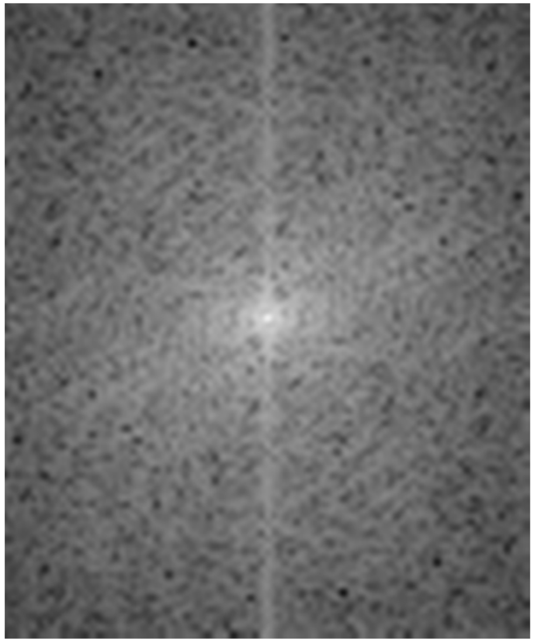
Fourier Transform 2D

The 2D Discrete Fourier Basis

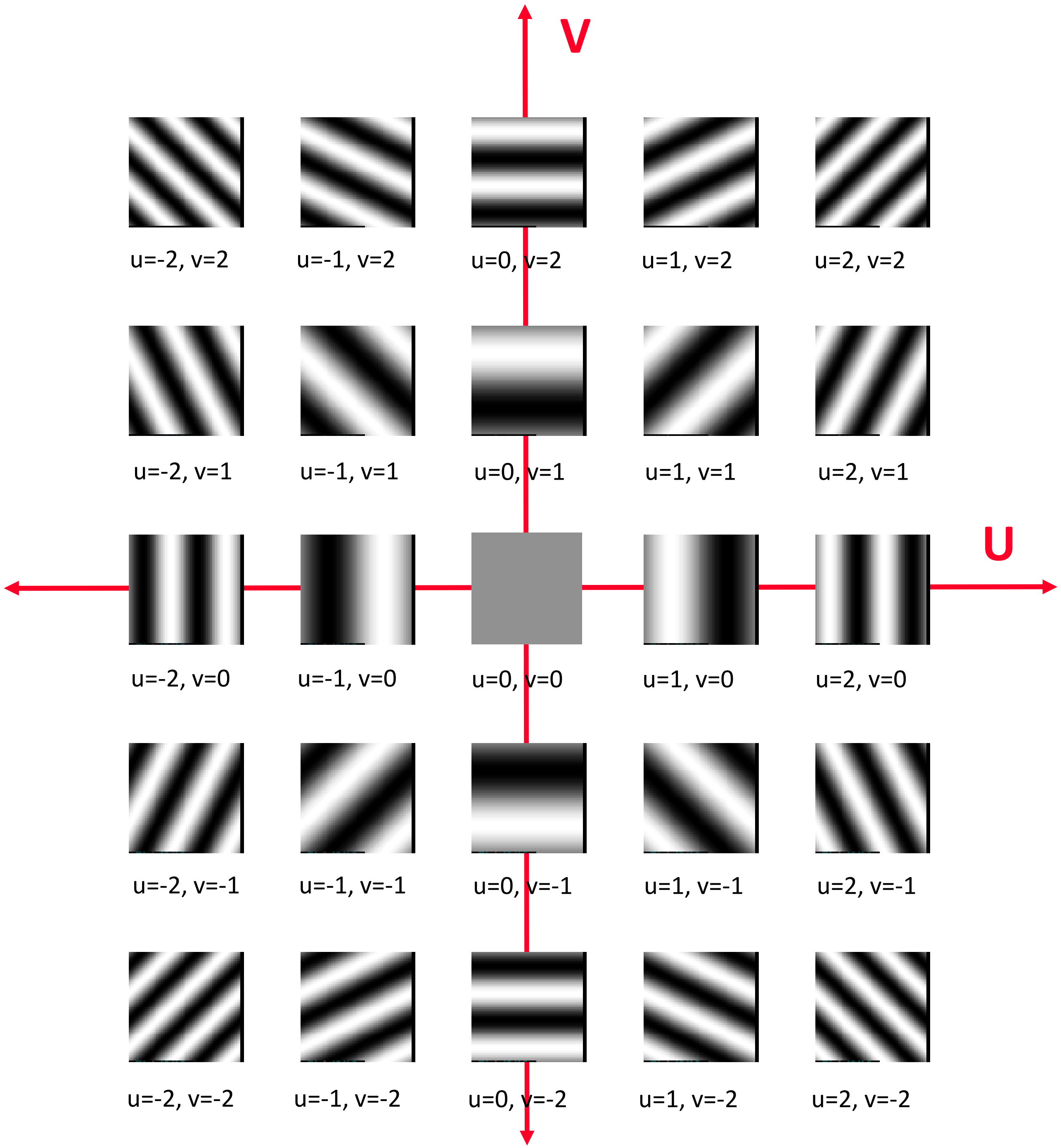
For a 2D image f(x,y) x=0..N-1, y=0..M-1, the DFT basis functions are 2D:

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

For frequency u,v the Fourier coefficient is:

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

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


V
U
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=-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

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, ..., N-1
v = 0, 1, 2, ..., M-1

Matlab: F=fft2(f);

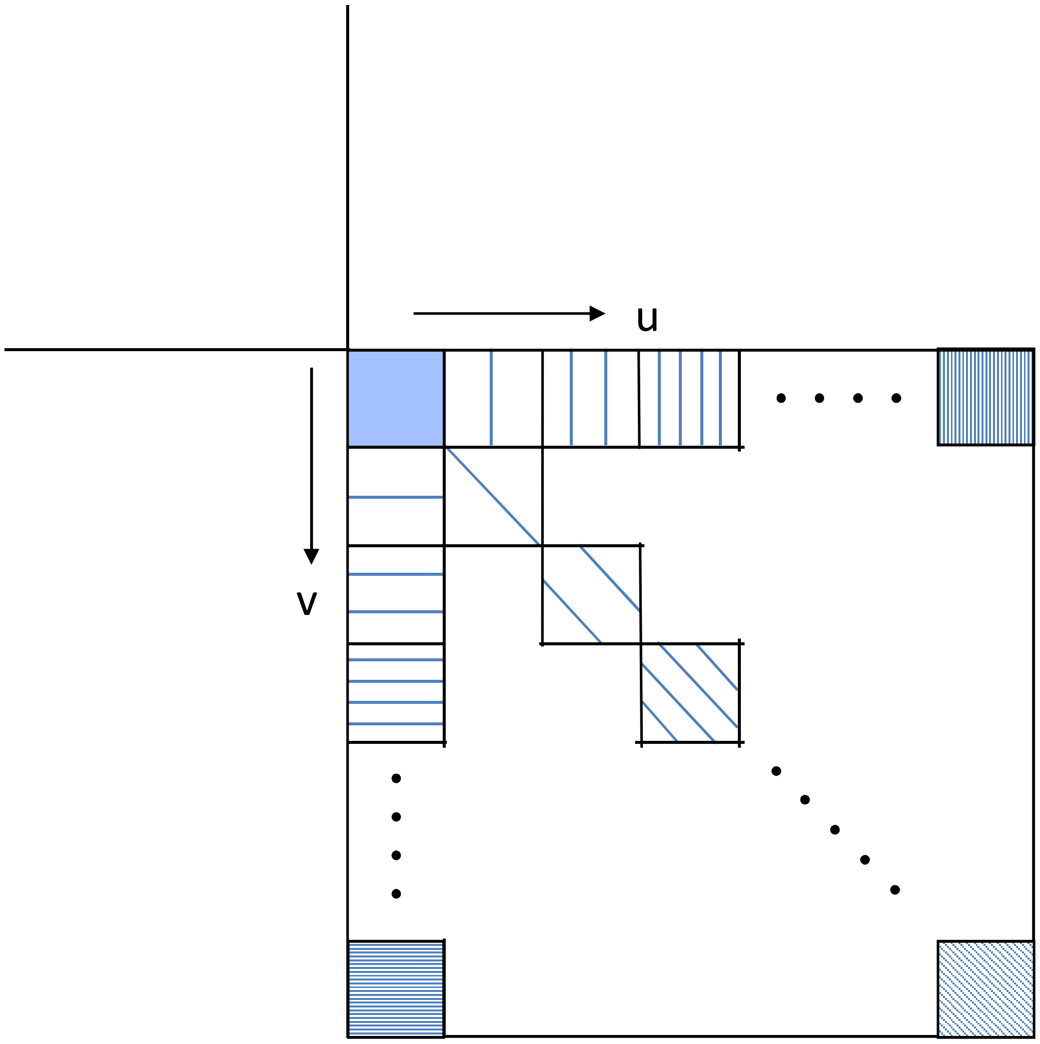
The Inverse Discrete Fourier Transform (IDFT) is defined as:

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

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

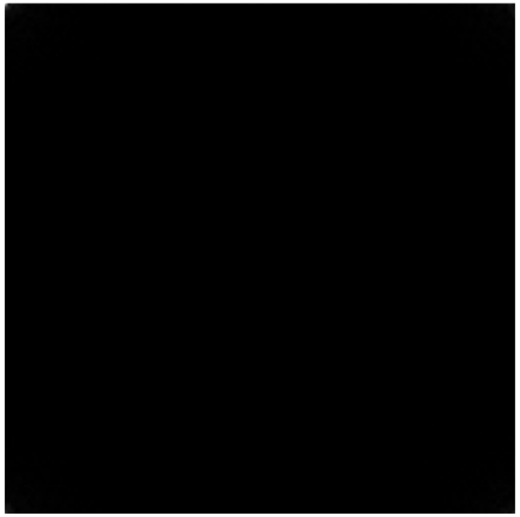
Matlab: f=ifft2(F);

Fourier Transform Image



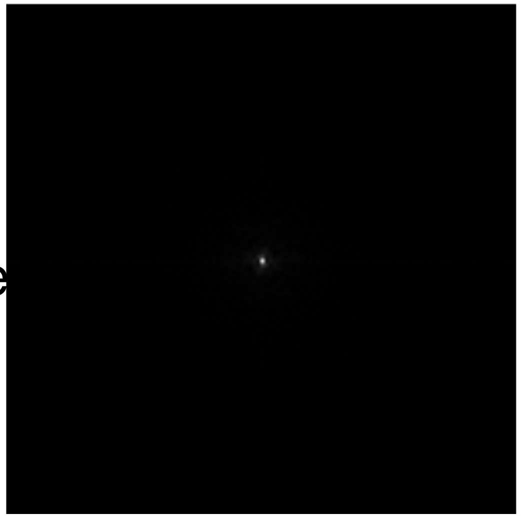
Fourier Transform – Image

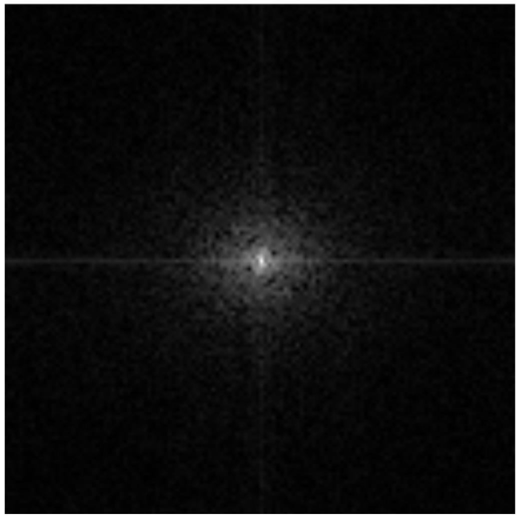
Original



Fourier Image
$|F(u,v)|$

Shifted
Fourier Image





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

Fourier Transform – Image

- 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]):

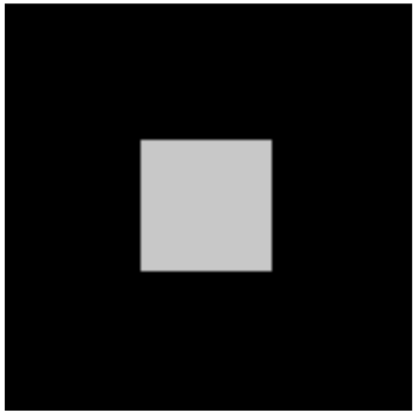
log(1+|F(u)|) = 4.62 1.61 1.01 0.69 0 0 0.69 1.01 1.61

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

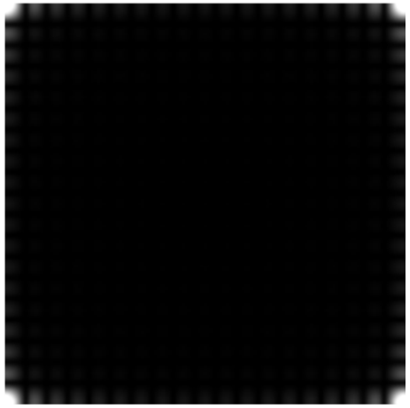
fftshift(log(1+|F(u)|) = 0 10 20 40 100 40 20 10 0

Fourier Transform – Image

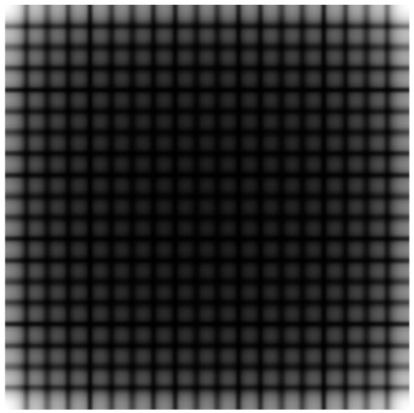
Original



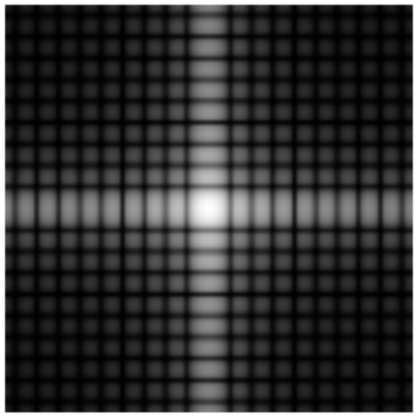
|F(u,v)|



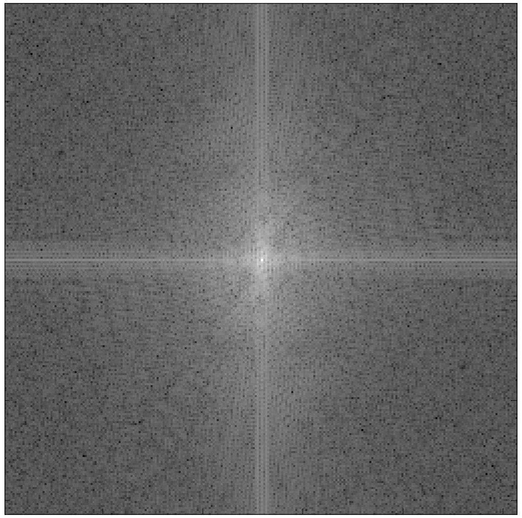
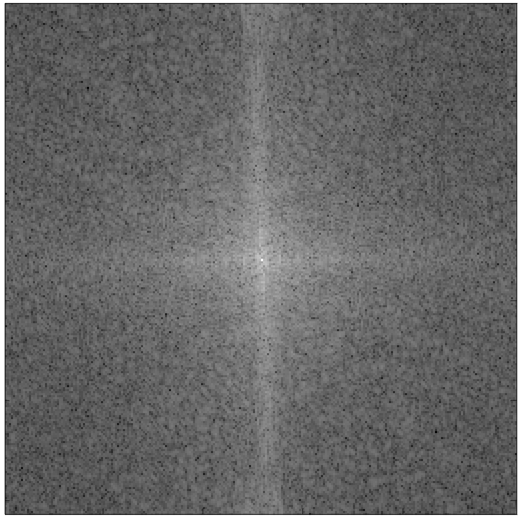
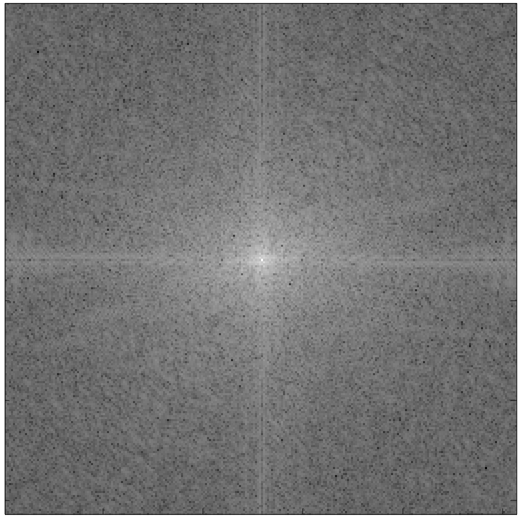
log(1 + |F(u,v)|)



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



Fourier Transform – Image

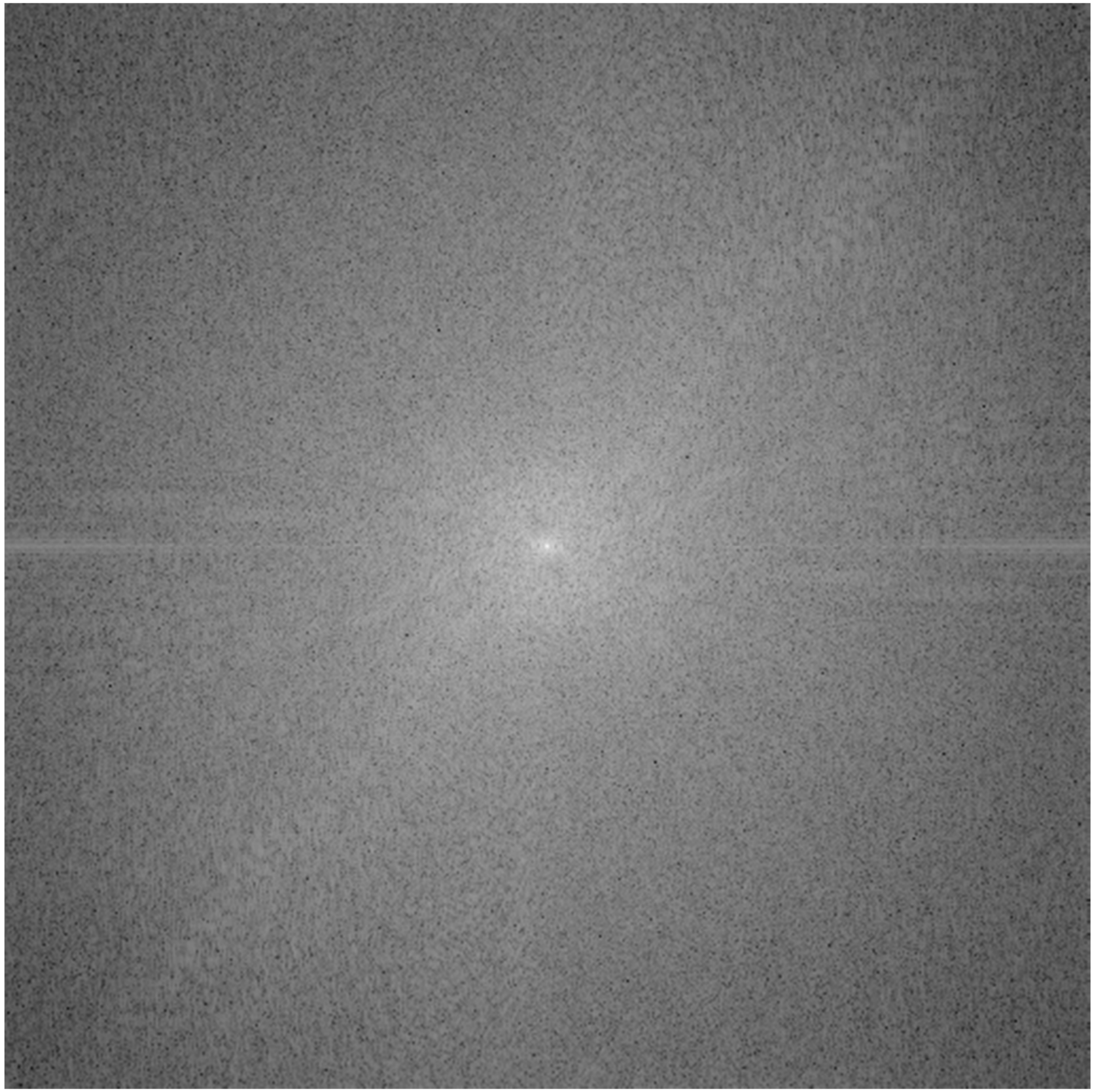
Fourier Transform – Image

- 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

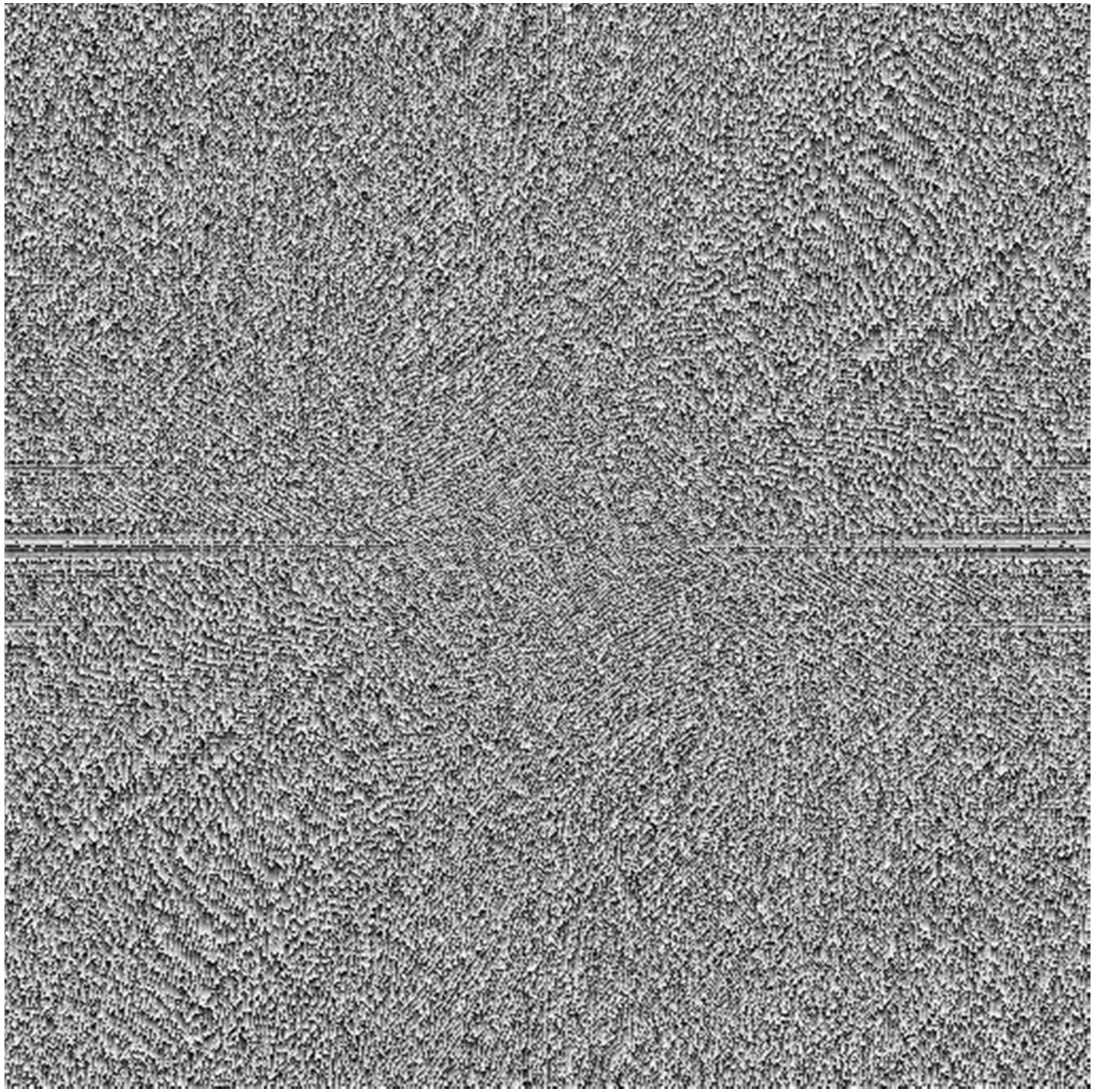
Magnitude transform of cheetah

Slide: Freeman & Durand



Magnitude transform of cheetah

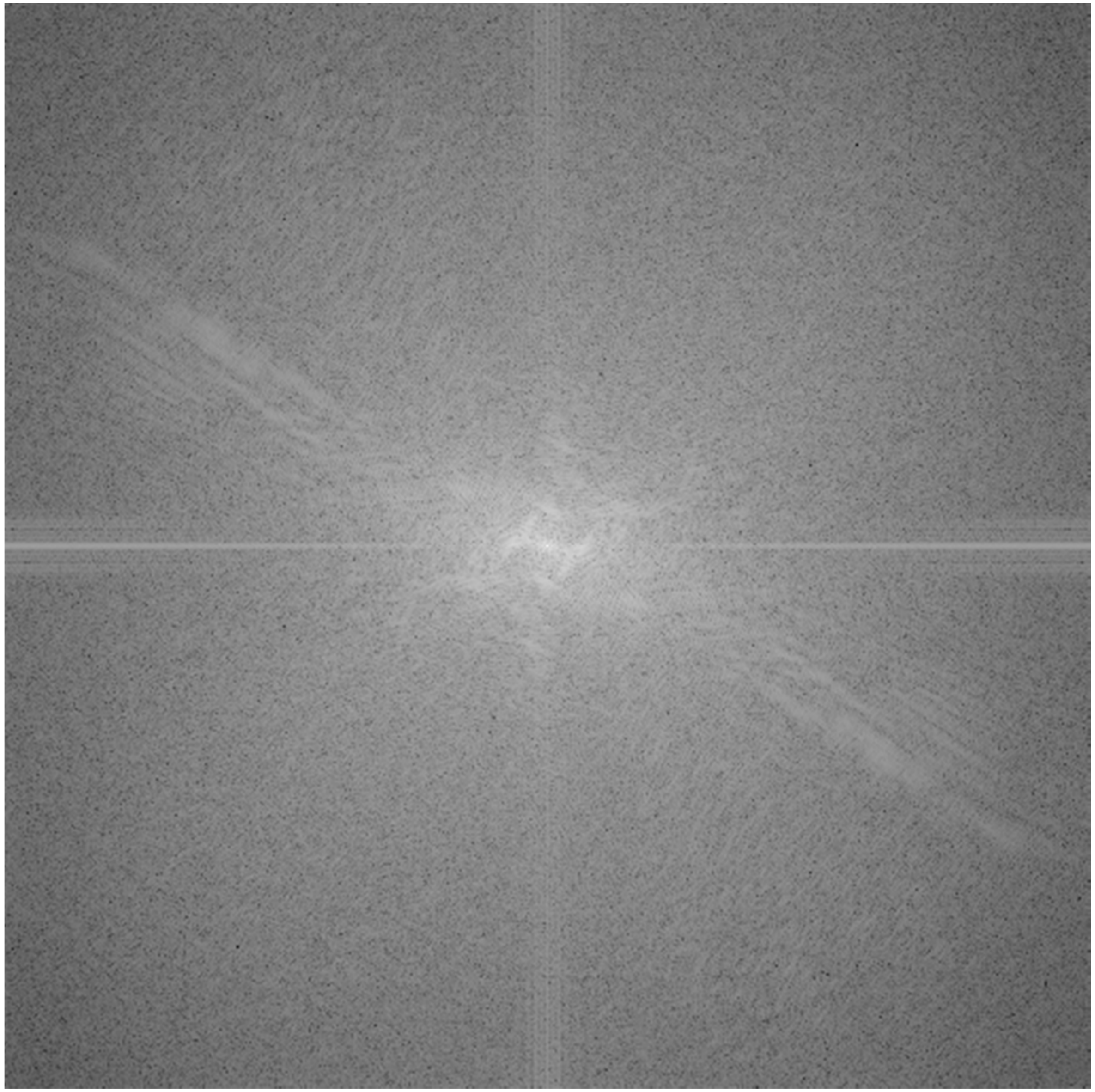
Slide: Freeman & Durand



Phase transform of cheetah

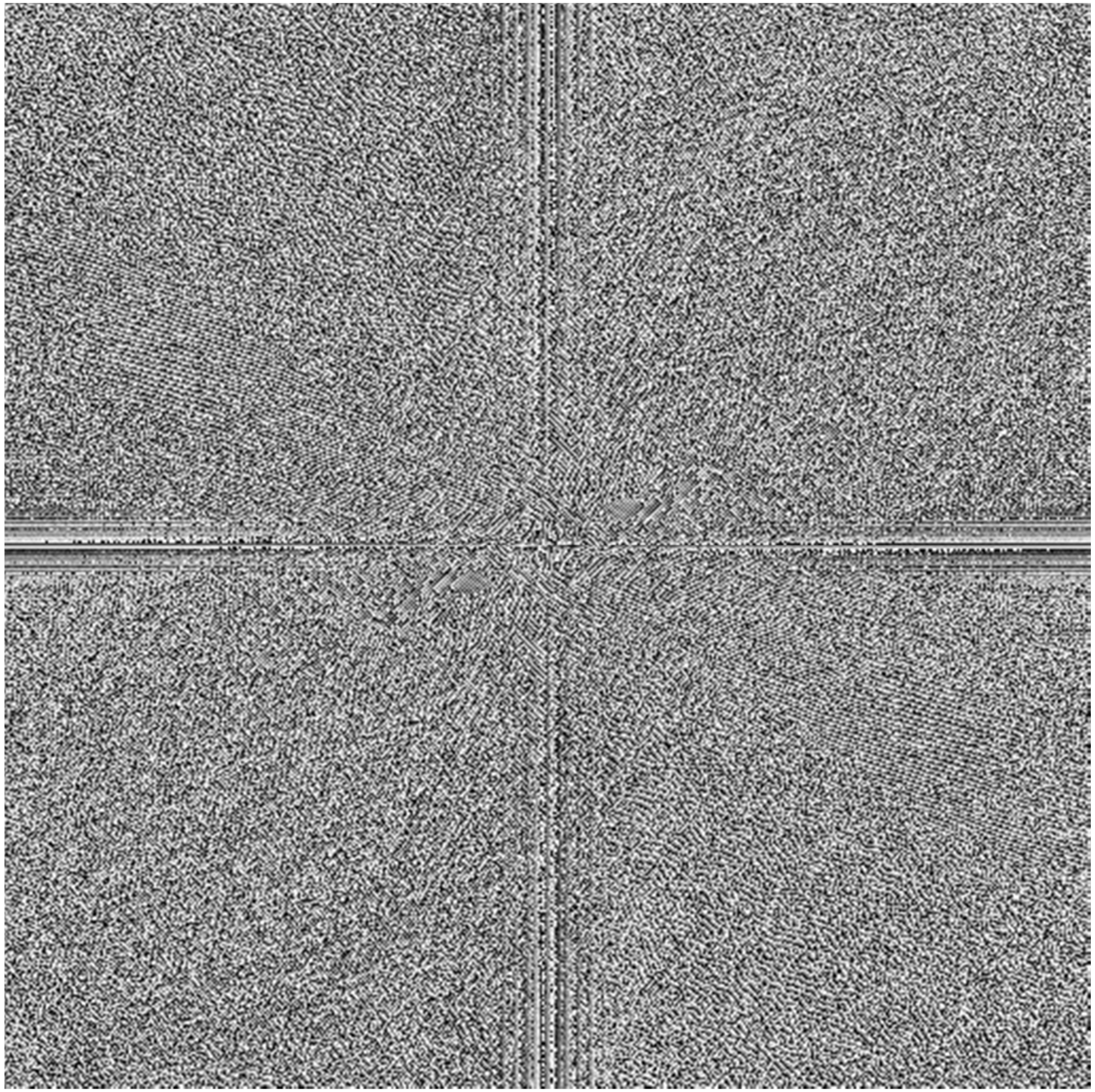
Slide: Freeman & Durand

Slide: Freeman & Durand



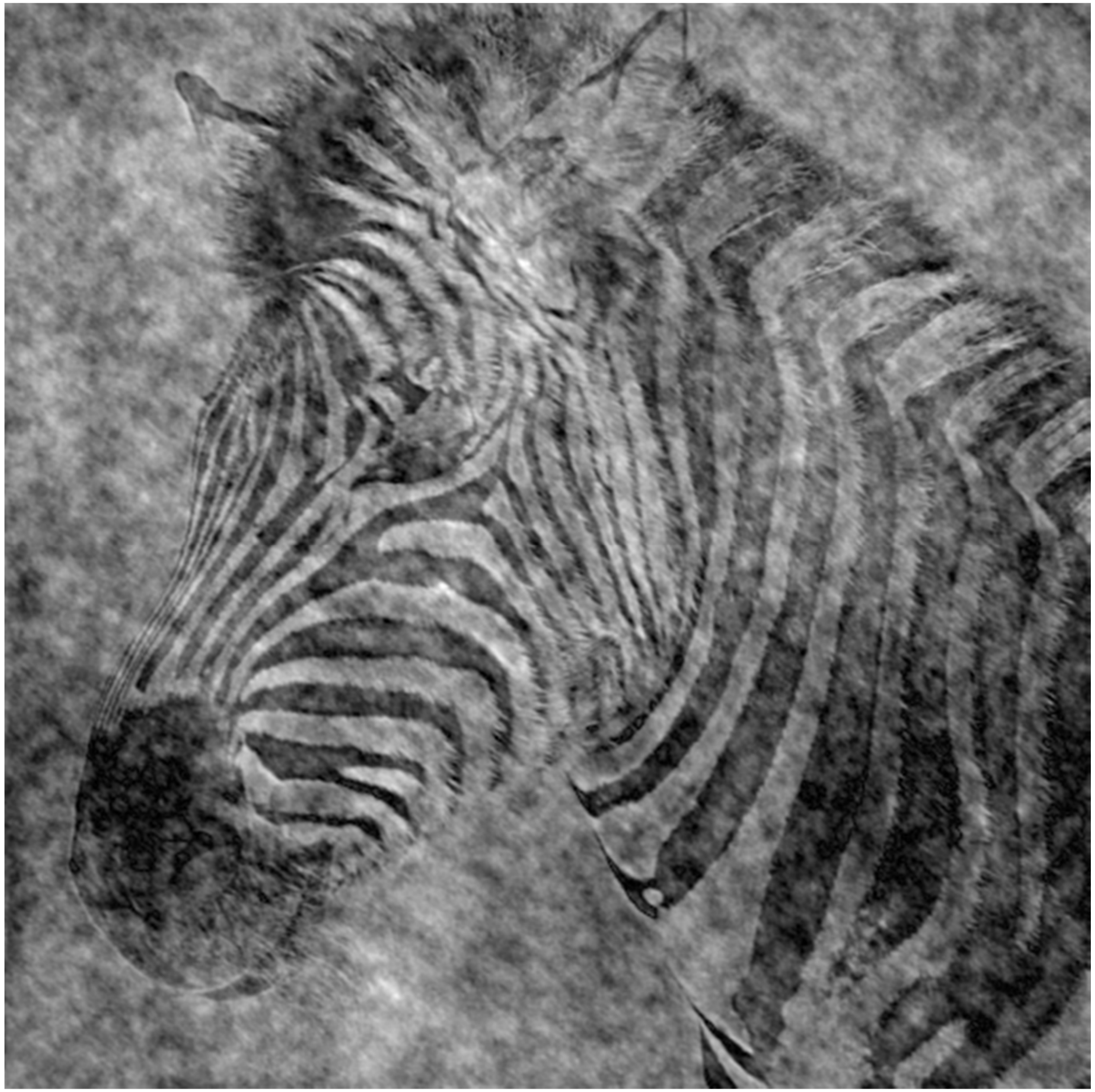
Magnitude transform of zebra

Slide: Freeman & Durand



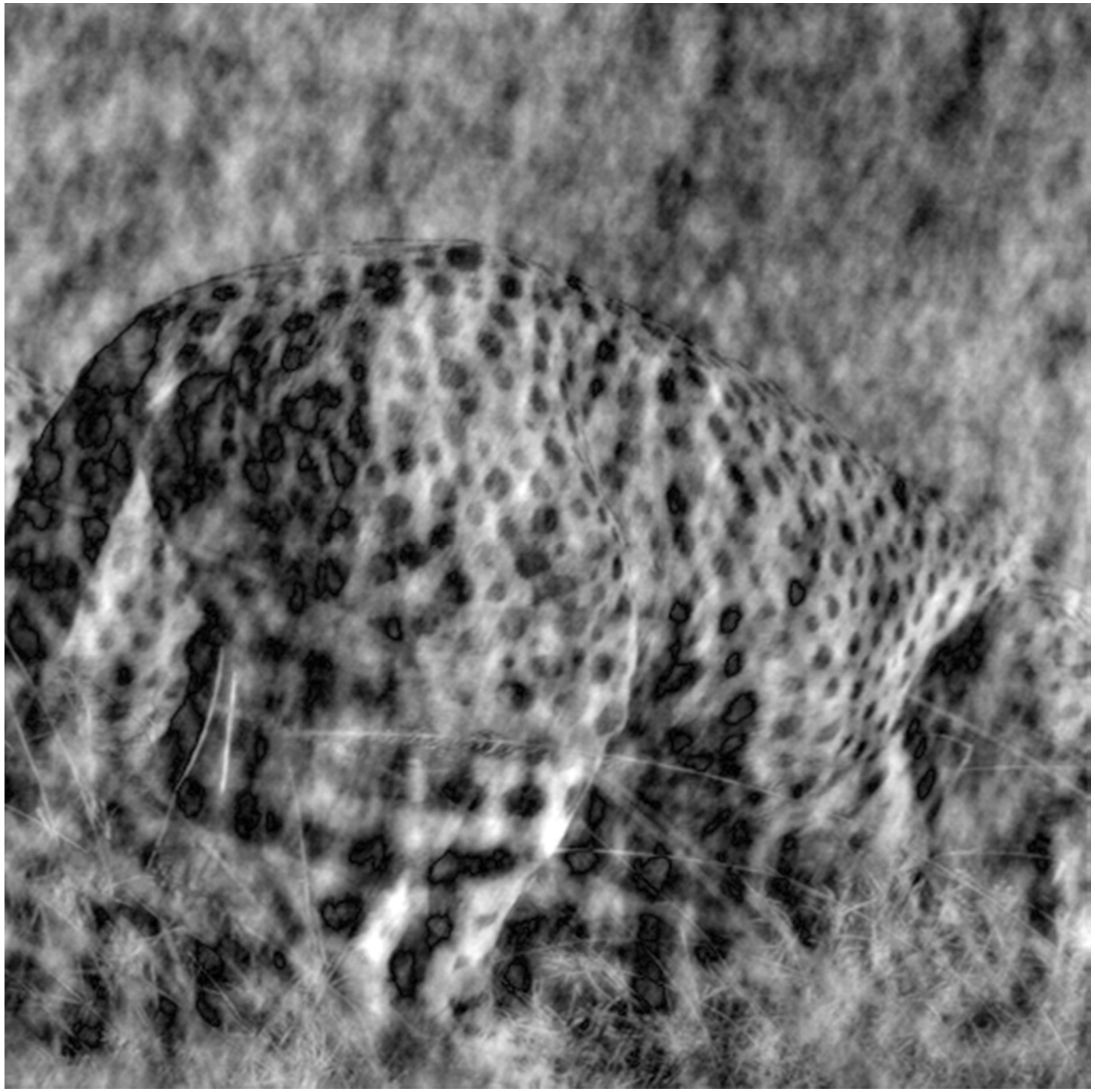
Phase transform of zebra

Slide: Freeman & Durand



Recon: Zebra Phase + Cheetah Magnitude

Slide: Freeman & Durand



Recon: Cheetah Phase + Zebra Magnitude

Slide: Freeman & Durand

Fourier Transform – Properties

- Linearity:

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

- Distributive (additivity):

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

- DC (average):

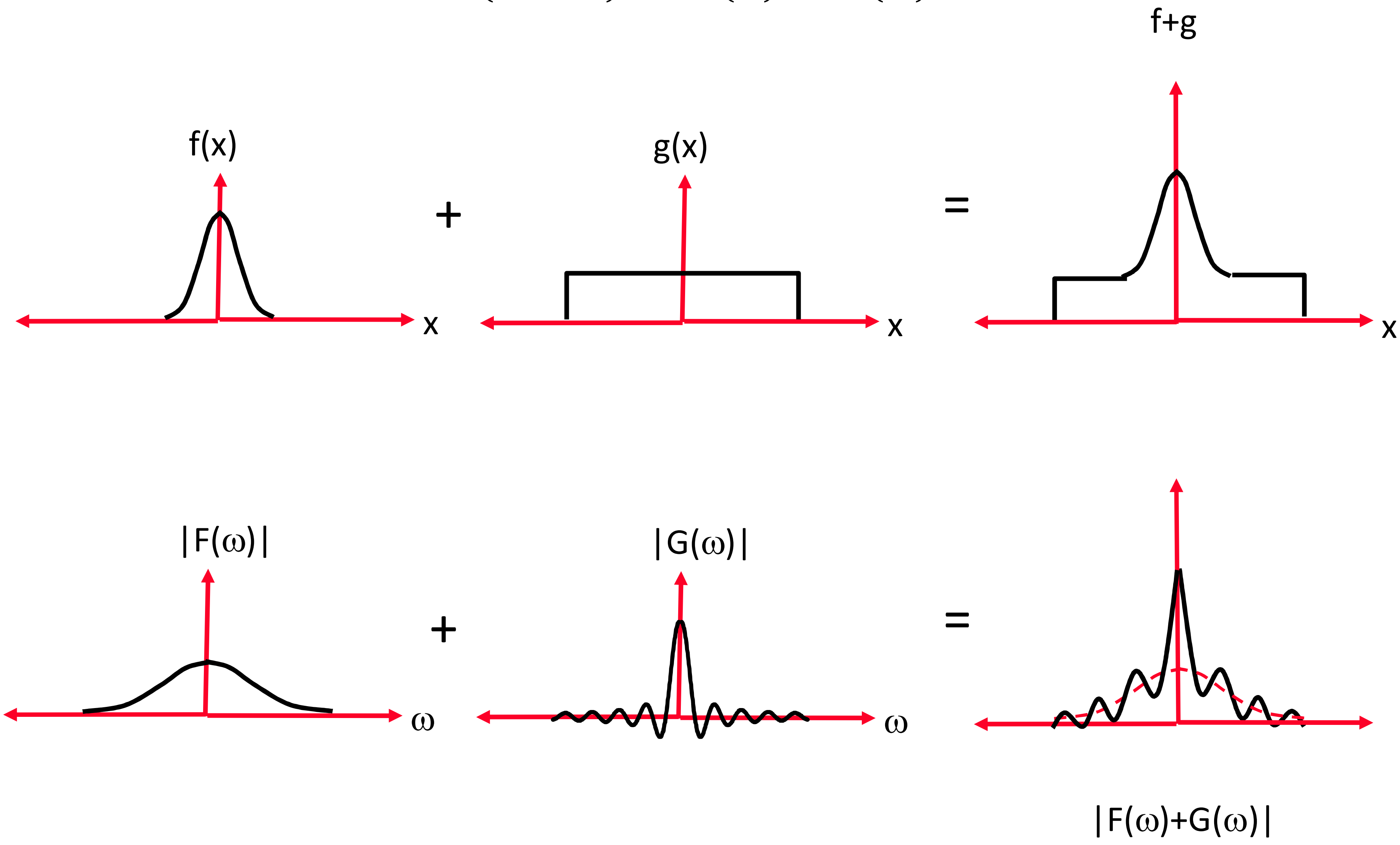
$$F(0,0) = \sum_x \sum_y f(x,y) e^0$$

- Parseval

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

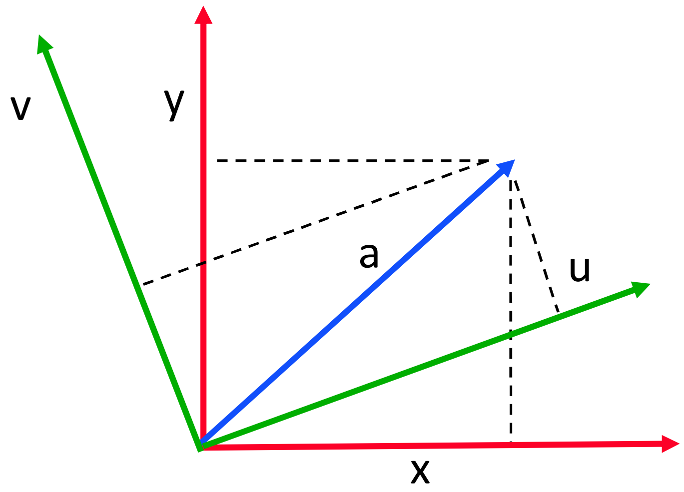
Distributive:

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



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,

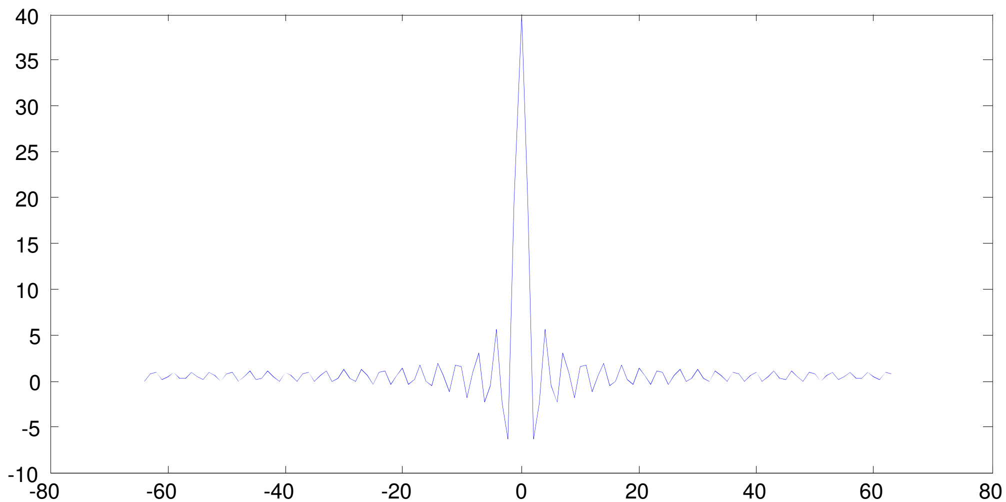
$$F(u,v) = F^*(-u,-v) \quad \text{thus} \quad |F(u,v)| = |F(-u,-v)|$$

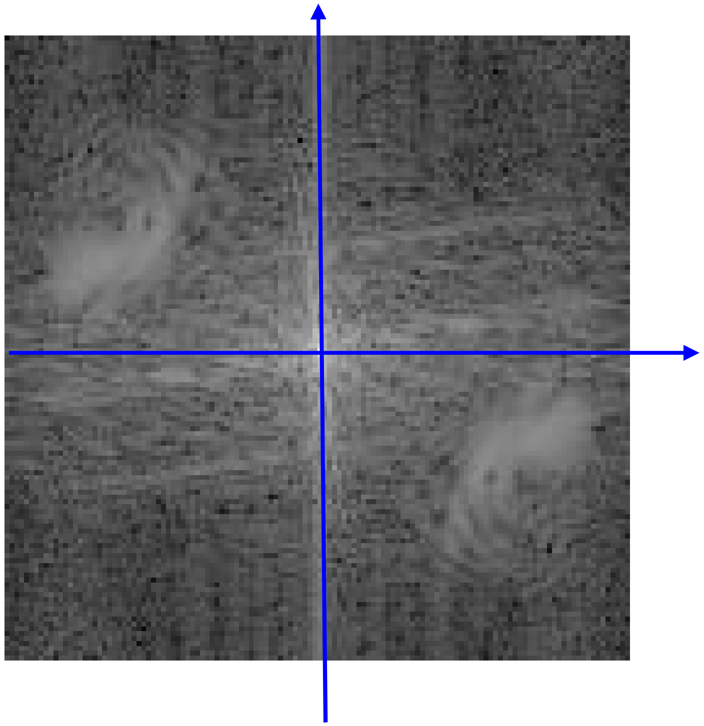
- Cyclic:
  if f(x,y) is discrete

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

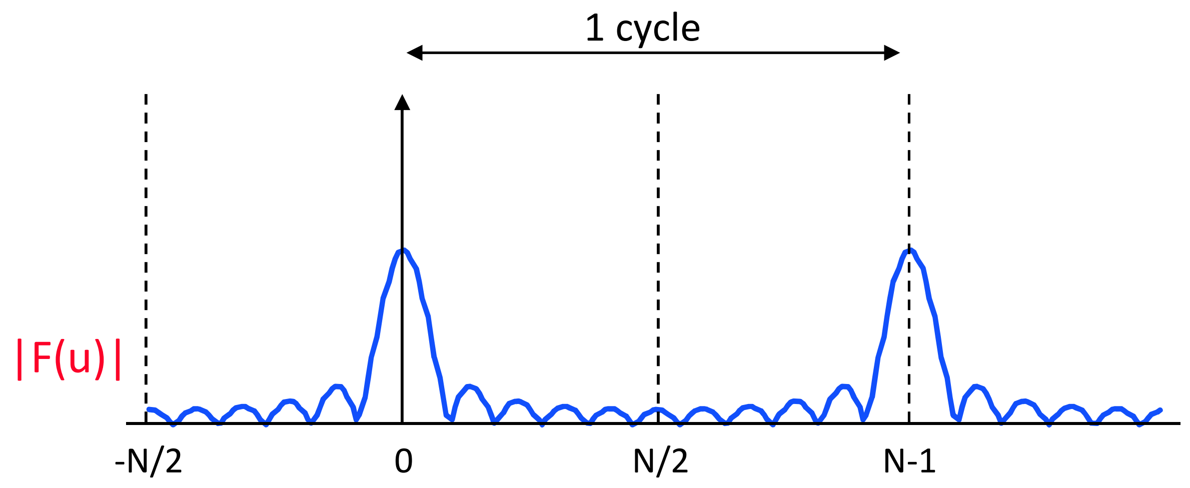
Symmetry of FT (for real signals):

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





Cyclic and Symmetry of FT :

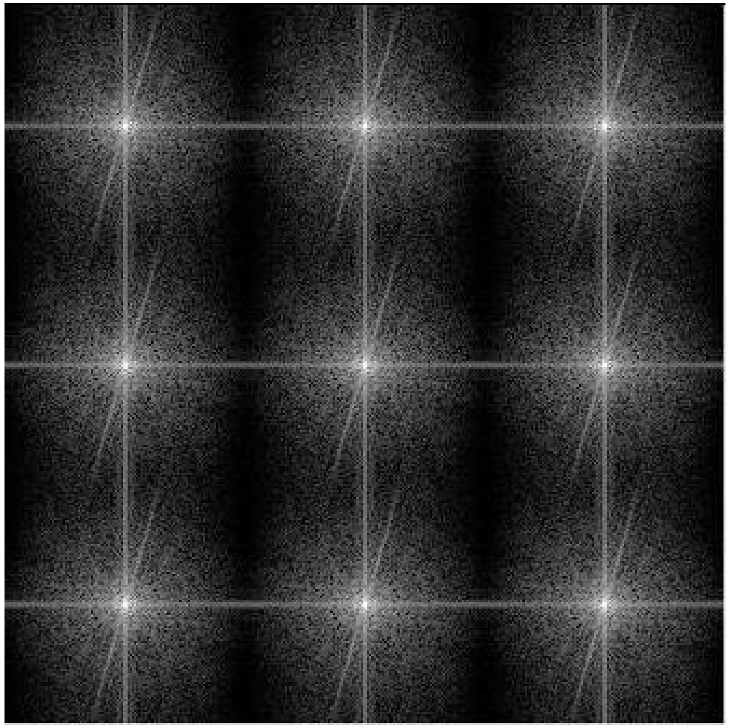


Due to replicas: $F(k)=F(N+k)$

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:

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

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

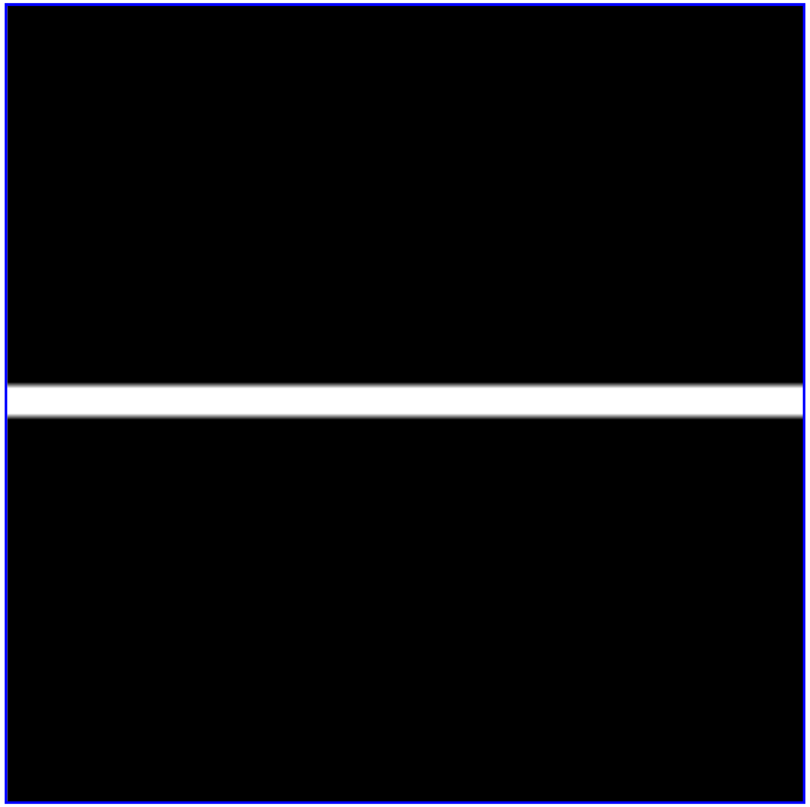
Thus, performing a 2D Fourier Transform is equivalent to performing 2 1D transforms:

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

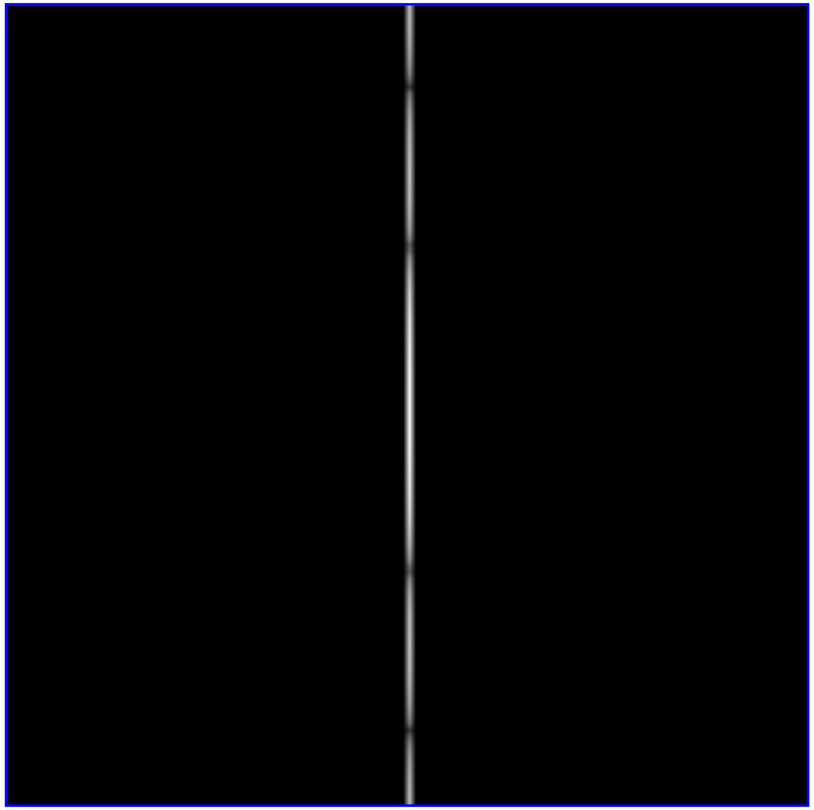


Image Transformations

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:

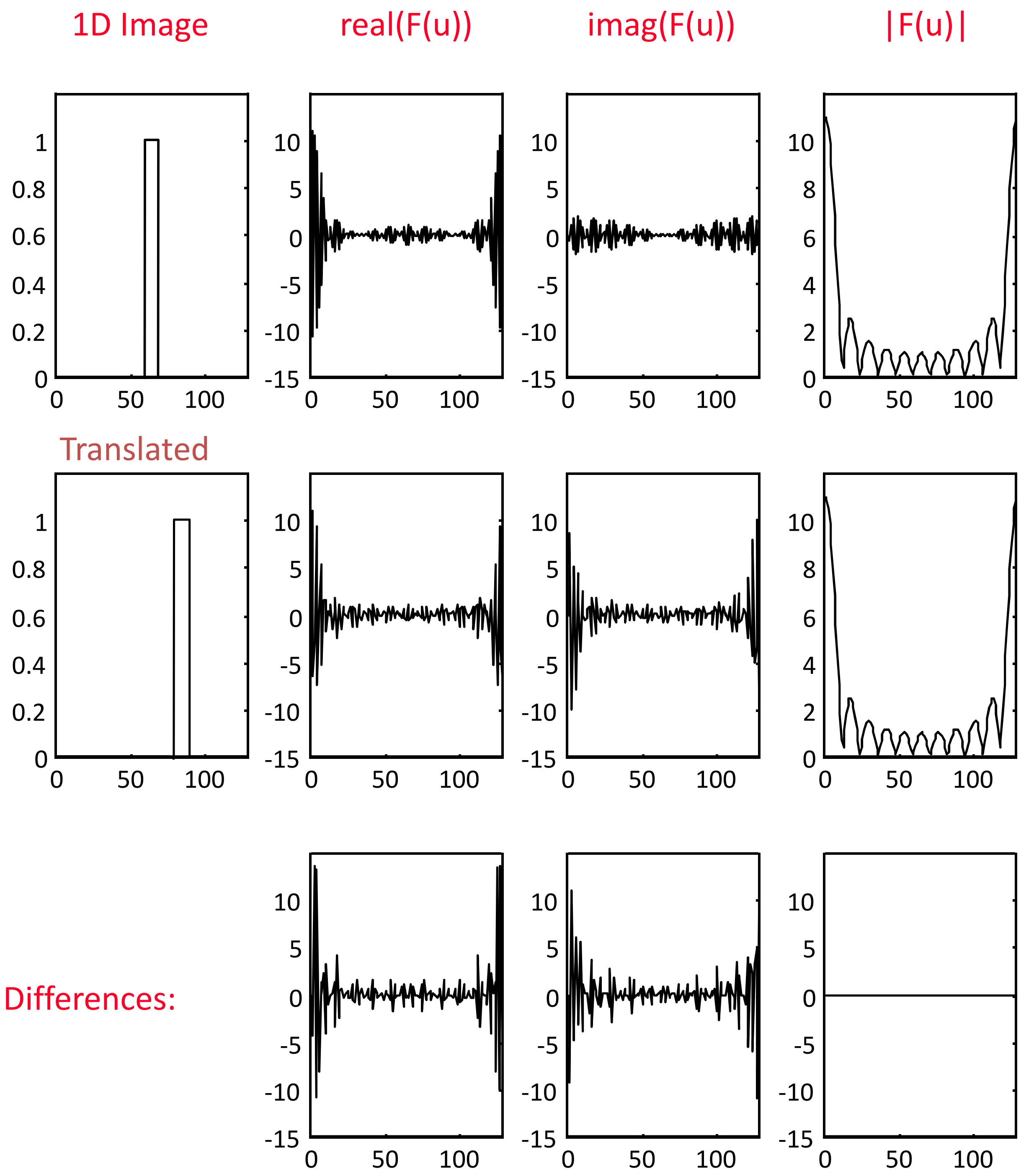


Image Transformations

Scaling:

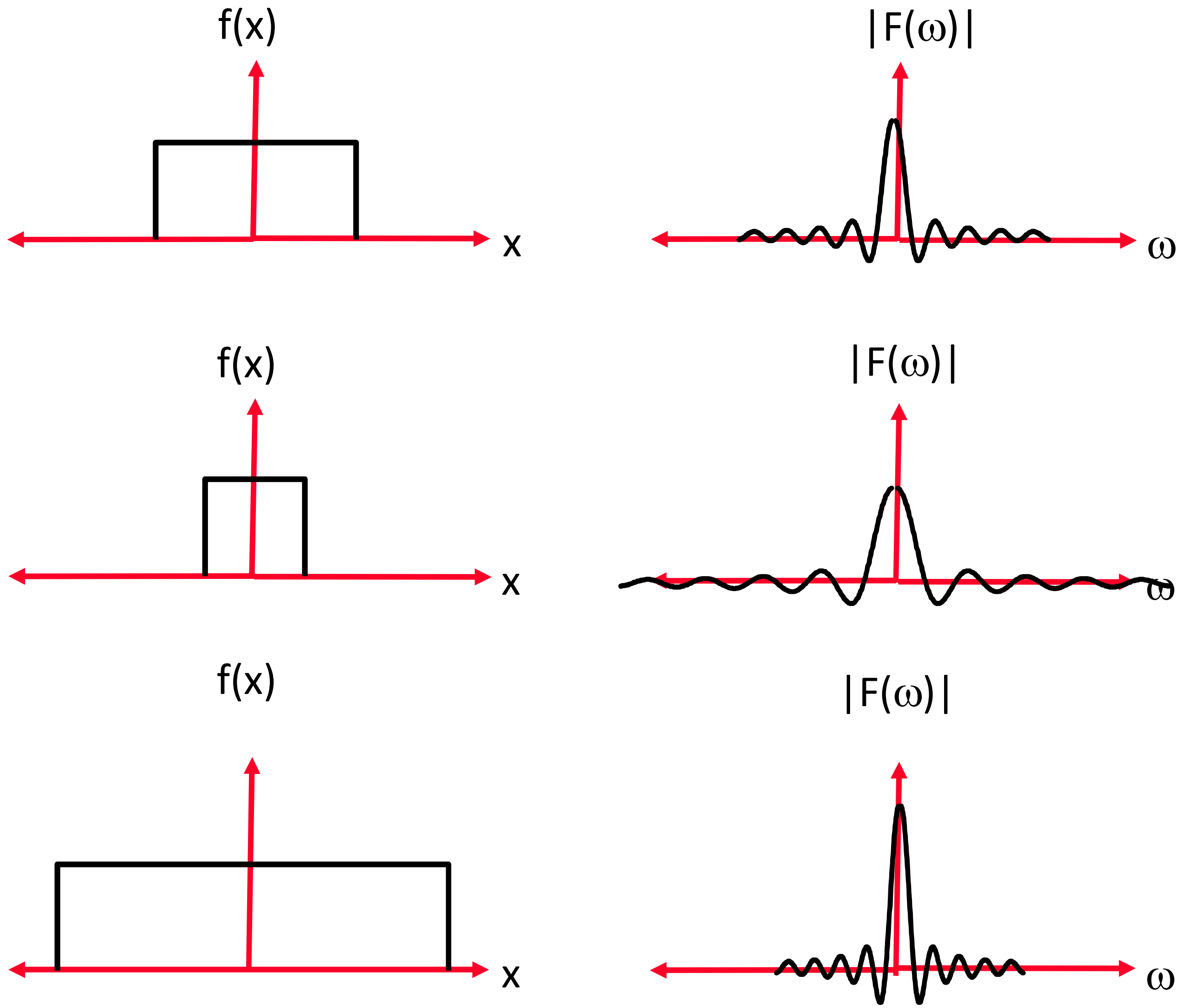
$$\tilde{F}\left[f\left(a\,x, b\,y\right)\right] = \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right)$$

Rotation:

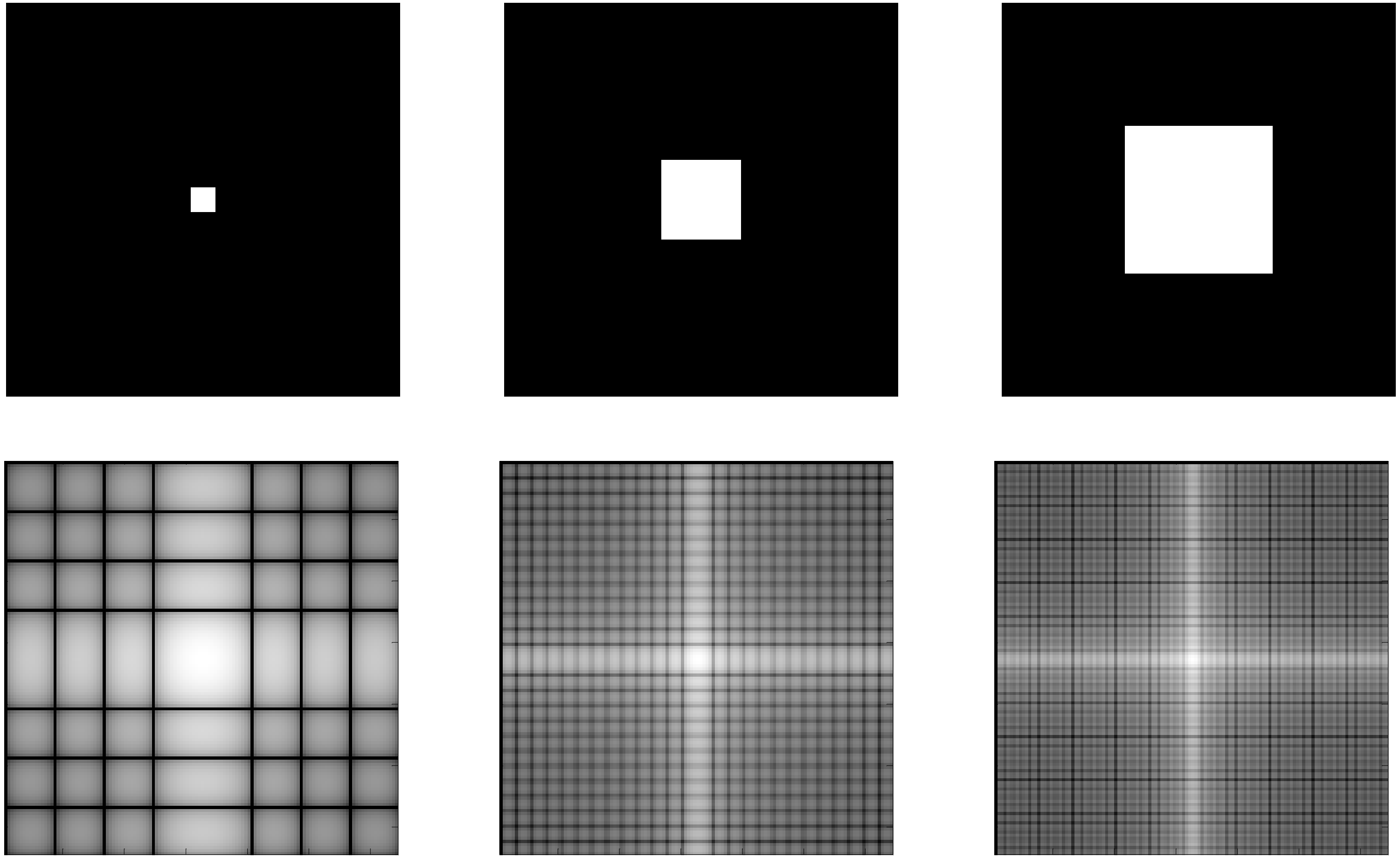
Rotation of f(x,y) by θ $\rightarrow$ rotation of F(u,v) by θ

Change of Scale:

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

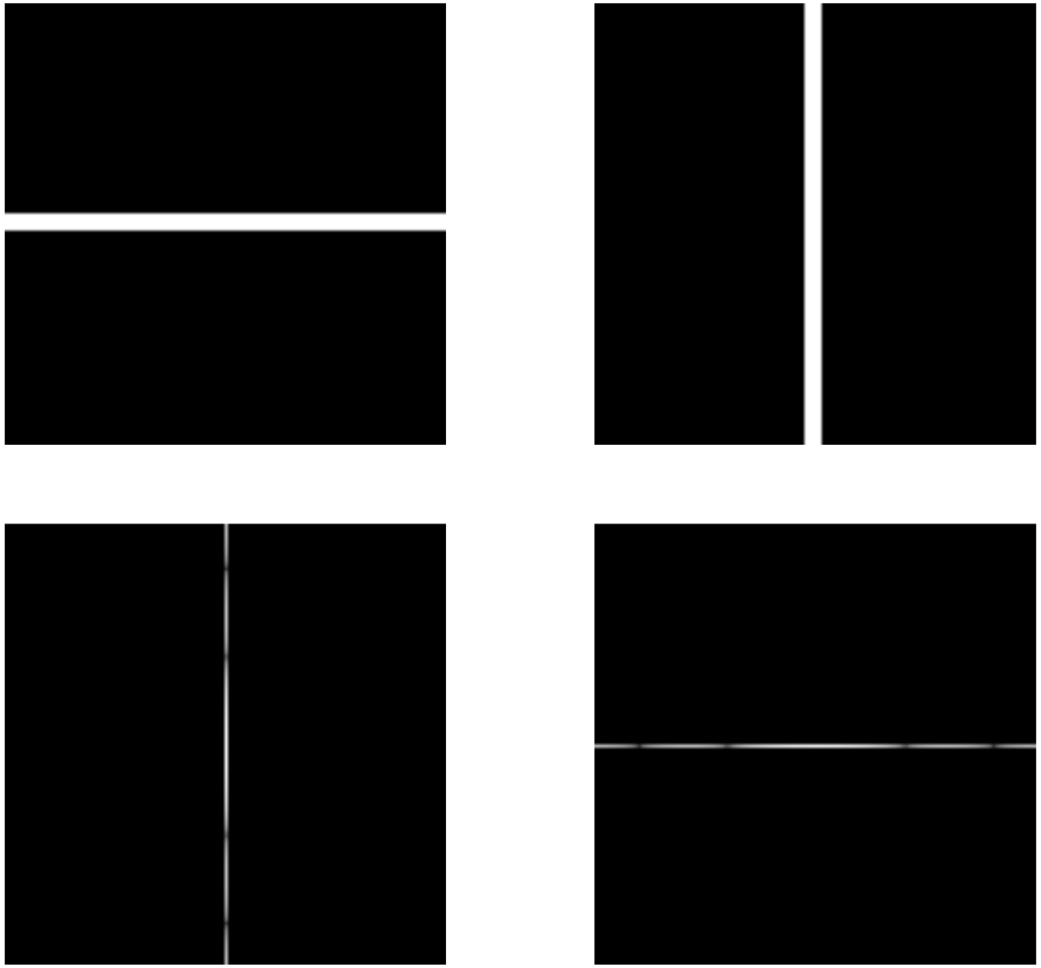


Change of Scale:



Rotation - Example

2D Image 2D Image - Rotated

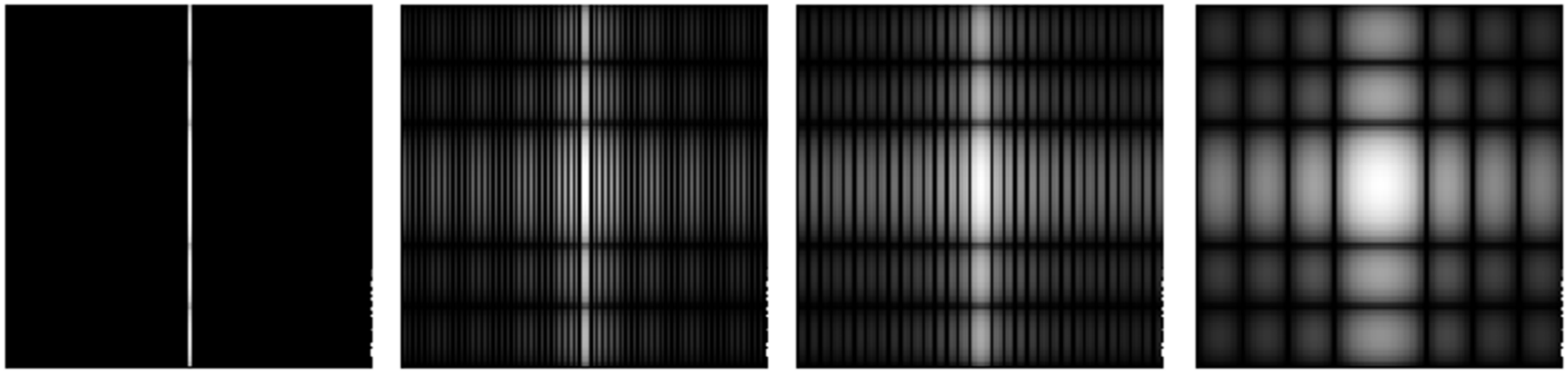


Fourier Spectrum Fourier Spectrum

Fourier Transform – Examples

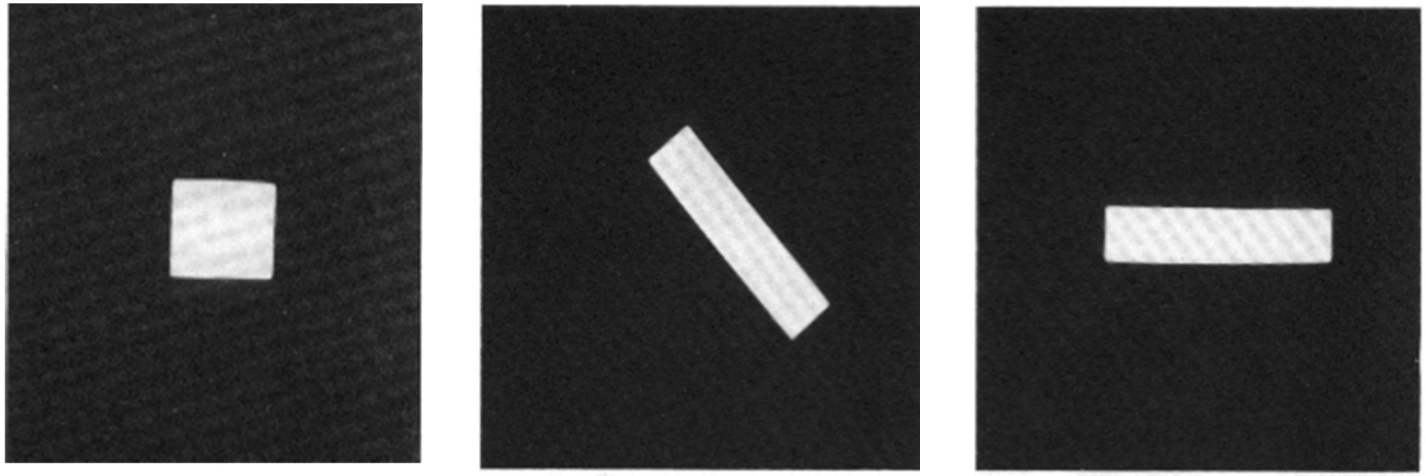
Image Domain

Frequency Domain

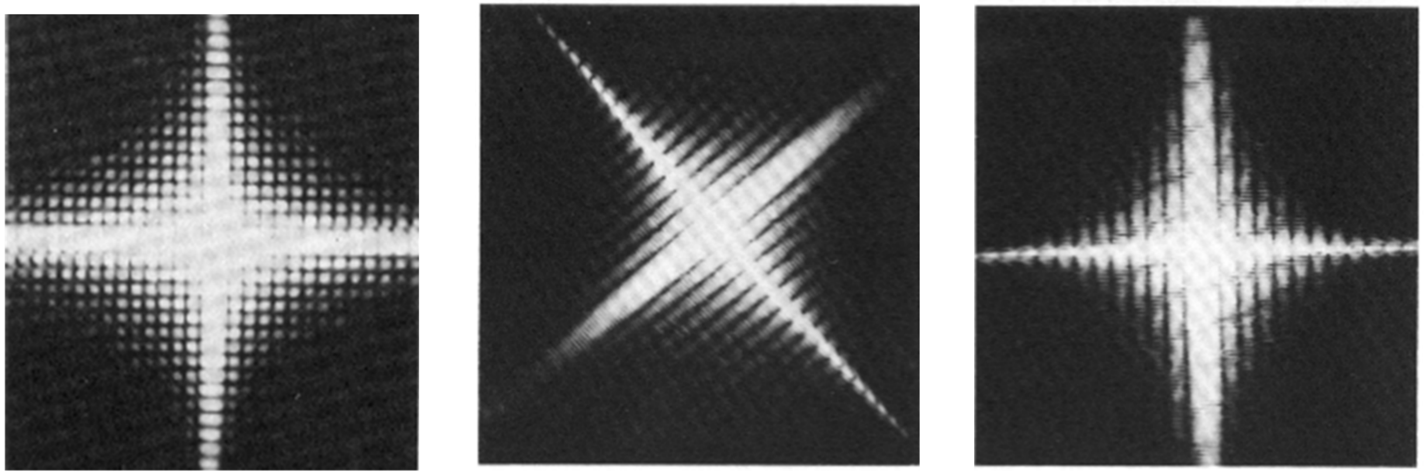


Fourier Transform – Examples

Image Domain

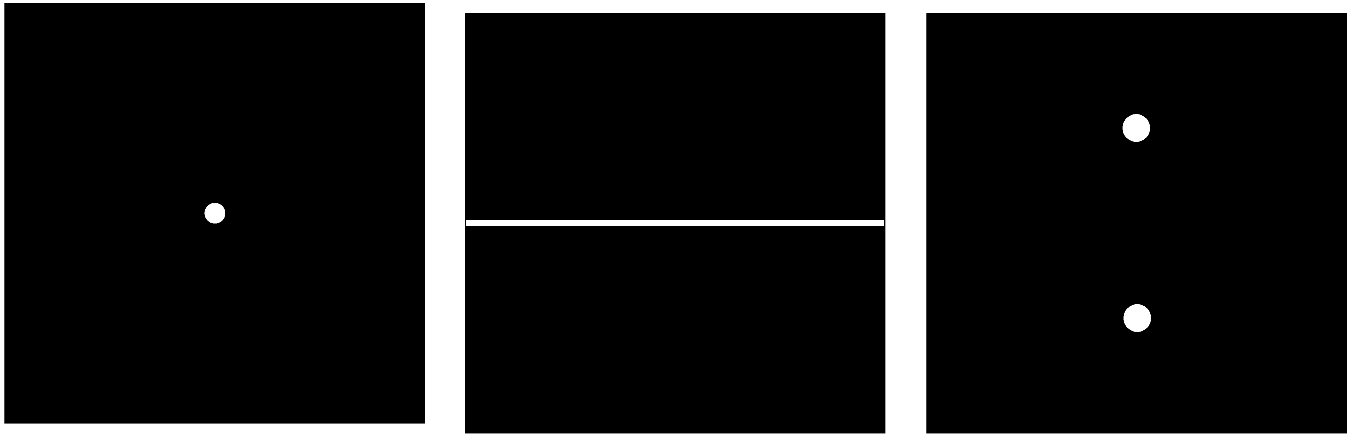


Frequency Domain

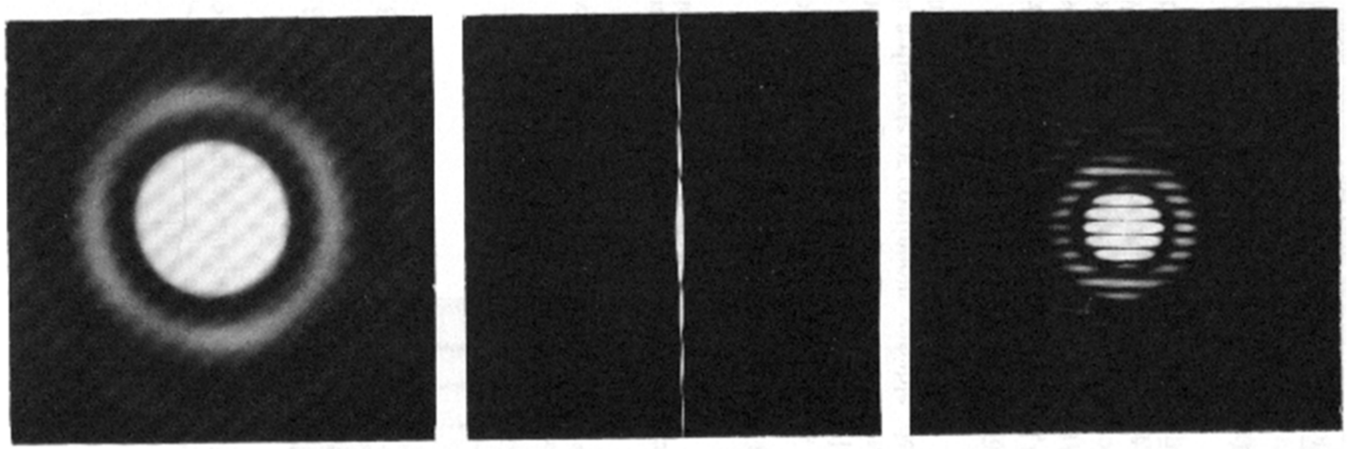


Fourier Transform – Examples

Image Domain

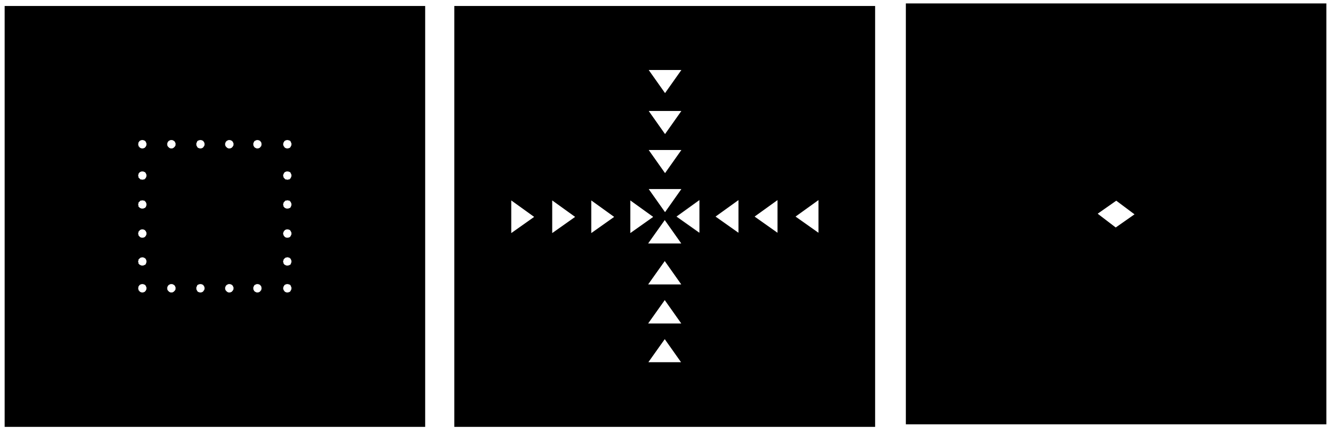


Frequency Domain

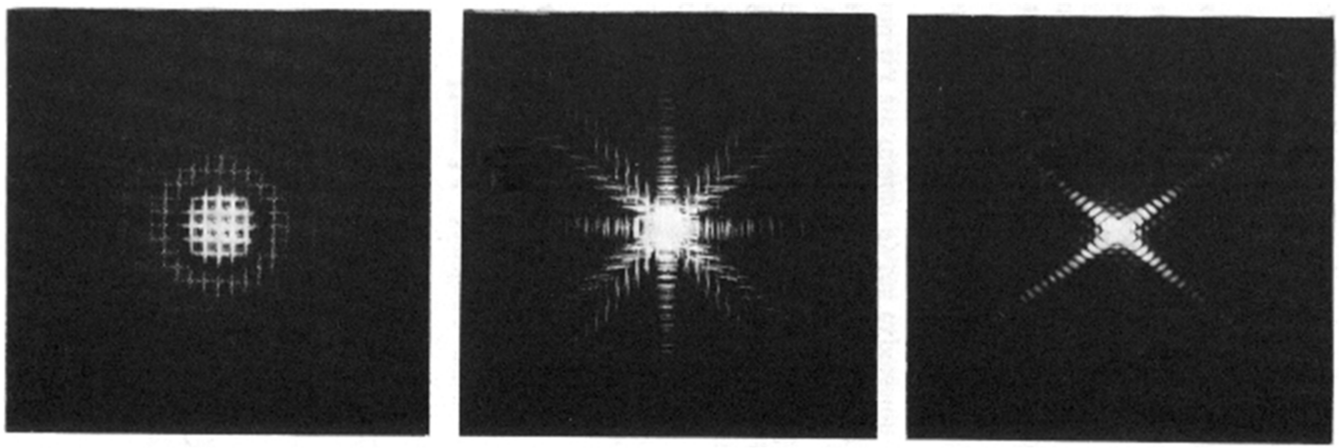


Fourier Transform – Examples

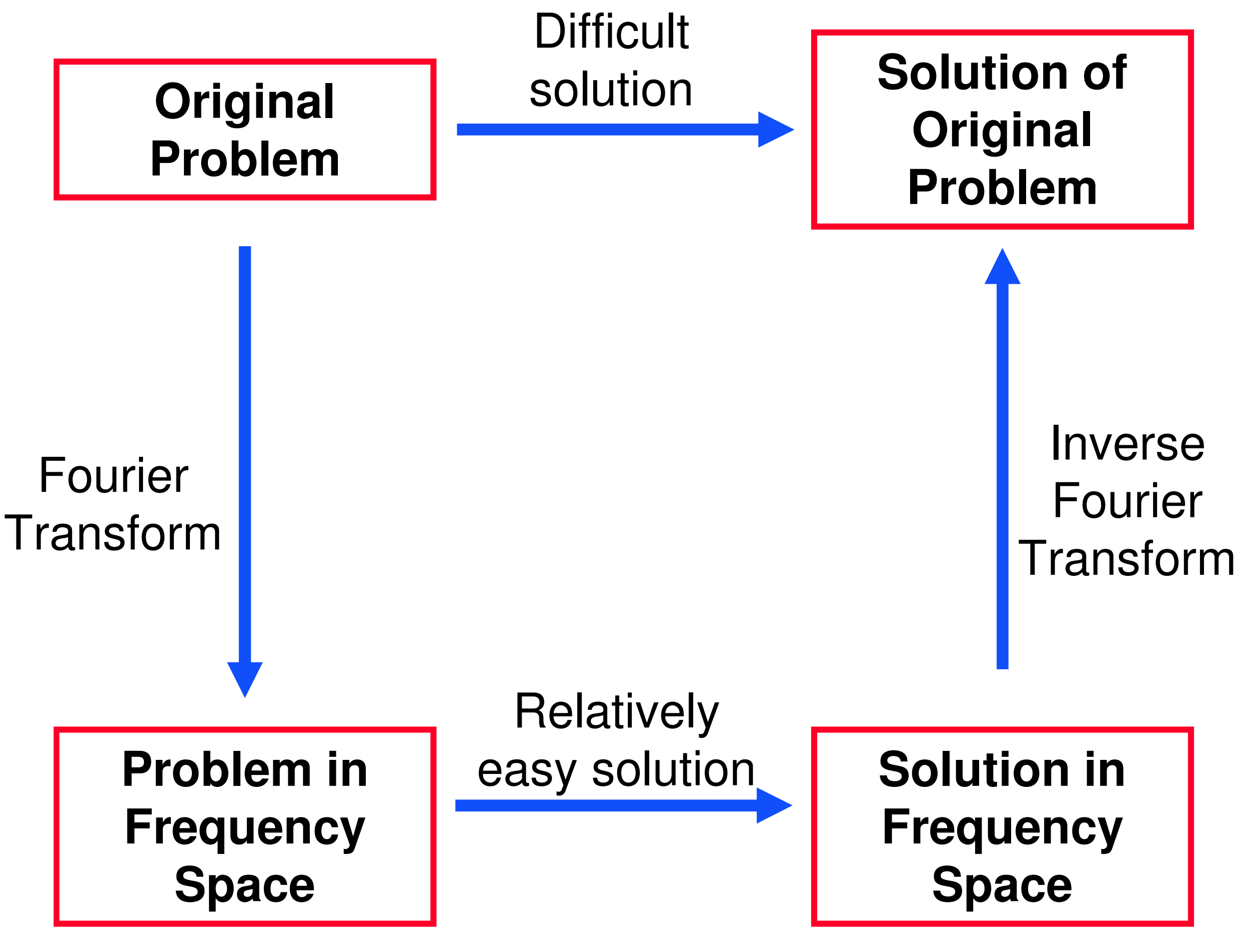
Image Domain



Frequency Domain



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

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

and likewise

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

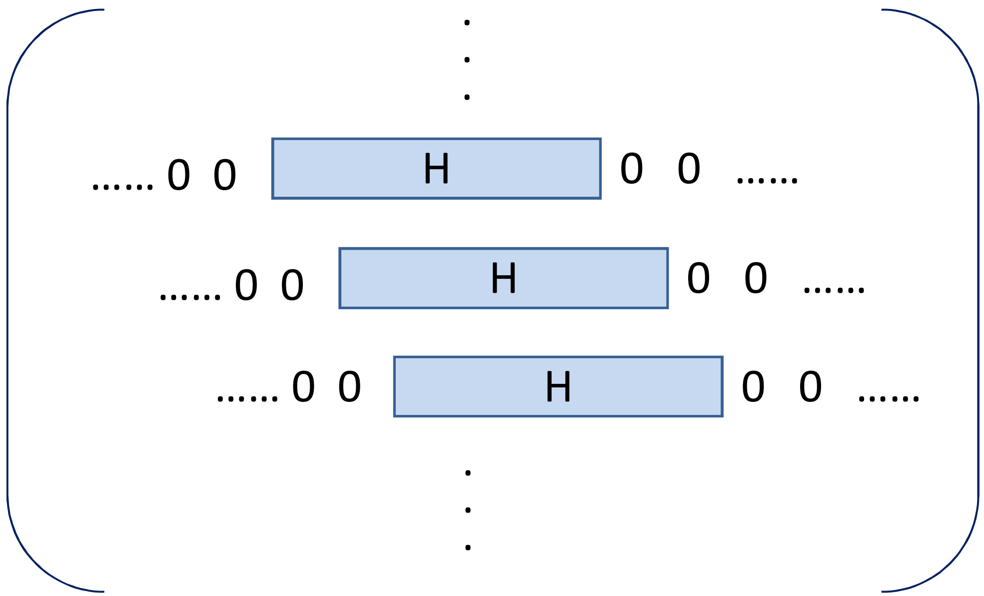
The Convolution Theorem - Proof

Convolution can be represented as a matrix multiplication:

$$y=Ax$$

where *A* is a circulant matrix.

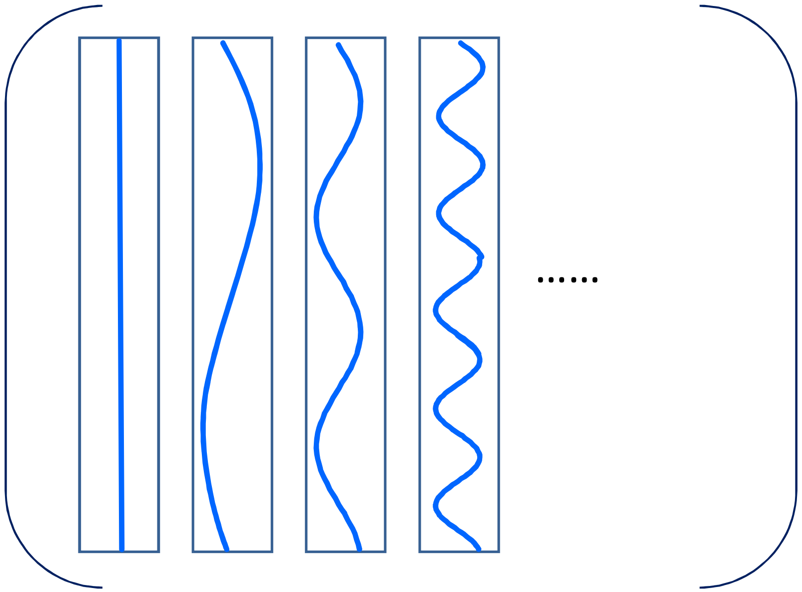
A =



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: $F_{nm} = \frac{1}{\sqrt{N}} e^{\frac{2\pi i mn}{N}} = F_{mn}$ 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.

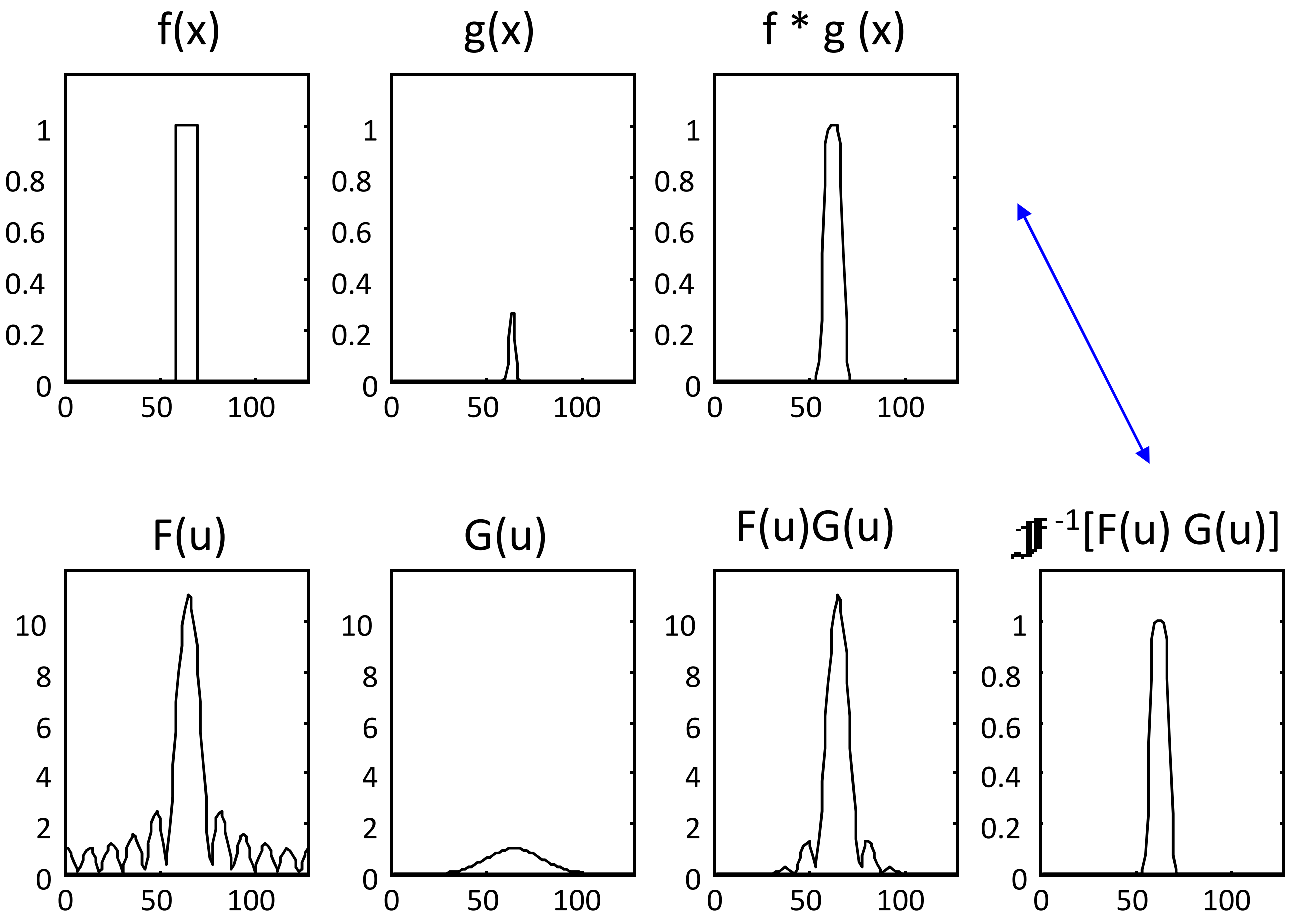
The Convolution Theorem - Proof

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

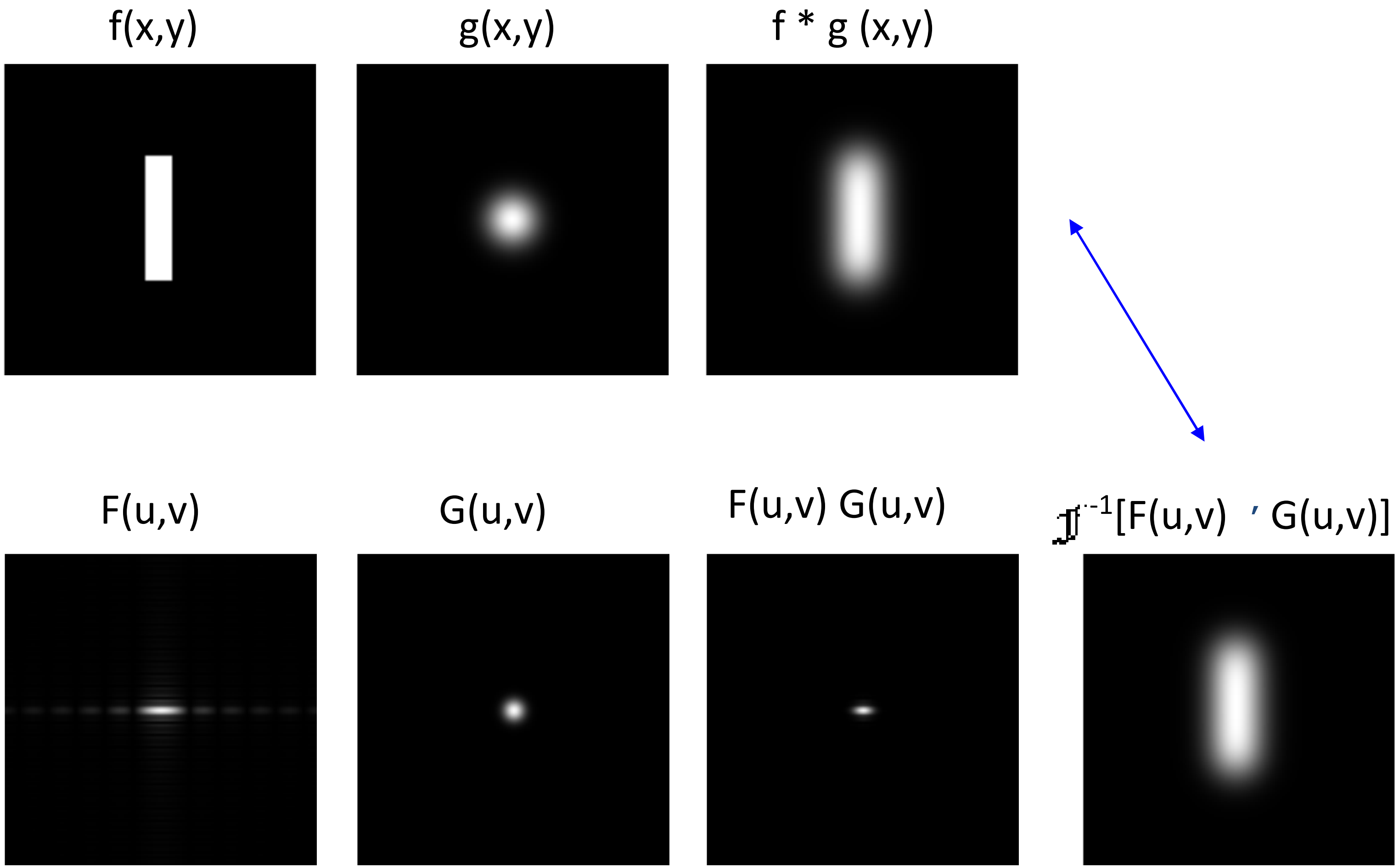
$$Y = D\ X$$

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

The Convolution Theorem - Example

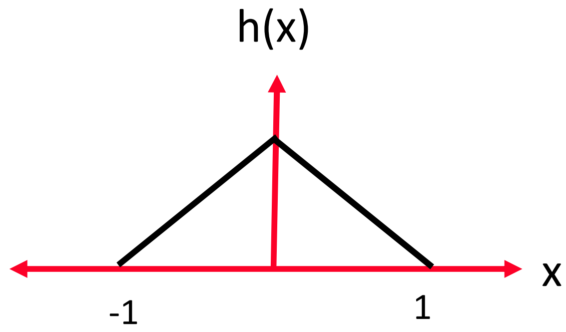


The Convolution Theorem - Example

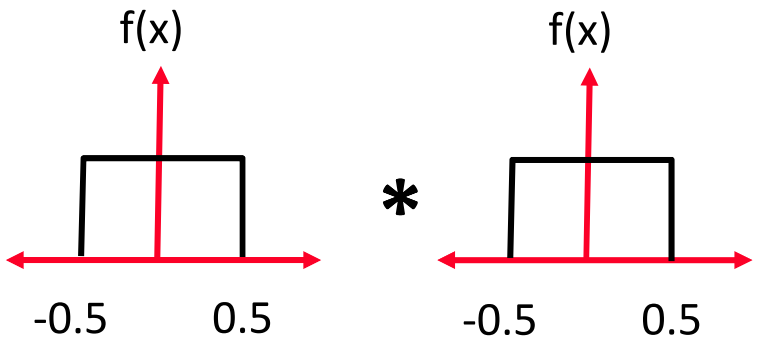


Convolution Theorem - Example

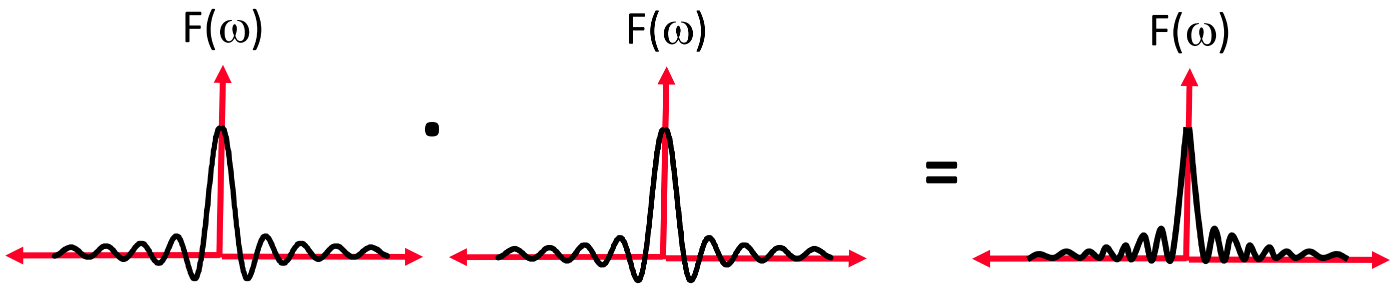
Example: What is the Fourier Transform of:



h(x) =



H(ω) =



Convolution Theorem - Example

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

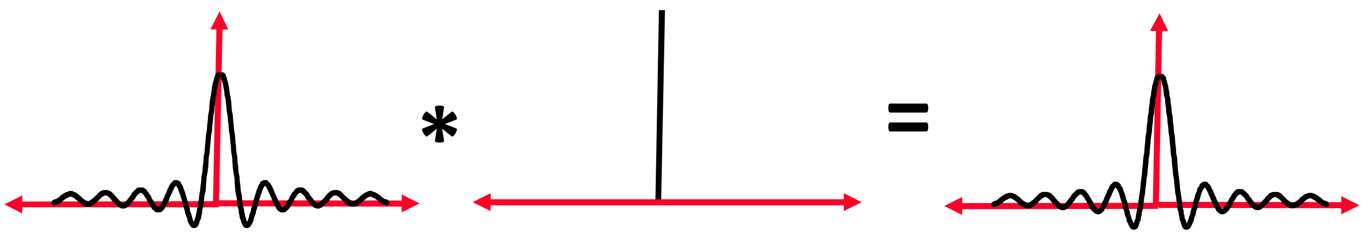
$$\delta(x) = \begin{cases} 1 & if \quad x = 0 \\ 0 & otherwise \end{cases}$$

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

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

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

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



Convolution Theorem - Example

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

$$g(x) = c$$

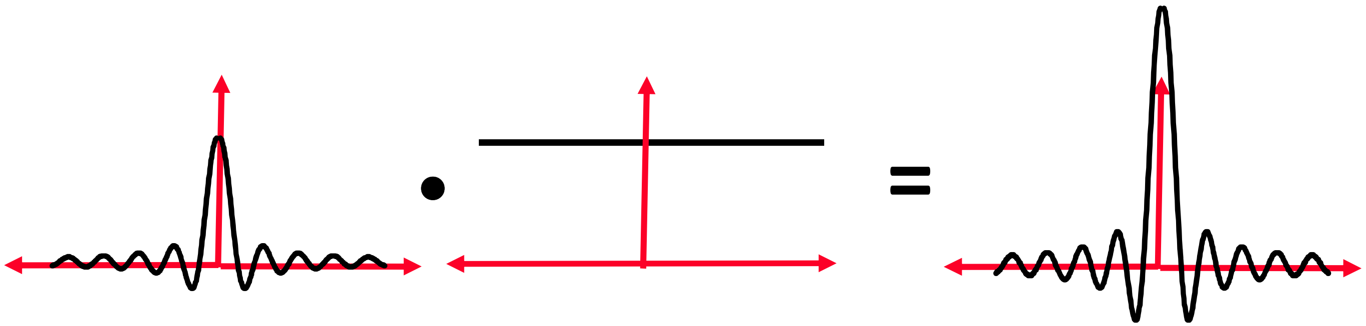
For any function **g(x)** :

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

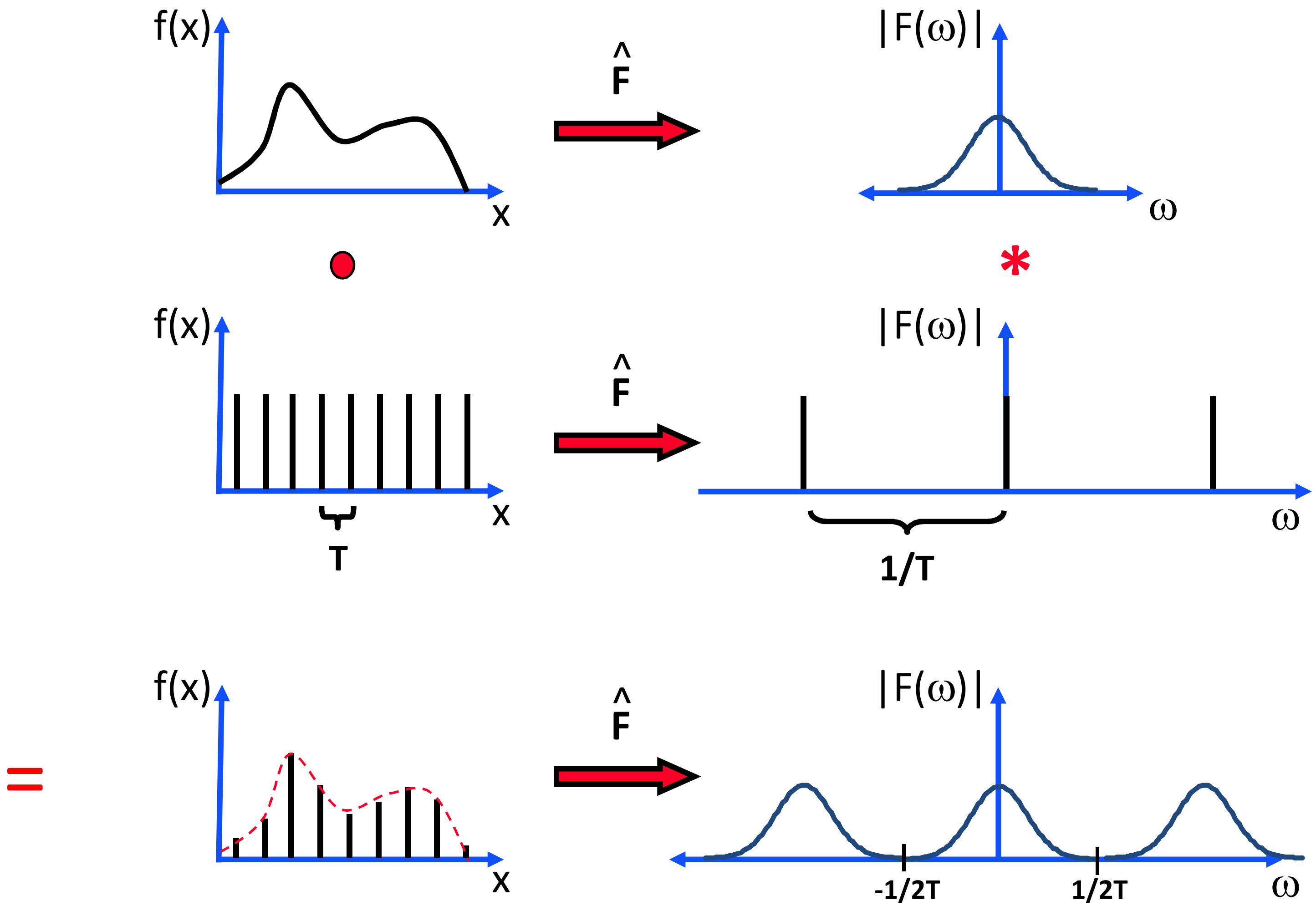
$$\downarrow \qquad \downarrow$$

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

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

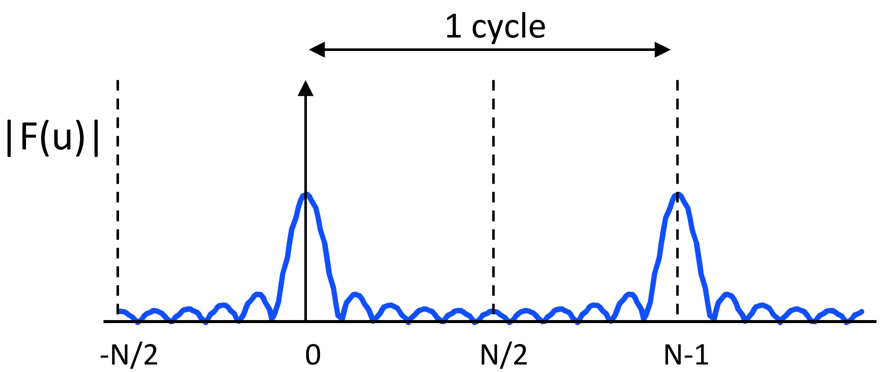


Sampling the Spatial Domain

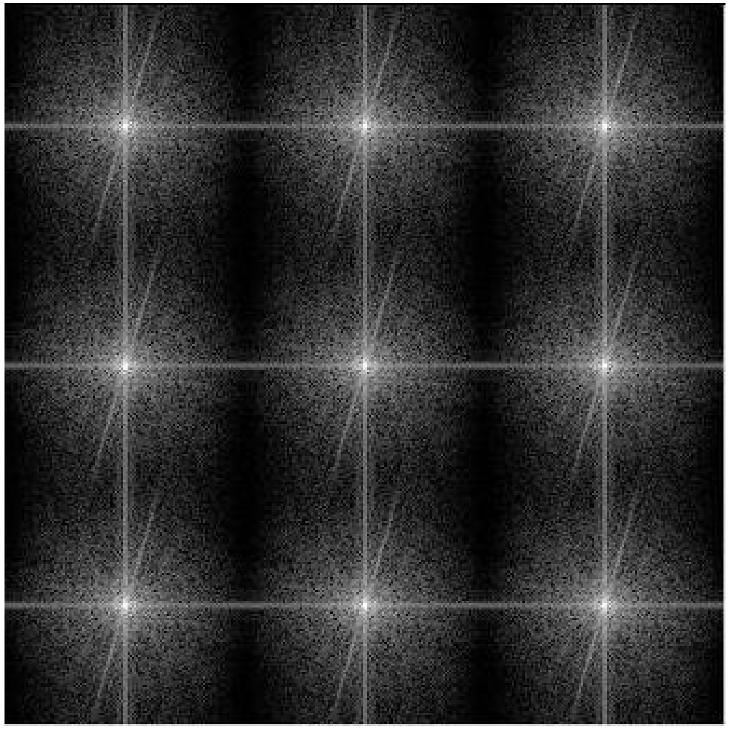


Sampling f(x) at cycle T produces replicas in the frequency domain with cycle 1/T.

Symmetry of FT :



F(k)=F(N+k)

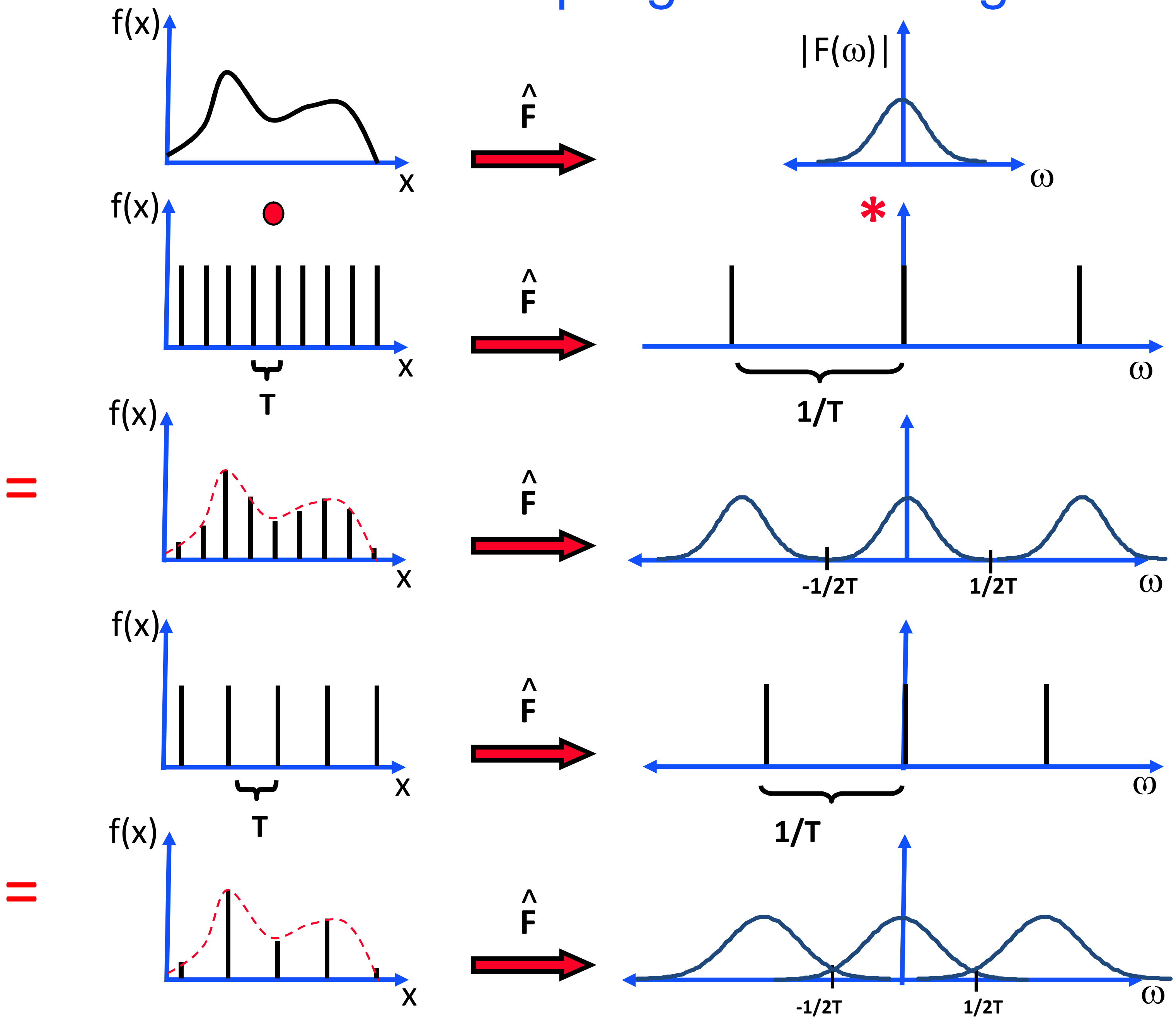


F(u,v) = F(u+N,v)

=F(u,v+M)

=F(u+N,v+M)

Undersampling and Aliasing



Critical Sampling

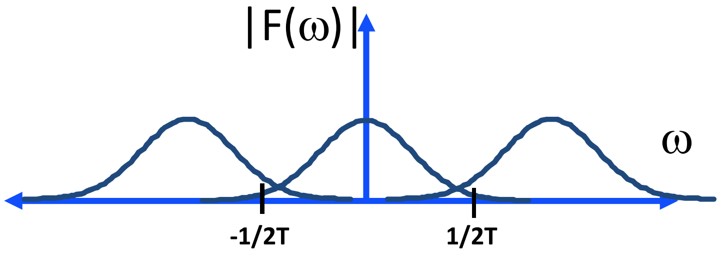
- If the maximal frequency of f(x) is ω_{max} , it is clear from the above replicas that ω_{max} should be smaller that 1/2T

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

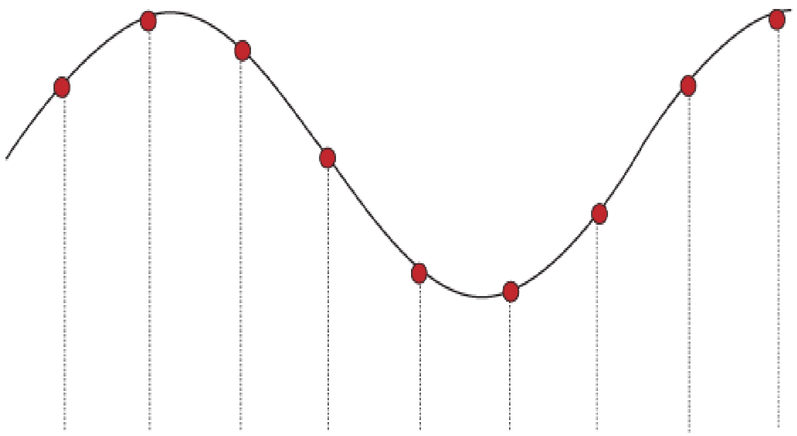
- Nyquist Theorem: If maximal frequency of f(x) is ω_{max} , sampling rate should be larger than $2\omega_{max}$ in order to fully reconstruct f(x) from its samples.

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

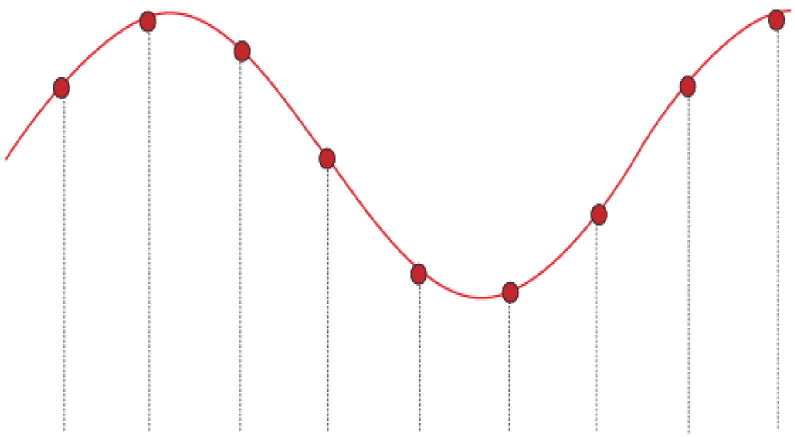
- If the sampling rate is smaller than $2\omega_{max}$ overlapping replicas produce aliasing.



Critical Sampling



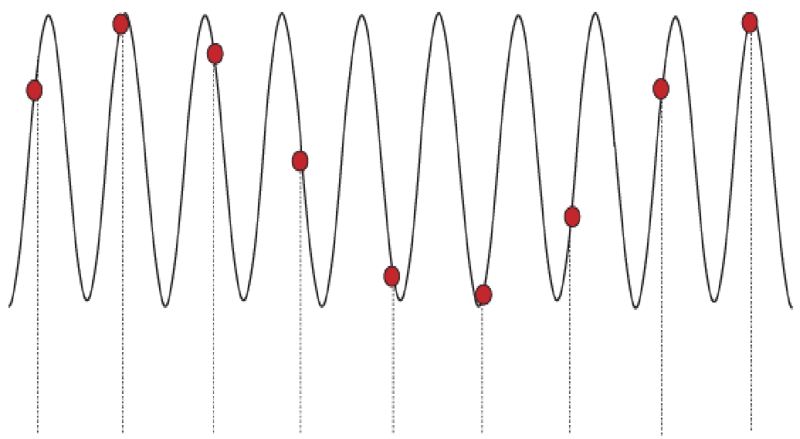
Input



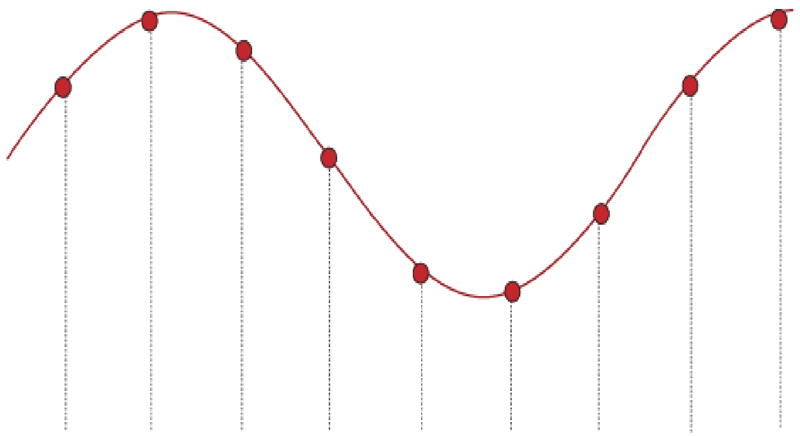
Reconstructed

Demo: B. Freeman

Aliasing



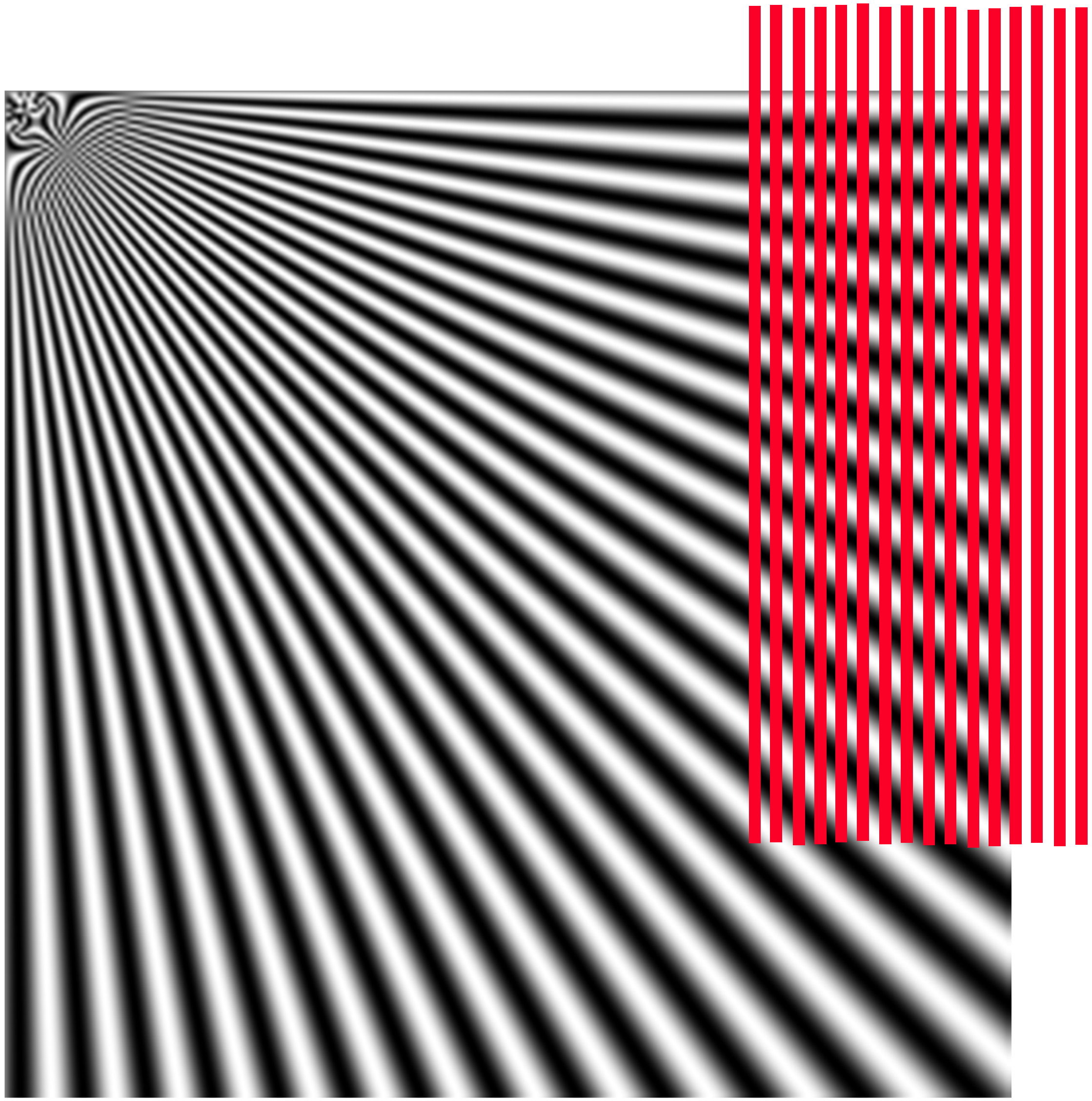
Input



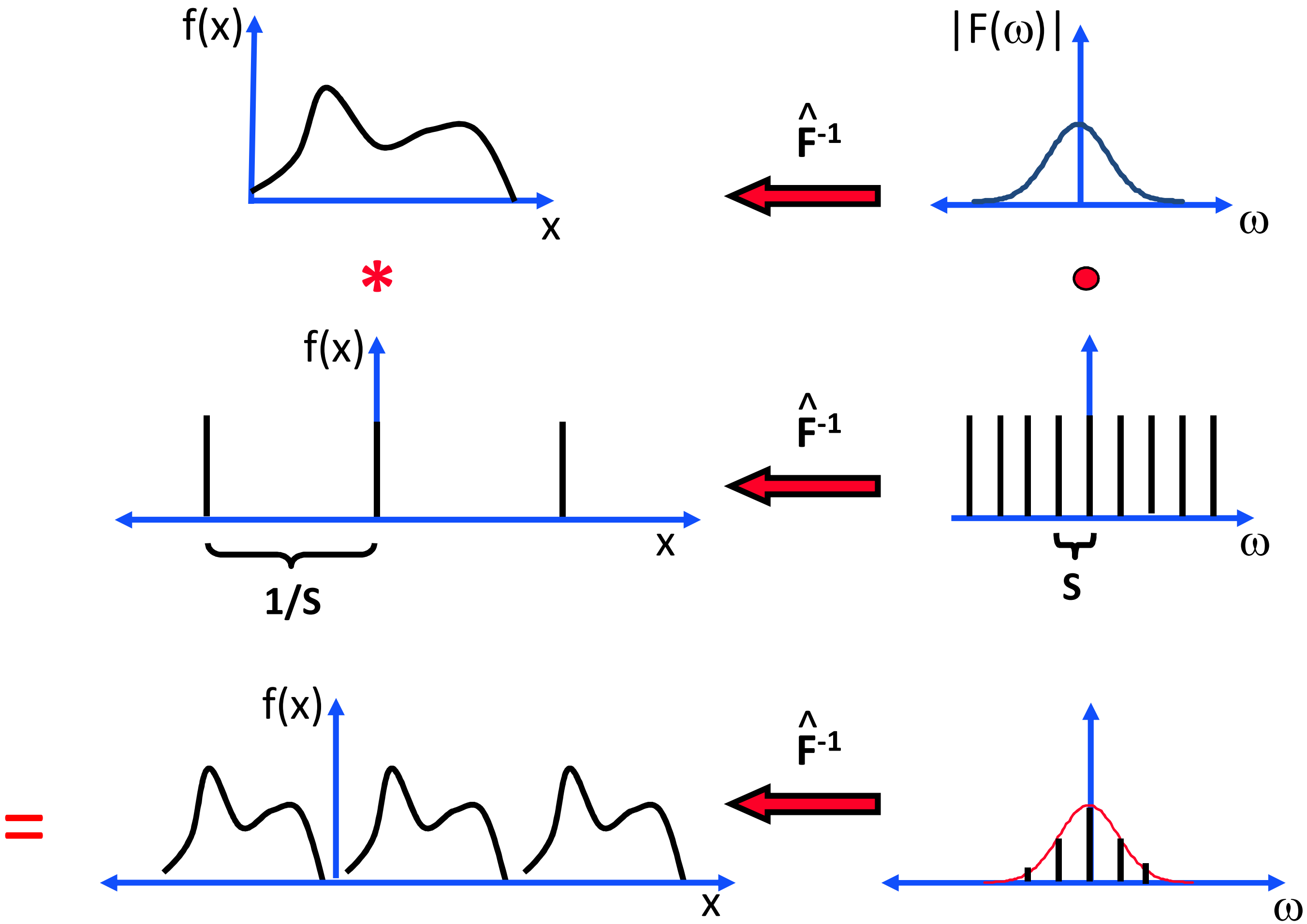
Reconstructed

Demo: B. Freeman

Aliasing



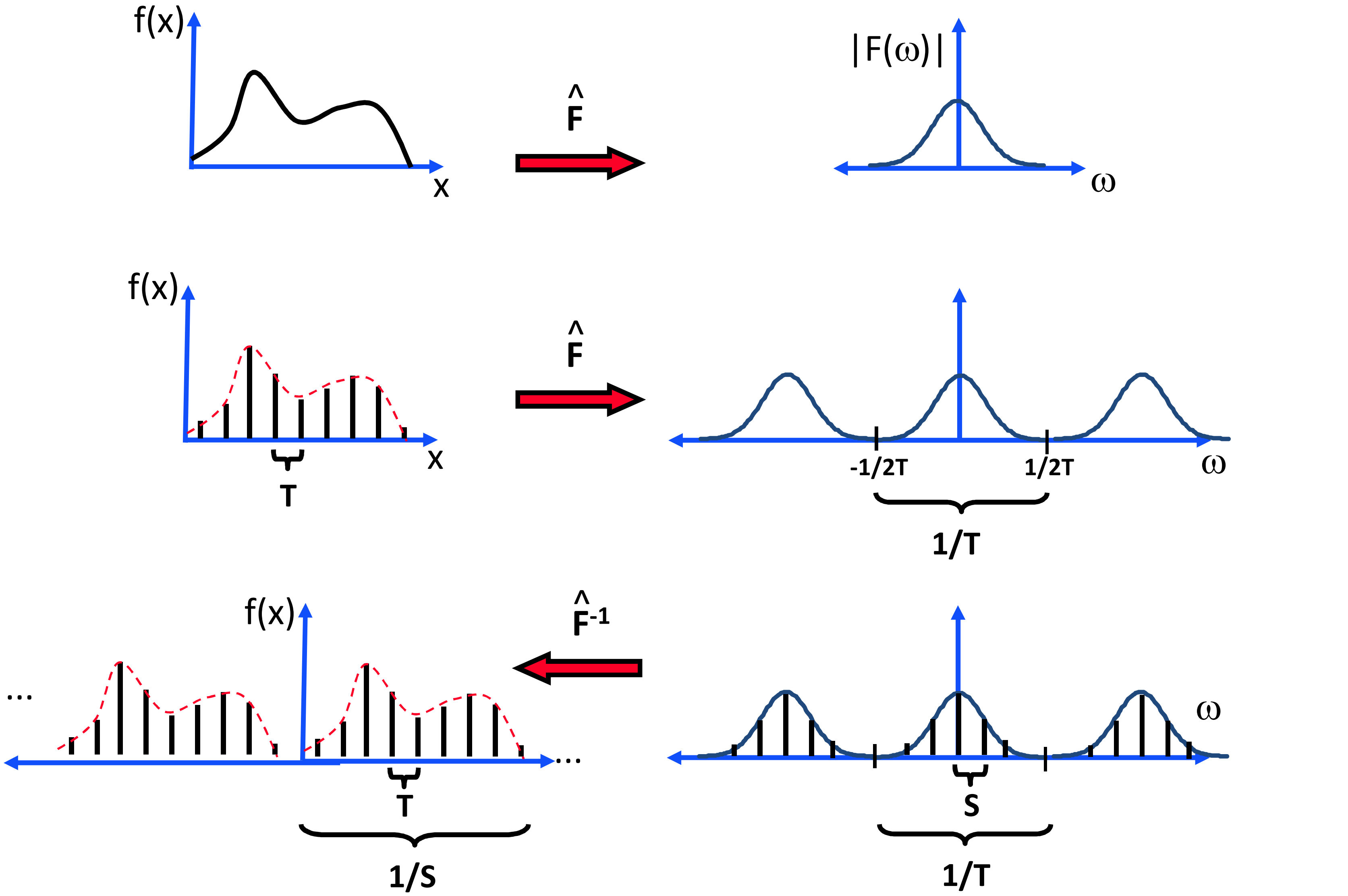
Sampling the Frequency Domain



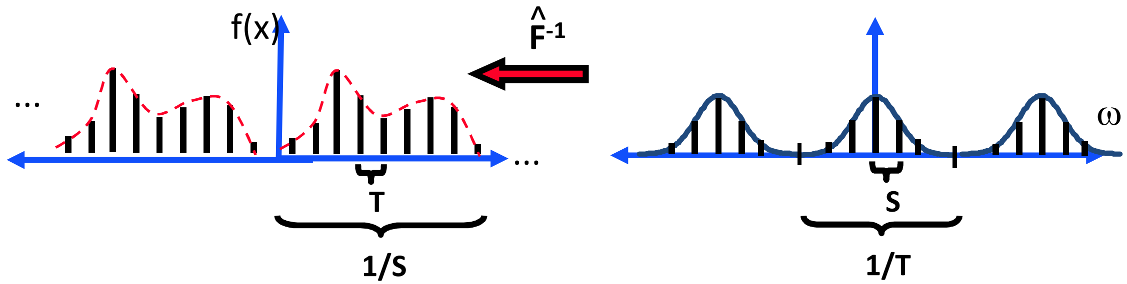
Sampling F(ω) at cycle S produces replicas in the image domain with cycle 1/S.

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).

$$M \geq N$$

Optimal Interpolation

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

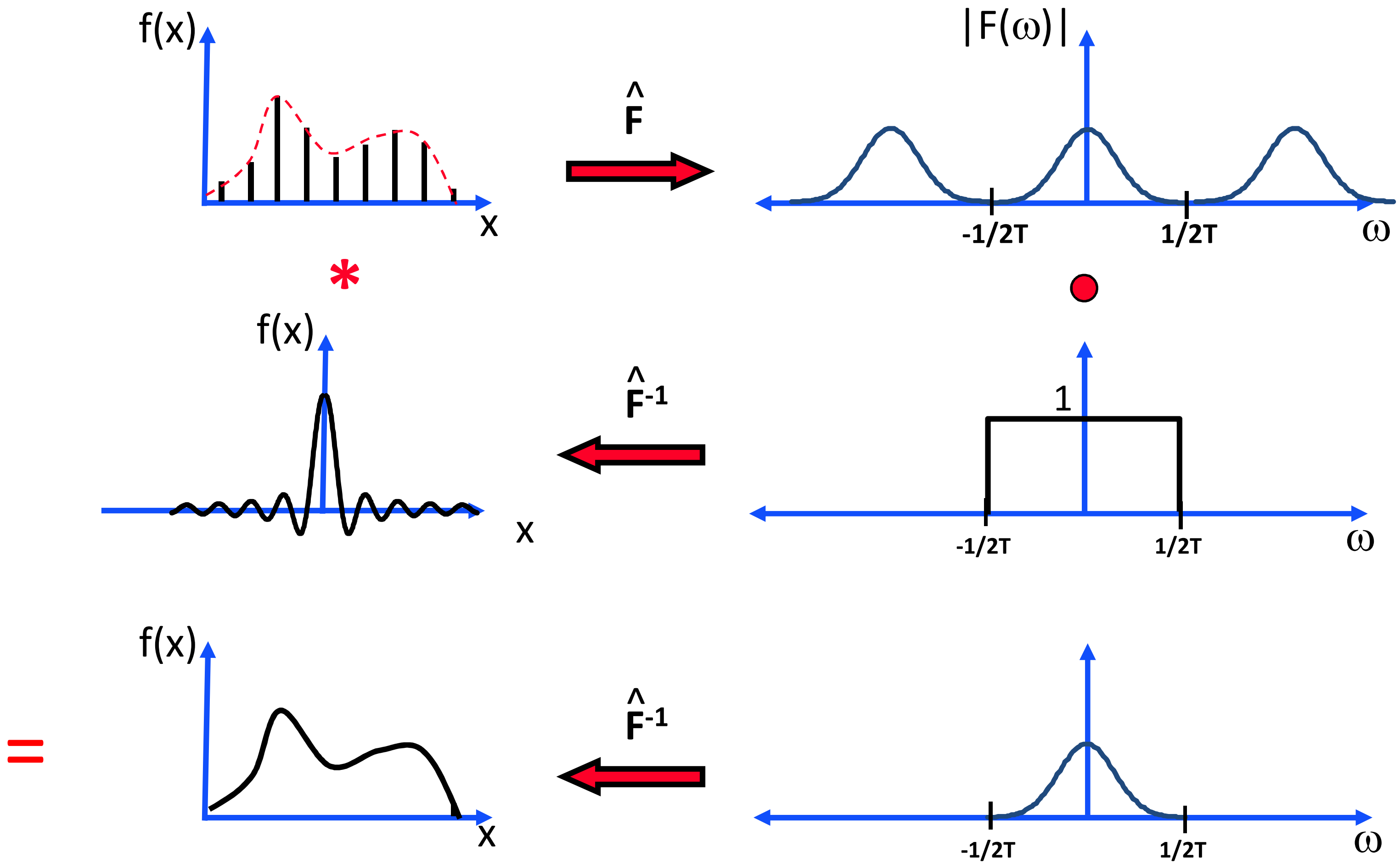


Image Scaling

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

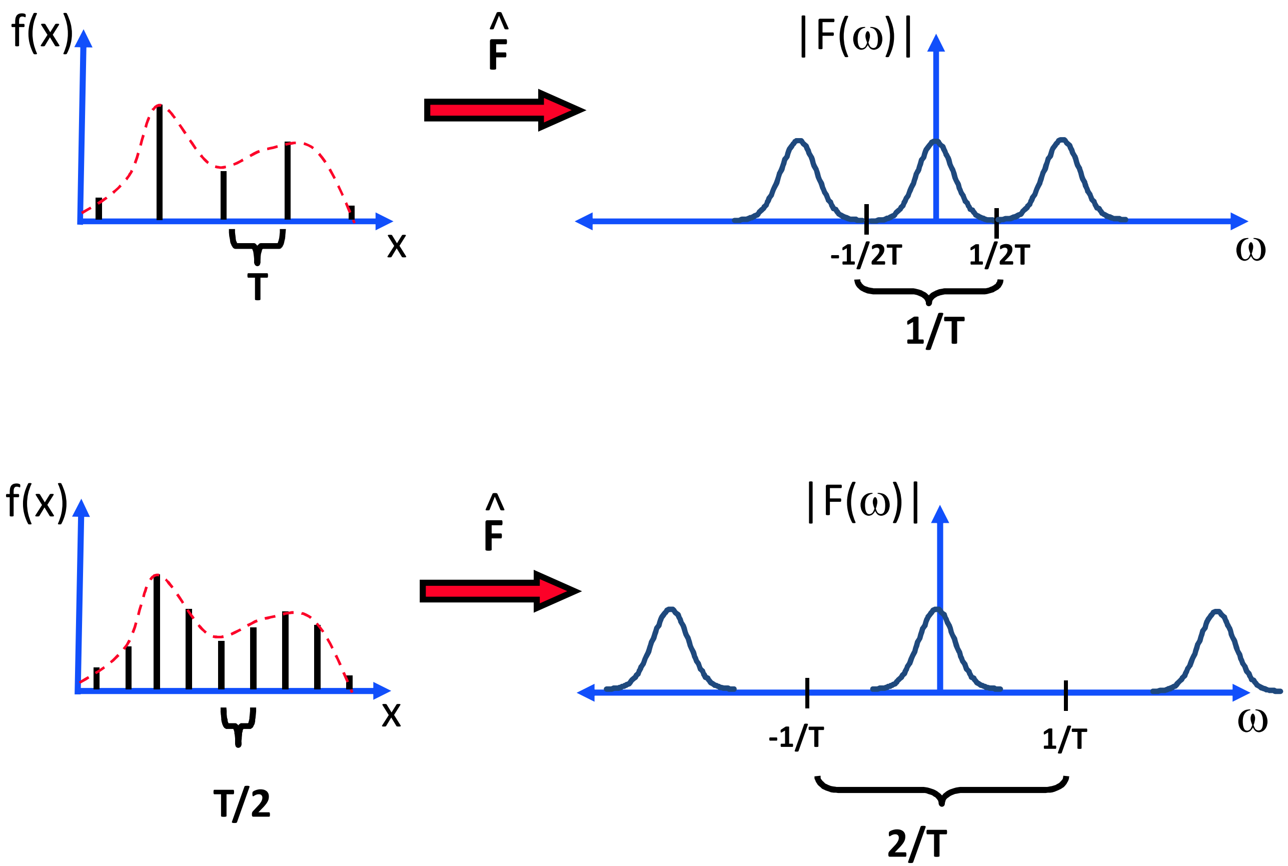


Image Scaling

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

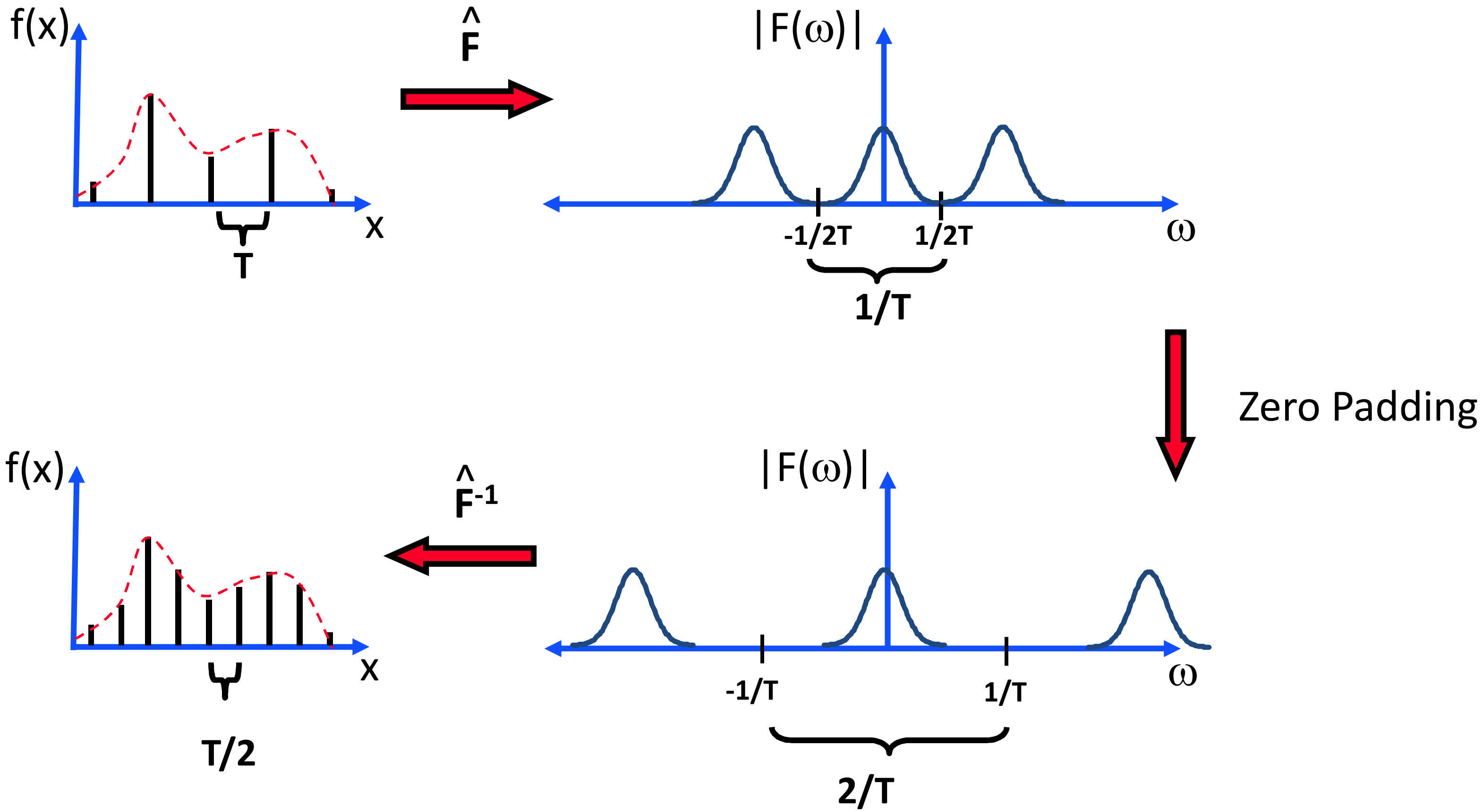
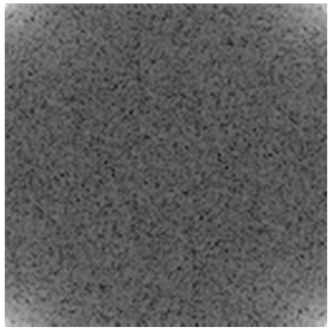


Image Scaling Example



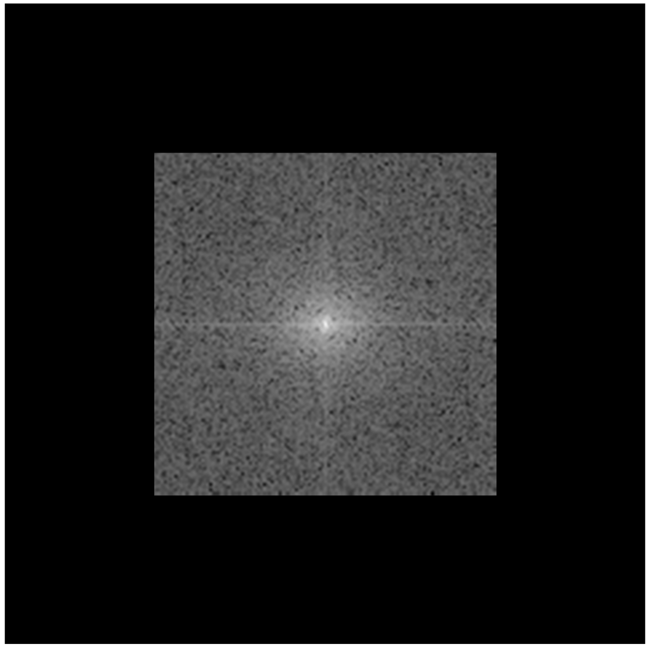


Image Scaling Example

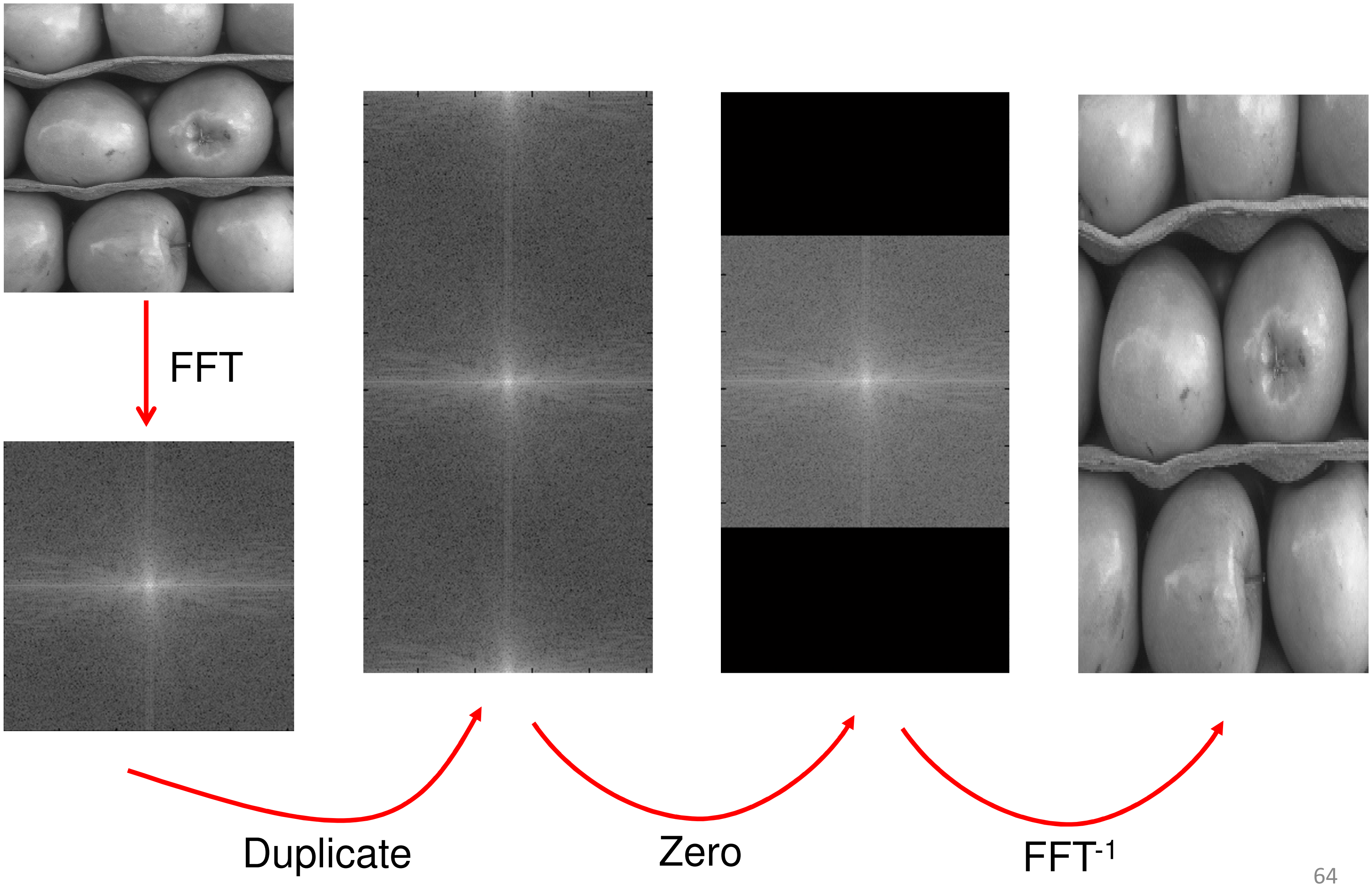
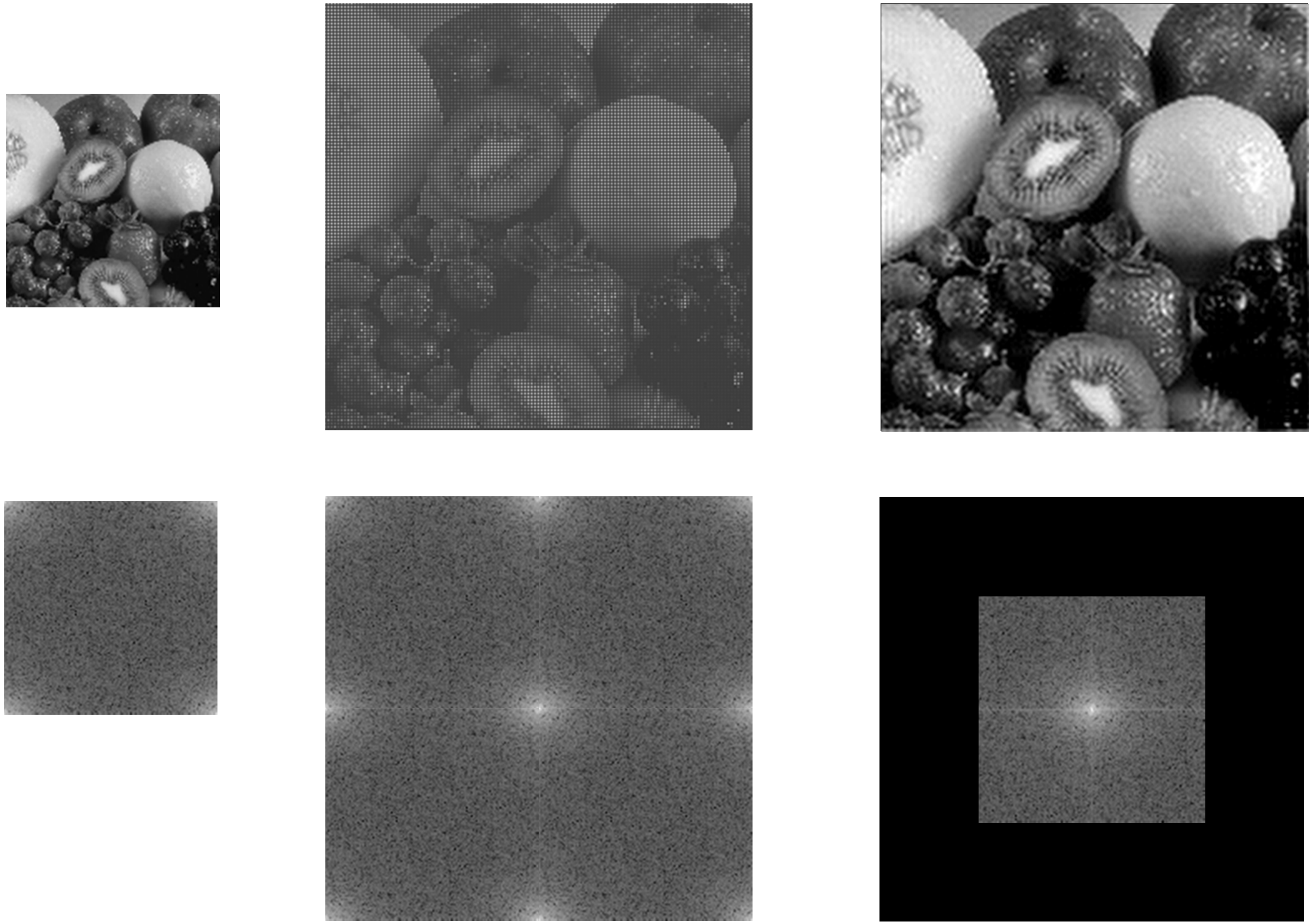
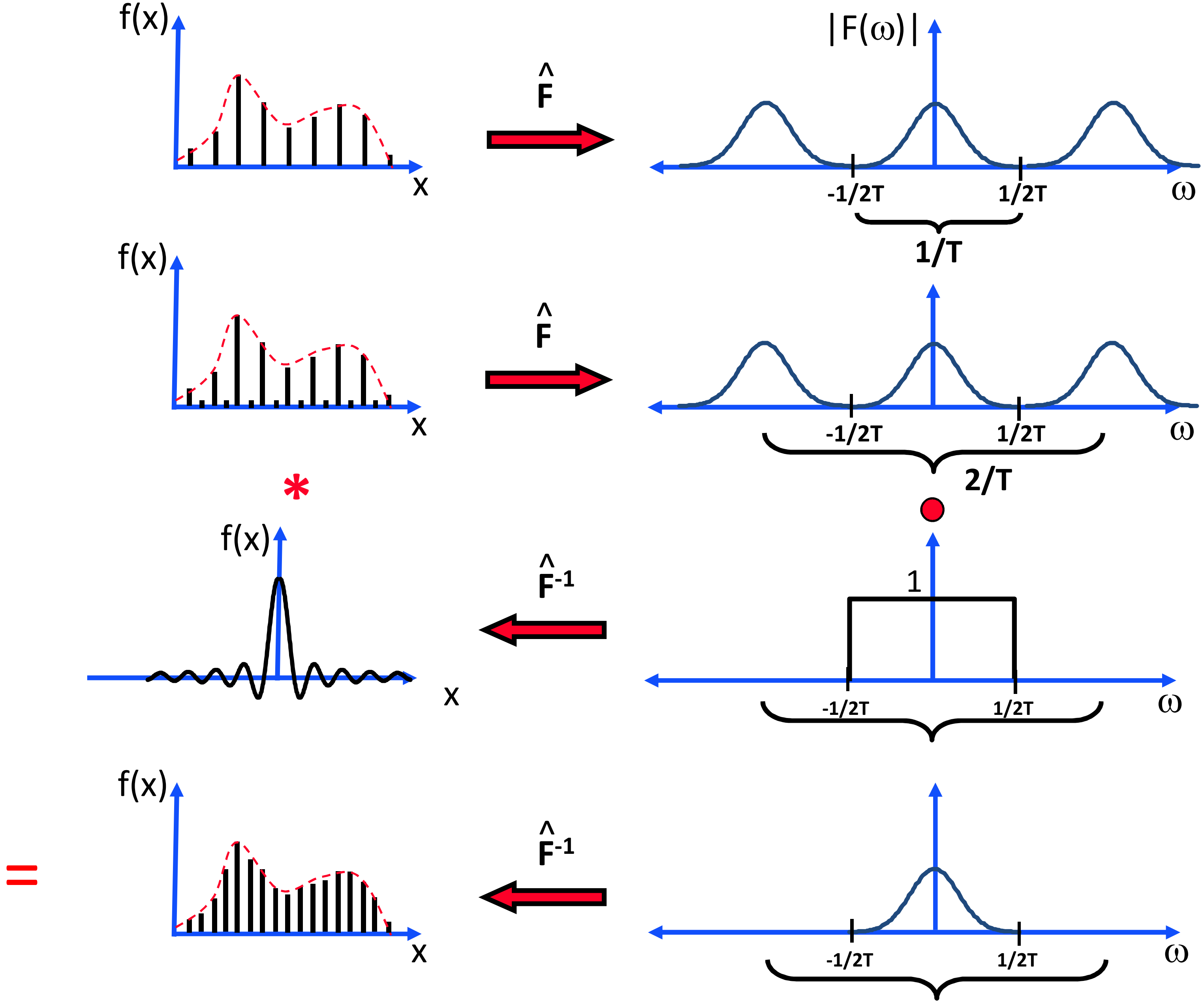


Image Scaling Example



Optimal Interpolation - Digital



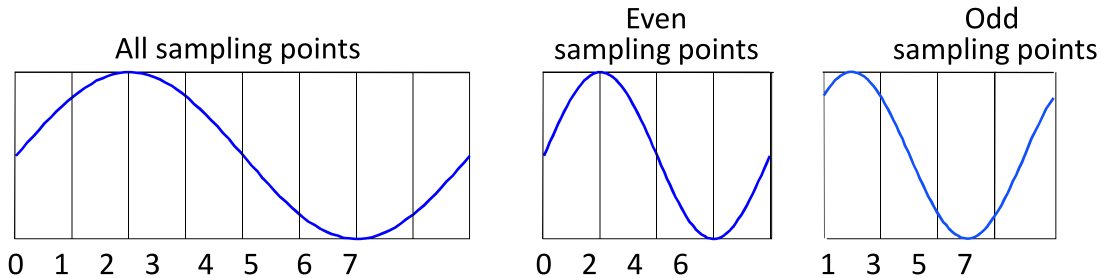
Fast Fourier Transform

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

$O(n^2)$ operations

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

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

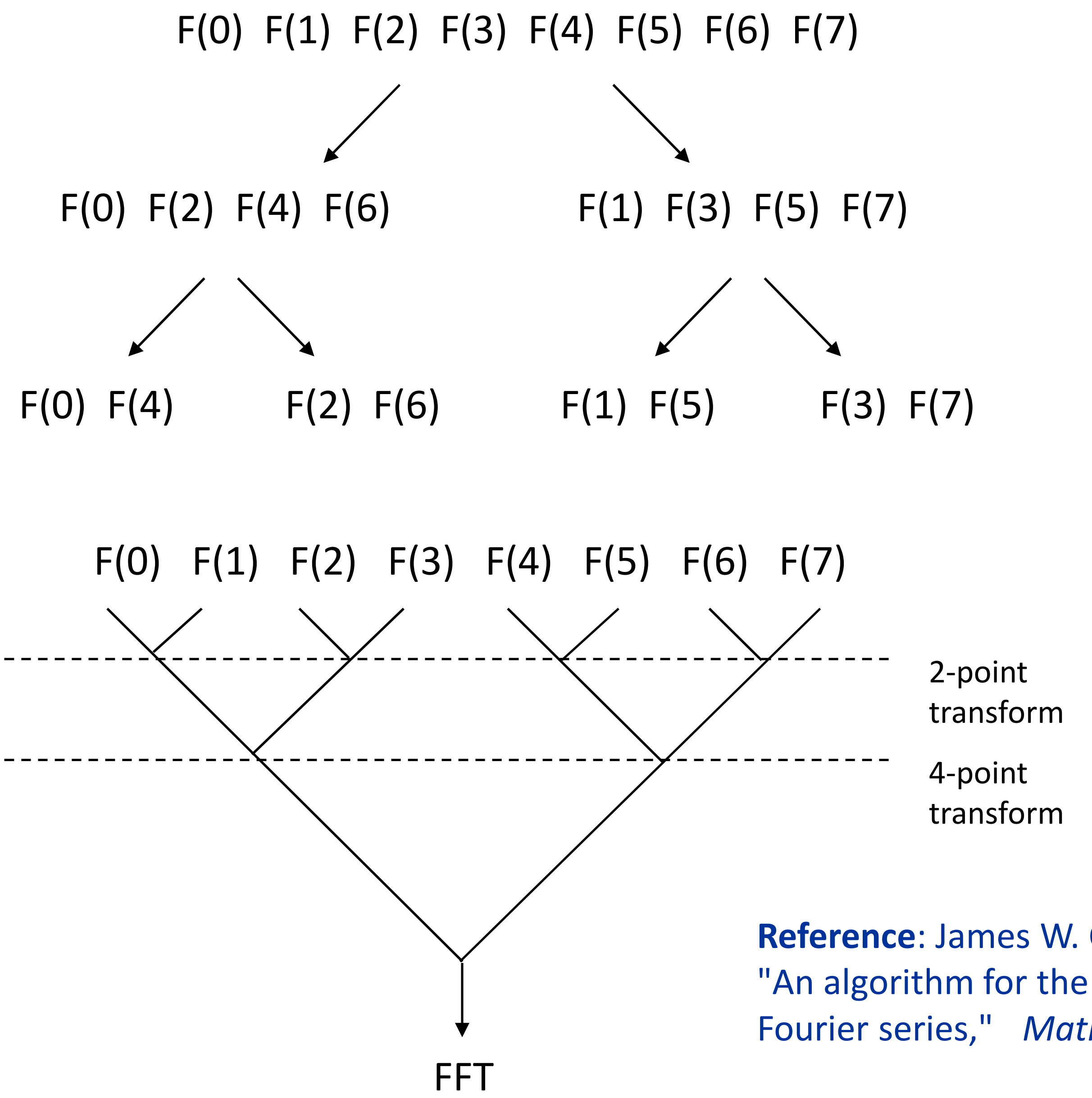


The Fourier transform of N inputs, can be performed as 2 Fourier Transforms of N/2 inputs each + one complex multiplication and addition for each value.
Thus, if F(N) is the computation complexity of FFT:

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).