

1/8

Source: S. Seitz

Image sub-sampling



1/2

1/4 (2x zoom)

1/8 (4x zoom)

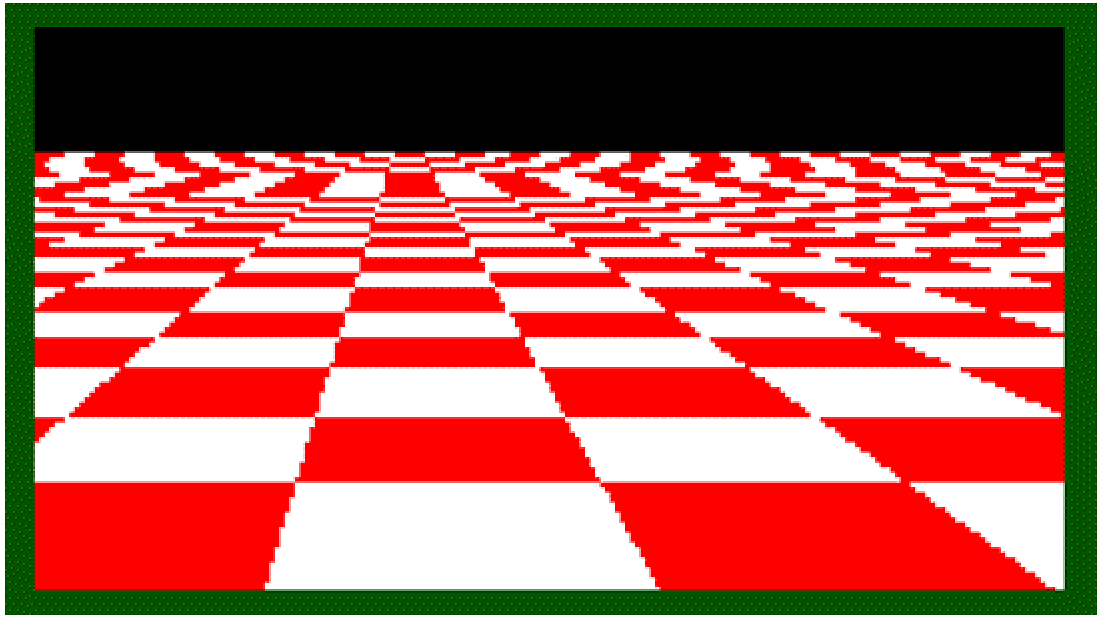
Why does this look so crufty?

Source: S. Seitz

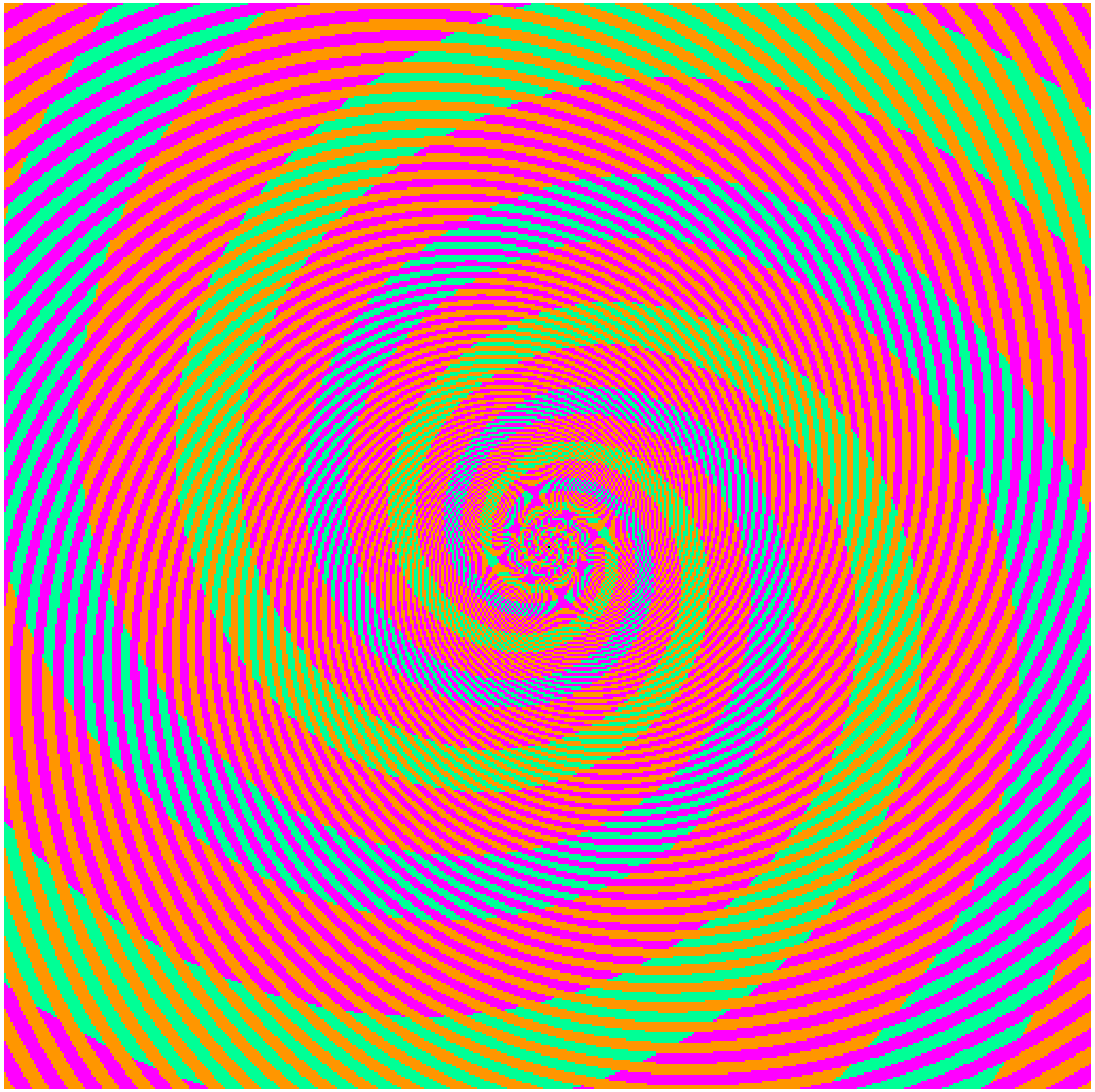
Image sub-sampling

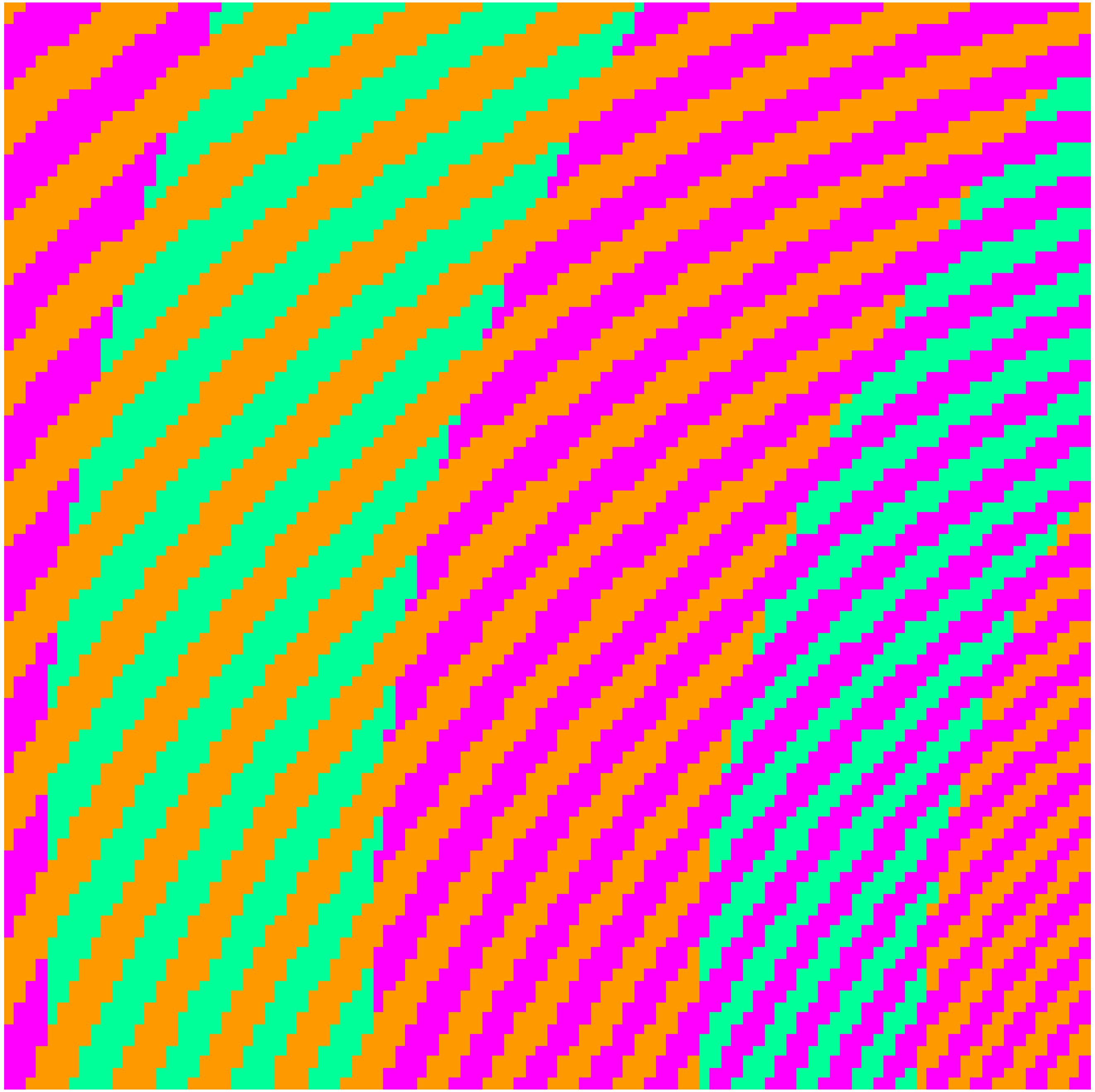
Source: F. Durand

Even worse for synthetic images



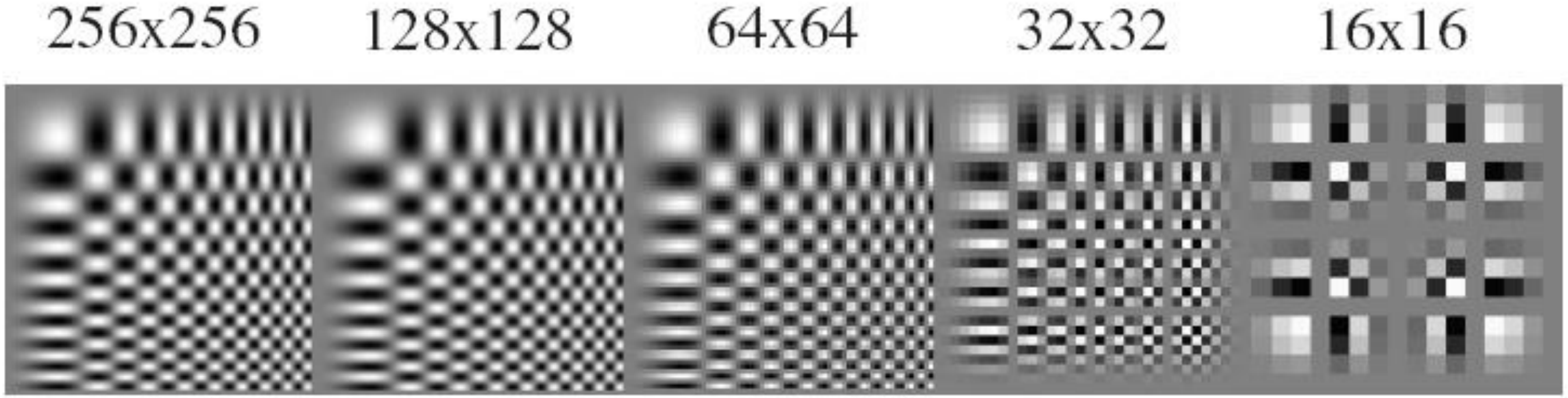
Source: L. Zhang





The blue and green colors are actually the same
http://blogs.discovermagazine.com/badastronomy/2009/06/24/the-blue-and-the-green/

Artifacts from sampling

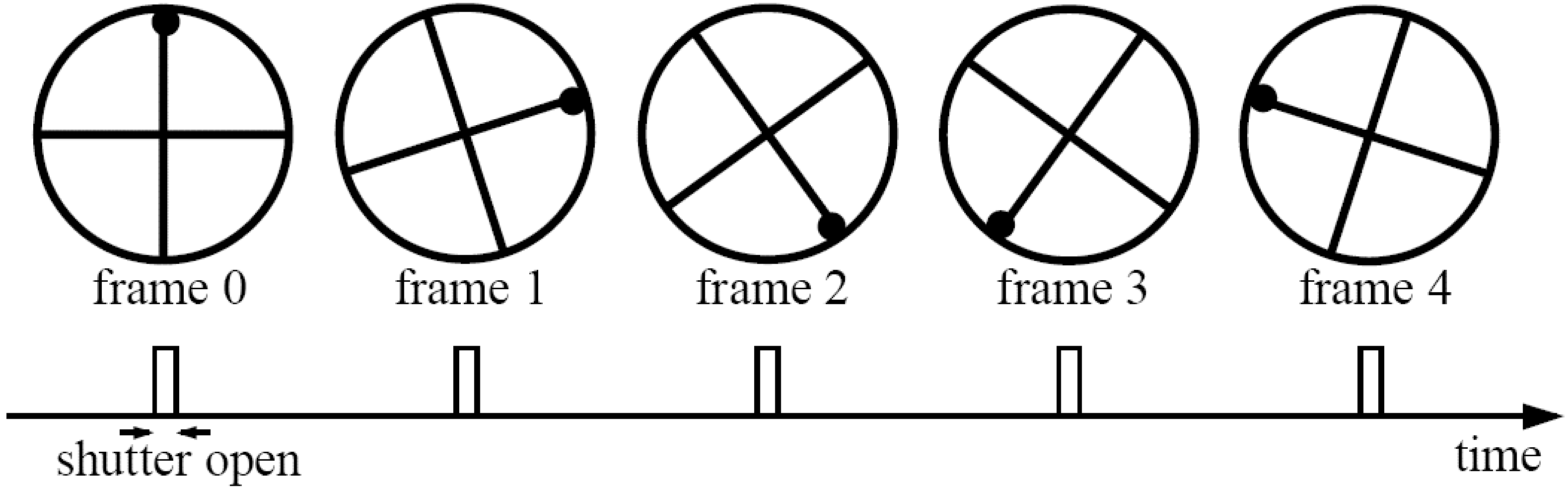


Interesting videos

Aliasing in video

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):

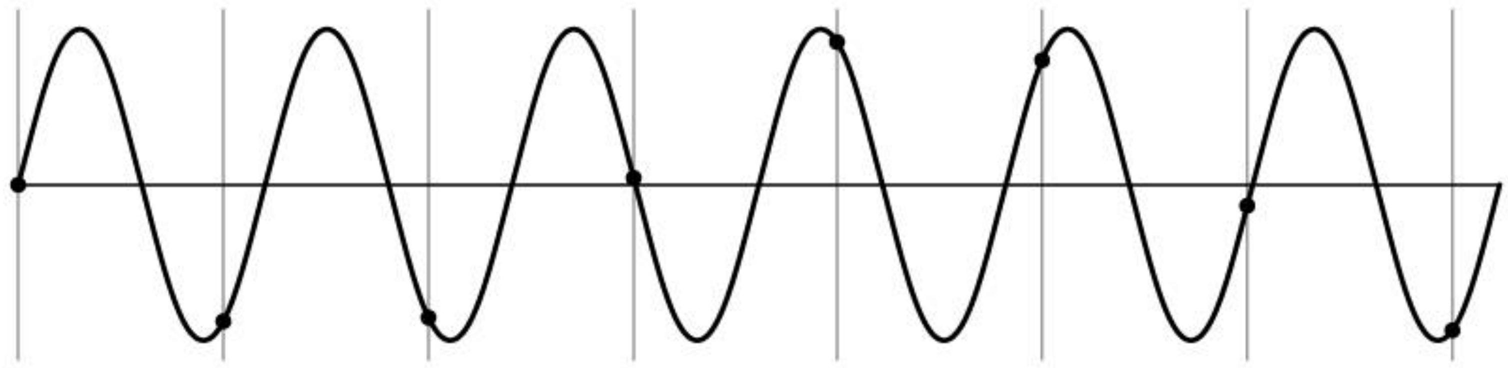


Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

Slide by Steve Seitz

Aliasing problem

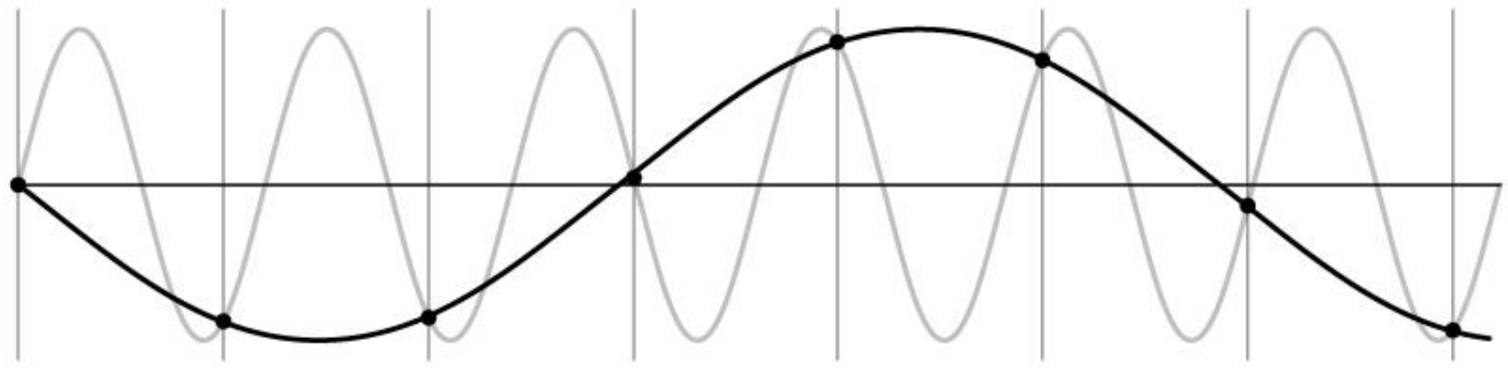
- 1D example (sinewave):



Source: S. Marschner

Aliasing problem

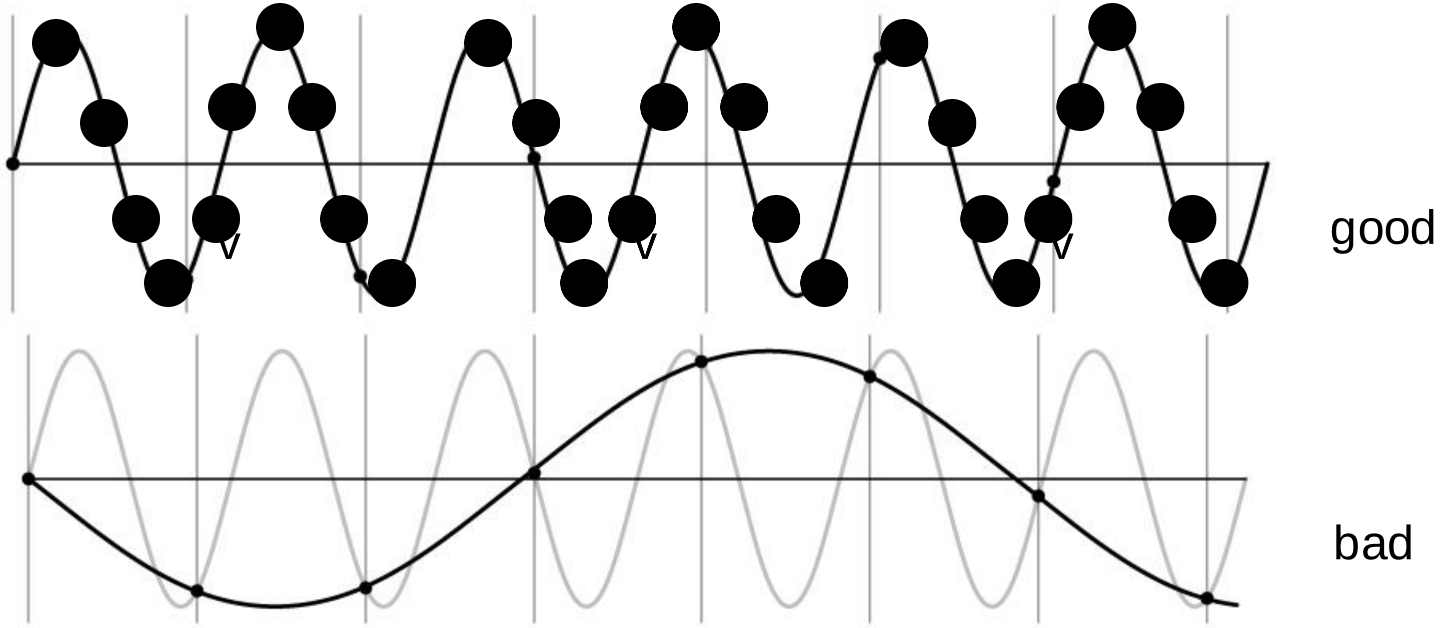
- 1D example (sinewave):



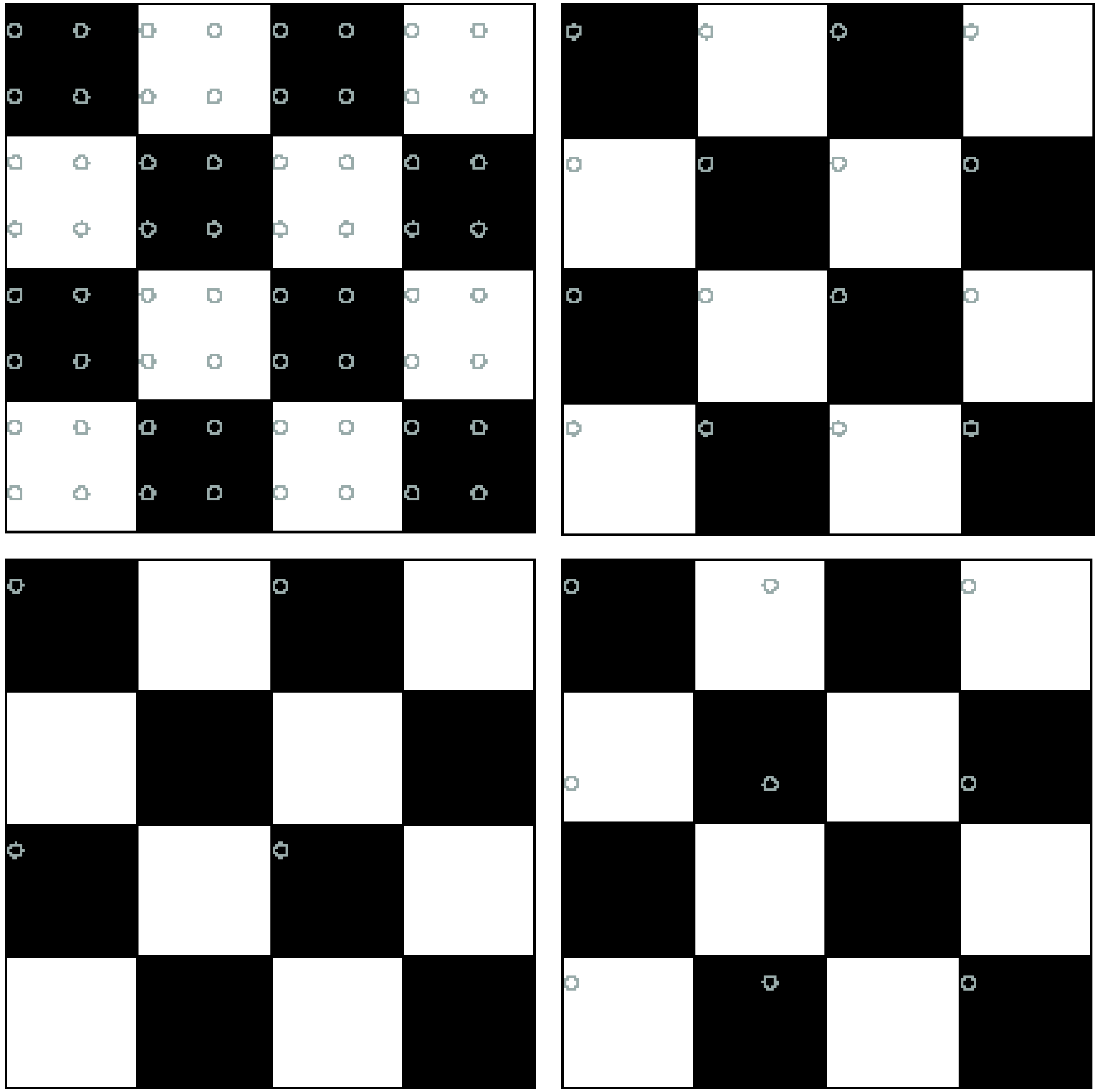
Source: S. Marschner

Nyquist-Shannon Sampling Theorem

- When sampling a signal at discrete intervals, the sampling frequency must be $\geq 2 \times f_{max}$
- f_{max} = max frequency



Nyquist limit – 2D example



Good sampling

How to sample?

Why aliasing precisely??

Bad sampling

Revisit FT

$$f(t) \xrightarrow{\mathcal{F}} F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt$$

$$F(\omega) \xrightarrow{\mathcal{F}^{-1}} f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega$$

$$f(t) * g(t) \xrightarrow{\mathcal{F}} F(\omega)G(\omega)$$

$$f(t)g(t) \xrightarrow{\mathcal{F}} F(\omega) * G(\omega)$$

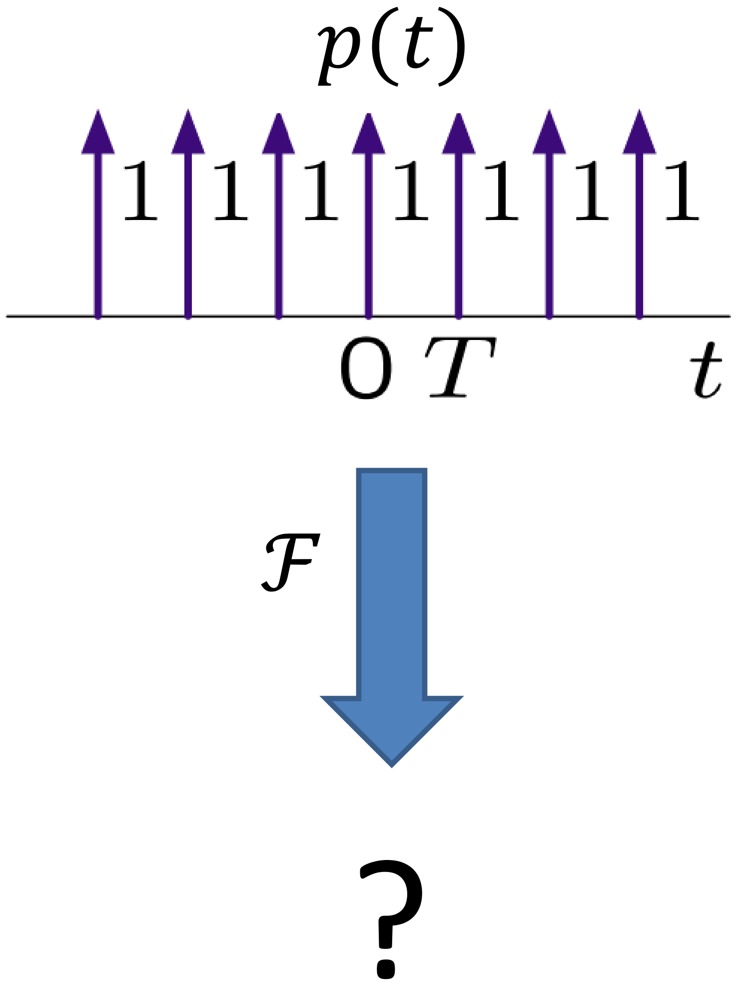
$$\mathcal{F}[f(t) * g(t)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau\, e^{-i\omega t}\, dt = \int_{-\infty}^{\infty} f(\tau)\int_{-\infty}^{\infty} g(t-\tau)e^{-i\omega(t-\tau)}dt\, e^{-i\omega\tau}d\tau$$

$$= \int_{-\infty}^{\infty} f(\tau)G(\omega)e^{-i\omega\tau}d\tau = G(\omega)\int_{-\infty}^{\infty} f(\tau)e^{-i\omega\tau}d\tau = F(\omega)G(\omega)$$

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega \xrightarrow[\omega\leftarrow t']{t\leftarrow -\omega'} f(-\omega') = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(t')e^{-i\omega' t'}dt' = \frac{1}{2\pi}\mathcal{F}[F(t')]\,(\omega')$$

$$f(t) \xrightarrow{\mathcal{F}} F(\omega) \Leftrightarrow F(t) \xrightarrow{\mathcal{F}} 2\pi f(-\omega)$$

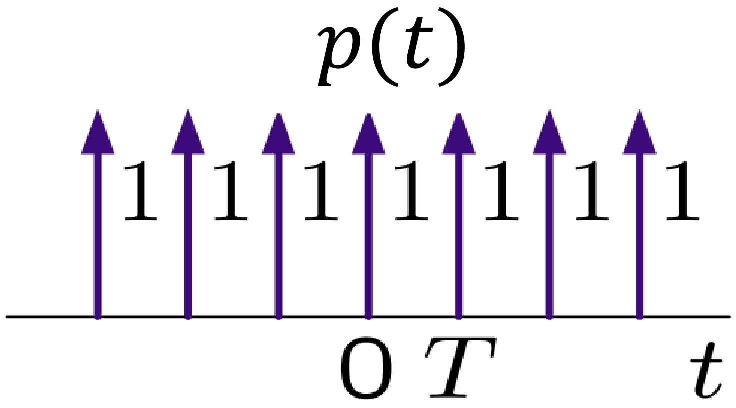
Pulse train



- A function $f(t)$ sampled at $t = nT$ is simply $f(t)p(t)$
- $\mathcal{F}[f(t)p(t)] = F(\omega) * P(\omega)$
- What is $P(\omega)$?

$$f(t)g(t) \xrightarrow{\mathcal{F}} F(\omega) * G(\omega)$$

Fourier Transform of Pulse train



$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

$$P(\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} \sum_{n=-\infty}^{\infty} \delta(t - nT)\, dt$$

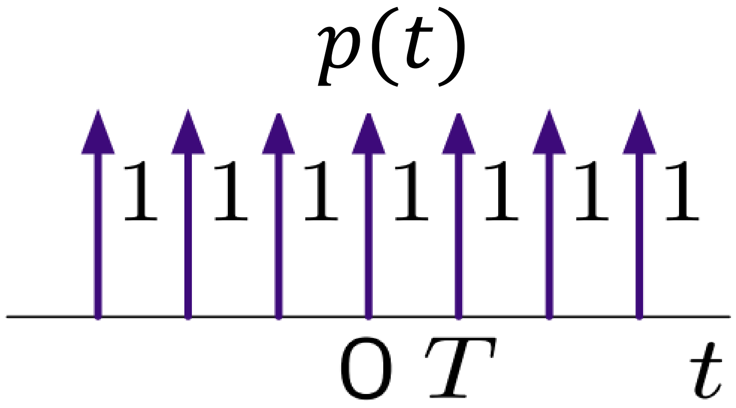
$$= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i\omega t}\, \delta(t - nT)\, dt = \sum_{n=-\infty}^{\infty} e^{-in\omega T}$$

$$\omega \leftarrow t$$
$$T \leftarrow -\Omega$$

$$\mathcal{F}\left[\sum_{n=-\infty}^{\infty} e^{int\Omega}\right] = 2\pi\left[\sum_{n=-\infty}^{\infty} \delta(\omega + n\Omega)\right] = 2\pi\left[\sum_{n=-\infty}^{\infty} \delta(\omega - n\Omega)\right]$$

$$f(t) \overset{\mathcal{F}}{\rightarrow} F(\omega) \Leftrightarrow F(t) \overset{\mathcal{F}}{\rightarrow} 2\pi f(-\omega)$$

Fourier Series of Pulse train



$$p(t) = \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{i2\pi n \frac{t}{T}}$$

$$\left\langle \sum_{n=-\infty}^{\infty} \delta(t - nT), e^{i2\pi m \frac{t}{T}} \right\rangle = \left\langle \sum_{n=-\infty}^{\infty} c[n] e^{i2\pi n \frac{t}{T}}, e^{i2\pi m \frac{t}{T}} \right\rangle$$

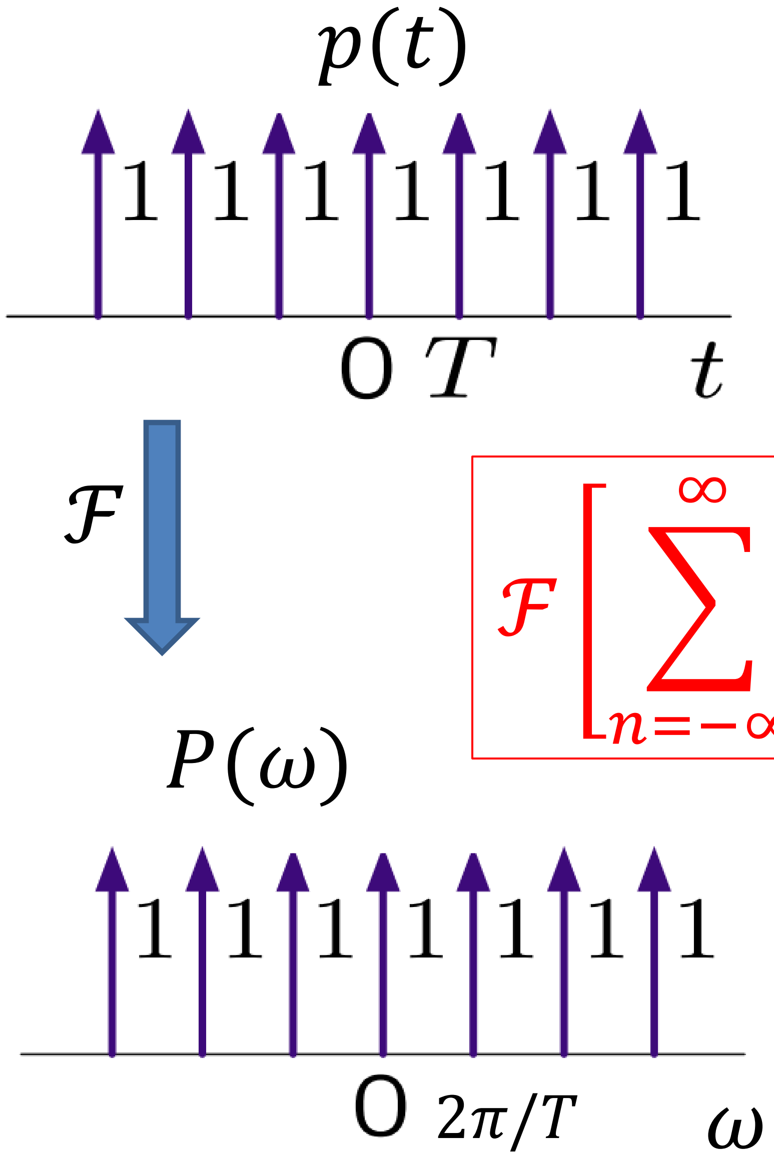
$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

$$\frac{1}{T} = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \sum_{n=-\infty}^{\infty} \delta(t - nT) e^{-i2\pi m \frac{t}{T}} dt = c[m]$$

$$\langle a(t), b(t) \rangle \triangleq \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} a(t) \overline{b(t)} dt$$

$$\left\langle e^{i2\pi n \frac{t}{T}}, e^{i2\pi m \frac{t}{T}} \right\rangle = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} e^{i2\pi (n-m) \frac{t}{T}} dt = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{otherwise} \end{cases}$$

Fourier Transform of Pulse train



$$p(t) = \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{i2\pi n \frac{t}{T}}$$

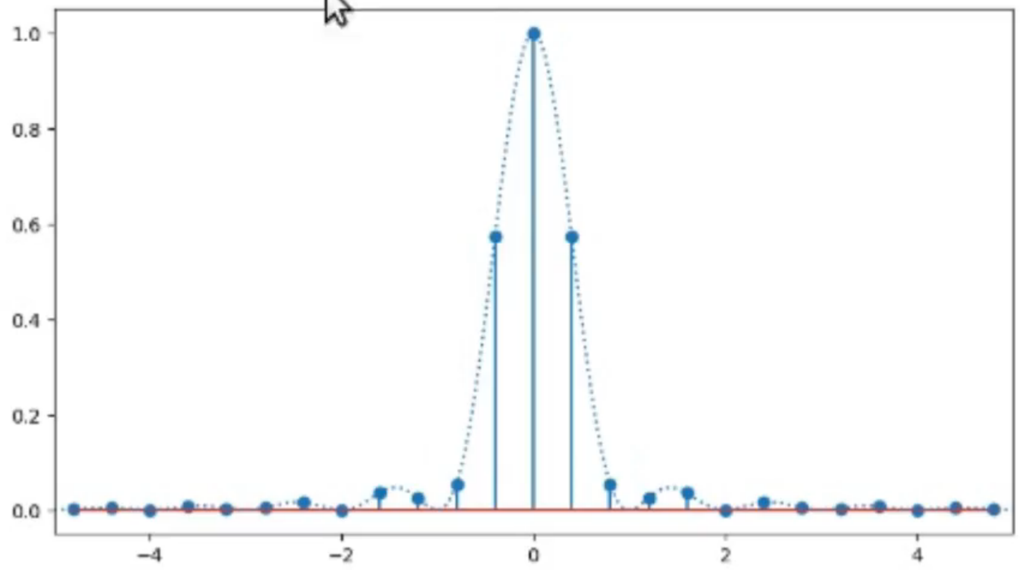
$$\mathcal{F}\left[\sum_{n=-\infty}^{\infty} \delta(t - nT)\right] = \frac{2\pi}{T}\left[\sum_{n=-\infty}^{\infty} \delta\left(f - n\left(\frac{2\pi}{T}\right)\right)\right]$$

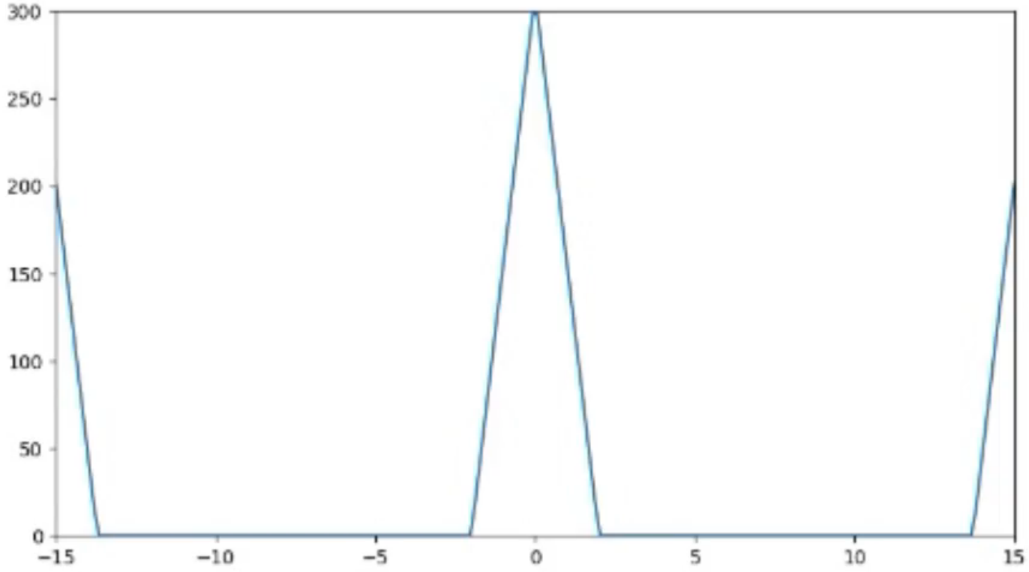
$$\mathcal{F}\left[\sum_{n=-\infty}^{\infty} e^{in\Omega t}\right] = 2\pi\left[\sum_{n=-\infty}^{\infty} \delta(\omega - n\Omega)\right]$$

Revisit Nyquist-Shannon Theorem

```
interact(plot_sinc,a=widgets.FloatSlider(min=1, max=30, step=0.05, value=15))
```

a 15.70



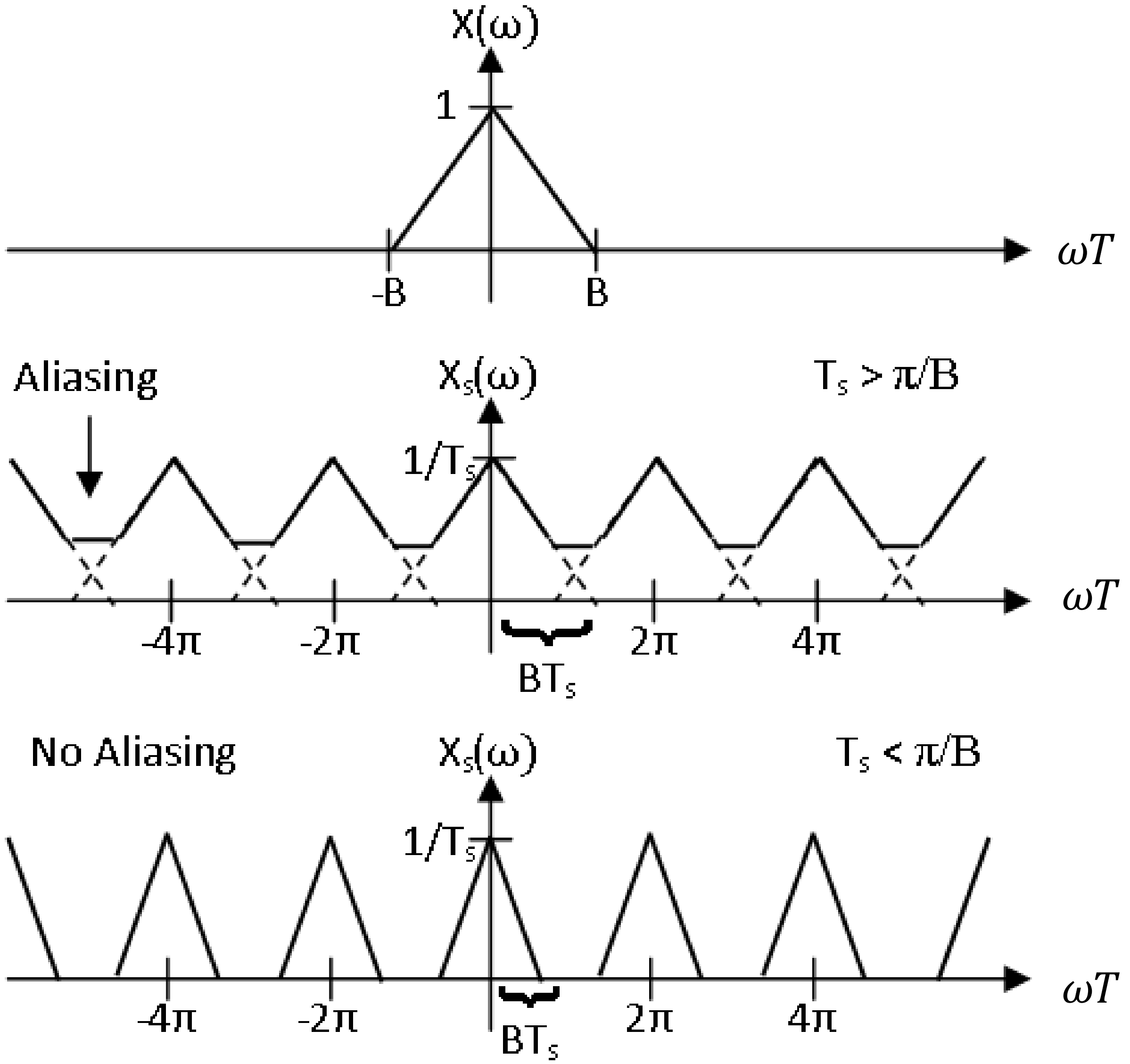


$$\mathcal{F}\left[\sum_{n=-\infty}^{\infty} \delta(t-nT)\right] = \frac{2\pi}{T}\left[\sum_{n=-\infty}^{\infty} \delta\left(\omega - n\left(\frac{2\pi}{T}\right)\right)\right]$$

$$f(t)g(t) \xrightarrow{\mathcal{F}} F(\omega) * G(\omega)$$

No aliasing as long as bandwidth $< \frac{2\pi}{T}$

Aliasing in downsampling

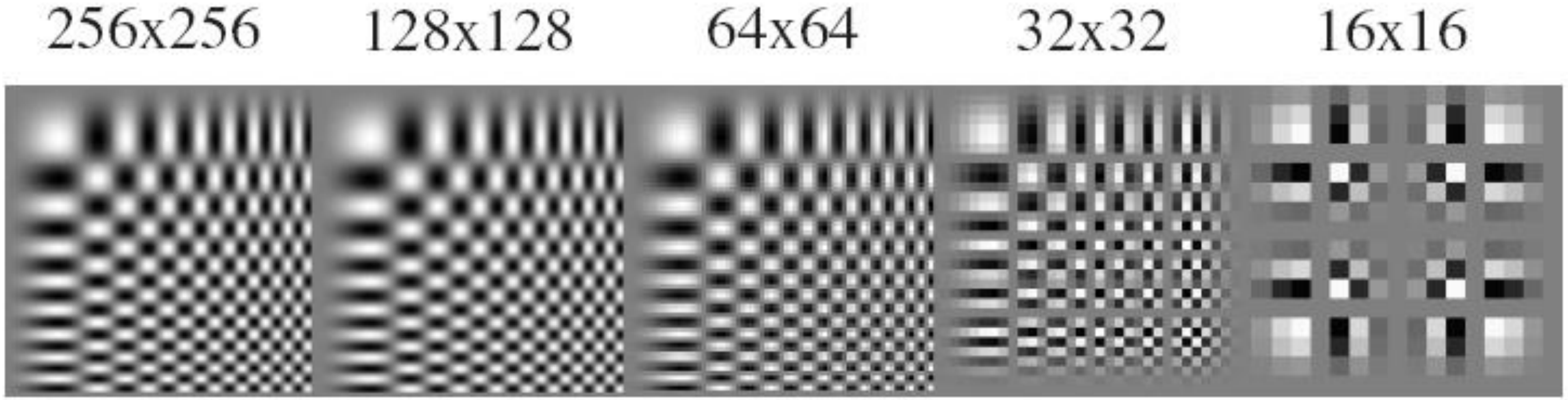


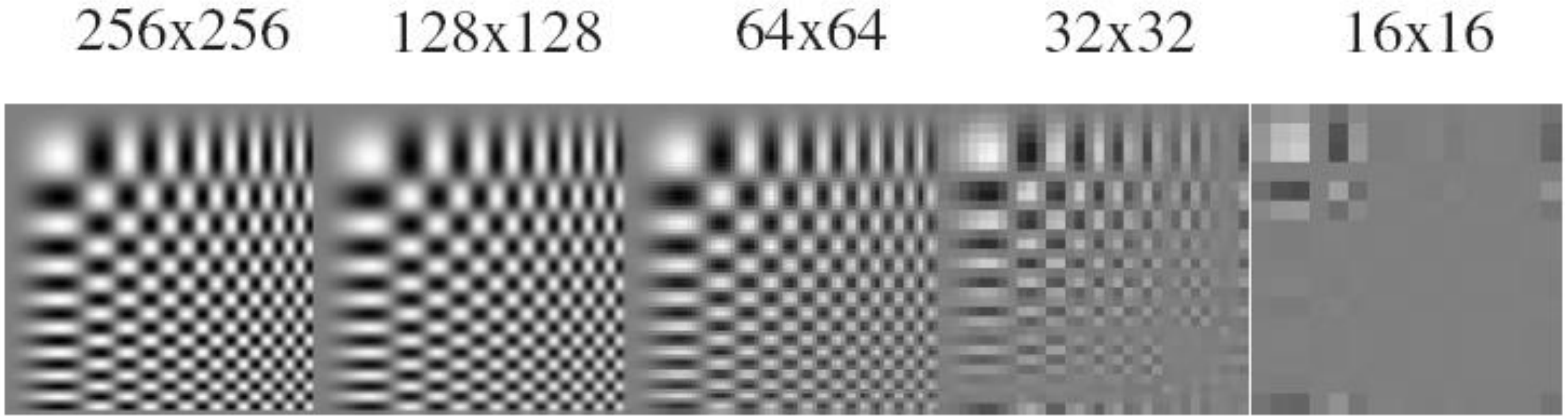
- Downsampling is just resampling at lower rate
- Aliasing if baseband overlaps

$$\mathcal{F}\left[\sum_{n=-\infty}^{\infty}\delta(t-nT)\right]=\frac{2\pi}{T}\left[\sum_{n=-\infty}^{\infty}\delta\left(\omega-n\left(\frac{2\pi}{T}\right)\right)\right]$$

$$f(t)g(t)\xrightarrow{\mathcal{F}}F(\omega)*G(\omega)$$

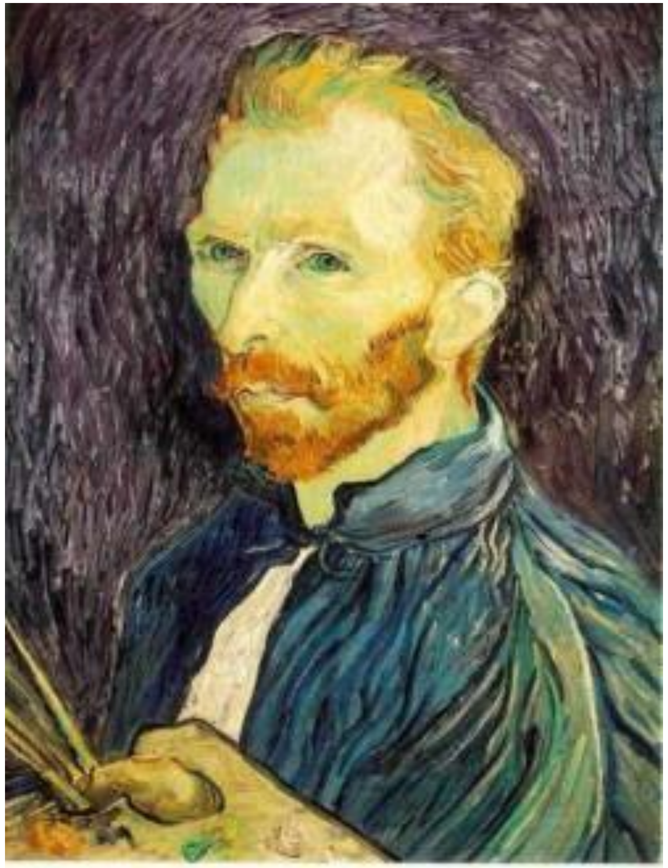
Anti-aliasing





Forsyth and Ponce 2002

Gaussian pre-filtering



Gaussian 1/2

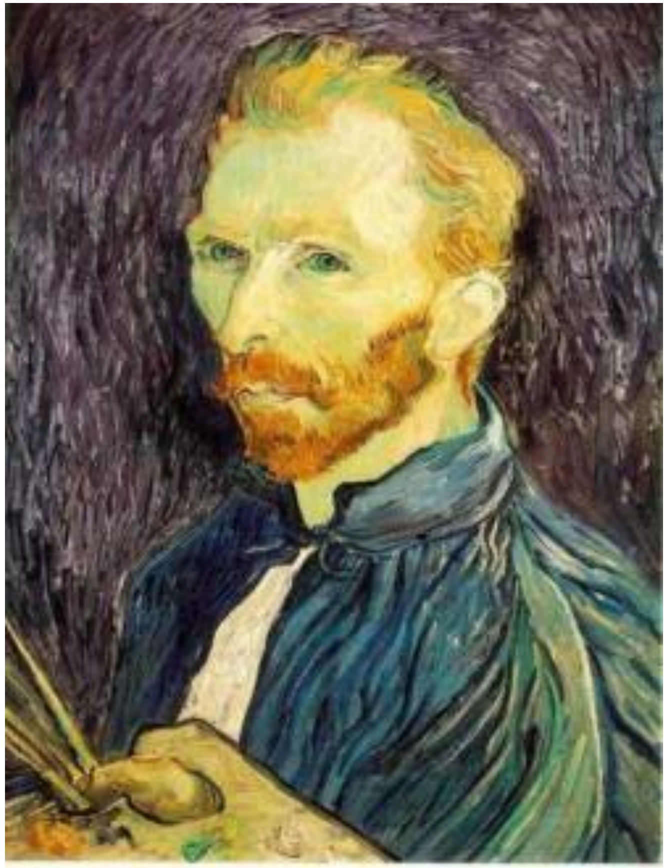
G 1/4

G 1/8

- Solution: filter the image, *then* subsample

Source: S. Seitz

Subsampling with Gaussian pre-filtering



Gaussian 1/2

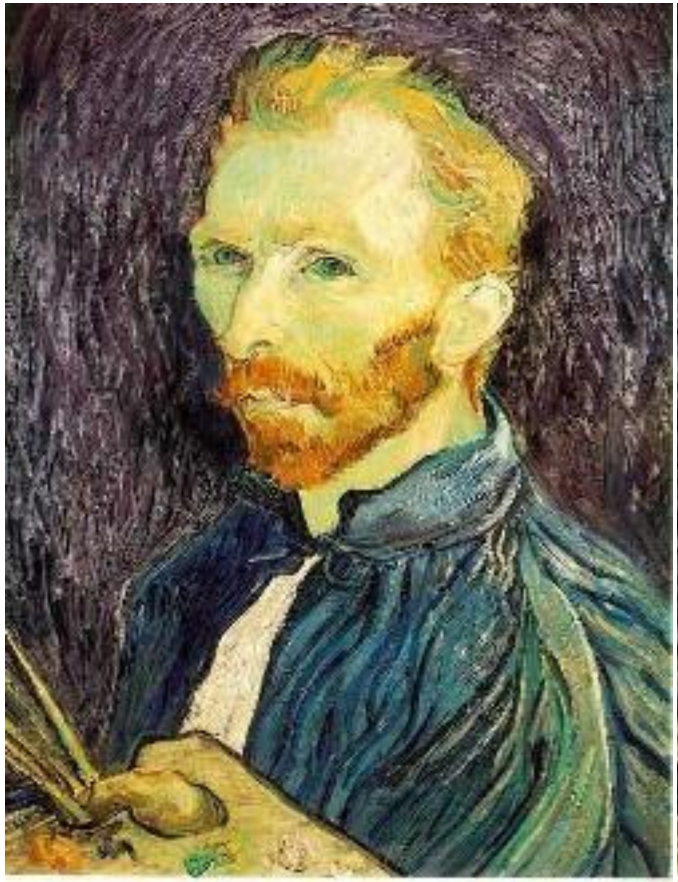
G 1/4

G 1/8

- Solution: filter the image, *then* subsample

Source: S. Seitz

Compare with...



1/2

1/4 (2x zoom)

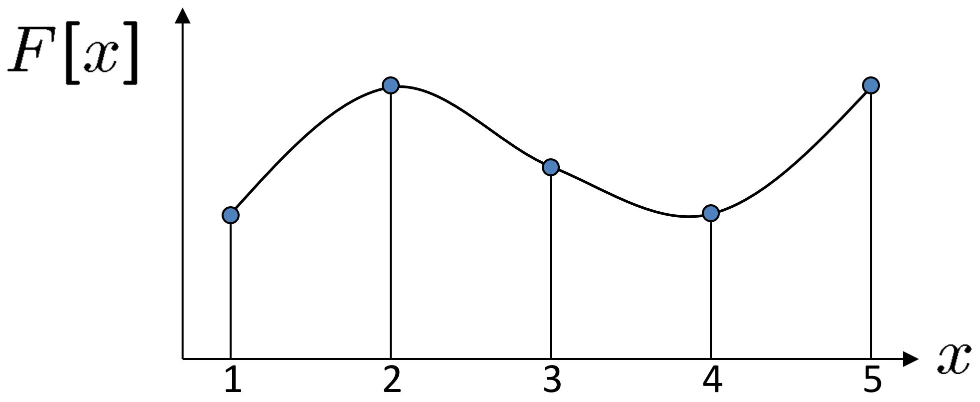
1/8 (4x zoom)

Source: S. Seitz

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

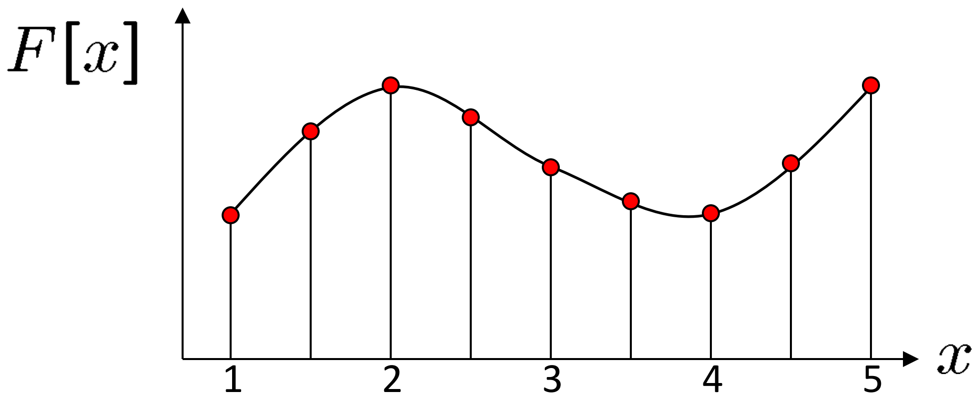
Recall how a digital image is formed

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

- 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

Adapted from: S. Seitz

Image interpolation



d = 1 in this example

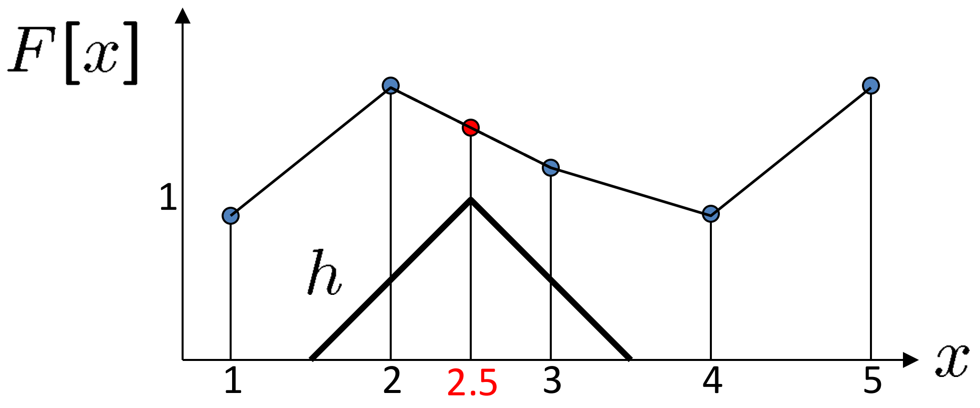
Recall how a digital image is formed

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

- 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

Adapted from: S. Seitz

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

Frequency representation

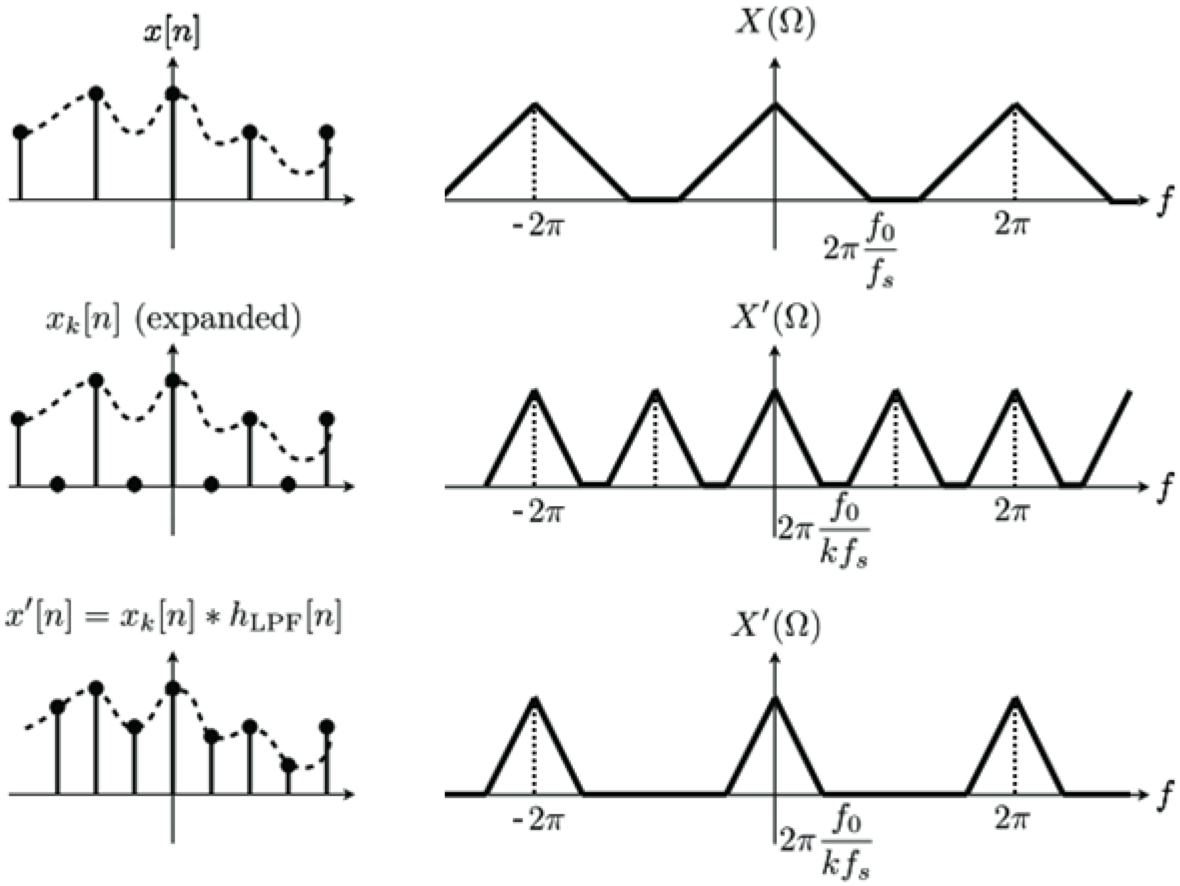
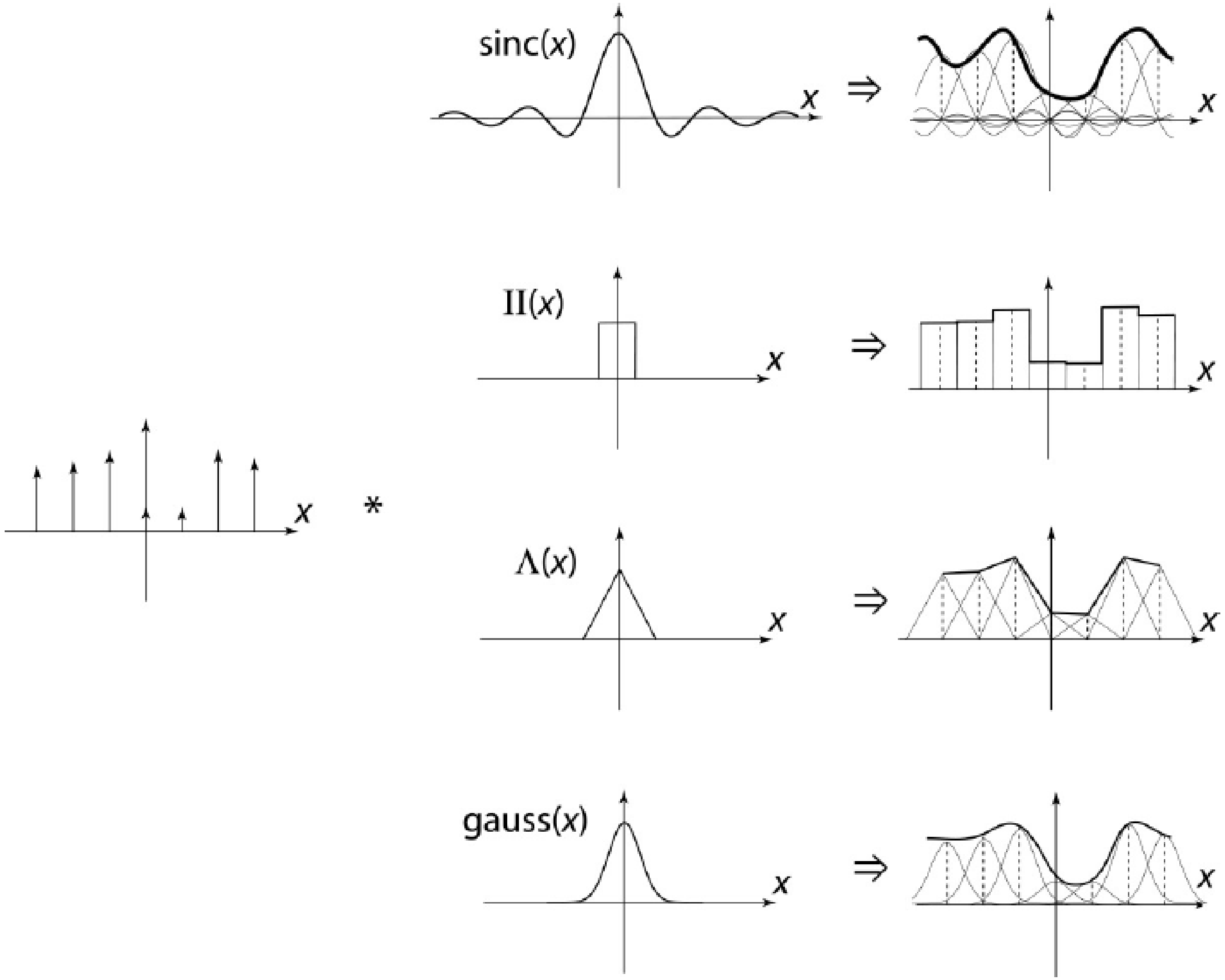


Image interpolation



"Ideal" reconstruction

Nearest-neighbor interpolation

Linear interpolation

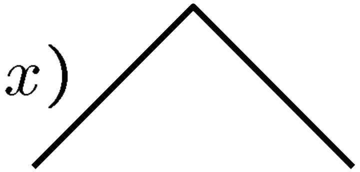
Gaussian reconstruction

Source: B. Curless

Reconstruction filters

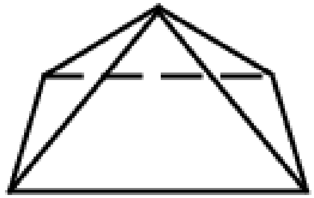
- What does the 2D version of this hat function look like?

$h(x)$



performs
linear interpolation

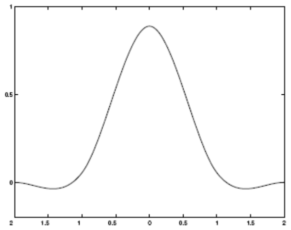
$h(x, y)$



(tent function) performs
bilinear interpolation

Better filters give better resampled images

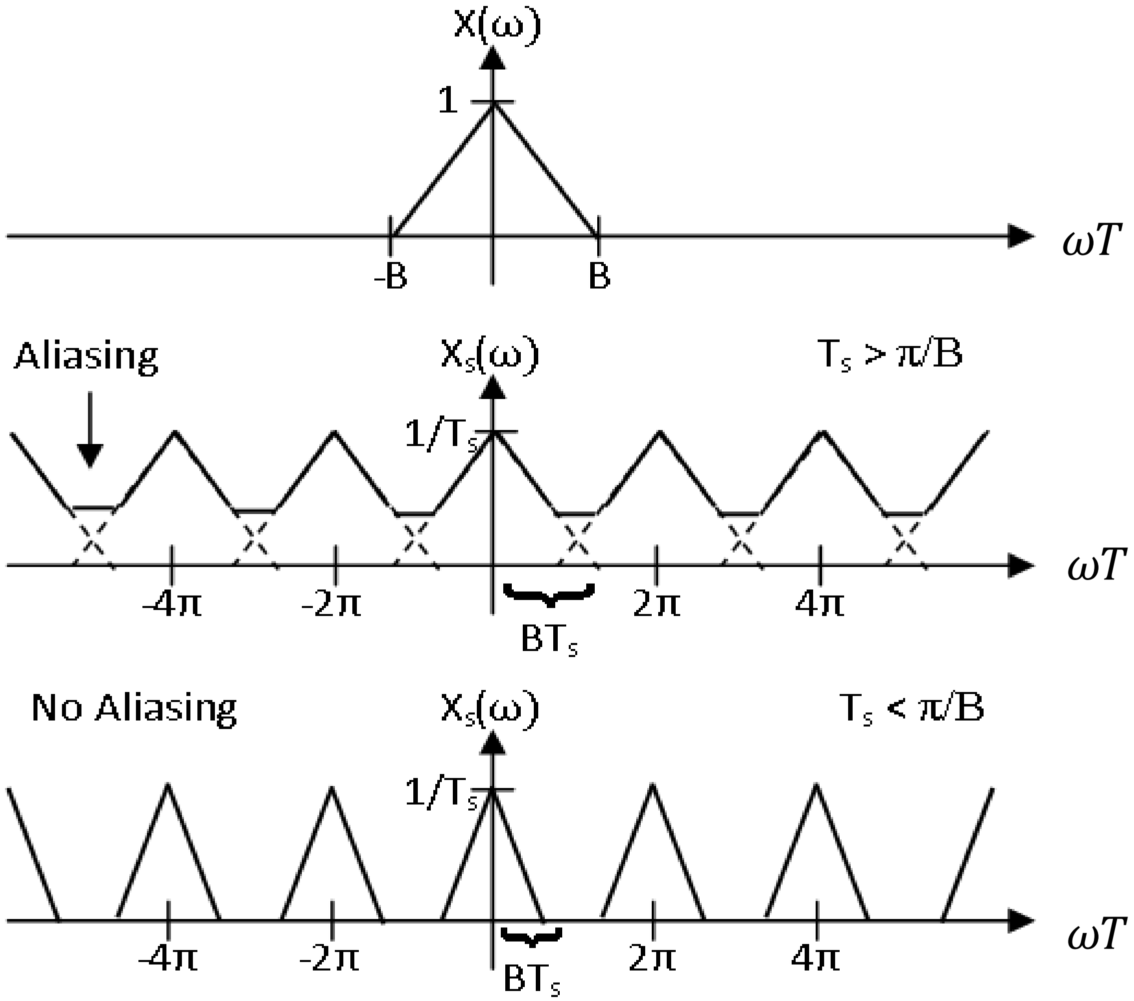
- **Bicubic** is common choice



Cubic reconstruction filter

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

Summary: downsampling and upsampling



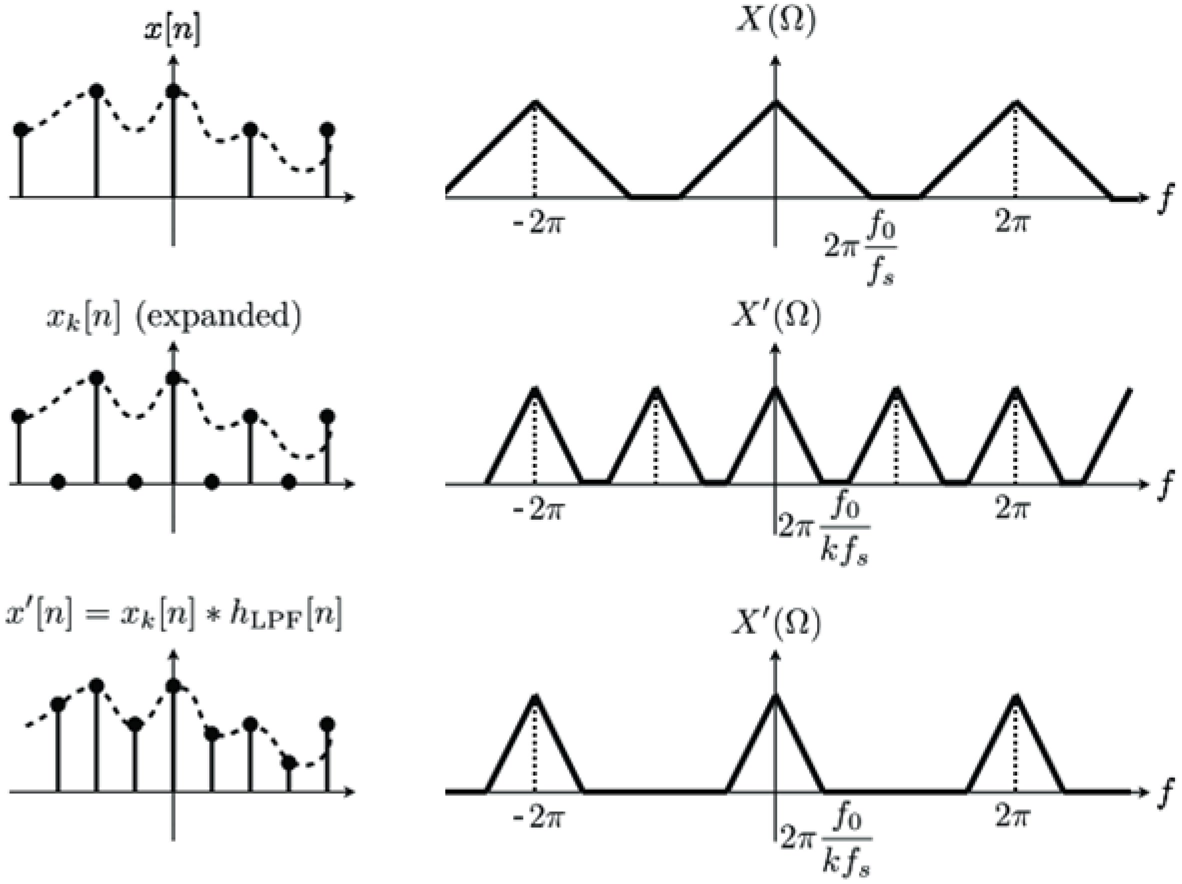


Image interpolation

Original image:  x 10

Nearest-neighbor interpolation

Bilinear interpolation

Bicubic interpolation

DSP Interpretation

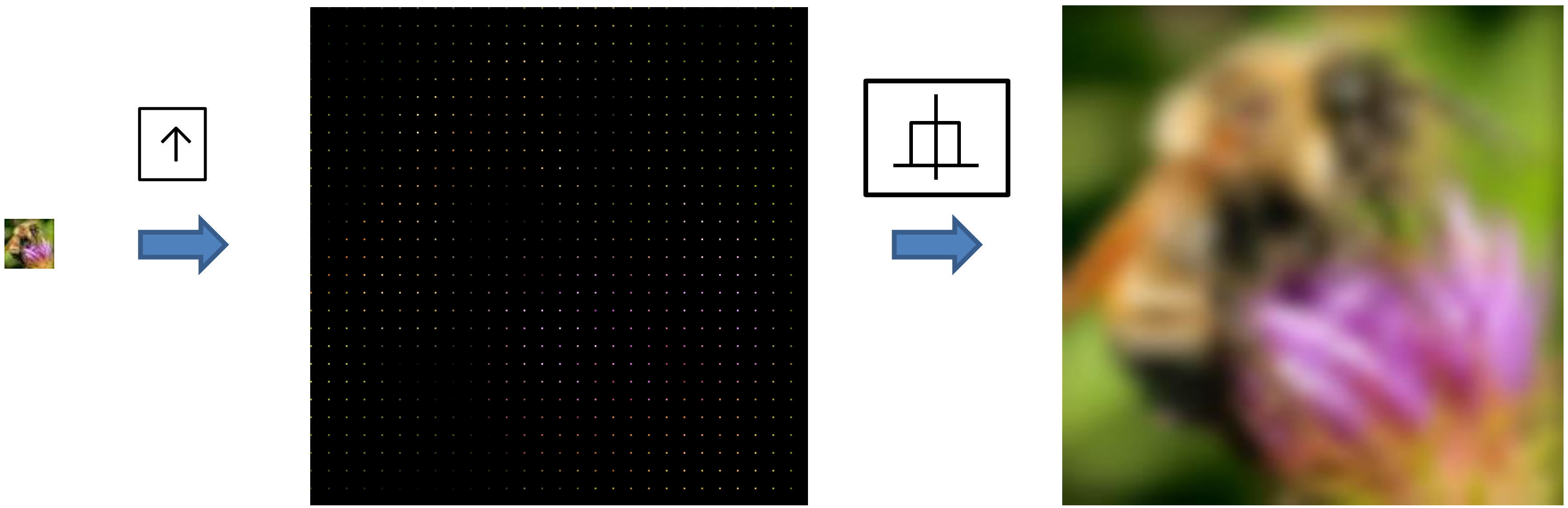
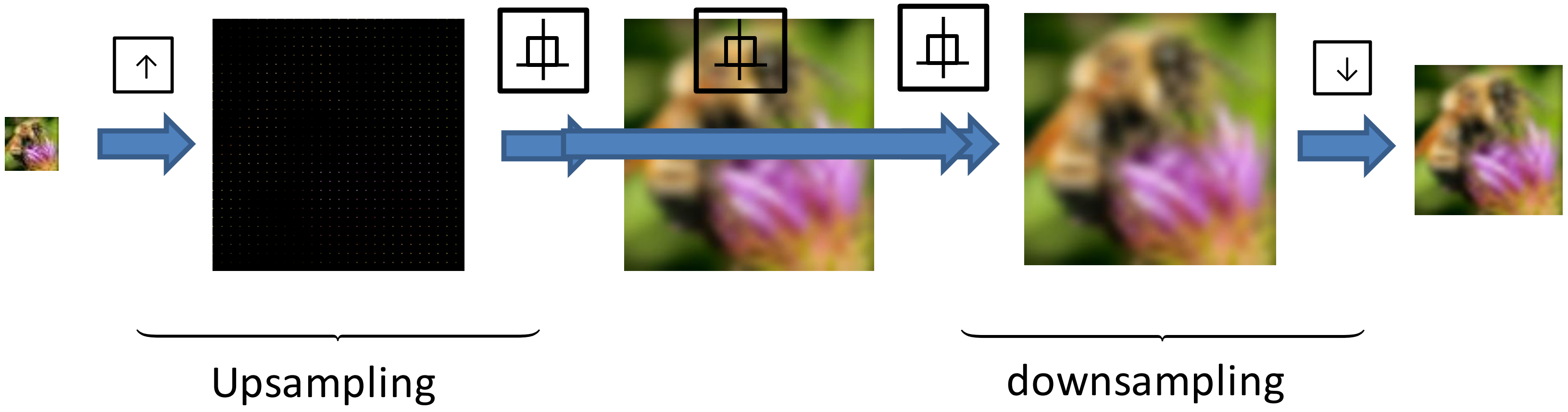


Image resampling



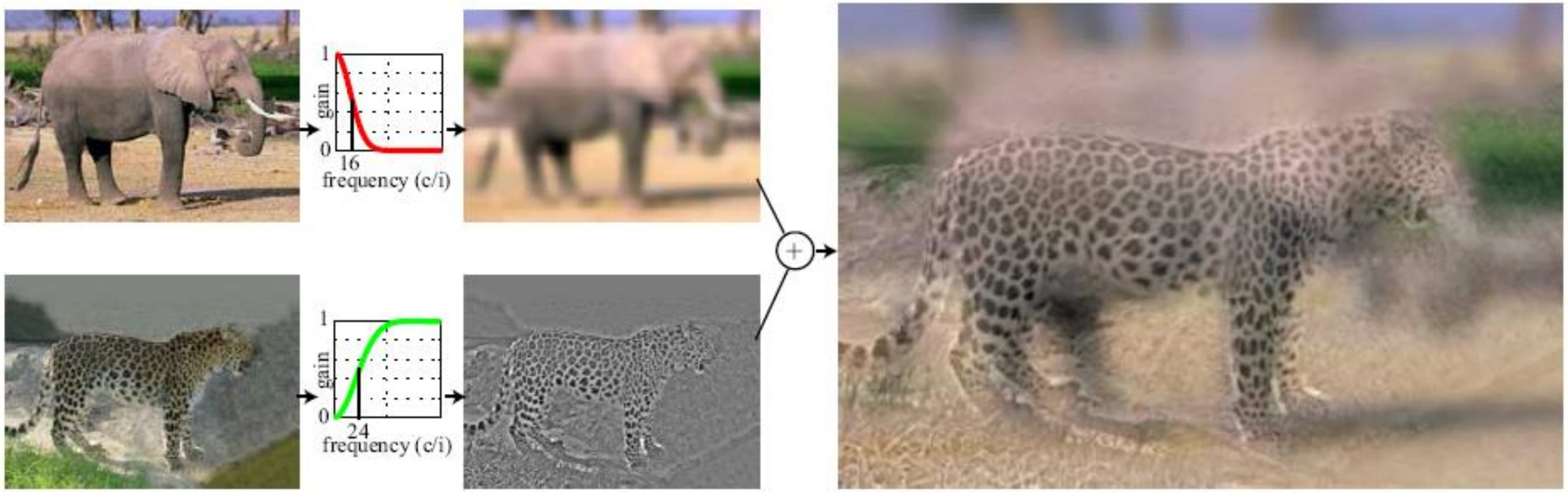
Hybrid Image

Salvador Dali, 1976

Another example

- Who is (s)he?

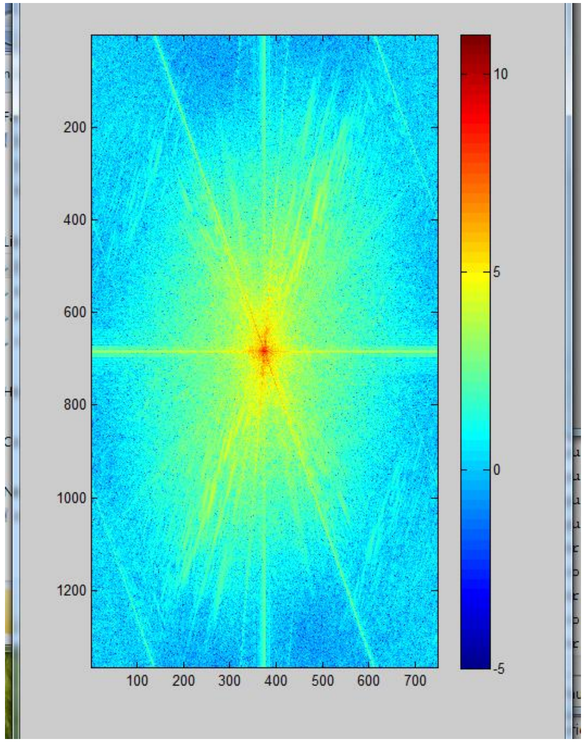
Hybrid Images



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

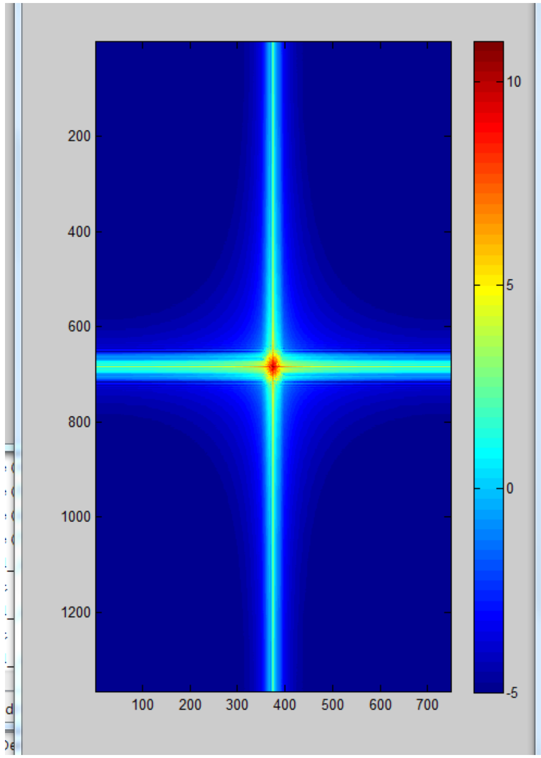
Hybrid Image in FFT

Hybrid Image



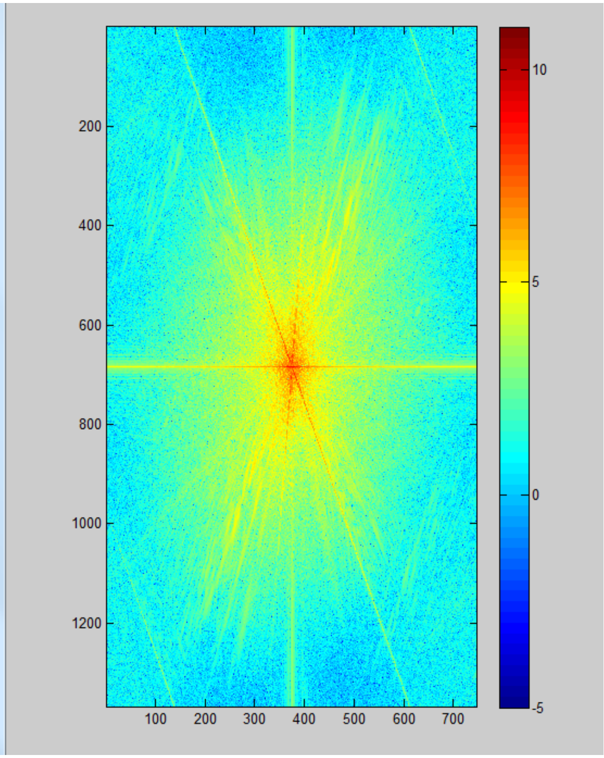
=

Low-passed Image

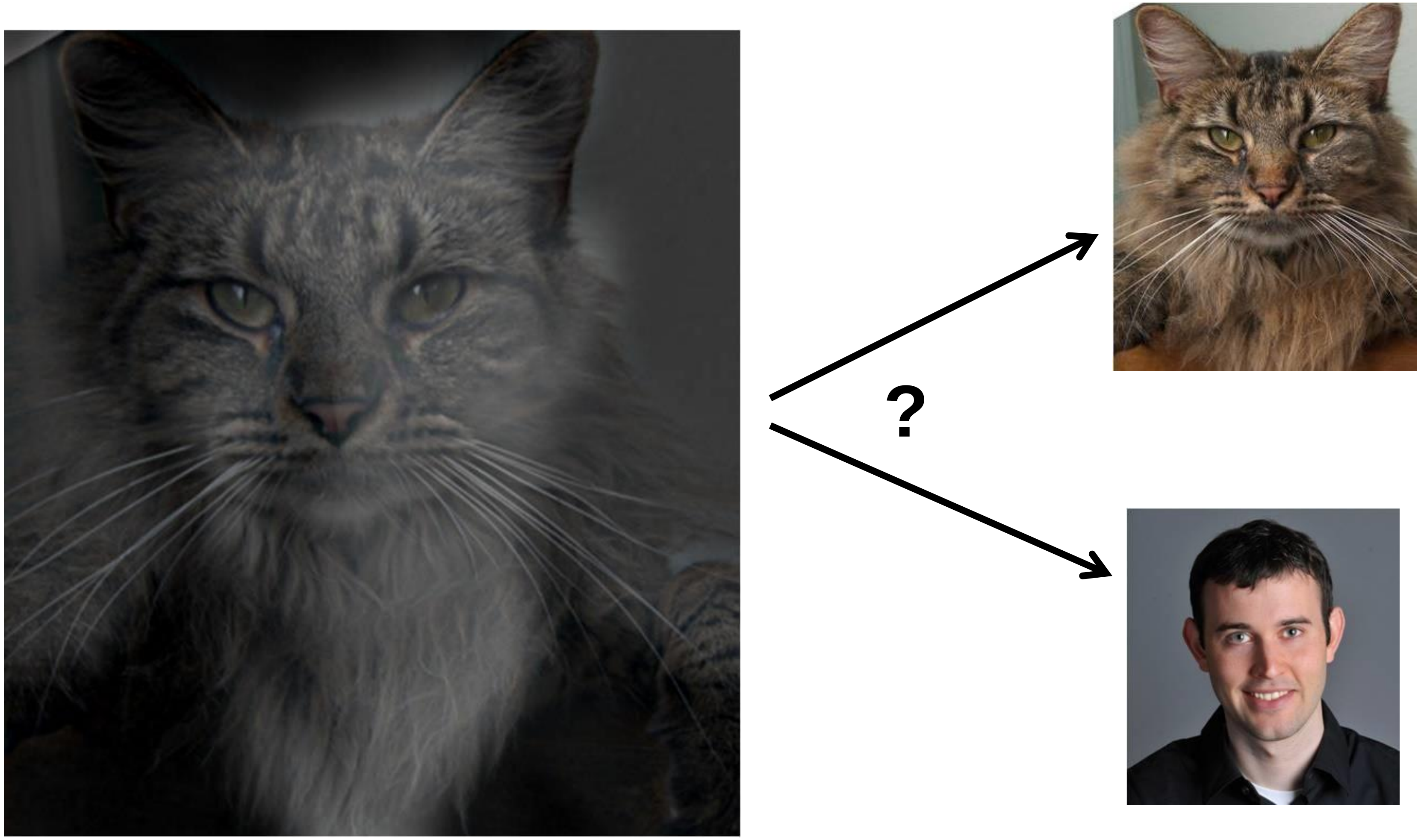


+

High-passed Image



Why do we get different, distance-dependent interpretations of hybrid images?



Hays

Campbell-Robson contrast sensitivity curve

Perceptual cues in the mid-high frequencies dominate perception.

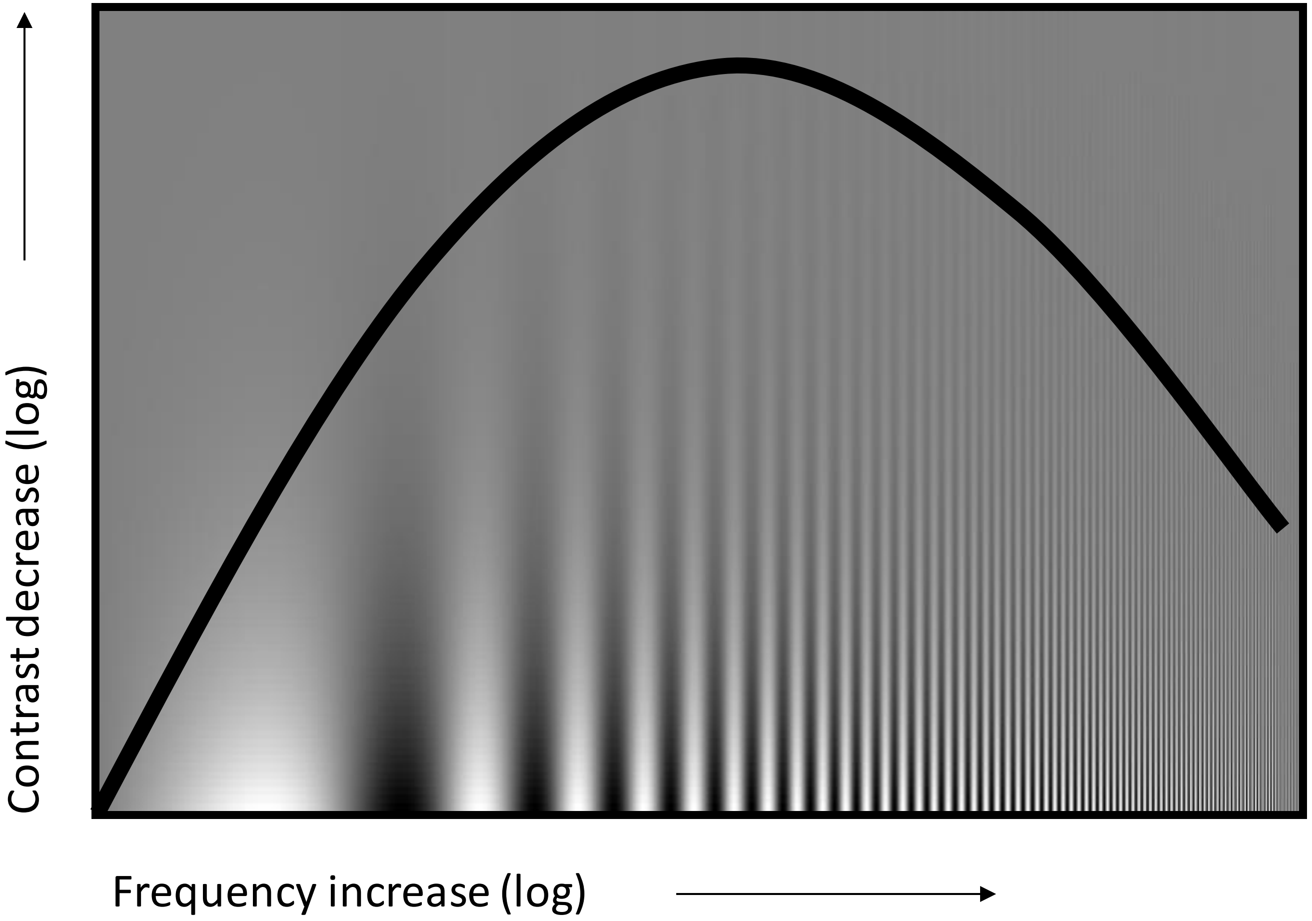
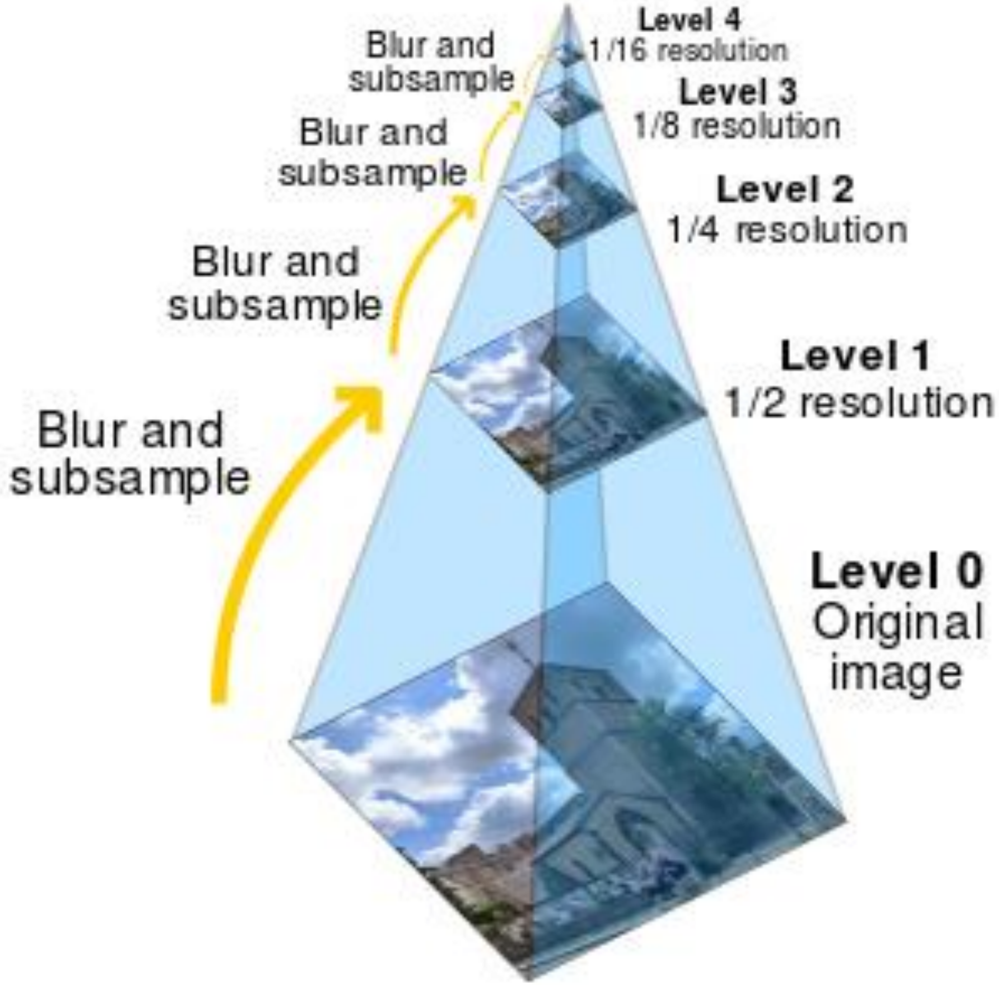


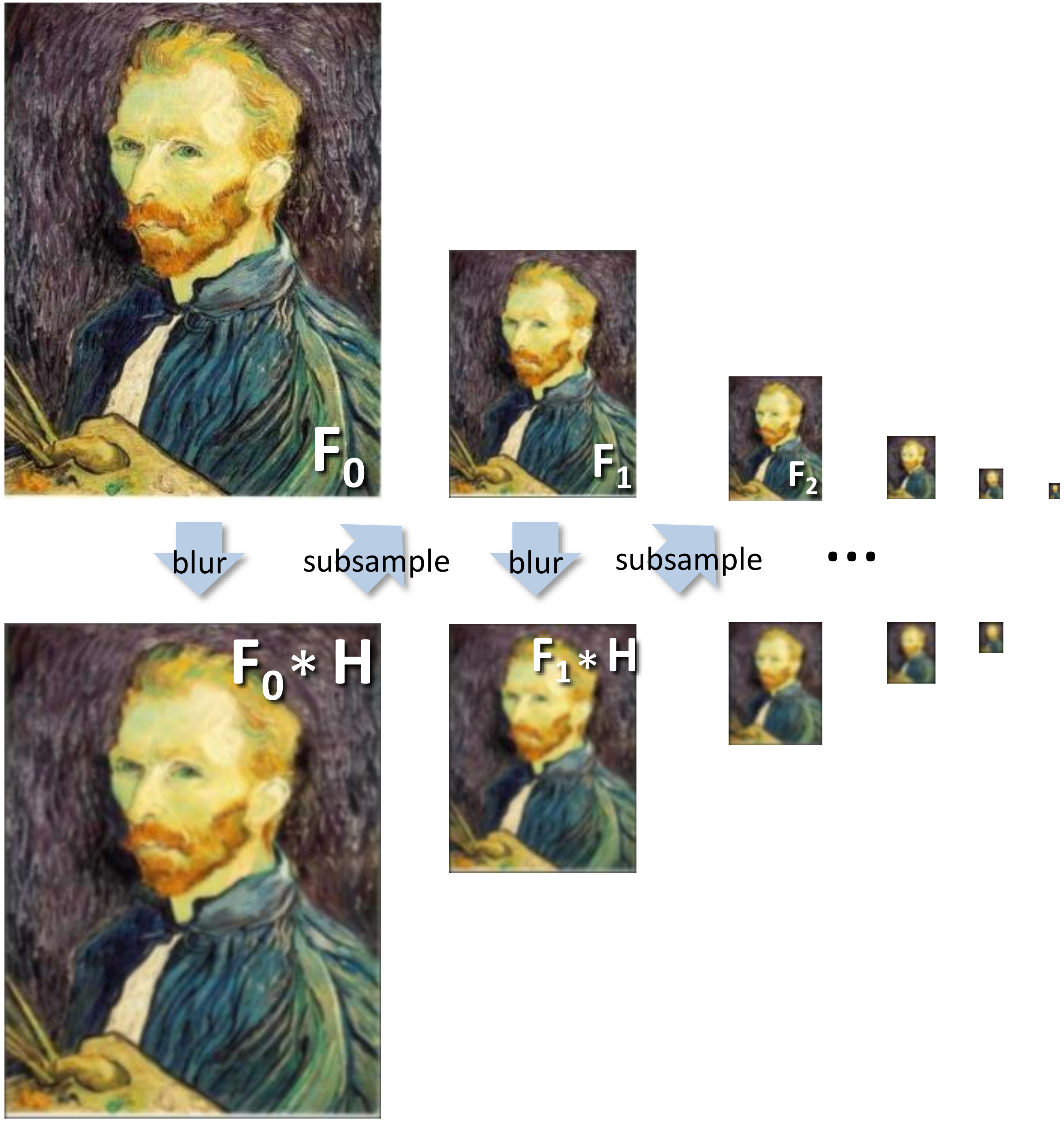
Image Pyramids

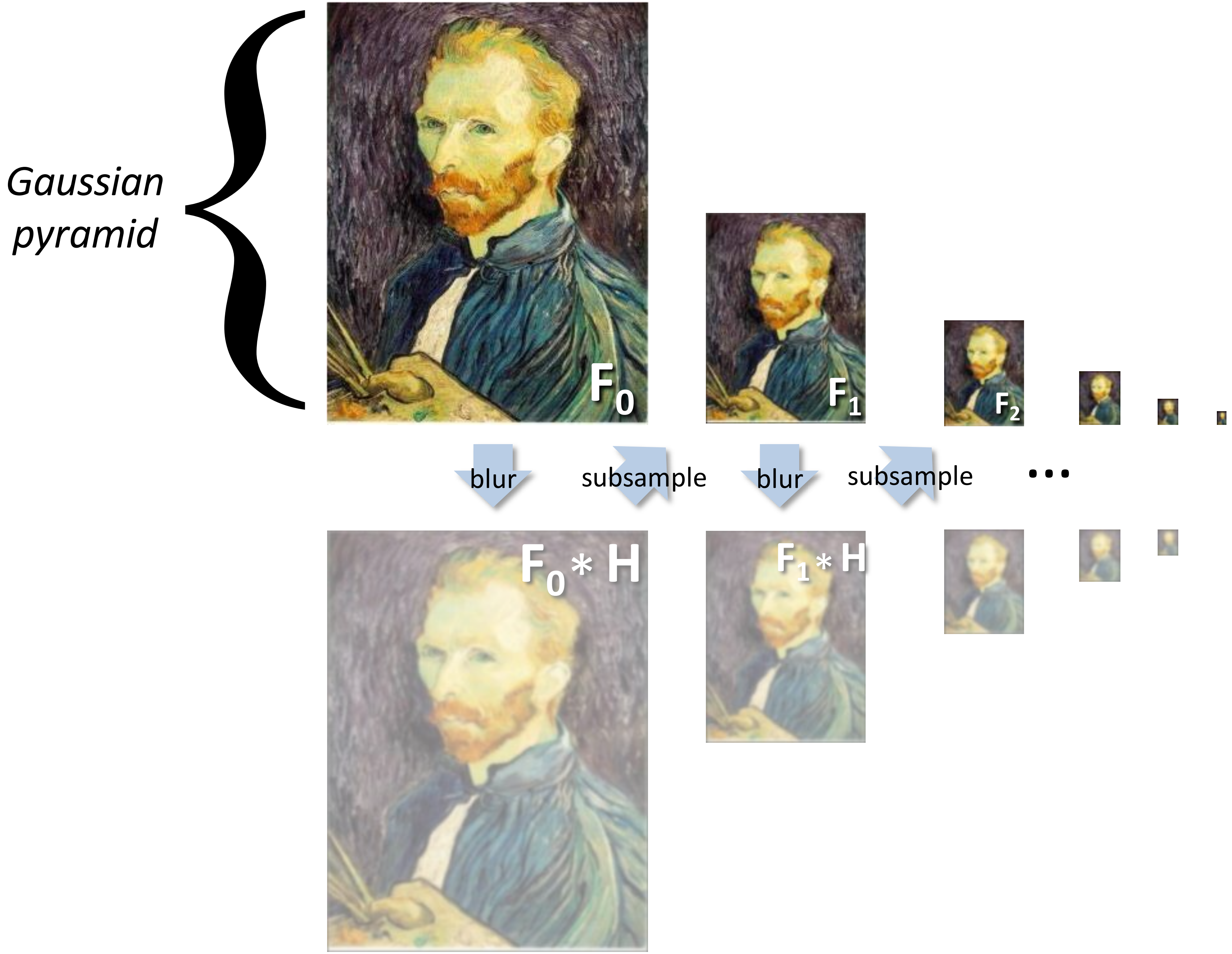


Project 1 function:
vis_hybrid_image.m

Wikipedia – Image Pyramids

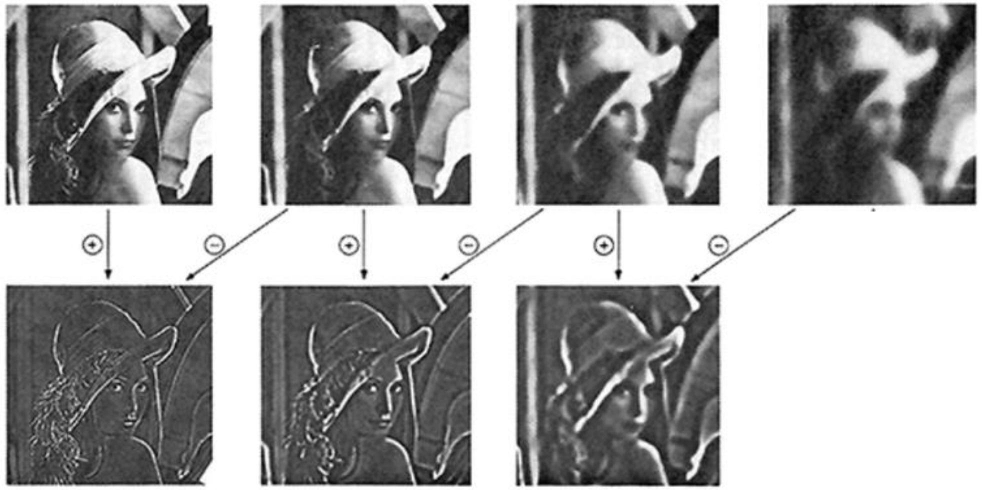
Gaussian pyramid




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

Laplace Pyramid

- Derive from Gaussian pyramid
  - G1=pydn(G0); G2=pydn(G1), …
  - One level of laplace pyramid is difference between approximated and original Gaussian pyramid levels
  - L0=G0-pyup(G1); L1 = G1-pyup(G2)



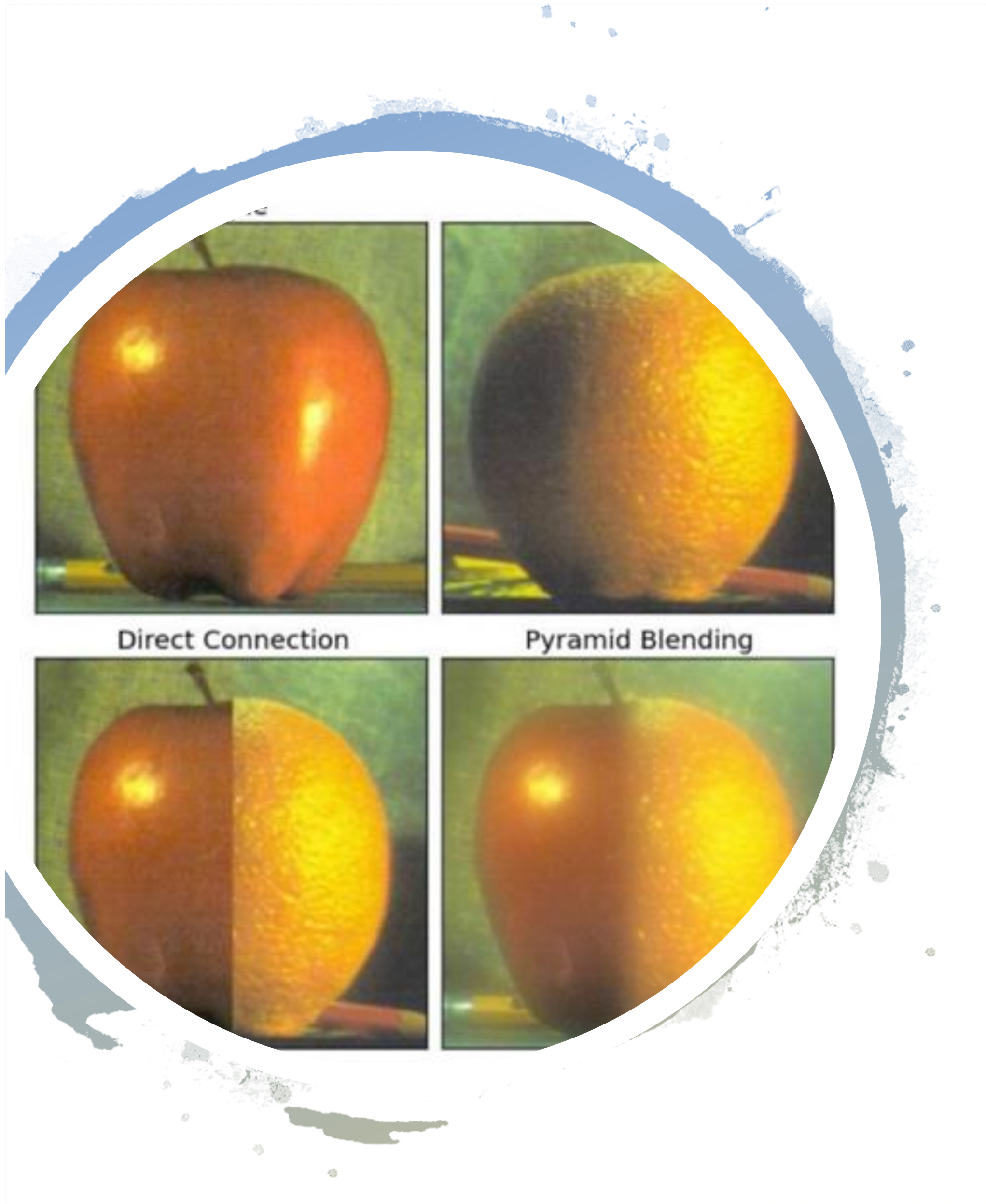


Image composting

- Generate L-pyramid of orange
- Generate L-pyramid of apple
- Combine two pyramids
  - For all levels, one half from one pyramid, the other half from another
- Reconstruct image from combine pyramid

Summary

- Product in time domain = convolution in freq domain
  - Sampling can be represented as signal multiplied by pulse train
  - Infinite repeated copy in frequency domain
  - When copies overlaps => aliasing
- Downsampling naively will lead to aliasing
  - Solution: apply low pass filter before downsample
- Should apply low pass filter after upsampling
- Laplace pyramid and Gaussian pyramid
- Hybrid image and composting image