Digital Image Processing
Fourier Transform

Instructor: Chen Yisong

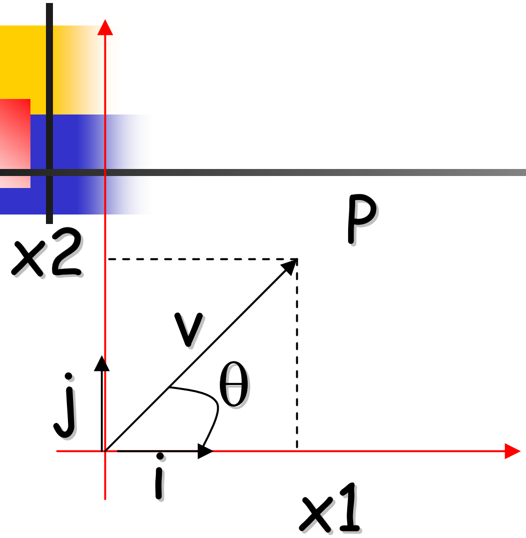
HCI & Multimedia Lab, Peking University

Let's start from quantum physics

- wave-particle duality of light(光的波粒二象性)
  - In physics and chemistry, wave–particle duality is the concept that all matter exhibits both wave-like and particle-like properties. (*Wikipedia*)
    - 一切物质同时具备波的特质及粒子的特质。
  - Digital image…… *is no exception*

  - Particle: a local point of view
  - Wave: a global point of view

Orthonormal Basis (标准正交基)



$$\mathbf{i} = (1,0) \qquad \|\mathbf{i}\| = 1$$

$$\mathbf{j} = (0,1) \qquad \|\mathbf{j}\| = 1$$

$$\mathbf{i} \cdot \mathbf{j} = 0$$

$$\mathbf{v} = (x_1, x_2) \qquad \mathbf{v} = \boxed{x_1.\mathbf{i} + x_2.\mathbf{j}} = \boxed{< \mathbf{v}, \mathbf{i} >}\mathbf{i} + \boxed{< \mathbf{v}, \mathbf{j} >}\mathbf{j}$$

$$\mathbf{v}.\mathbf{i} = (x_1.\mathbf{i} + x_2.\mathbf{j}).\mathbf{i} = x_1.1 + x_2.0 = x_1$$

$$\mathbf{v}.\mathbf{j} = (x_1.\mathbf{i} + x_2.\mathbf{j}).\mathbf{j} = x_1.0 + x_2.1 = x_2$$

All 2D vectors can be seen as the linear combination of i and j

How to compute the coefficients of the linear combination?

Spatial orthonormal basis for digital image

$$\underset{b_1}{\begin{array}{|c|c|c|c|}\hline 1&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline\end{array}}\quad \underset{b_2}{\begin{array}{|c|c|c|c|}\hline 0&1&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline\end{array}}\quad \underset{b_3}{\begin{array}{|c|c|c|c|}\hline 0&0&1&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline\end{array}}\quad \underset{b_4}{\begin{array}{|c|c|c|c|}\hline 0&0&0&1\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline\end{array}}$$

$$\underset{b_5}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 1&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline\end{array}}\quad \underset{b_6}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&1&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline\end{array}}\quad \underset{b_7}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&0&1&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline\end{array}}\quad \underset{b_8}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&0&0&1\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline\end{array}}$$

$$\underset{b_9}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 1&0&0&0\\\hline 0&0&0&0\\\hline\end{array}}\quad \underset{b_{10}}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&1&0&0\\\hline 0&0&0&0\\\hline\end{array}}\quad \underset{b_{11}}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&1&0\\\hline 0&0&0&0\\\hline\end{array}}\quad \underset{b_{12}}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&1\\\hline 0&0&0&0\\\hline\end{array}}$$

$$\underset{b_{13}}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 1&0&0&0\\\hline\end{array}}\quad \underset{b_{14}}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&1&0&0\\\hline\end{array}}\quad \underset{b_{15}}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&1&0\\\hline\end{array}}\quad \underset{b_{16}}{\begin{array}{|c|c|c|c|}\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&0\\\hline 0&0&0&1\\\hline\end{array}}$$

- $|b_i|=1$
- $< b_i , b_j >=0$
- These 16 blocks build the orthonormal bases of the 16-space for a 4*4 image.

14	255	25	33
201	198	66	101
9	188	85	85
161	27	7	73

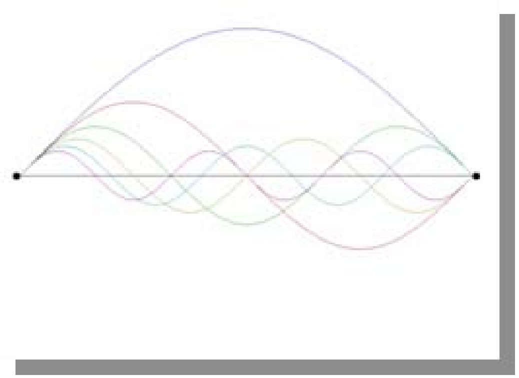
$14b_1 + 255b_2 + 25b_3 + 33b_4 +$
$201b_5 + 198b_6 + 66b_7 + 101b_8 +$
$9b_9 + 188b_{10} + 85b_{11} + 85b_{12} +$
$161b_{13} + 27b_{14} + 7b_{15} + 73b_{16}$

Signal decomposition to bases

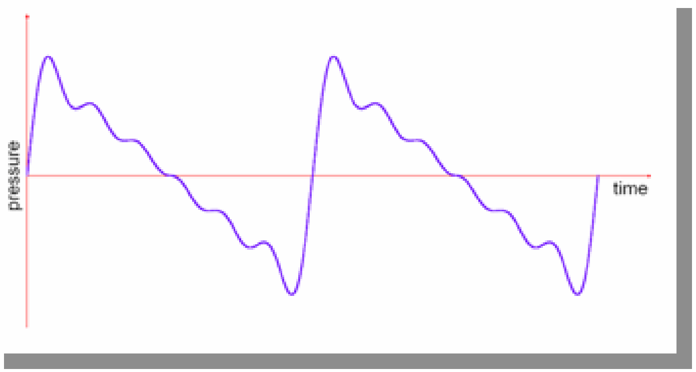
- Vision: decomposition in spatial domain
- Hearing: decomposition in frequency domain

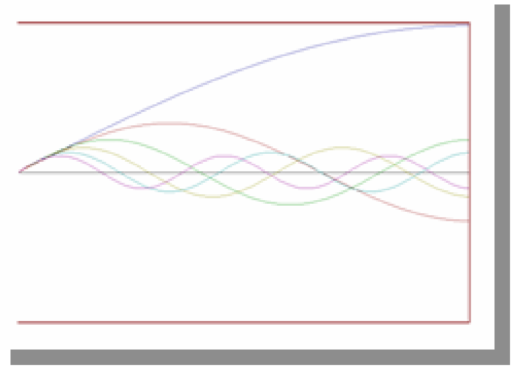
Sound Waves:

Emerge from the superposition of the modes (模态的叠合).

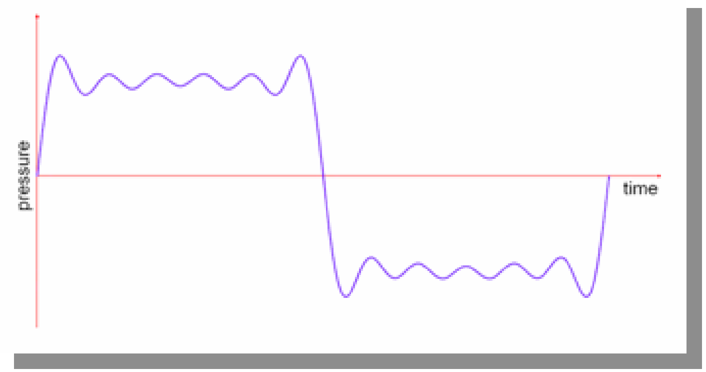


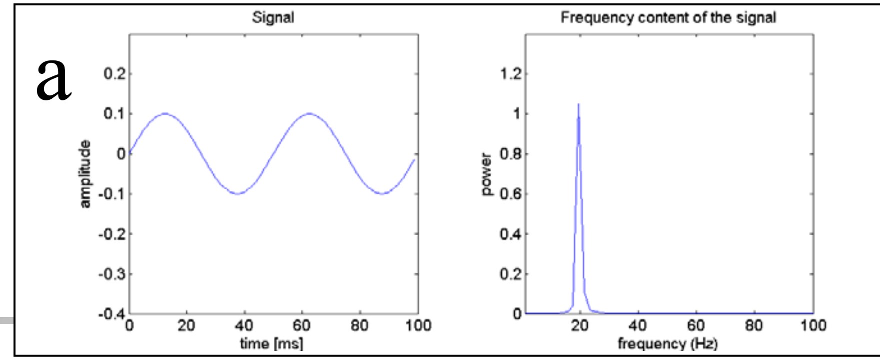
string sound



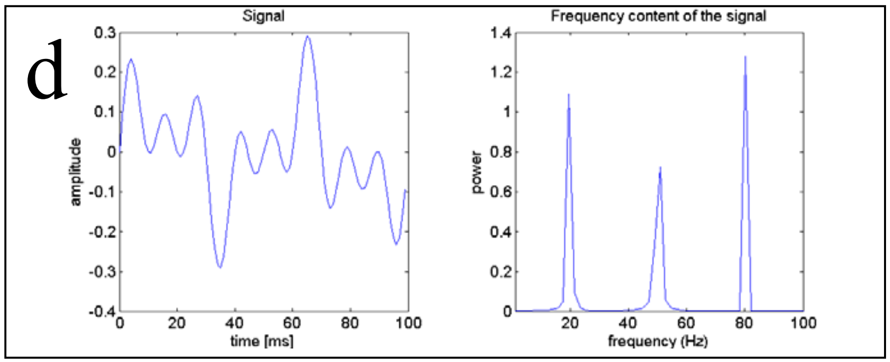


pipe sound

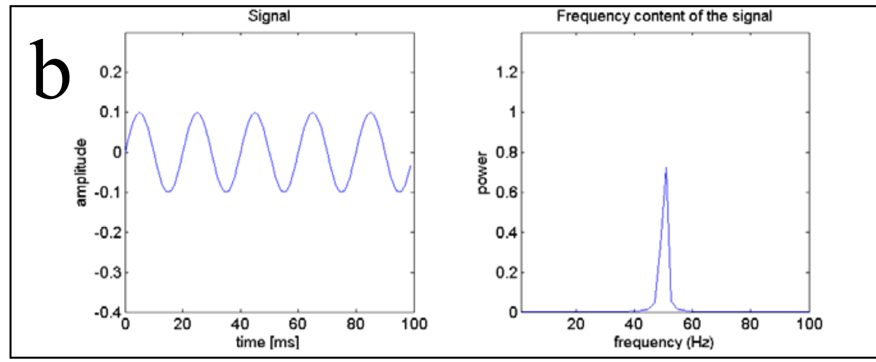




time domain frequency domain

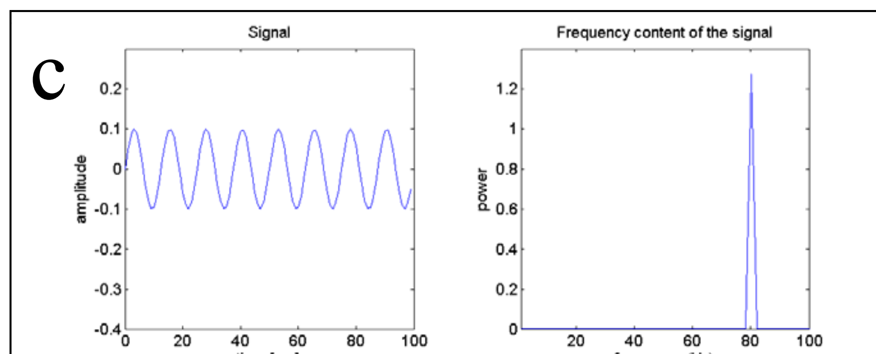


time domain frequency domain



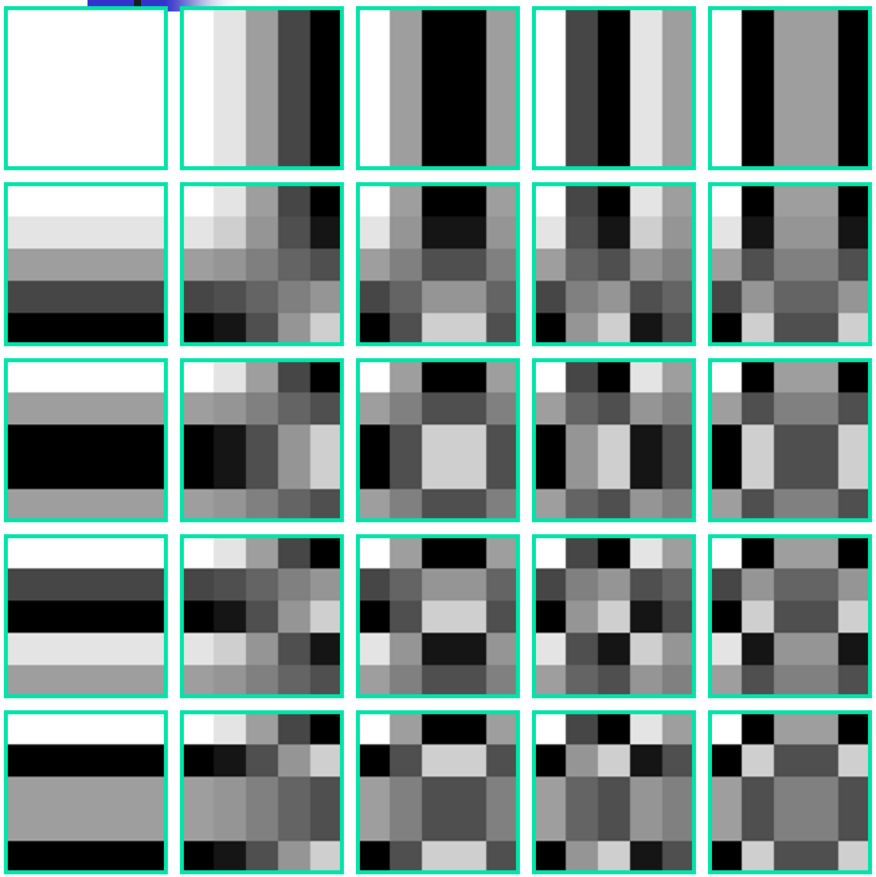
time domain frequency domain

$$d = a + b + c$$



time domain frequency domain

Frequency orthonormal basis for digital image



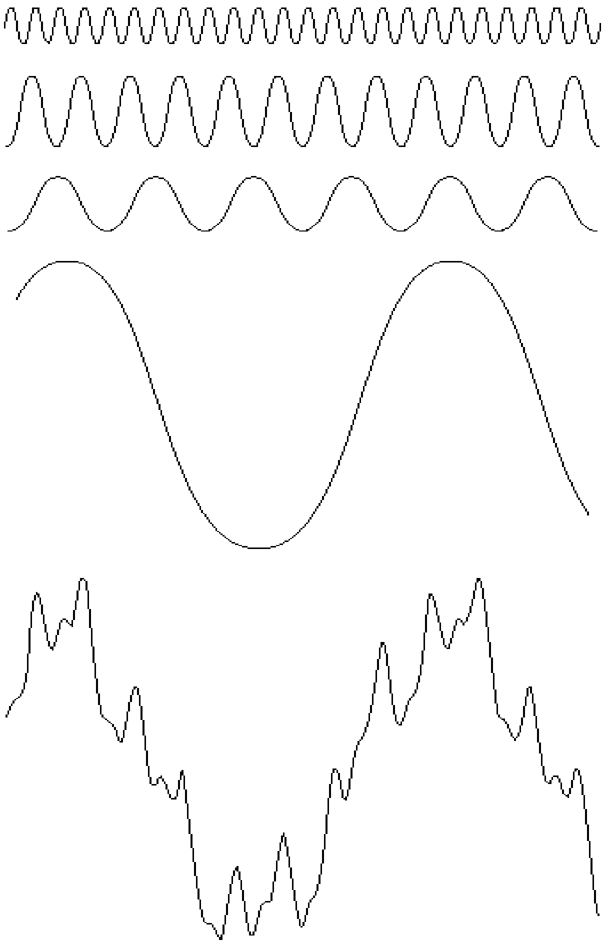
- $|b_i|=1$
- $< b_i , b_j >=0$
- These 25 blocks build the frequency orthonormal bases for a 5*5 image.

14	255	25	33	200
201	198	66	101	76
9	188	85	85	42
161	27	7	73	99
153	40	22	113	10

$(\ )b_1+(\ )b_2+(\ )b_3+(\ )b_4+(\ )b_5+(\ )b_6+(\ )b_7+(\ )b_8+(\ )b_9+(\ )b_{10}+(\ )b_{11}+(\ )b_{12}+(\ )b_{13}+$
$(\ )b_{14}+(\ )b_{15}+(\ )b_{16}+(\ )b_{17}+(\ )b_{18}+(\ )b_{19}+(\ )b_{20}+(\ )b_{21}+(\ )b_{22}+(\ )b_{23}+(\ )b_{24}+(\ )b_{25}$

Fourier Transform(傅立叶变换)

- Basic ideas:
  - A periodic function can be represented by the **sum** of sines functions of different frequencies, multiplied by a different coefficient.
  - Non-periodic functions can also be represented as the **integral** of sines/cosines multiplied by weighing function.



Joseph Fourier (1768-1830)

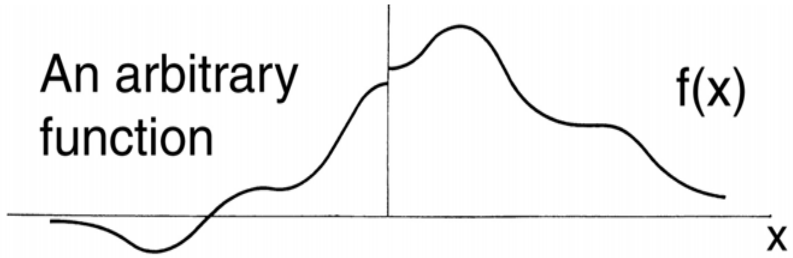
Fourier was obsessed with the physics of heat and developed the Fourier transform theory to model heat-flow problems.

Joseph Fourier, 21 March 1768–16 May 1830. (*By permission of the Bibliothèque Municipale de Grenoble.*)

Fourier transform basis functions

Approximating a square wave as the sum of sine waves.

Any function can be written as the sum of an even and an odd function



Fourier Cosine Series

Because cos(*mt*) is an even function, we can write an even function, *f(t),* as:

$$f(t) = \frac{1}{\pi} \sum_{m=0}^{\infty} F_m \cos(mt)$$

where series F_m is computed as

$$\boxed{F_m = \int_{-\pi}^{\pi} f(t) \cos(mt)\, dt}$$

Here we suppose *f(t)* is over the interval ($-\pi$, π).

Fourier Sine Series

Because sin(*mt*) is an odd function, we can write any odd function, *f(t),* as:

$$f(t) = \frac{1}{\pi}\sum_{m=0}^{\infty} F'_m \sin(mt)$$

where the series F'_m is computed as

$$\boxed{F'_m = \int_{-\pi}^{\pi} f(t)\sin(mt)\,dt}$$

Fourier Series

So if $f(t)$ is a general function, neither even nor odd, it can be written:

$$\boxed{f(t) = \frac{1}{\pi}\sum_{m=0}^{\infty} F_m \cos(mt) + \frac{1}{\pi}\sum_{m=0}^{\infty} F'_m \sin(mt)}$$

Even component

Odd component

$$F_m = \int f(t)\cos(mt)\,dt \qquad F'_m = \int f(t)\sin(mt)\,dt$$

The Inner Product: a Measure of Similarity

The similarity between functions f and g on the interval $(-\lambda/2, \lambda/2)$ can be defined by

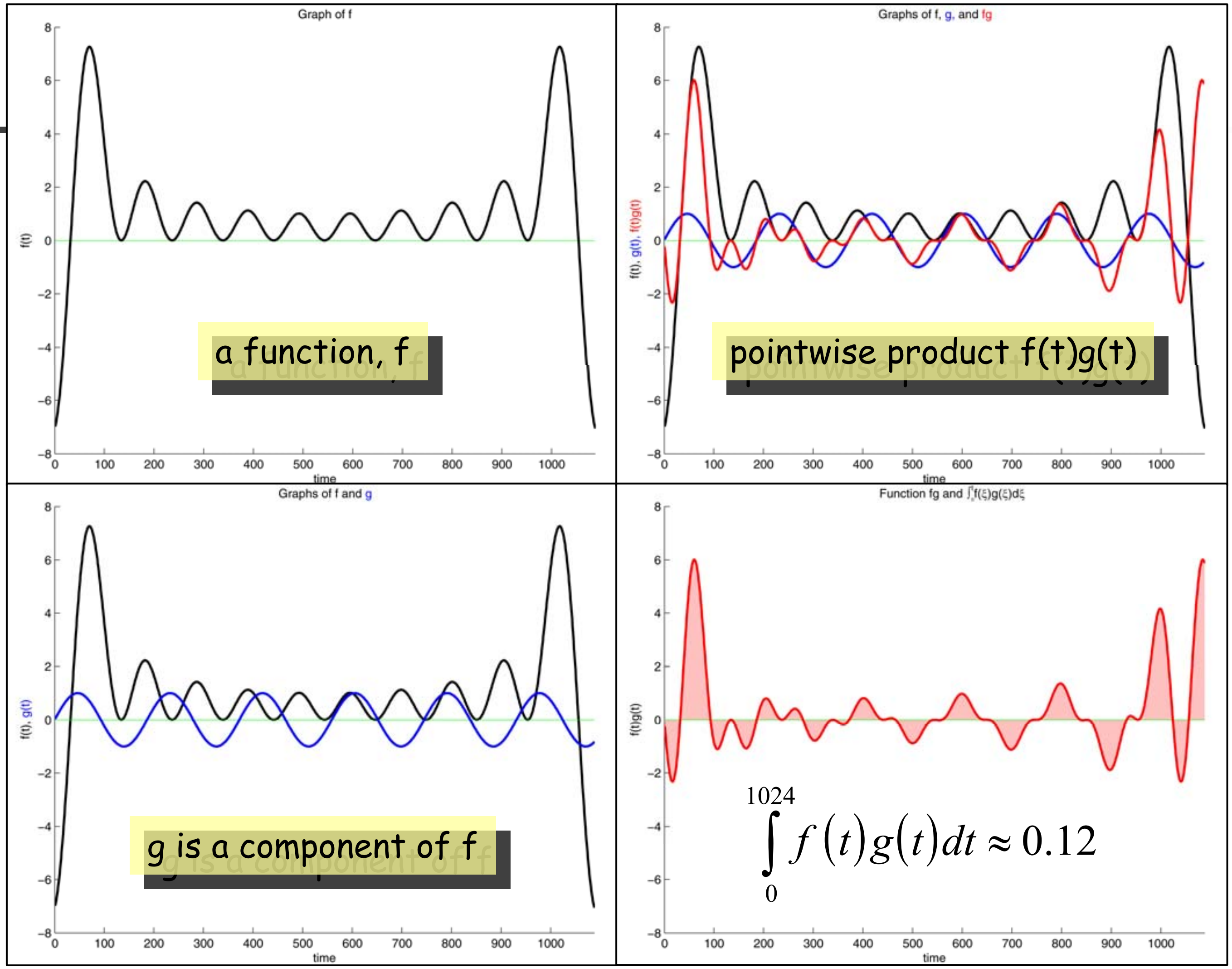
$$\langle f, g \rangle = \int_{-\lambda/2}^{\lambda/2} f(t)\, g^{*}(t)\, dt$$

where $g^{*}(t)$ is the complex conjugate of $g(t)$.

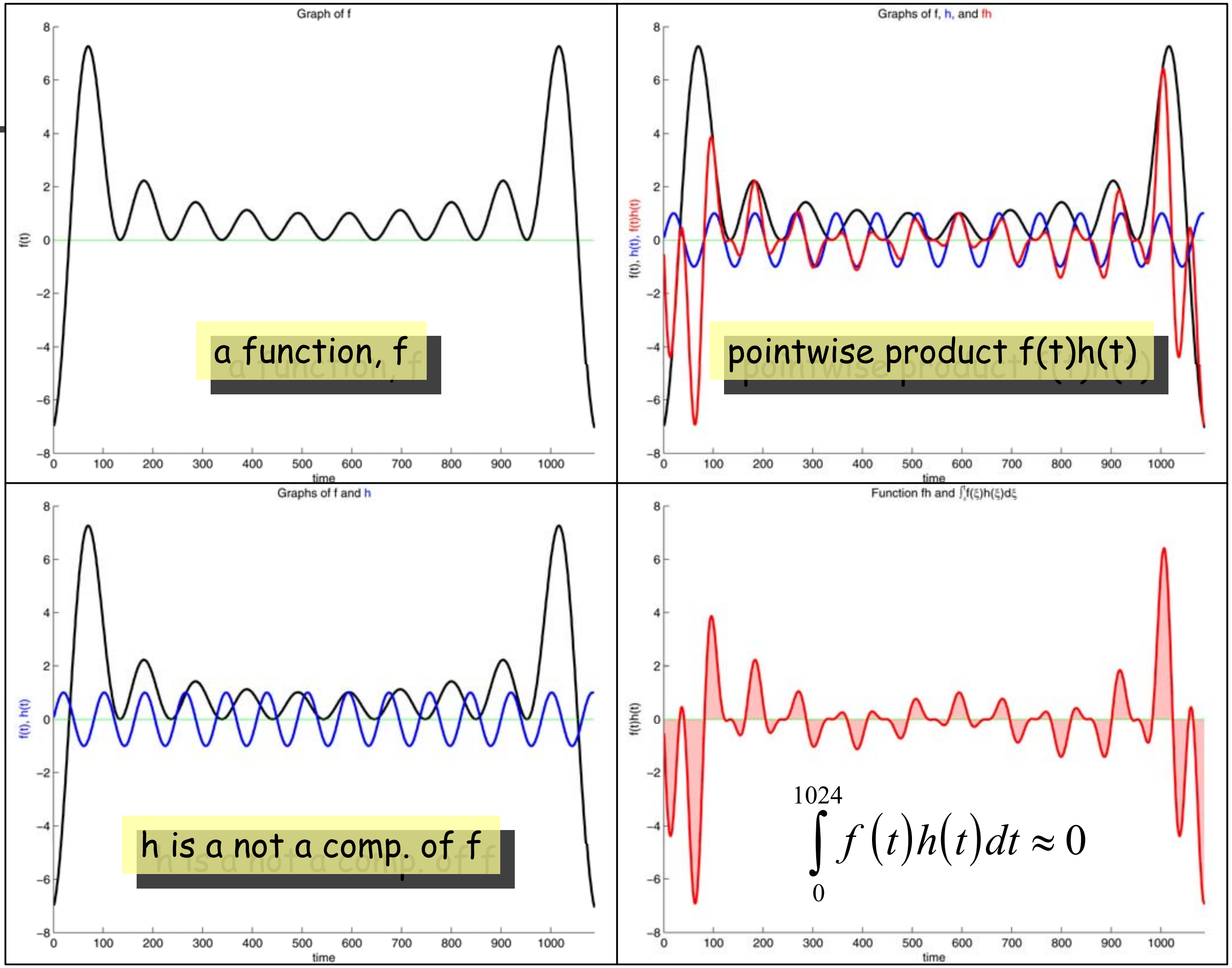
This number, called the *inner product* of f and g, can also be thought of as the amount of g in f or as the projection of f onto g.

If f and g have the same energy, then their inner product is maximal if $f = g$. On the other hand if $\langle f, g \rangle = 0$, then f and g have nothing in common.

Inner Products



Inner Products



Inner Product of a Periodic Function and a Sinusoid

$$\langle f, g\rangle = \int_{-\lambda/2}^{\lambda/2} f(t)\sin\left(\tfrac{2\pi}{\lambda}t\right)dt$$

$$\langle f, g\rangle = \int_{-\lambda/2}^{\lambda/2} f(t)\cos\left(\tfrac{2\pi}{\lambda}t\right)dt$$

$$\begin{aligned}\langle f, g\rangle &= \int_{-\lambda/2}^{\lambda/2} f(t)\left[\cos\left(\tfrac{2\pi}{\lambda}t\right) - i\sin\left(\tfrac{2\pi}{\lambda}t\right)\right]dt \\ &= \int_{-\lambda/2}^{\lambda/2} f(t)e^{-i\frac{2\pi}{\lambda}t}\,dt \\ &= \int_{-\lambda/2}^{\lambda/2} f(t)e^{-i\omega t}\,dt\end{aligned}$$

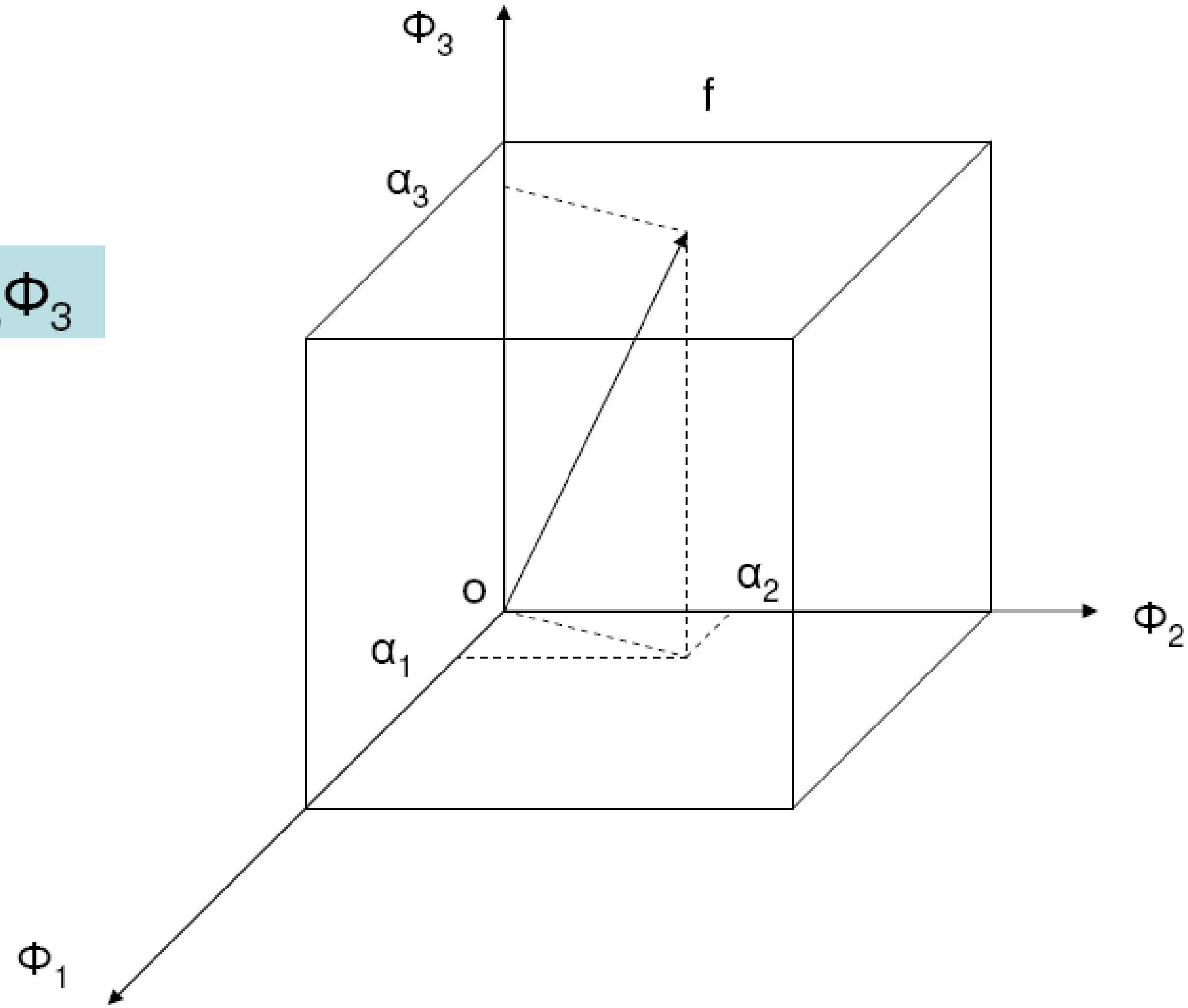
3 different representations

$$e^{-i\frac{2\pi}{\lambda}t} = \cos\left(\tfrac{2\pi}{\lambda}t\right) - i\sin\left(\tfrac{2\pi}{\lambda}t\right)$$

$$\omega = \frac{2\pi}{\lambda}$$

Illustration of Decomposition

$$f = \alpha_1\Phi_1+\alpha_2\Phi_2+\alpha_3\Phi_3$$



Decomposition

- Ortho-normal basis function

$$\int_{-\infty}^{\infty} \phi(x,u_1)\phi^*(x,u_2)dx = \begin{cases} 1, & u_1 = u_2 \\ 0, & u_1 \neq u_2 \end{cases}$$

- Forward Signal decomposition(Coefficient computation)

$$F(u) = < f(x), \phi(x,u) > = \int_{-\infty}^{\infty} f(x)\phi^*(x,u)dx$$

- Inverse Signal reconstruction

$$f(x) = \int_{-\infty}^{\infty} F(u)\phi(x,u)du$$

Fourier Transform

- Basis function

$$\phi(x,u) = e^{j2\pi ux}, \quad u \in (-\infty, +\infty).$$

- Forward Transform Signal decomposition(Coefficient computation)

$$F(u) = F\{f(x)\} = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

- Inverse Transform Signal reconstruction

$$f(x) = F^{-1}\{F(u)\} = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

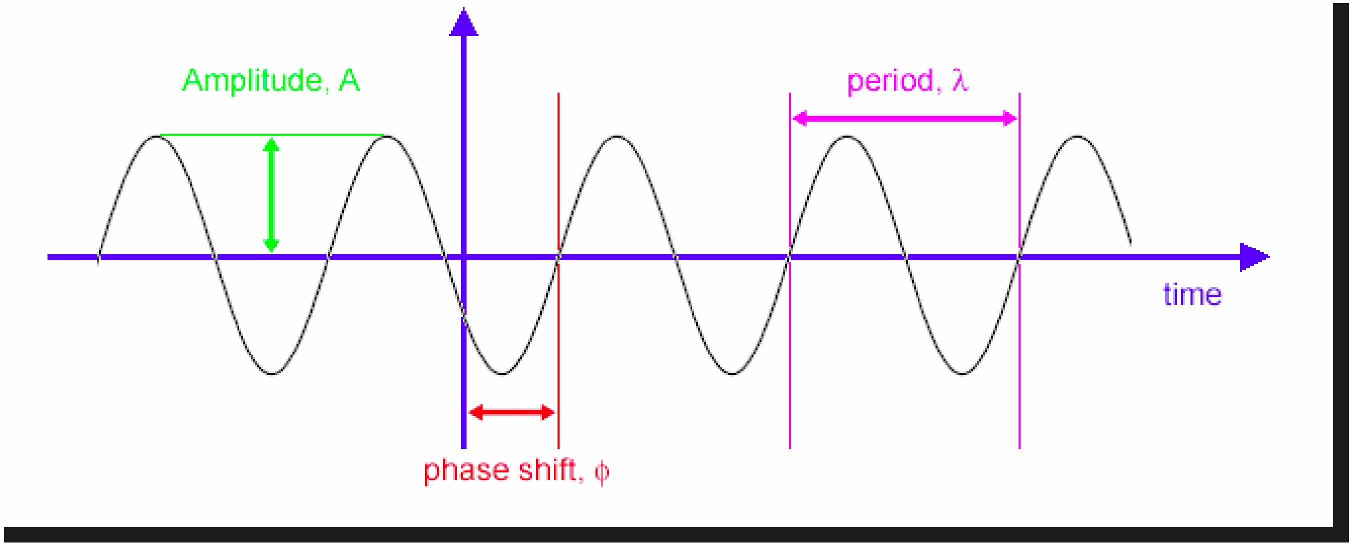
What if f(x) is real?

- Real world signals f(x) are usually real
- F(u) is still complex, but has special properties

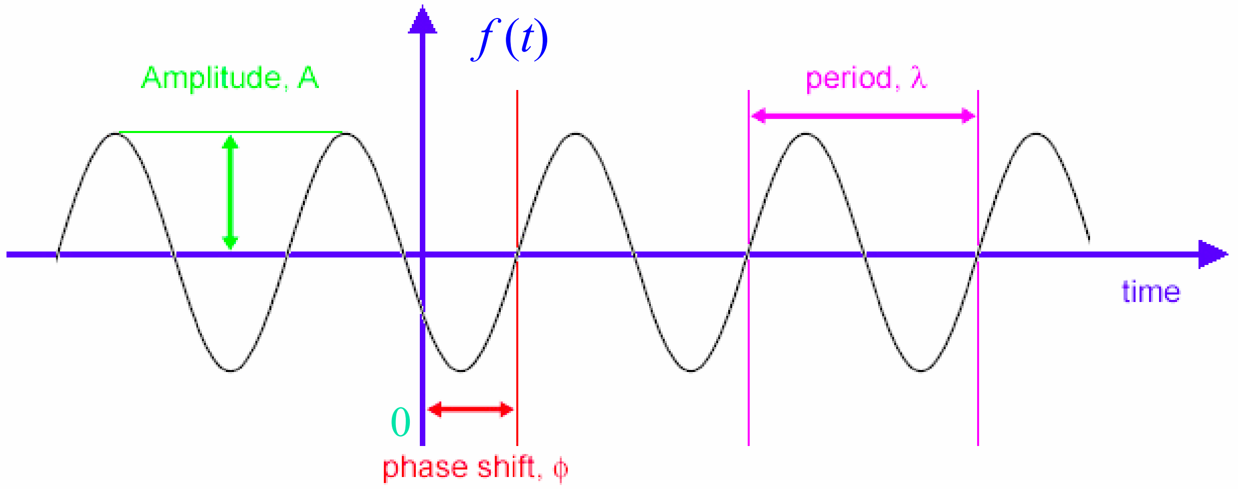
$$F^*(u) = F(-u)$$

$$R(u) = R(-u), A(u) = A(-u), P(u) = P(-u) : \text{even function}$$

$$I(u) = -I(-u), \phi(u) = -\phi(-u) : \text{odd function}$$



Anatomy of a Sinusoid



$$f(t) = A\sin\left(\frac{2\pi}{\lambda}t - \phi\right)$$

$1/\lambda$ is the frequency of the sinusoid (Hz).

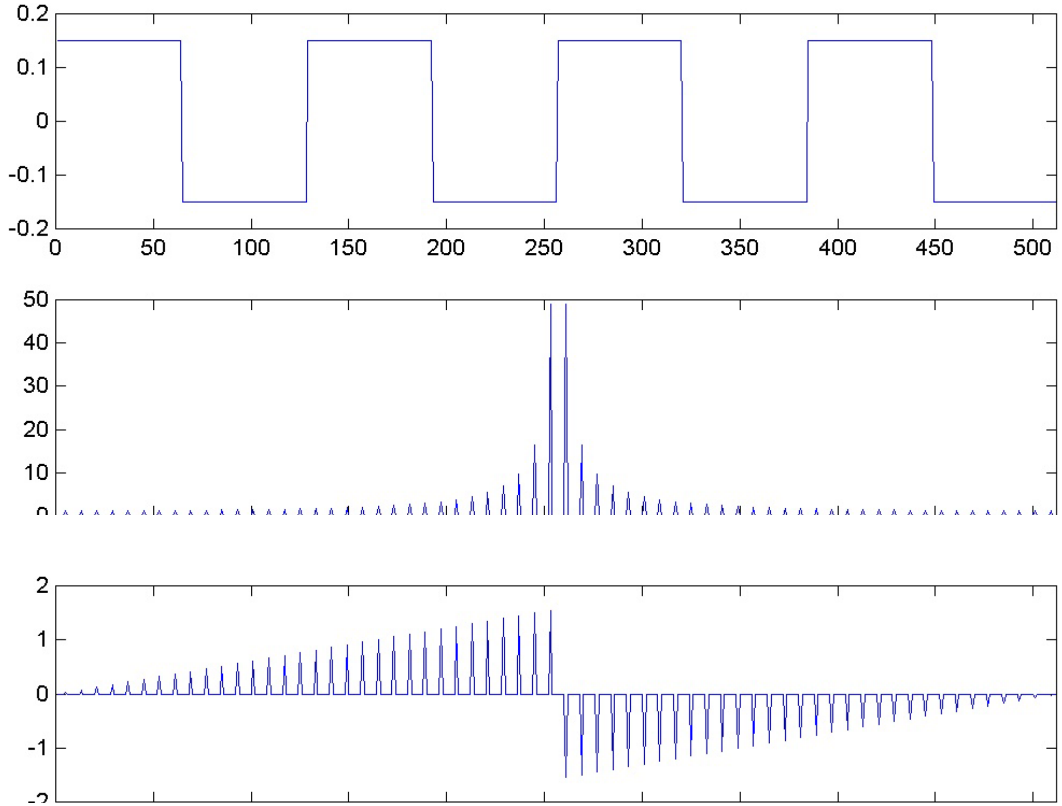
$2\pi/\lambda$ is the angular frequency (radians/s).

Fourier Transform of a Square Wave

Time-domain signal

Fourier magnitude

Fourier phase



Property of Fourier Transform

- Duality

$$f(t) \Leftrightarrow F(u)$$
$$F(t) \Leftrightarrow f(-u)$$

- Linearity

$$F\{a_1 f_1(x) + a_2 f_2(x)\} = a_1 F\{f_1(x)\} + a_2 F\{f_2(x)\}$$

- Scaling

$$F\{af(x)\} = aF\{f(x)\}$$

- Translation

$$f(x - x_0) \Leftrightarrow F(u)e^{-j2\pi x_0 u}, \quad f(x)e^{j2\pi u_0 x} \Leftrightarrow F(u - u_0)$$

- Convolution

$$f(x) \otimes g(x) = \int f(x - \alpha) g(\alpha) d\alpha$$
$$f(x) \otimes g(x) \Leftrightarrow F(u)G(u)$$

We will review convolution later!

Two Dimension Fourier Transform

- Basis functions

$$\phi(x, y; u, v) = e^{j(2\pi ux + 2\pi vy)} = e^{j2\pi ux} e^{j2\pi vy}, \quad u, v \in (-\infty, +\infty).$$

- Forward – Transform

$$F(u,v) = F\{f(x,y)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-j2\pi(ux+vy)} dxdy$$

- Inverse – Transform

$$f(x,y) = F^{-1}\{F(u,v)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) e^{j2\pi(ux+vy)} dudv$$

- Property
  - All the properties of 1D FT apply to 2D FT

Separability of 2D FT and Separable Signal

- Separability of 2D FT

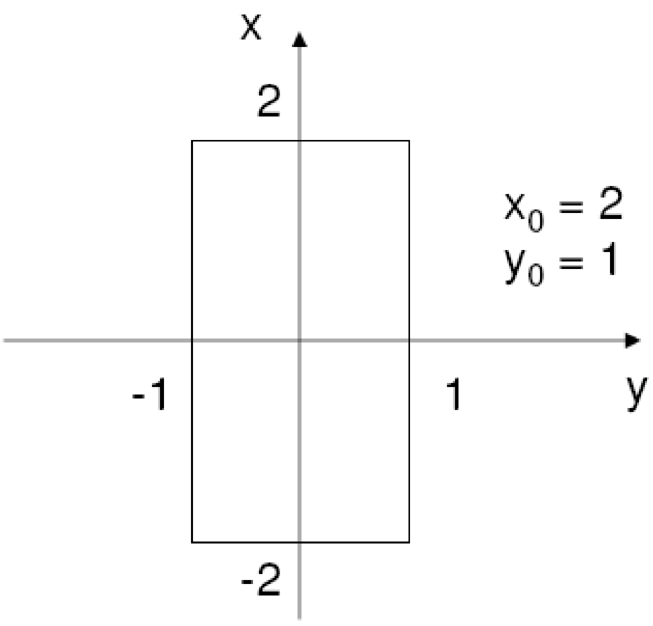
$$F_2\{f(x,y)\} = F_y\{F_x\{f(x,y)\}\} = F_x\{F_y\{f(x,y)\}\}$$

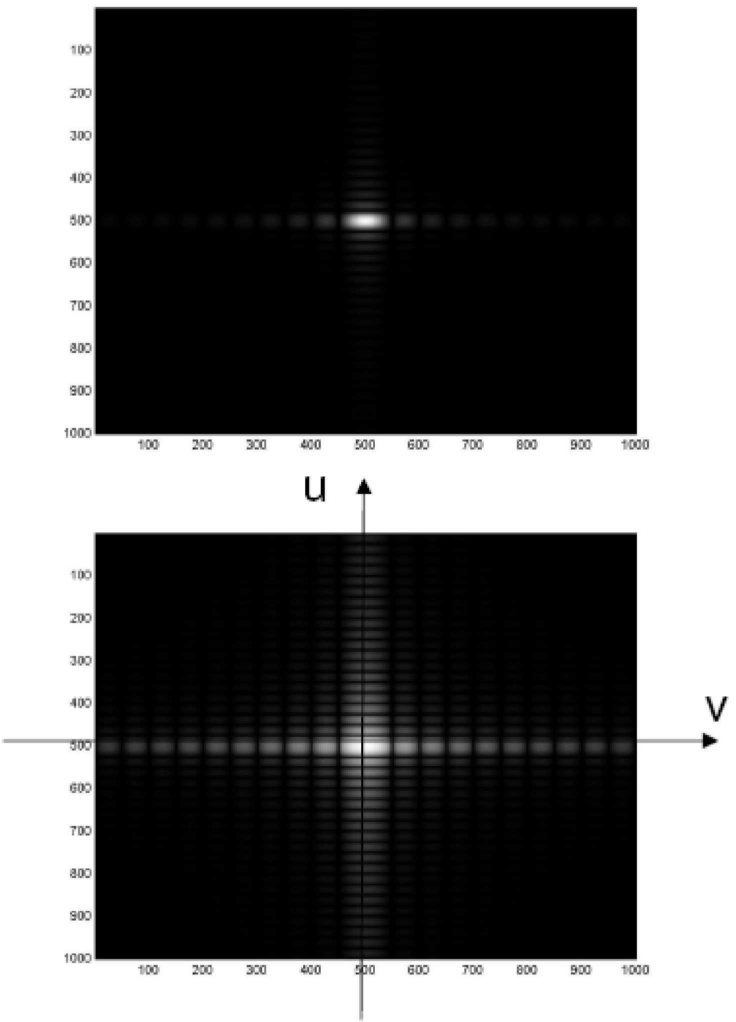
  - where F_x, F_y are 1D FT along x and y.
  - one can do 1DFT for each row of original image, then 1D FT along each column of resulting image
- Separable Signal
  - $f(x,y) = f_x(x)f_y(y)$
  - $F(u,v) = F_x(u)F_y(v)$,
    - where $F_x(u) = F_x\{f_x(x)\}$, $F_y(u) = F_y\{f_y(y)\}$
  - For separable signal, one can simply compute two 1D transforms!

Example 2

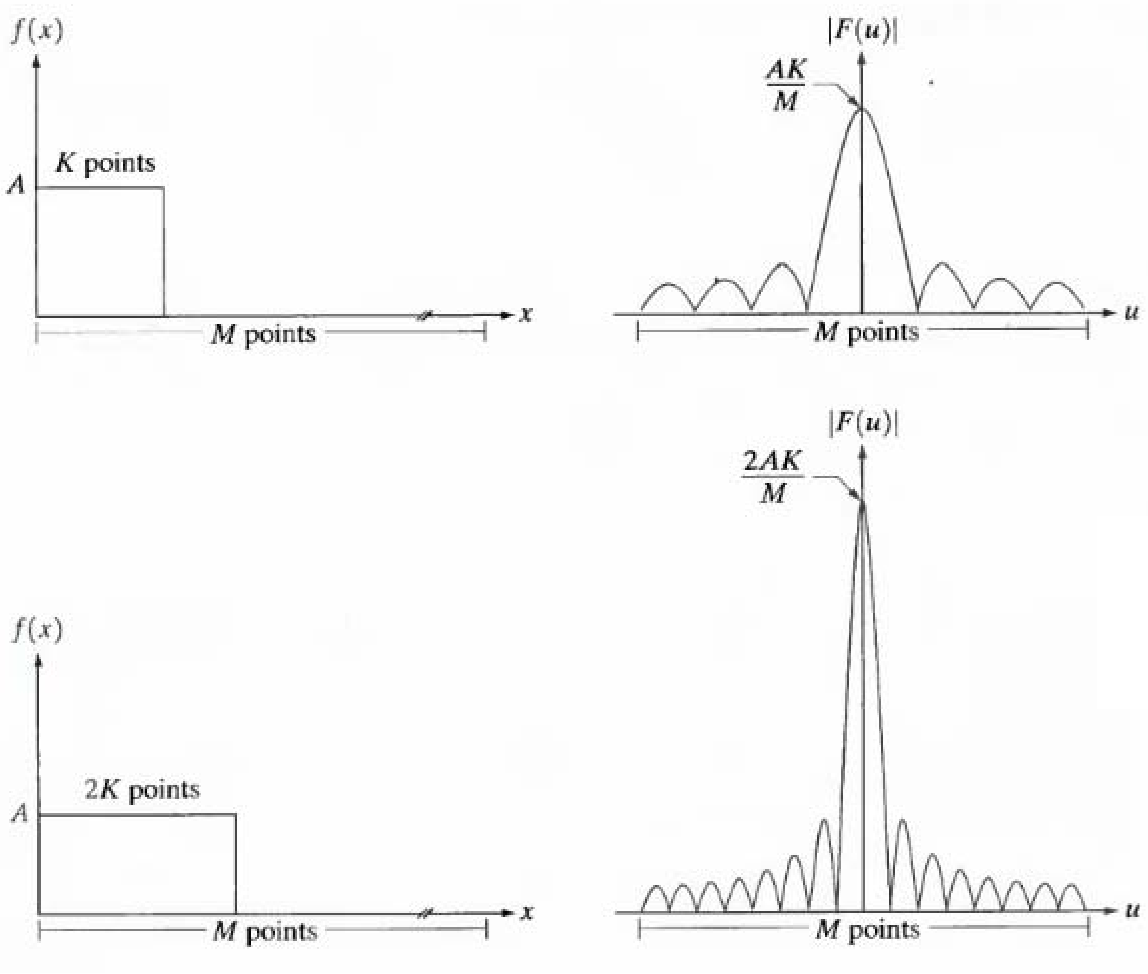
$$f(x,y) = \begin{cases} 1, & |x| \le x_0, |y| \le y_0 \\ 0, & otherwise \end{cases} \Rightarrow$$

$$F(u,v) = 4x_0 y_0 \operatorname{sinc}(2x_0 u)\operatorname{sinc}(2y_0 v)$$





Fourier Transform: Scaling



a b
c d

FIGURE 4.2 (a) A discrete function of M points, and (b) its Fourier spectrum. (c) A discrete function with twice the number of nonzero points, and (d) its Fourier spectrum.

$f(ax) \Leftrightarrow 1/|a|F(u/a)$

Rotation

- Let $x = r\cos\theta, \quad y = r\sin\theta, \quad u = \rho\cos\omega, \quad v = \rho\sin\omega.$
- 2D FT in polar coordinate (r, θ) and (ρ, Ø)

$$F(\rho,\phi) = \int_0^{\infty}\int_0^{2\pi} f(r,\theta)e^{-j2\pi(r\cos\theta\,\rho\cos\phi + r\sin\theta\,\rho\sin\phi)}rdrd\theta$$
$$= \iint f(r,\theta)e^{-j2\pi r\rho\cos(\theta-\phi)}rdrd\theta$$

- Property

$$f(r,\theta+\theta_0) \quad \Leftrightarrow \quad F(\rho,\phi+\theta_0)$$

Example of Rotation

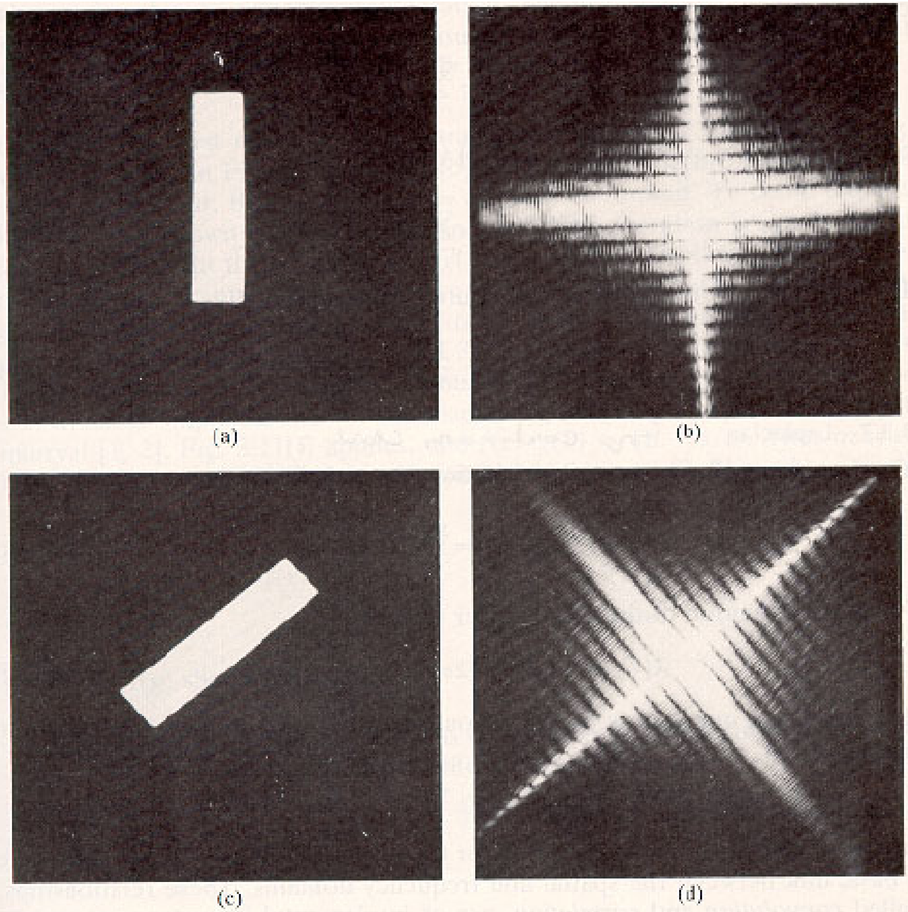


Figure 3.10 *Rotational properties of the Fourier transform: (a) a simple image; (b) spectrum; (c) rotated image; (d) resulting spectrum.*

Fourier Transform: Summary

Let *F*(*m*) incorporates both cosine and sine series coefficients, with the sine series distinguished by making it the imaginary component:

$$F(m) = F_m - jF_m' = \int f(t)\cos(mt)dt - j\cdot\int f(t)\sin(mt)dt$$

Let's now allow *f(t)* range from $-\infty$ to ∞, we rewrite:

$$\boxed{\Im\{f(t)\} = F(u) = \int_{-\infty}^{\infty} f(t)\exp(-j2\pi ut)dt}$$

F*(*u*)** is called the **Fourier Transform** of *f(t)*. We say that *f(t)* lives in the "**time domain**," and ***F*(*u*)** lives in the "**frequency domain**." ***u is called the frequency variable.

The Inverse Fourier Transform

We go from *f(t)* to *F(u)* by

Signal decomposition

Coefficient computation

$$\Im\{f(t)\} = F(u) = \int_{-\infty}^{\infty} f(t)\exp(-j2\pi ut)dt$$

Fourier Transform

Given *F*(*u*), *f(t)* can be obtained by the inverse Fourier transform

Signal reconstruction

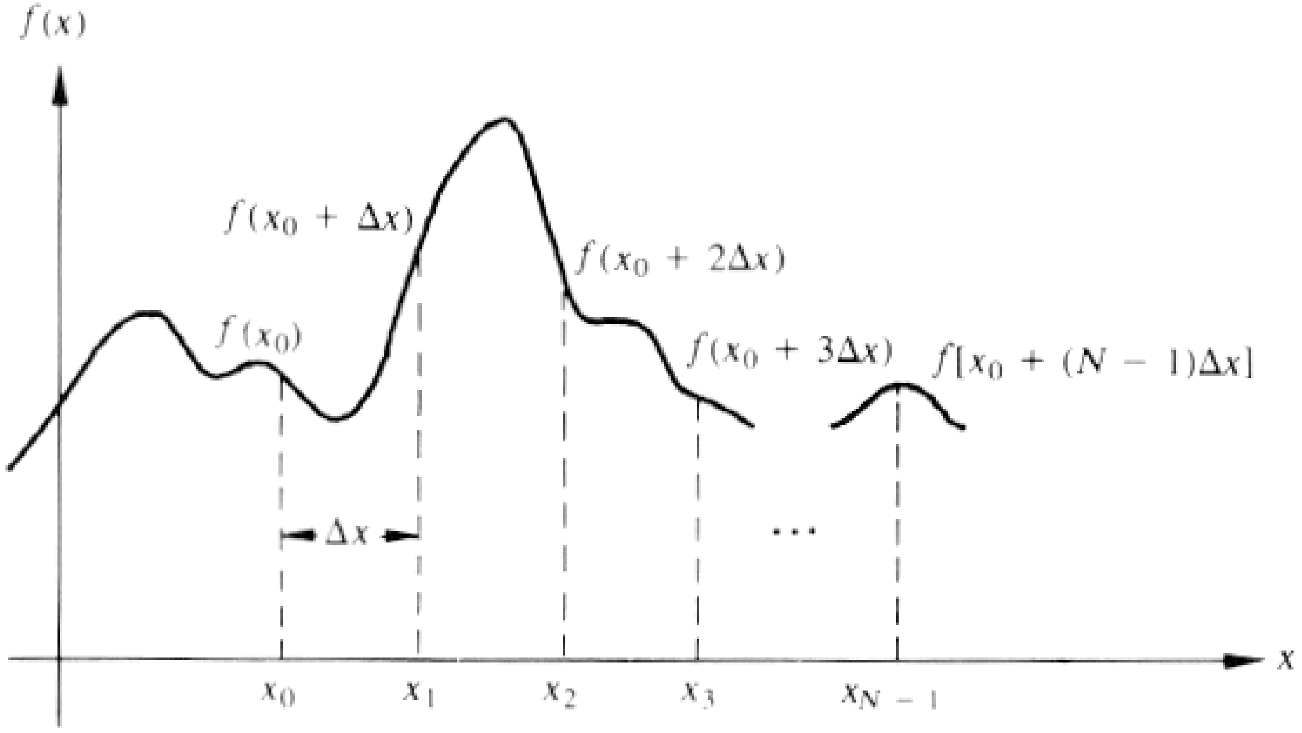
$$\Im^{-1}\{F(u)\} = f(t) = \int_{-\infty}^{\infty} F(u)\exp(j2\pi ut)du$$

Inverse Fourier Transform

Discrete Fourier Transform (DFT)

- A continuous function f(x) is discretized as:

$$\{f(x_0), f(x_0+\Delta x), f(x_0+2\Delta x), \ldots, f(x_0+(N-1)\Delta x)\}$$



Discrete Fourier Transform (DFT)

Let x denote the discrete values (x=0,1,2,…,M-1), i.e.

$$f(x) = f(x_0 + x\Delta x)$$

$$\{f(x_0), f(x_0 + \Delta x), f(x_0 + 2\Delta x), \ldots, f(x_0 + (M-1)\Delta x)\}$$

$$\Downarrow$$

$$\{f(0), f(1), f(2), \ldots, f(M-1)\}$$

Discrete Fourier Transform (DFT)

- The discrete Fourier transform pair that applies to sampled functions is given by:

Signal decomposition

Coefficient computation

$$\boxed{F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)\exp(-j2\pi ux/M)}$$

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

and

Signal reconstruction

$$\boxed{f(x) = \sum_{u=0}^{M-1} F(u)\exp(j2\pi ux/M)}$$

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

2-D Discrete Fourier Transform

- In 2-D case, the DFT pair is:

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp(-j2\pi(ux/M + vy/N))$$

u=0,1,2,…,M-1 and v=0,1,2,…,N-1

and:

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\exp(j2\pi(ux/M + vy/N))$$

x=0,1,2,…,M-1 and y=0,1,2,…,N-1

Polar Coordinate Representation of FT

- The Fourier transform of a real function is generally complex and we use polar coordinates:

$$F(u,v) = R(u,v) + j \cdot I(u,v)$$

$\Downarrow$ Polar coordinate

$$F(u,v) = |F(u,v)| \exp(j\phi(u,v))$$

Magnitude: $$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$$

Phase: $$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

Symmetry of FT for Real Image

- For real image f(x,y), FT is conjugate symmetric(共轭对称):

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

- The magnitude of FT is symmetric:

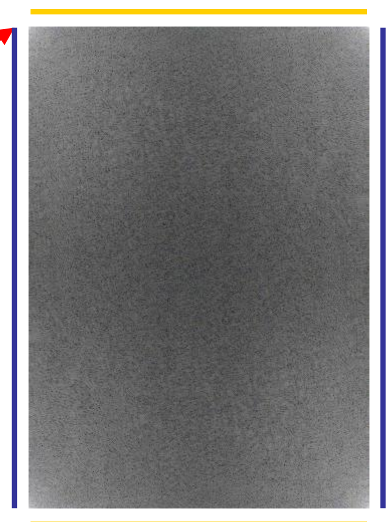
$$|F(u,v)| = |F(-u,-v)|$$

Matlab's `fftshift` and `ifftshift`

`I = ifftshift(J):`

origin

from FFT2
or ifftshift



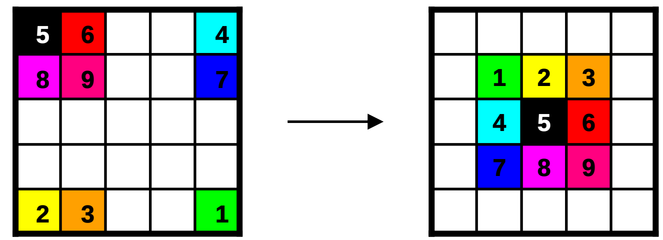
$$J\left(\lfloor R/2 \rfloor + 1, \lfloor C/2 \rfloor + 1\right) \rightarrow I(1,1)$$

where $\lfloor x \rfloor$ = `floor(x)` = the largest integer smaller than x.

Matlab's `fftshift` and `ifftshift`

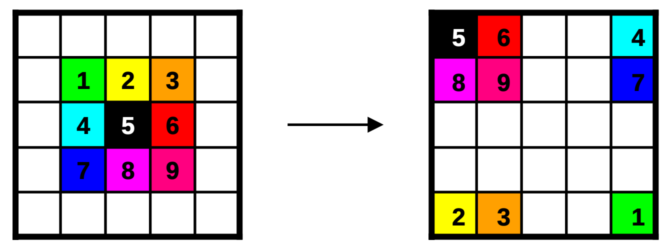
`J = fftshift(I):`

$$I(1,1) \rightarrow J(\lfloor R/2 \rfloor + 1, \lfloor C/2 \rfloor + 1)$$



`I = ifftshift(J):`

$$J(\lfloor R/2 \rfloor + 1, \lfloor C/2 \rfloor + 1) \rightarrow I(1,1)$$

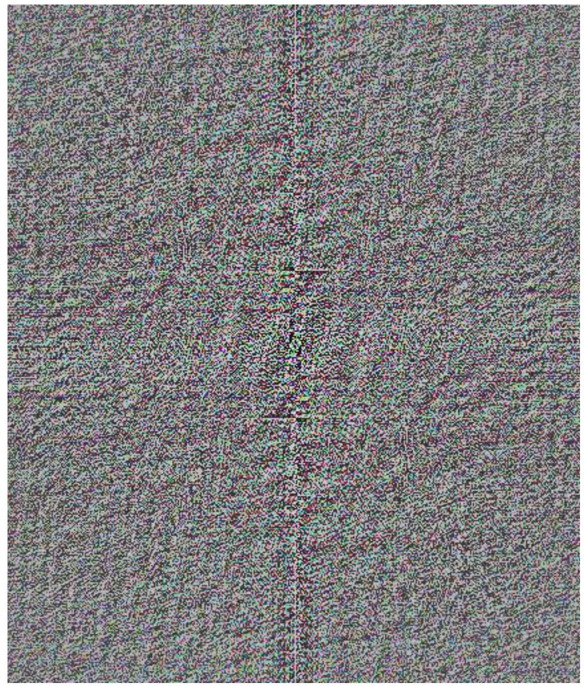


where $\lfloor x \rfloor$ = `floor(x)` = the largest integer smaller than x.

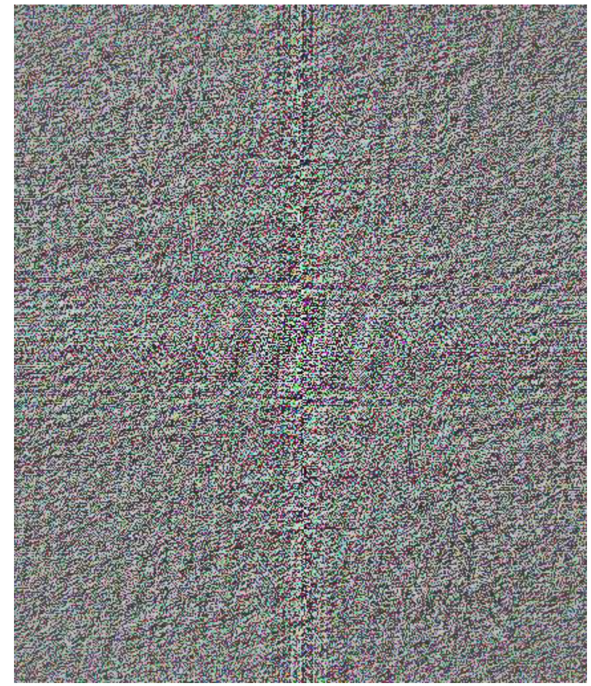
FT of an Image (Real + Imaginary)

Symmetry…

I



$\mathrm{Re}[\mathrm{F}\{I\}]$

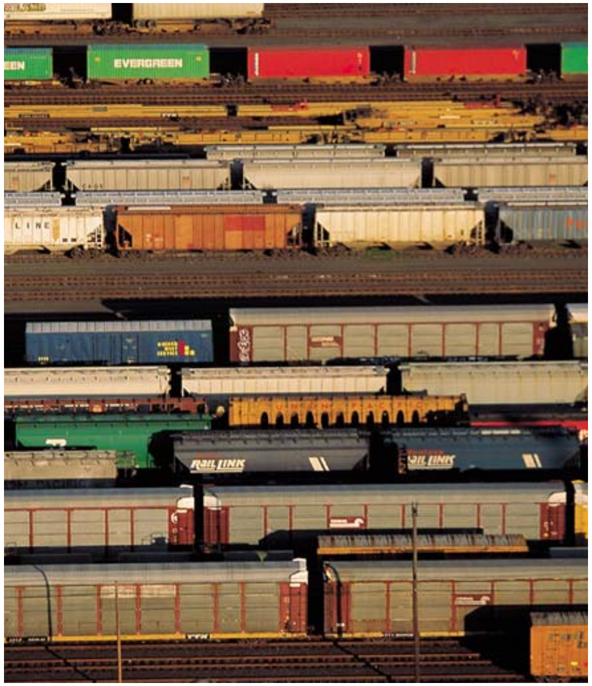


$\mathrm{Im}[\mathrm{F}\{I\}]$

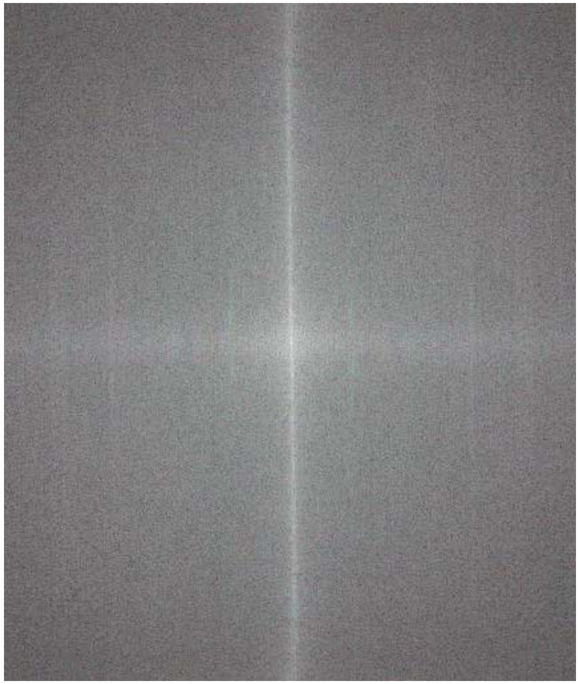
$$F(u,v) = F^*(-u,-v)$$

Re(u)=Re(-u), Im(u)=-Im(-u)

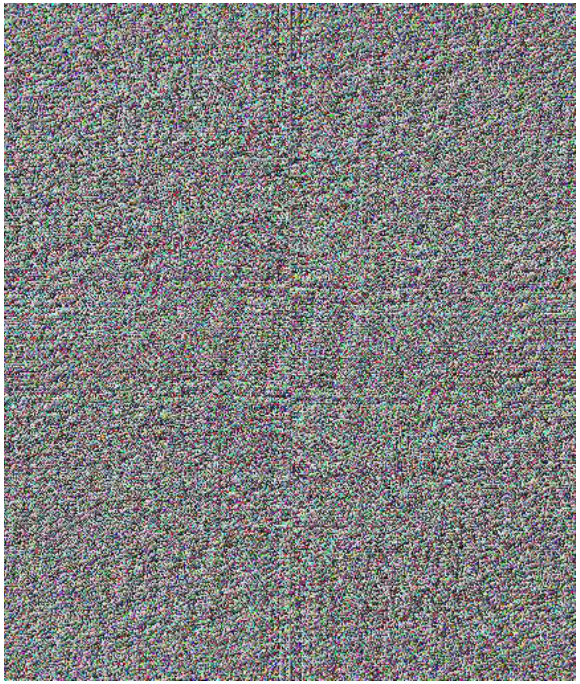
FT of an Image (Magnitude + Phase)



I



$\log\{|\mathrm{F}\{I\}|^2+1\}$



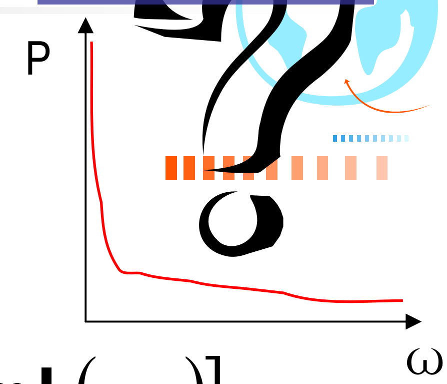
$\angle[\mathrm{F}\{I\}]$

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

$A(u)=A(-u),\ \Phi(u)=-\Phi(-u)$

The Power Spectrum (功率谱)

For display, the log of the power spectrum is often used.



The power spectrum of a signal is the square of the magnitude of its Fourier Transform.

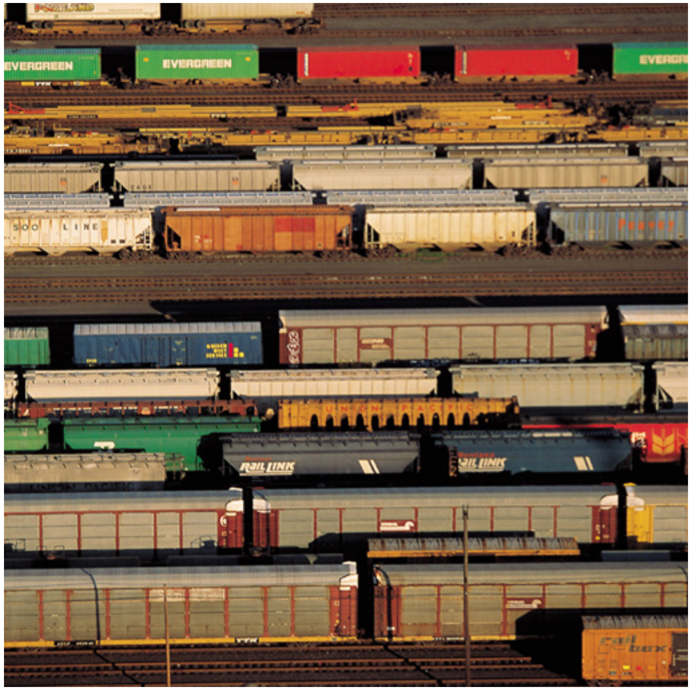
$$
\begin{aligned}
|I(u,v)|^2 &= I(u,v)\,I^*(u,v) \\
&= [\mathrm{Re}\,I(u,v)+i\,\mathrm{Im}\,I(u,v)][\mathrm{Re}\,I(u,v)-i\,\mathrm{Im}\,I(u,v)] \\
&= [\mathrm{Re}\,I(u,v)]^2+[\mathrm{Im}\,I(u,v)]^2.
\end{aligned}
$$

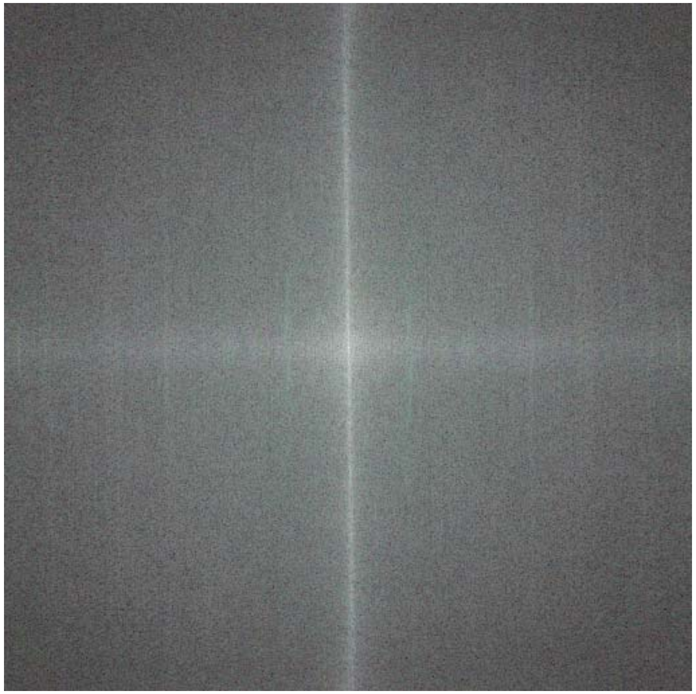
At each location (u,v) it indicates the squared intensity of the frequency component with period $\lambda = 1/\sqrt{u^2+v^2}$ and orientation $\theta = \tan^{-1}(v/u)$.

For display in Matlab:

```
PS = fftshift(2*log(abs(fft2(I))+1));
```

Power Spectrum of an Image





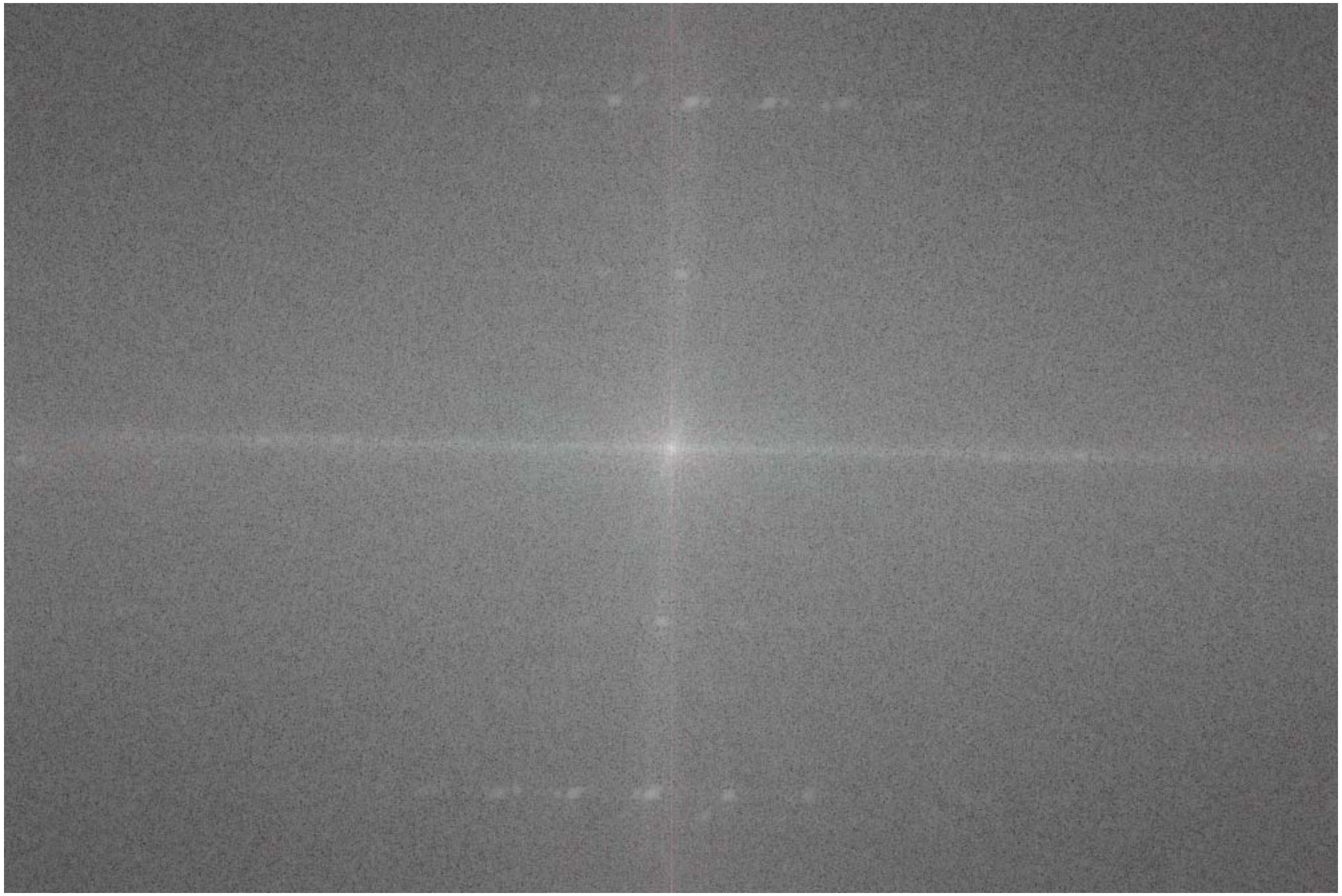
$\log\{|\mathrm{F}\{I\}|^2+1\}$

Fourier Magnitude and Phase

I

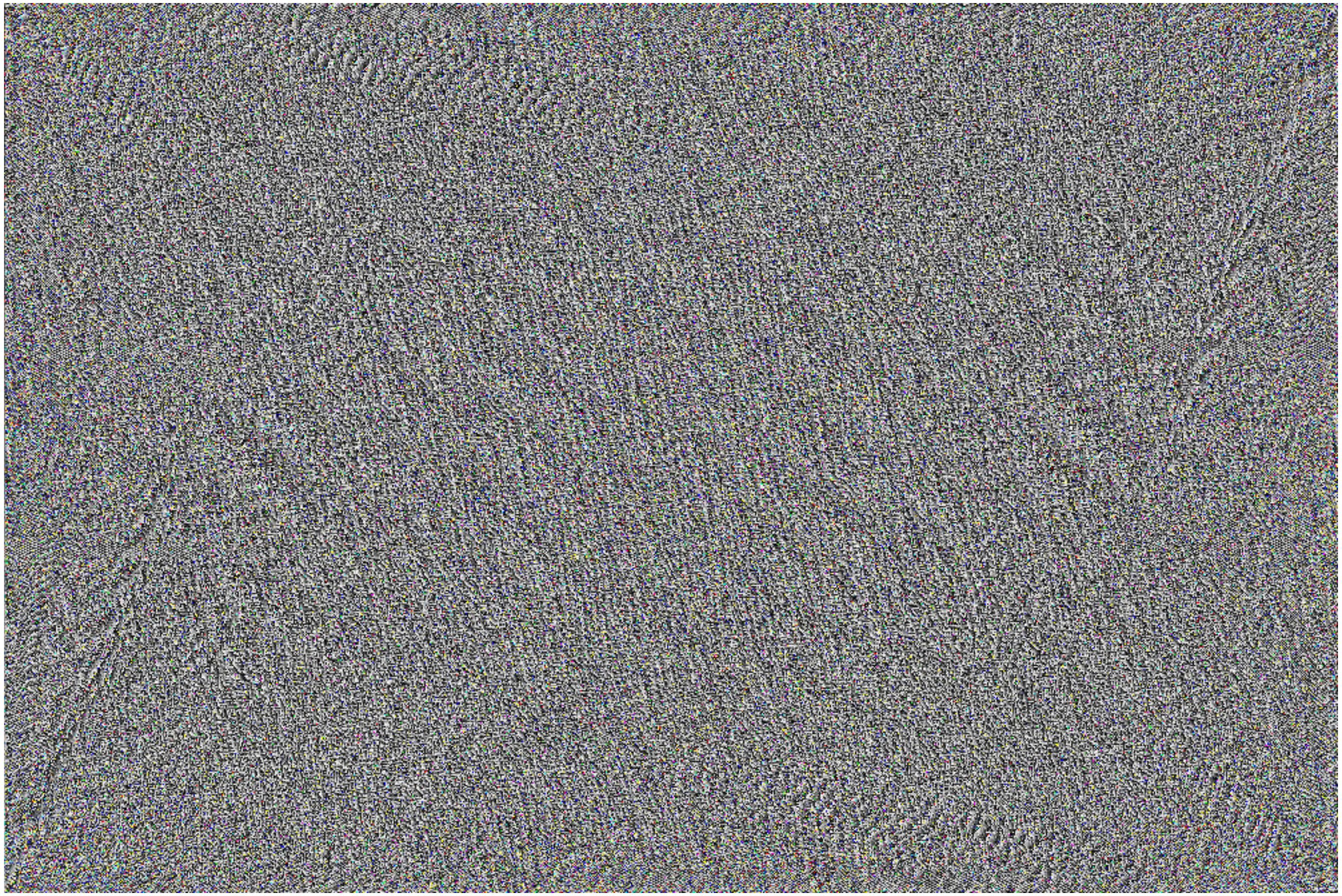
Fourier Magnitude

$\log\left|\mathbf{F}\{I\}\right|$



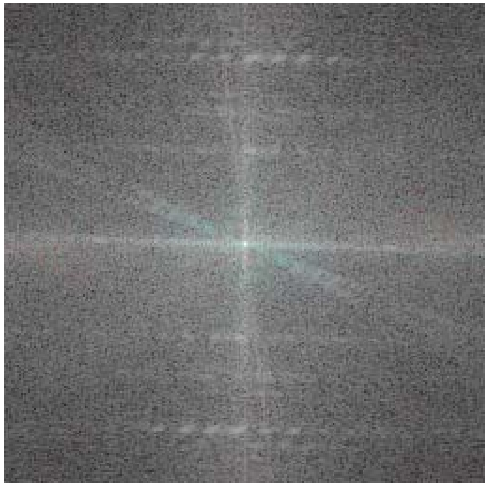
Fourier Phase

$\angle \mathbf{F}\{I\}$

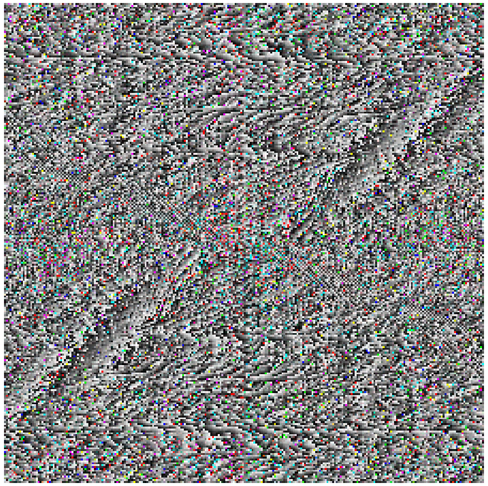


Q: Which contains more visually relevant information: magnitude or phase?

original image

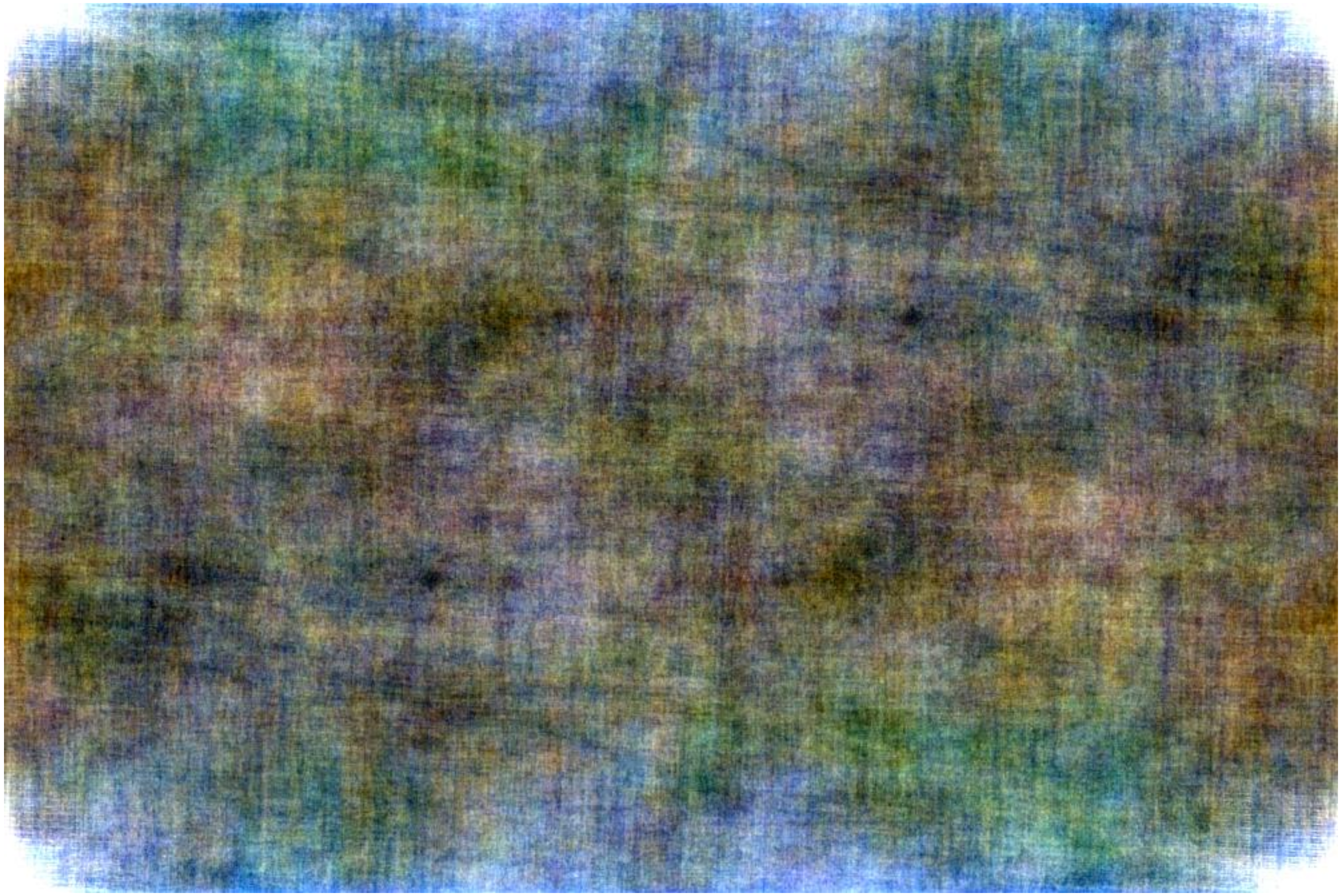


Fourier log magnitude



Fourier phase

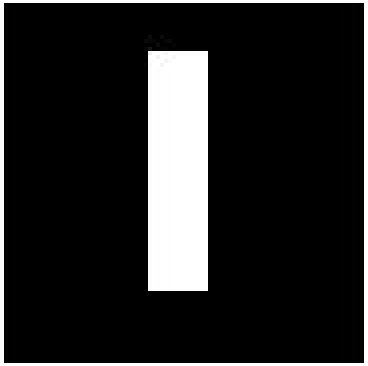
Magnitude Only Reconstruction

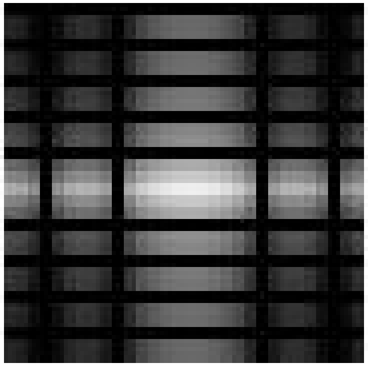


Phase Only Reconstruction

DFT – Matlab demo

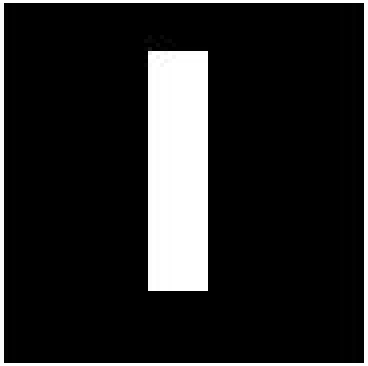
```
f = zeros(30,30);
f(5:24,13:17) = 1.00;
figure('Position',[200 200 90 90]);
imshow(f,'InitialMagnification','fit');
F = fft2(f);
F2=fftshift(F);
F3 = log(abs(F2)+0.0000001);
figure('Position',[500 500 90 90]);
imshow(F3,[-1 5],'InitialMagnification','fit');
```

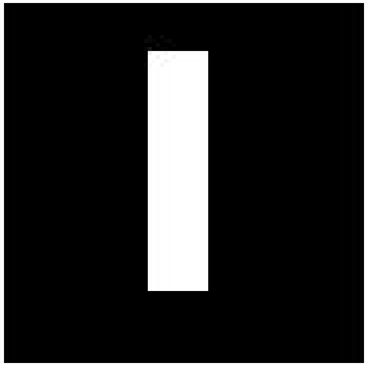




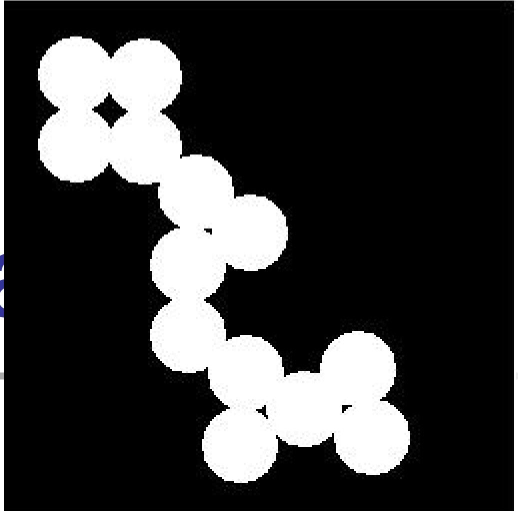
DFT Rebuild – Matlab demo

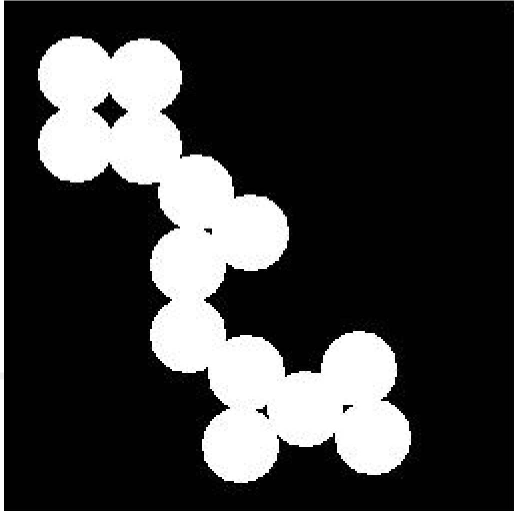
```
f = zeros(30,30);
f(5:24,13:17) = 1.00;
figure('Position',[200 200 90 90]);
imshow(f,'InitialMagnification','fit');
F = fft2(f);
%F2=fftshift(F);
rebuiltf=ifft2(F);
figure('Position',[500 500 90 90]);
imshow(rebuiltf,'InitialMagnification','fit');
```

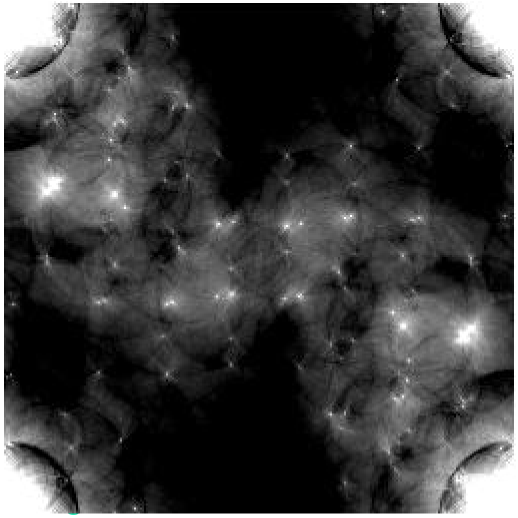


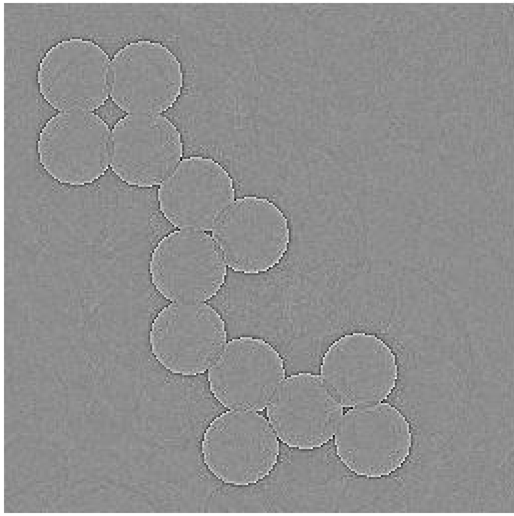


DFT – Matla



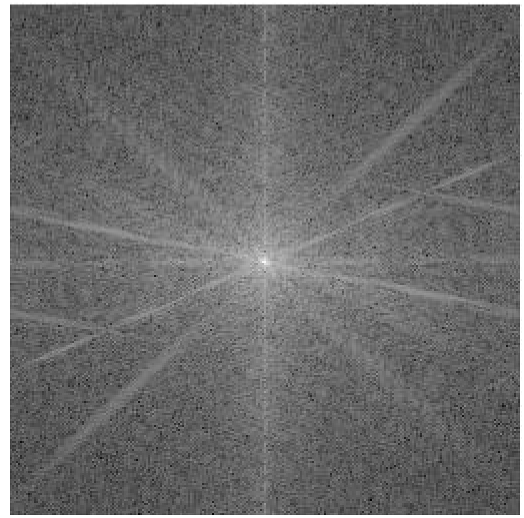


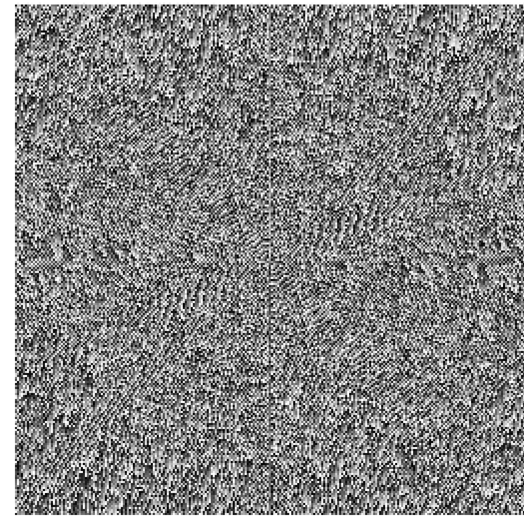




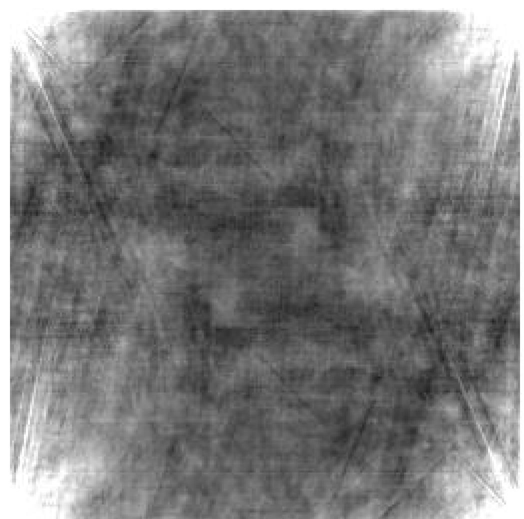
```
clear all;
close all;
a=imread('circles.png');
b=double(a);
figure;imshow(b);        %output image 1: original
Fb = fft2(b);
rebuiltb=ifft2(Fb);
figure;imshow(rebuiltb);        %output image 2: rebuilt image
rebuiltmb=ifft2(abs(Fb));        %magnitude-only reconstruction
figure;imshow(rebuiltmb);        %output image 3: abs-rebuilt image
rebuiltpb=ifft2( Fb./(abs(Fb)+0.00000001) );    %phase-only reconstruction
figure;imshow(rebuiltpb,[]);     %output image 4: phase-rebuilt image
```

DFT – Matlab demo

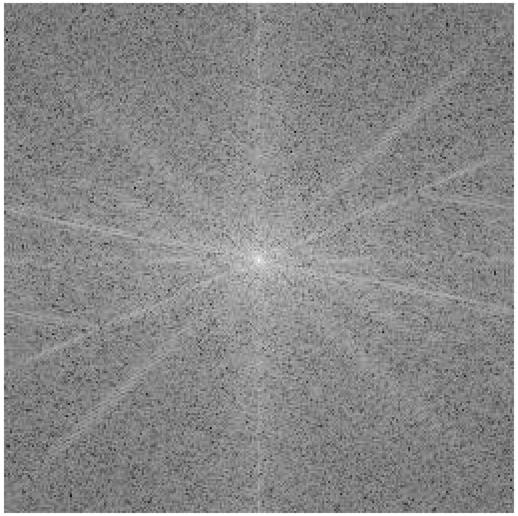


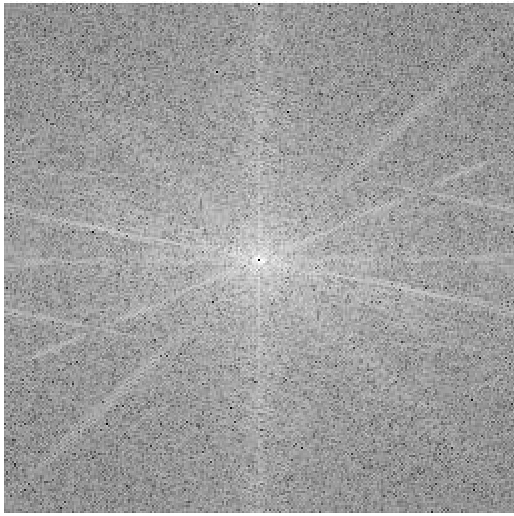


```
clear all;close all;
a=imread('cameraman.tif');b=im2double(a);
figure;imshow(b);
Fb = fft2(b); Fbshift=fftshift(Fb);
figure;imshow(log(abs(Fbshift)),[]);
figure;imshow(angle(Fbshift),[]);
rebuiltmb=ifft2(abs(Fb));
figure;imshow(rebuiltmb);
rebuiltpb=ifft2(Fb./abs(Fb));
figure;imshow(rebuiltpb,[]);
```



DFT – Matlab demo





```
clear all;close all;
a=imread('cameraman.tif');b=im2double(a);
figure;imshow(b);
Fb = fft2(b);Fbshift=fftshift(Fb);
figure;imshow(log(abs(real(Fbshift))),[]);
figure;imshow(log(abs(imag(Fbshift))),[]);
rebuiltrb=ifft2(real(Fb));
figure;imshow(rebuiltrb,[]);
rebuiltib=ifft2(Fb-real(Fb));
figure;imshow(rebuiltib,[]);
```