Fourier

Deepayan Bhowmik

Fourier: The Basics

Fourier transform of an image is a global transform.
Produces a representation of the image in Fourier space:

- i.e. the frequency domain.

Performed by Discrete Fourier Transform (DFT) operator:

- Input: spatial domain image (pixels).

Fourier transform:

- Decomposes image into sine and cosine frequency components.
- Point in the output image represents a particular frequency contained in the input. Spatial domain (real space) image.
- Used in image analysis, compression … .

Fourier: Terminology

Spatial (or Real) domain (or “real space”):

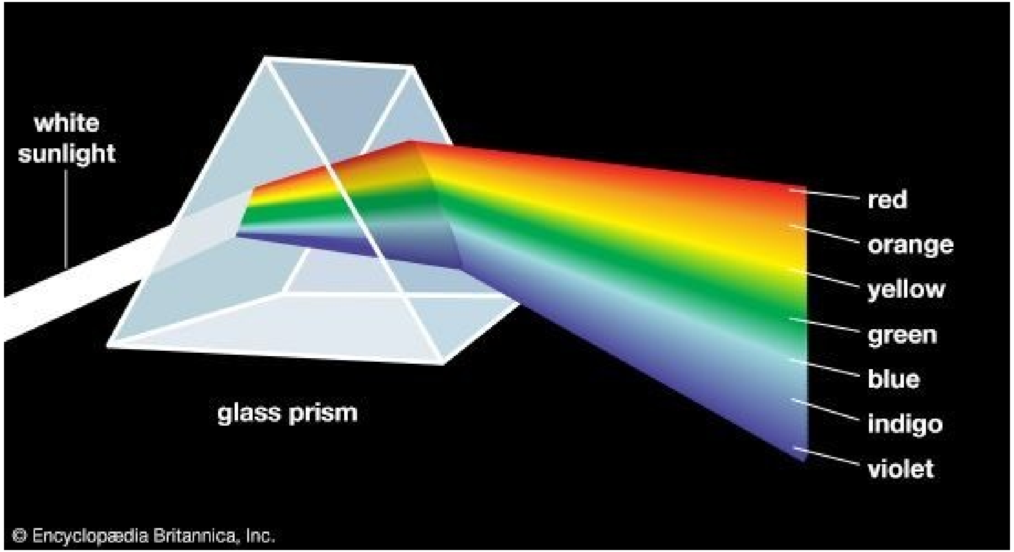
- Where images (and signals) are represented by a spatial layout of samples that are real numbers:
  - i.e. pixels for images (2D).
  - i.e. samples for signals (1D).

Frequency (or Fourier) domain (or “frequency/Fourier space”):

- Where images (and signals) are represented by (complex number) coefficients of Sine and cosine basis frequencies:
  - 2D spatial frequencies for images.
  - 1D waveform frequencies for signals.

Fourier: Terminology

“A useful **analogy to the Fourier Transform** is a glass prism.
A glass prism is a physical device that separates light into various colours on its wavelength (or frequency) content.
The Fourier transform may be viewed as a **'mathematical prism' that separates a function [or image] into components based on frequency content**.”



Fourier: Terminology

Had crazy 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 big wigs.
- Not translated into English until 1878!

But it's true!

- Called Fourier Series.
- Possibly the greatest tool used in Engineering.

Fourier: Example (1D)

Our building block:

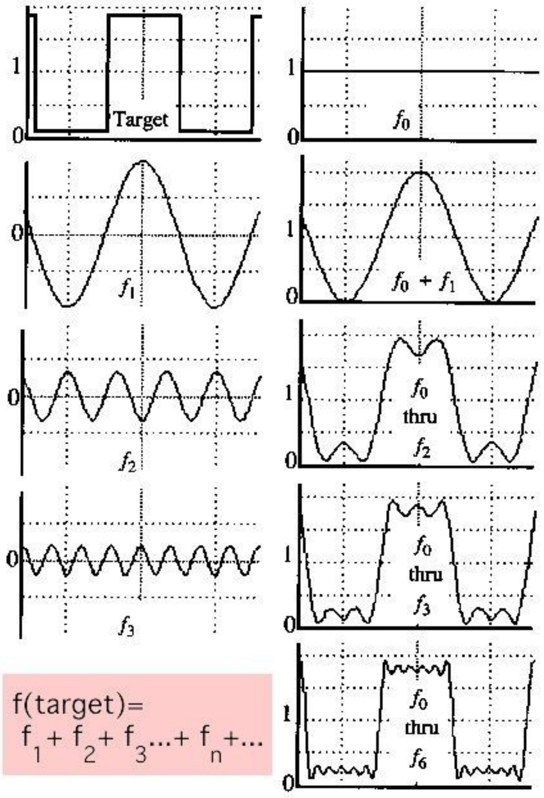
$$A \sin(\omega x + \varphi)$$

Add enough of them to get any signal *f(x)* you want:

f(x) = f(target)

= f1 + f2 + f3 + f4 + … + fn

Which one encodes the coarse vs. fine structure of the signal?



Fourier: Example (1D)

We want to understand the frequency of our 1D signal. So, let's reparametrize the signal by instead of *x*:

$$f(x) \longrightarrow \boxed{\text{Fourier Transform}} \longrightarrow F(\omega)$$

For every ω from *0* to *inf*, *F()* holds the amplitude *A* and phase ϕ of the corresponding sine:

$$A \sin(\omega x + \varphi)$$

How can *F* hold both? Complex number trick!

$$F(\omega) = R(\omega) + iI(\omega)$$

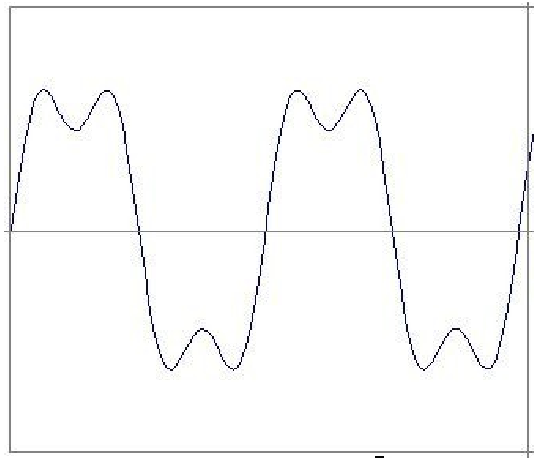
$$A = \pm\sqrt{R(\omega)^2 + I(\omega)^2}$$

$$\phi = \tan^{-1}\frac{I(\omega)}{R(\omega)}$$

$$F(\omega) \longrightarrow \boxed{\text{Inverse Fourier Transform}} \longrightarrow f(x)$$

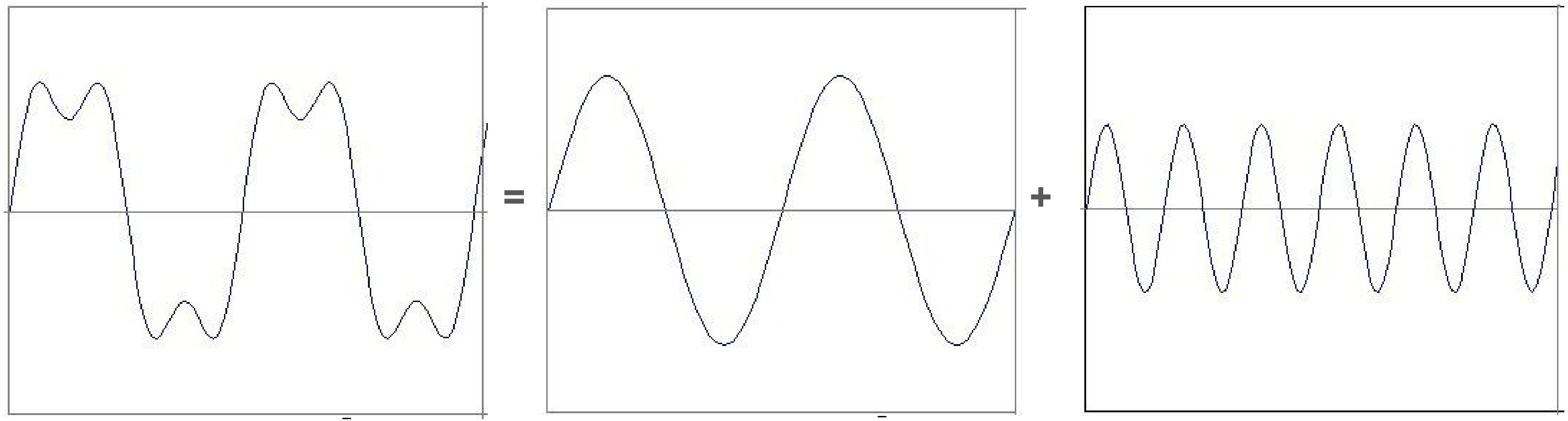
Fourier: Example (1D)

Example 1D signal: *f(x) = sin(2pf x) + (1/3)sin(2p(3f) x)*



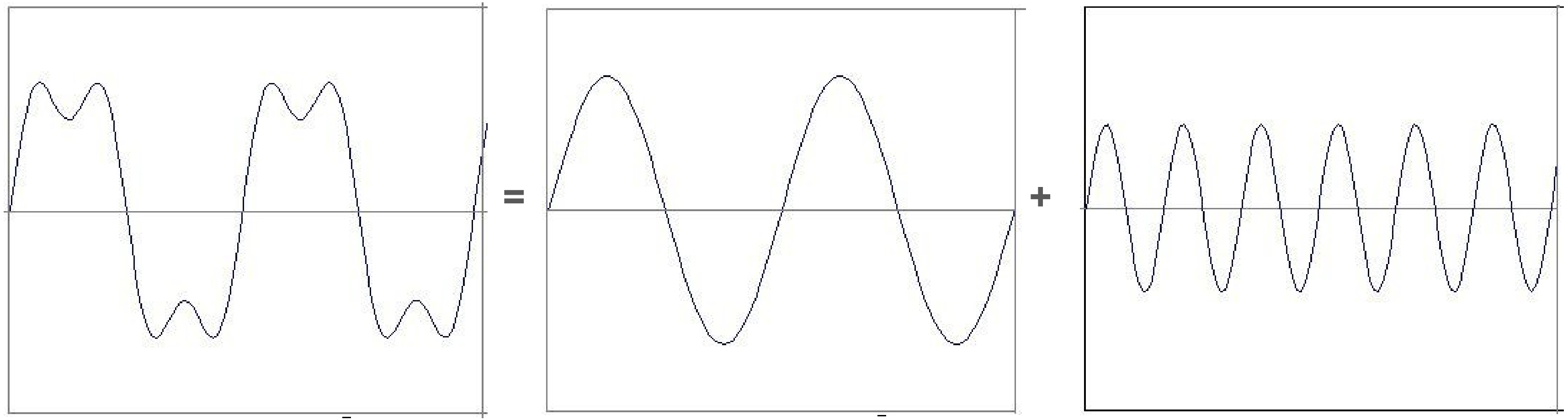
Fourier: Example (1D)

Example 1D signal: $f(x) = sin(2pf\ x) + (1/3)sin(2p(3f)\ x)$

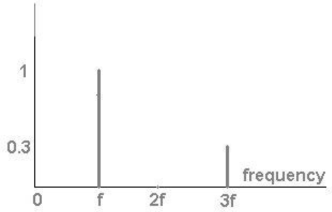


Fourier: Example (1D)

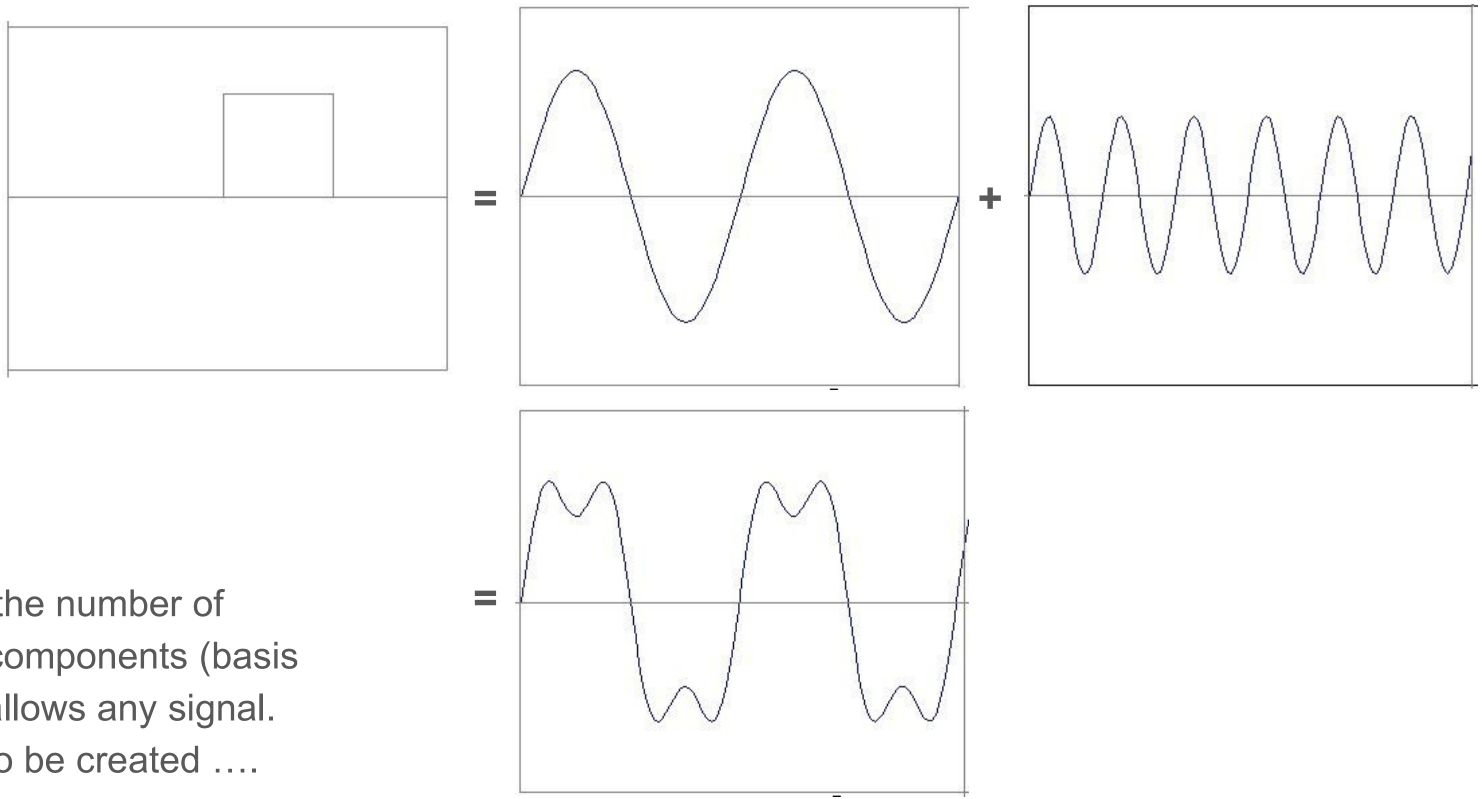
Example 1D signal: *f(x) = sin(2pf x) + (1/3)sin(2p(3f) x)*



Gives us frequency coefficients {1, 1/3}
for sinusoidal basis functions sin(2p(cf) x)
where p, f are constants and c = {0, 1, 2, 3, …, n}

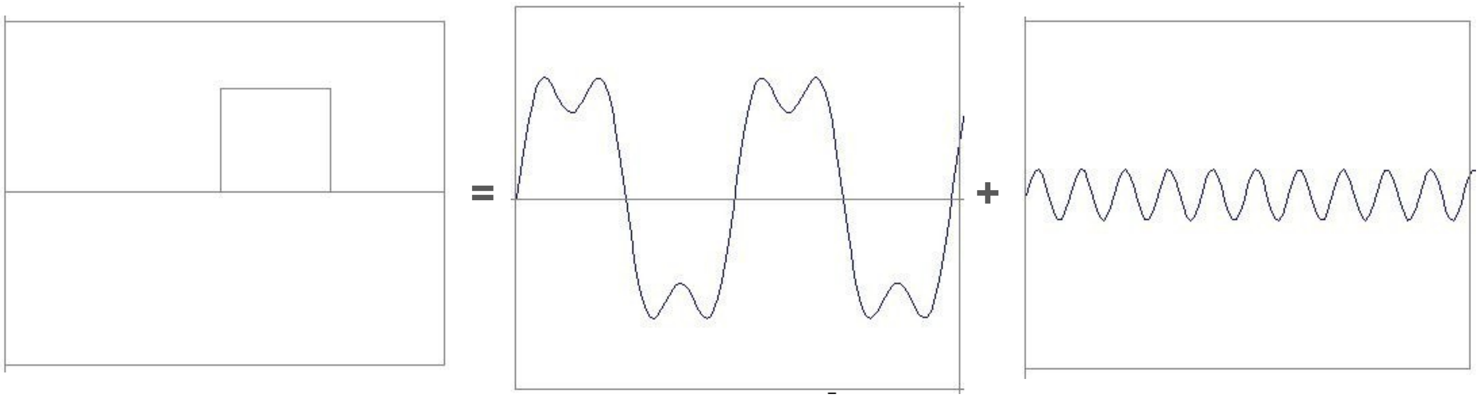


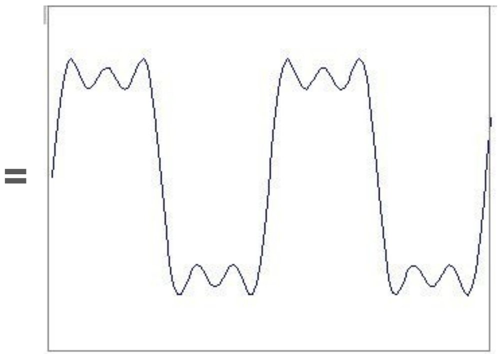
Fourier: Example (1D)



Increasing the number of frequency components (basis functions) allows any signal. waveform to be created ….

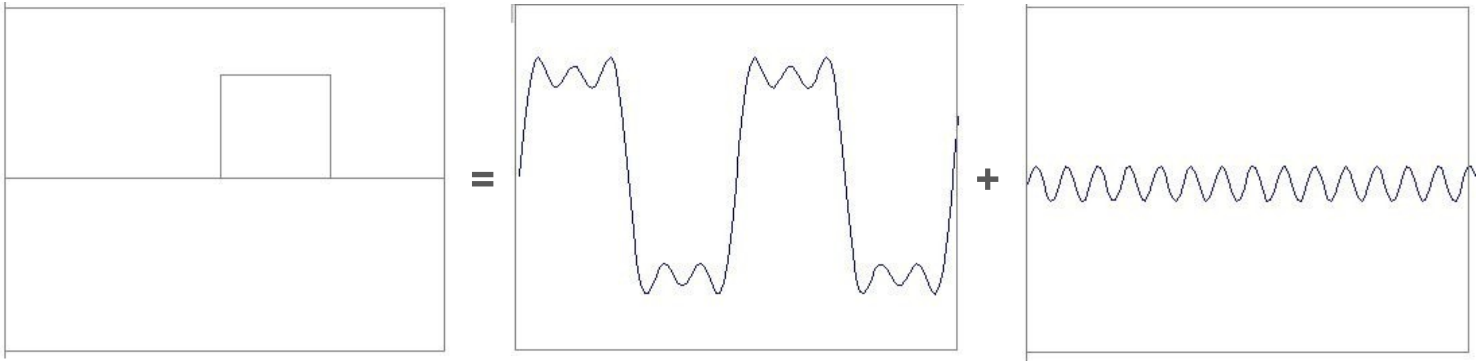
Fourier: Example (1D)

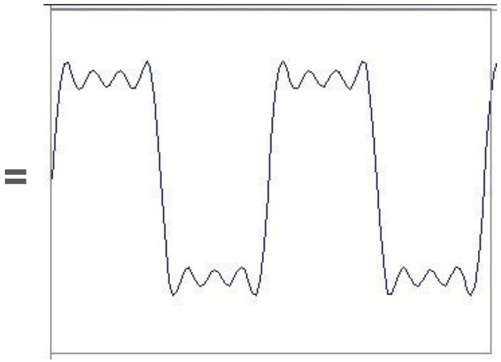




Increasing the number of frequency components (basis functions) allows any signal. waveform to be created ….

Fourier: Example (1D)





Increasing the number of frequency components (basis functions) allows any signal. waveform to be created ….

Fourier: Example (1D)

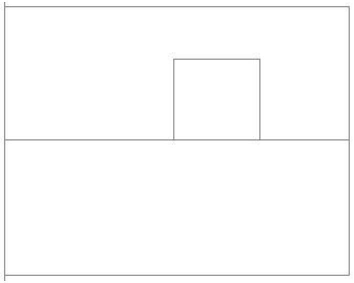
In fact, as we increase the number of frequency components the later, **higher frequency components contribute less to the overall shape (i.e. coarse outline) of the signal and more to the finer detail of its shape**.

Low frequency components influence the overall shape (i.e. coarse outline) of the signal.

Higher frequency components influence the finer (i.e. edge) detail of the signal.

Fourier: Example (1D)

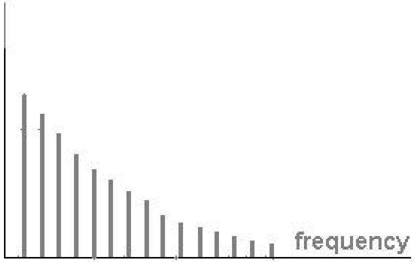
In fact, as we increase the frequency then each new component contributes less and less to the overall reconstruction of the signal …



$$= \frac{4}{\pi}\sum_{k=1}^{\infty}\frac{\sin\left(2\pi(2k-1)ft\right)}{2k-1}$$

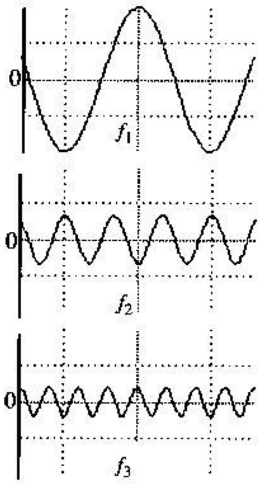
$$= \frac{4}{\pi}\left(\sin(2\pi ft) + \frac{1}{3}\sin(6\pi ft) + \frac{1}{5}\sin(10\pi ft) + \cdots\right)$$

As such, the corresponding coefficients get smaller as frequency increases … .

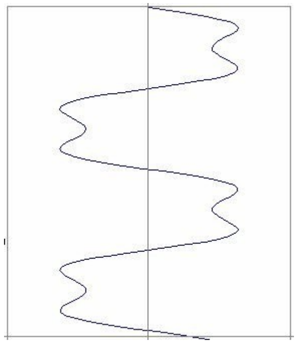


Fourier: Transform

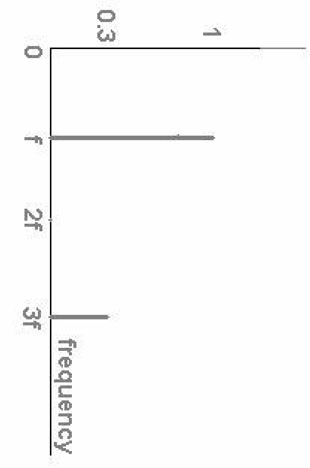
Fourier Transform: Just a change of mathematical basis: $M * f(x) = F(\omega)$



*



=



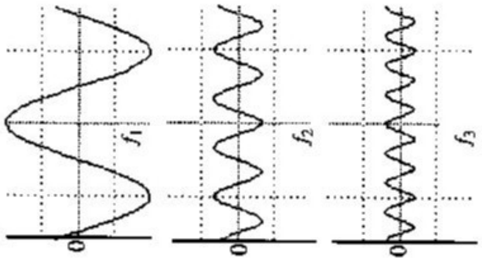
F() = Fourier domain representation (as coefficients of frequencies)

f(x) = spatial domain representation (as samples of a signal)

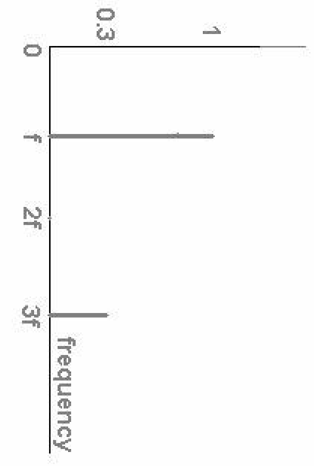
M = set of basis functions

Fourier: Transform

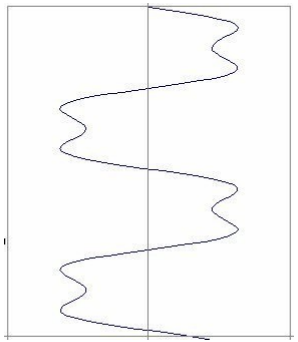
Inverse Fourier Transform: Just a change of mathematical basis: $M^{-1} * F(\omega) = f(x)$



*



=



F() = Fourier domain representation (as coefficients of frequencies)

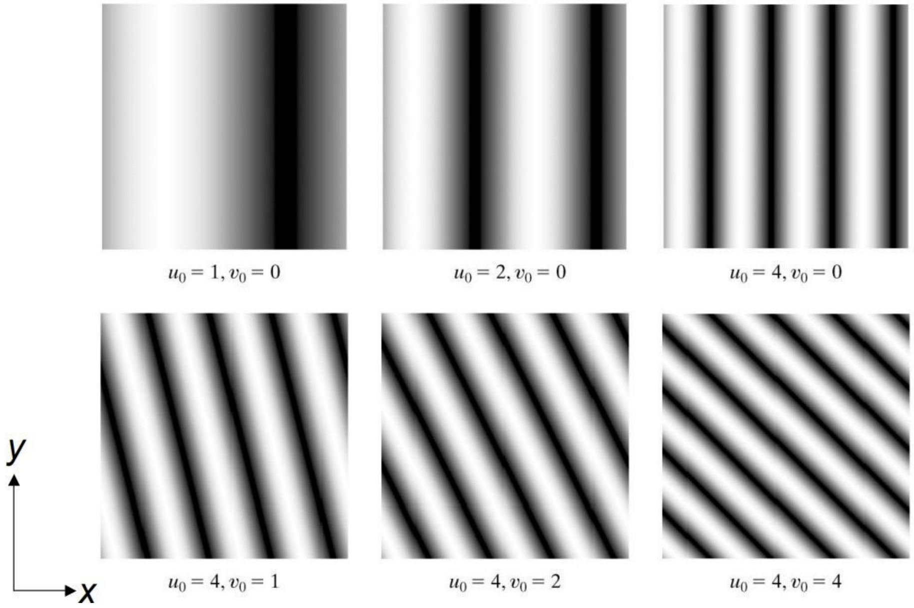
f(x) = spatial domain representation (as samples of a signal)

M = set of basis functions

Fourier: 2D

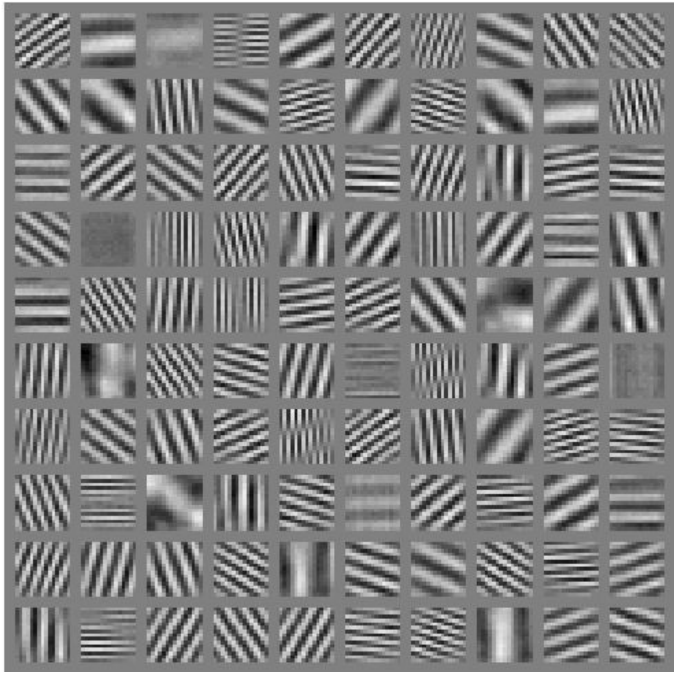
Using 2D spatial frequency sine waves as basis functions:

$$s(x,y) = \sin\left[2\pi\left(u_0 x + v_0 y\right)\right]$$



Fourier: 2D

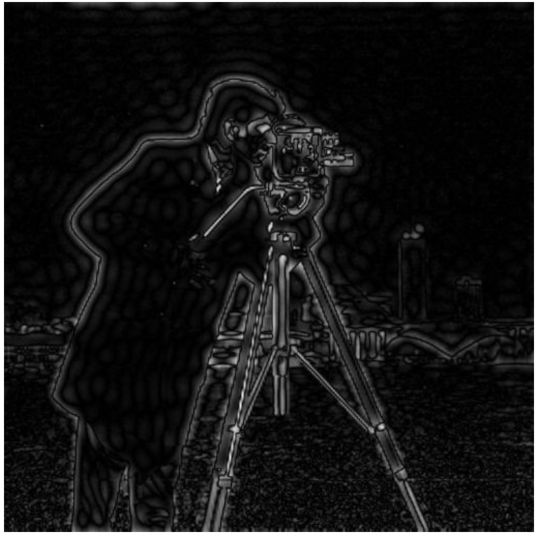
Using 2D spatial frequency sine waves as basis functions:



Fourier: Image Frequencies

Input

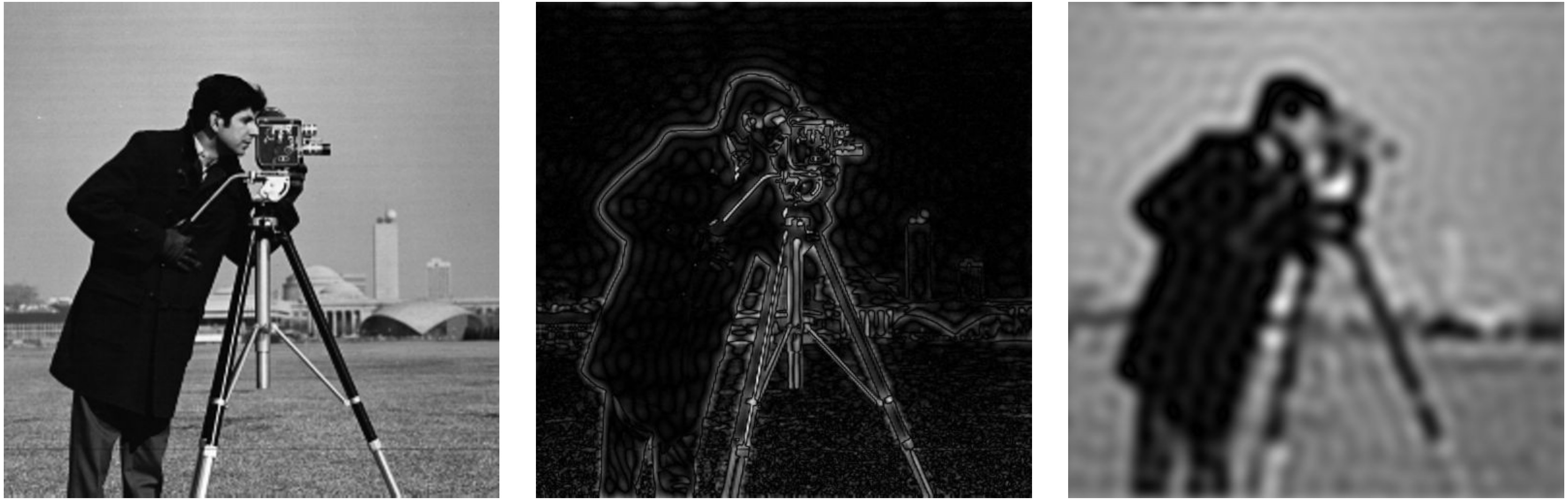
High Frequencies



Low Frequencies

Fourier: Image Frequencies

Higher frequency components influence the finer (i.e. edge) detail of the image.
Low frequency components influence the overall shape (i.e. coarse outline) of the image.



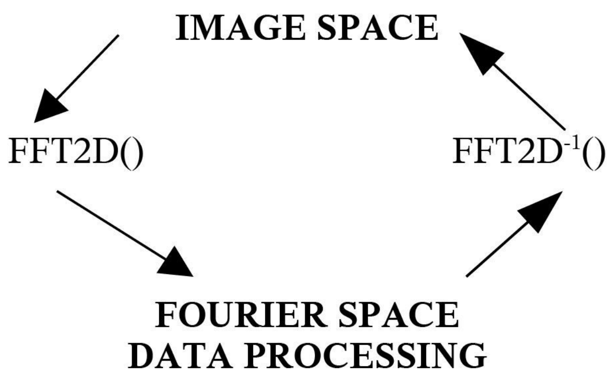
Fourier: Discrete Transform (and Inverse Transform)

Fourier Transform of an image via Discrete Fourier Transform (DFT):

- Inverse Discrete Fourier Transform (DFT-1):
    - Why “discrete” ?: due to discrete nature of image sampling (pixels).

Computable Efficiently and Rapidly via 2D Fast Fourier Transform (FFT) algorithm on *N* discrete samples:

- 2D FFT is a series of 2N 1D FFT
- 1D FFT is $O(N\log_2 N)$
- Algorithm details:beyond scope of course



Fourier: Discrete Convolution Theorem

Fourier Space Filters:

- Multiplicative image operations.
- Operate on the Discrete Fourier Transform (DFT) of image.
- Why “filters”?
  - Suppress certain frequencies whilst leaving others unchanged.
  - Synonymous to filtering in the frequency domain.

Convolution operation in spatial domain (pixel space):

$$f_{output}(i,j) = \sum_{k=-\lfloor N/2 \rfloor}^{\lfloor N/2 \rfloor} \sum_{l=-\lfloor M/2 \rfloor}^{\lfloor M/2 \rfloor} m_{kl} I_{input}(i+k, j+l)$$

S, *M* and *F* denote DFT's of f_{output}, m_{kl} and f_{input} respectively then **convolution in Fourier space** transforms to:

$$S = P\,F$$

[Known as the (Discrete) Convolution Theorem]

Fourier: Discrete Convolution Theorem

Convolution in the frequency domain (Fourier space) reduces to multiplication in the spatial domain (Real space).

and (more importantly) vice versa … .

Multiplication in the frequency domain reduces to convolution in the spatial domain.

$$f_{output} = \sum \sum m I_{input} \quad == \quad DFT^{-1} (S = P F)$$

Fourier: Discrete Convolution Theorem

Why are we interested?

If m_{kl} is small then:

- Compute convolution directly in real space.

But, if m_{kl} is large then:

- More efficient to use a Fast Fourier Transform (FFT) to obtain a Fourier space version the image finput.
- Then perform the convolution-based filtering as multiplication in Fourier space.

Size of kernel m_{kl}: number of elements:

Answer: computational efficiency

Fourier: Complex Series

One of a number of linear polynomials which can be used to ‘model’ a continuous or discrete 2D function *f(x, y)*:

$$f(x,y)=\sum_{nm} c_{nm}\varphi_{nm}(x,y)$$

where:

$\Phi_{nm}(x, y)$ - **basis function**.

$c_{nm}(x, y)$ - (complex) **coefficients**.

$$\sum_{nm} \equiv \sum_{n}\sum_{m} \equiv \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}$$

Complex Fourier Series basis functions: $\Phi_{nm}(x, y) = exp(x)\ exp(y)$.

- model signals/images as a set of coefficients cnm
  - determine the relative weight/influence of each basis functions

Problem reduced to finding the complex coefficients c_{nm}.

Fourier: Visualising

DFT output = complex number valued output image:

- Containing the coefficients of the 2D Fourier series:
  - Known as the “Fourier spectrum” of the image, I_{input}.
- Dimension *n x m* (same as input).
- Can be displayed as the real or imaginary parts of the image.

$$F_{nm} = G_{nm} + iH_{nm}$$

where:

- $G_{nm} = real[F_{nm}]$
- $H_{nm} = imaginary[F_{nm}]$

Fourier: Visualising

Components commonly used for visualising (i.e. seeing!) the resulting Fourier spectrum itself:

Amplitude (magnitude) Spectrum: $$|F_{nm}| = \sqrt{G_{nm}^2 + H_{nm}^2}$$

Phase Spectrum: $$\varphi_{nm} = \tan^{-1}\left(\frac{H_{nm}}{G_{nm}}\right)$$

Power Spectrum: $$|F_{nm}|^2$$

Fourier: Visualising

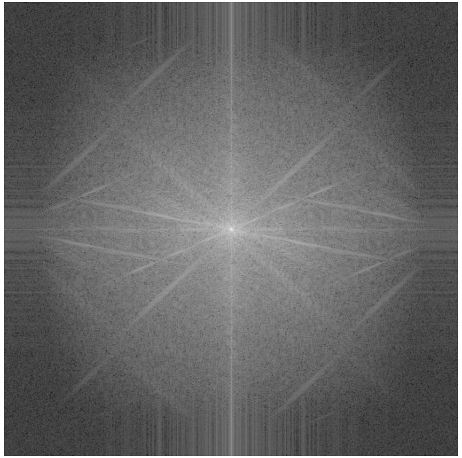
Transformed so image mean, *F(0,0)*, is centred (by convention):

- Known as the DC-component = average brightness of the image.
- Highest frequency present = *F(N-1,N-1)*.

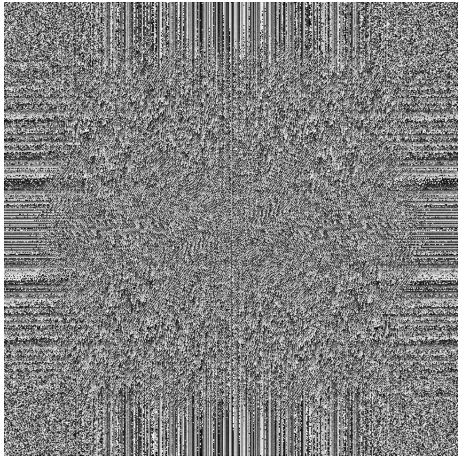
Display can be presented based on a logarithmic scale $ln(1+|F_{nm}|)$ as (floating point) range very large.

Input

Magnitude



Phase



Fourier: Visualising

Image contains all frequencies:

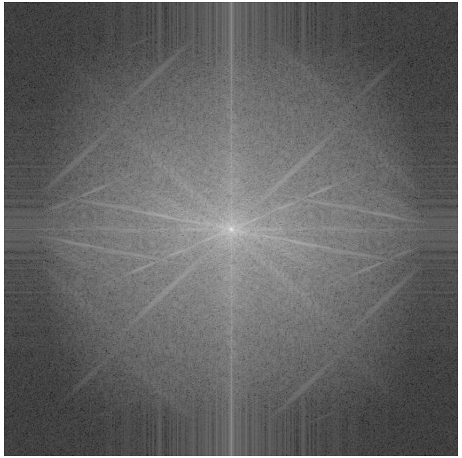
- Lower frequencies (close to amplitude image centre) larger than the higher frequencies present (outsides of the amplitude image). Thus: more information in lower frequencies.

Two predominant directions in the image - horizontally and vertically.

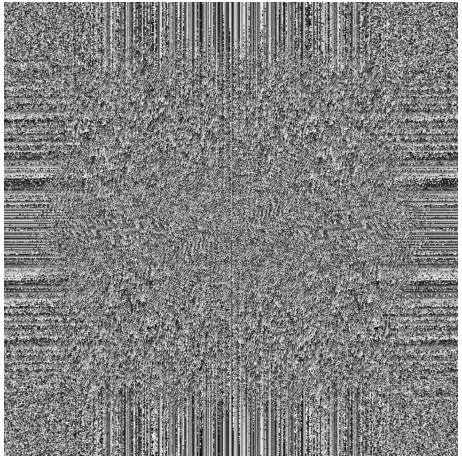
- Originate from the regular background patterns in original.

Input

Magnitude



Phase

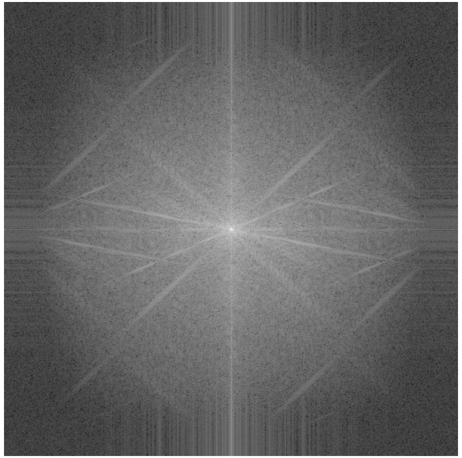


Fourier: Visualising

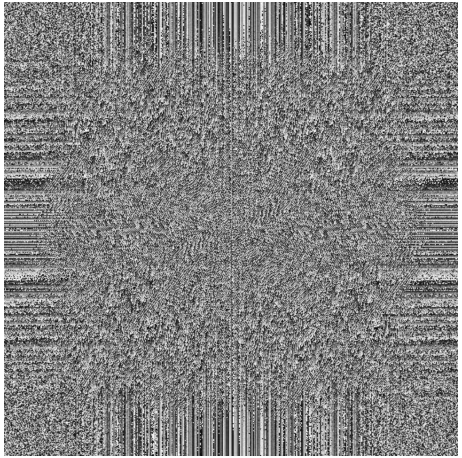
Phase spectrum = phase part of the corresponding frequency.
Vertical and horizontal features correspond to image patterns.
In general: not much new information about the structure of the spatial image.
Crucial in reconstructing the inverse Fourier transform – otherwise resulting spatial image is completely corrupted (as signals need phase + frequency information).

Input

Magnitude

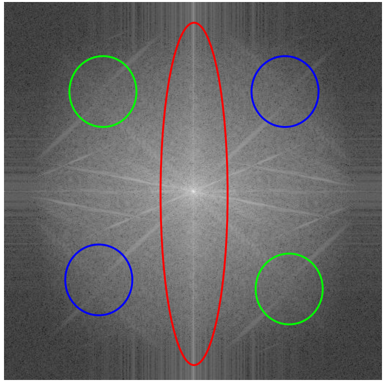


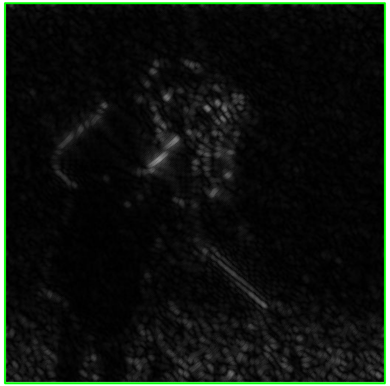
Phase

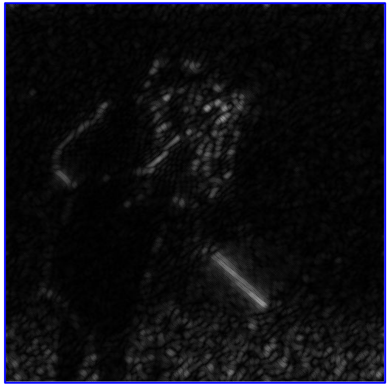


Fourier: Frequency Components

Removing frequencies in the Fourier domain results in frequency filtering in the spatial domain.







Fourier: Frequency Filtering

Removed frequency components = multiplied by a constant (effectively):

- Here (in previous example): constant = zero.
- (i.e. set to “black” = 0).

By doing multiplication in the Fourier domain:

- We achieved a low pass filter of the image.
- (e.g. smoothing, we keep the low frequency information).
- == convolution in the spatial domain.
- This is the crux of the Discrete Convolution Theorem.

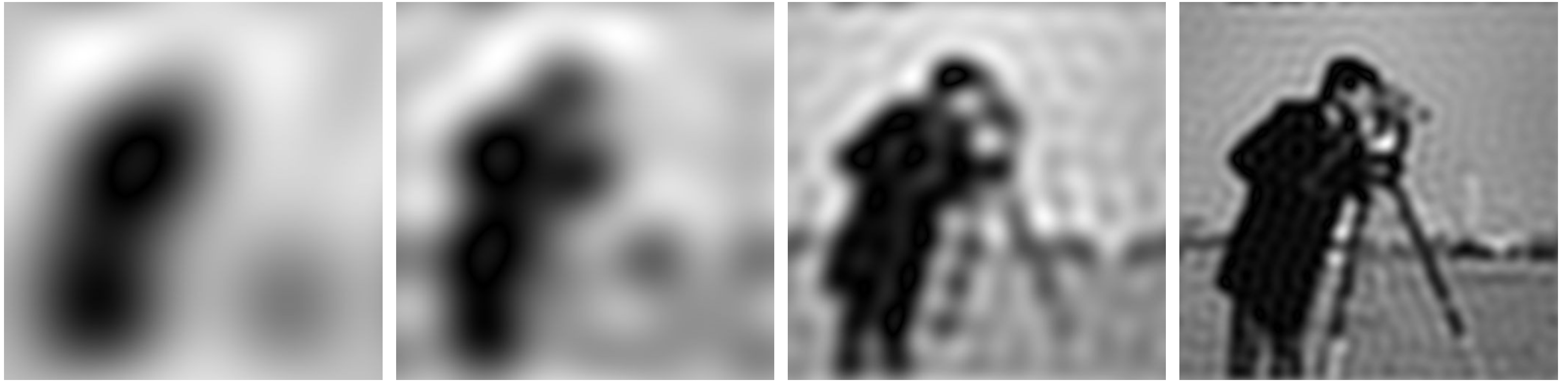
Translate the convolution mask to a Fourier image itself and then multiply by this mask:

$$S = P\,F$$

- where Fourier mask P: > 0 keep the frequencies and 0 remove frequencies.

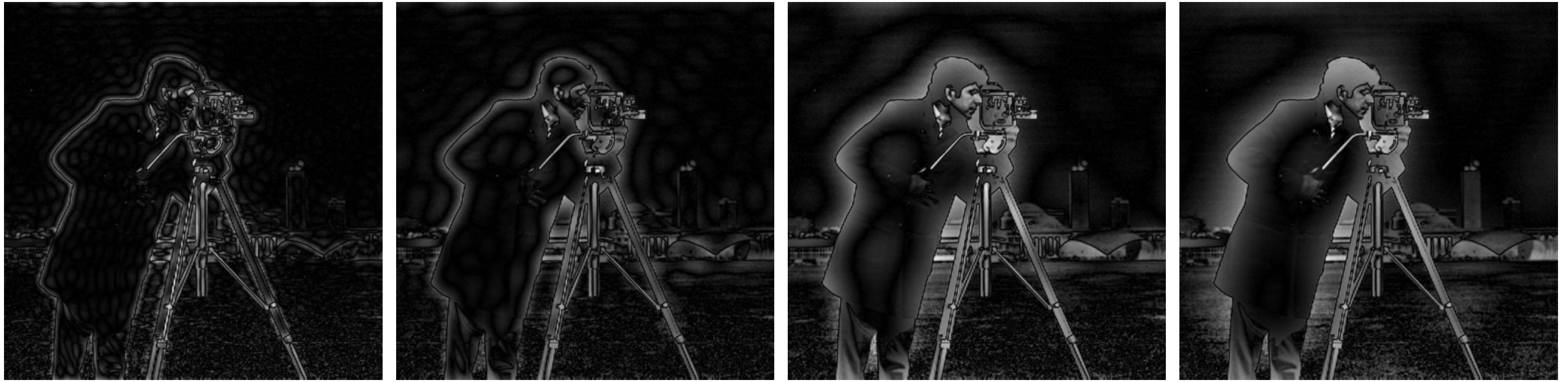
Fourier: Frequency Filtering

Below are low-pass filtered images representing the lowest spatial frequency elements of the original:

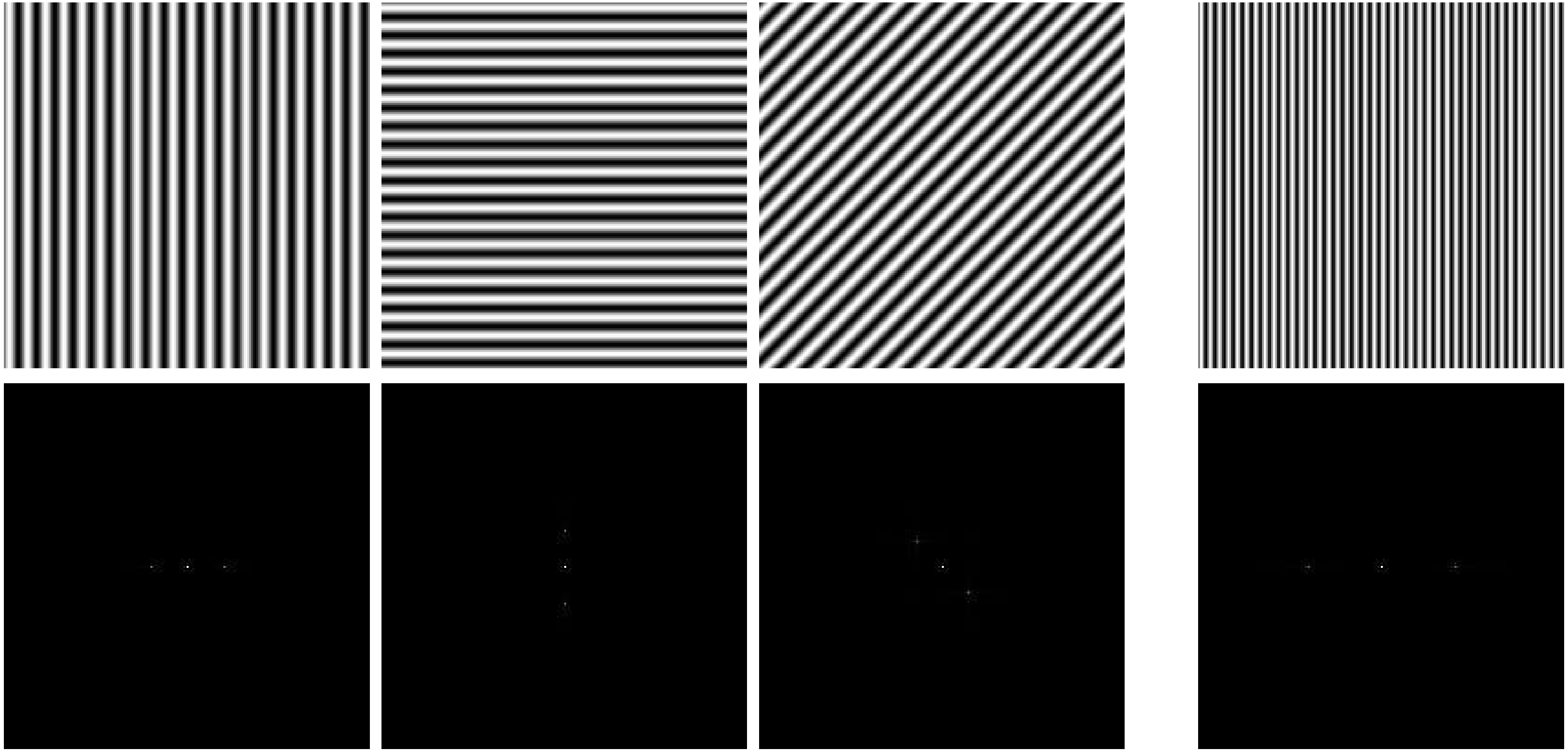


Fourier: Frequency Filtering

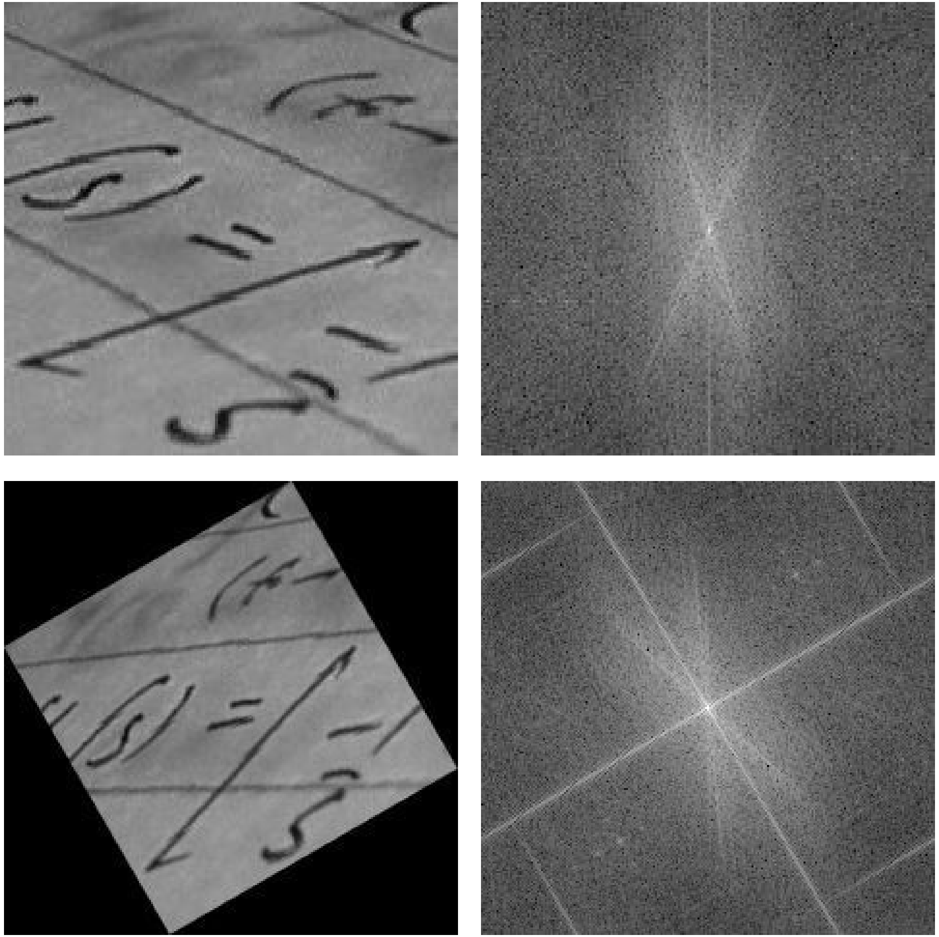
Below are high-pass filtered images representing the highest spatial frequency elements of the original:



Fourier: DFT Examples



Fourier: DFT Examples



Fourier: DFT

The DFT is only an approximation to the Fourier Transform of the continuous image from which the digital image was obtained!

Edge Effects:

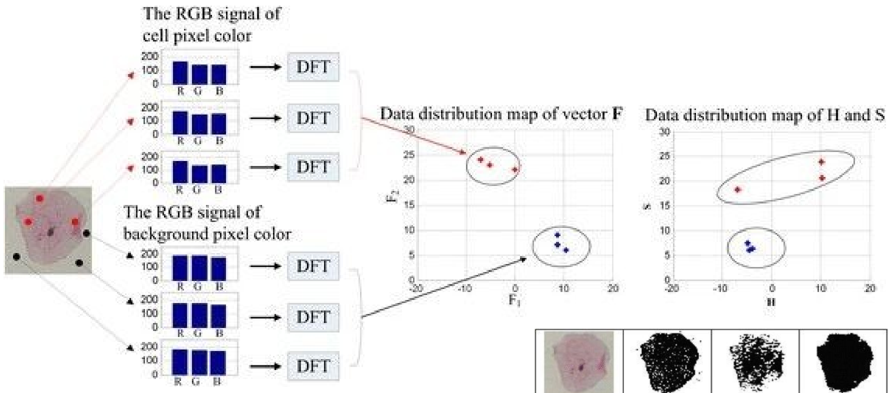
- DFT assumes digital image is one period of a periodic function (in all directions!).
- If the values at the opposite edges of the function are not the same, the result is additional edge discontinuities.

Frequency Aliasing:

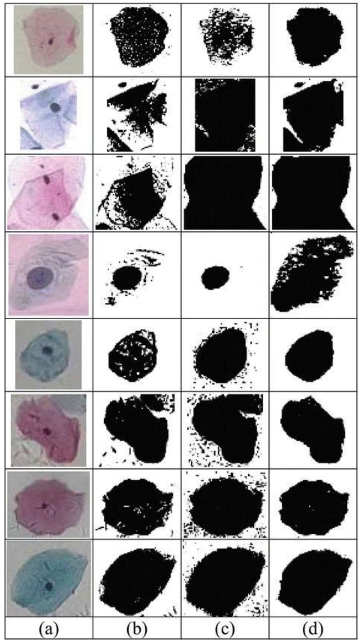
- DFT inaccuracy from under-sampling causes frequency aliasing.
- Practice highest frequency not usually known exactly:
  - Low pass filter to impose a limit.

Applications

The proposed color space treats RGB as a sampled spectral signal, called a RGB signal. Then, based on the Fourier spectrum analysis of the RGB signal, the 1-D discrete Fourier transform is introduced to describe the color features of microscopic images. K-means clustering experiments on two microscopic image datasets validate the superiority of the proposed IF1F2 color space compared with the classical RGB and HSV color spaces.



Guan, T., Zhou, D., Xu, C. et al. A novel RGB Fourier transform-based color space for optical microscopic image processing. Robot. Biomim. 1, 16 (2014)



K -means clustering: a) Original microscopic images. (b) Results of the RGB space. (c) Results of the HSV space. (d) Results of the proposed color space.