Review: Image Filtering

Image Filtering

- Point Operation
- Neighborhood Operation -> Filtering-> **Correlation** & **Convolution**

$$(f * I)(x, y) = \sum_{i,j=-\infty}^{\infty} f(i, j)I(x + i, y + j)$$

$$(f * I)(x, y) = \sum_{i,j=-\infty}^{\infty} f(i, j)I(x - i, y - j)$$

Image Filtering

- Point Operation
- Neighborhood Operation -> Filtering-> **Correlation** & **Convolution**

$$(f * I)(x,y) = \sum_{i,j=-\infty}^{\infty} f(i,j)I(x+i, y+j) \qquad (f * I)(x,y) = \sum_{i,j=-\infty}^{\infty} f(i,j)I(x-i, y-j)$$

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Image

a	b	c
d	e	f
g	h	i

Kernel
[filter]

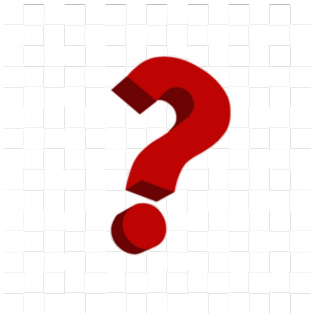


Image Filtering

- Neighborhood Operation -> Filtering-> **Correlation** & **Convolution**

$$(f * I)(x,y) = \sum_{i,j=-\infty}^{\infty} f(i,j)I(x+i, y+j)$$

$$(f * I)(x,y) = \sum_{i,j=-\infty}^{\infty} f(i,j)I(x-i, y-j)$$

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Image

a	b	c
d	e	f
g	h	i

Kernel
[filter]

i	h	g
f	e	d
c	b	a

Correlation

a	b	c
d	e	f
g	h	i

Convolution

Image Filtering

- Neighborhood Operation -> Filtering-> **Correlation** & **Convolution**
- Separable filters [Cost & Property Understanding]

Gaussian $\frac{1}{16}$

1	2	1
2	4	2
1	2	1

Sobel

1	0	-1
2	0	-2
1	0	-1

1	2	1
0	0	0
-1	-2	-1

Box [Average] $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Prewitt

1	0	-1
1	0	-1
1	0	-1

1	1	1
0	0	0
-1	-1	-1

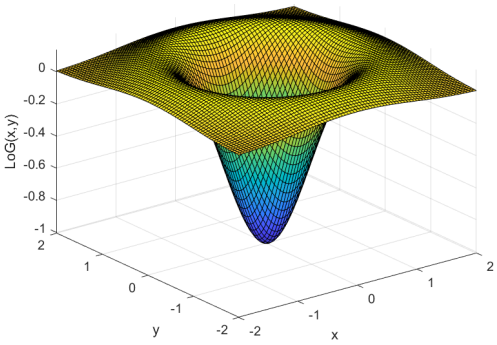
Edge Detection

Sobel

Prewitt

OpenCV has a built-in function **cv2.filter2D()** to convolve a kernel with an image.

https://theailearner.com/tag/cv2-filter2d/



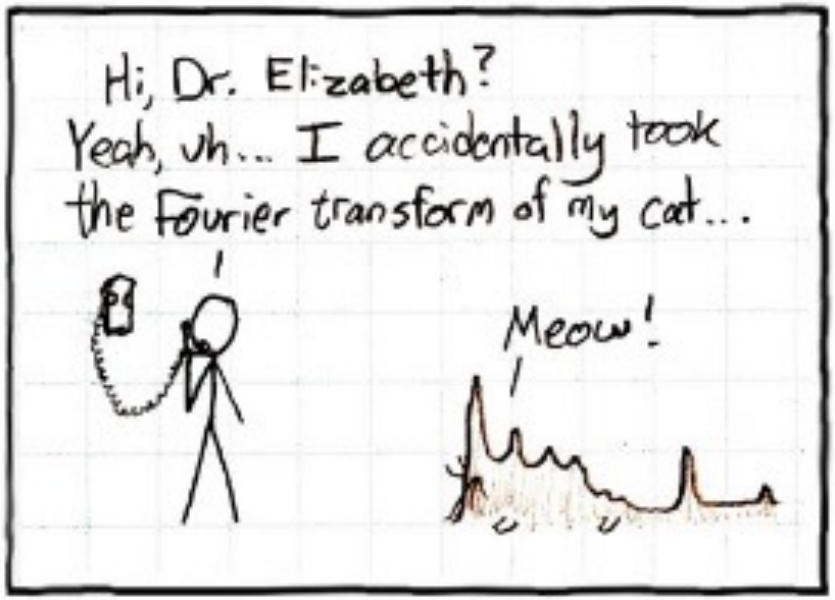
Laplacian Filter [2nd-order Derivative]

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

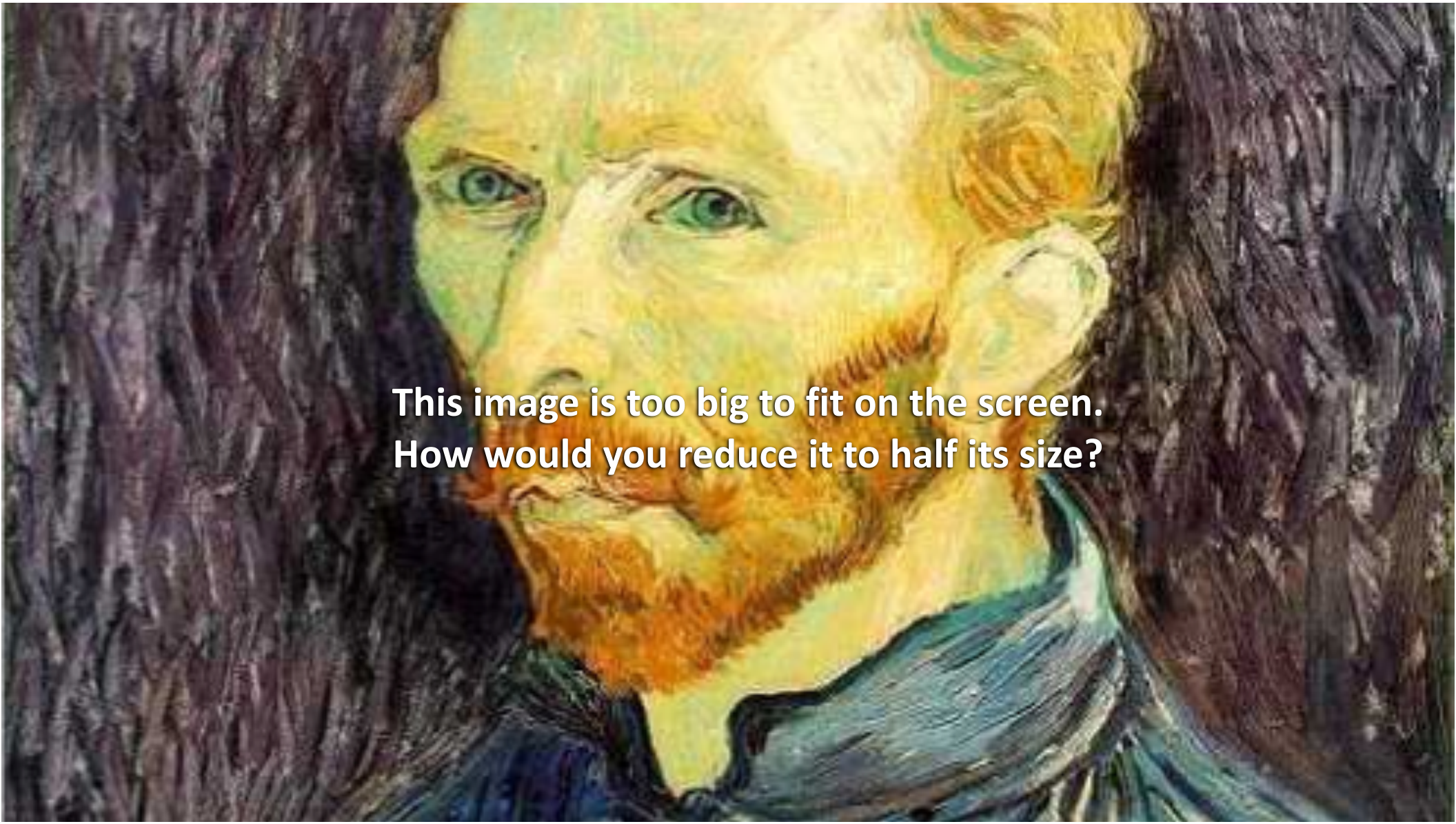
https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm

Image pyramids and frequency domain

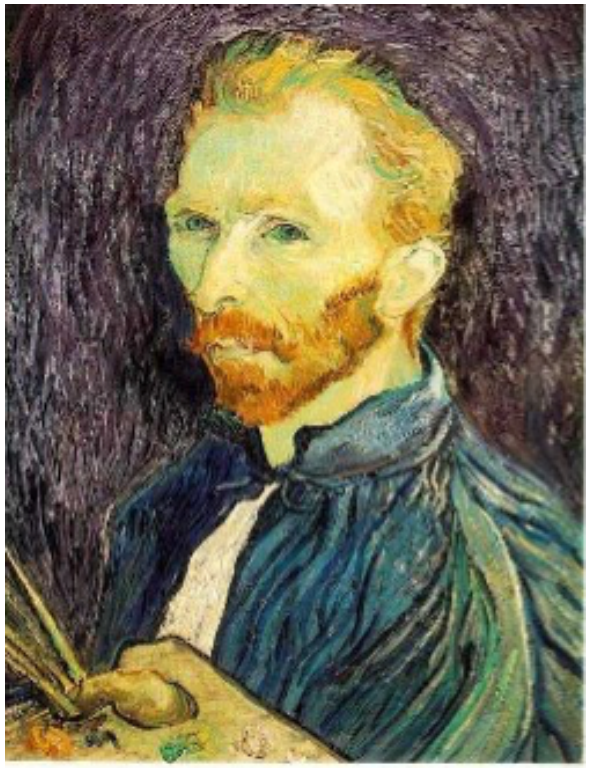


Computer Vision
Fall 2022, Lecture 3

Image downsampling


This image is too big to fit on the screen.
How would you reduce it to half its size?

Naïve image downsampling



1/2

Throw away half the rows and columns

delete even rows
delete even columns

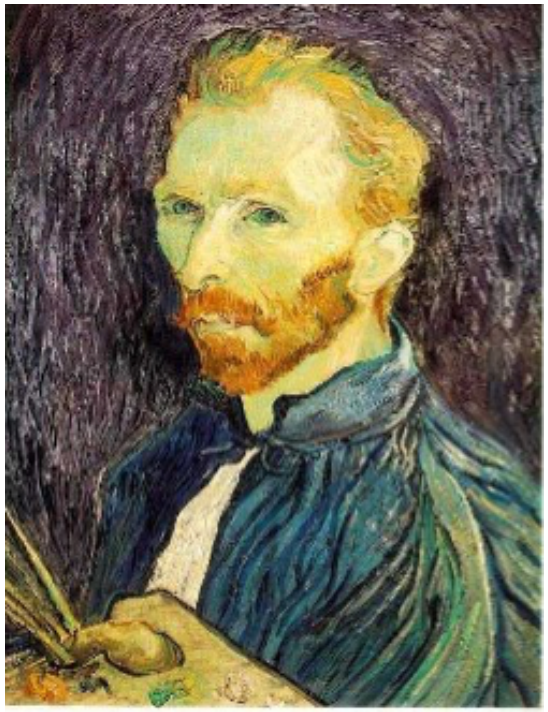
1/4

delete even rows
delete even columns

1/8

What is the problem with this approach?

Naïve image downsampling



1/2

1/4 (2x zoom)

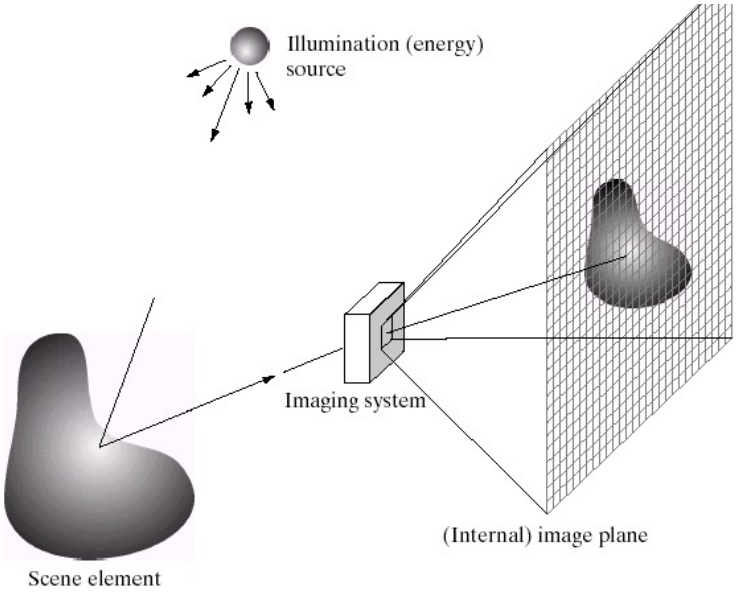
1/8 (4x zoom)

Why is the 1/8 image so pixelated (and do you know what this effect is called)?

Aliasing

Aliasing occurs when an oscilloscope does not sample the signal fast enough to construct an accurate waveform record. The signal frequency is misidentified, and the waveforms displayed on an oscilloscope become indistinguishable. Aliasing is basically a form of undersampling.

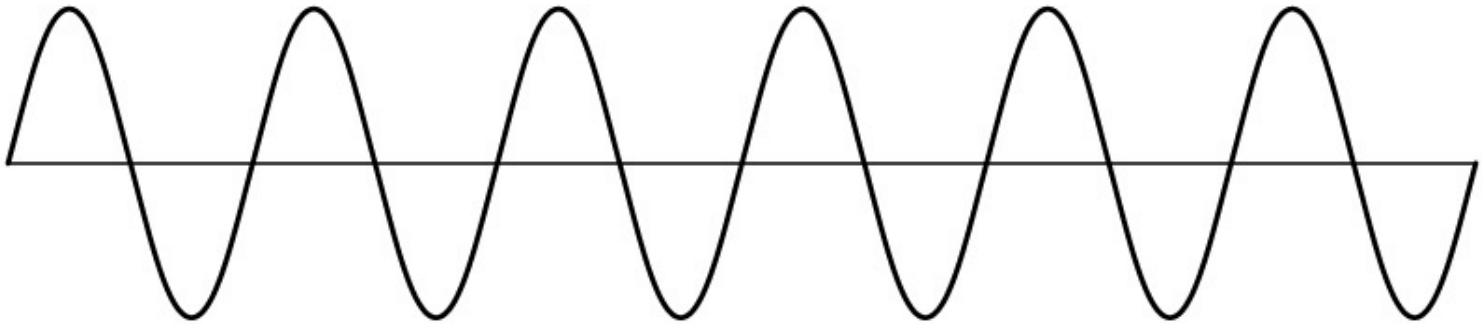
Reminder



Images are a *discrete*, or *sampled*, representation of a *continuous* world

Sampling

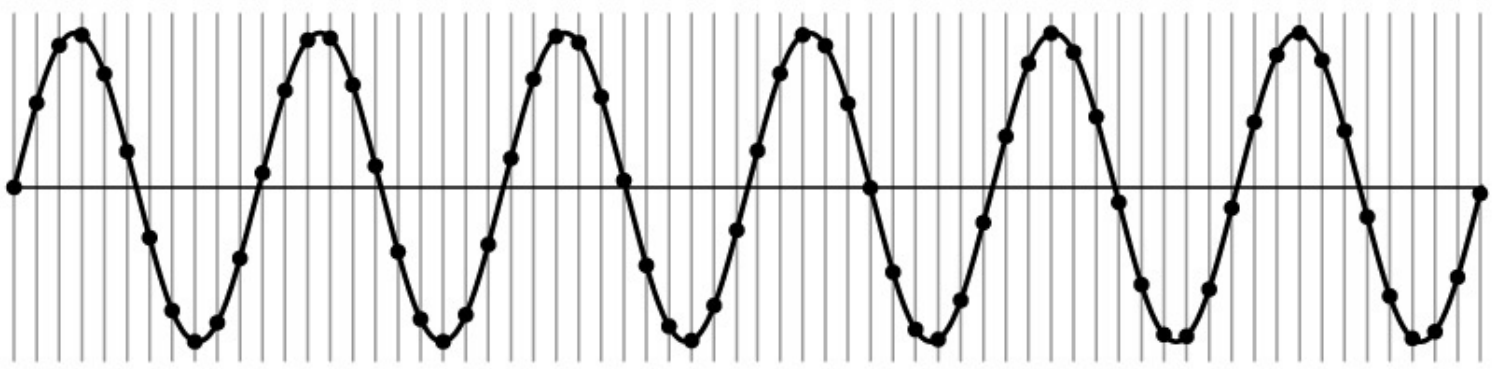
Very simple example: a sine wave



How would you discretize this signal?

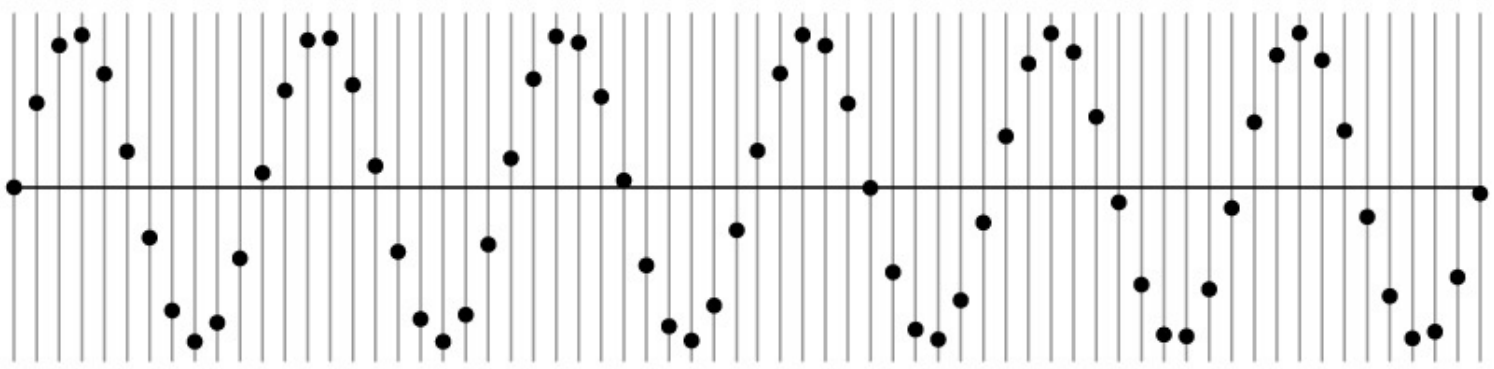
Sampling

Very simple example: a sine wave



Sampling

Very simple example: a sine wave

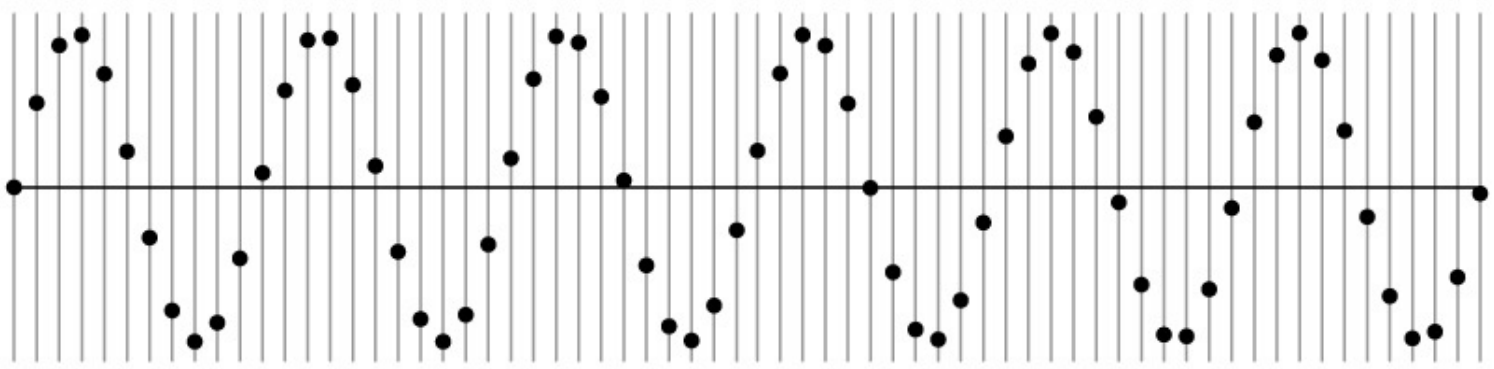


How many samples should I take?

Can I take as *many* samples as I want?

Sampling

Very simple example: a sine wave

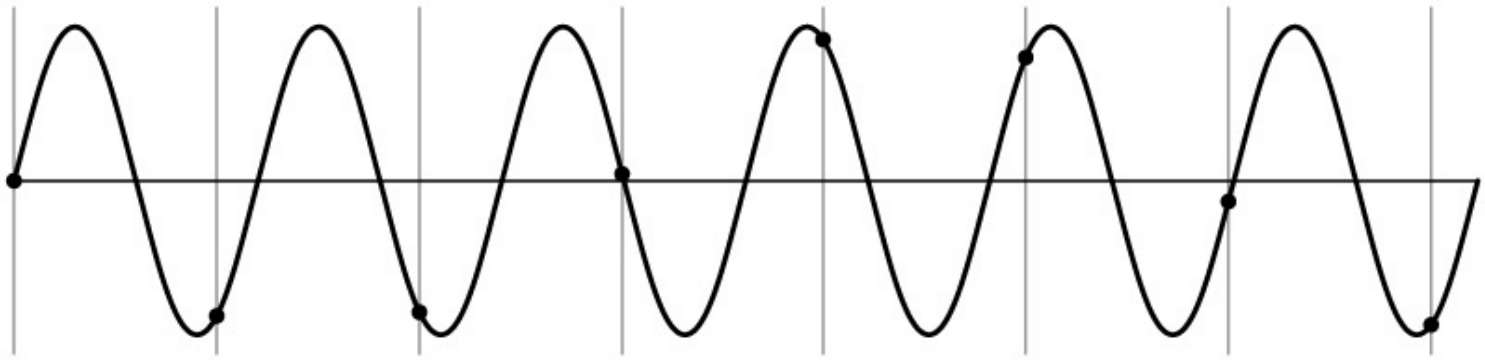


How many samples should I take?

Can I take as *few* samples as I want?

Undersampling

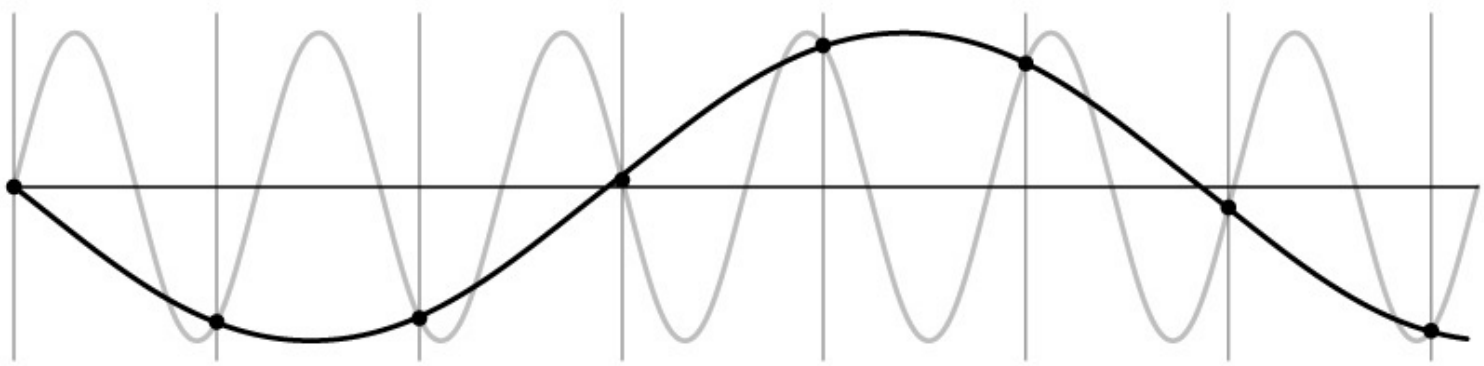
Very simple example: a sine wave



Unsurprising effect: information is lost.

Undersampling

Very simple example: a sine wave



Unsurprising effect: information is lost.

Surprising effect: can confuse the signal with one of *lower* frequency.

Aliasing in textures

Aliasing in photographs

This is also known as “**moire**” pattern.

Anti-aliasing

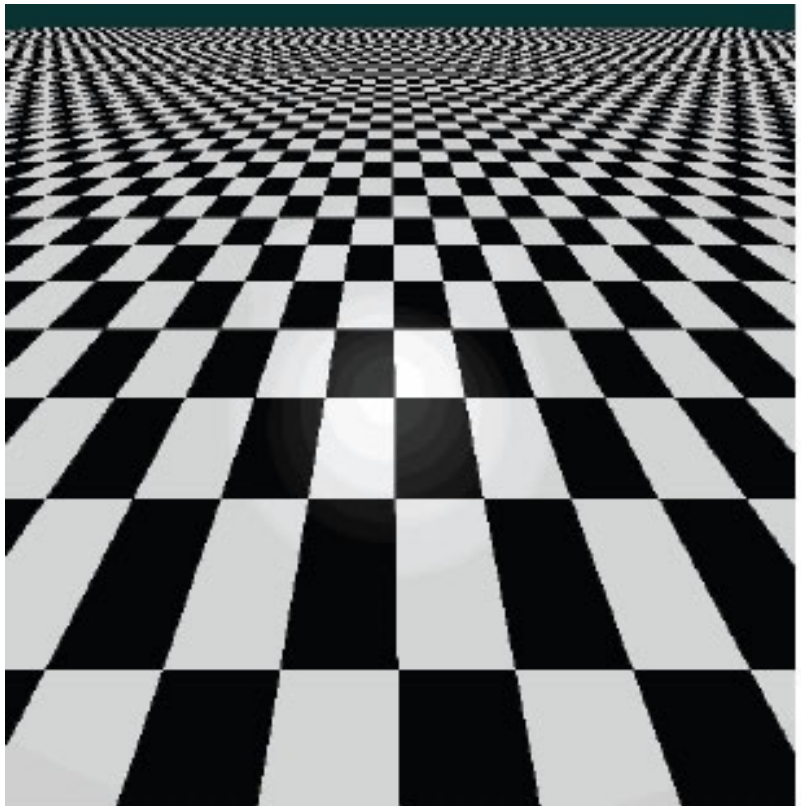
How would you deal with aliasing?

Anti-aliasing

How would you deal with aliasing?

Approach 1: Oversample the signal

Anti-aliasing in textures



aliasing artifacts

anti-aliasing by oversampling

Anti-aliasing

How would you deal with aliasing?

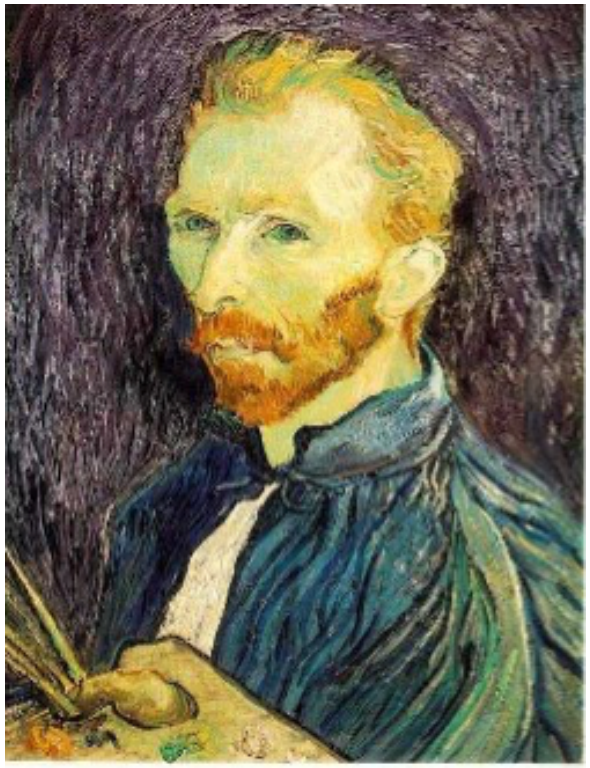
Approach 1: Oversample the signal

Approach 2: Smooth the signal

- Remove some of the detail effects that cause aliasing.
- Lose information, but better than aliasing artifacts.

How would you smooth a signal?

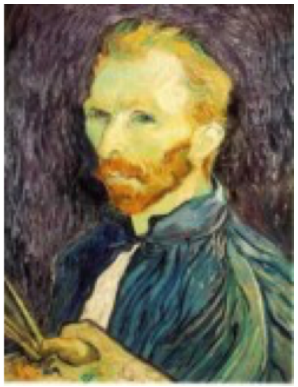
Better image downsampling



1/2

Apply a smoothing filter first, then throw away half the rows and columns

Gaussian filter
delete even rows
delete even columns

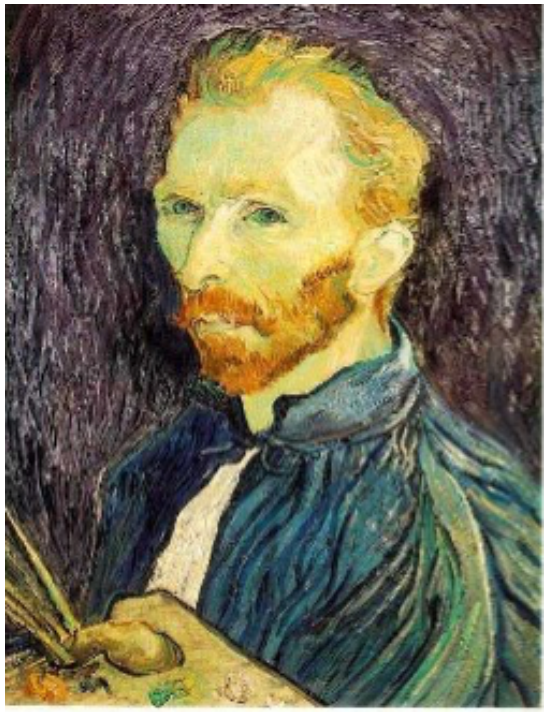


1/4

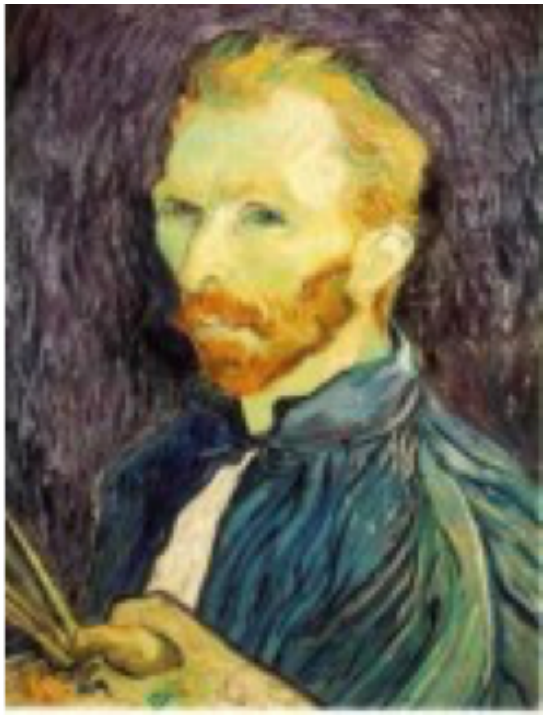
Gaussian filter
delete even rows
delete even columns

1/8

Better image downsampling



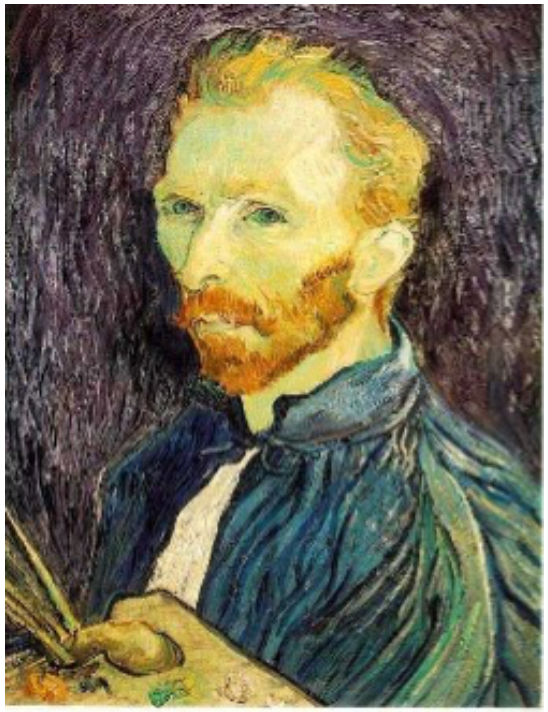
1/2



1/4 (2x zoom)

1/8 (4x zoom)

Naïve image downsampling



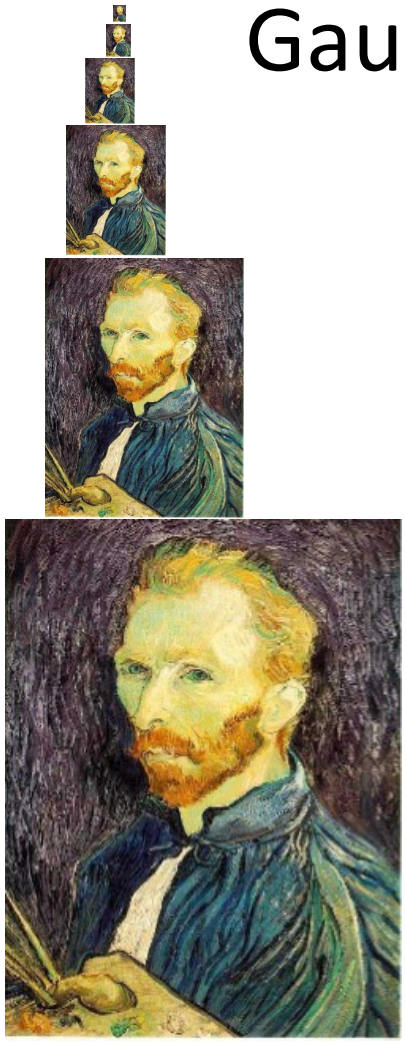
1/2

1/4 (2x zoom)

1/8 (4x zoom)

Gaussian image pyramid

Gaussian image pyramid



The name of this sequence of subsampled images

Constructing a Gaussian pyramid

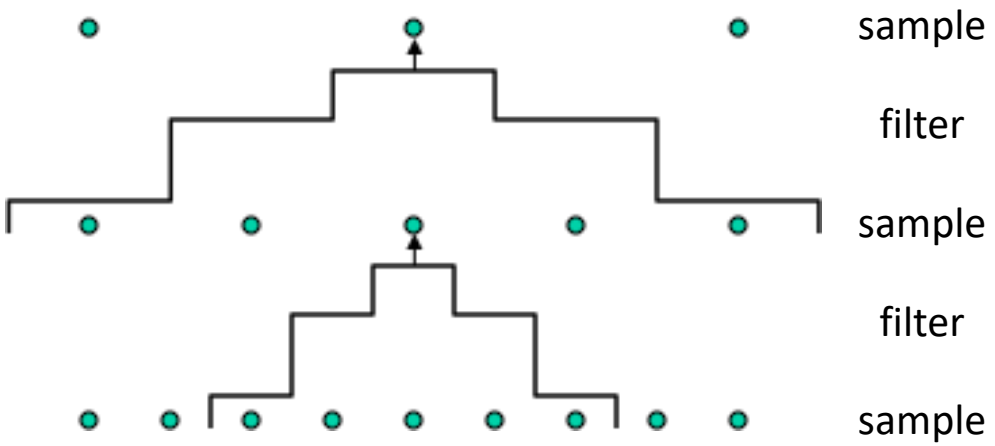
Algorithm

repeat:

filter

subsample

until min resolution reached



Question: How much bigger than the original image is the whole pyramid?

Constructing a Gaussian pyramid

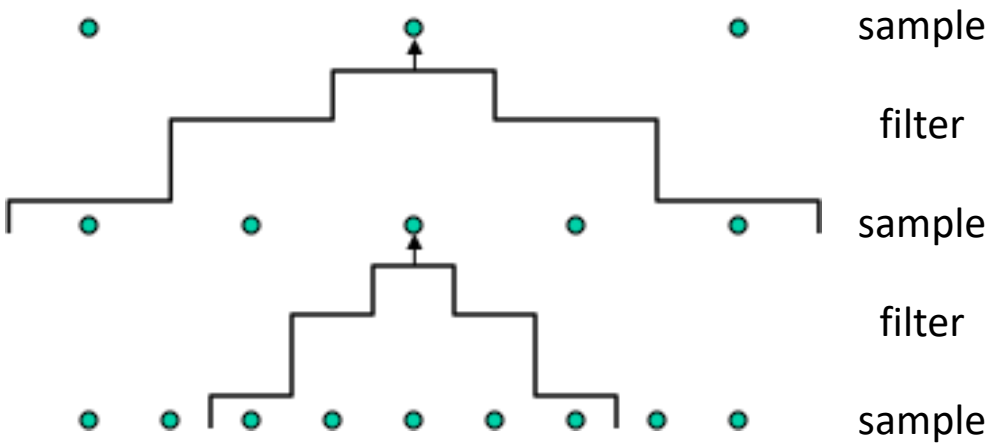
Algorithm

repeat:

filter

subsample

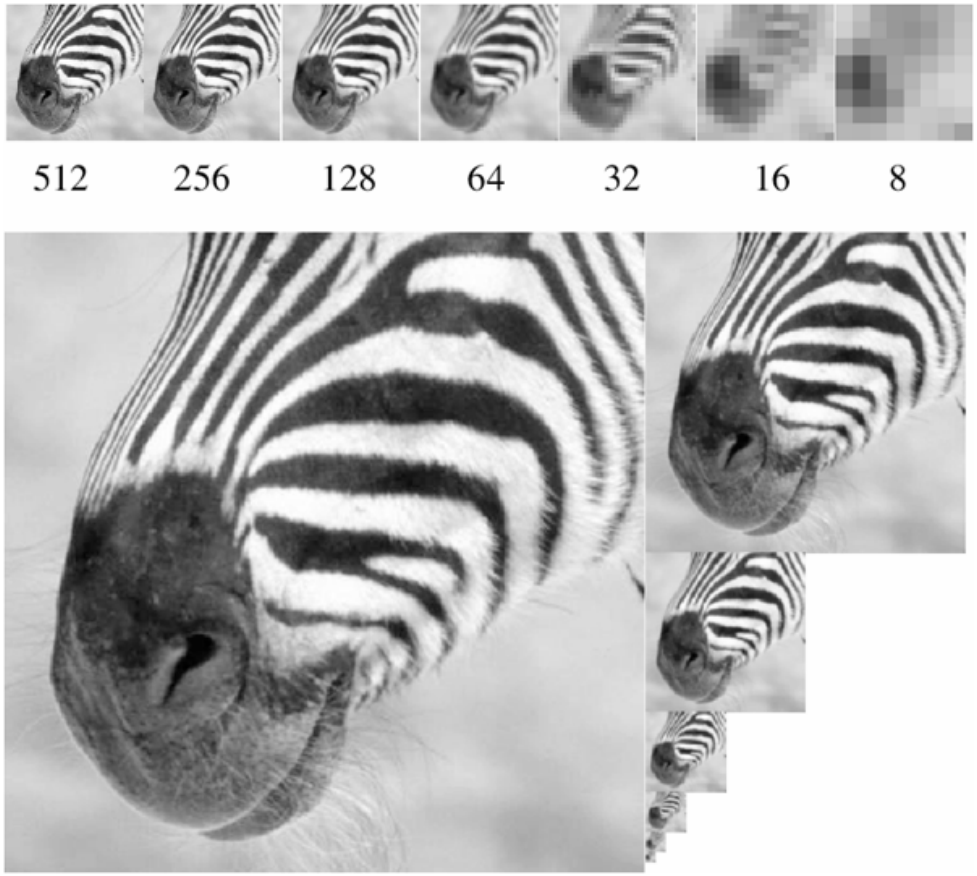
until min resolution reached



Question: How much bigger than the original image is the whole pyramid?

Answer: Just 4/3 times the size of the original image! (How did I come up with this number?)

Some properties of the Gaussian pyramid



What happens to the details of the image?

- They get smoothed out as we move to higher levels.

What is preserved at the higher levels?

- Mostly large uniform regions in the original image.

How would you reconstruct the original image from the image at the upper level?

- That's not possible.

Blurring is lossy

level 0

level 1 (before downsampling)

What does the residual look like?

Blurring is lossy

level 0

-

level 1 (before downsampling)

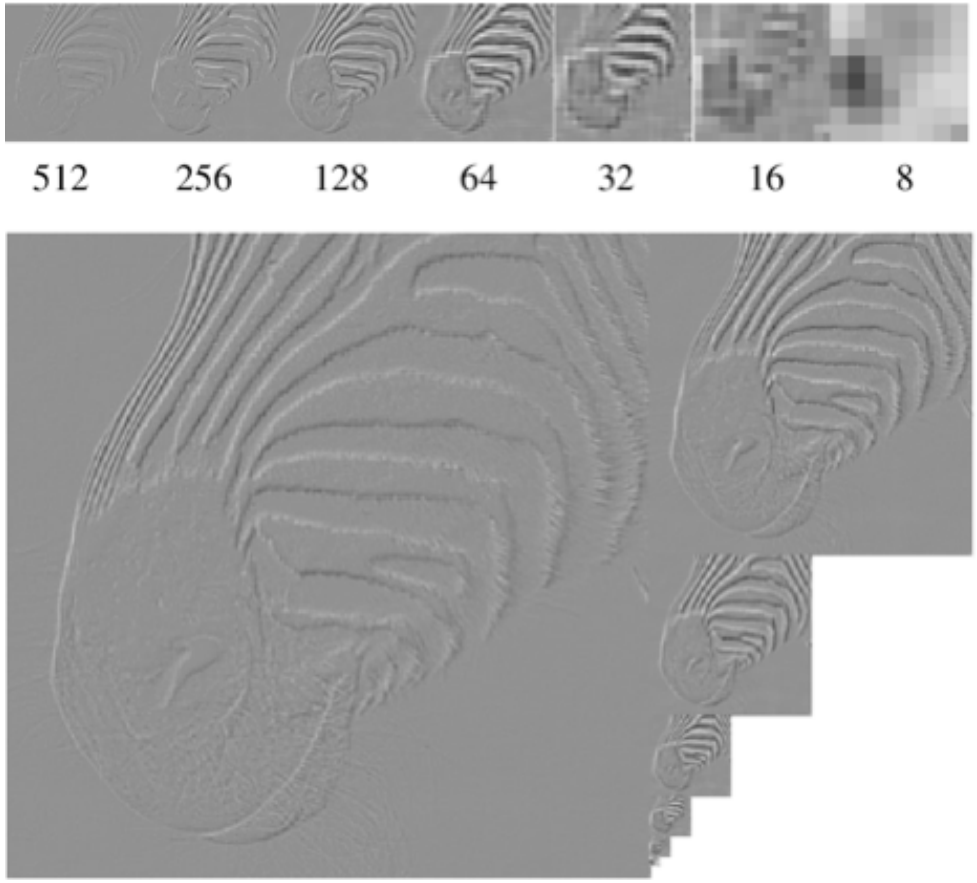
=

residual

Can we make a pyramid that is lossless?

Laplacian image pyramid

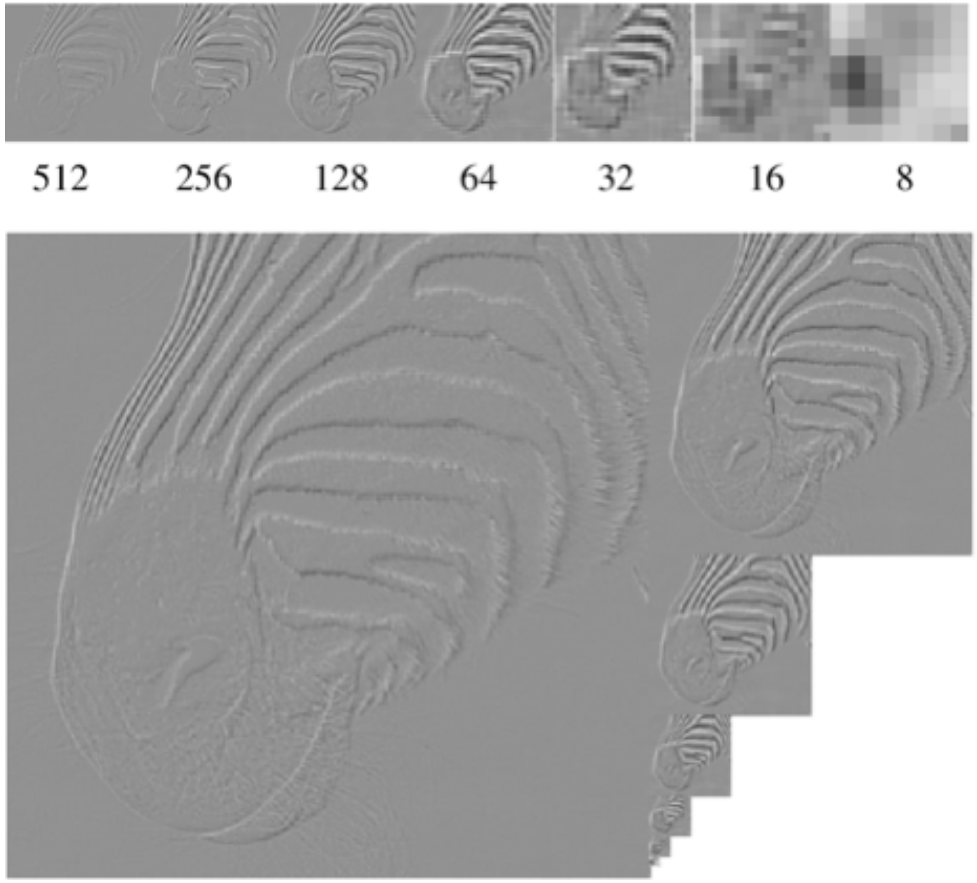
Laplacian image pyramid



At each level, retain the residuals instead of the blurred images themselves.

Can we reconstruct the original image using the pyramid?

Laplacian image pyramid



At each level, retain the residuals instead of the blurred images themselves.

Can we reconstruct the original image using the pyramid?

- Yes we can!

Deep residual learning for image recognition
K He, X Zhang, S Ren, J Sun - Proceedings of the IEEE …, 2016 - openaccess.thecvf.com
Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We …
☆ Save 99 Cite Cited by 131288 Related articles »

Let’s start by looking at just one level

level 0

=

level 1 (upsampled)

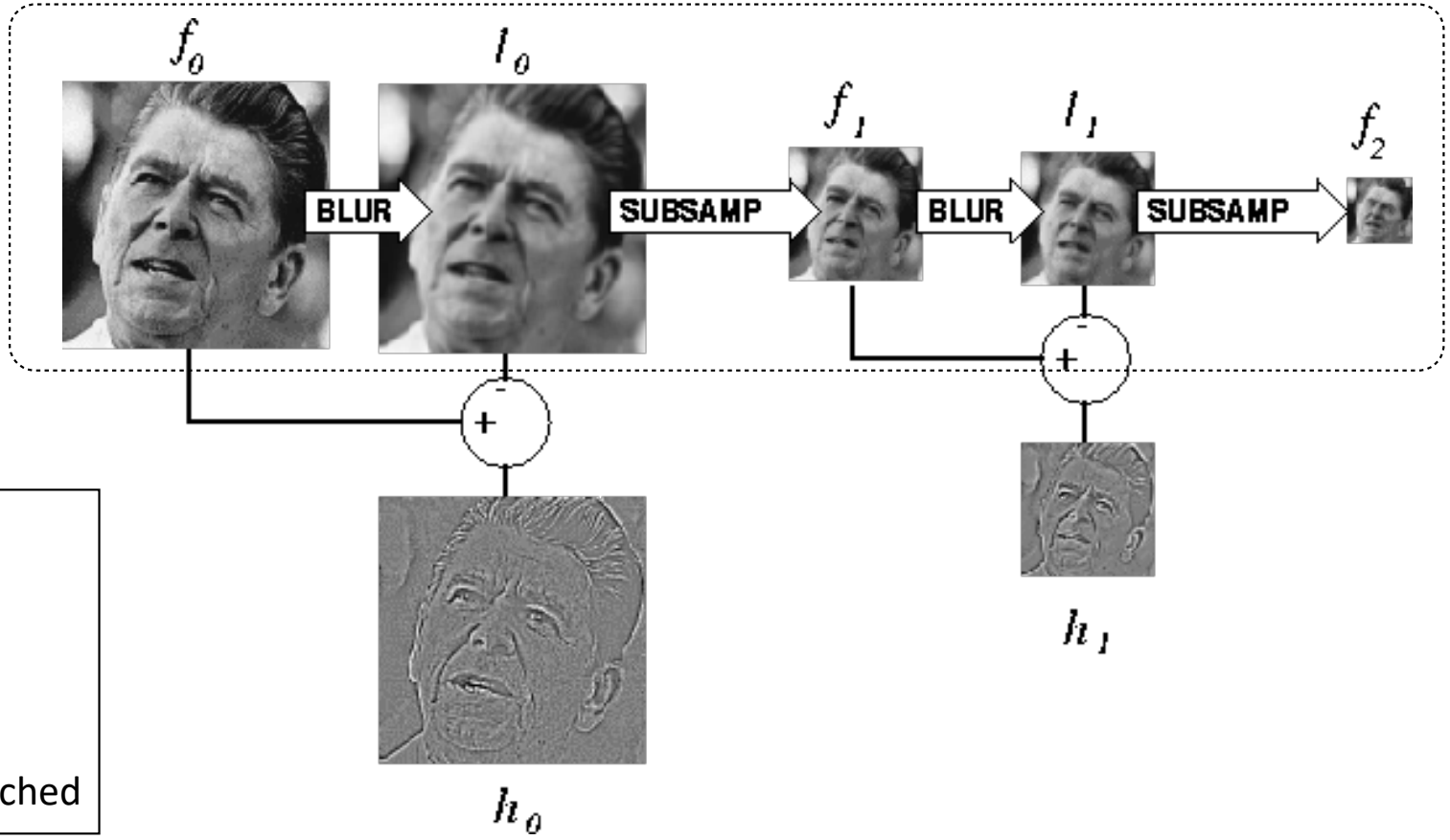
+

residual

Does this mean we need to store both residuals and the blurred copies of the original?

Constructing a Laplacian pyramid

What is this part?



Algorithm

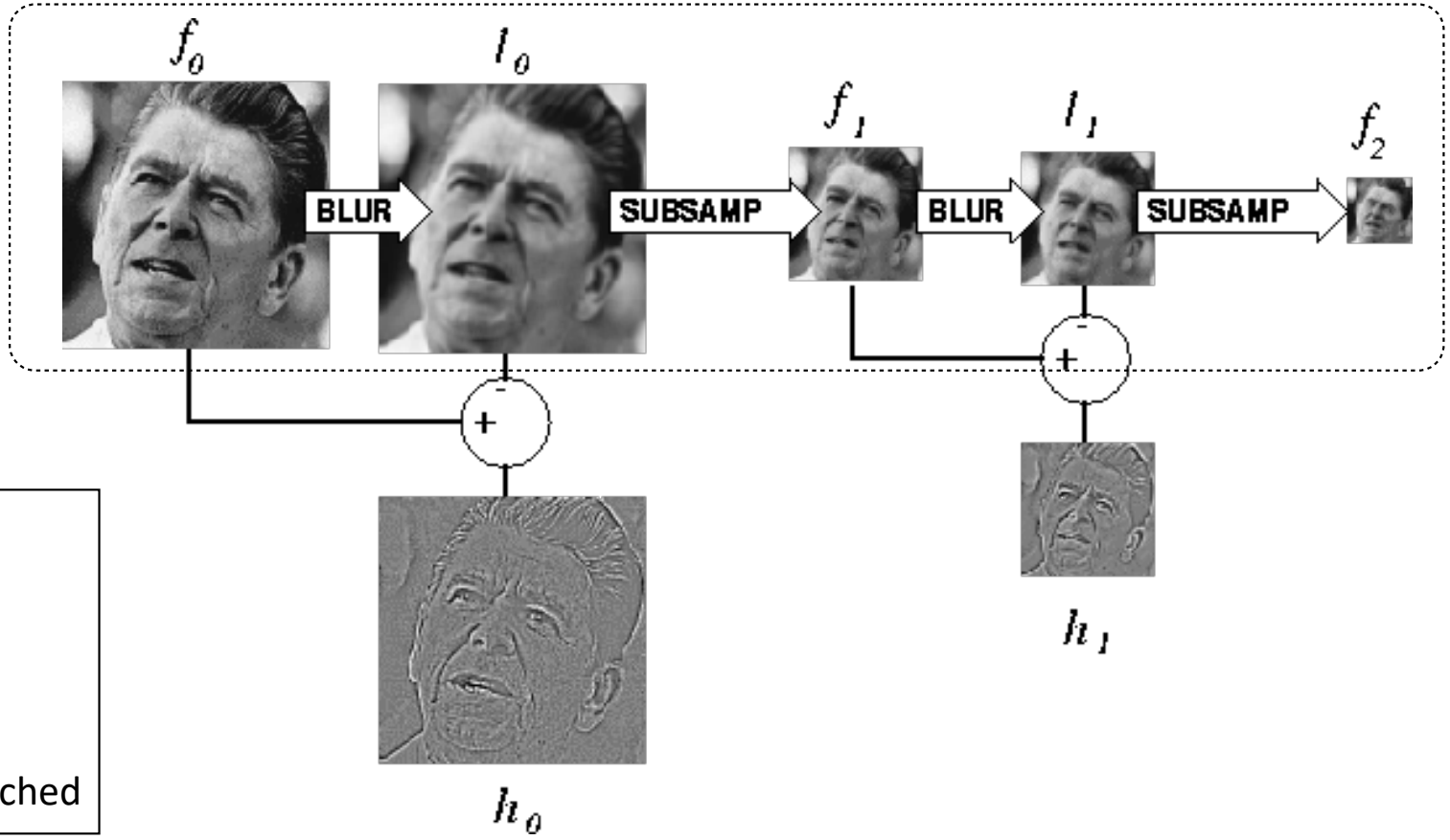
repeat:

- filter
- compute residual
- subsample

until min resolution reached

Constructing a Laplacian pyramid

It's a Gaussian pyramid.



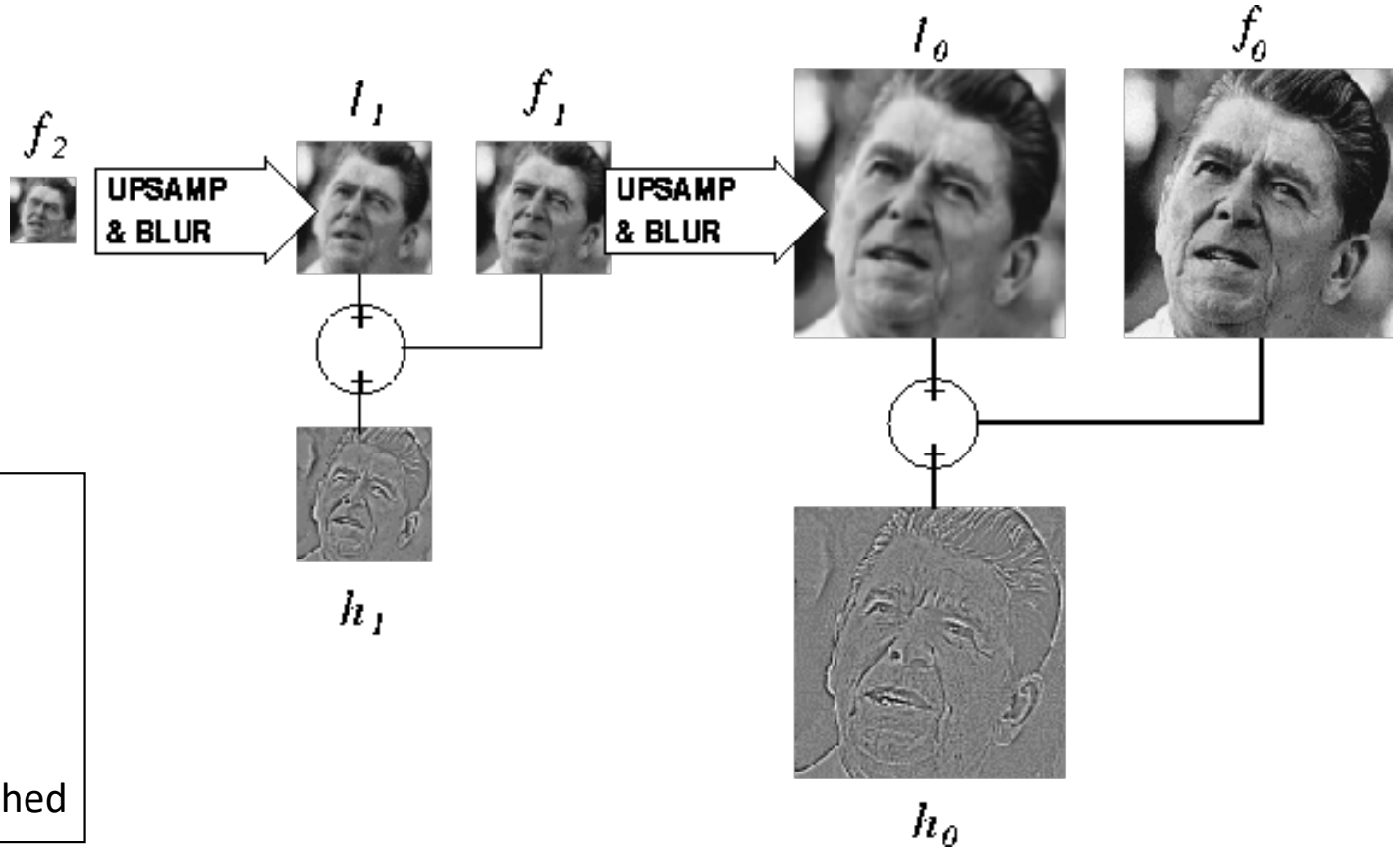
Algorithm

repeat:

- filter
- compute residual
- subsample

until min resolution reached

Reconstructing the original image



Algorithm

repeat:

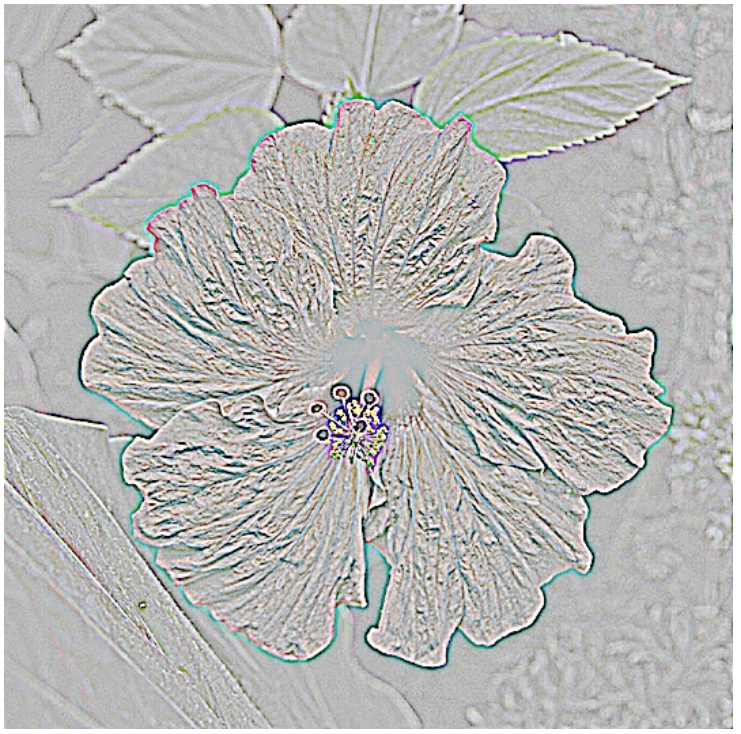
upsample

sum with residual

until orig resolution reached

Gaussian vs Laplacian Pyramid

Shown in opposite order for space.



Which one takes more space to store?

What are image pyramids used for?

image compression

multi-scale texture mapping

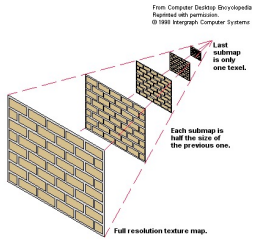
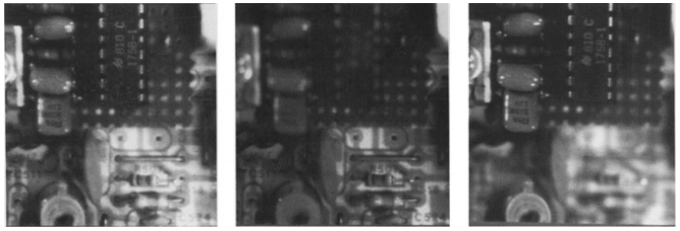
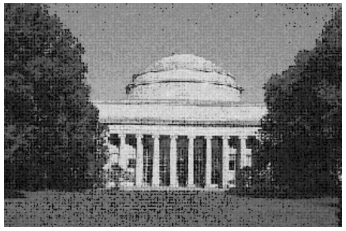


image blending

focal stack compositing

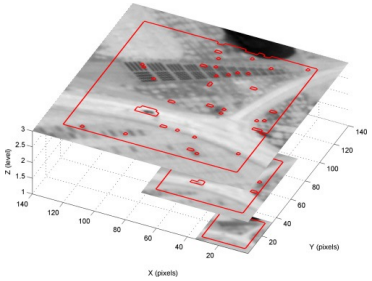


denoising



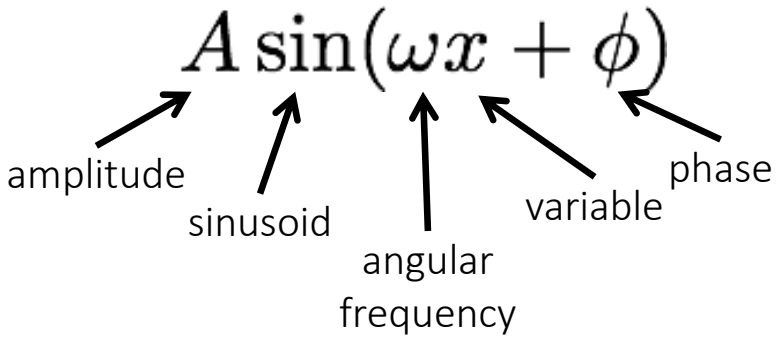
multi-scale detection

multi-scale registration



Fourier series

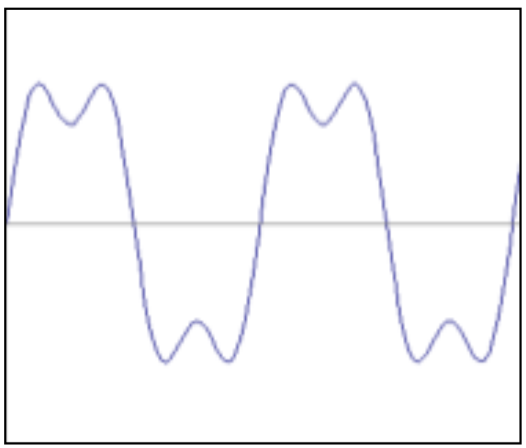
Basic building block



Fourier's claim: Add enough of these to get any *periodic* signal you want!

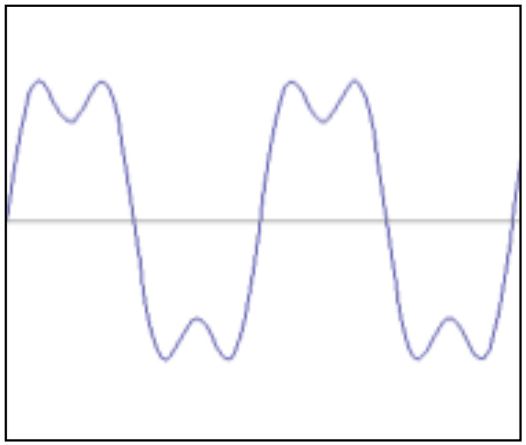
Examples

How would you generate this function?

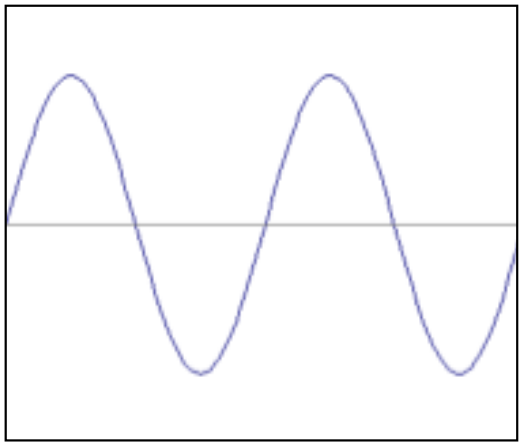
 = ? + ?

Examples

How would you generate this function?



=

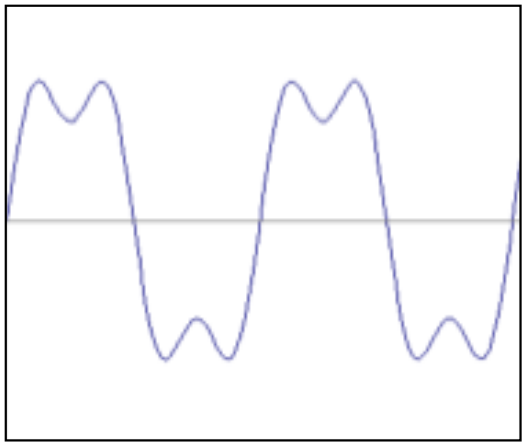


$\sin(2\pi x)$

+ ?

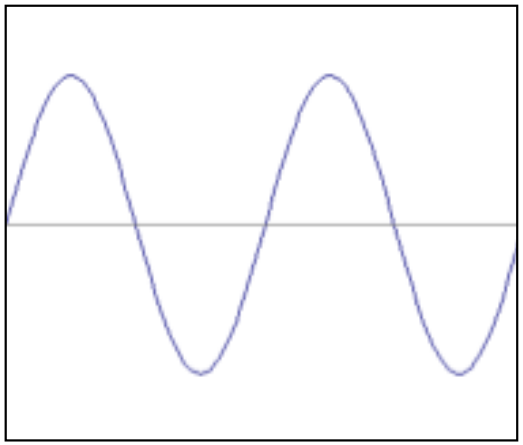
Examples

How would you generate this function?



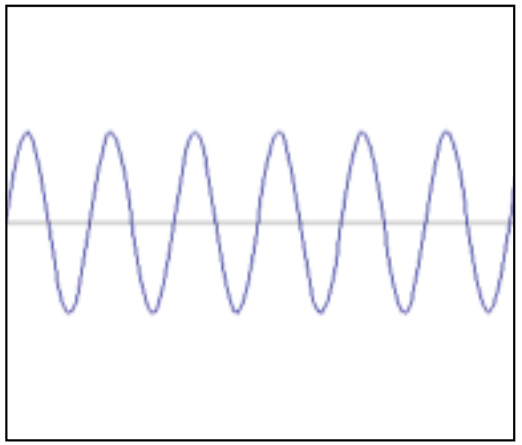
$f(x) = \sin(2\pi x) + \frac{1}{3}\sin(2\pi 3x)$

=



$\sin(2\pi x)$

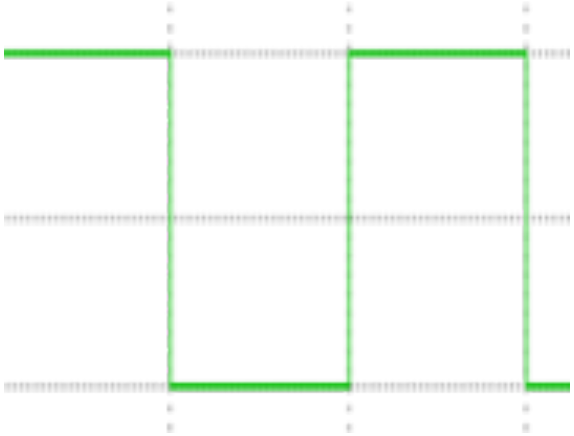
+



$\frac{1}{3}\sin(2\pi 3x)$

Examples

How would you generate this function?

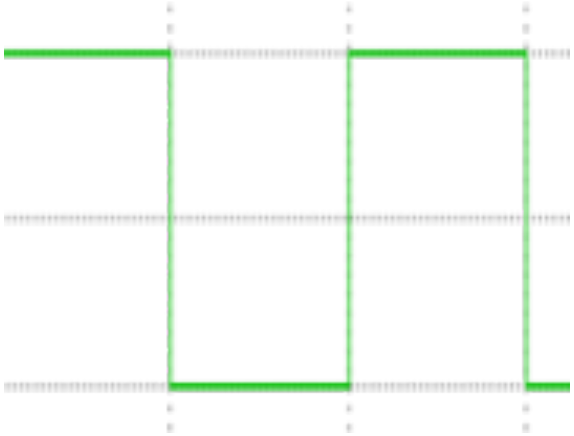


square wave

= ? + ?

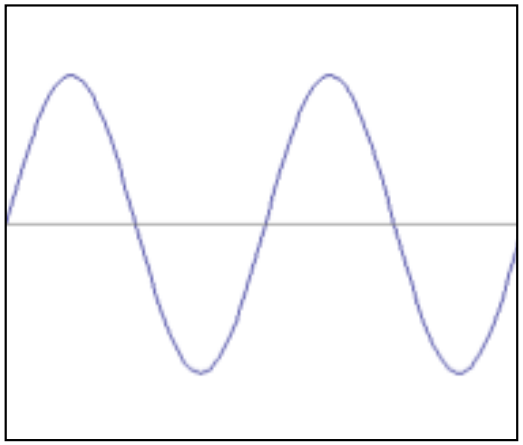
Examples

How would you generate this function?

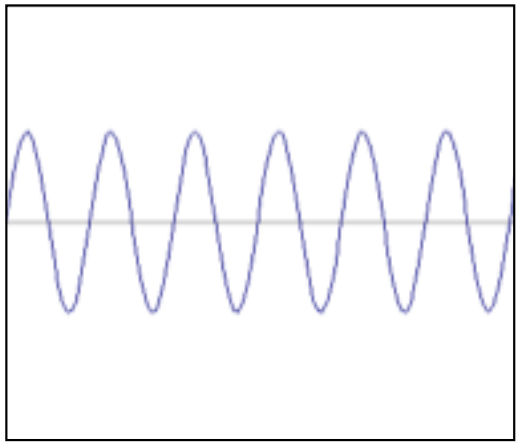


square wave

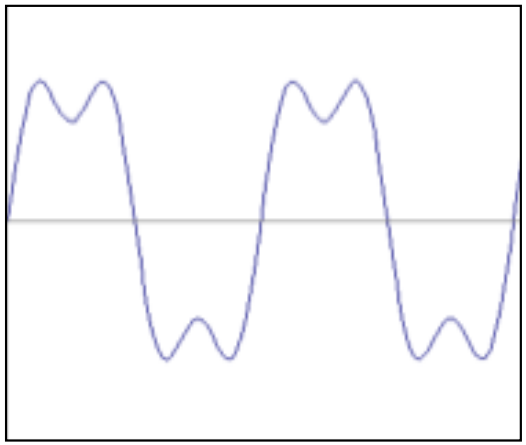
≈



+

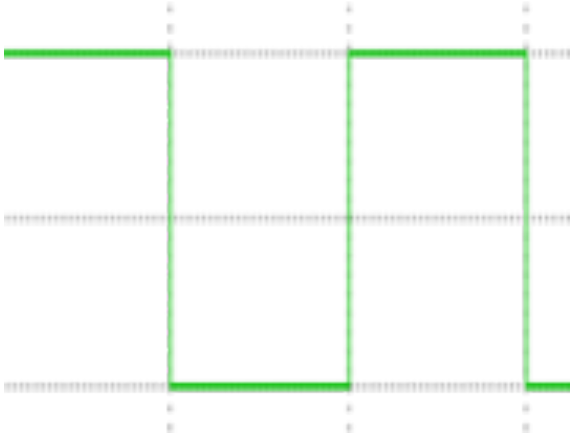


=



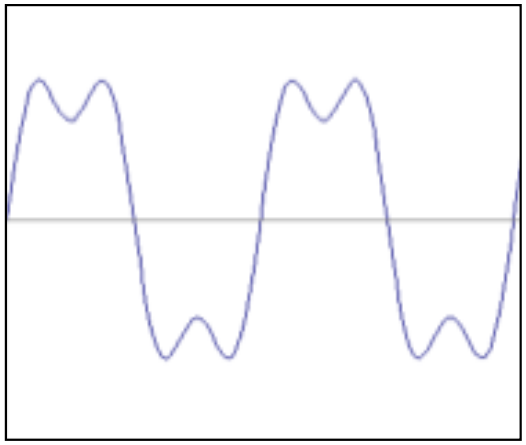
Examples

How would you generate this function?

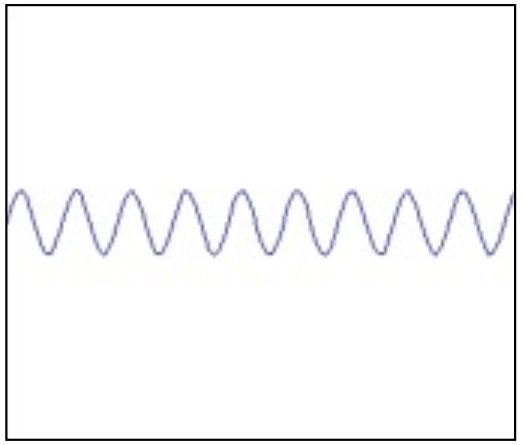


square wave

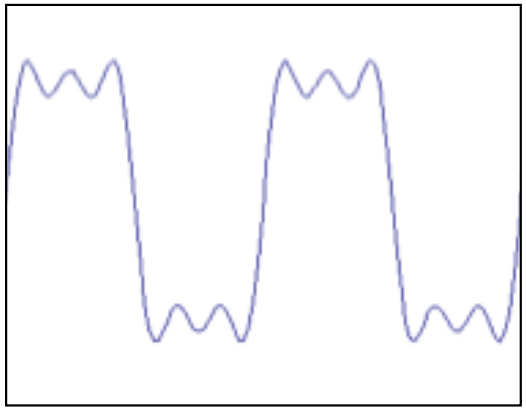
≈



+

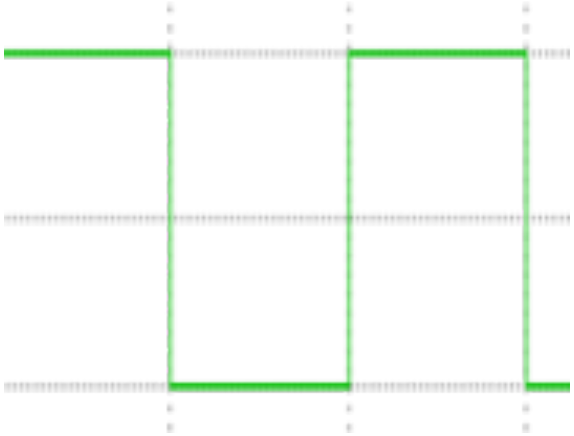


=

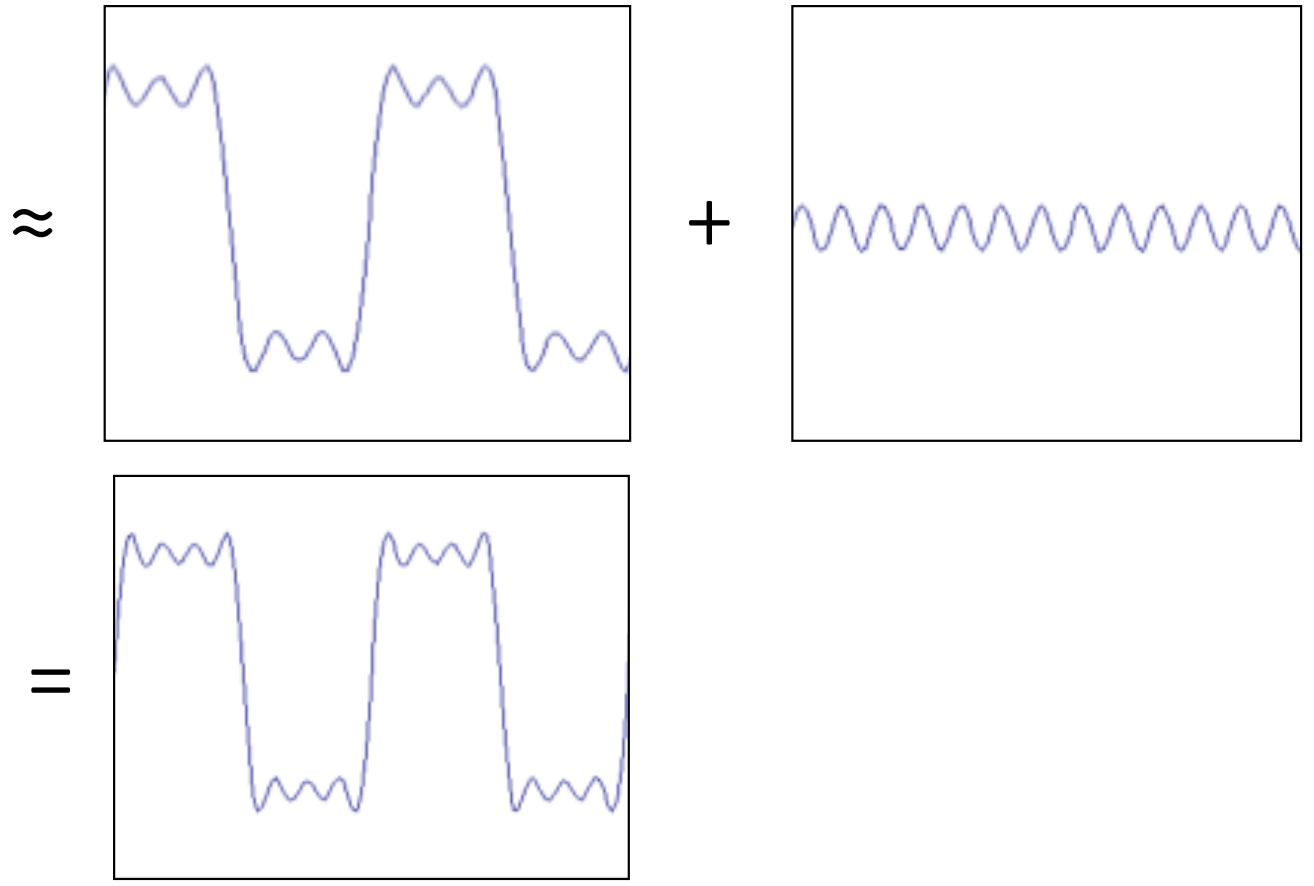


Examples

How would you generate this function?

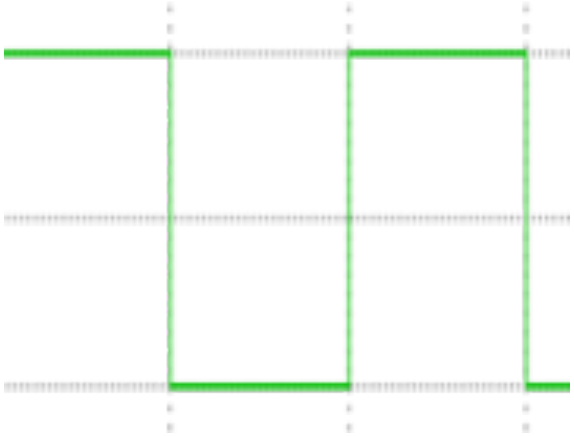


square wave

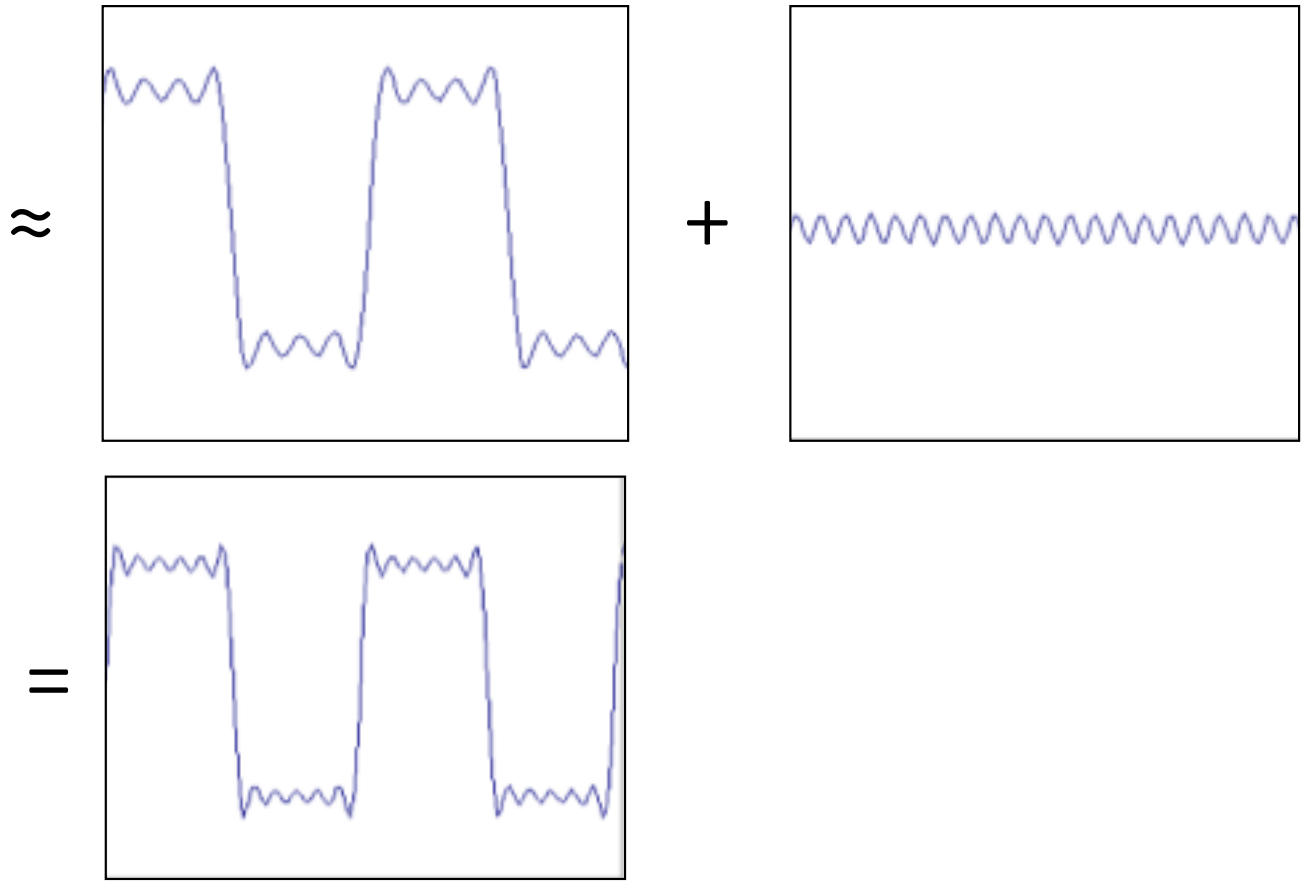


Examples

How would you generate this function?

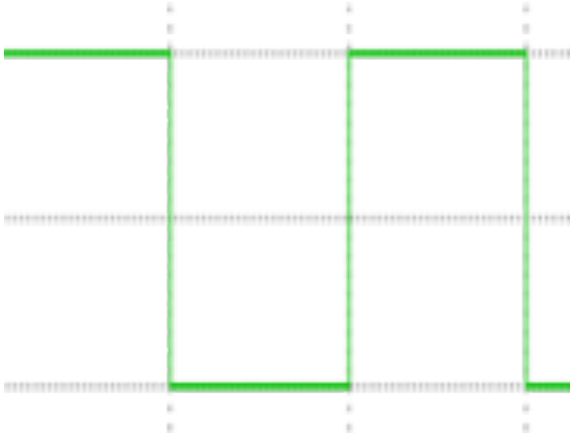


square wave



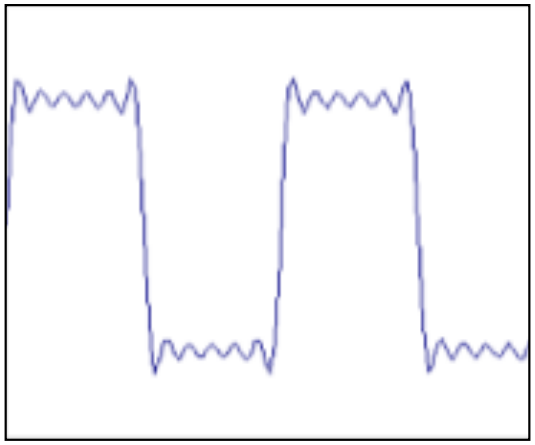
Examples

How would you generate this function?

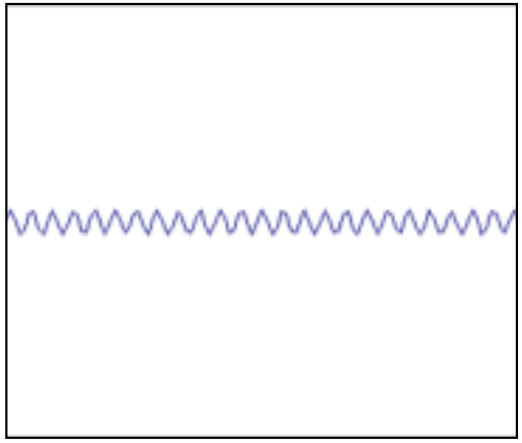


square wave

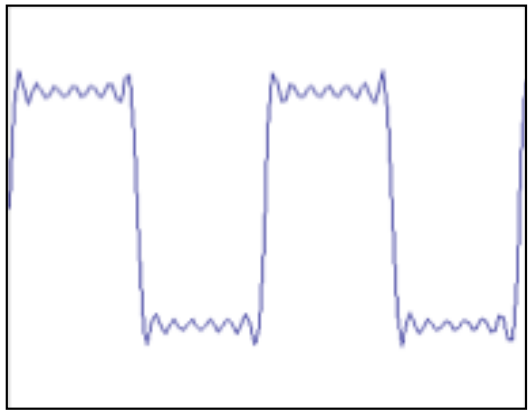
≈



+

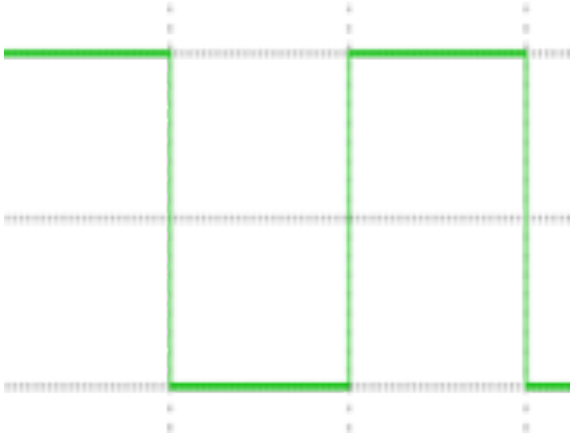


=



How would you express this mathematically?

Examples



square wave

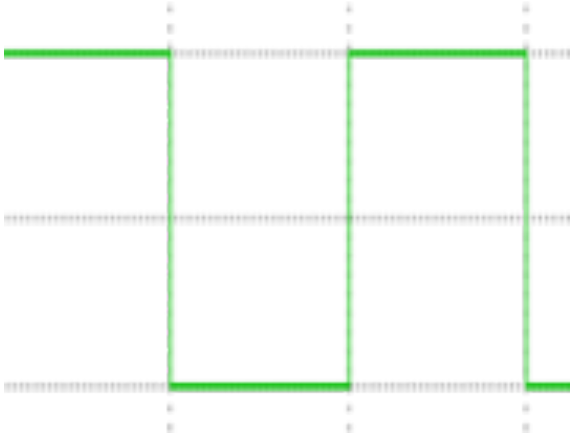
$=$

$$A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi kx)$$

infinite sum of sine waves

How would could you visualize this in the frequency domain?

Examples

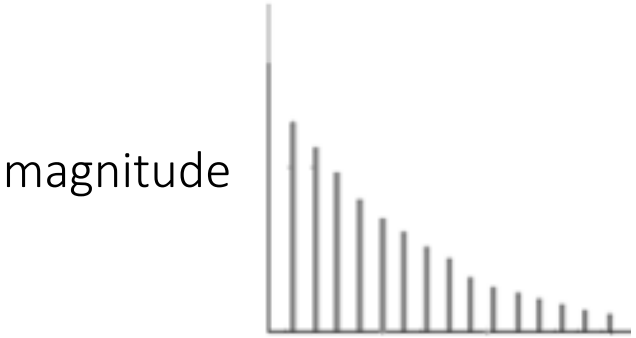


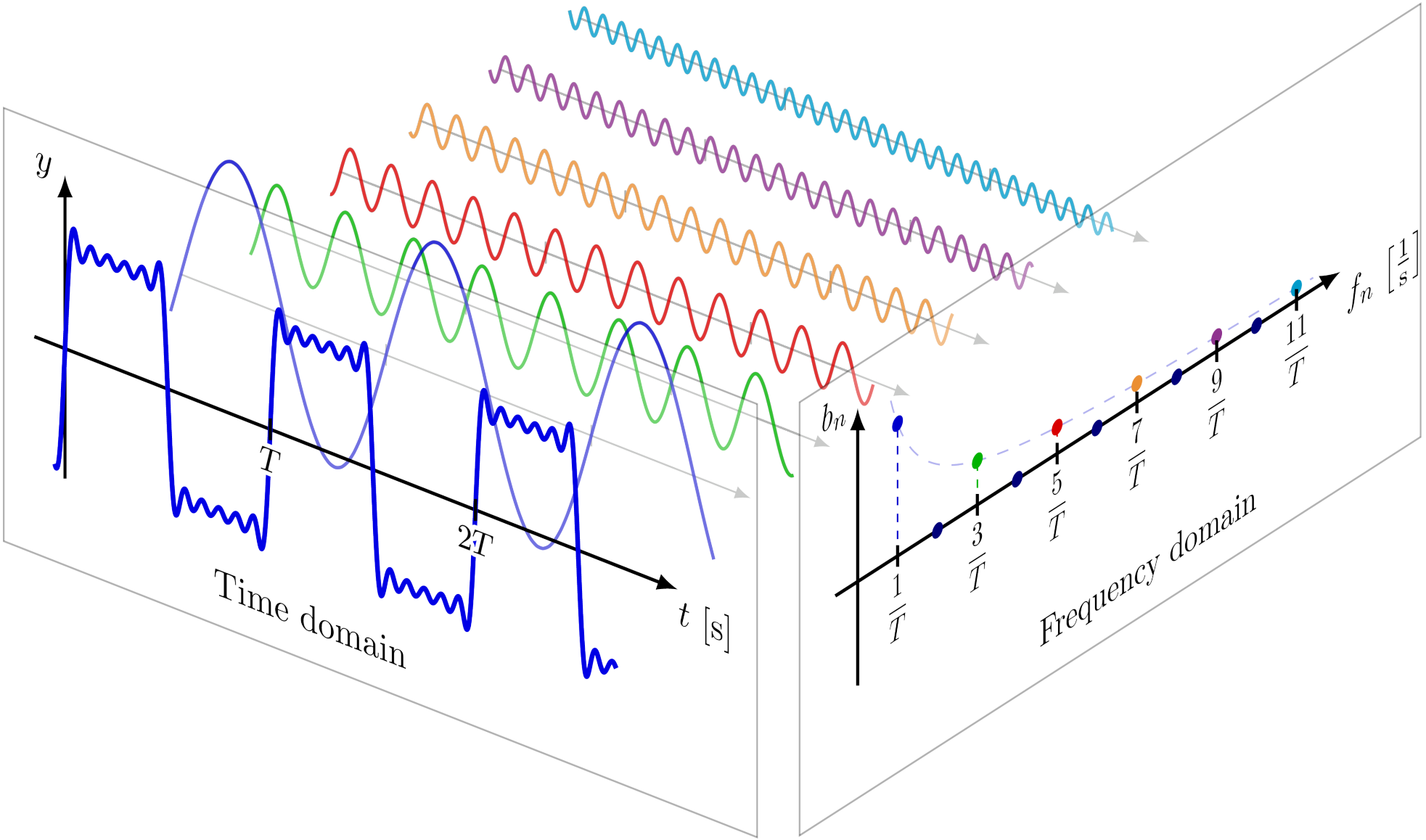
square wave

=

$$A \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi k x)$$

infinite sum of sine waves




y
T
2T
Time domain
t [s]
bn
1/T
3/T
5/T
7/T
9/T
11/T
Frequency domain
fn [1/s]

Fourier transform

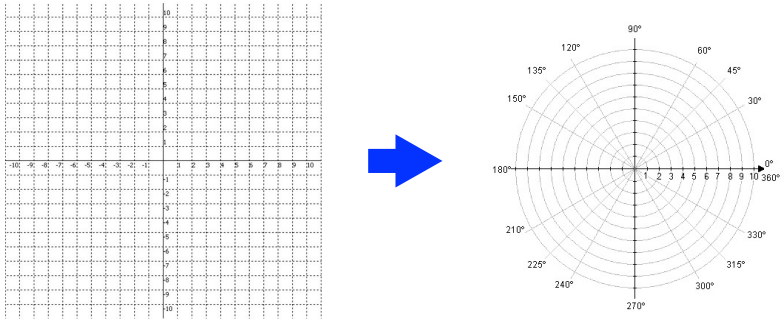
Recalling some basics

Complex numbers have two parts:

rectangular coordinates

$$R + jI$$

real imaginary



Alternative reparameterization:

polar coordinates

$$r(\cos\theta + j\sin\theta)$$

polar transform

$$\theta = \tan^{-1}(\frac{I}{R}) \quad r = \sqrt{R^2 + I^2}$$

or equivalently

$$re^{j\theta}$$

Euler's formula

$$e^{j\theta} = \cos\theta + j\sin\theta$$

exponential form

This will help us understand the Fourier transform equations

Fourier transform

The connection to the 'summation of sine waves' idea

$$F(k) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi kx/N}$$

$$f(x) = \sum_{k=0}^{N-1} F(k) e^{j2\pi kx/N}$$

Euler's formula

$$e^{j\theta} = \cos\theta + j\sin\theta$$

sum over frequencies

$$f(x) = \sum_{k=0}^{N-1} F(k) \left\{ \cos\left(2\pi kx/N\right) + j\sin\left(2\pi kx/N\right) \right\}$$

scaling parameter

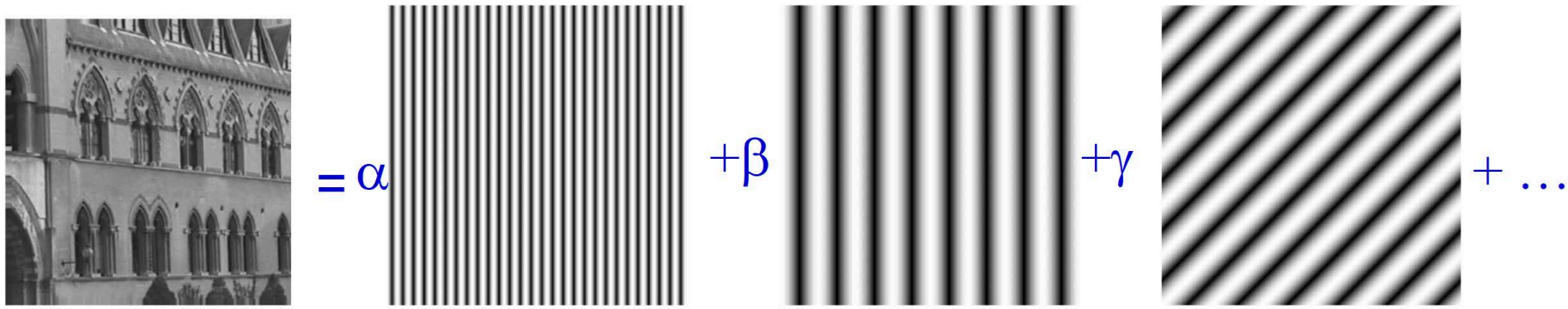
wave components

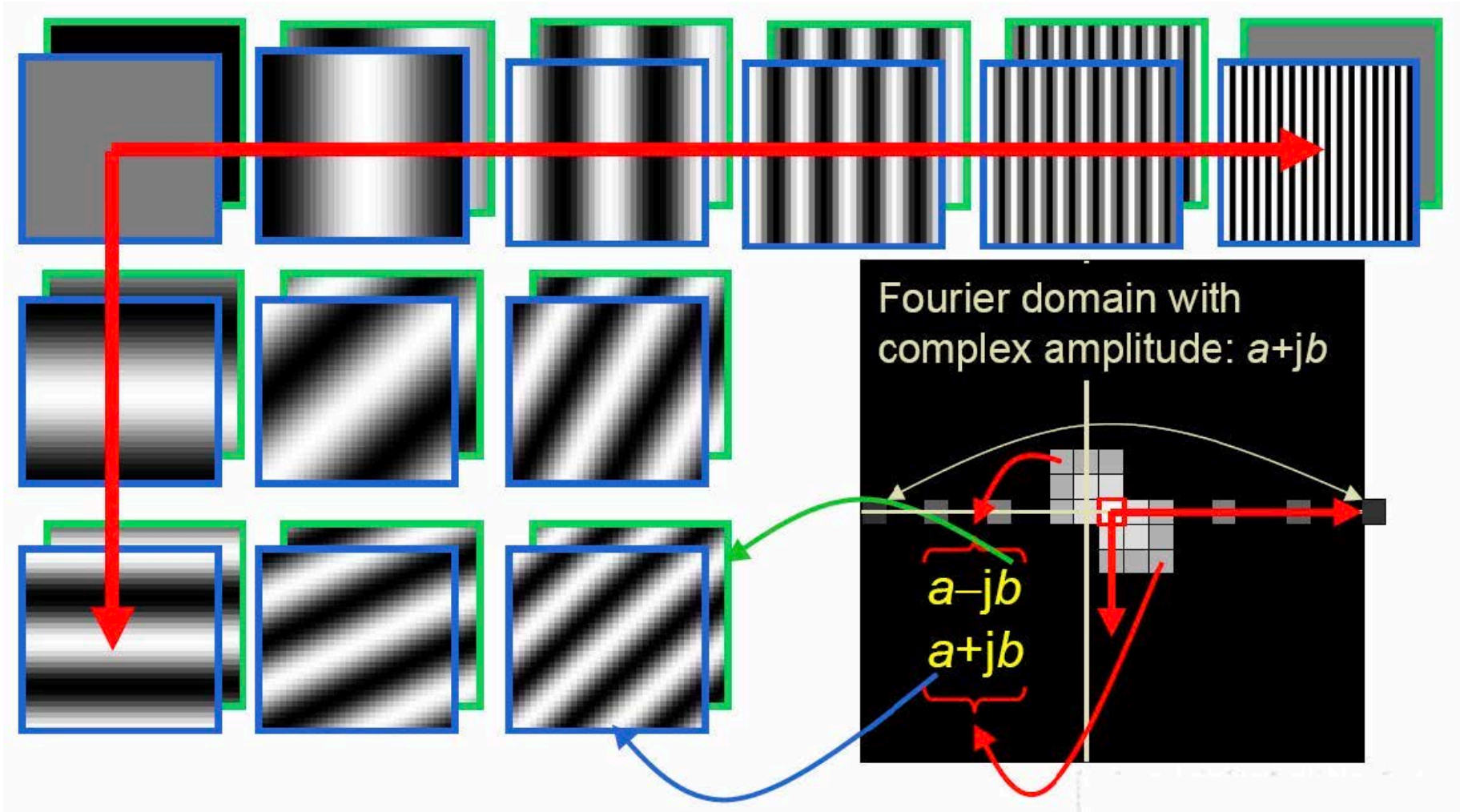
2D Fourier transform

$$F[k,l] = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m,n] e^{-j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

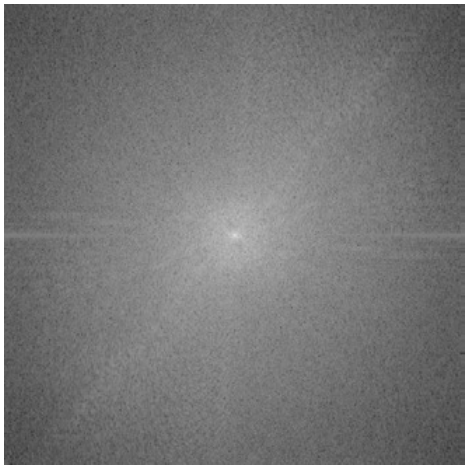
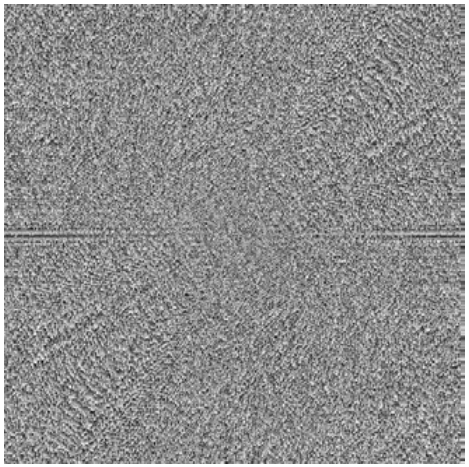
Inverse 2D DFT

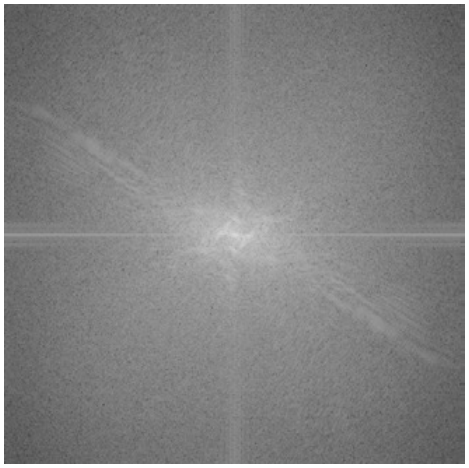
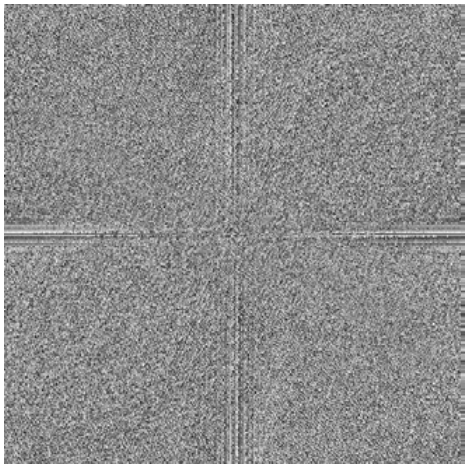
$$f[m,n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F[k,l] e^{j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$




Fourier domain with
complex amplitude: a+jb
a–jb
a+jb

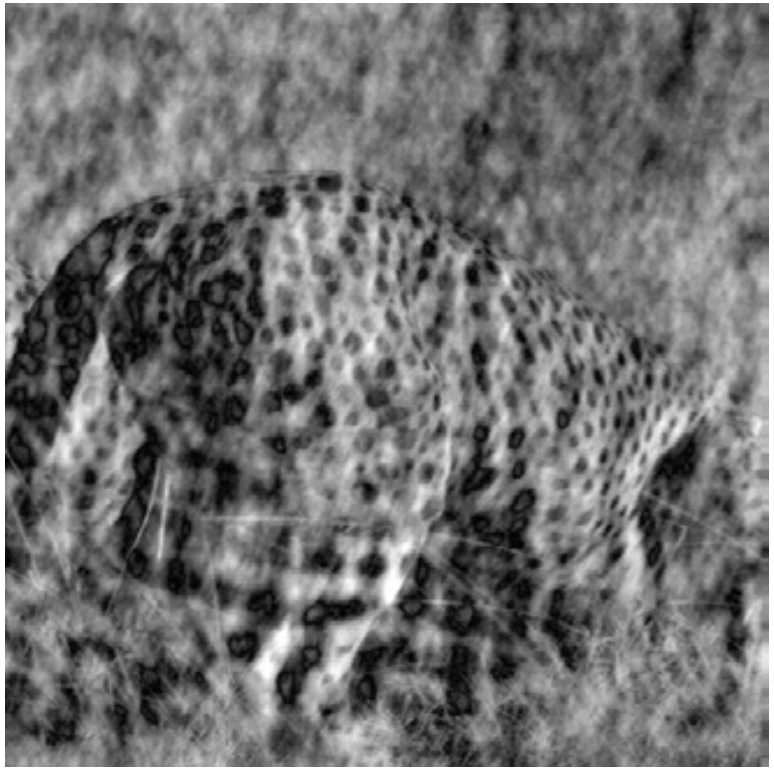
Fourier transforms of natural images

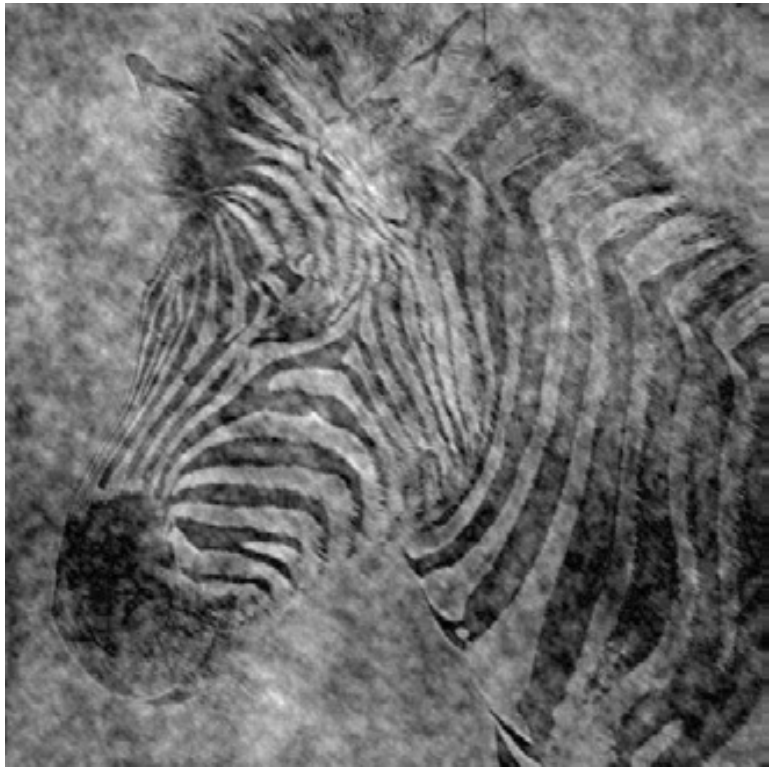



original amplitude phase

Fourier transforms of natural images



cheetah phase with zebra amplitude



zebra phase with cheetah amplitude

The convolution theorem

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

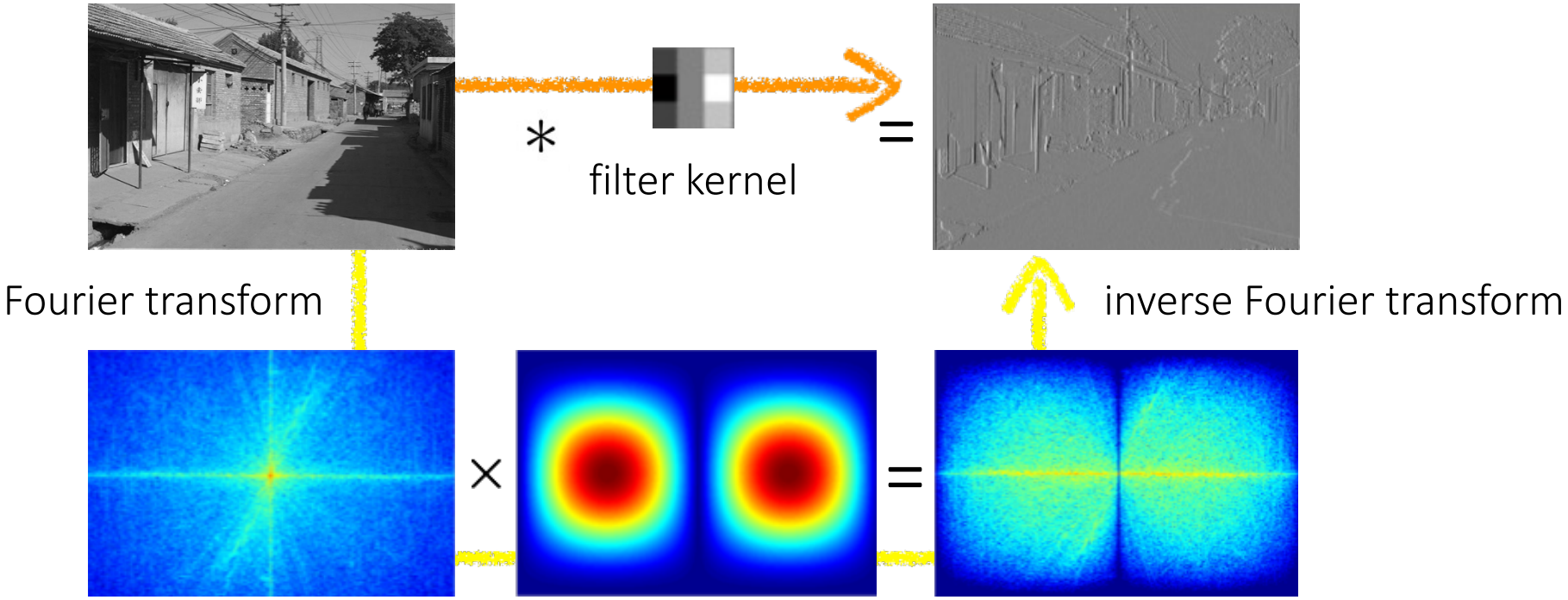
$$\mathcal{F}\{g * h\} = \mathcal{F}\{g\}\mathcal{F}\{h\}$$

The inverse Fourier transform of the product of two Fourier transforms is the convolution of the two inverse Fourier transforms:

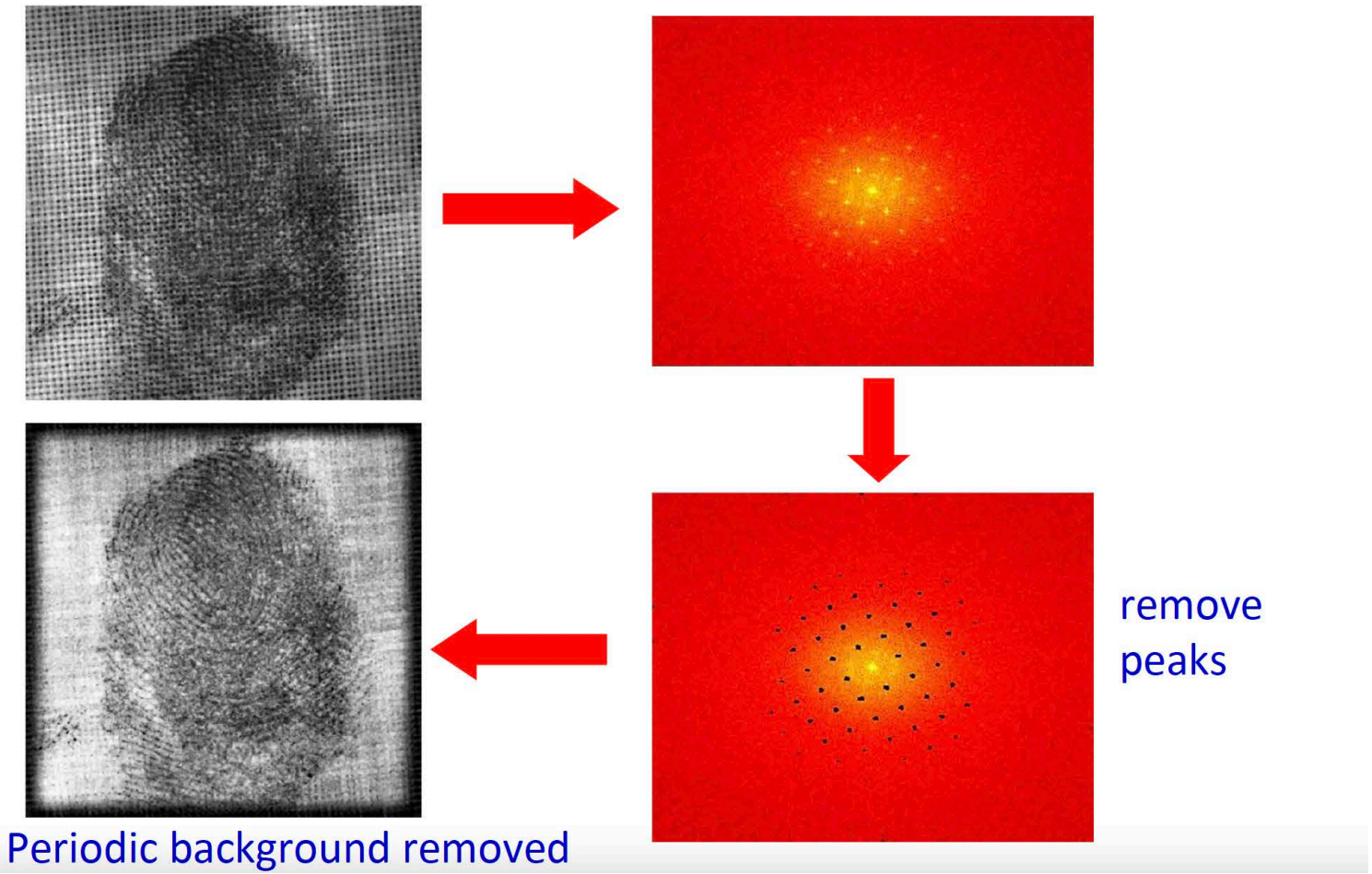
$$\mathcal{F}^{-1}\{gh\} = \mathcal{F}^{-1}\{g\} * \mathcal{F}^{-1}\{h\}$$

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

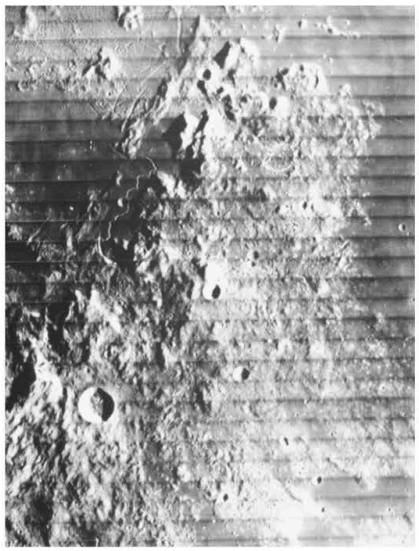
Spatial domain filtering

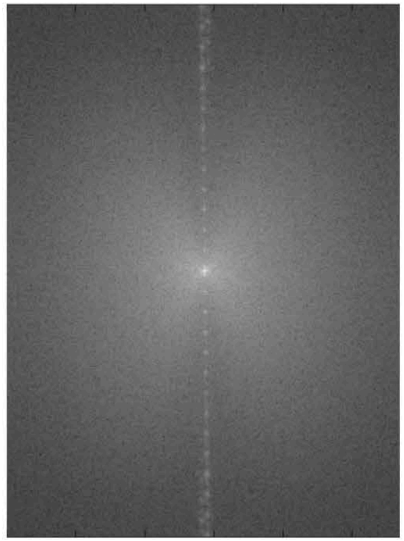


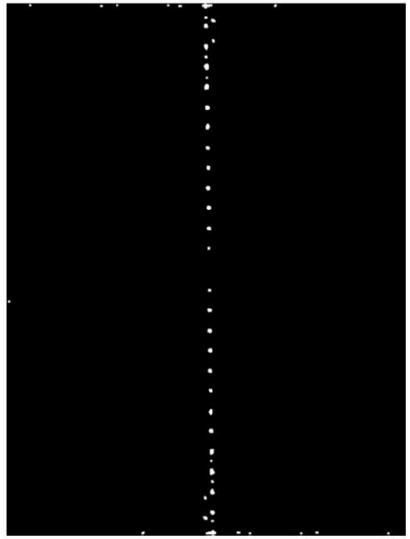
Frequency domain filtering


remove
peaks
Periodic background removed

Lunar orbital image (1966)







remove peaks

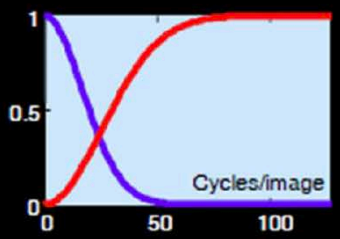
join lines removed

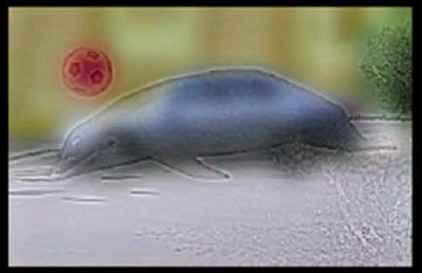
Perception of hybrid images

SIGGRAPH2006

1- Frequency cut = 16 cycles/image

Frequency domain

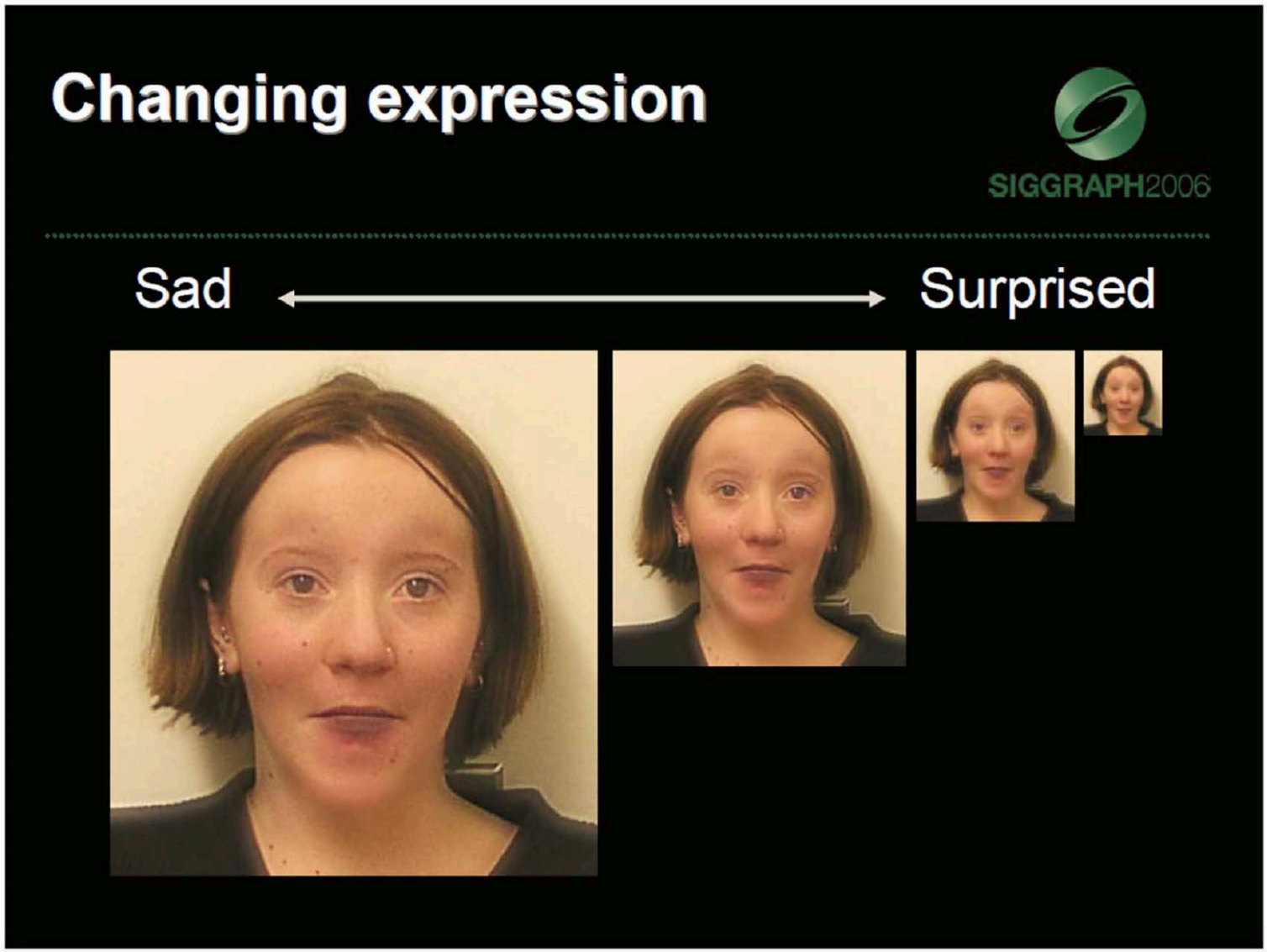




2- Frequency cut = 36 cycles/image


© 2006 Antonio Torralba and Aude Oliva

Changing expression



Aude Oliva & Antonio Torralba & Philippe G Schyns, SIGGRAPH 2006