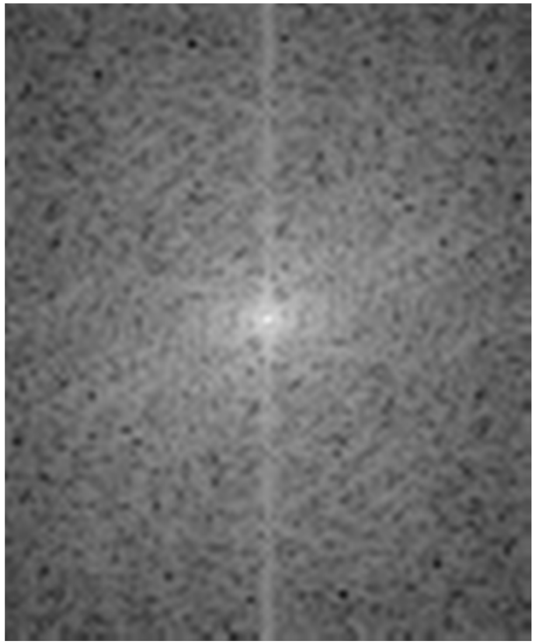
The Fourier Transform

The Fourier Transform

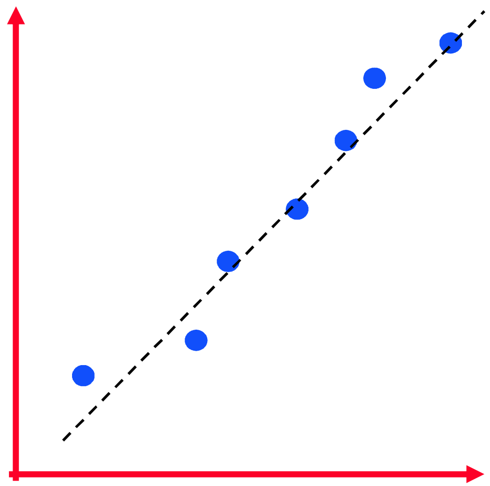


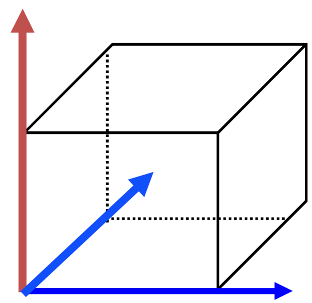
Jean Baptiste Joseph Fourier

1768-1830

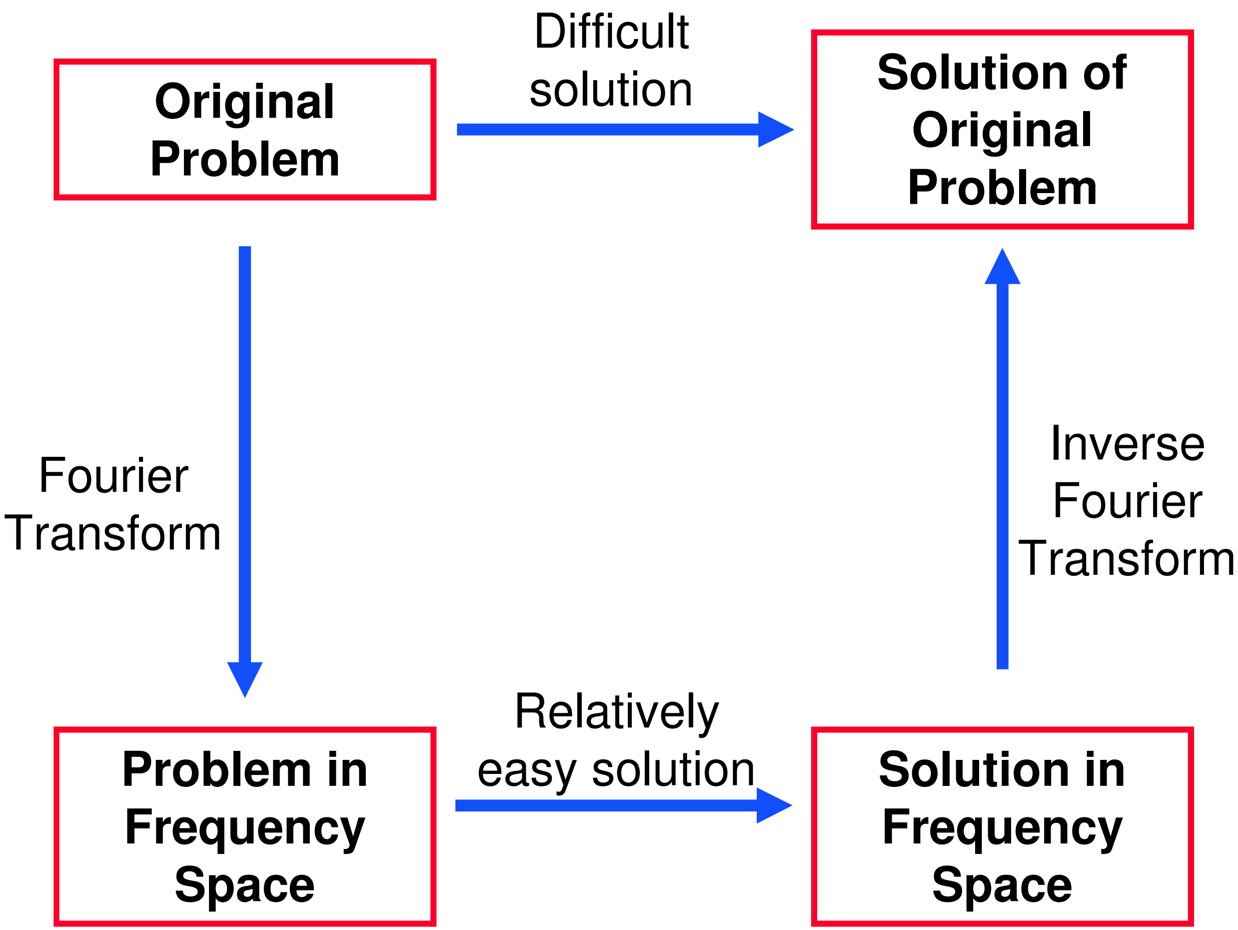
Efficient Data Representation

- Data can be represented in many ways.
- Advantage using an appropriate representation.
- Examples:
  - Noisy points along a line
  - Color space red/green/blue v.s. Hue/Brightness





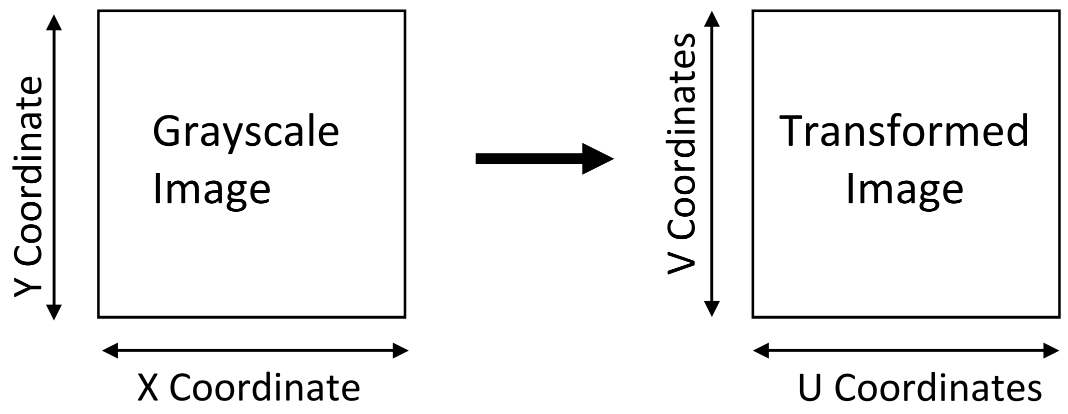
Why do we need representation in the frequency domain?



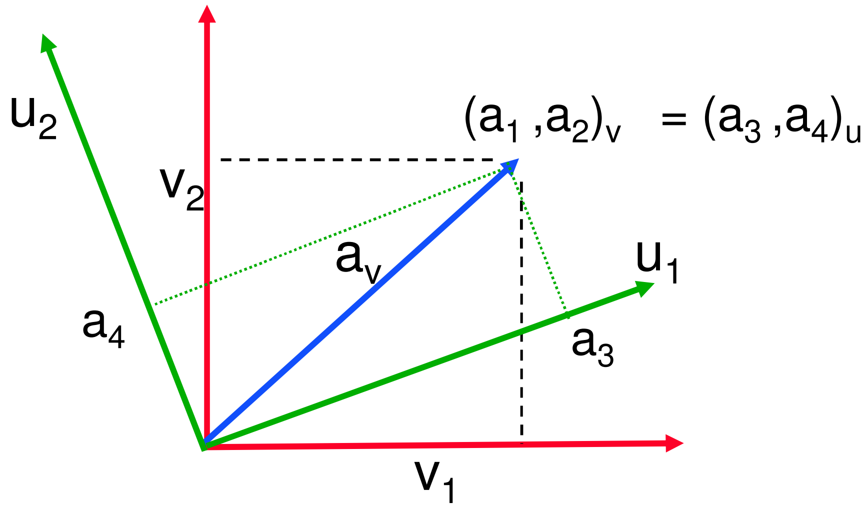
How can we enhance such an image?

Transforms

1. Basis Functions.
2. Method for finding the image given the transform coefficients.
3. Method for finding the transform coefficients given the image.



Change of Basis



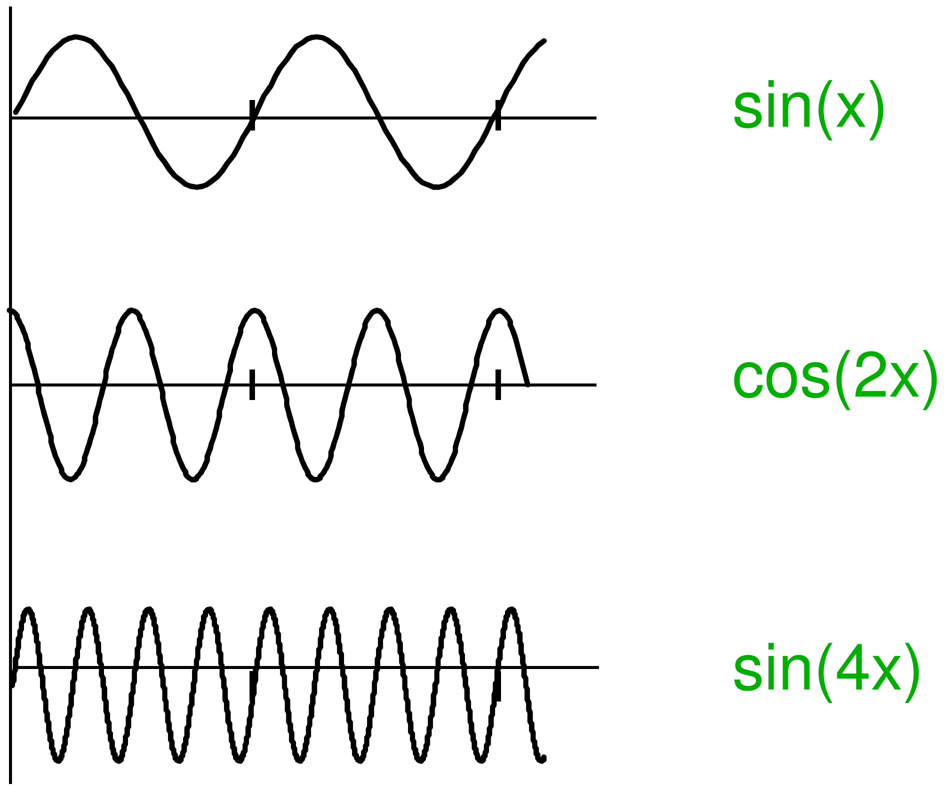
$$a_{iu} = \langle a_v, u_i \rangle$$

$$a_v = \sum_i a_{iu} u_i$$

where $\langle c, b \rangle = c^{T*} b = \sum c^*(i) b(i)$

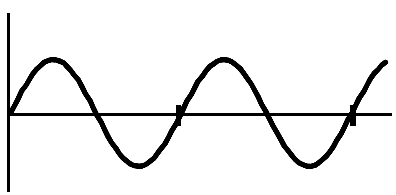
The Fourier basis functions

Basis Functions are sines and cosines

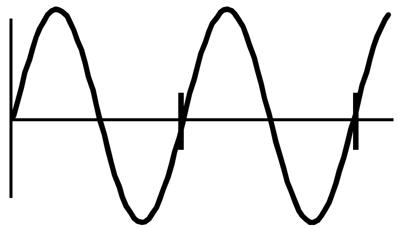


The Fourier basis functions

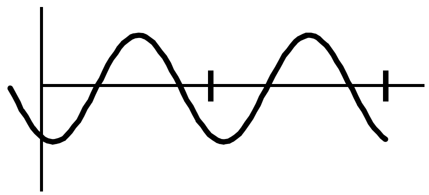
The transform coefficients determine the amplitude and phase:



$a \sin(2x)$



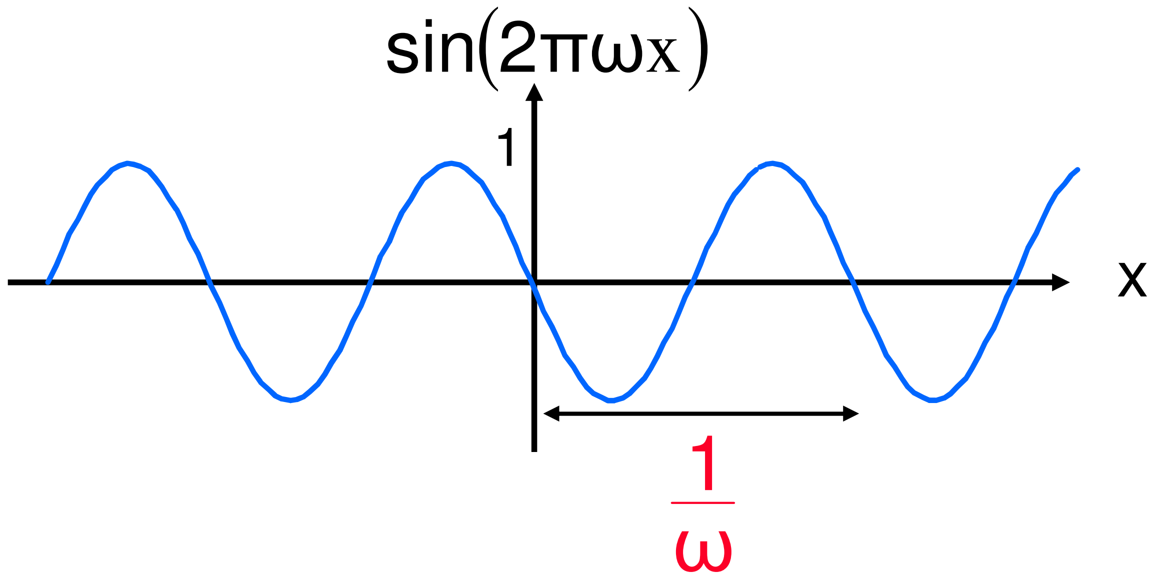
$2a \sin(2x)$



$-a \sin(2x+\phi)$

The Fourier basis functions

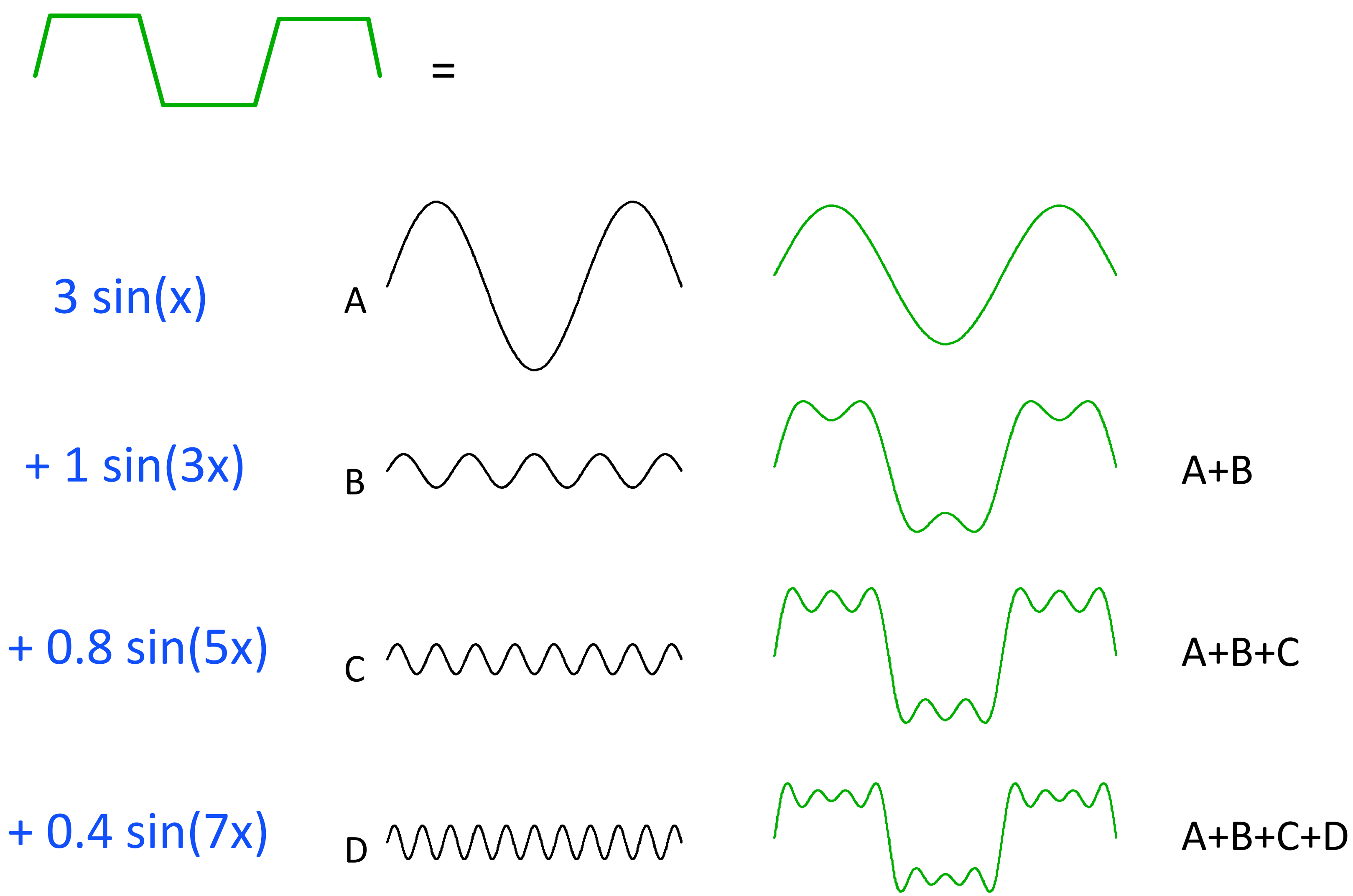
– Define $K=2\pi\omega$



– The *wavelength* of $\sin(2\pi\omega x)$ is $1/\omega$.

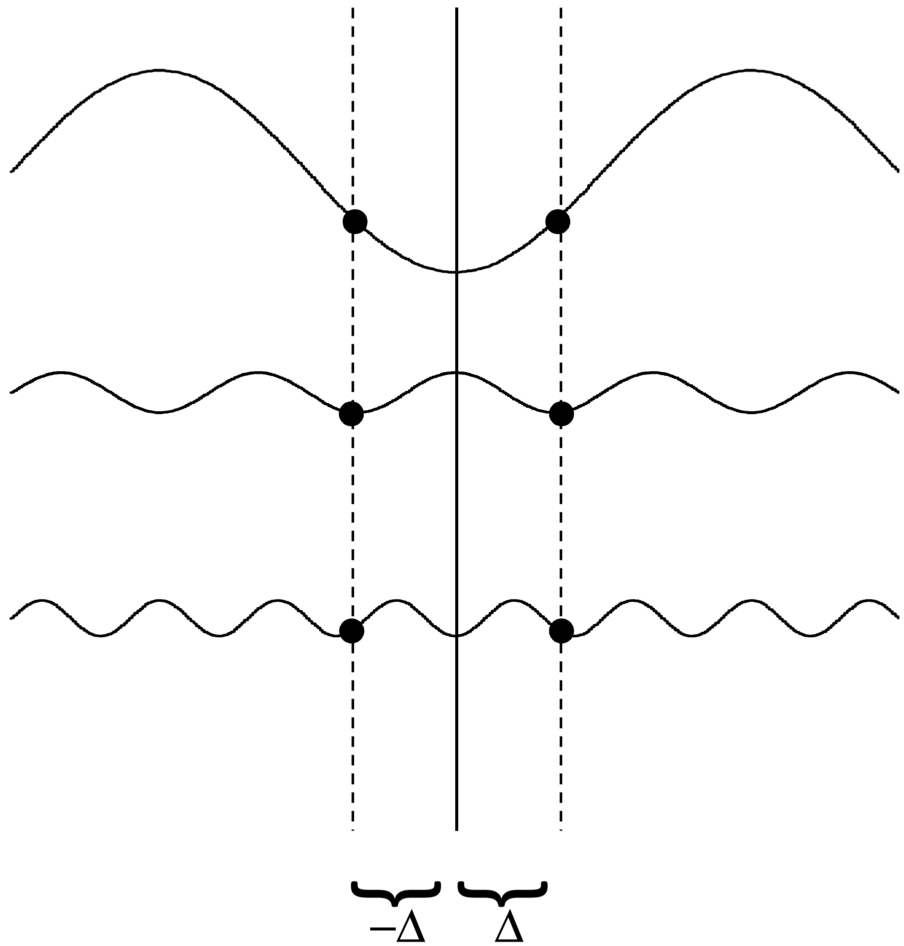
– The *frequency* is ω .

Every function equals a sum of sines and cosines



Sum of cosines only $\longrightarrow$ symmetric functions

Sum of sines only $\longrightarrow$ antisymmetric functions



Fourier Transform

$$f(x) = C_0 + \underbrace{C_1\cos(x) + S_1\sin(x)} + \ldots + \underbrace{C_k\cos(kx) + S_k\sin(kx)} + \ldots$$

Terms are considered in pairs:

$$C_k\cos(kx) + S_k\sin(kx) = R_k \sin(kx + \theta_k)$$

$$\text{where} \quad R_k = \sqrt{C_k^{\,2} + S_k^{\,2}} \quad \text{and} \quad \theta_k = \tan^{-1}\left(\frac{S_k}{C_k}\right)$$

Using Complex Numbers:

$$\cos(kx)\,,\ \sin(kx) \longrightarrow e^{ikx} \qquad \boxed{e^{ikx} = \cos(kx) + i\sin(kx)}$$

$$C_k\cos(kx) + S_k\sin(kx) \longrightarrow \underbrace{R\, e^{i\theta}}_{\text{Amplitude+phase}} e^{ikx}$$

Fourier Transform

Fourier Basis:

$$B_\omega(x) = e^{i2\pi\omega x} = \cos(2\pi\omega x) + i\sin(2\pi\omega x)$$

Fourier Coefficients:

$$F(\omega) = R_\omega \, e^{i\theta_\omega}$$

(Determines Amplitude+Phase of Basis)

Fourier Transform:

$$f(x) = R_0 \, e^{i\theta_0} e^{i2\pi 0x} + R_1 \, e^{i\theta_1} e^{i2\pi 1x} + \ldots + R_\omega \, e^{i\theta_\omega} e^{i2\pi\omega x} + \ldots$$

The 1D Continuous Fourier Transform

The Continuous Fourier Transform finds $F(\omega)$ given the (cont.) signal $f(x)$:

$$\boxed{F(\omega) = \int_x f(x)\, e^{-i2\pi\omega x}\, dx}$$

$B_\omega(x) = e^{i2\pi\omega}$ is a complex wave function for each w .

The Inverse Continuous Fourier Transform composes a signal $f(x)$ given $F(\omega)$:

$$\boxed{f(x) = \int_\omega F(\omega)\, e^{i2\pi\omega x}\, d\omega}$$

The discrete Fourier basis functions are

$$\boxed{b_k(x) = \frac{1}{\sqrt{N}} e^{\frac{2\pi ikx}{N}} \qquad \begin{matrix} k = 0..N-1 \\ x = 0..N-1 \end{matrix}}$$

For frequency k the Fourier coefficient is:

$$C_k \cos\left(\frac{2\pi kx}{N}\right) + S_k \sin\left(\frac{2\pi kx}{N}\right) \rightarrow \left(R_k e^{i\theta_k}\right) e^{i2\pi kx/N}$$

$$\boxed{F(k) = R_k e^{i\theta_k}}$$

The 1D Discrete Fourier Transform (DFT)

$$F(k) = \langle f(x), b_k(x) \rangle = \sum_{x=0}^{N-1} f(x) e^{\frac{-2\pi ikx}{N}}$$

k = 0, 1, 2, ..., N-1

Matlab: F=fft(f);

The Inverse Discrete Fourier Transform (IDFT) is defined as:

$$f(x) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) e^{\frac{2\pi ikx}{N}}$$

x = 0, 1, 2, ..., N-1

Matlab: F=ifft(f);

<u>Remark</u>: Normalization constant might be different!

Discrete Fourier Transform - Example

f(x) = [2 3 4 4]

$$F(0) = \sum_{x=0}^{3} f(x) e^{\frac{-2\pi 0 x}{4}} = \sum_{x=0}^{3} f(x) \cdot 1 =$$

$$= (f(0) + f(1) + f(2) + f(3)) = (2+3+4+4) = 13$$

$$F(1) = \sum_{x=0}^{3} f(x) e^{\frac{-2\pi i x}{4}} = \ = [2e^{0}+3e^{-i\pi/2}+4e^{-\pi i}+4e^{-i3\pi/2}] = [-2+i]$$

$$F(2) = \sum_{x=0}^{3} f(x) e^{\frac{-4\pi i x}{4}} = [2e^{0}+3e^{-i\pi}+4e^{-2\pi i}+4e^{-3\pi i}] = [-1-0i] = -1$$

$$F(3) = \sum_{x=0}^{3} f(x) e^{\frac{-6\pi i x}{4}} = [2e^{0}+3e^{-i3\pi/2}+4e^{-3\pi i}+4e^{-i9\pi/2}] = [-2-i]$$

$$\boxed{\text{DFT of } [2\ \ 3\ \ 4\ \ 4] \text{ is } [\,13\ \ (-2+i)\ \ -1\ \ (-2-i)\,]}$$

The 2D Discrete Fourier Basis

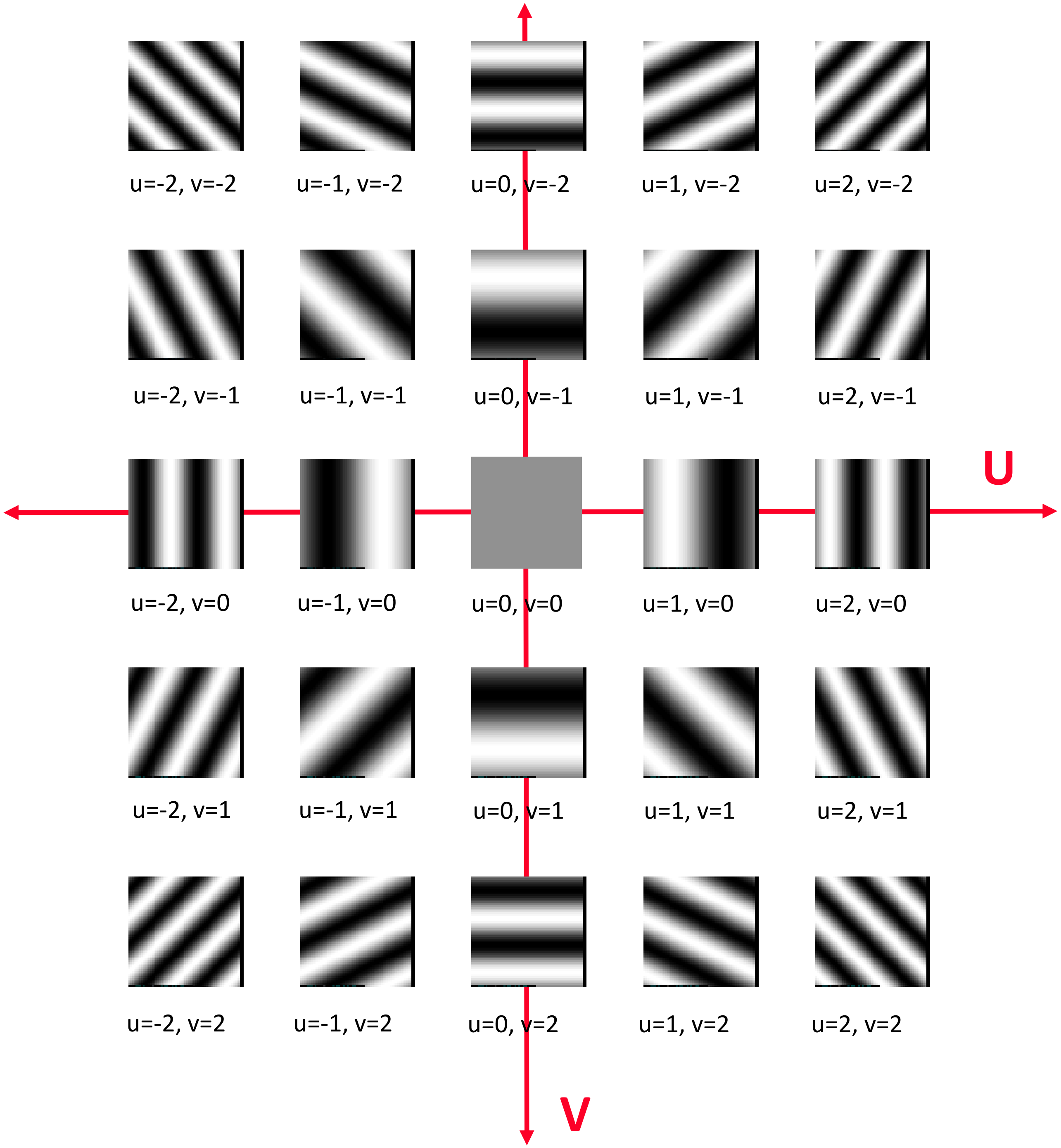
For a 2D image f(x,y) x=0..N-1, y=0..M-1, the DFT basis functions are 2D:

$$B_{u,v}(x,y) = \frac{1}{\sqrt{MN}} e^{2\pi i\left(\frac{ux}{N}+\frac{vy}{M}\right)} \qquad u=0..N-1,\ \ v=0..M-1$$

For frequency u,v the Fourier coefficient is:

$$F(u,v) = \left\langle f(x,y), B_{u,v}(x,y)\right\rangle =$$

$$= \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y) B^{*}_{u,v}(x,y)$$


u=-2, v=-2
u=-1, v=-2
u=0, v=-2
u=1, v=-2
u=2, v=-2
u=-2, v=-1
u=-1, v=-1
u=0, v=-1
u=1, v=-1
u=2, v=-1
U
u=-2, v=0
u=-1, v=0
u=0, v=0
u=1, v=0
u=2, v=0
u=-2, v=1
u=-1, v=1
u=0, v=1
u=1, v=1
u=2, v=1
u=-2, v=2
u=-1, v=2
u=0, v=2
u=1, v=2
u=2, v=2
V

The 2D Discrete Fourier Transform

For a 2D image f(x,y) x=0..N-1, y=0..M-1,

the 2D Discrete Fourier Transform is defined as:

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y-0}^{M-1} f(x,y) e^{-2\pi i (ux/N + vy/M)}$$

$u = 0, 1, 2, \ldots, N\text{-}1$
$v = 0, 1, 2, \ldots, M\text{-}1$

Matlab: F=fft2(f);

The Inverse Discrete Fourier Transform (IDFT) is defined as:

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{N-1} \sum_{v-0}^{M-1} F(u,v) e^{2\pi i (ux/N + vy/M)}$$

$y = 0, 1, 2, \ldots, N\text{-}1$
$x = 0, 1, 2, \ldots, M\text{-}1$

Matlab: f=ifft2(F);

The Fourier Transform - Summary

- F(k) is the Fourier transform of f(x):

$$\tilde{F}\{f(x)\} = F(k)$$

- f(x) is the inverse Fourier transform of F(k):

$$\tilde{F}^{-1}\{F(k)\} = f(x)$$

- f(x) and F(k) are a Fourier pair.

- f(x) is a representation of the signal in the Spatial Domain and F(k) is a representation in the Frequency Domain.

- The Fourier transform F(k) is a function over the complex numbers:

$$F(k) = R_k e^{i\theta_k}$$

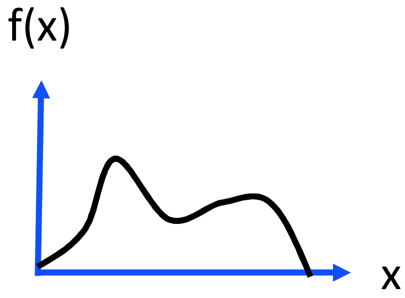
  - R_k tells us how much of frequency k is needed.
  - θ_k tells us the shift of the Sine wave with frequency k.

- Alternatively:

$$F(k) = a_k + ib_k$$

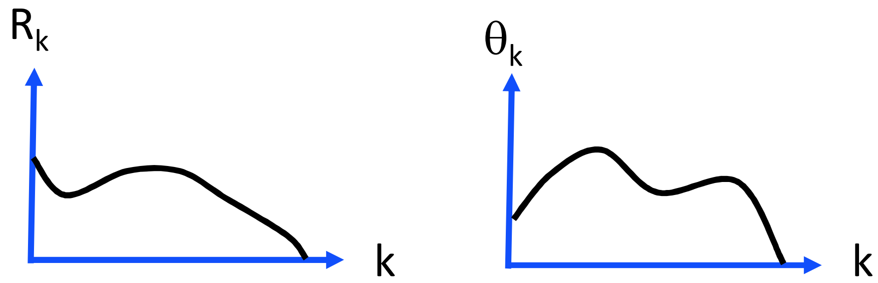
  - a_k tells us how much of cos with frequency k is needed.
  - b_k tells us how much of sin with frequency k is needed.

The Frequency Domain



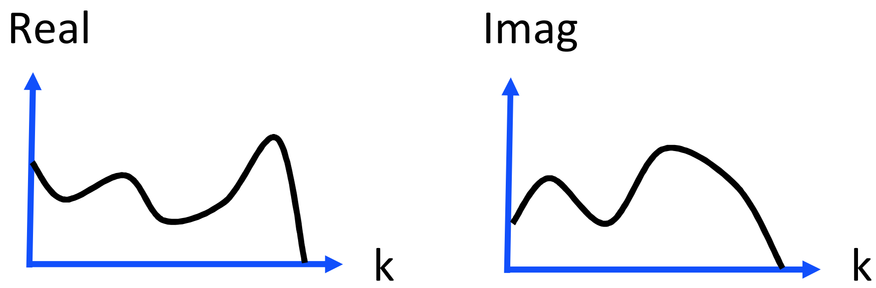
The signal f(x)

$$F(k) = R_k e^{i\theta_k}$$



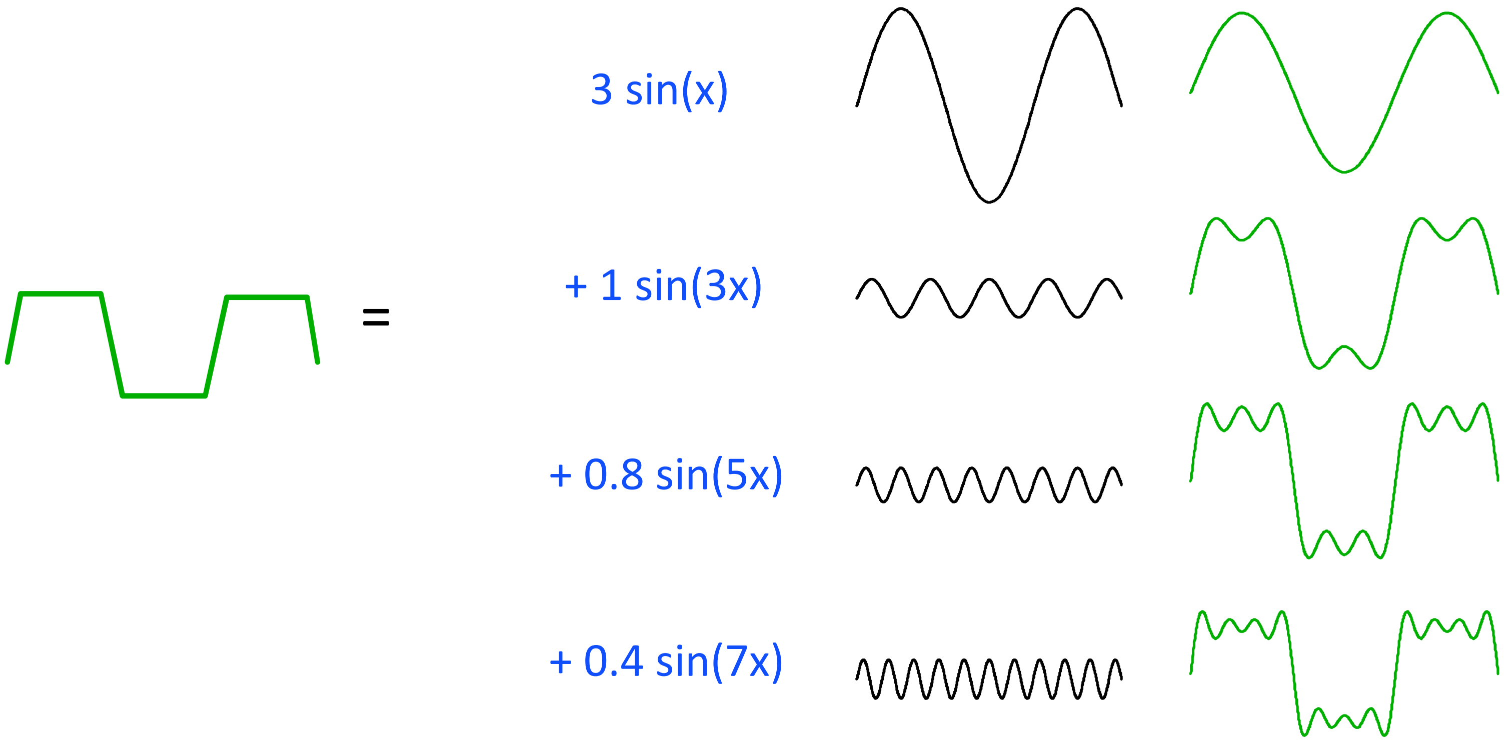
Amplitude (spectrum) and Phase

$$F(k) = a_k + ib_k$$



Real and Imaginary

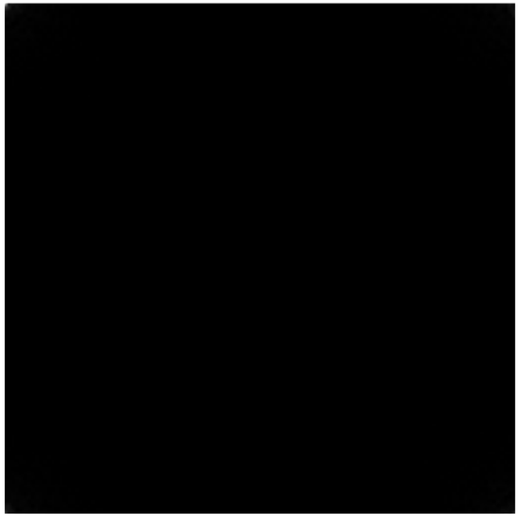
- $|R_k|^2 = F^*(k)\, F(k)$ - is the power spectrum of F(k) .
- If f(x) has a lot of fine details, $|R_k|^2$ will be high for high k.
- If f(x) is "smooth", $|R_k|^2$ will be low for high k.



Demo

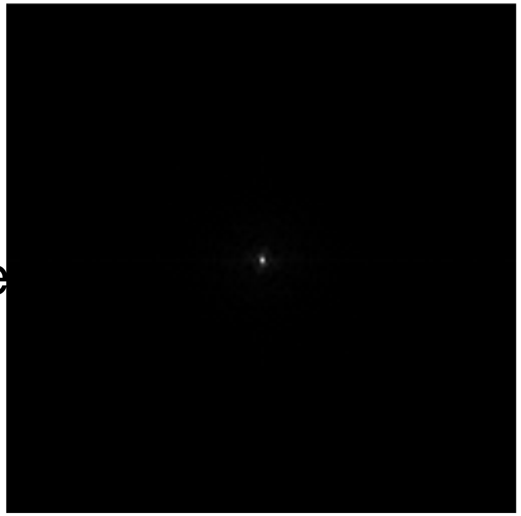
Displaying the Fourier Transform

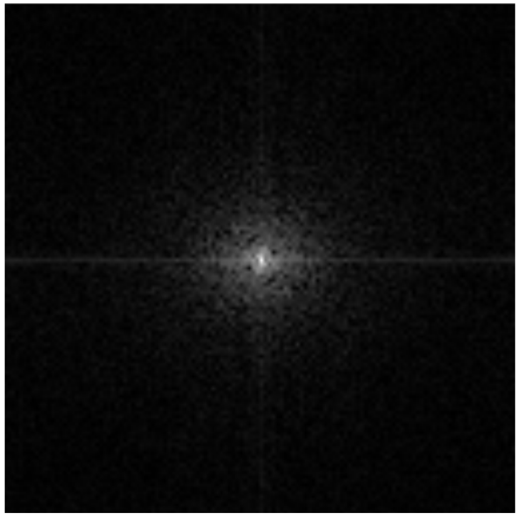
Original



Fourier Image
|F(u,v)|

Shifted
Fourier Image





Shifted
Log Fourier
log(1+ |F(u,v)|)

Displaying the Fourier Transform

- F(u,v) is a Fourier transform of f(x,y) and it has complex entries.

F = fft2(f);

- In order to display the Fourier Spectrum |F(u,v)|
  - Reduce dynamic range of |F(u,v)| by displaying the log:

D = log(1+abs(F));

  - Cyclically rotate the image so that F(0,0) is in the center:

D = fftshift(D);

Example:

|F(u)| = 100 4 2 1 0 0 1 2 4

Display in Range ([0..100]):

log(1+|F(u)|) = 4.62 1.61 1.01 0.69 0 0 0.69 1.01 1.61

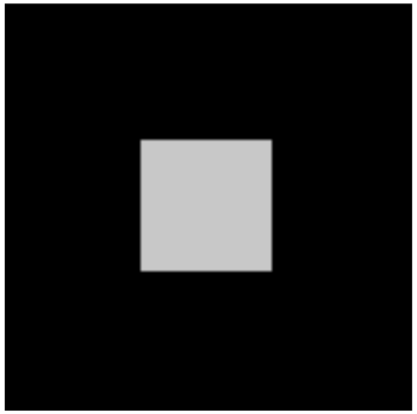
log(1+|F(u)|)/0.0462 = 100 40 20 10 0 0 10 20 40

fftshift(log(1+|F(u)|) = 0 10 20 40 100 40 20 10 0

imagesc(D);

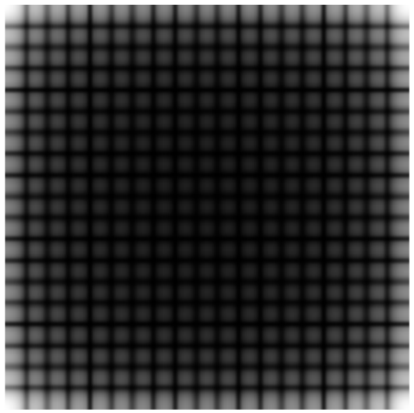
Displaying the Fourier Transform

Original

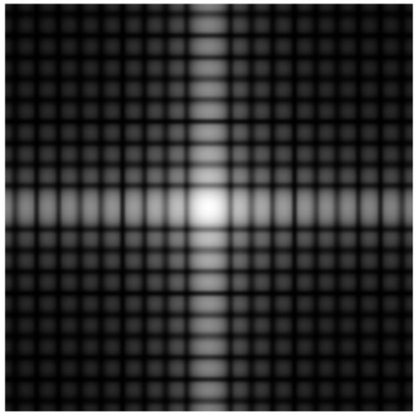


|F(u,v)|

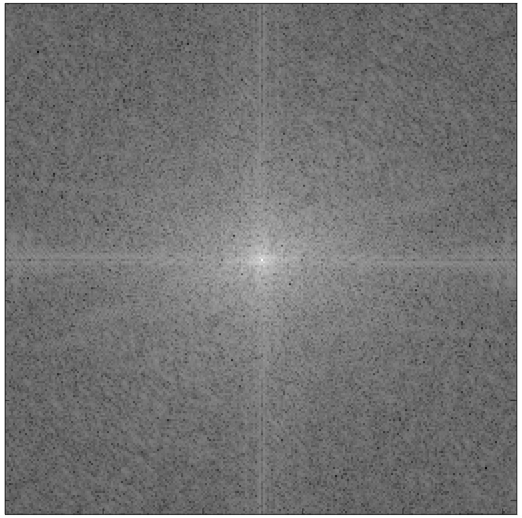
log(1 + |F(u,v)|)

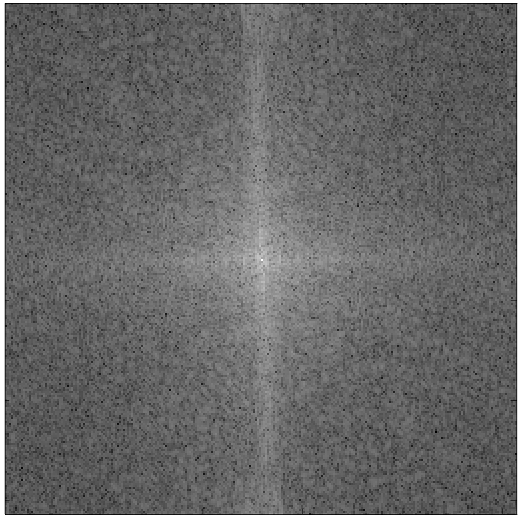


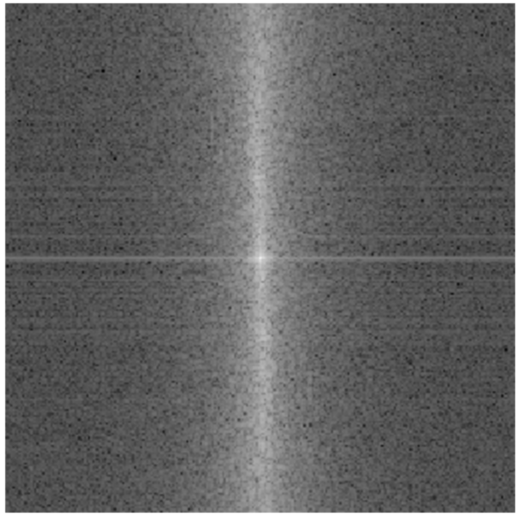
fftshift(log(1 + |F(u,v)|))



Fourier Transform – Image Examples







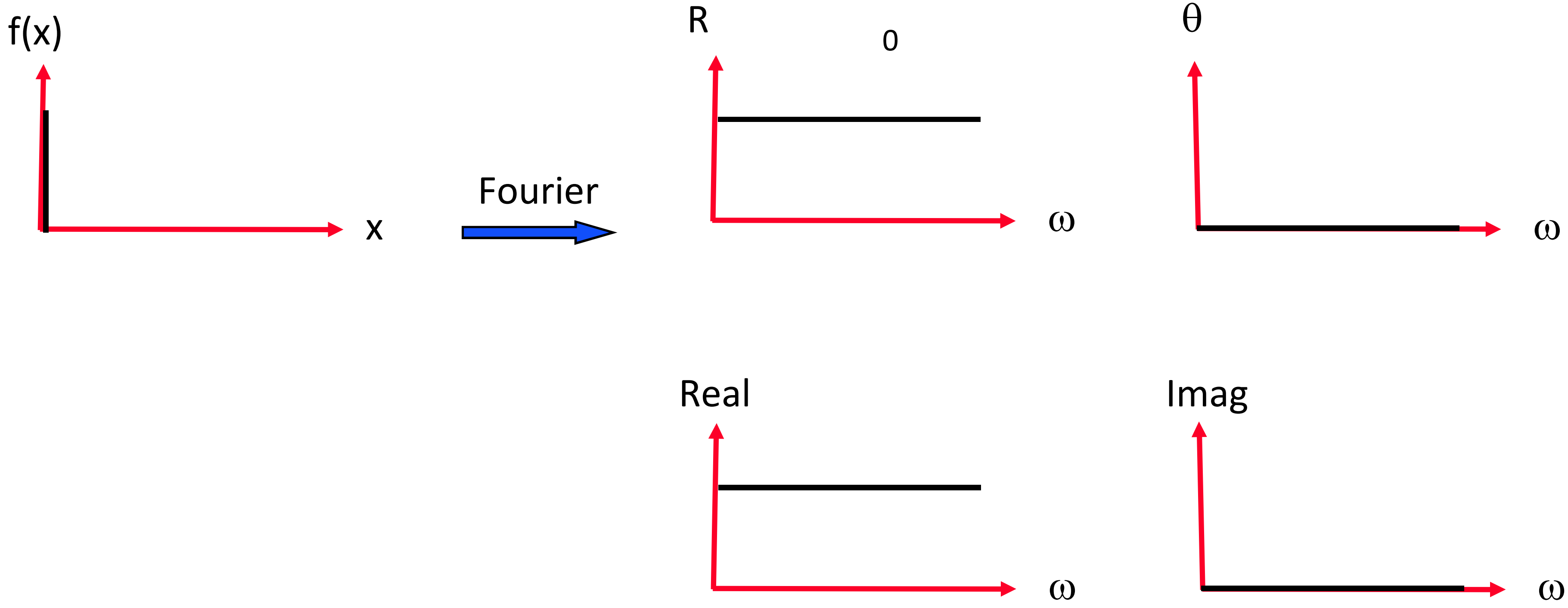
Examples

The Delta Function:

$$f(x) = \delta(x)$$

$$\left(\lim_{x \to 0} \delta(x) = \infty \;;\; \int \delta(x)\,dx = 1 \quad \int g(x)\delta(x - x_0)\,dx = g(x_0) \right)$$

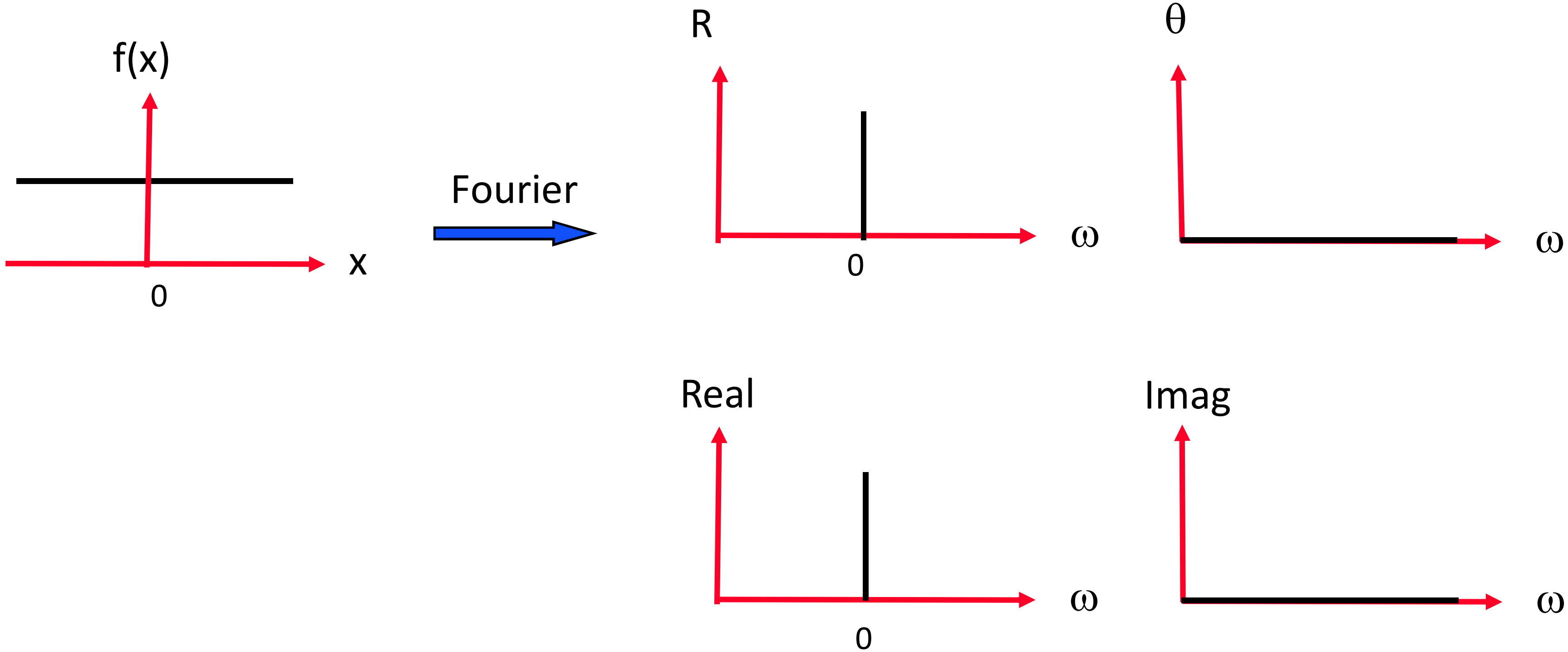
$$F(\omega) = \int_{-\infty}^{\infty} \delta(x) \cdot e^{-i2\pi\omega x}\,dx = 1$$



The Constant Function:

$$f(x) = 1$$

$$F(\omega) = \int_{-\infty}^{\infty} e^{-i2\pi\omega x}\, dx = \delta(\omega)$$

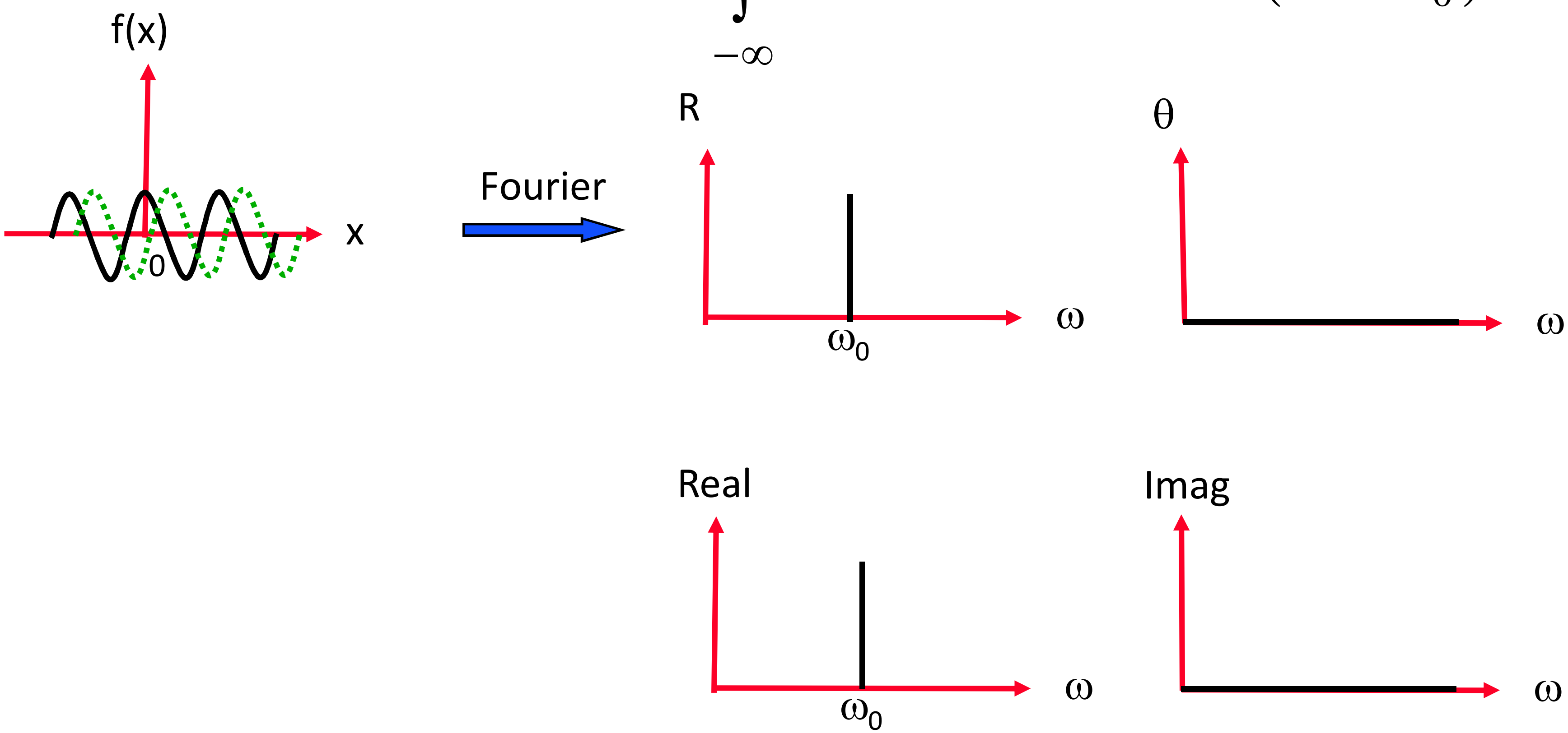


A Basis Function:

$$f(x) = e^{i2\pi\omega_0 x}$$

$$F(\omega) = \int_{-\infty}^{\infty} e^{i2\pi\omega_0 x} e^{-i2\pi\omega x}\, dx$$

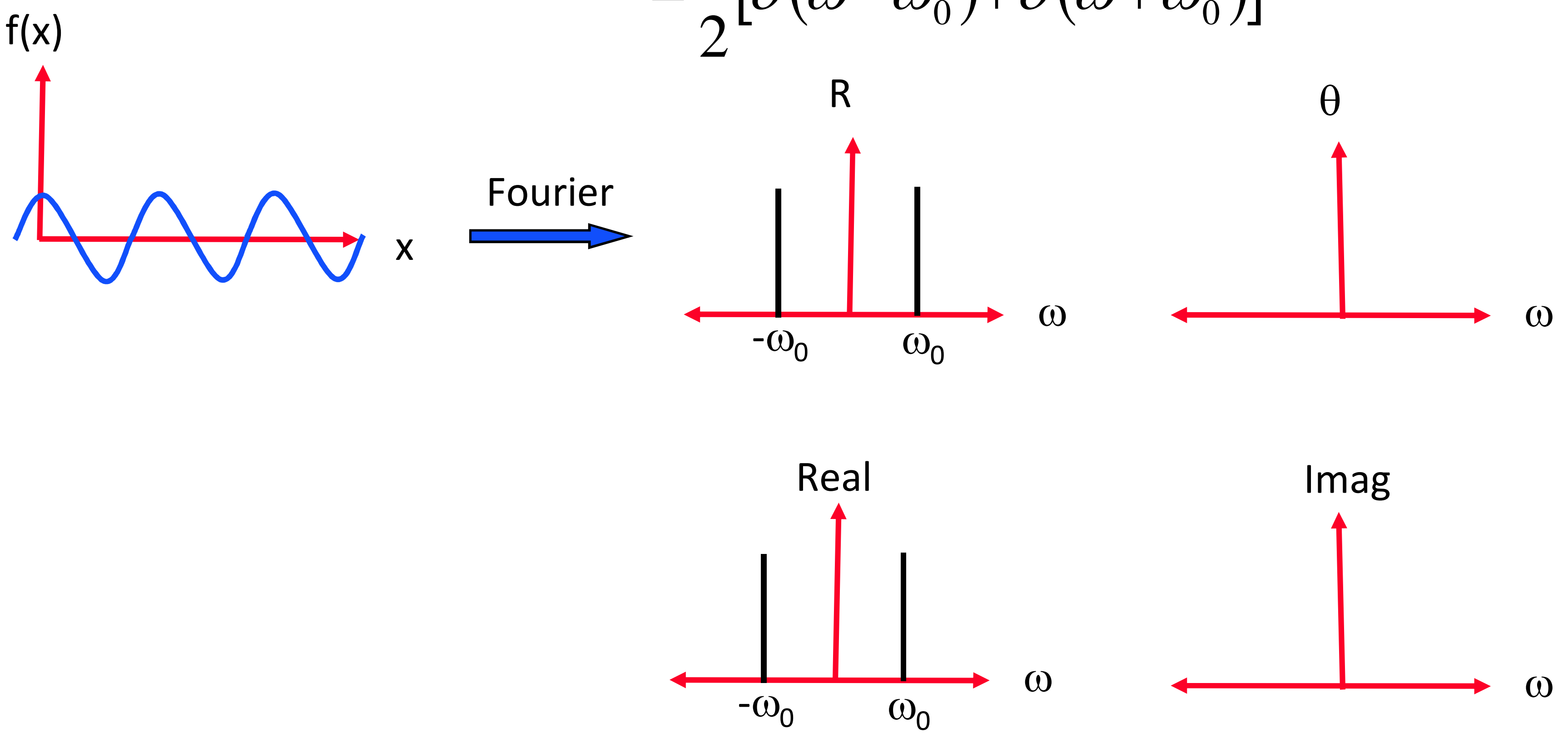
$$= \int_{-\infty}^{\infty} e^{-i2\pi(\omega-\omega_0)x}\, dx = \delta(\omega - \omega_0)$$



The CosineFunction: $f(x)=\cos\left(2\pi\omega_0 x\right)$

$$F(\omega)=\int_{-\infty}^{\infty}\frac{1}{2}\left(e^{i2\pi\omega_0 x}+e^{-i2\pi\omega_0 x}\right)\cdot e^{-i2\pi\omega x}\,dx=$$

$$=\frac{1}{2}\left[\delta\left(\omega-\omega_0\right)+\delta\left(\omega+\omega_0\right)\right]$$

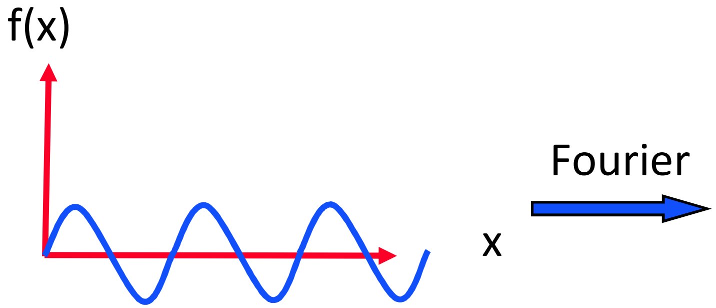


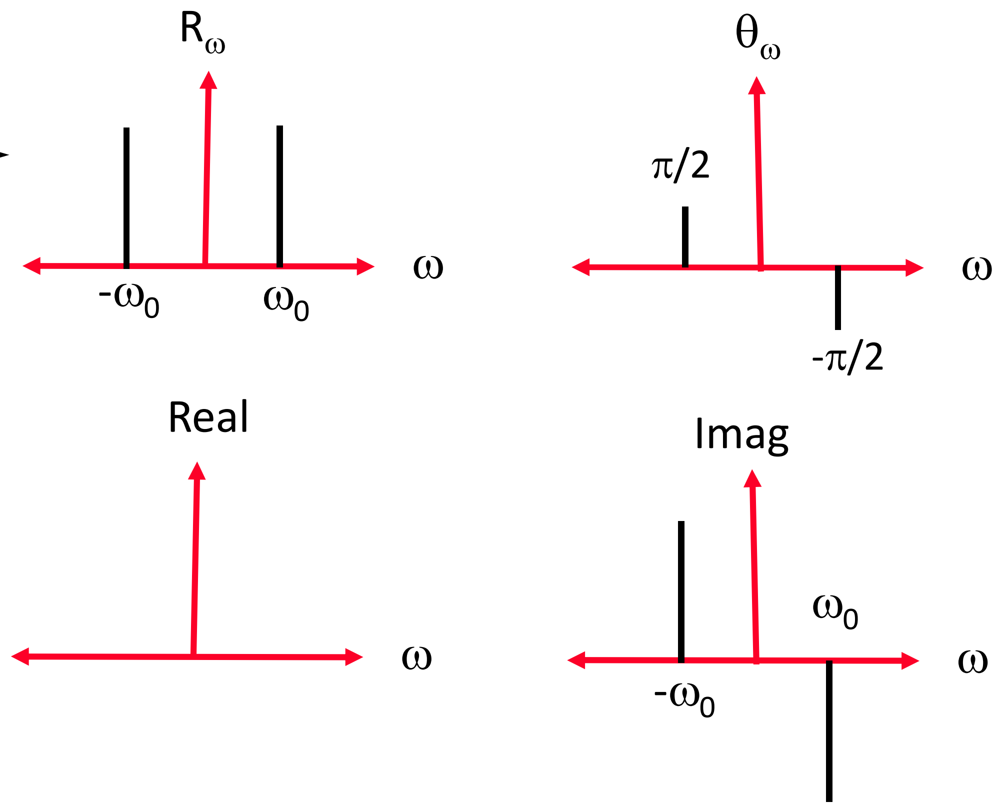
The Sine Function:

$$f(x) = \sin(2\pi\omega_0 x)$$

$$F(\omega) = \int_{-\infty}^{\infty} \frac{i}{2}\left(e^{-i2\pi\omega_0 x} - e^{i2\pi\omega_0 x}\right) \cdot e^{-i2\pi\omega x}\, dx =$$

$$= \frac{i}{2}\left[\delta(\omega+\omega_0) - \delta(\omega-\omega_0)\right]$$

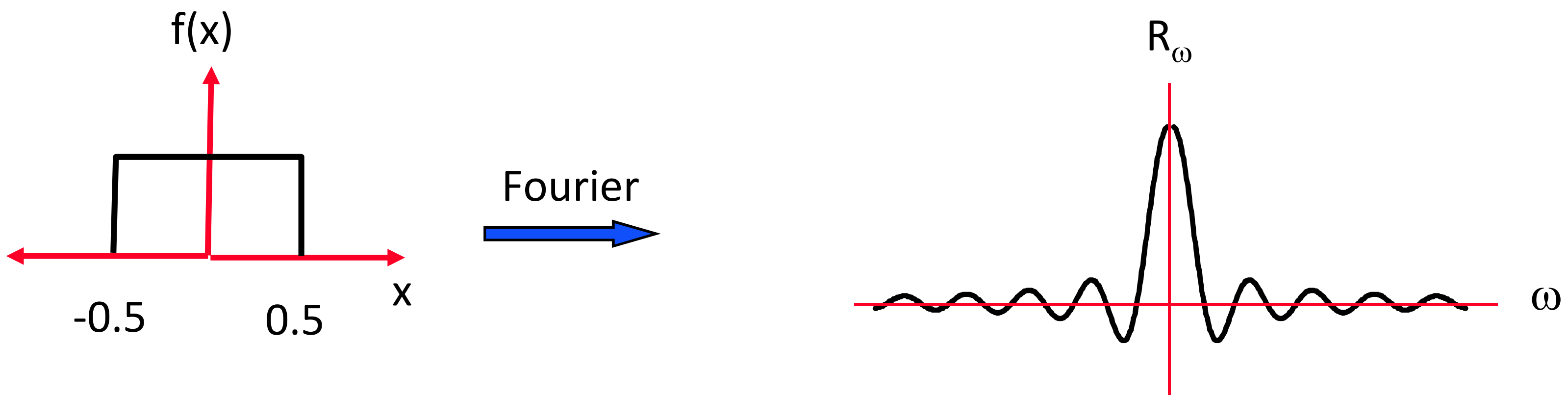




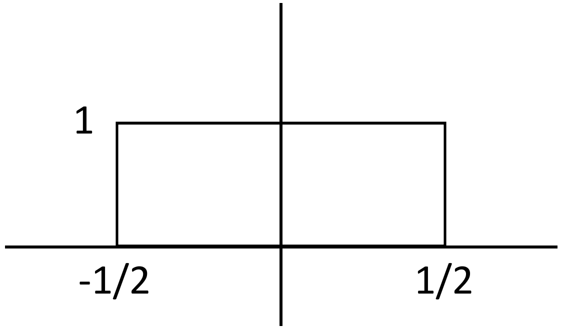
The Window Function (rect):

$$\mathrm{rect}_{1/2}(x) = \begin{cases} 1 & \text{if } |x| < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

$$F(\omega) = \int_{-0.5}^{0.5} e^{-i2\pi\omega x}\, dx = \frac{\sin(\pi\omega)}{\pi\omega} = \mathrm{sinc}(\pi\omega)$$



Proof:



$$f(x) = \mathrm{rect}_{1/2}(x) = \begin{cases} 1 & |x| \leq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

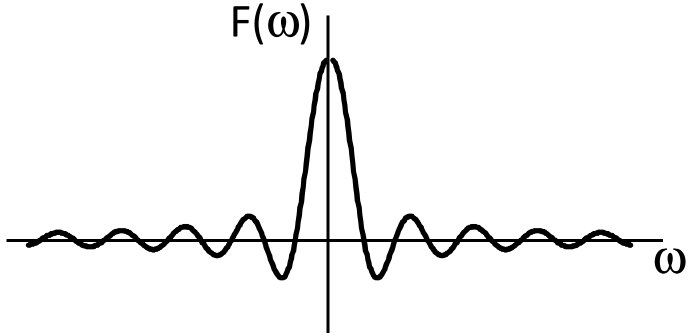
$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i \omega x} dx = \int_{-1/2}^{1/2} e^{-2\pi i \omega x} dx$$

$$= \frac{1}{-2\pi i \omega} \left[e^{-2\pi i \omega x} \right]_{-1/2}^{1/2}$$

$$= \frac{1}{-2\pi i \omega} \left[e^{-\pi i \omega} - e^{\pi i \omega} \right]$$

$$= \frac{1}{-2\pi i \omega} \left[\cancel{\cos(\pi\omega)} - i\sin(\pi\omega) - \cancel{\cos(\pi\omega)} - i\sin(\pi\omega) \right]$$

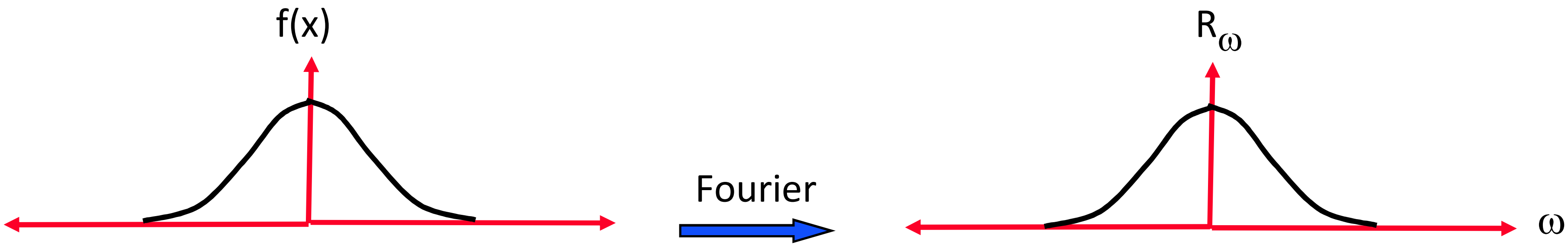
$$= \frac{\sin(\pi\omega)}{\pi\omega} = \boxed{\mathrm{SINC}(\omega)}$$



The Gaussian Function:

$$f(x)=e^{-\pi x^2}$$

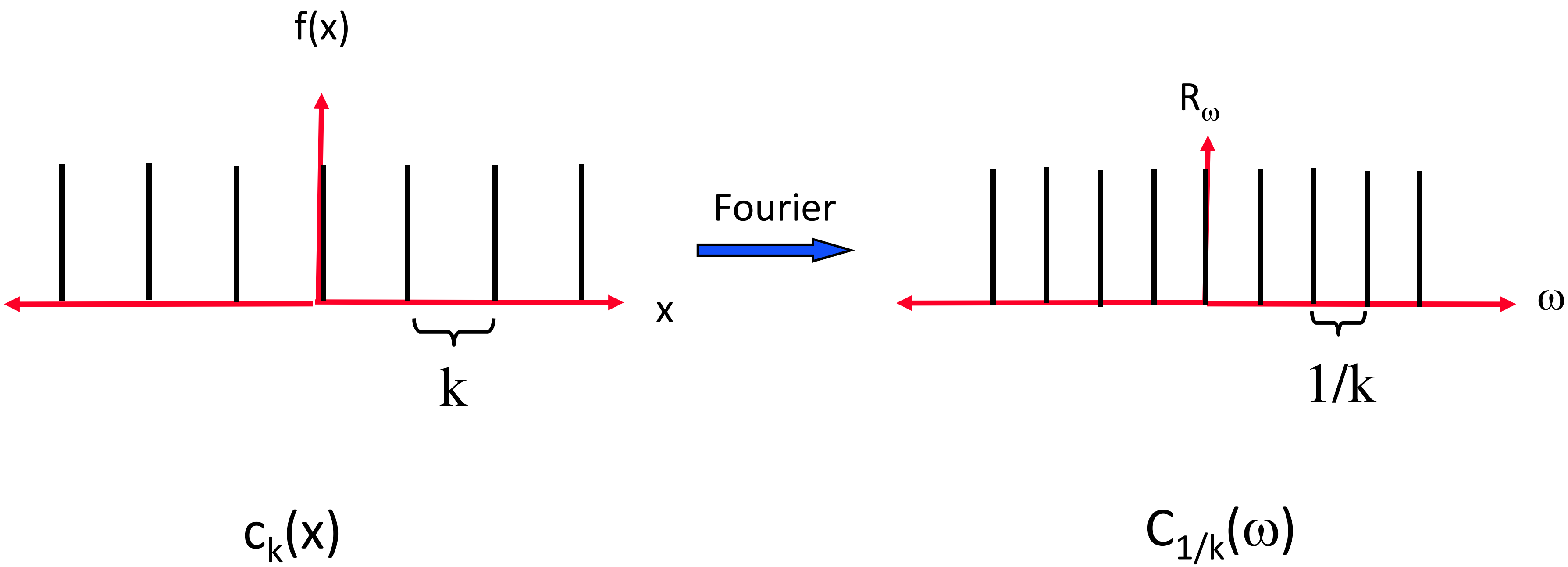
$$F(\omega)=e^{-\pi\omega^2}$$



The Comb Function:

$$c_k(x) = \delta(x \mod k)$$

$$\tilde{F}\{c_k\} = \delta\left(\omega \mod \frac{1}{k}\right) = C_{1/k}(\omega)$$



Fourier Transform – Properties

- Homogeneity:

$$\tilde{F}[\alpha f] = \alpha \tilde{F}[f]$$

- Distributive (additivity):

$$\tilde{F}[f_1 + f_2] = \tilde{F}[f_1] + \tilde{F}[f_2]$$

Linearity

- DC (average):

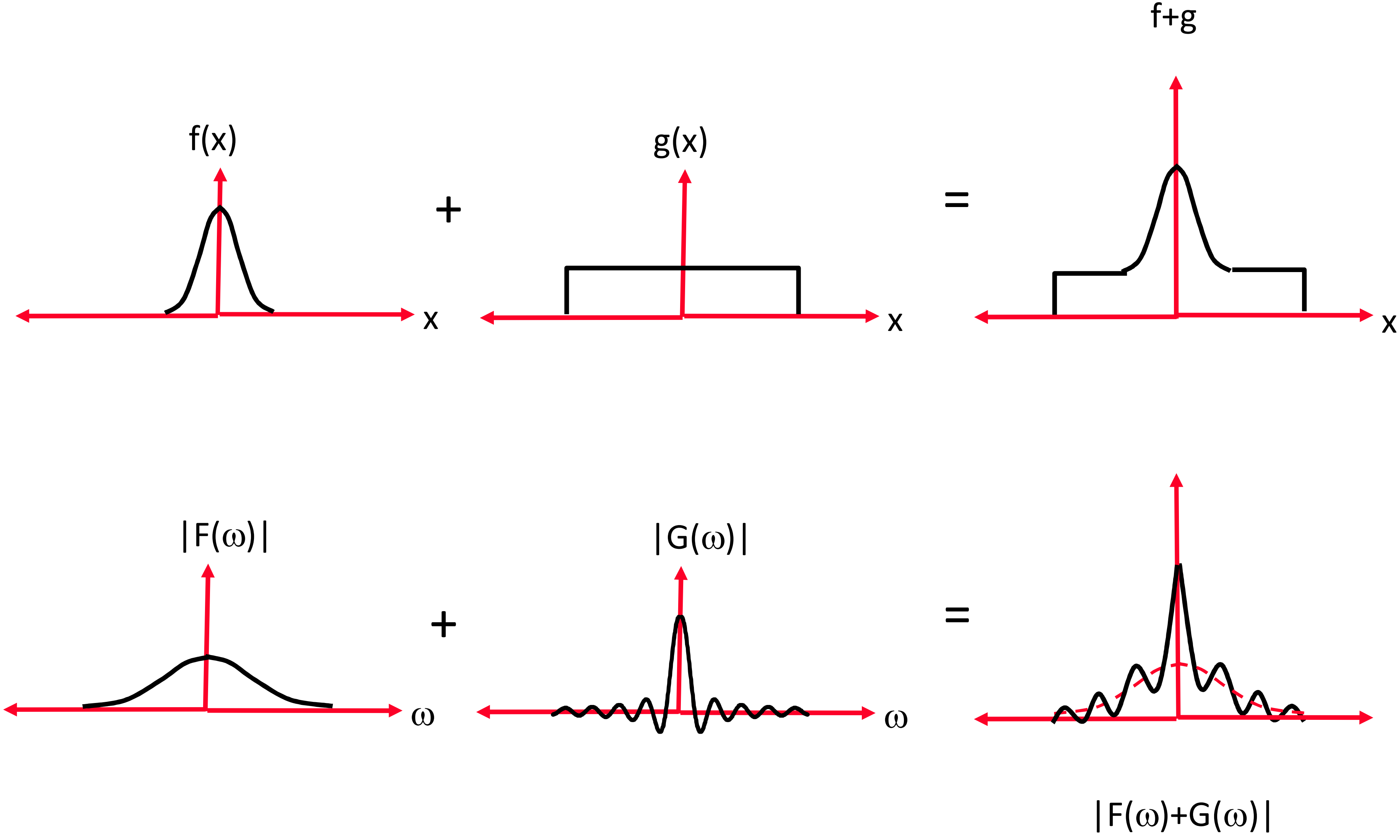
$$F(0,0) = \sum_x \sum_y f(x,y) e^0$$

- Parseval

$$\sum_x \sum_y \| f(x,y) \|^2 = \sum_u \sum_v \| F(u,v) \|^2$$

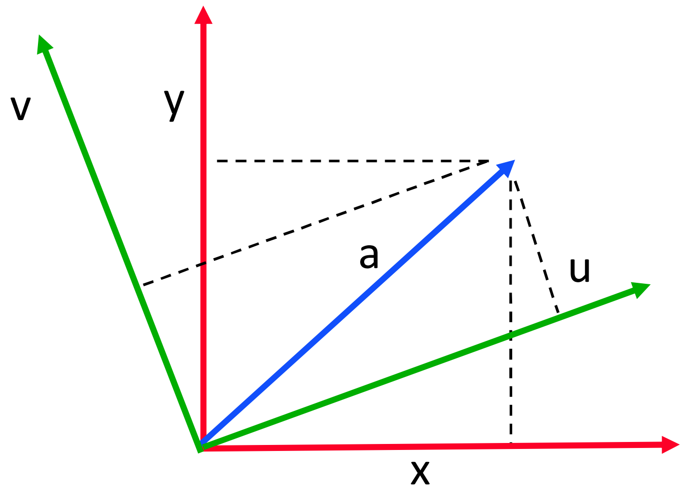
Distributive:

$$\tilde{F}\{f+g\}=\tilde{F}\{f\}+\tilde{F}\{g\}$$



Parseval's Theorem:

$$\sum_x \sum_y \left\| f(x,y) \right\|^2 = \sum_u \sum_v \left\| F(u,v) \right\|^2$$



Fourier Transform – Properties

- Symmetric:

  If f(x,y) is real then,

$$F(u,v) = F^*(-u,-v) \quad \text{thus} \quad |F(u,v)| = |F(-u,-v)|$$

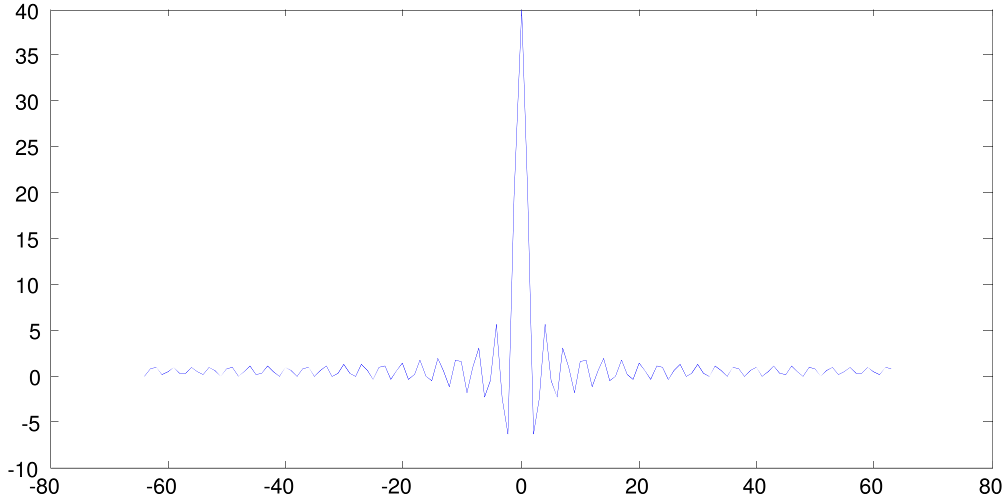
- Cyclic:

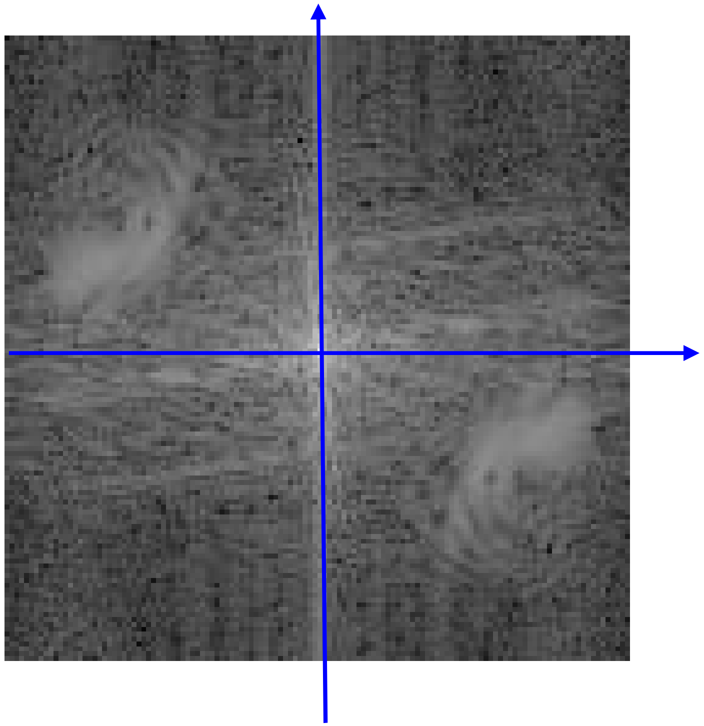
  if f(x,y) is discrete

$$F(u,v) = F(u+N,v) = F(u,v+M) = F(u+N,v+M)$$

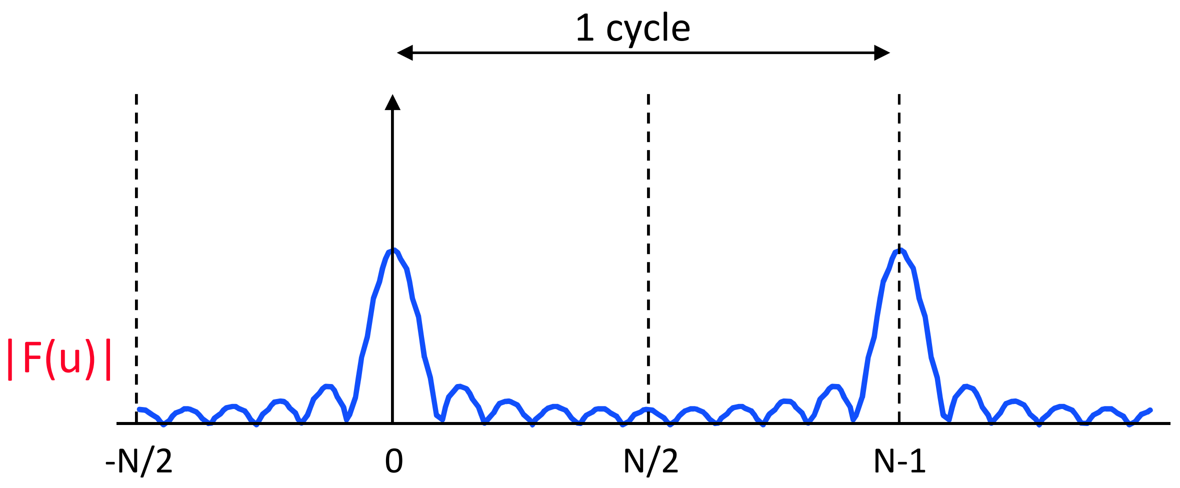
Symmetry of FT (for real signals):

$$F(u,v) = F^*(-u,-v)$$





Cyclic and Symmetry of FT :

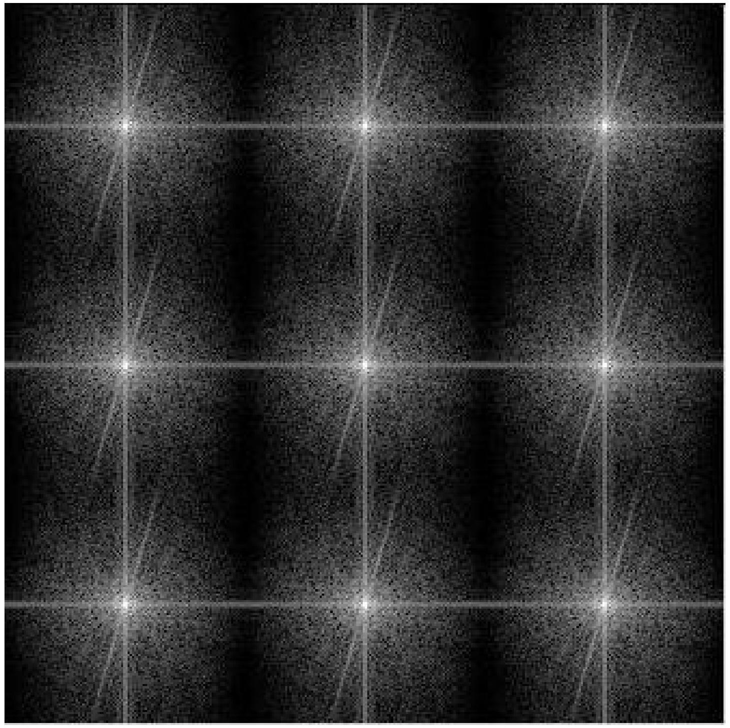


Due to replicas: $F(k)=F(N+k)$

Due to symmetry: $F(k)=F^*(-k)=F^*(N-k)$

Cyclic and Symmetry of FT :

In 2D: $$F(u,v) = F(u+N,v) = F(u,v+M) = F(u+N,v+M)$$



Fourier Transform – Properties

Seperability:

$$F(u,v) = \sum_x \sum_y f(x,y) e^{-2\pi i\left(\frac{ux}{N}+\frac{vy}{M}\right)} =$$

$$= \sum_x \left(\sum_y f(x,y) e^{-2\pi i \frac{vy}{N}} \right) e^{-2\pi i \frac{ux}{N}} = \sum_x F(x,v) e^{-2\pi i \frac{ux}{N}}$$

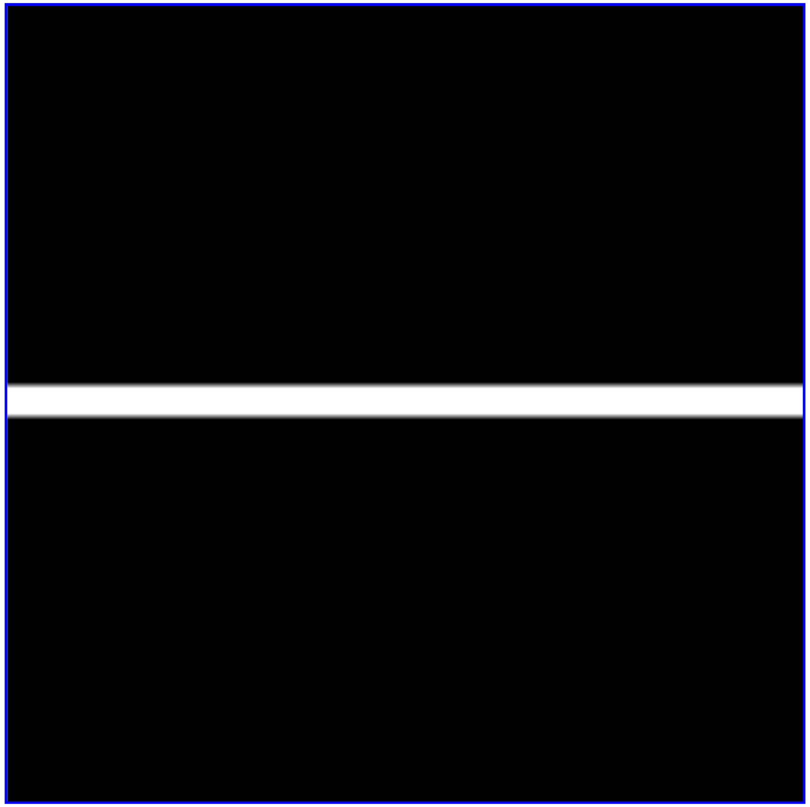
Thus, performing a 2D Fourier Transform is equivalent to performing 2 1D transforms:

1. 1D transform on EACH column of image f(x,y), obtaining F(x,v).
2. 1D transform on EACH row of F(x,v), obtaining F(u,v).

Higher Dimensions: Fourier in any dimension can be performed by applying 1D transform on each dimension.

Example - Seperability:

2D Image



Fourier Spectrum

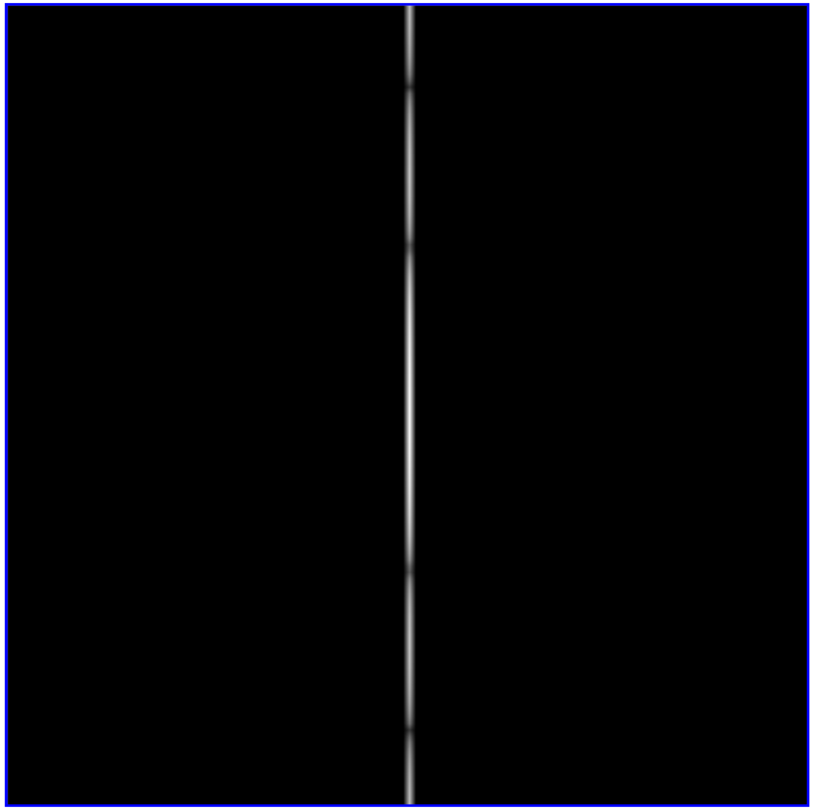


Image Transformations

Translation:

$$\tilde{F}\left[f\left(x-x_0, y-y_0\right)\right] = F\left(u,v\right)e^{-2\pi i\left(\frac{ux_0}{N}+\frac{vy_0}{M}\right)}$$

The Fourier Spectrum remains unchanged under translation:

$$\left|F\left(u,v\right)\right| = \left|F\left(u,v\right)e^{-2\pi i\left(\frac{ux_0}{N}+\frac{vy_0}{M}\right)}\right|$$

Example Translation:

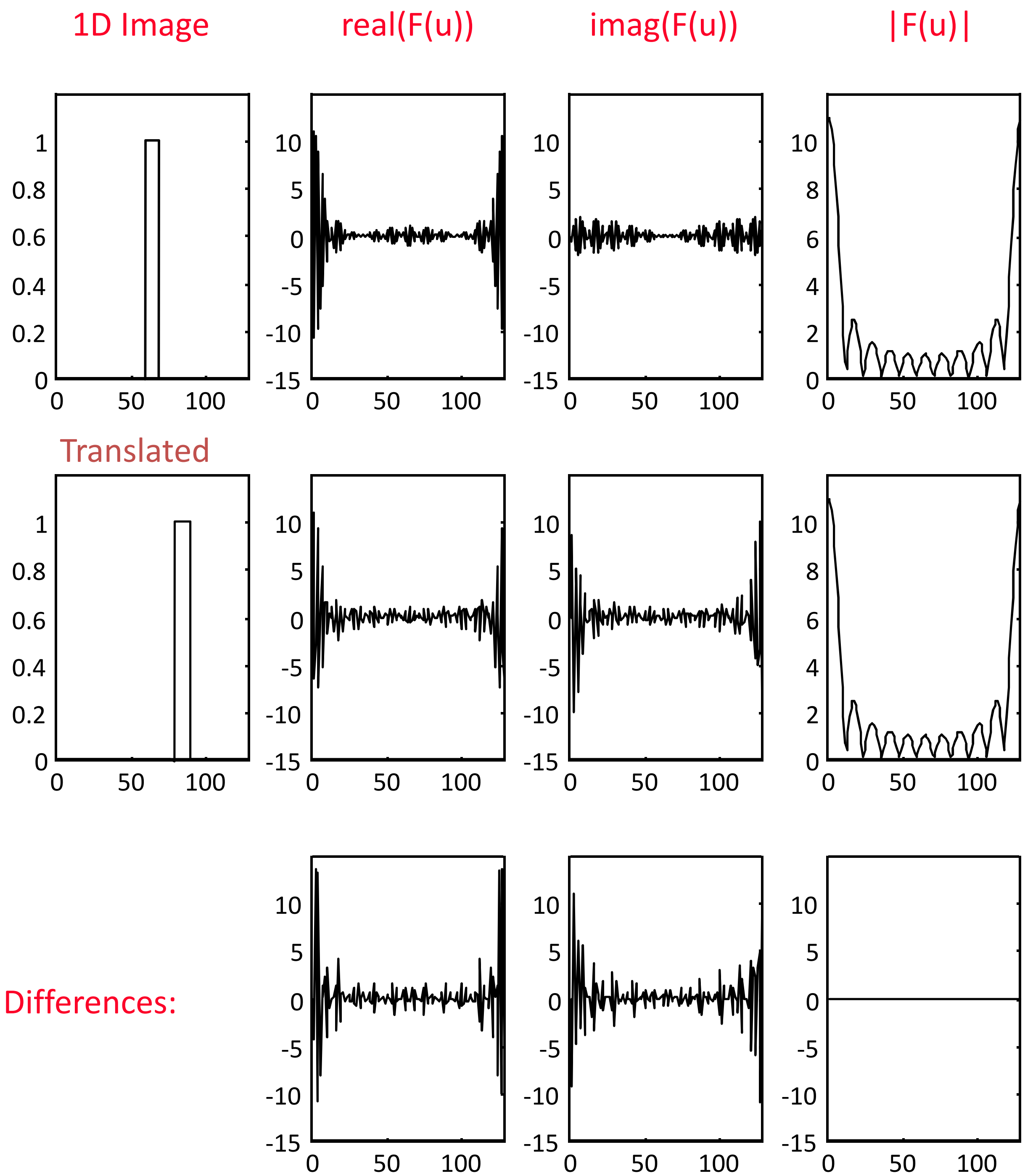


Image Transformations

Scaling:

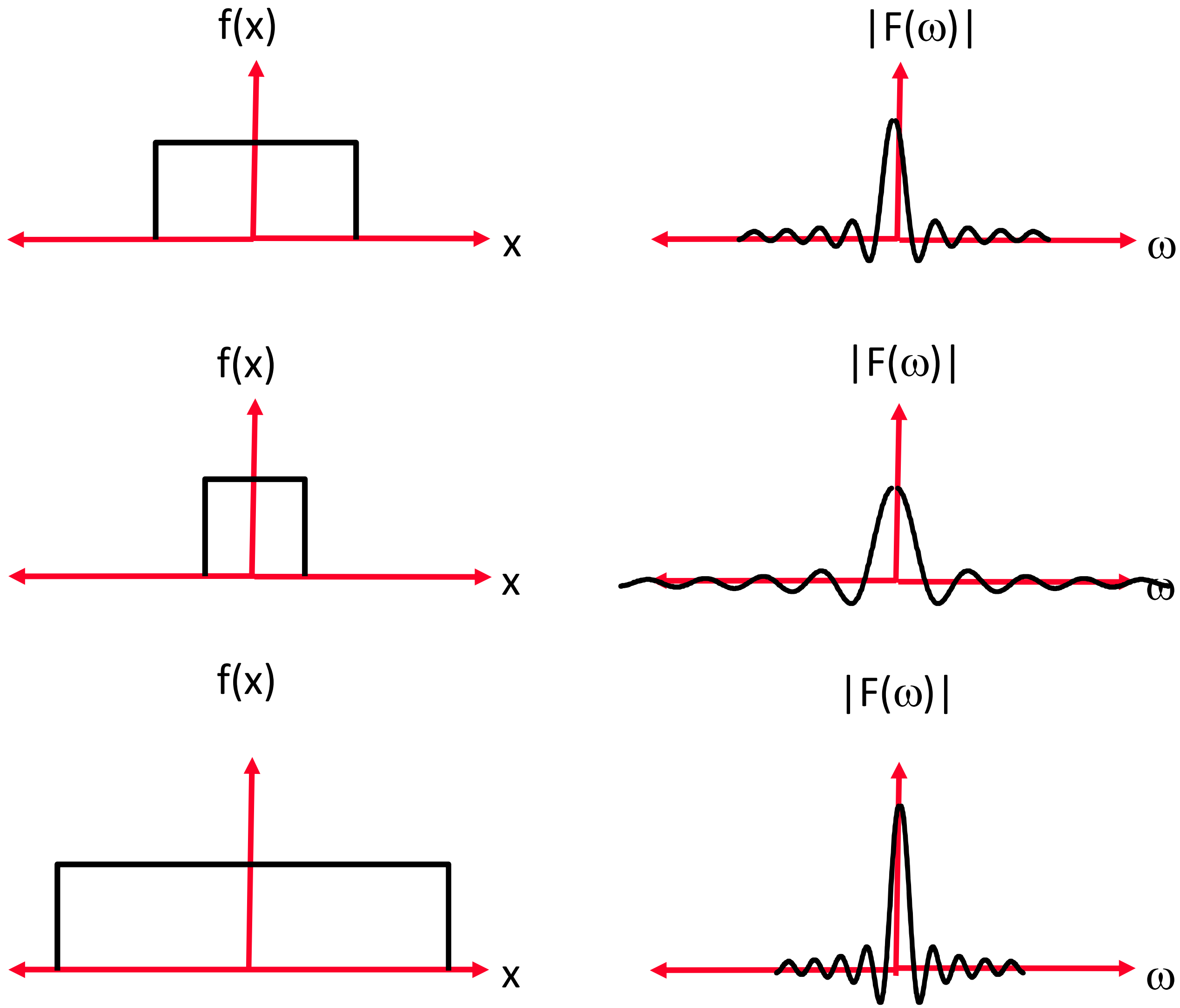
$$\tilde{F}\left[f\left(a\,x, b\,y\right)\right] = \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right)$$

Rotation:

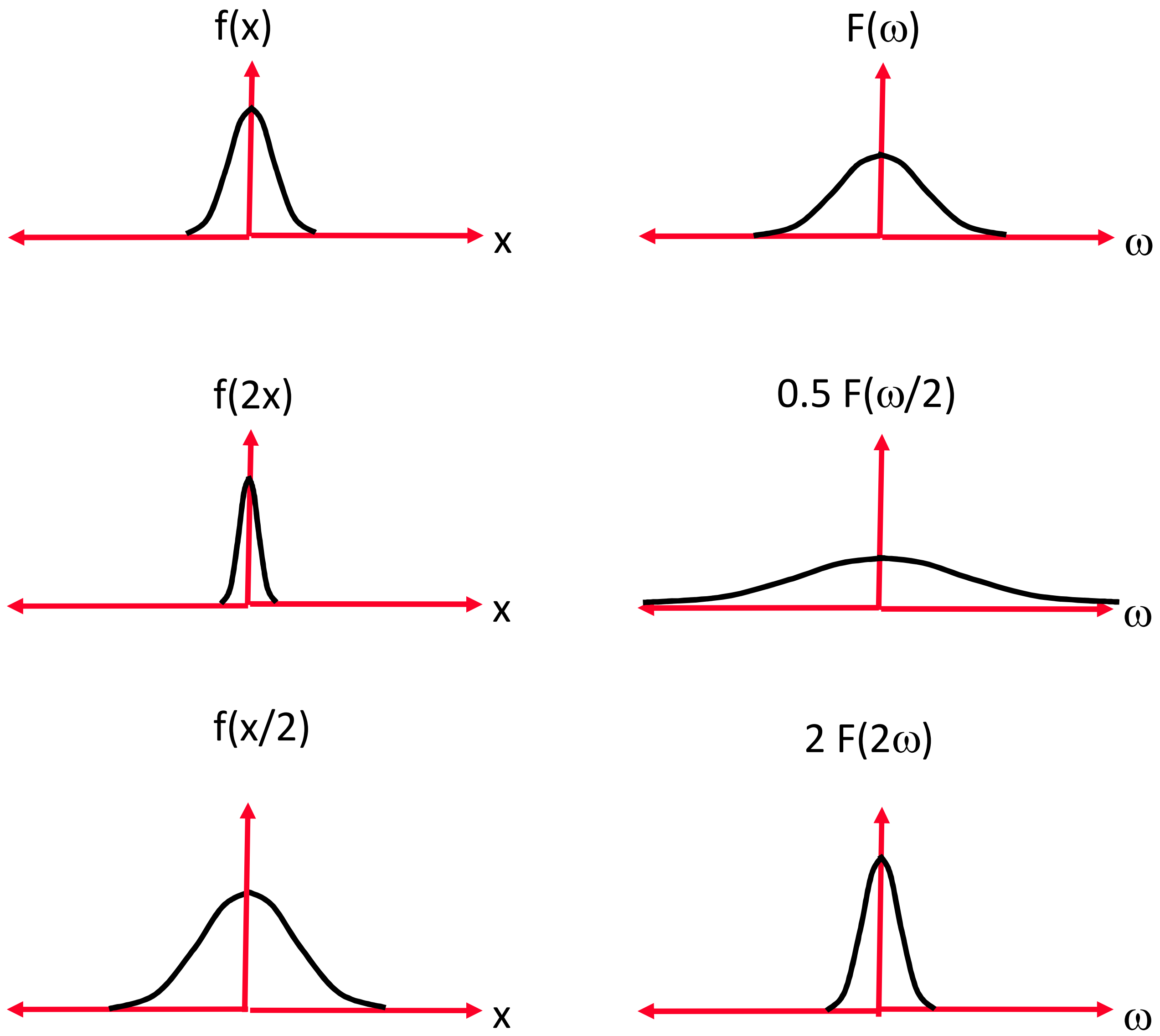
Rotation of f(x,y) by θ $\rightarrow$ rotation of F(u,v) by θ

Change of Scale:

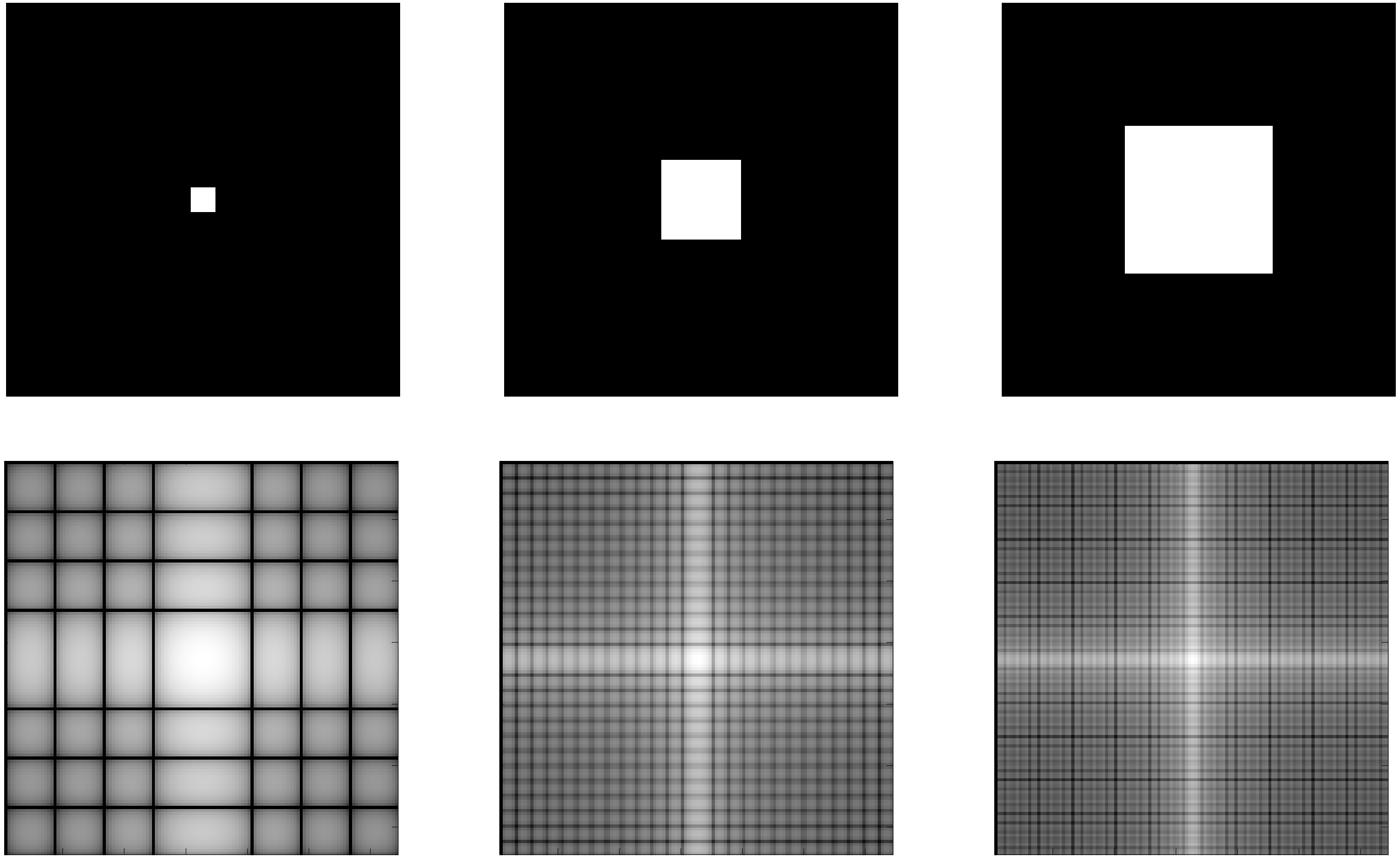
$$\text{if } \tilde{F}\{f(x)\} = F(\omega) \text{ then } \tilde{F}\{f(ax)\} = \frac{1}{|a|} F\left(\frac{\omega}{a}\right)$$



Change of Scale:



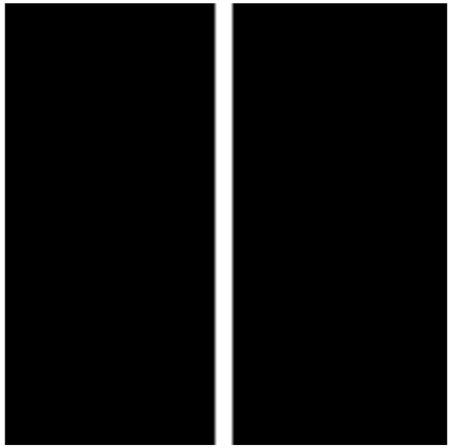
Change of Scale:

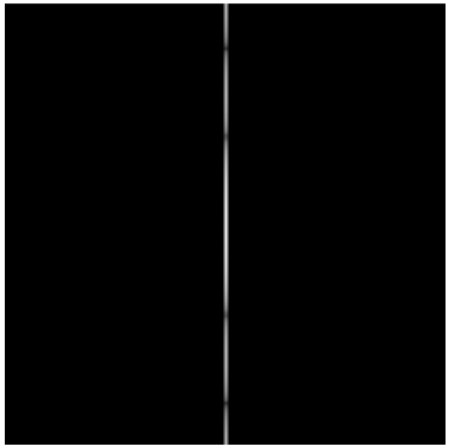


Rotation - Example

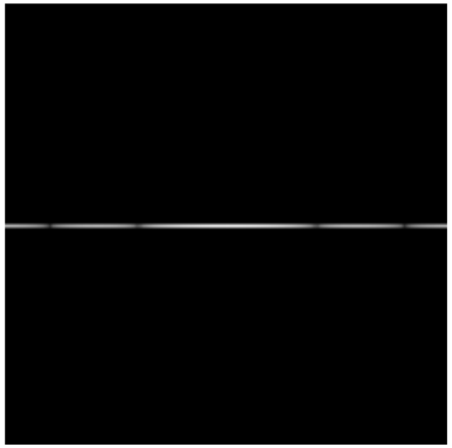
2D Image

2D Image - Rotated





Fourier Spectrum



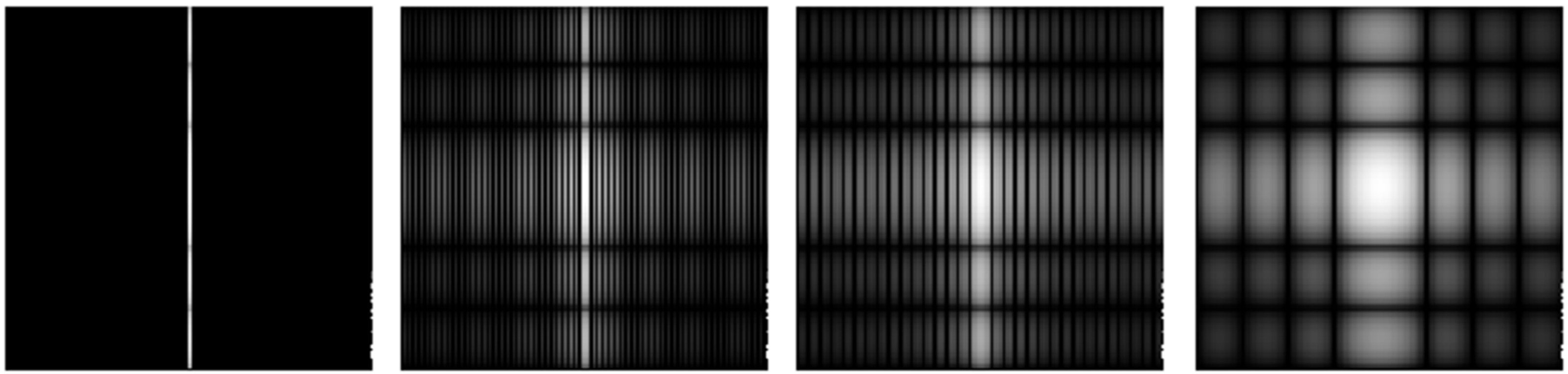
Fourier Spectrum

Example Demo

Fourier Transform – Examples

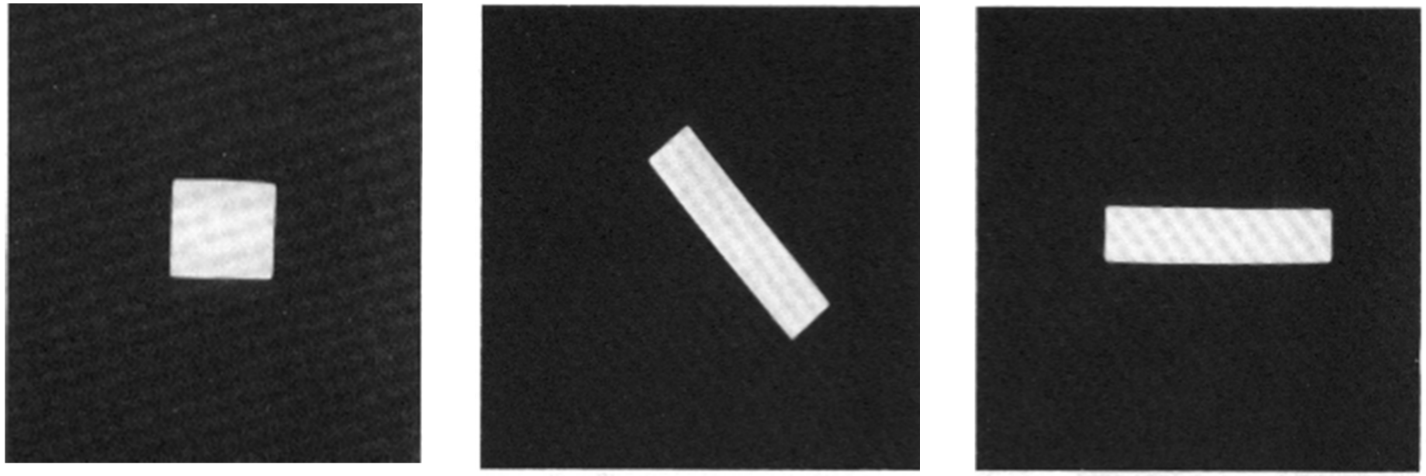
Image Domain

Frequency Domain

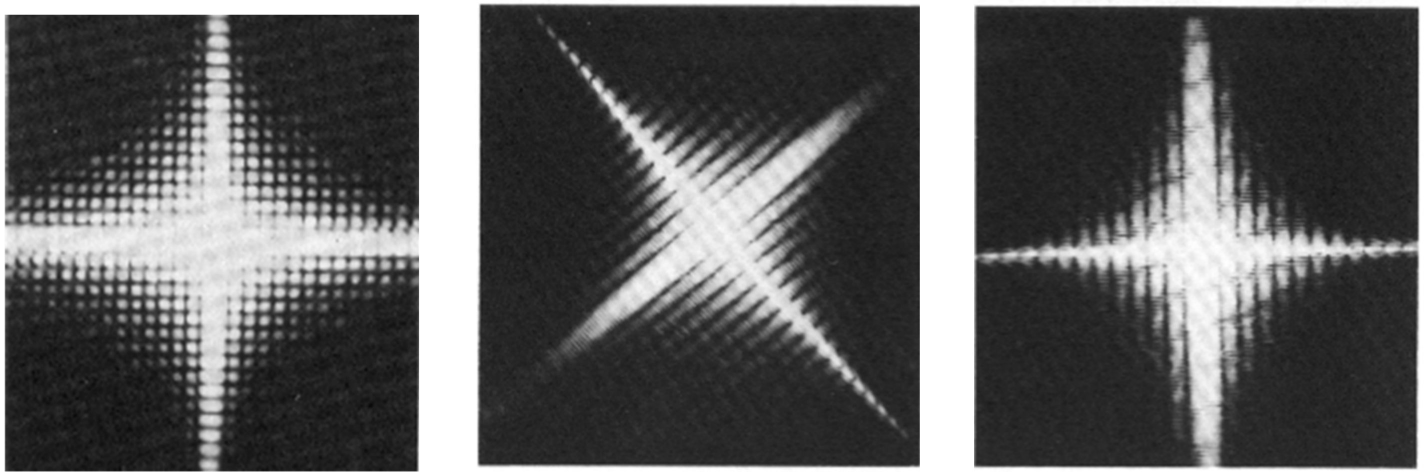


Fourier Transform – Examples

Image Domain

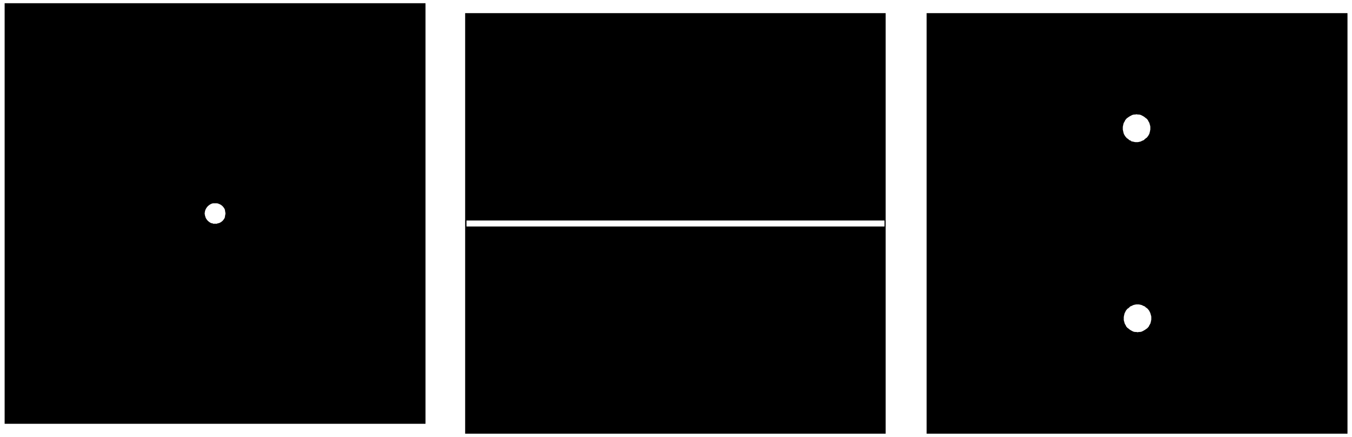


Frequency Domain

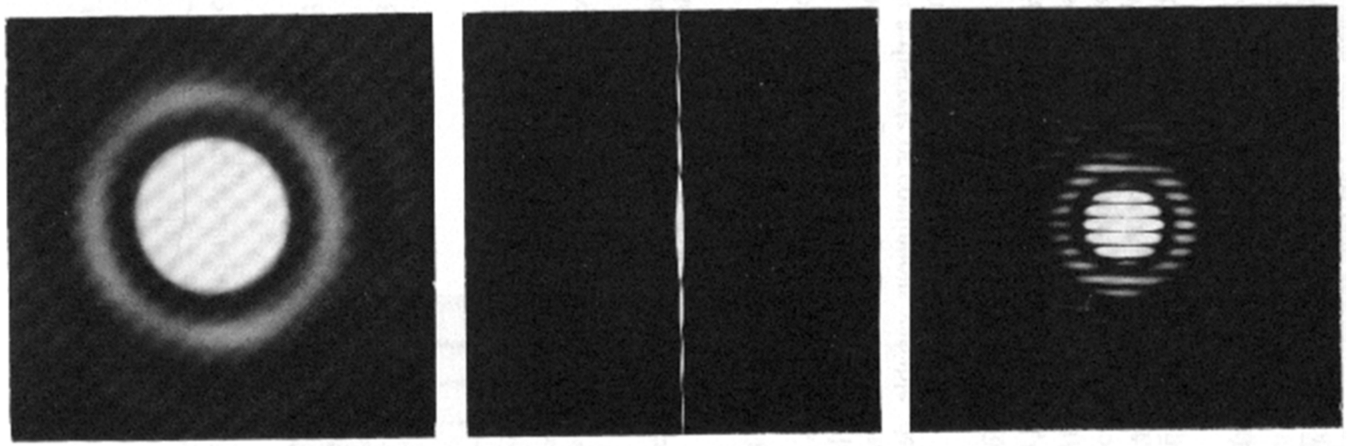


Fourier Transform – Examples

Image Domain

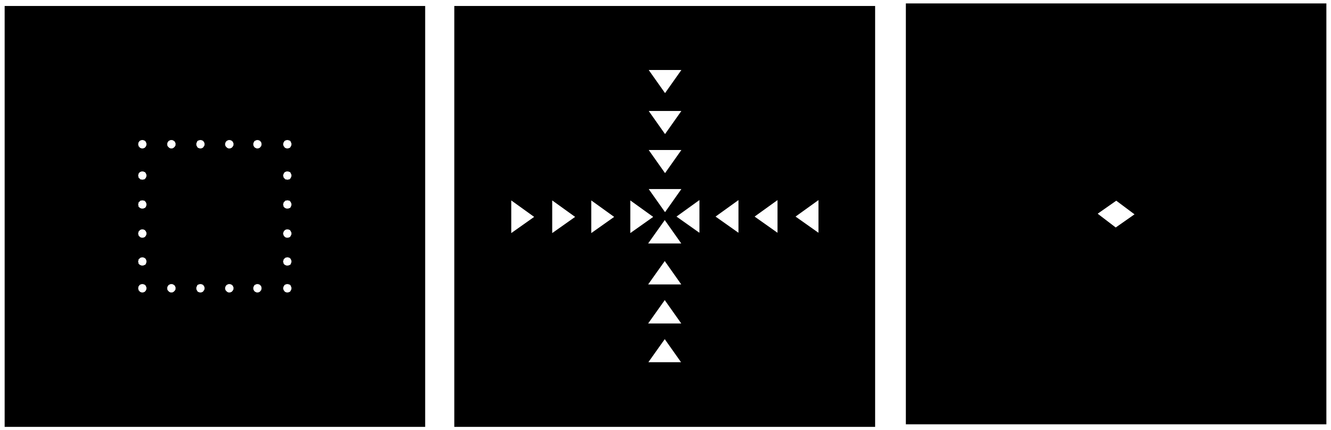


Frequency Domain

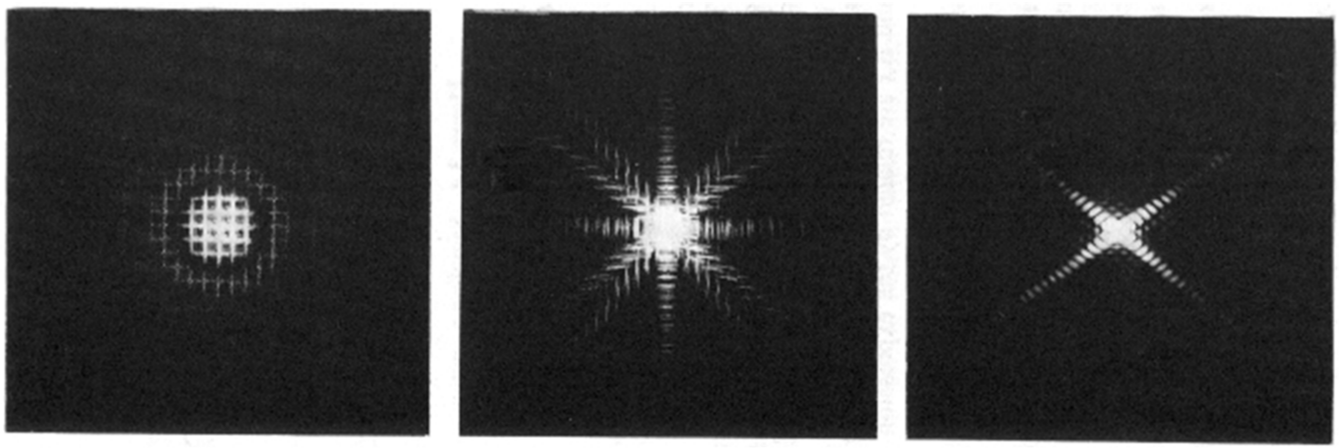


Fourier Transform – Examples

Image Domain



Frequency Domain



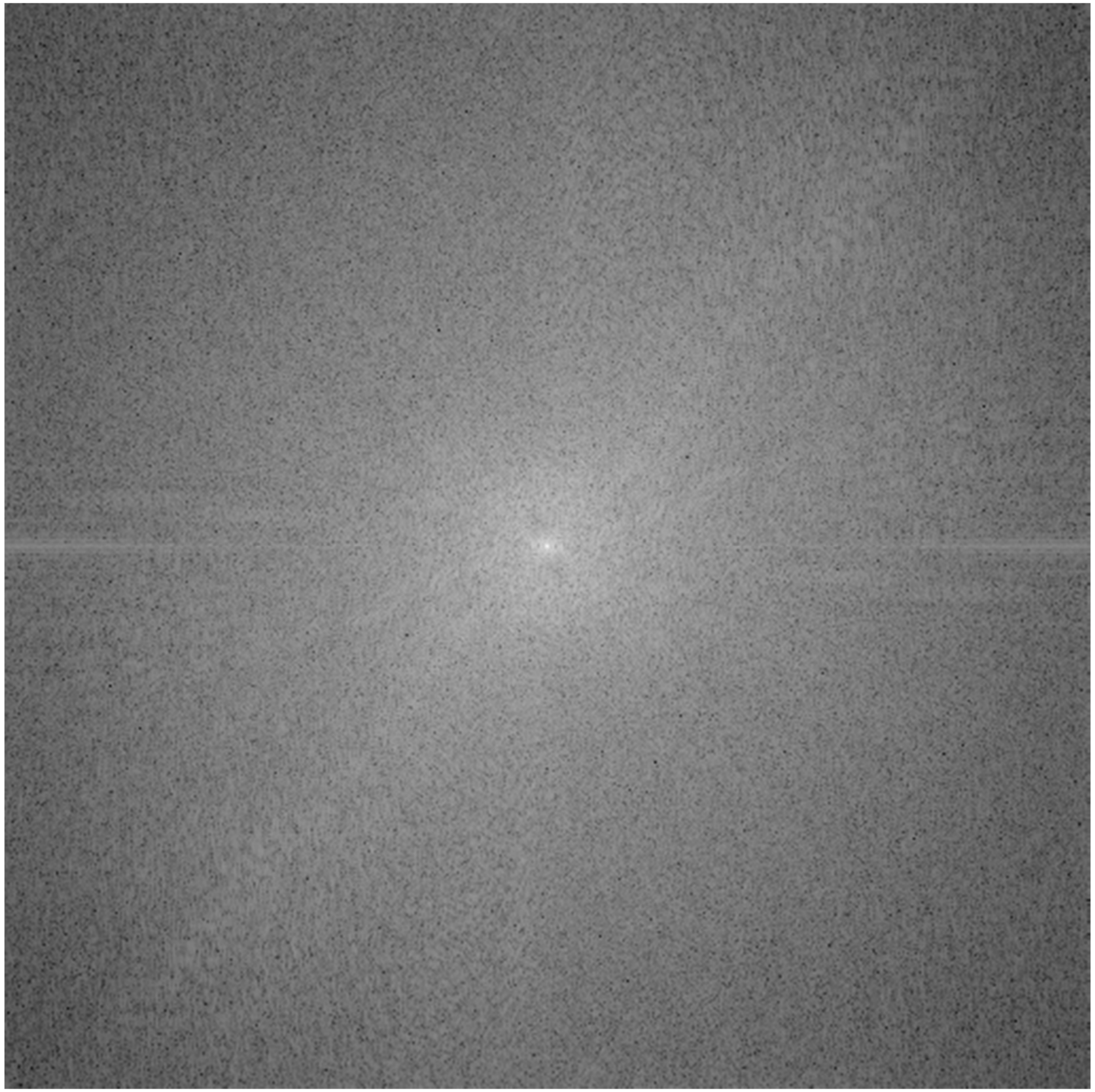
Fourier Transform – Image

- Curious fact
  - all natural images have about the same magnitude transform
  - hence, phase seems to matter, but magnitude largely doesn't
- Demonstration
  - Take two pictures, swap the phase transforms, compute the inverse - what does the result look like?

Slide: Freeman & Durand

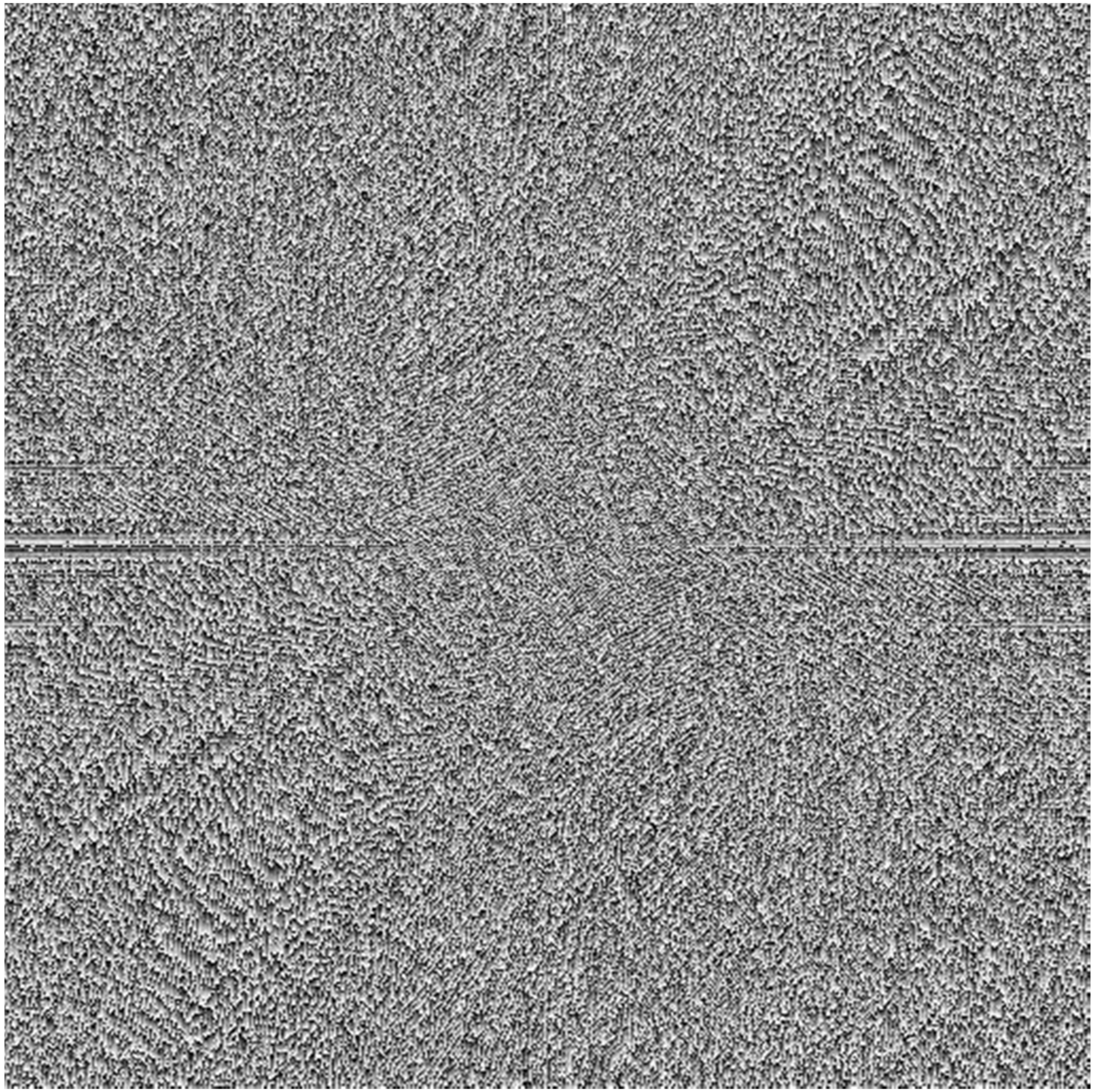
Magnitude transform of cheetah

Slide: Freeman & Durand



Magnitude transform of cheetah

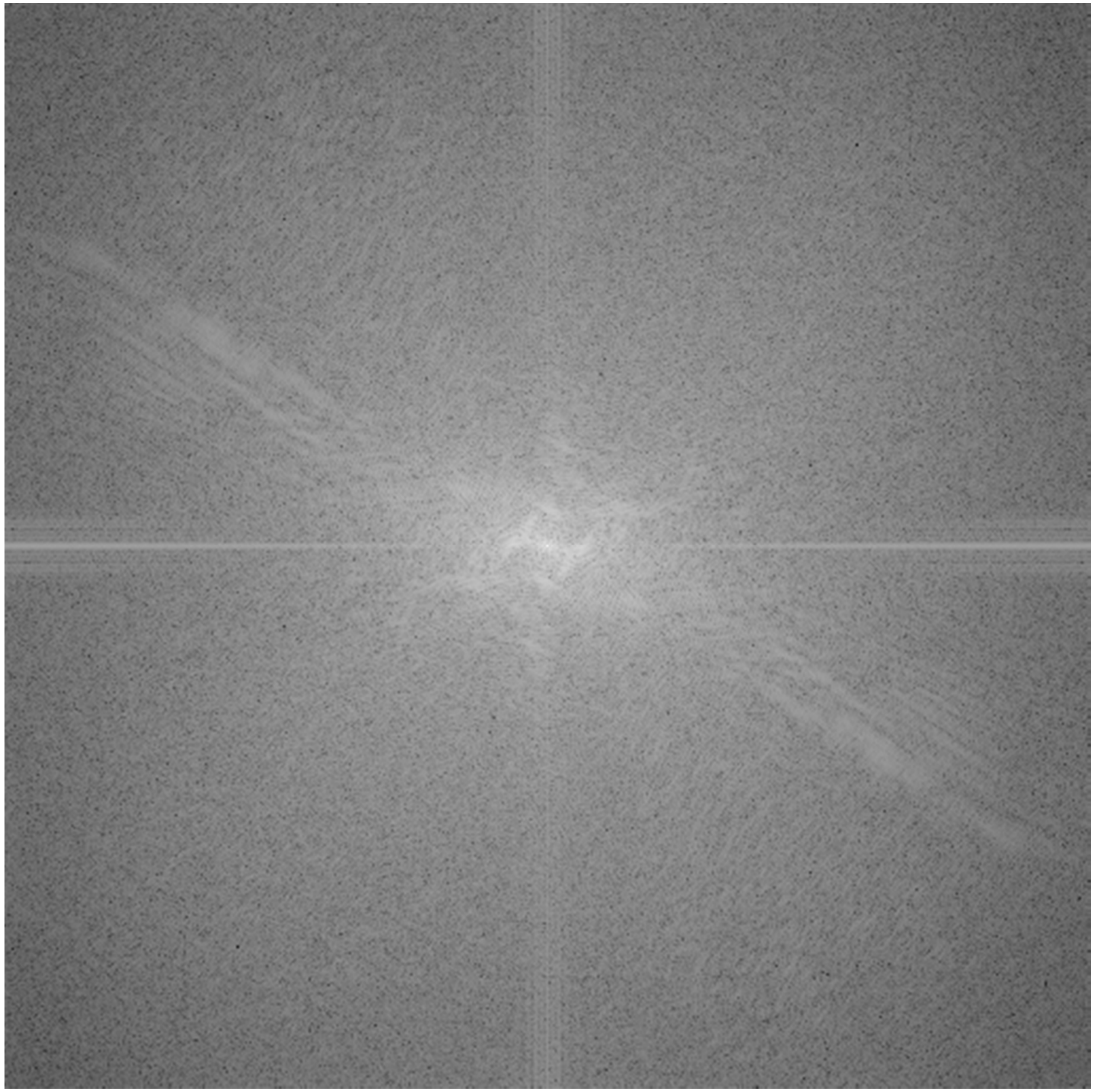
Slide: Freeman & Durand



Phase transform of cheetah

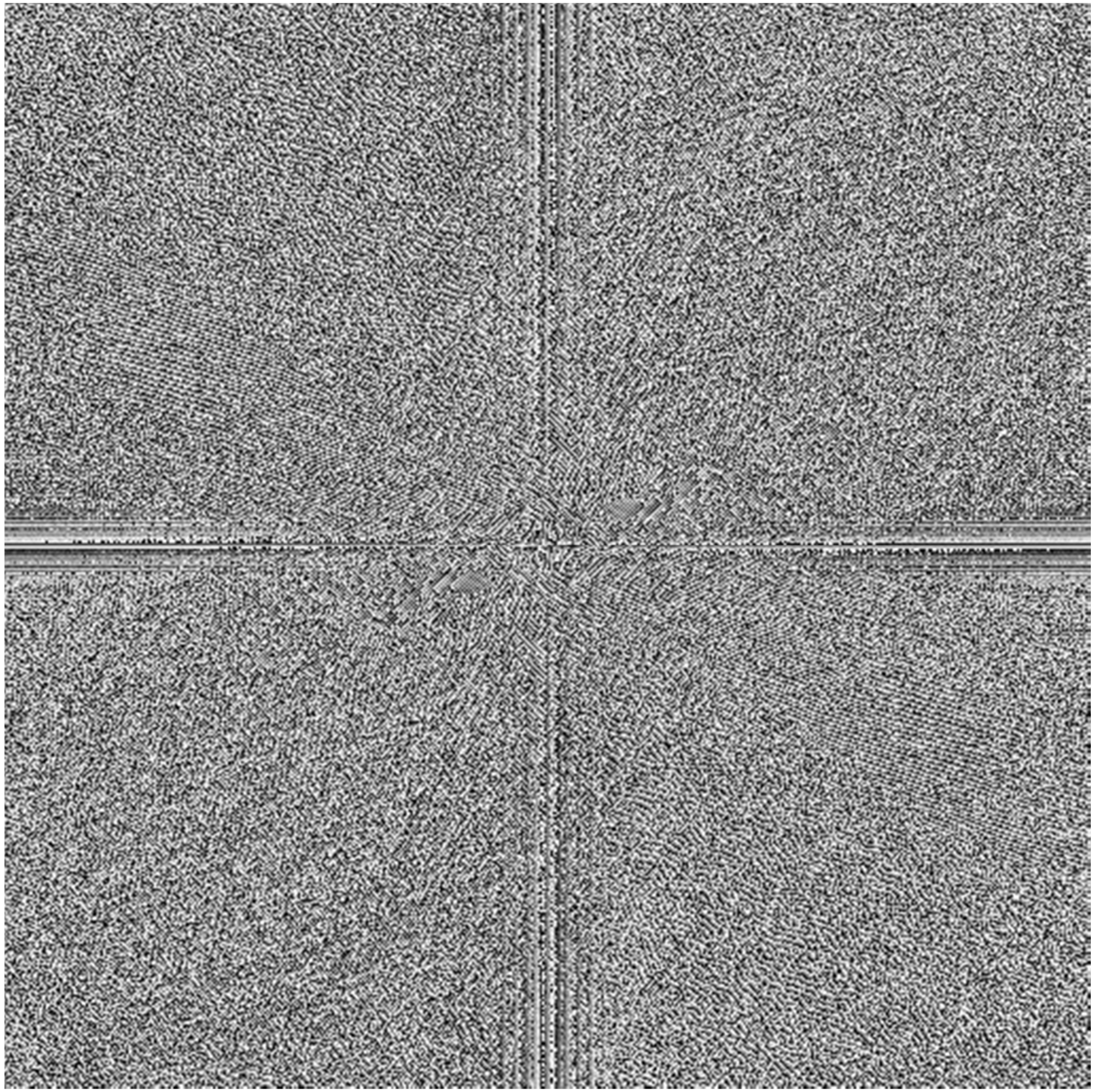
Slide: Freeman & Durand

Slide: Freeman & Durand



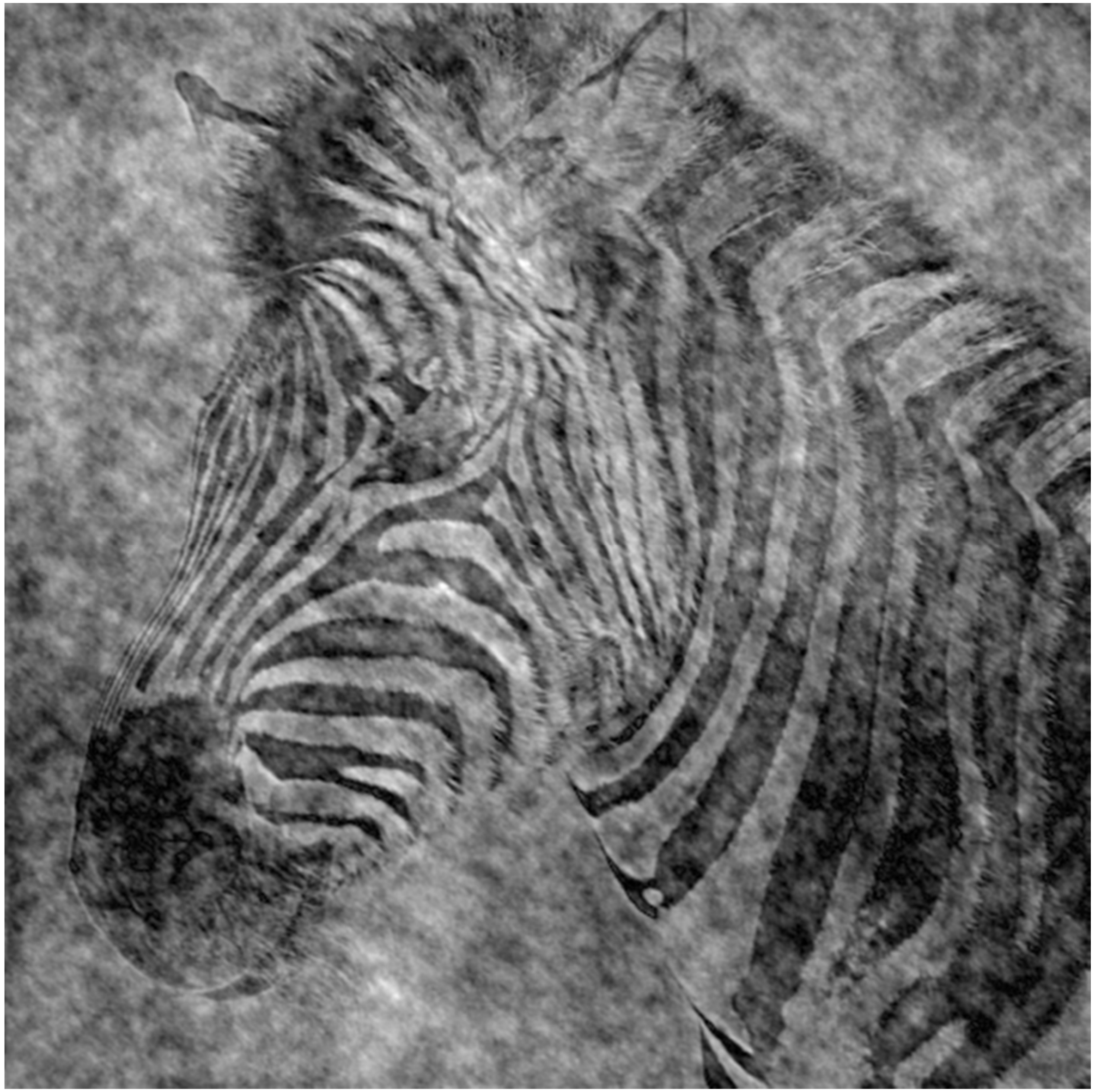
Magnitude transform of zebra

Slide: Freeman & Durand



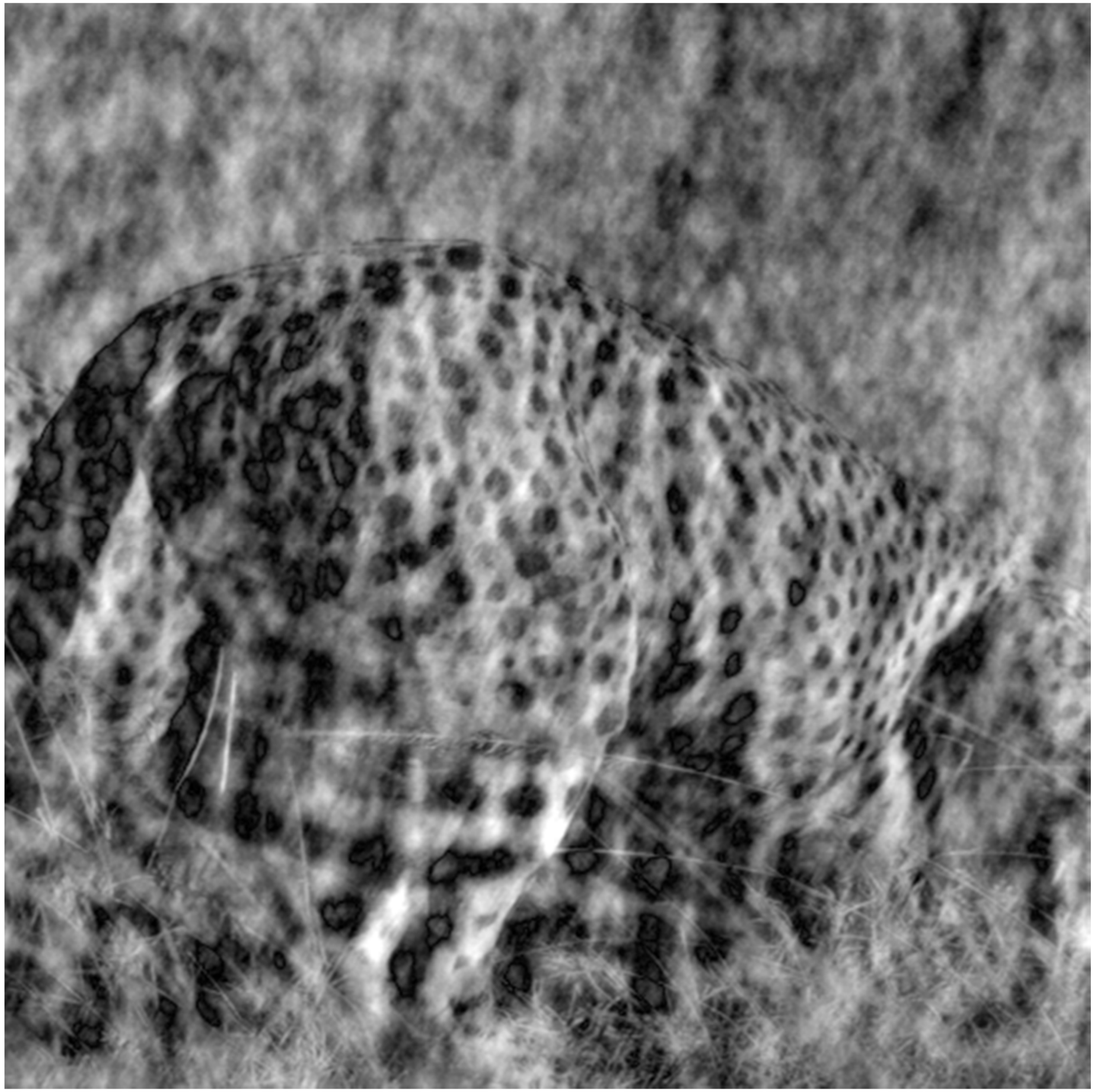
Phase transform of zebra

Slide: Freeman & Durand



Recon: Zebra Phase + Cheetah Magnitude

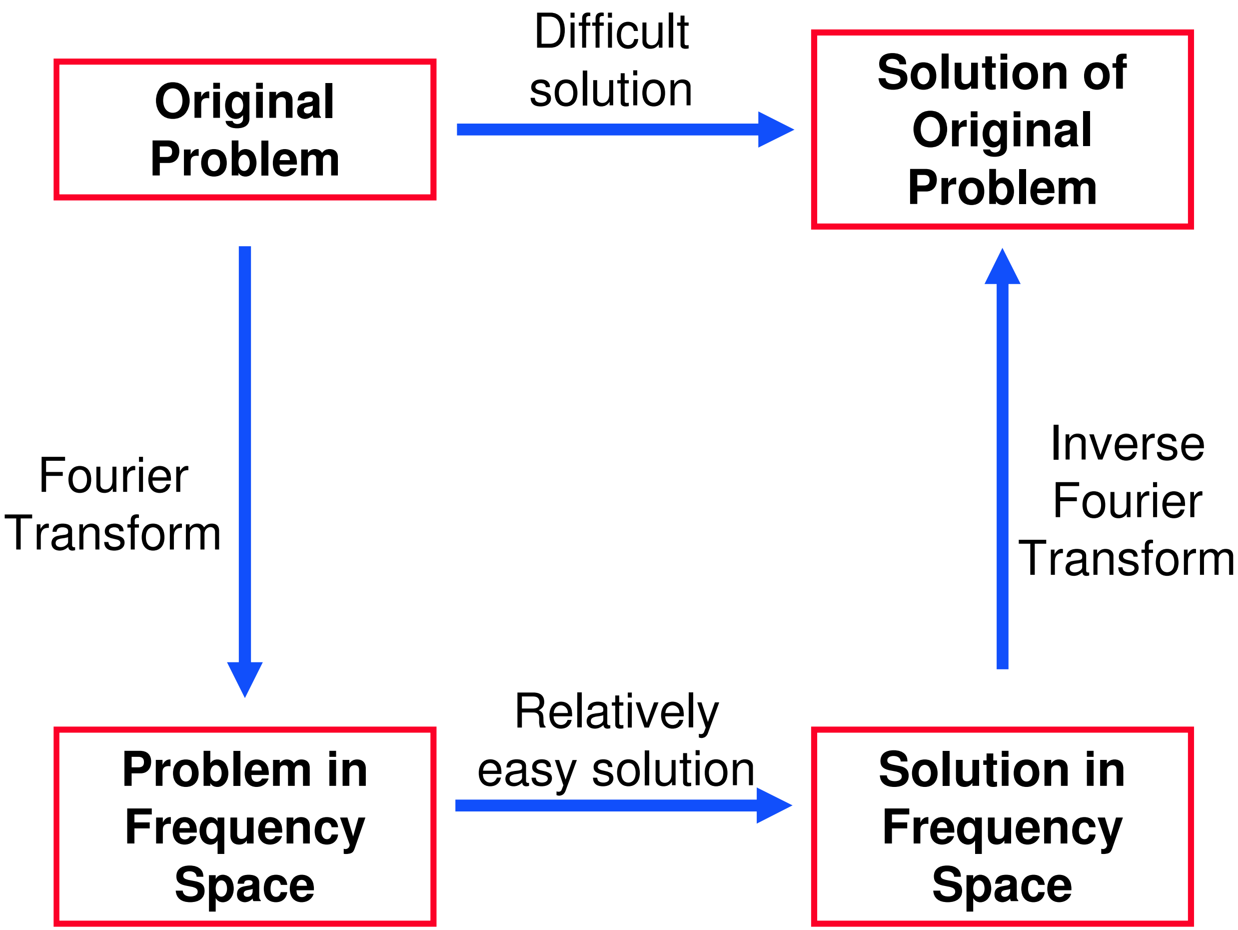
Slide: Freeman & Durand



Recon: Cheetah Phase + Zebra Magnitude

Slide: Freeman & Durand

Why do we need representation in the frequency domain?



The Convolution Theorem

$g = f * h$	$g = f\,h$
implies	implies
$G = F\,H$	$G = F * H$

Convolution in one domain is multiplication in the other and vice versa

The Convolution Theorem

$$\boxed{\tilde{F}\{f(x) * g(x)\} = \tilde{F}\{f(x)\}\tilde{F}\{g(x)\}}$$

and likewise

$$\boxed{\tilde{F}\{f(x)g(x)\} = \tilde{F}\{f(x)\} * \tilde{F}\{g(x)\}}$$

The Convolution Theorem - Proof

Convolution can be represented as a matrix multiplication:

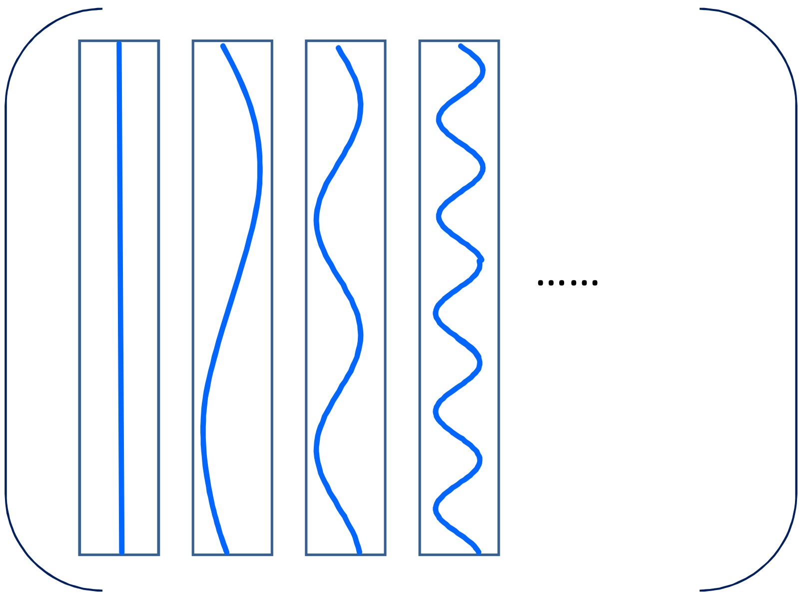
$$y=Ax$$

where A is a circulant matrix.

$$A = \begin{pmatrix} & & \vdots & & \\ \ldots\ldots\ 0\ \ 0 & \boxed{\qquad h \qquad} & 0\ \ 0\ \ \ldots\ldots & & \\ & \ldots\ldots\ 0\ \ 0 & \boxed{\qquad h \qquad} & 0\ \ 0\ \ \ldots\ldots & \\ & & \ldots\ldots\ 0\ \ 0 & \boxed{\qquad h \qquad} & 0\ \ 0\ \ \ldots\ldots \\ & & \vdots & & \end{pmatrix}$$

The Convolution Theorem - Proof

Let F be a matrix composed of the Fourier bases:

$F =$ 

Transformed signal is then: $X = F^T x$

Note 1: $F_{nm} = \frac{1}{\sqrt{N}} e^{\frac{2\pi i m n}{N}} = F_{mn}$ thus: $F = F^T$

Note 2: $F^* F^T = F^T F^* = I$

The Convolution Theorem - Proof

Spatial Domain $$y = Ax$$

Frequency Domain $$F^T y = F^T A x$$

$$F^T y = F^T A (F^* F^T)\, x$$

$$= (F^T A F^*) F^T x$$

$$= D F^T x$$

Where $D = F^T A F^*$ is a diagonal matrix with the Fourier coefficients of filter h on its diagonal.

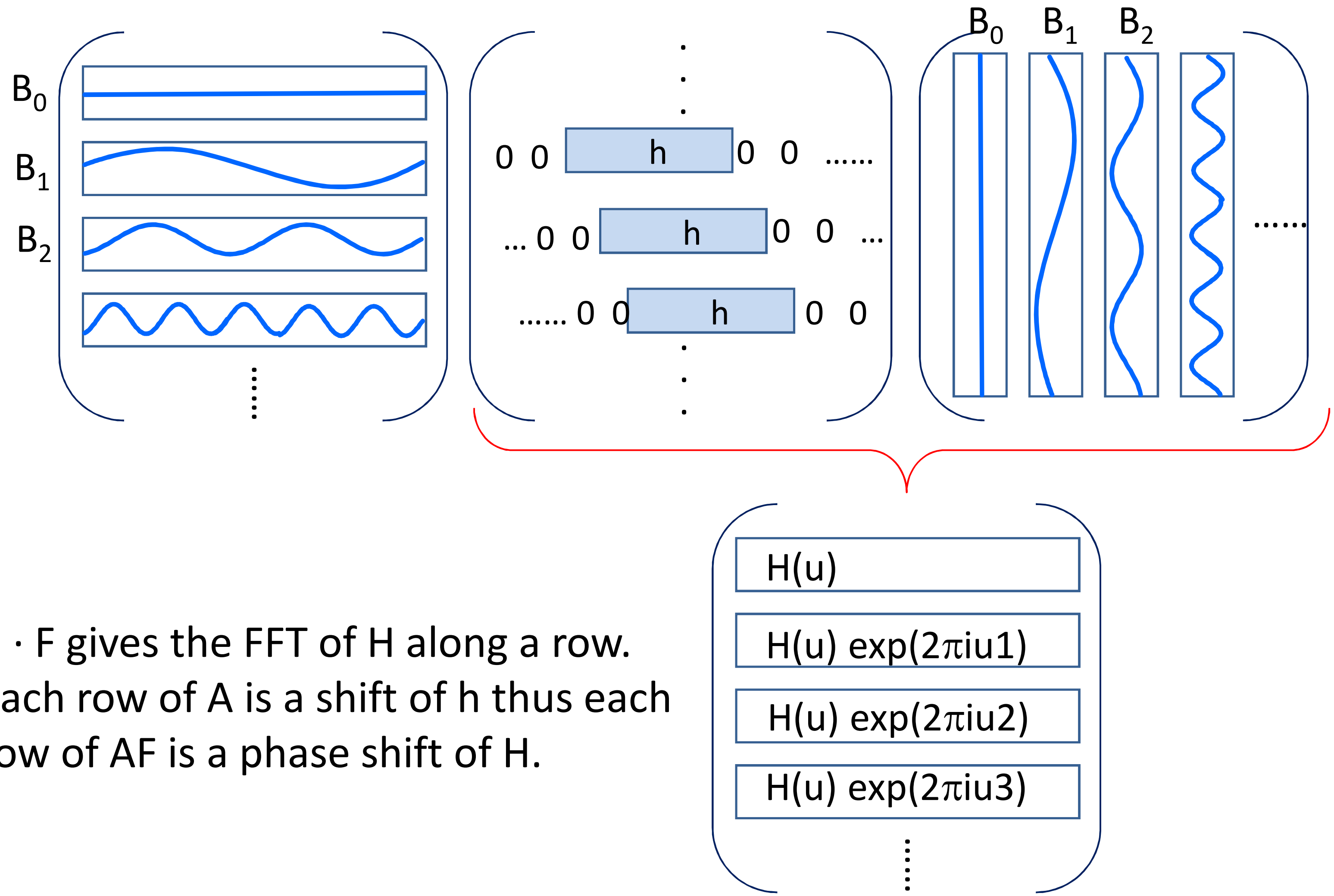
The Convolution Theorem - Proof

$$F^T y = D F^T x$$

$$Y = D\ X$$

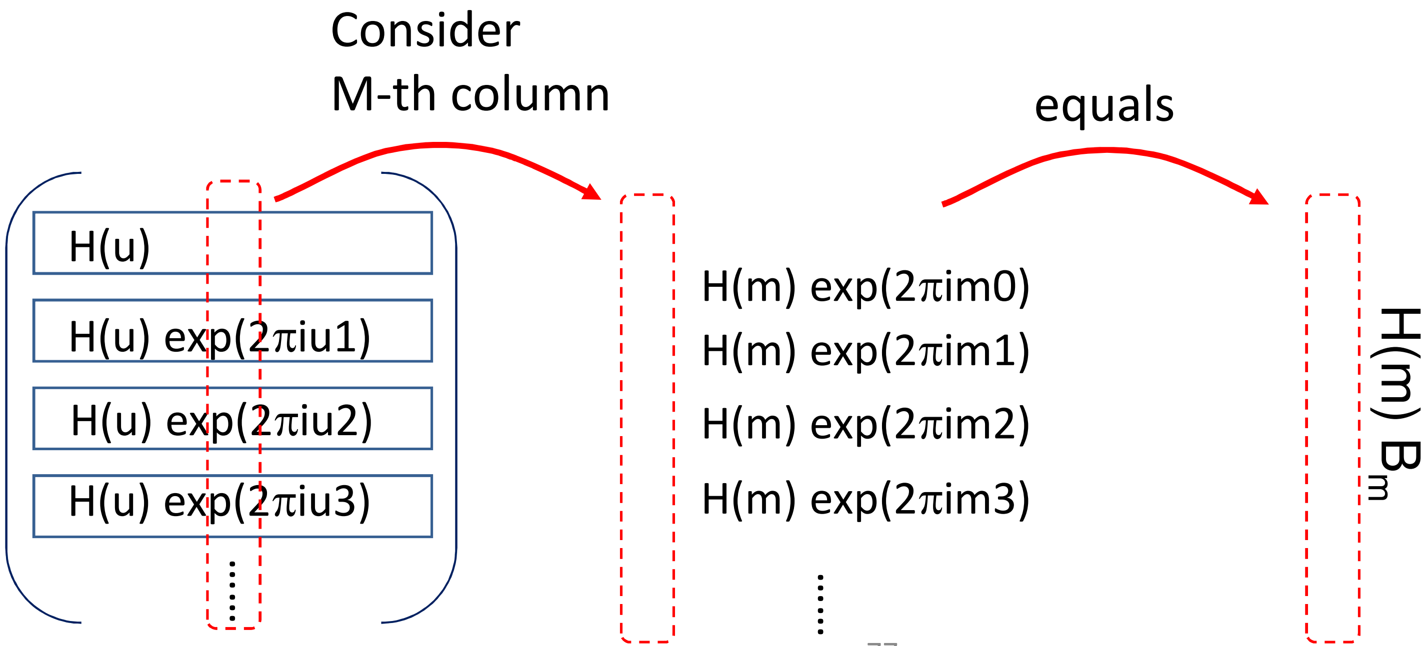
Thus, the Convolution theorem is nothing more than a system diagonalization.

The Convolution Theorem - Proof



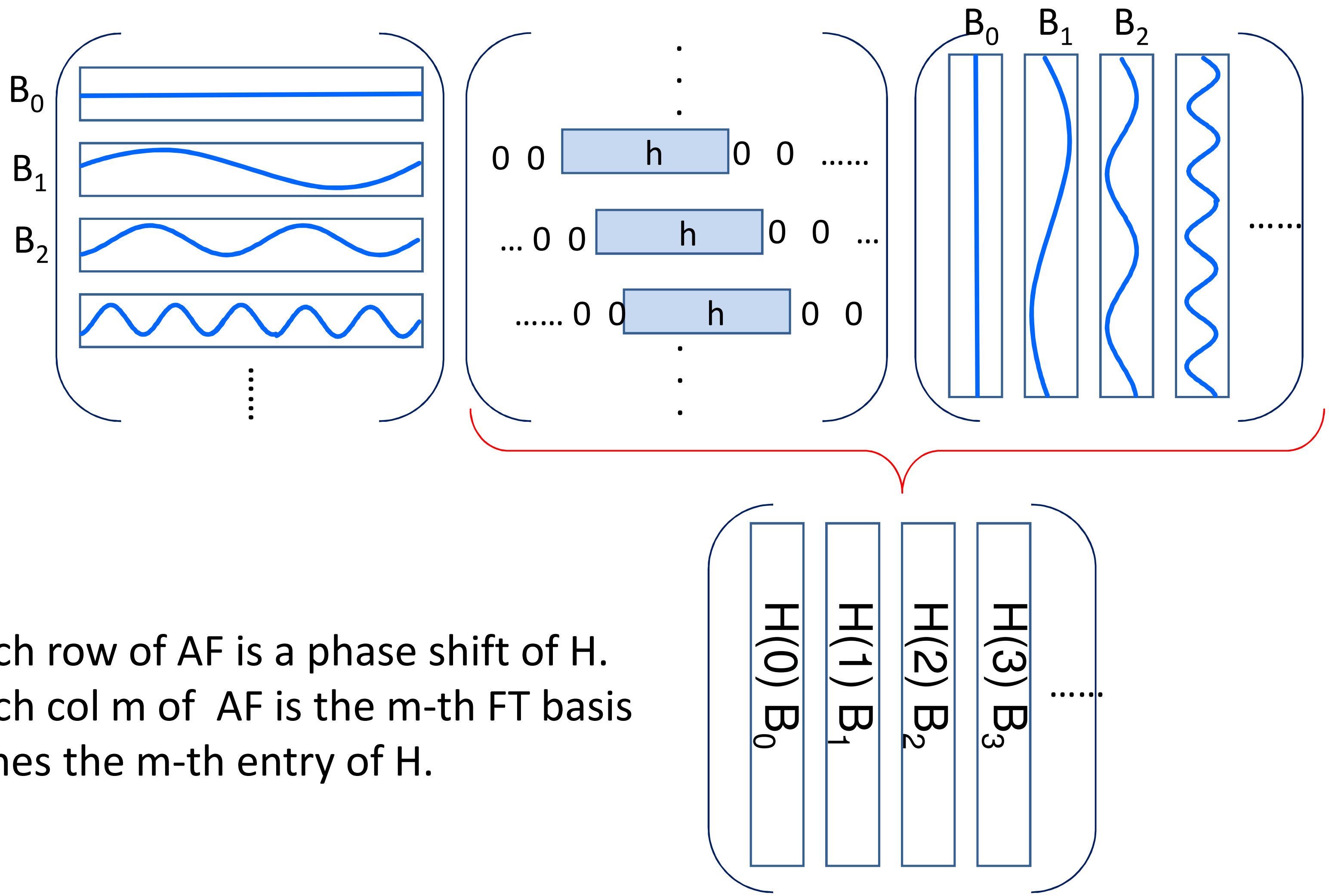
h · F gives the FFT of H along a row. Each row of A is a shift of h thus each row of AF is a phase shift of H.

The Convolution Theorem - Proof



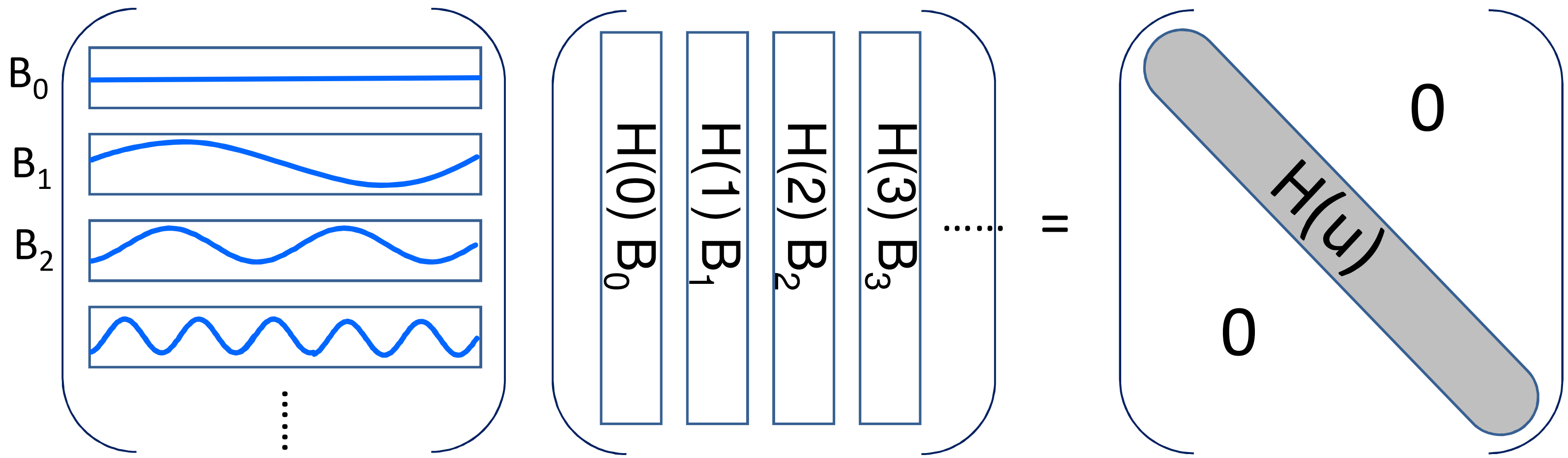
Each col of AF is H(m) times the m-th basis of Fourier.

The Convolution Theorem - Proof



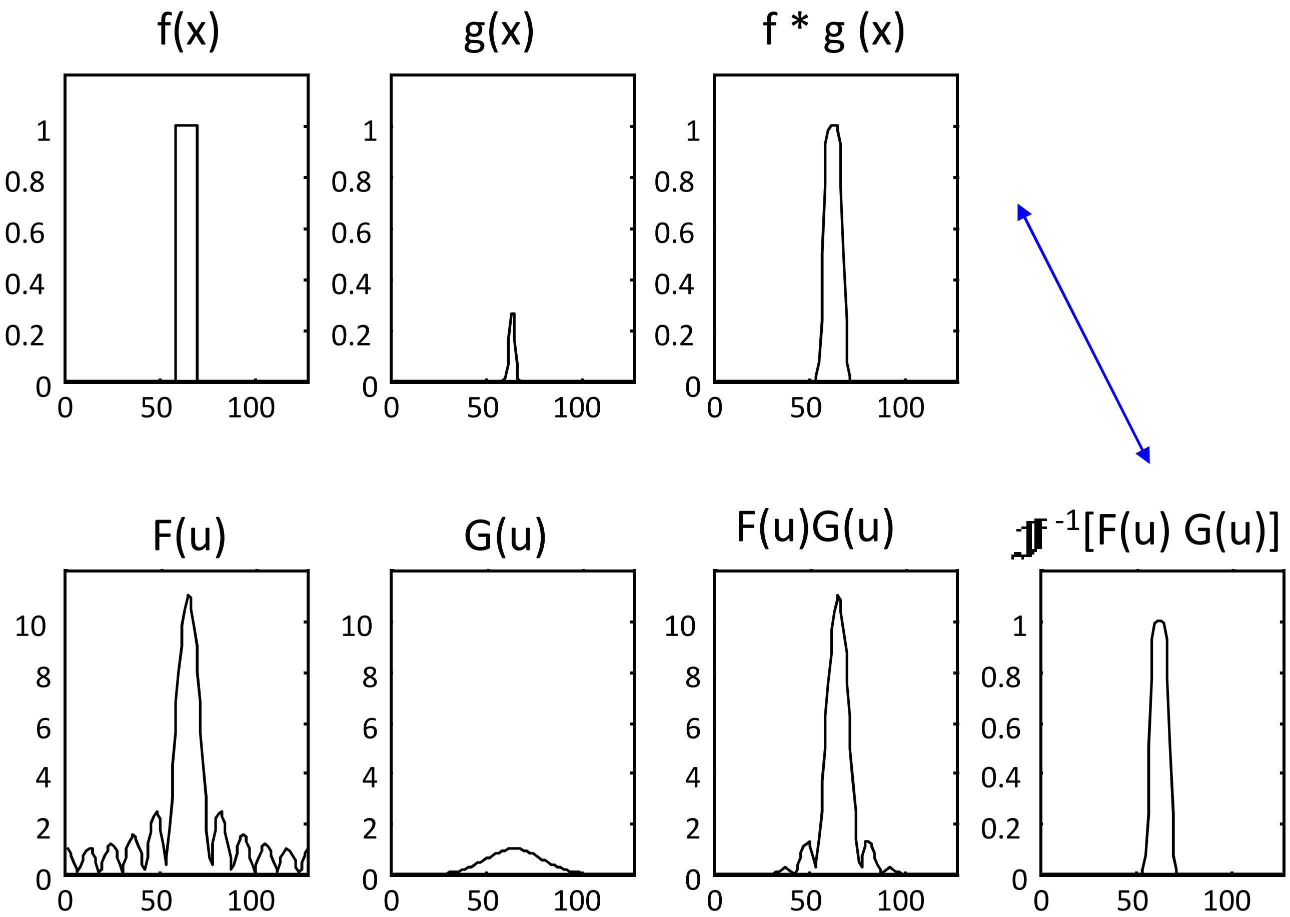
Each row of AF is a phase shift of H.
Each col m of AF is the m-th FT basis times the m-th entry of H.

The Convolution Theorem - Proof

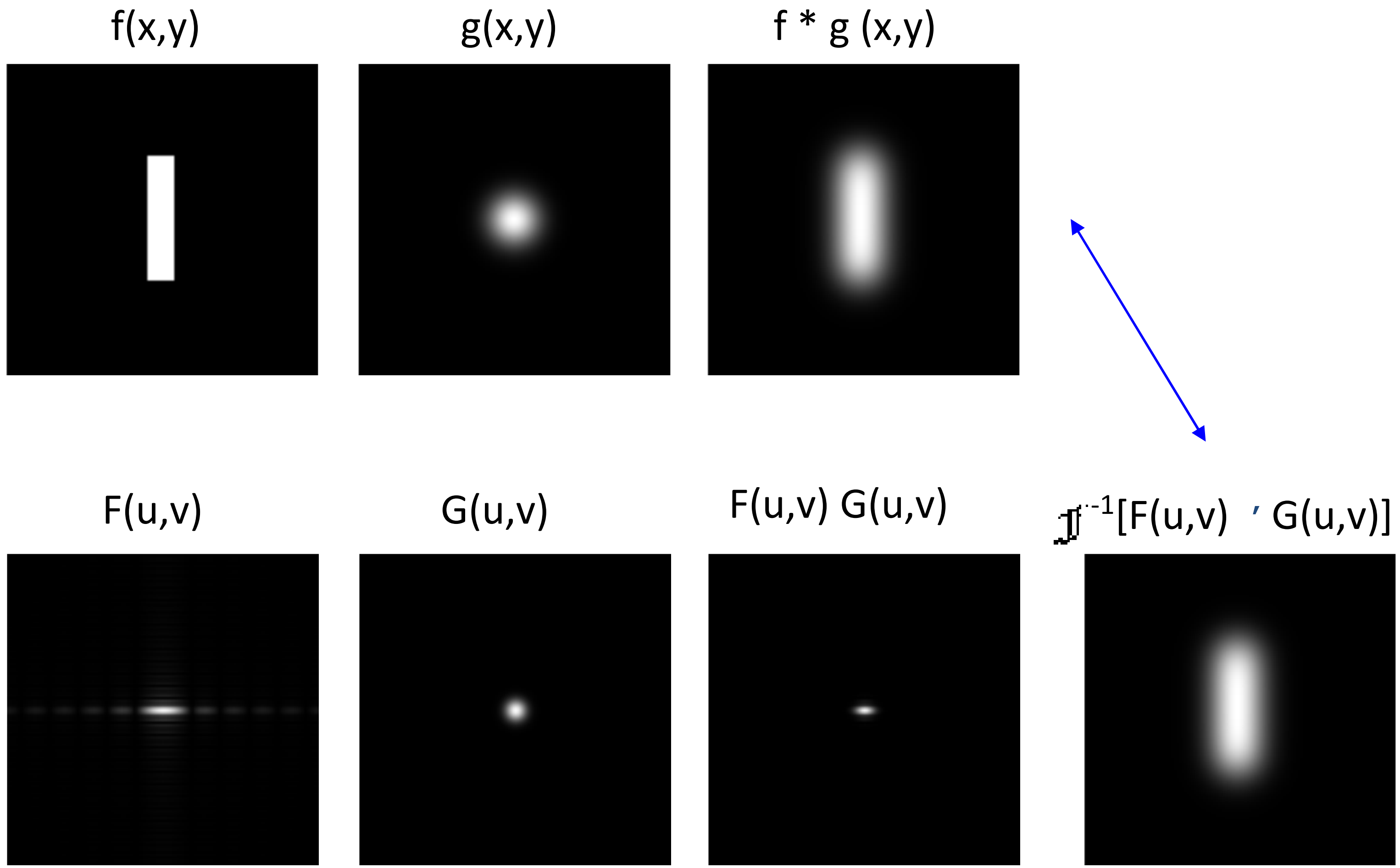


D is a diagonal matrix due to orthogonality of B_m .
H is on the diagonal of D.

The Convolution Theorem - Example

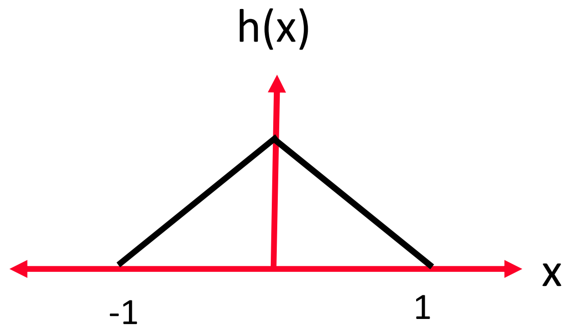


The Convolution Theorem - Example

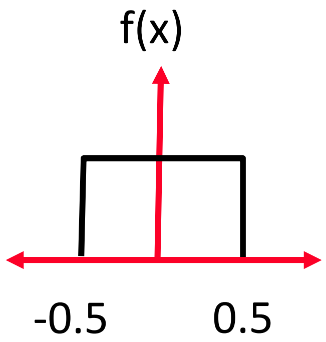


Convolution Theorem - Example

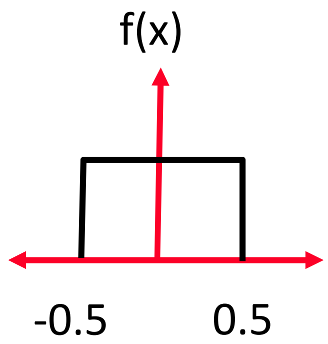
Example: What is the Fourier Transform of:



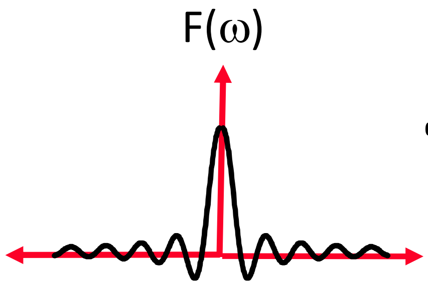
h(x) =



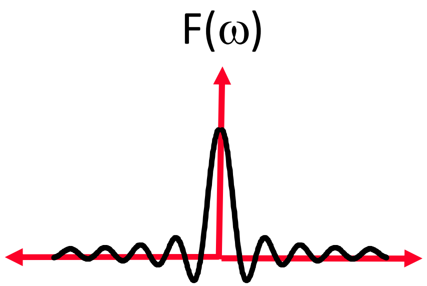
$*$



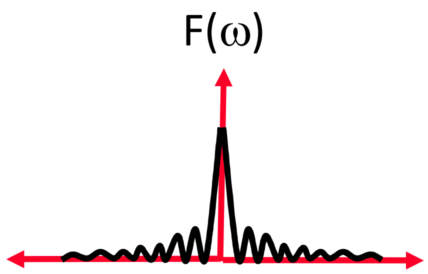
H(ω) =



$\cdot$



=



Convolution Theorem - Example

Example: What is the Fourier Transform of the Dirac Function?

$$\delta(x) = \begin{cases} 1 & if \quad x = 0 \\ 0 & otherwise \end{cases}$$

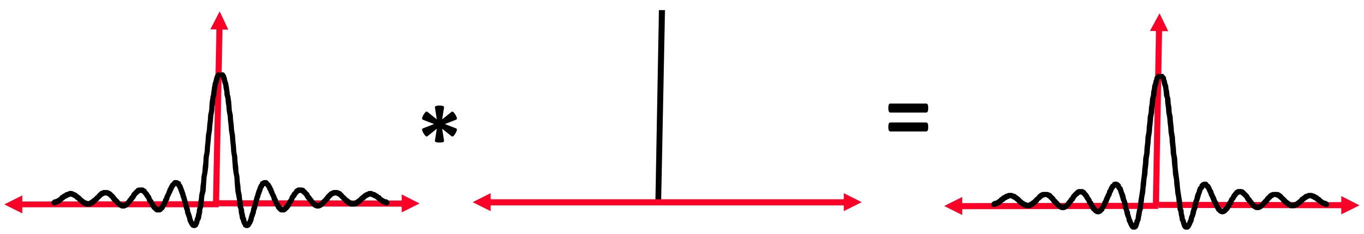
For any function **f(x)** :

$$f(x) \; * \; \delta(x) \; = f(x)$$

$$\downarrow \qquad\qquad\qquad \downarrow$$

$$F(u) \cdot \mathrm{F}[\delta(x)] = F(u)$$

$$\boxed{\tilde{F}[\delta(x)] = 1}$$



Convolution Theorem - Example

Example: What is the Fourier Transform of a constant Function?

$$g(x) = c$$

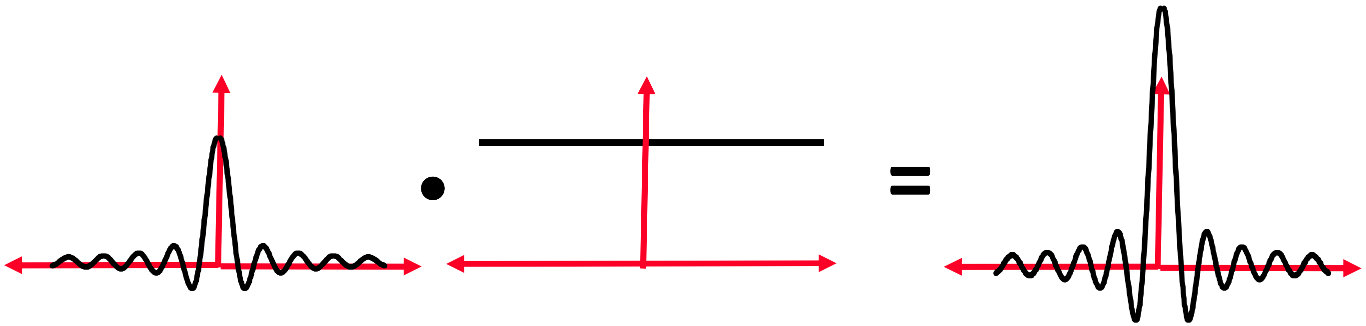
For any function **g(x)** :

$$f(x)\,g(x) = cf(x)$$

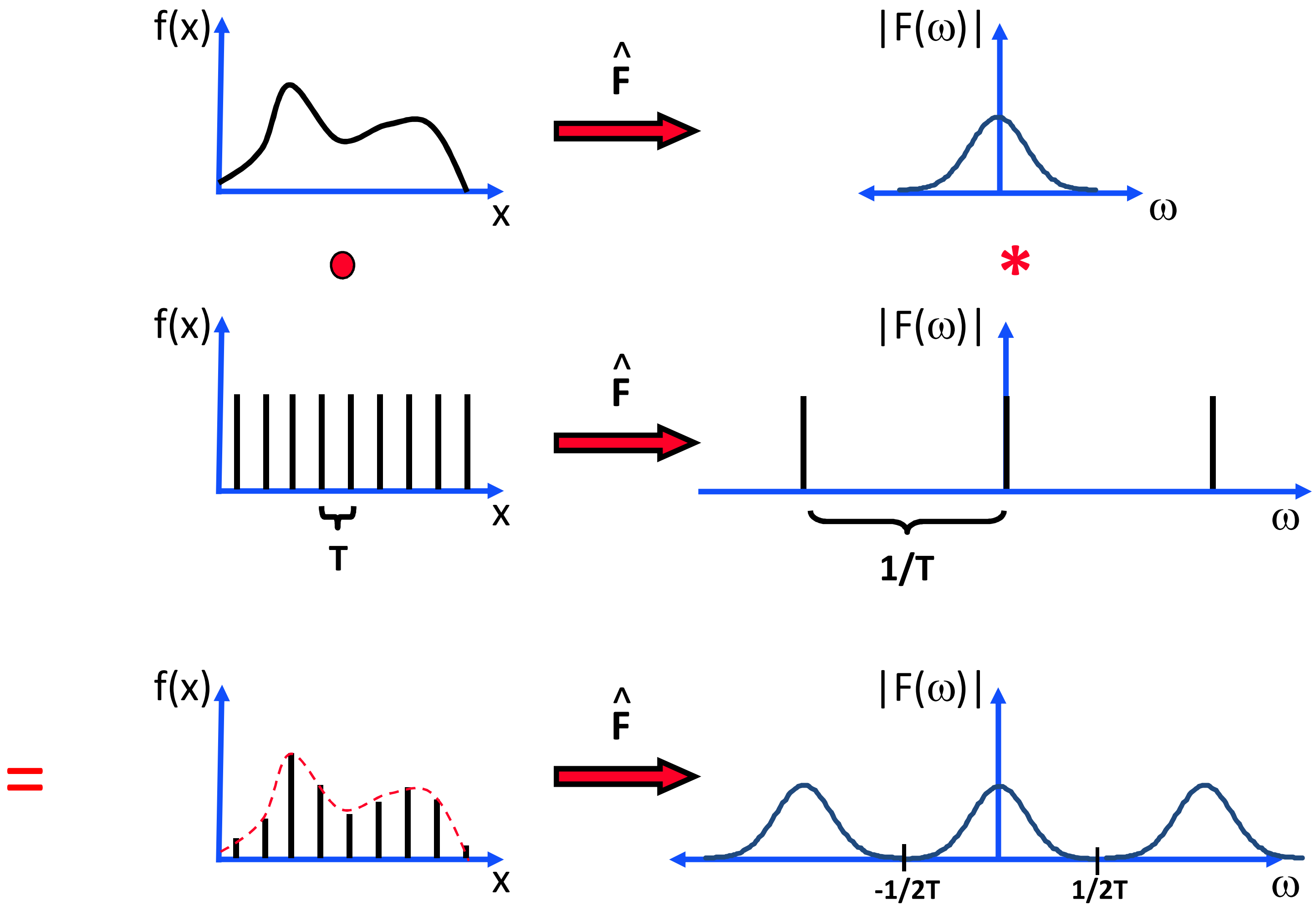
$$\downarrow \qquad\qquad \downarrow$$

$$F(u) * G(u) = cF(u)$$

$$\boxed{\tilde{F}[c] = c\delta(u)}$$

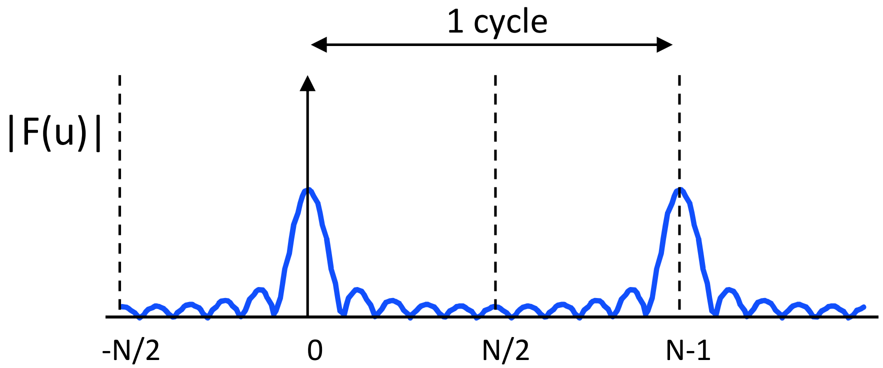


Sampling the Spatial Domain

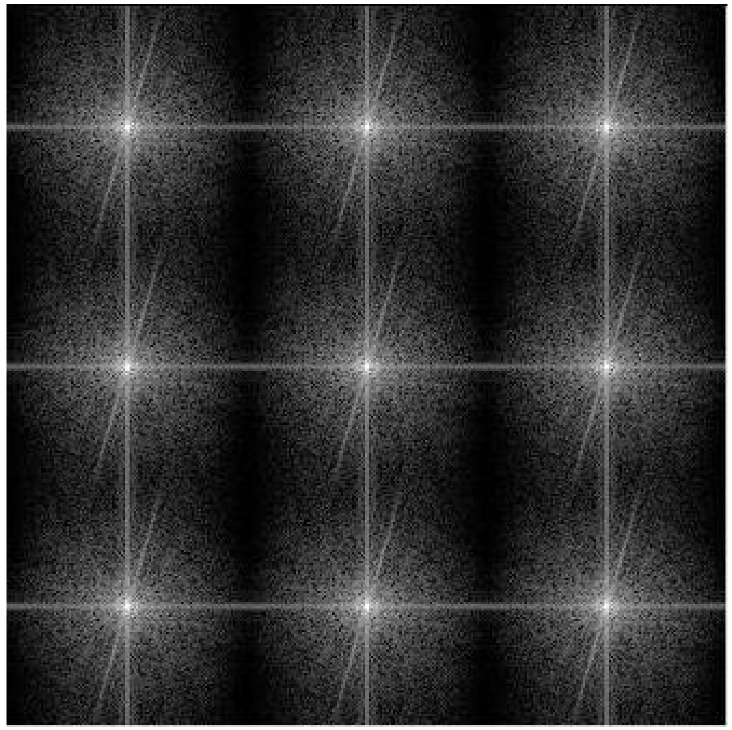


Sampling f(x) at cycle T produces replicas in the frequency domain with cycle 1/T.

Symmetry of FT :

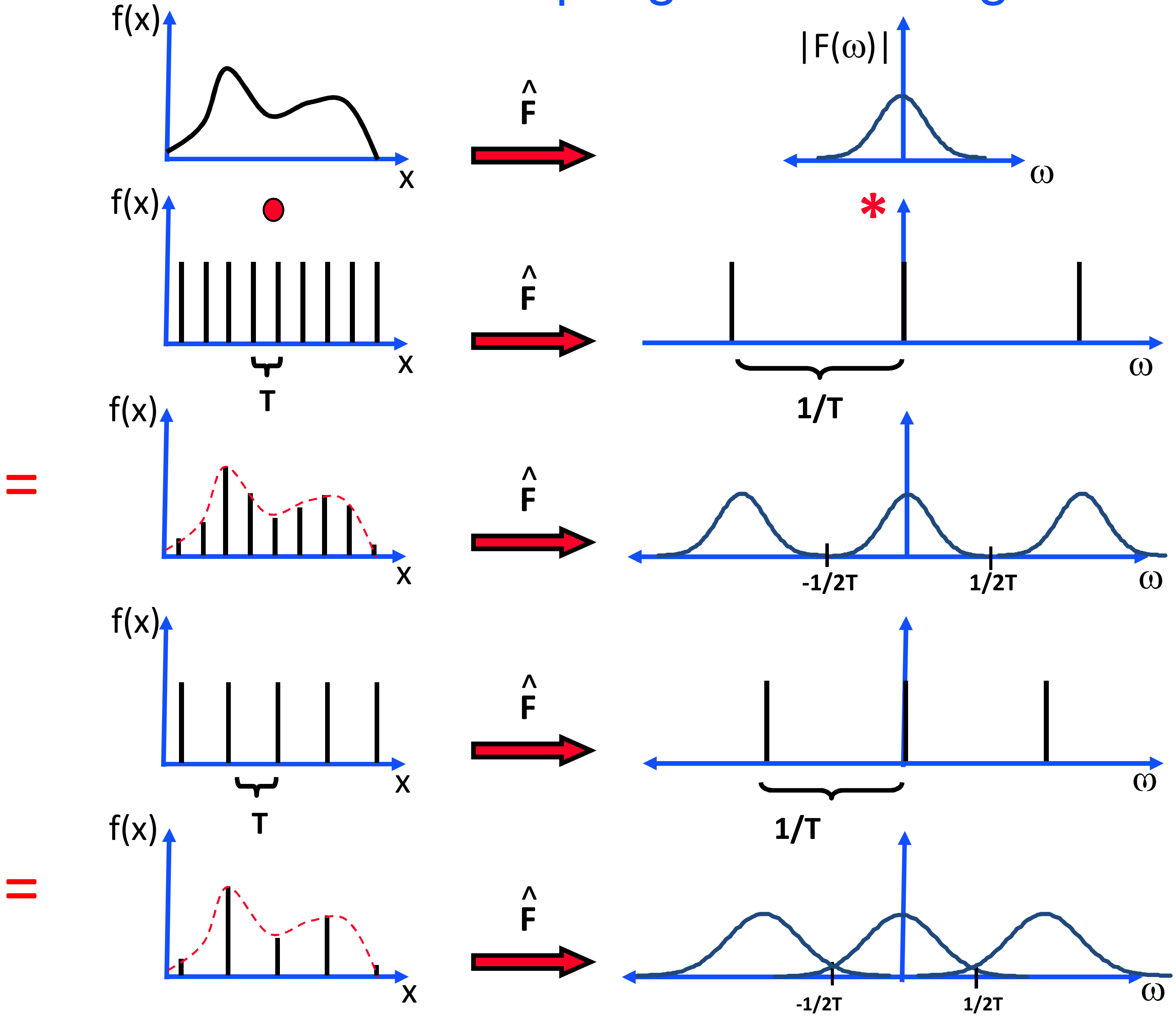


$$F(k)=F(N+k)$$



$$\begin{aligned} F(u,v) &= F(u+N,v) \\ &= F(u,v+M) \\ &= F(u+N,v+M) \end{aligned}$$

Undersampling and Aliasing



Critical Sampling

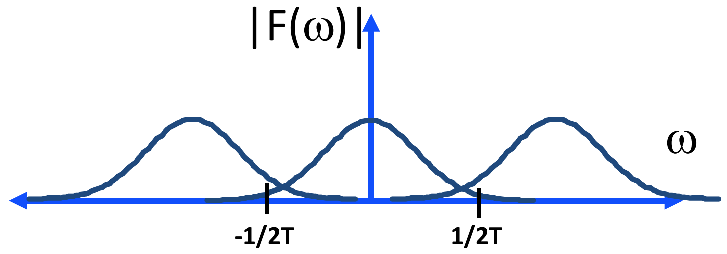
- If the maximal frequency of f(x) is ω_{max} , it is clear from the above replicas that ω_{max} should be smaller that 1/2T

$$\boxed{\omega_{sampling} = \frac{1}{T} > 2\omega_{\max}}$$

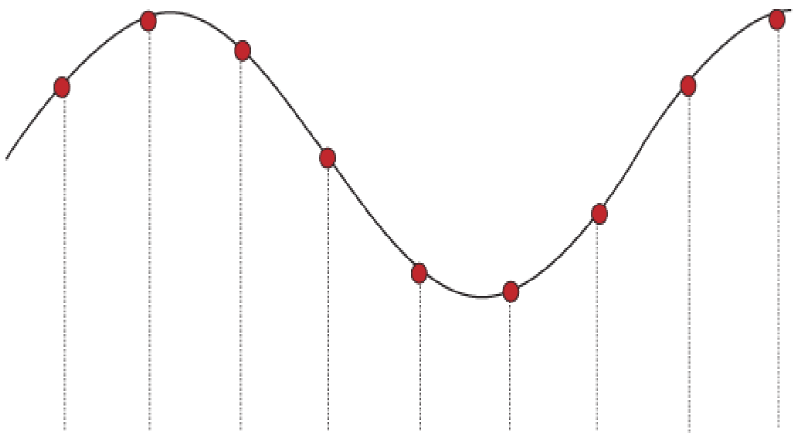
- <u>Nyquist Theorem</u>: If maximal frequency of f(x) is ω_{max} , sampling rate should be larger than $2\omega_{max}$ in order to fully reconstruct f(x) from its samples.

  $2\omega_{max}$ is the Nyquist frequency.

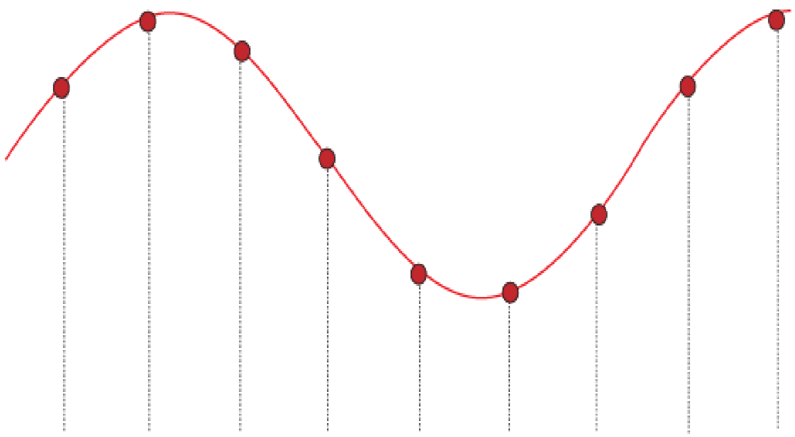
- If the sampling rate is smaller than $2\omega_{max}$ overlapping replicas produce aliasing.



Critical Sampling



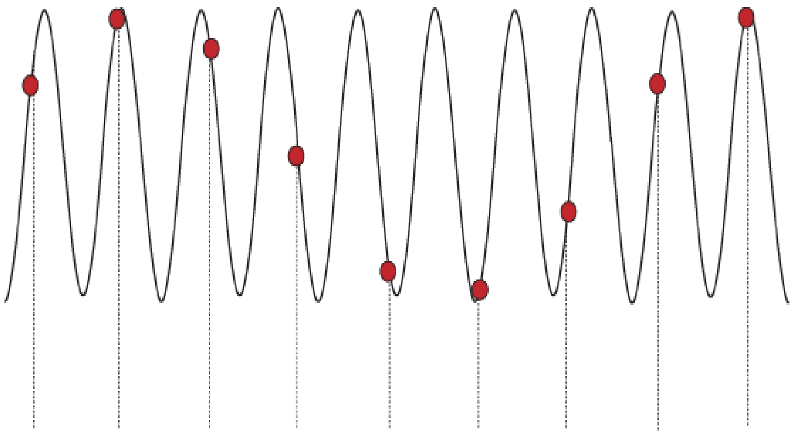
Input



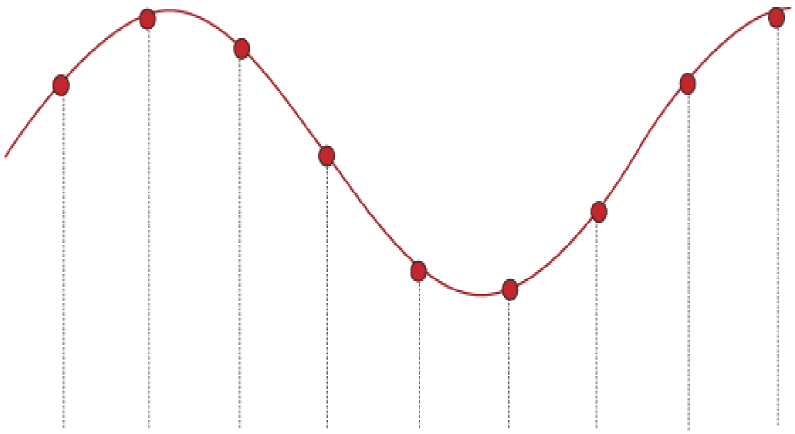
Reconstructed

Demo: B. Freeman

Aliasing



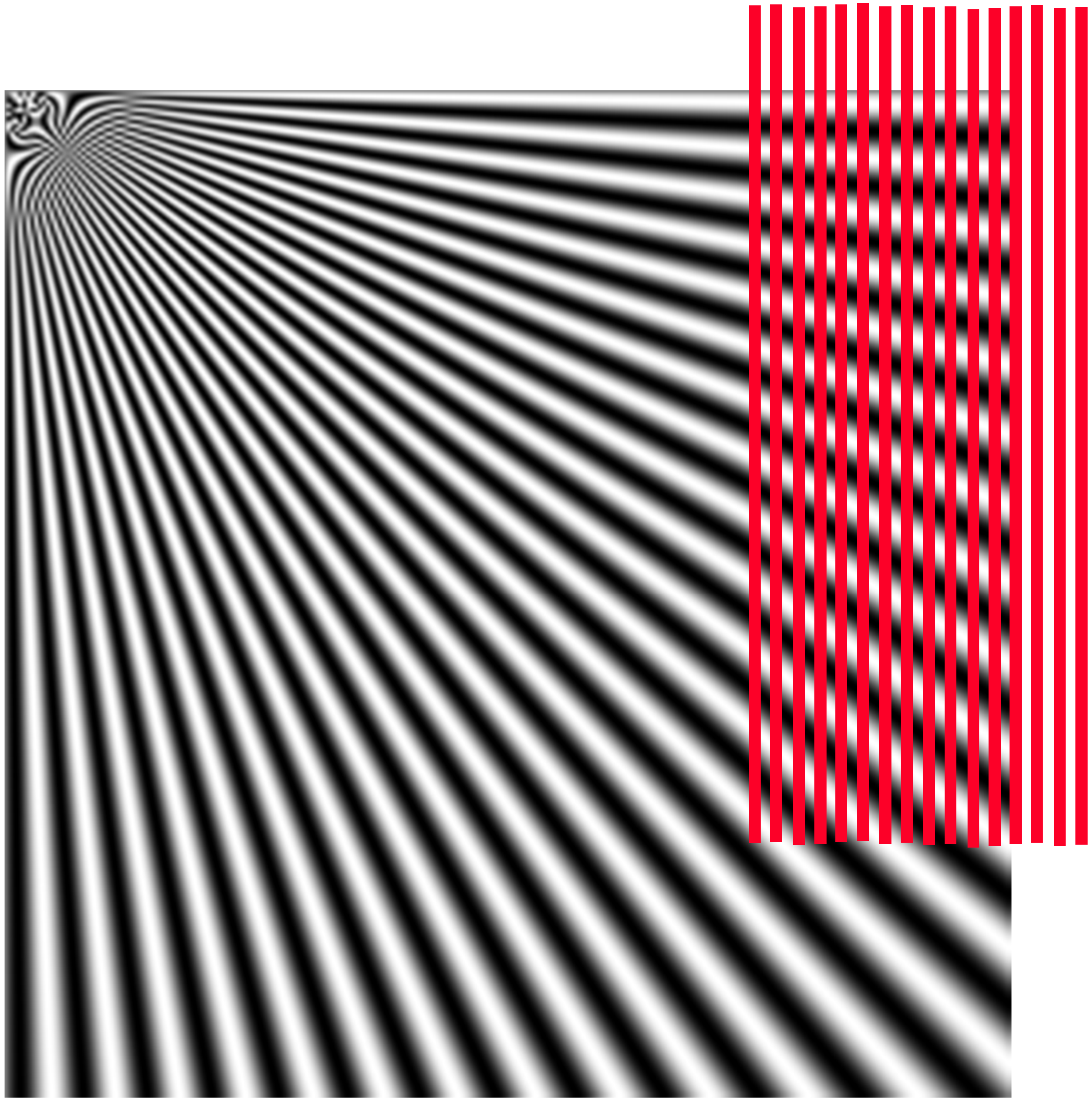
Input



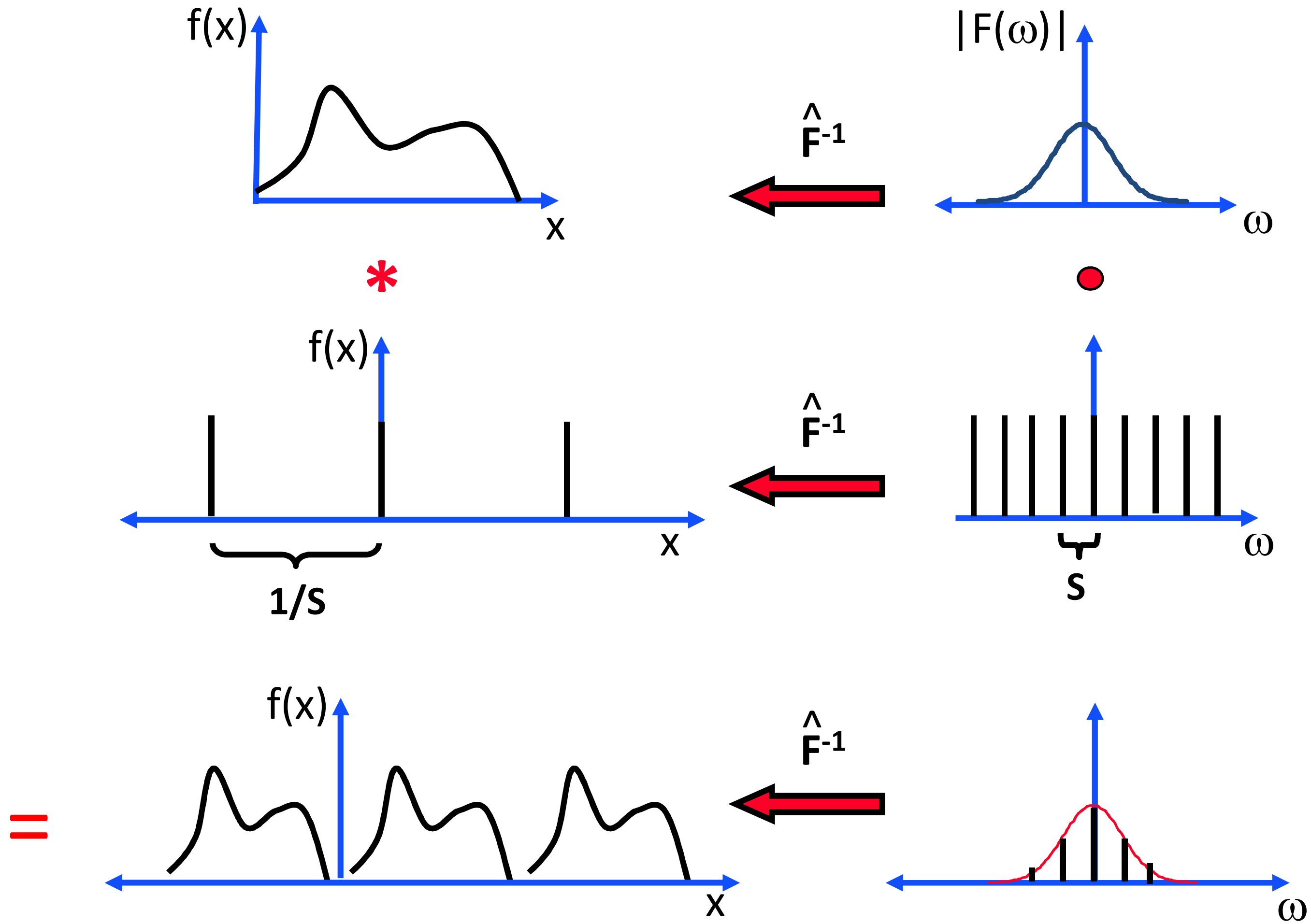
Reconstructed

Demo: B. Freeman

Aliasing



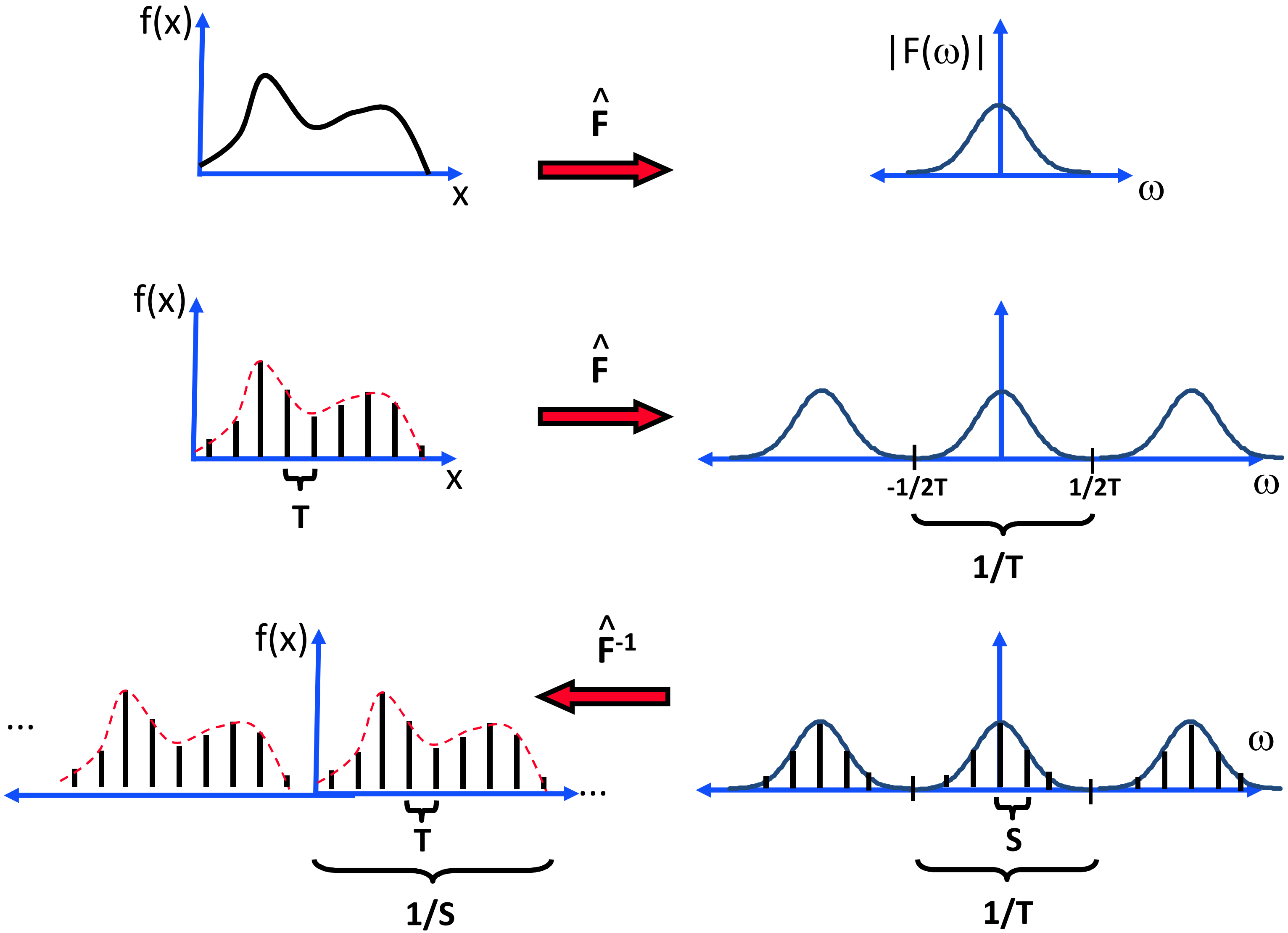
Sampling the Frequency Domain



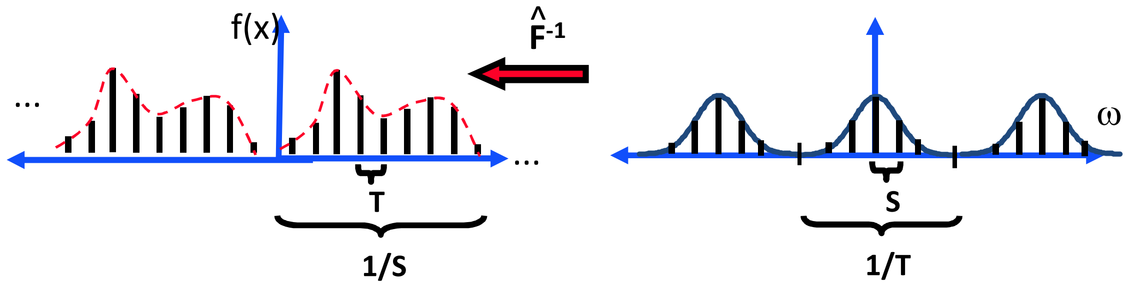
Sampling F(ω) at cycle S produces replicas in the image domain with cycle 1/S.

Sampling both Image and Frequency Domain

Sampling both f(x) with impulses of cycle T and F(ω) with impulses of cycle S:



Sampling both Image and Frequency Domain



Question: Assuming f(x) was samples with N samples. What is the minimal number of samples M in F(ω) in order to fully reconstruct f(x) ?

Answer:

If we sample f(x) with N samples of cycle T, the support of f(x) is NT.

The support of F(ω) is 1/T in the frequency domain.

If we sample F(ω) with M samples, the sample cycle is 1/MT.

The replicas in the spatial domain are each MT.

In order to avoid replicas overlap, MT should be greater or equal to NT (the function support).

$$\boxed{M \geq N}$$

Optimal Interpolation

If sampling rate is above Nyquist – it is possible to fully reconstruct f(x) from its samples.

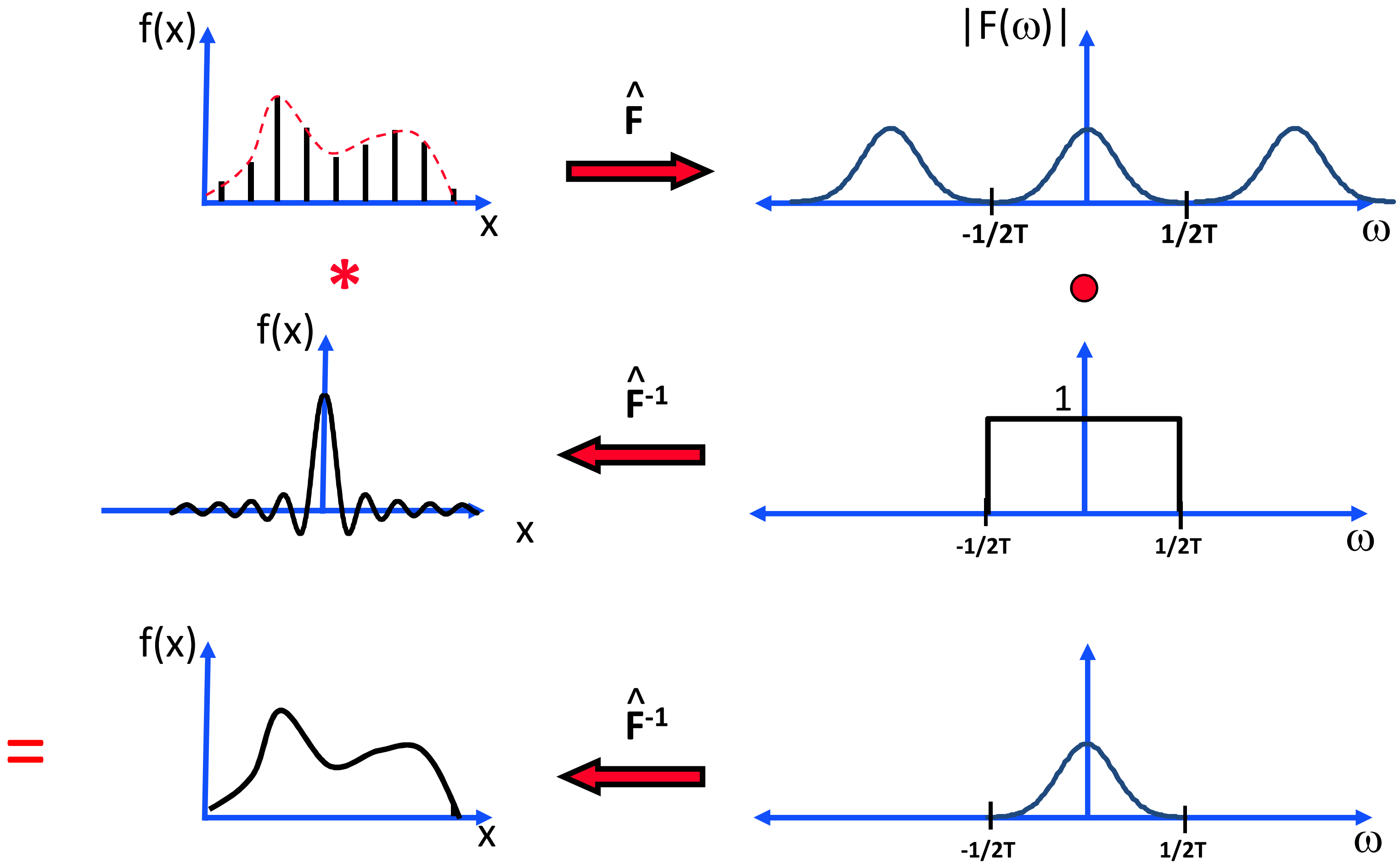


Image Scaling

If sampling rate is above Nyquist – it is possible to interpolate f(x) from its samples.

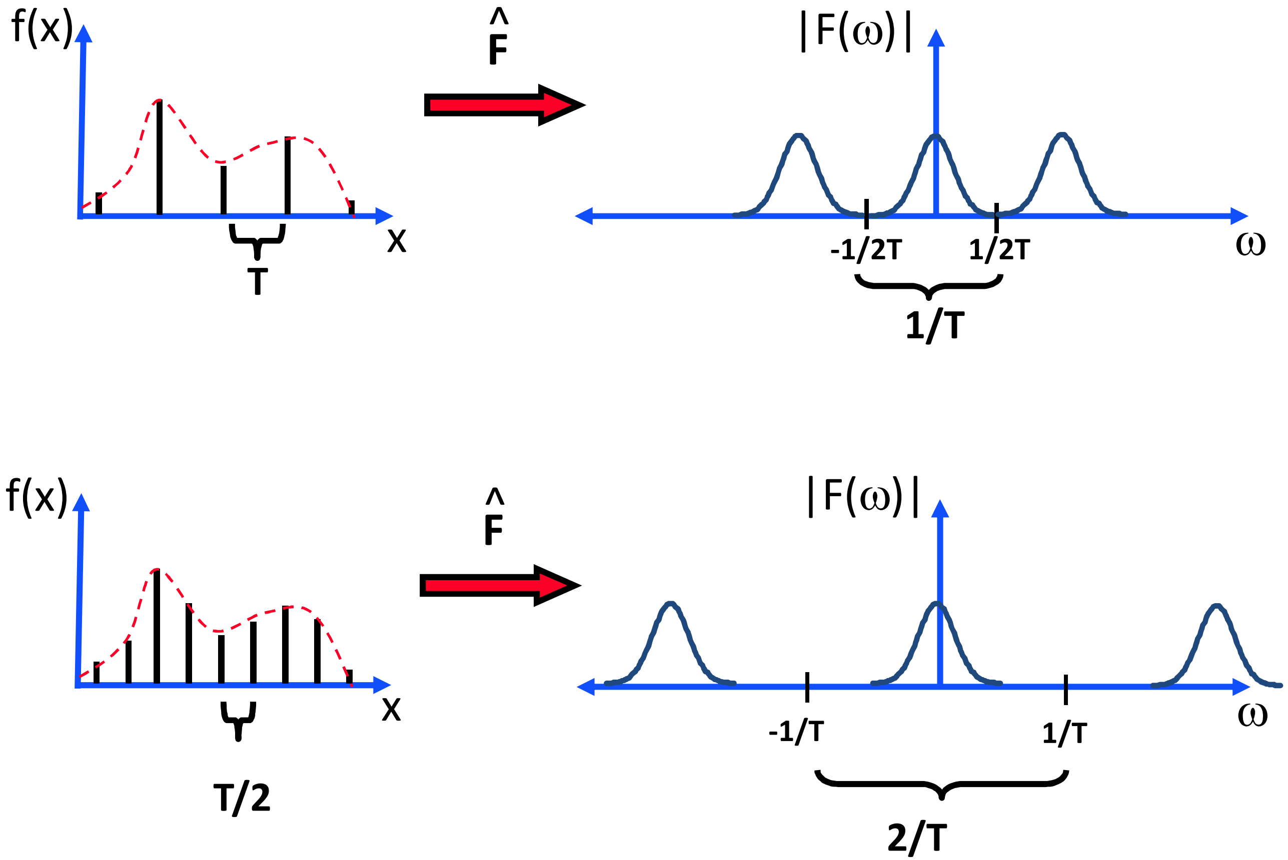


Image Scaling

If sampling rate is above Nyquist – it is possible to interpolate f(x) from its samples.

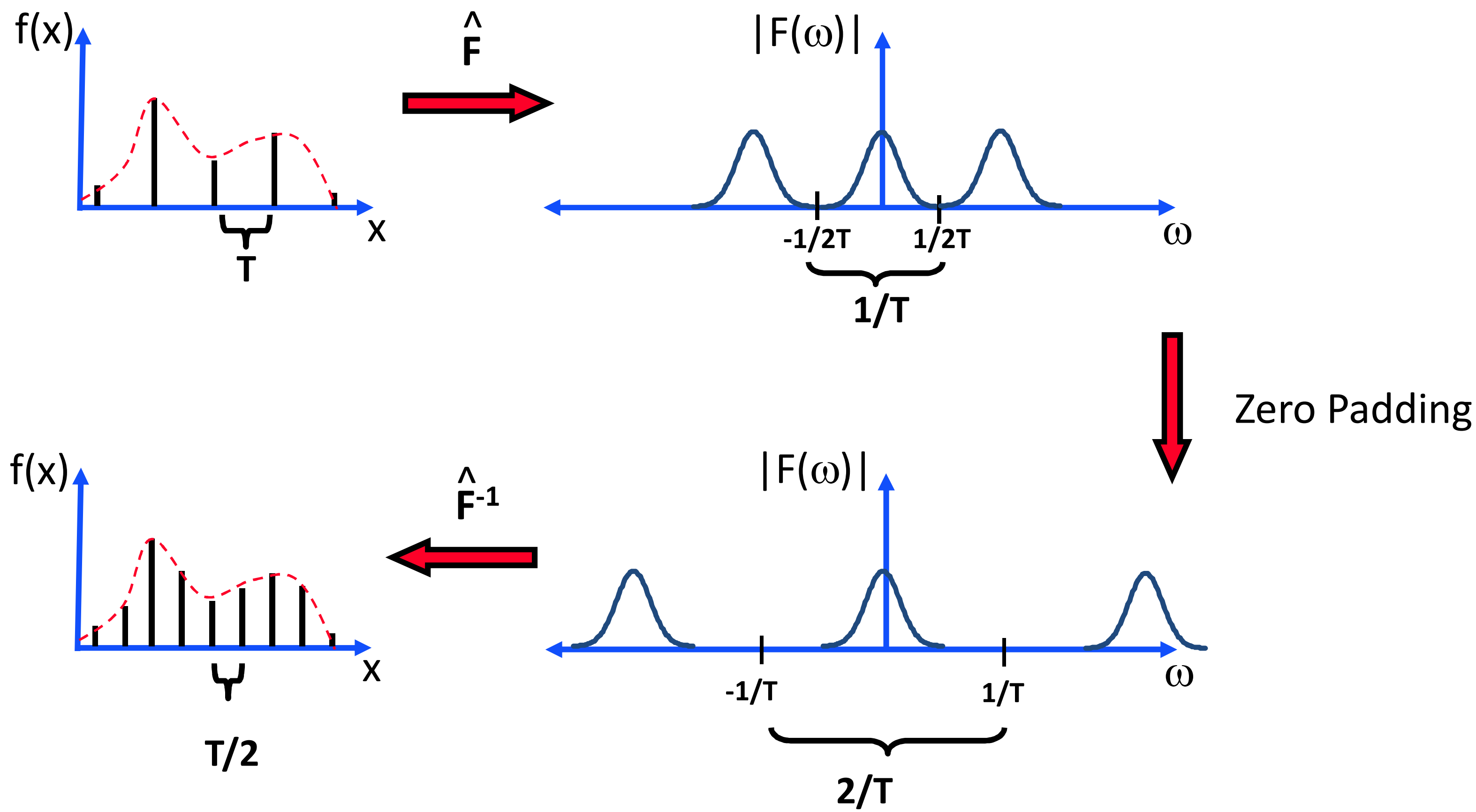


Image Scaling Example

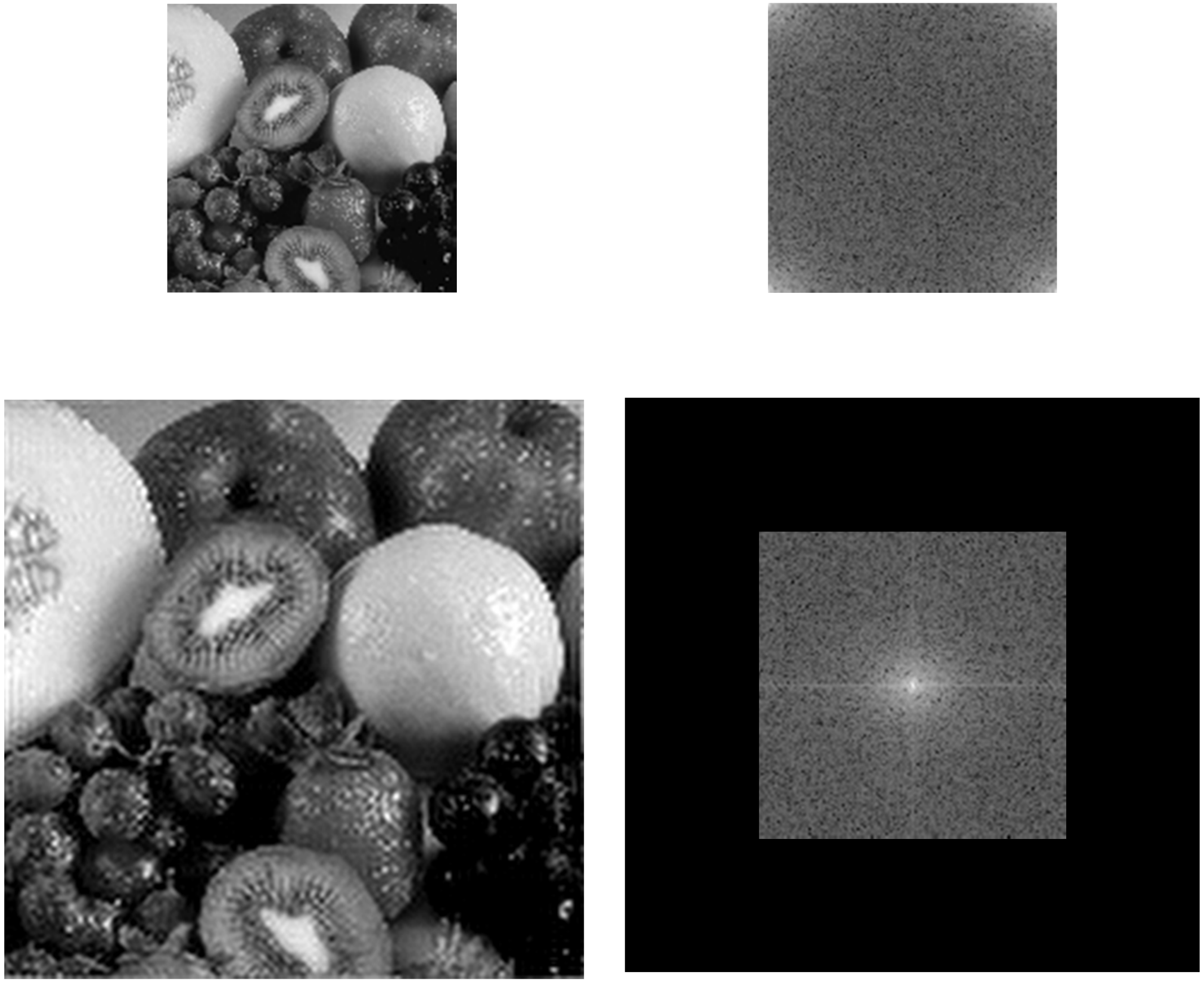


Image Scaling Example

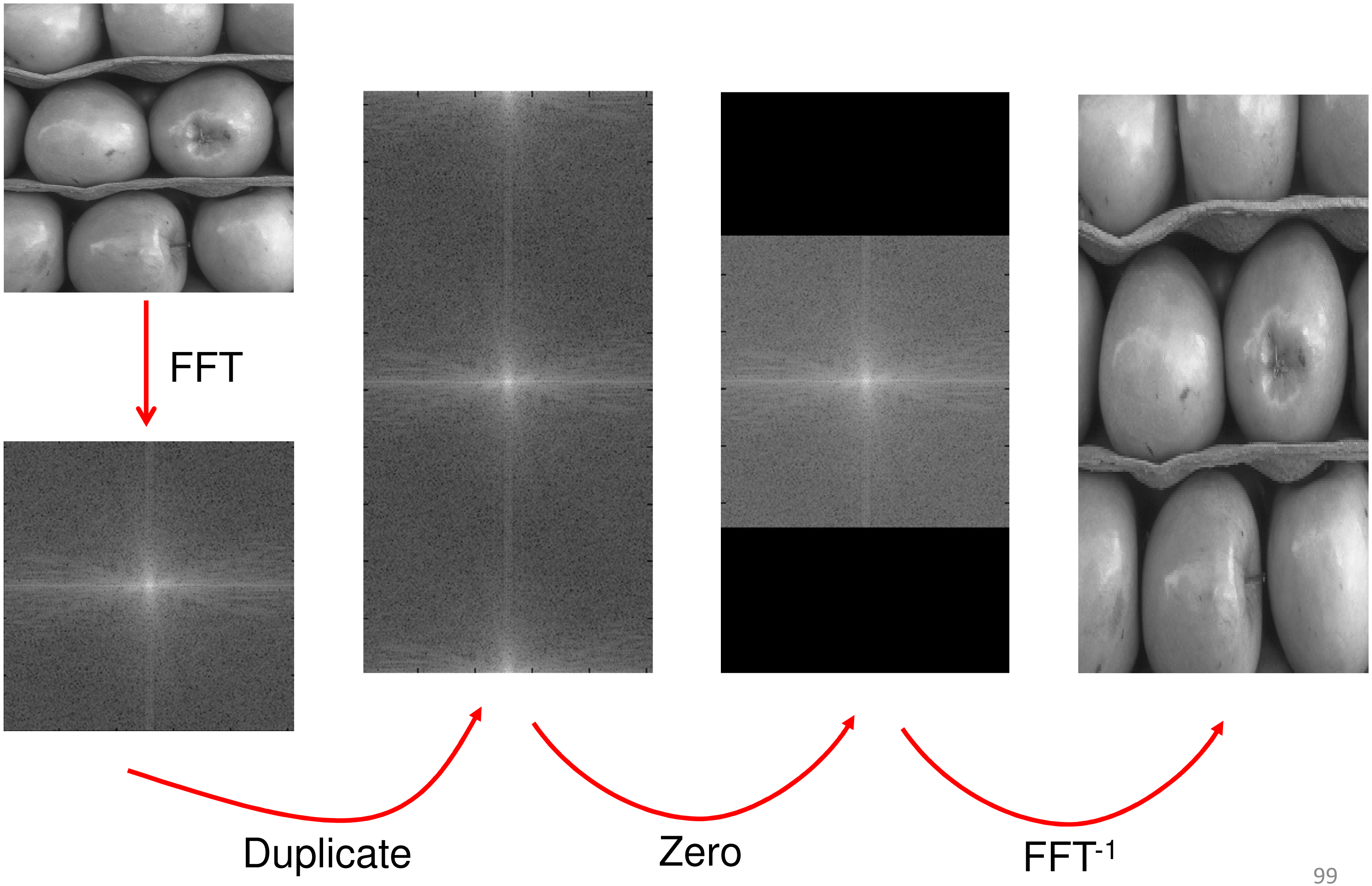
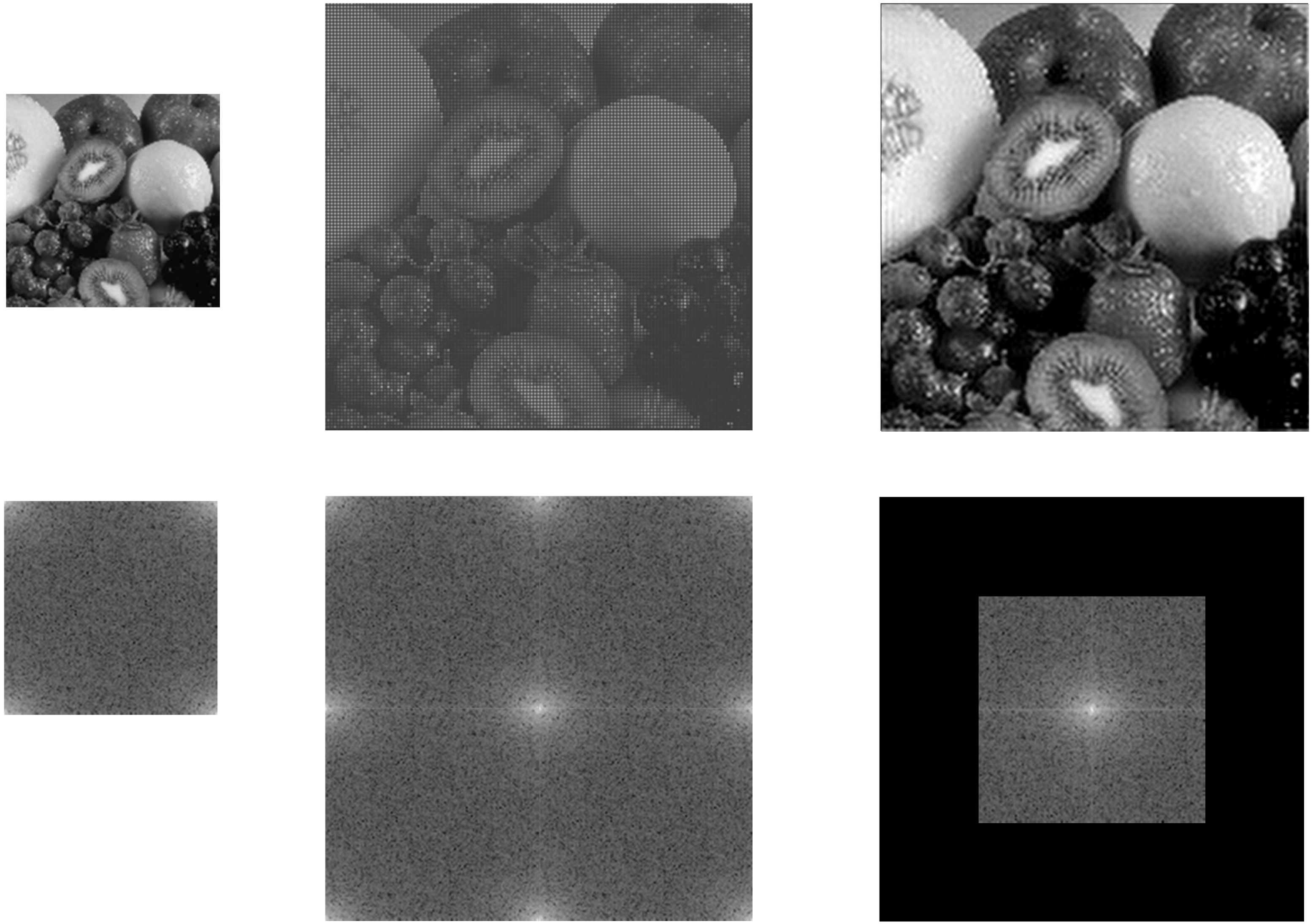
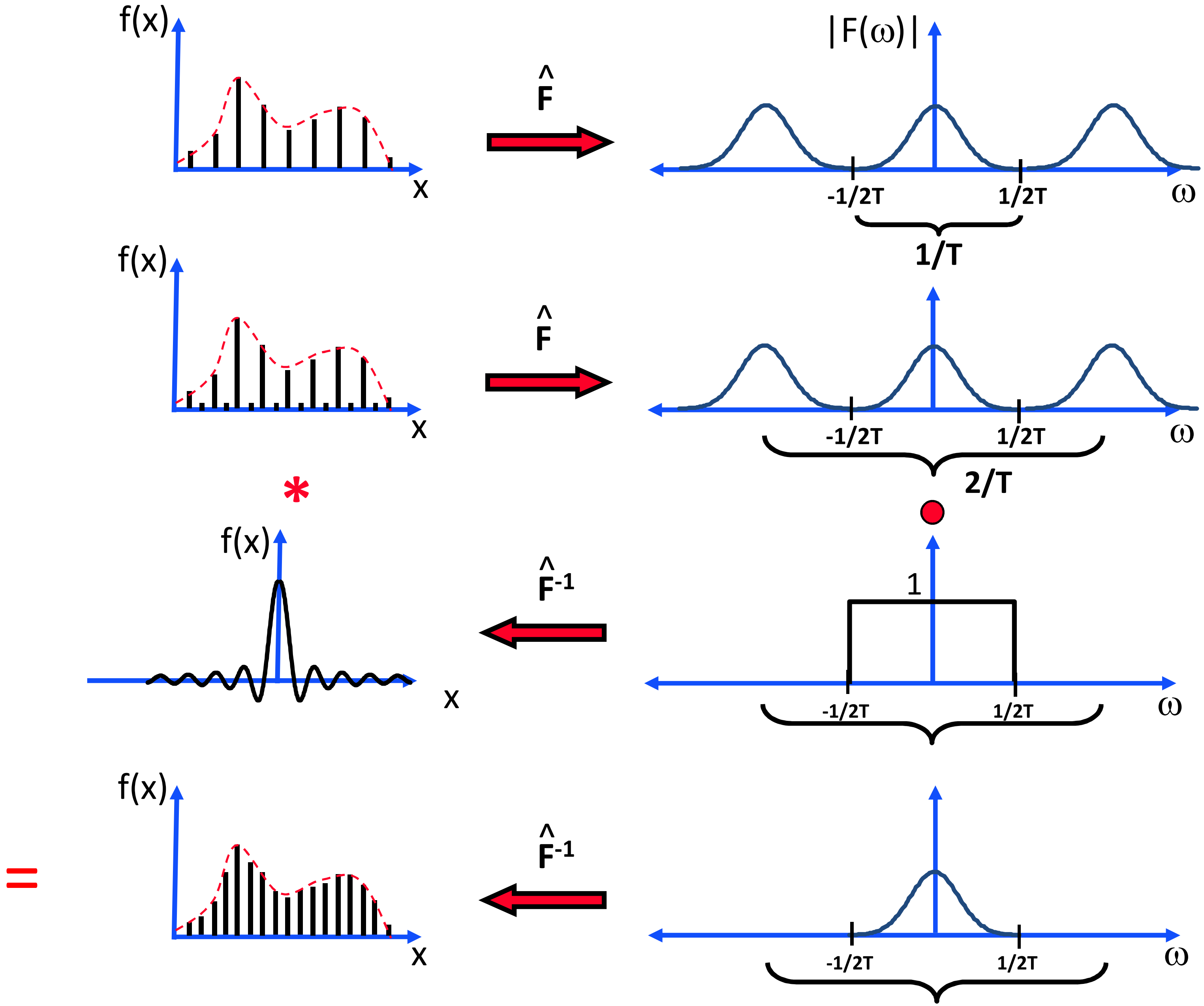


Image Scaling Example



Optimal Interpolation - Digital



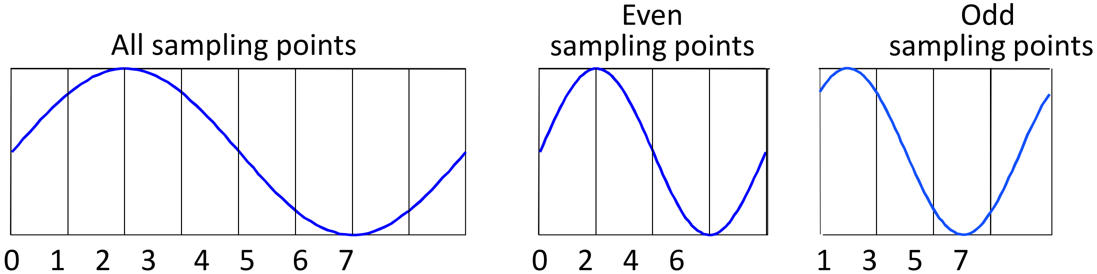
Fast Fourier Transform

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{\frac{-2\pi i u x}{N}} \qquad u = 0, 1, 2, ..., N-1$$

$O(n^2)$ operations

$$F(u) = \underbrace{\frac{1}{N}\sum_{x=0}^{N/2-1} f(2x)e^{\frac{-2\pi i u 2x}{N}}}_{\text{even } x} + \underbrace{\frac{1}{N}\sum_{x=0}^{N/2-1} f(2x+1)e^{\frac{-2\pi i u(2x+1)}{N}}}_{\text{odd } x}$$

$$= \frac{1}{2}\left[\underbrace{\frac{1}{N/2}\sum_{x=0}^{N/2-1} f(2x)e^{\frac{-2\pi i u x}{N/2}}}_{\text{Fourier Transform of of N/2 even points}} + e^{\frac{-2\pi i u}{N}}\underbrace{\frac{1}{N/2}\sum_{x=0}^{N/2-1} f(2x+1)e^{\frac{-2\pi i u x}{N/2}}}_{\text{Fourier Transform of of N/2 odd points}}\right]$$

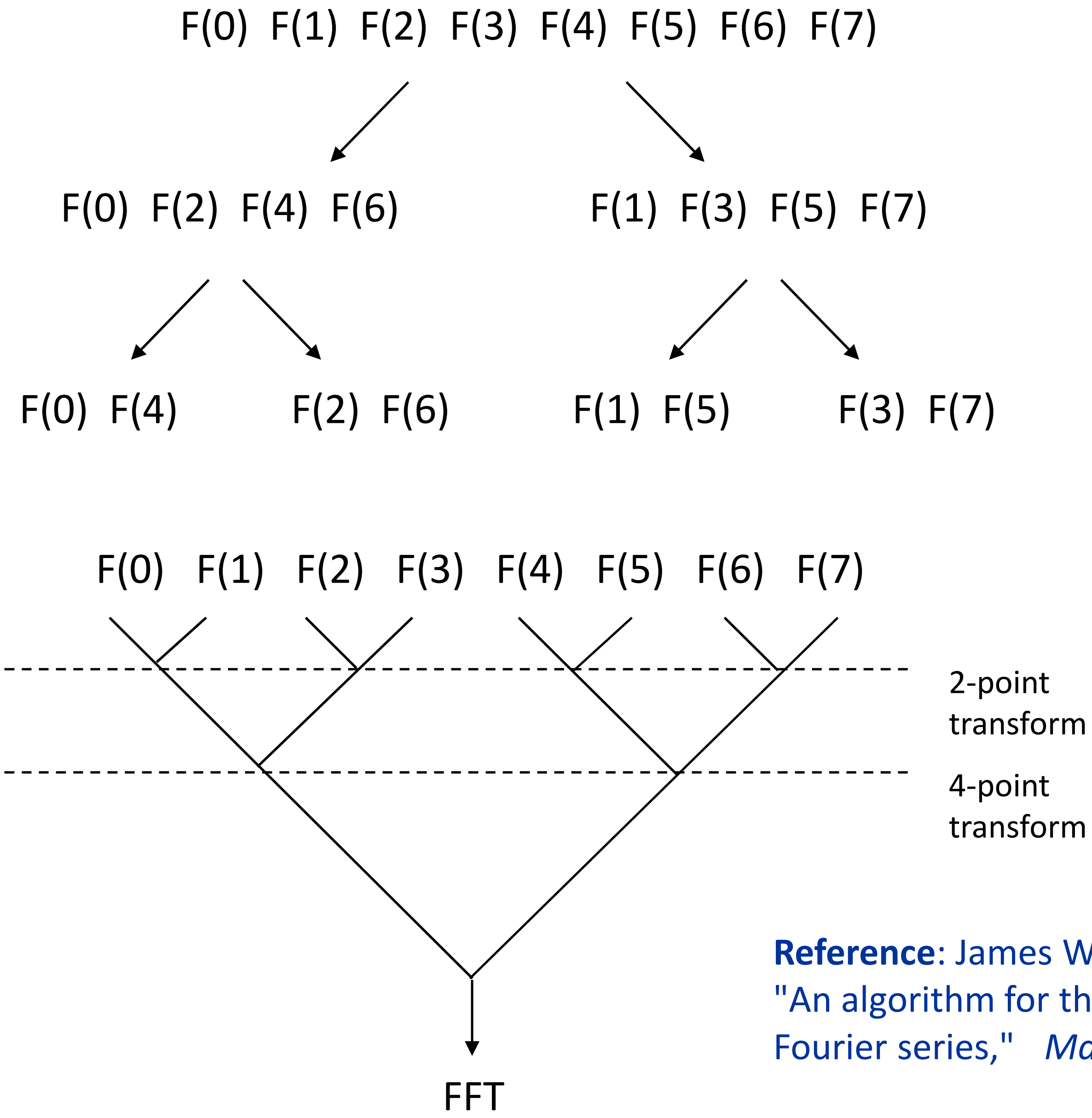


The Fourier transform of N inputs, can be performed as 2 Fourier Transforms of N/2 inputs each + one complex multiplication and addition for each value.
Thus, if F(N) is the computation complexity of FFT:

F(N)=F(N/2)+F(N/2)+O(N)

$\Rightarrow$ F(N)=N logN

Fast Fourier Transform



FFT 1D:

$O(N \log(N))$ operations

FFT of 2D (NxN) :

$O(N^2 \log(N))$ operations

Reference: James W. Cooley and John W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.* **19**, 297–301 (1965).