Introduction to Medical Imaging

Signal Processing Basics

Klaus Mueller

Computer Science Department

Stony Brook University

Strange Effects

Ever tried to reduce the size of an image and you got this?

We call this effect ‘aliasing’

Better

But what you really wanted is this:

We call this ‘anti-aliasing’

Why Is This Happening?

The smaller image resolution cannot represent the image detail captured at the higher resolution

- skipping this small detail leads to these undesired artifacts

Overview

So how do we get the nice image?

For this you need to understand:

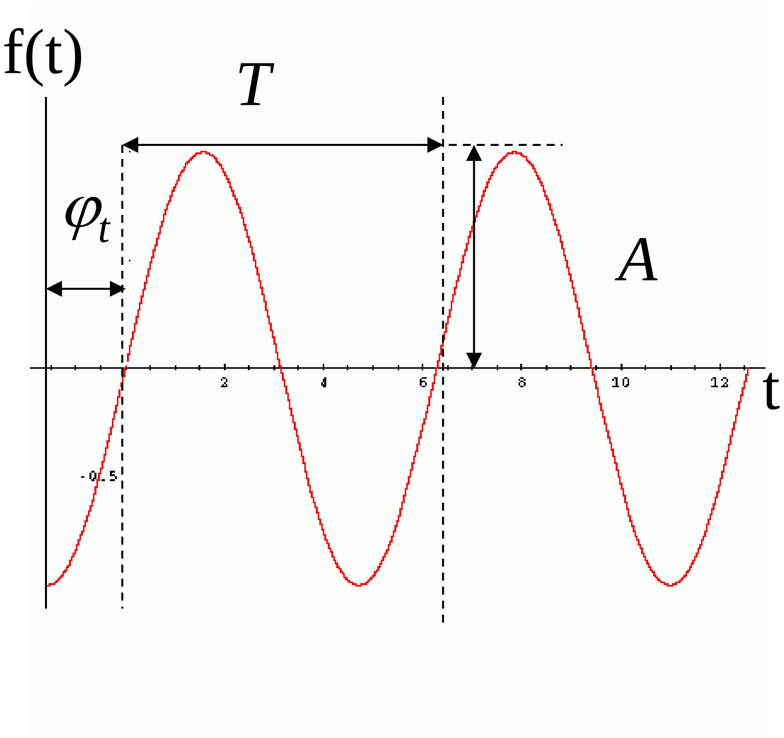
- Fourier theory
- Sampling theory
- Digital filters

Don’t be scared, we’ll cover these topics gently

Periodic Signals

A signal is periodic if $s(t+T) = s(t)$

- we call T the period of the signal
- if there is no such T then the signal is aperiodic



Sinusoids are periodic functions

- sinusoids play an important role

Write as:

$$A\sin(\frac{2\pi t}{T} + \varphi_t)$$

- where φ_t is the phase shift and A is the amplitude

Alternatively:

$$A\sin(2\pi ft + \varphi_t) = A\sin(\omega t + \varphi_t)$$

- where $f=1/T$ is the *frequency*
- we may write $\omega = 2\pi f$

Fourier Theory

Jean Baptiste Joseph Fourier (1768-1830)

His idea (1807):

- *Any periodic function can be rewritten as a weighted sum of sines and cosines of different frequencies.*

Don’t believe it?

- neither did Lagrange, Laplace, Poisson and other major mathematicians of his time
- in fact, the theory was not translated into English until 1878

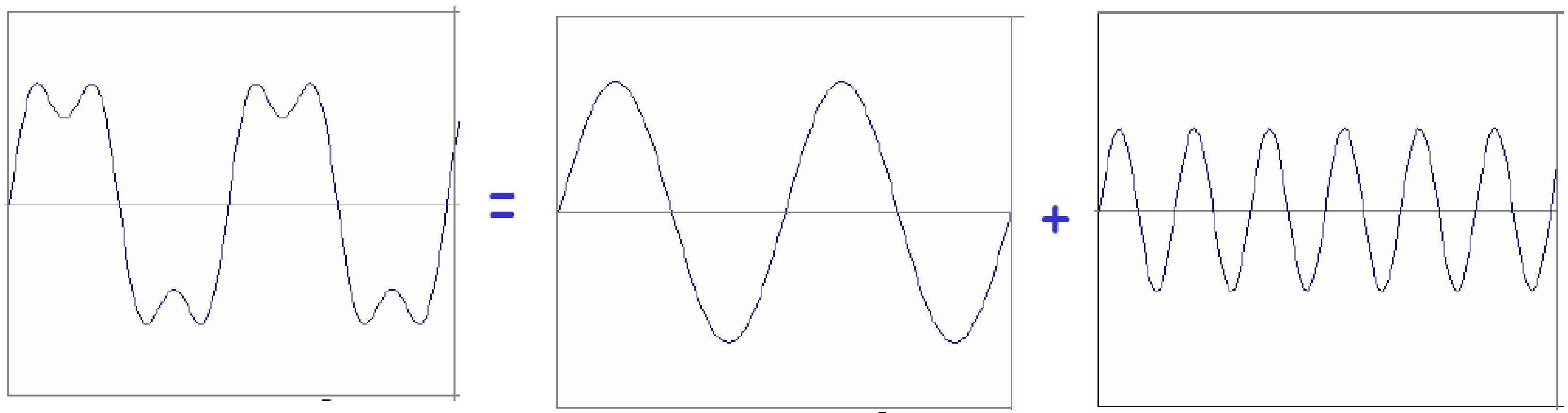
But it’s true!

- it is called the *Fourier Series*

Example

Consider the function:

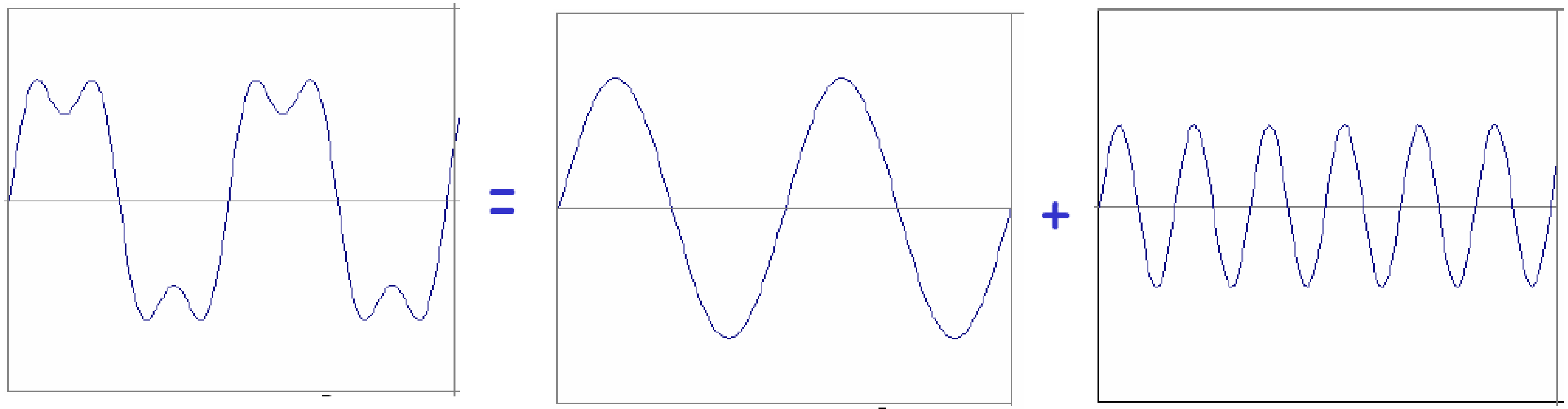
$$g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi(3f)\, t)$$

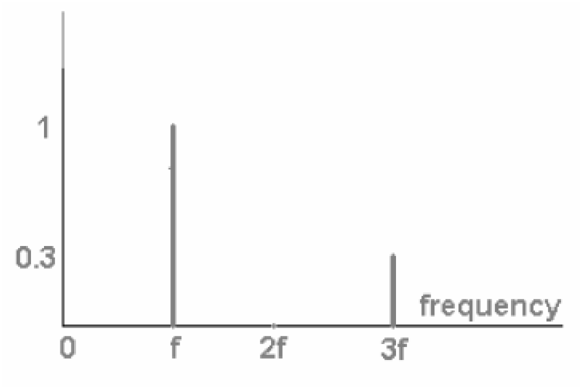


Frequency Spectrum

Consider the function:

$$g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi(3f)\, t)$$

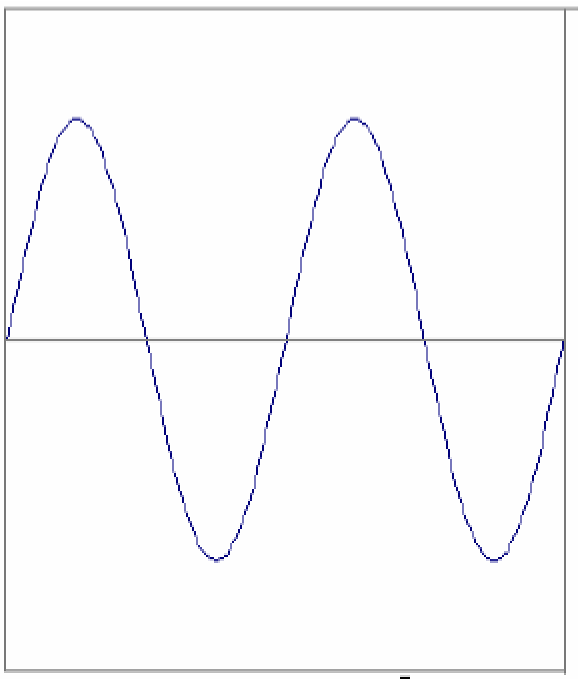




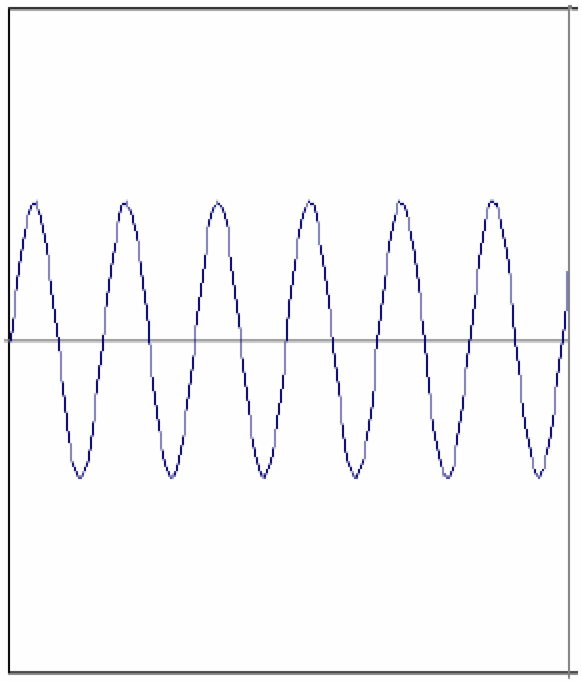
the function's frequency spectrum

A. Efros, CMU

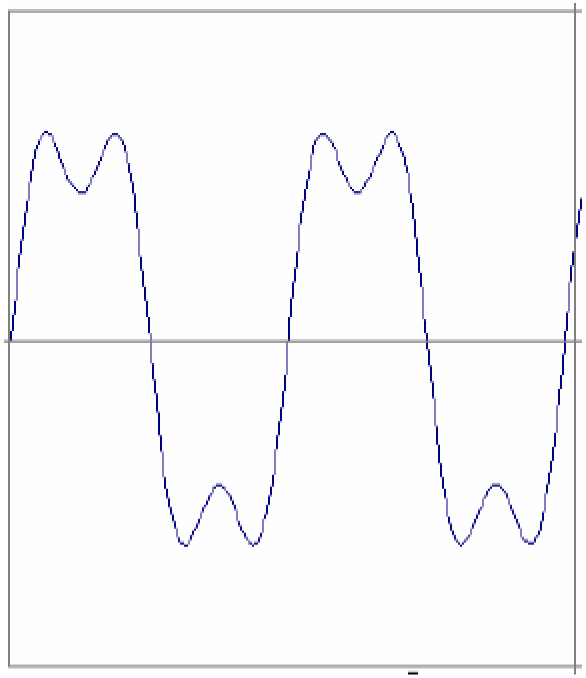
Further Example (1)



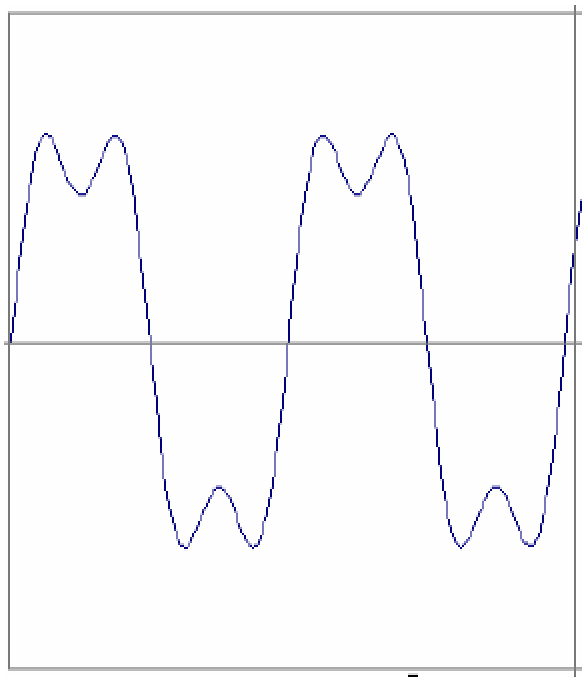
+



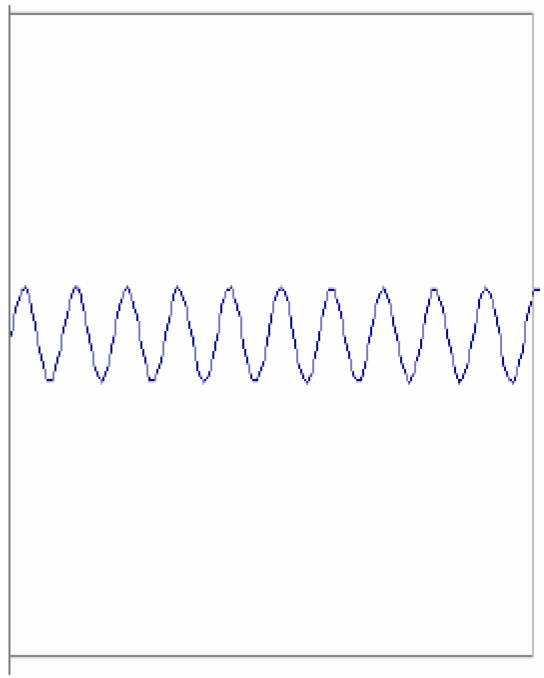
=



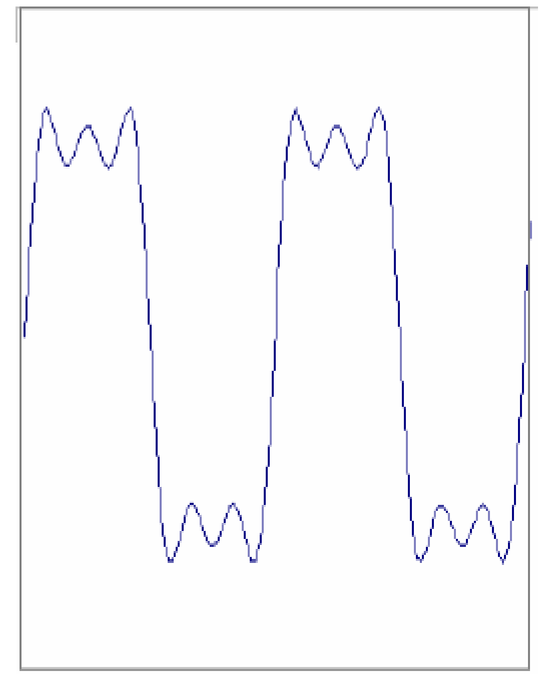
Further Example (2)



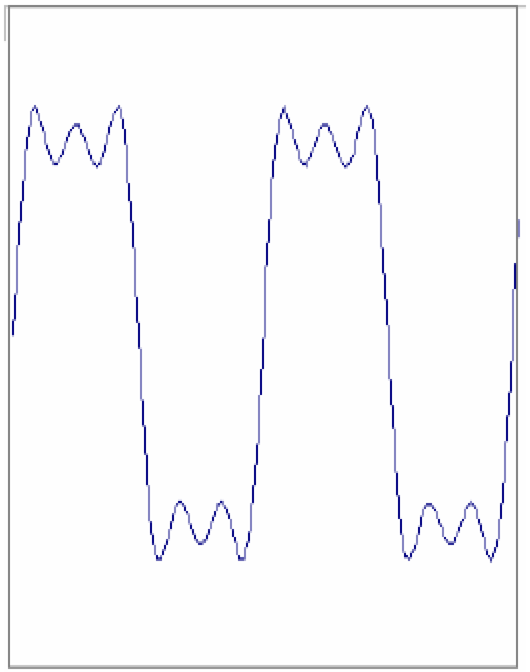
+



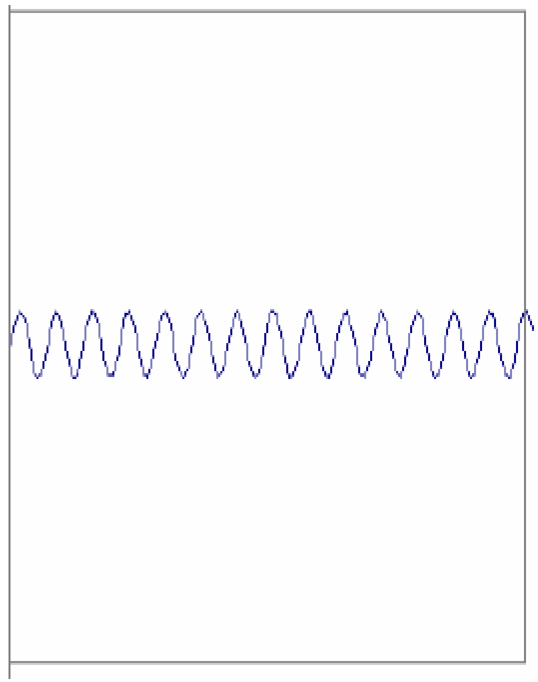
=



Further Example (3)



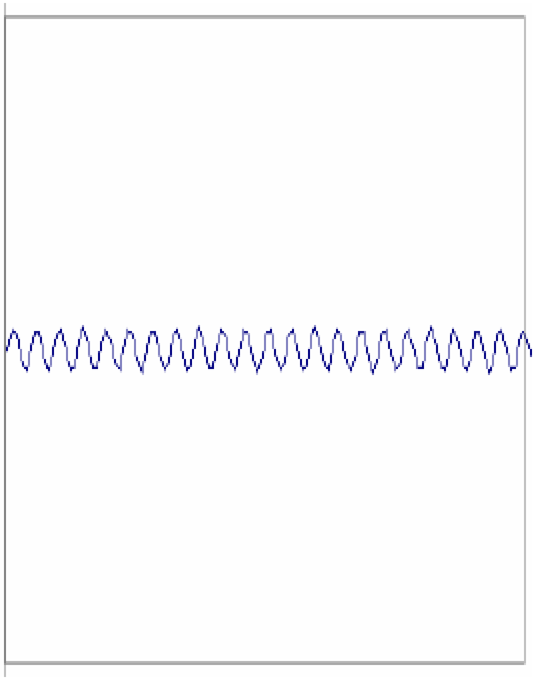
+



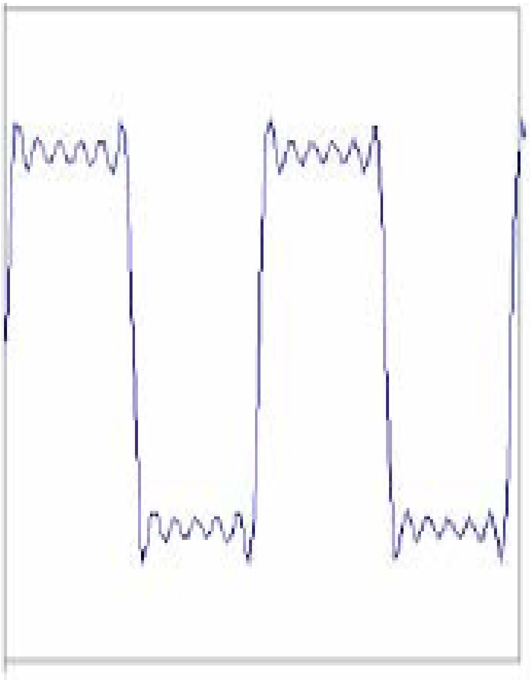
=

Further Example (4)

+



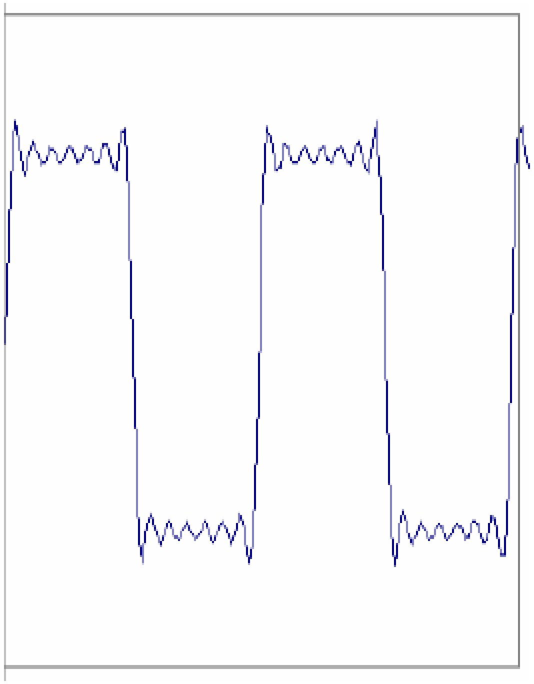
=



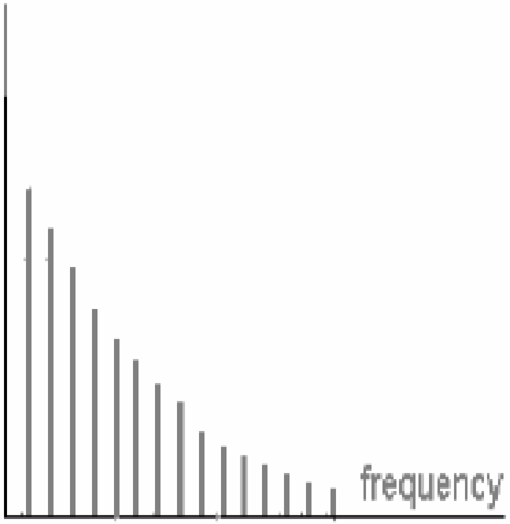
The Importance of the Frequency Spectrum

We observe:

- oscillations of different frequencies add to form the signal
- there is a characteristic frequency spectrum to any signal
- sharp edges can only be represented (generated) by high frequencies



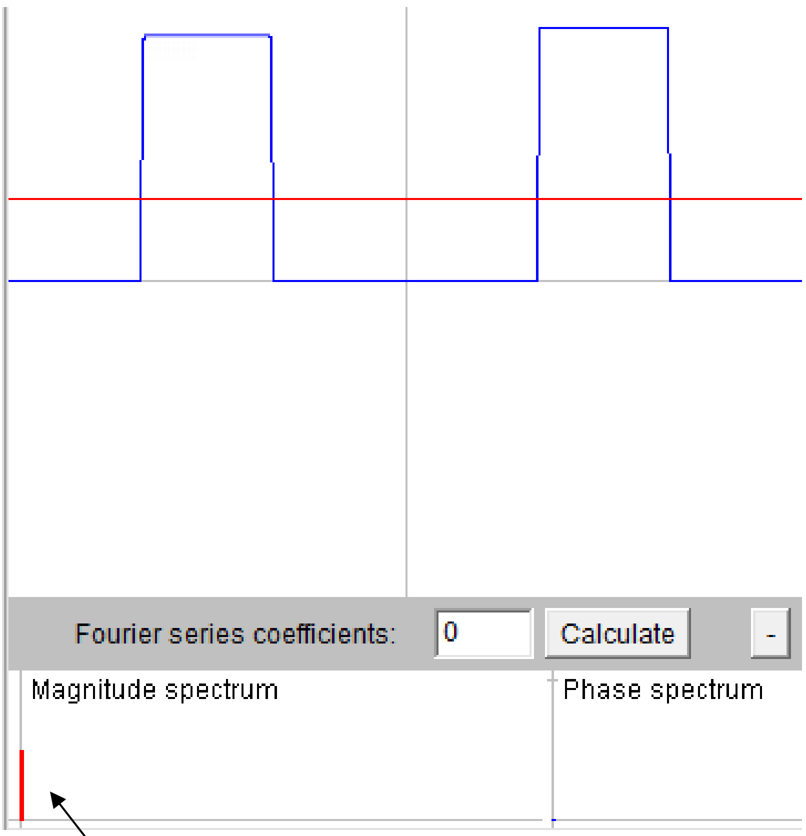
signal
(approximate square/box function)

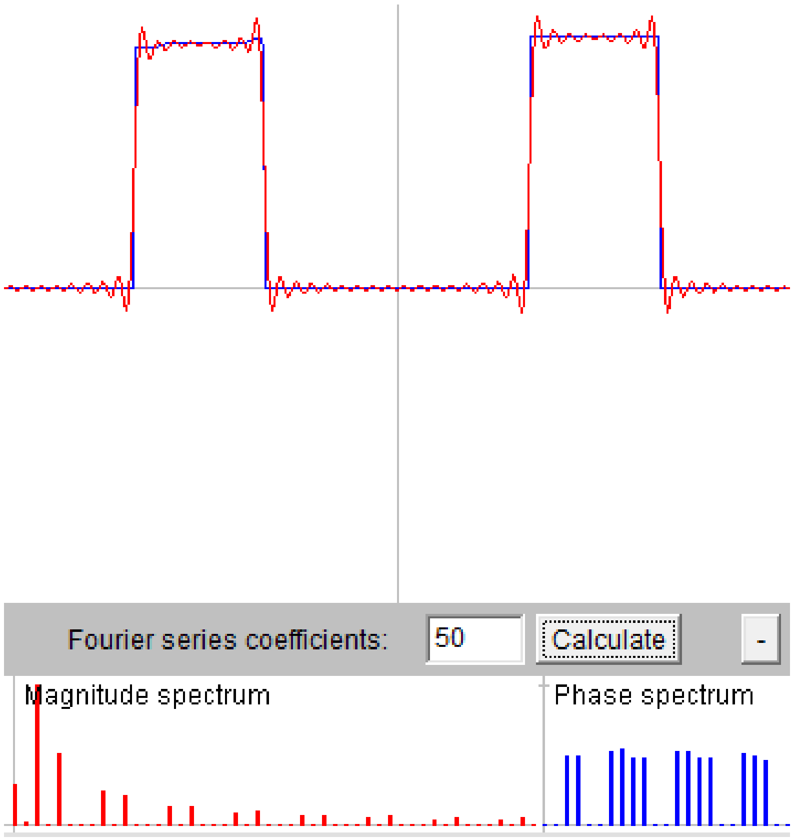


its frequency spectrum

The DC Component

The first component of the spectrum is the *signal average DC*





‘DC component’ = signal average

The Math…

The example just seen has the following Fourier Series:

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

- most of the time the phase is not interesting, so we shall omit it

In fact, this is an interesting series: the *sinc* function

- we shall see more of it soon

We can convert any discrete signal into its Fourier Series (and back)

- this is called the *Fourier Transform* (*Inverse Fourier Transform*)

$$\begin{array}{ccc} & \boxed{\text{Fourier Transform}} & \\ s(t) & \rightleftarrows & S(k) \\ & \boxed{\text{Inverse Fourier Transform}} & \end{array}$$

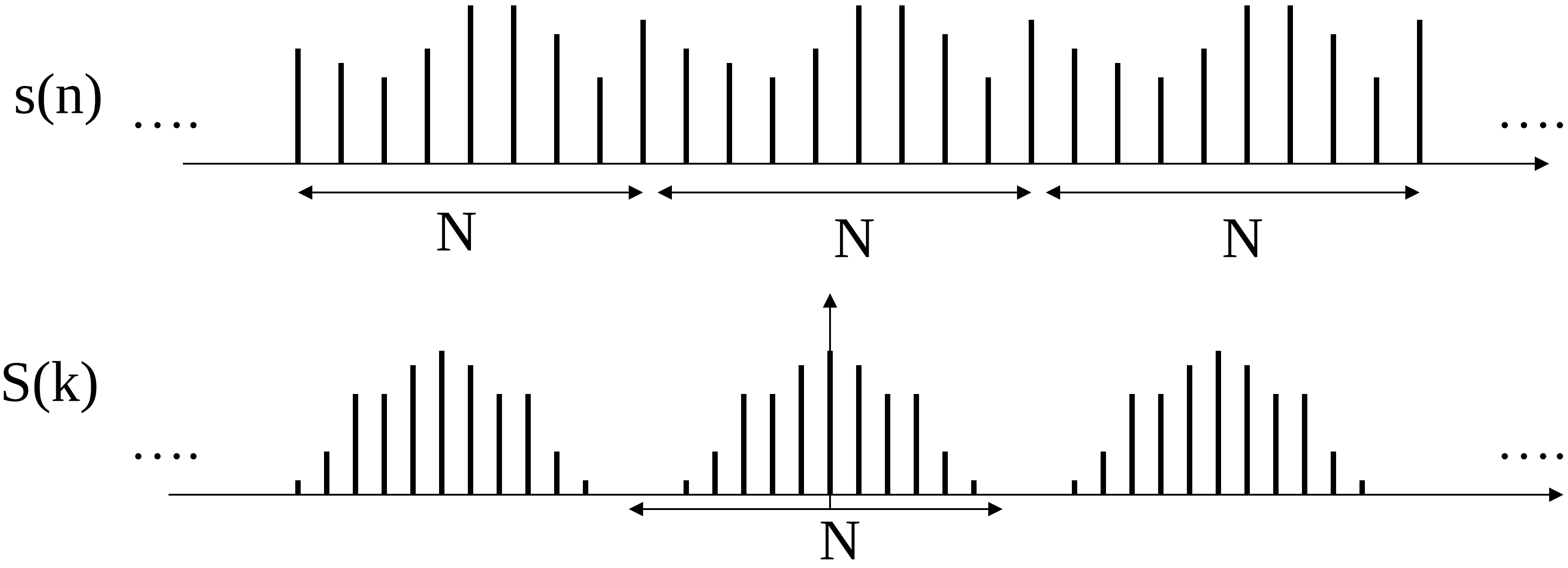
Fourier Transform of Discrete Signals: DFT

Discrete Fourier Transform (DFT)

- assumes that the signal is discrete and finite

$$S(k)=\sum_{n=0}^{N-1} s(n)e^{\frac{-i2\pi kn}{N}} \qquad s(n)=\frac{1}{N}\sum_{n=0}^{N-1} S(k)e^{\frac{i2\pi kn}{N}}$$

- we have *N* samples, from which we can calculate *N* frequencies
- the frequency spectrum is <u>discrete</u> and it is <u>periodic</u> in *N*



Periodicity

Images are discrete signals

- so their frequency spectra are finite and periodic (see last slide)
- and therefore they have an upper limit (a maximum frequency)

Images are also finite (in size)

- the DFT assumes that they are also periodic
- as odd as this may sound, this is the underlying assumption

Therefore:

- frequency spectra are finite and periodic
- images are also finite and periodic

Keep this in mind for now

- it will help explain the strange resizing effects presented before

Now, What About the Complex Exponential…

It is Fourier's way to encode phase and amplitude into one representation

- to understand it better, let's first review complex numbers
- and then see what it means in the Fourier context

Note, we only discuss this to illustrate the full picture

- essential for this class is only to know the concept of frequency spectrum discussed thus far

Recall: Complex Numbers

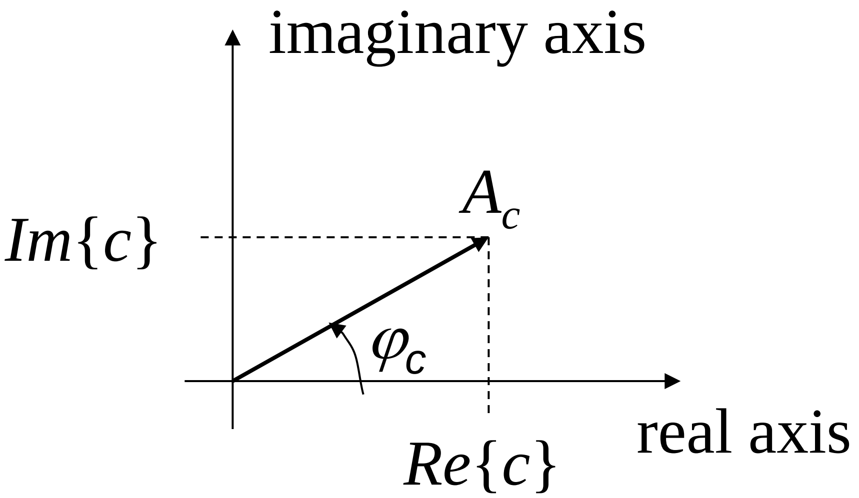
A complex number *c* has a real and an imaginary part:

- $c = Re\{c\} + i\,Im\{c\}$ (cartesian representation) $i = \sqrt{-1}$
- here, *i* always denotes the complex part

We can also use a polar representation:

$$A_c = \sqrt{\mathrm{Re}\{c\}^2 + \mathrm{Im}\{c\}^2}$$

$$\varphi_c = \tan^{-1}\left(\frac{\mathrm{Im}\{c\}}{\mathrm{Re}\{c\}}\right)$$

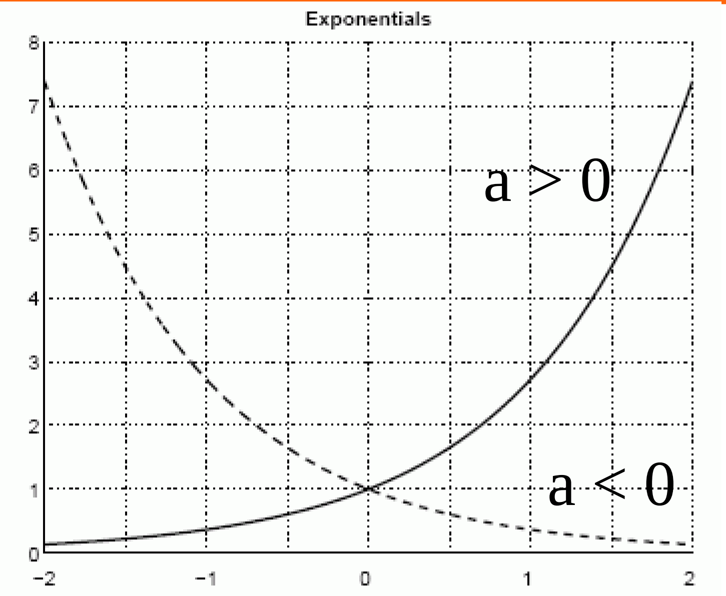


Application: Complex Sinusoids

Exponential *exp*

$$\exp(ax) = e^{ax}$$

- when *a* > 0 then *exp* increases with increasing *x*
- when *a* < 0 then *exp* approximates 0 with increasing *x*

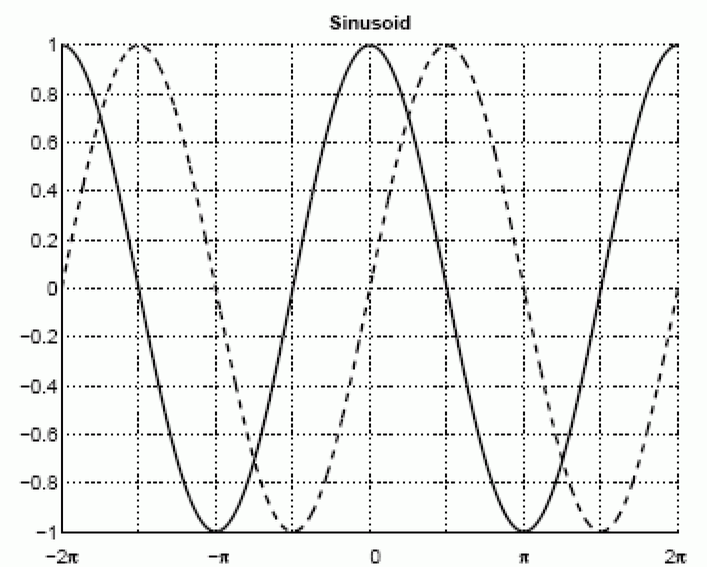


Complex exponential / sinusoid:

$$A_k e^{i(2\pi kt+\varphi)} = A_k(\cos(2\pi kt+\varphi)+i\sin(2\pi kt+\varphi))$$

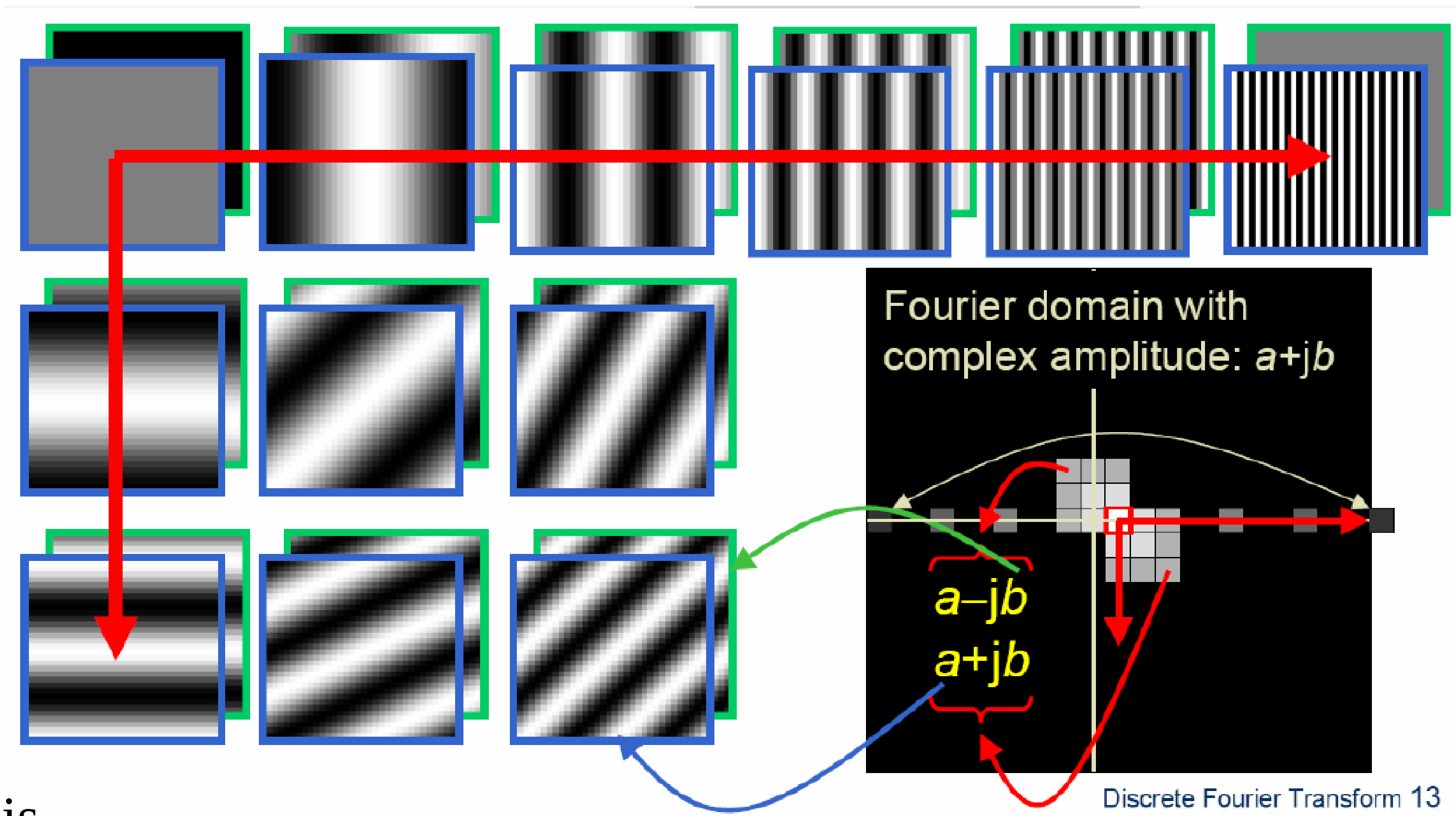
As before

- the *cos* term is the signal's real part
- the *sin* term is the signal's imaginary part
- *A* is the amplitude, φ the phase shift, *k* determines the frequency



Two-Dimensional Fourier Spectrum

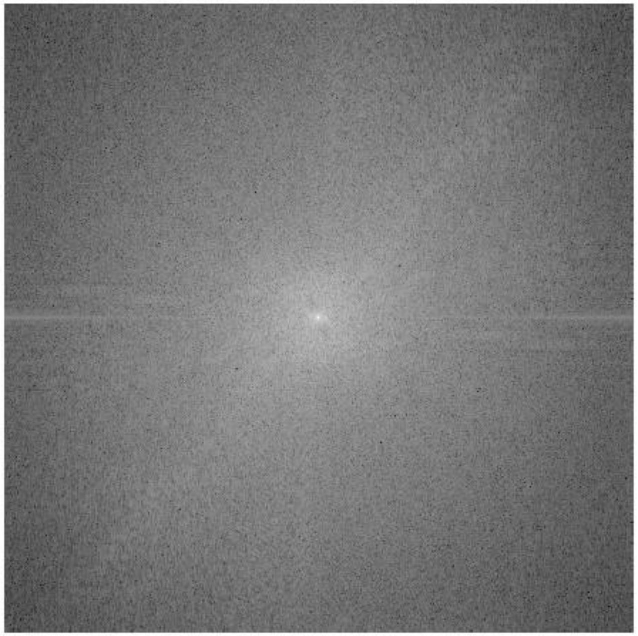
u-axis

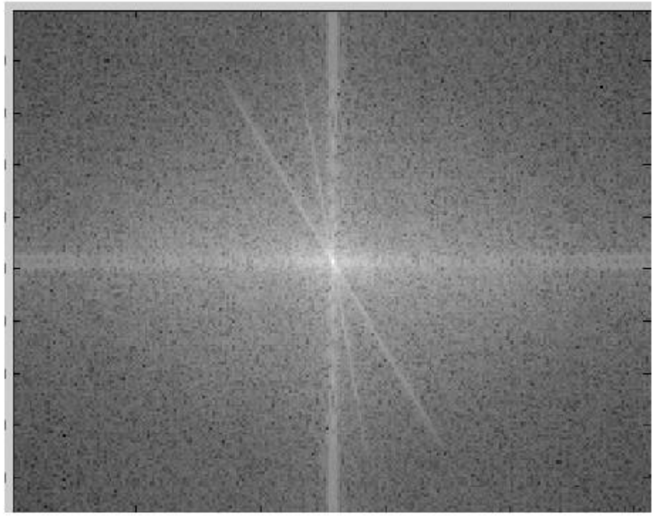


v-axis

A. Efros, CMU

Some Example Spectra





A. Efros, CMU

Effects of Missing Spectra Portions: Axial

(a) Spectrum along u determines detail along spatial x

(b) Spectrum along v determines detail along spatial y

(a)

(b)

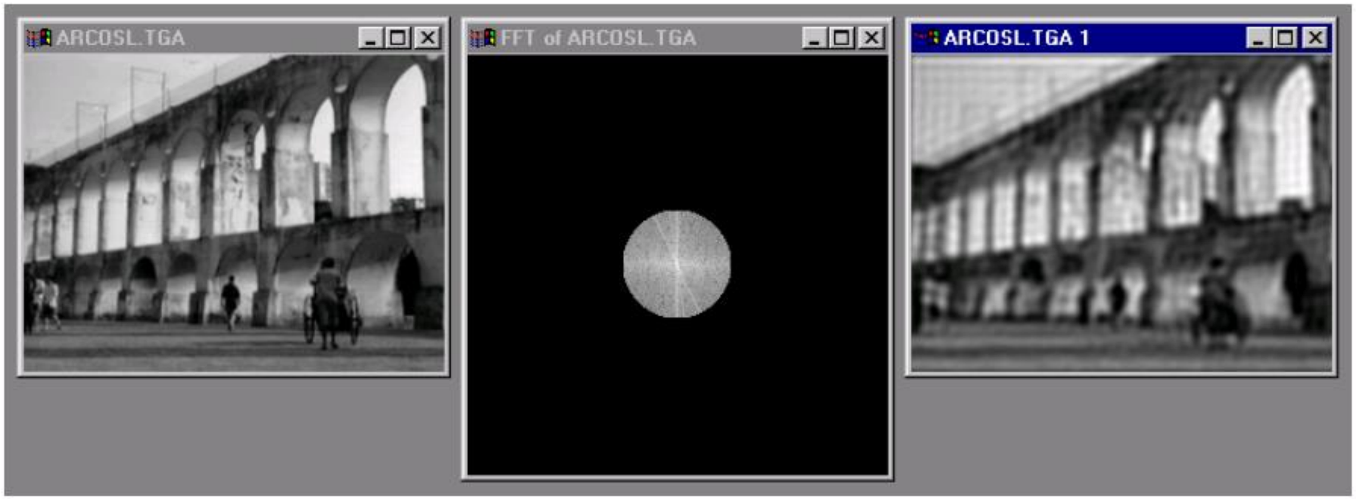
A. Efros, CMU

Effects of Missing Spectra Portions: Radial

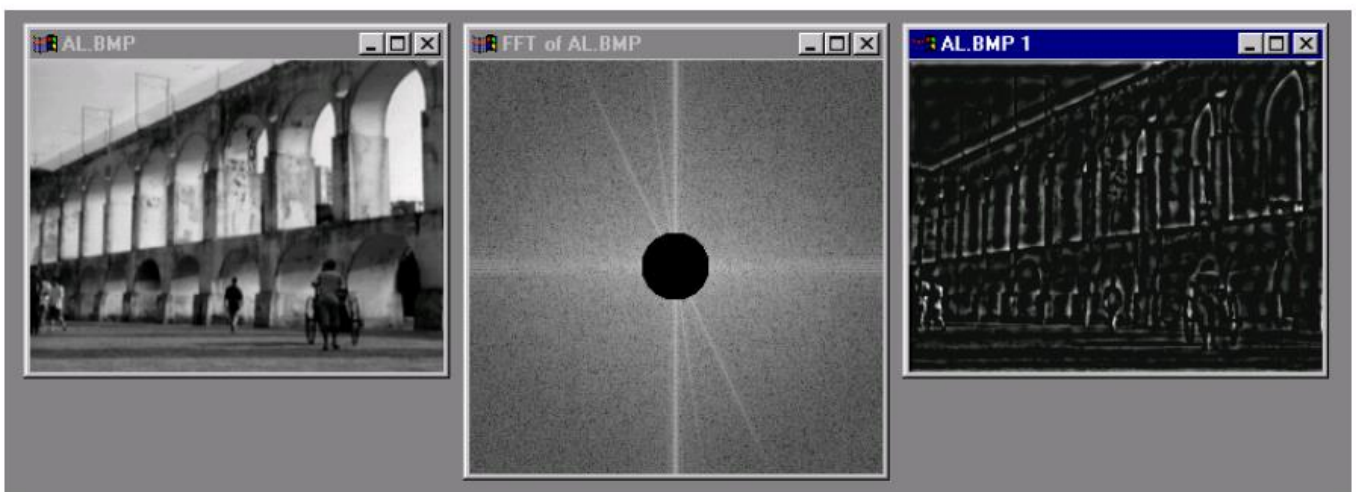
(a) Lower frequencies (close to origin) give overall structure

(b) Higher frequencies (periphery) give detail (sharp edges)

(a)



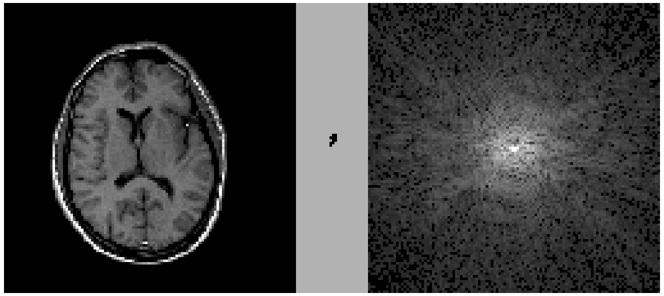
(b)



A. Efros, CMU

The Math… 2D DFT

The 2D transform:



$$S(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} s(n,m) e^{\frac{-i2\pi(kn+lm)}{NM}}$$

$$s(n,m) = \frac{1}{NM}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} S(k,l) e^{\frac{i2\pi(kn+lm)}{NM}}$$

Separability:

$$S(k,l) = \frac{1}{NM}\sum_{m=0}^{M-1} e^{\frac{-i2\pi lm}{M}} P(k,m) \quad \text{where } P(k,m) = \sum_{n=0}^{N-1} s(n,m) e^{\frac{-i2\pi kn}{N}}$$

$$s(n,m) = \frac{1}{NM}\sum_{l=0}^{M-1} e^{\frac{-i2\pi lm}{M}} p(n,l) \quad \text{where } \mathrm{p}(n,l) = \sum_{k=0}^{N-1} S(n,m) e^{\frac{-i2\pi kn}{N}}$$

- if M=N, complexity is $2\cdot O(2N^3)$

Fast Fourier Transform (FFT)

Recursively breaks up the FT sum into odd and even terms:

$$S(k) = \sum_{n=0}^{N-1} s(n) e^{\frac{-i2\pi kn}{N}} = \sum_{n=0}^{N/2-1} s(2n) e^{\frac{-i2\pi k2n}{N}} + \sum_{n=0}^{N/2-1} s(2n+1) e^{\frac{-i2\pi k(2n+1)}{N}}$$

$$= \sum_{n=0}^{N/2-1} s_{even}(n) e^{\frac{-i2\pi kn}{N/2}} + e^{\frac{-i2\pi k}{N}} \sum_{n=0}^{N/2-1} s_{odd}(n) e^{\frac{-i2\pi kn}{N/2}}$$

Results in an O(n·log(n)) algorithm (in 1D)

- O(n^2·log(n)) for 2D (and so on)

Fast Fourier Transform (FFT)

Gives rise to the well-known butterfly Divide + Conquer architecture

- invented by Cooley-Tuckey, 1965)