Filtering in the Frequency Domain

Fourier Transform and Applications

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

Any univariate (單因素) function can be rewritten as a weighted sum of sines and cosines of different frequencies

Joseph Fourier (1768-1830)

Basic principles:

- A periodic function can be represented by the **sum** of sines/cosines 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.

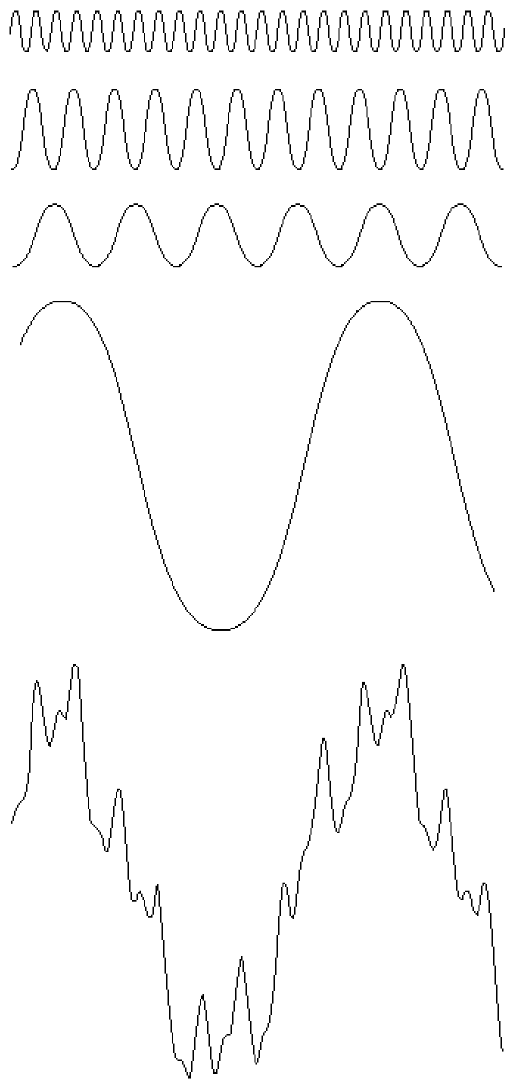
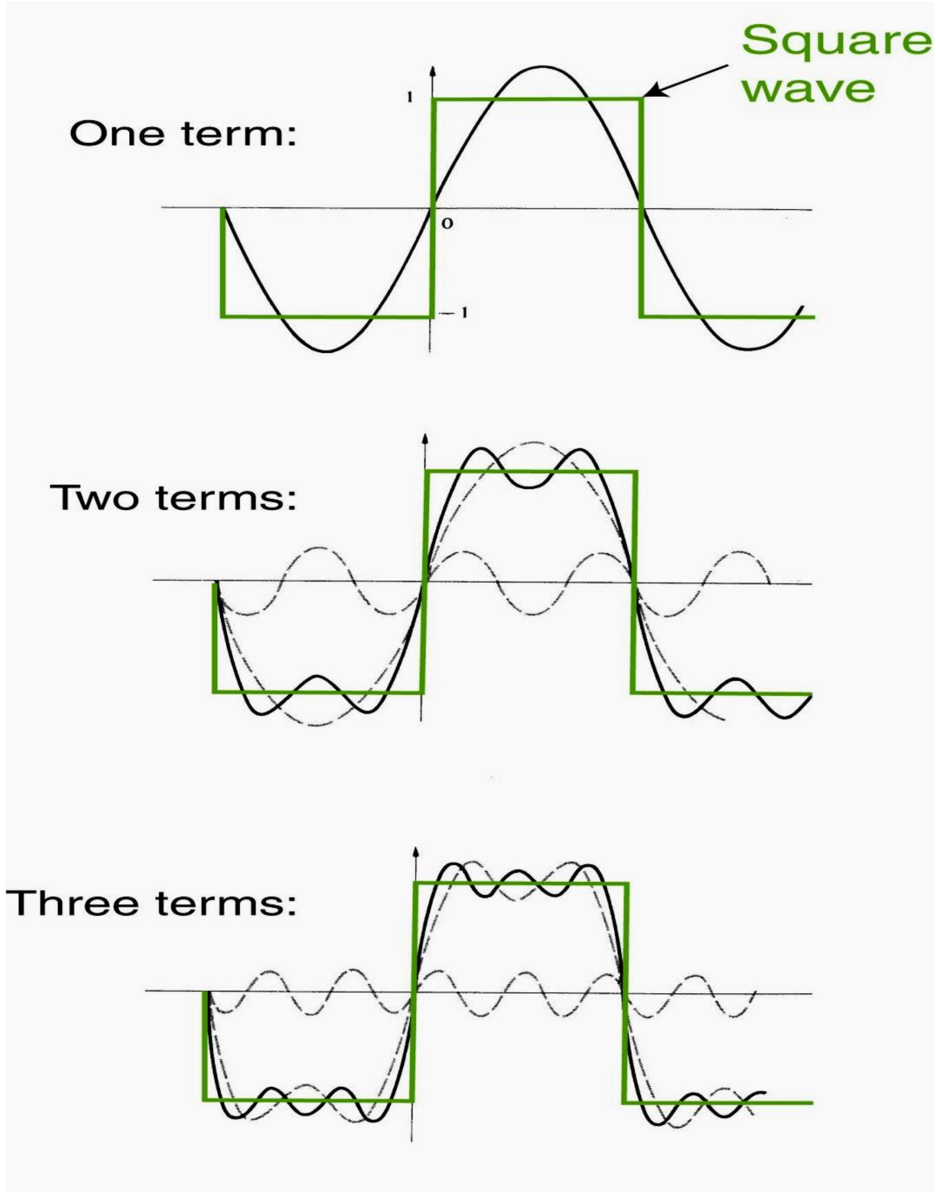


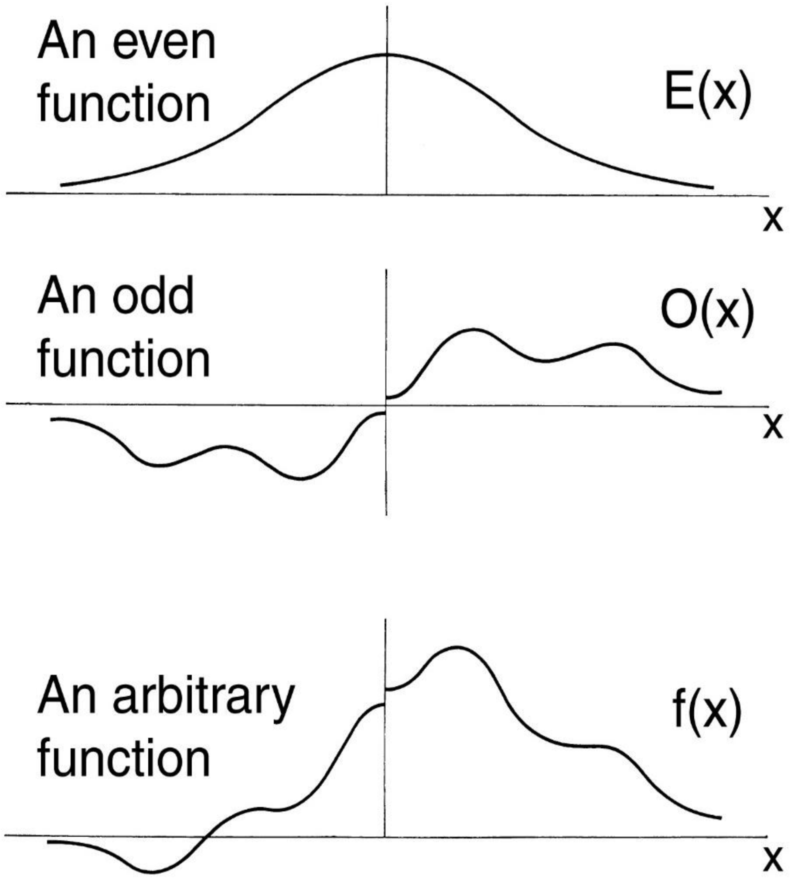
FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

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



$$E(x) \equiv [f(x) + f(-x)]/2$$

$$O(x) \equiv [f(x) - f(-x)]/2$$

$$\Downarrow$$

$$f(x) = E(x) + O(x)$$

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 (–π,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

where the Fourier series is

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

- Any function that periodically repeats itself can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficients.
- This sum is called a ***Fourier series*****.**

Fourier Series

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos nt + b_n \sin nt)$$

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) dt$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos nt \, dt$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin nt \, dt$$

Fourier Transform

- A function that is not periodic but the area under its curve is finite can be expressed as the integral of sines and/or cosines multiplied by a weighing function.
- The formulation in this case is ***Fourier transform***

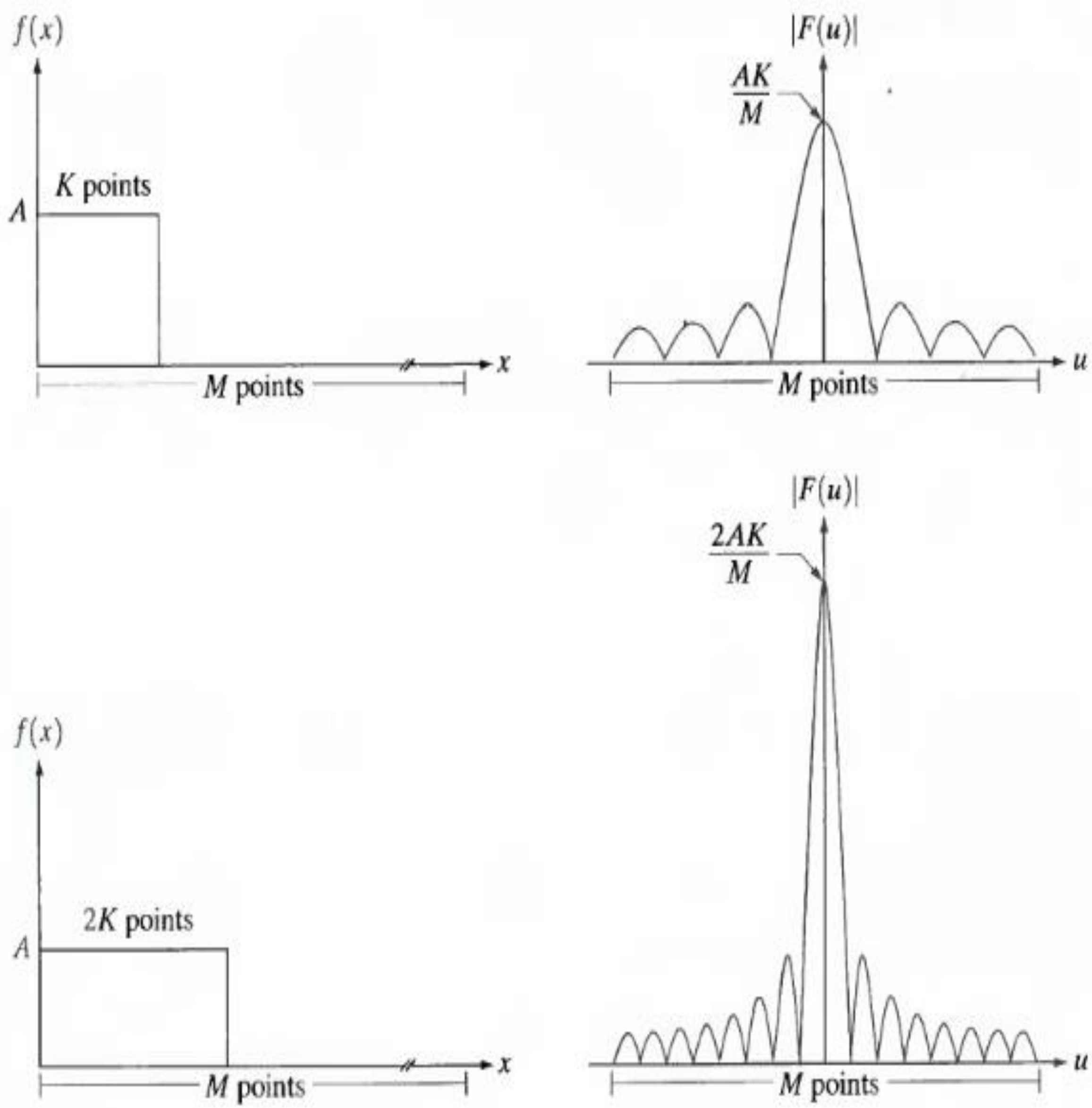
- Let $F(m)$ incorporates both cosine and sine series coefficients, with the sine series distinguished by making it the imaginary component:

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

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

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

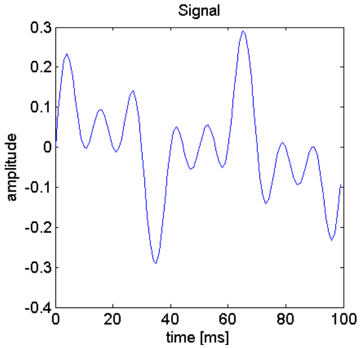
- ***F(u)*** is called the **Fourier Transform** of *f(t)*. We say that *f(t)* lives in the "**time domain**," and ***F(u)*** lives in the "**frequency domain**." ***u*** is called the frequency variable.

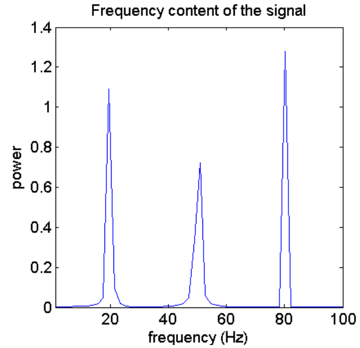


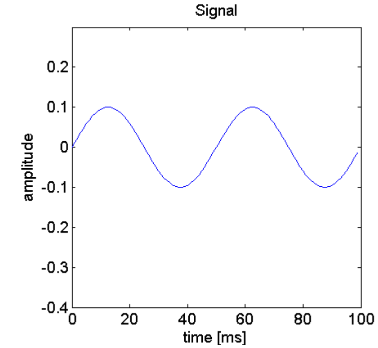
a b
c d

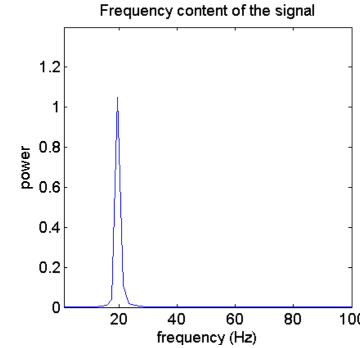
FIGURE 4.2 (a) A discrete function of M points, and (b) its Fourier spectrum. (c) A discrete function with twice the number of nonzero points, and (d) its Fourier spectrum.

$$\Delta u \cdot \Delta x \leq 1/M$$

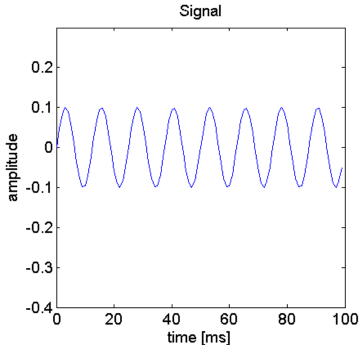


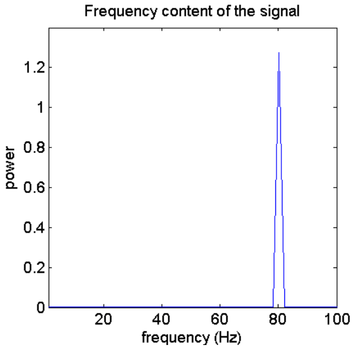


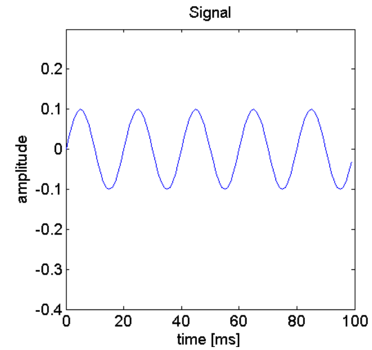


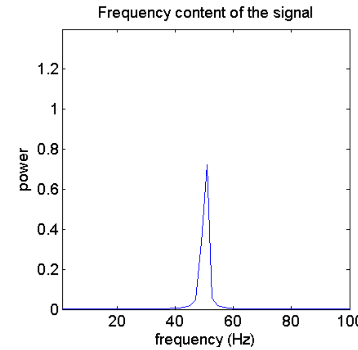


time domain **frequency domain** **time domain** **frequency domain**









time domain **frequency domain** **time domain** **frequency domain**

The Inverse Fourier Transform

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

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

Fourier Transform

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

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

Inverse Fourier Transform

Fourier Theory

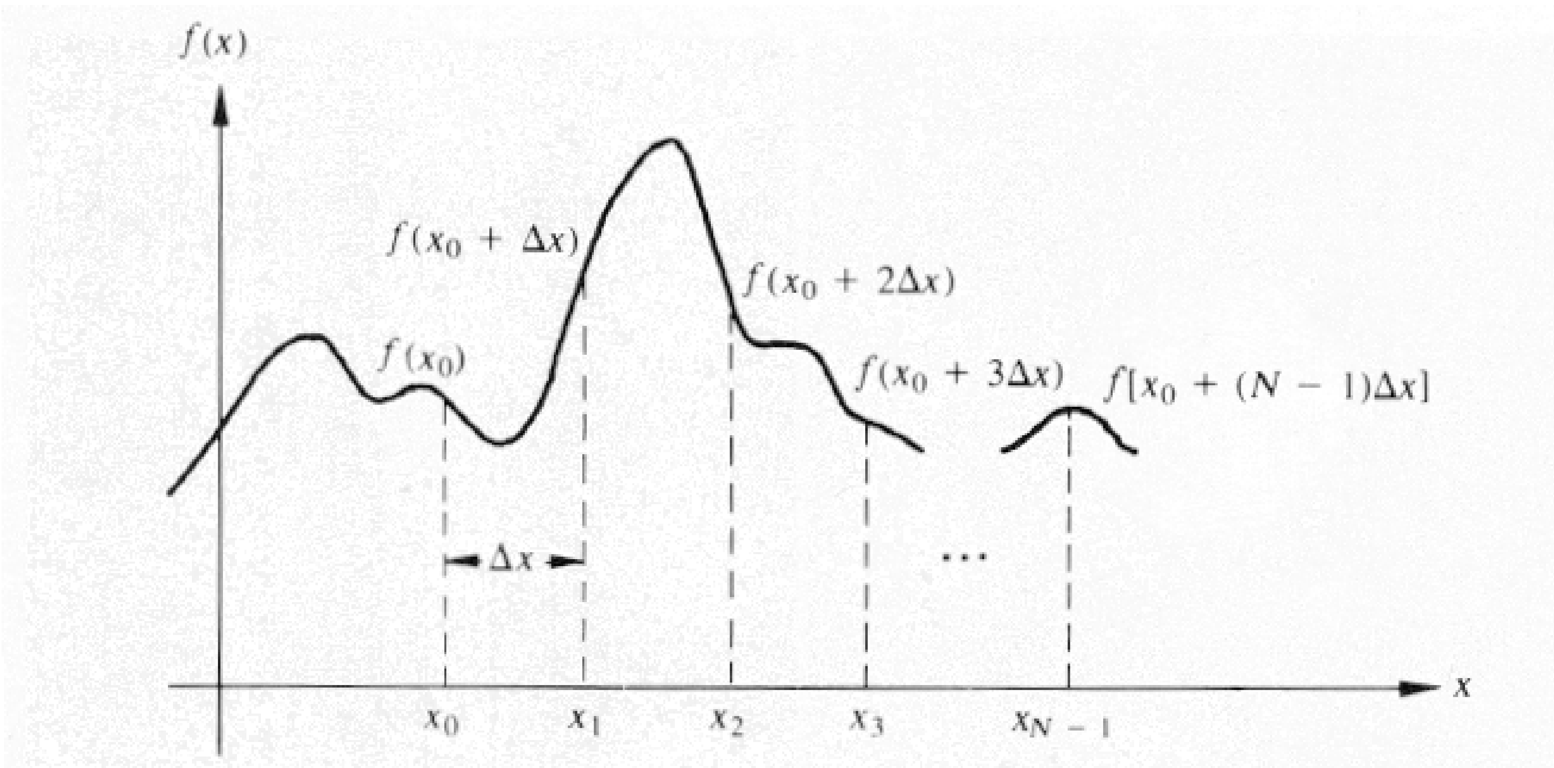
- Reconstruction
  - A function expressed in either a Fourier series or transform can be reconstructed (recovered) completely via an inverse process, without loss of information.

Discrete Fourier Transform (DFT)

- A continuous function f(x) is discretized into a sequence:

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

by taking N or M samples Δx units apart



$f(x)$
$f(x_0 + \Delta x)$
$f(x_0 + 2\Delta x)$
$f(x_0)$
$f(x_0 + 3\Delta x)$
$f[x_0 + (N-1)\Delta x]$
Δx
x
x_0
x_1
x_2
x_3
x_{N-1}

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

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

then

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

$$\Downarrow$$

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

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

$$\boxed{F(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(x)\exp(-j2\pi ux/M)} \quad u=0,1,2,\ldots,M\text{-}1$$

and

$$\boxed{f(x)=\sum_{u=0}^{M-1} F(u)\exp(j2\pi ux/M)} \quad x=0,1,2,\ldots,M\text{-}1$$

$$F(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-j2\pi u\frac{x}{M}} \qquad u=[0,1,2,\ \ldots,\ M\text{-}1]$$

$$F(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(x)\left[\cos 2\pi u\frac{x}{M}-j\sin 2\pi u\frac{x}{M}\right]$$

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

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

2-D Discrete Fourier Transform

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

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

and:

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

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

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

Polar coordinate

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

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

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

- The DFT is the sampled Fourier Transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to fully describe the spatial domain image.
- The number of frequencies corresponds to the number of pixels in the spatial domain image, *i.e.* **the image in the spatial and Fourier domain are of the same size**.

- The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components.
- The output of the transformation represents the image in the *Fourier* or *frequency domain*, while the input image is the *spatial domain* equivalent.
- In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image.

- The Fourier Transform is used in a wide range of applications
  - image analysis
  - image filtering
  - image reconstruction
  - image compression.

Implementation

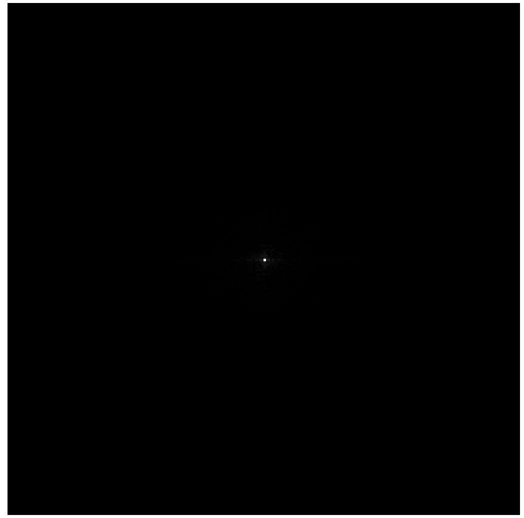
- The spatial domain image is first transformed into an intermediate image using *N* one-dimensional Fourier Transforms
- This intermediate image is then transformed into the final image, again using *N* one-dimensional Fourier Transforms.
- Expressing the two-dimensional Fourier Transform in terms of a series of *2N* one-dimensional transforms decreases the number of required computations.

- The Fourier Transform produces a complex number valued output image which can be displayed with two images, either with the *real* and *imaginary* part or with *magnitude* and *phase*.
- In image processing, often only the magnitude of the Fourier Transform is displayed, as it contains most of the information of the geometric structure of the spatial domain image.

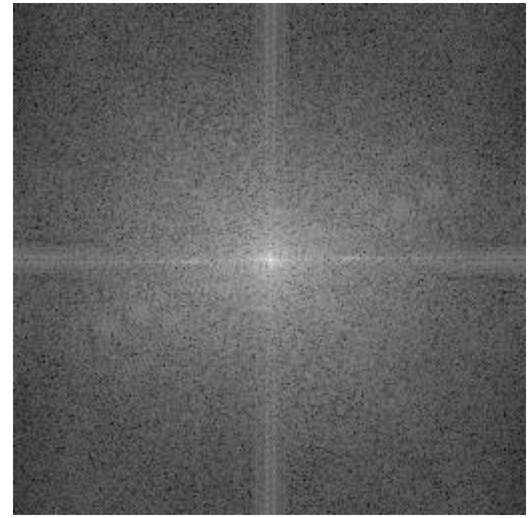
- If we want to re-transform the Fourier image into the correct spatial domain after some processing in the frequency domain, we must make sure to preserve both magnitude and phase of the Fourier image.
- The Fourier domain image has a much greater range than the image in the spatial domain. Hence, to be sufficiently accurate, its values are usually calculated and stored in float values.

- The Fourier Transform is used if we want to access the geometric characteristics of a spatial domain image
- Because the image in the Fourier domain is decomposed into its sinusoidal components, it is easy to examine or process certain frequencies of the image, thus influencing the geometric structure in the spatial domain.

(a)



(b)

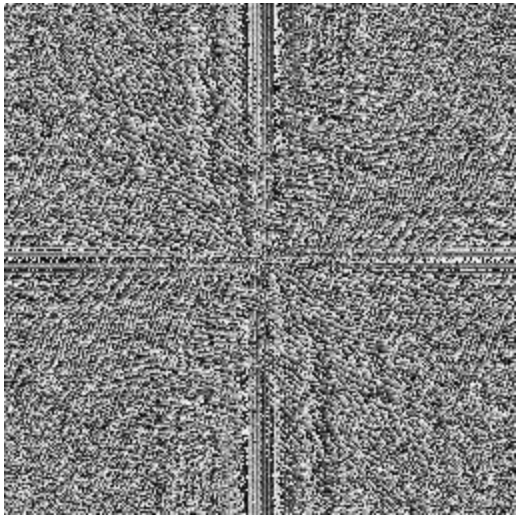


(c)

- The magnitude calculated from the complex result is shown in (b)
- The dynamic range of the Fourier coefficients (*i.e.* the intensity values in the Fourier image) is too large to be displayed on the screen, therefore all other values appear as black
- We apply a logarithmic transformation to the image we obtain

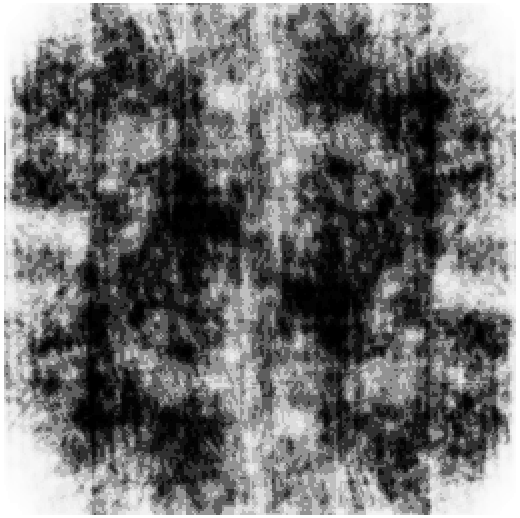
- The result shows that the image contains components of all frequencies, but that their magnitude gets smaller for higher frequencies.
- Hence, low frequencies contain more image information than the higher ones.
- The transform image also tells us that there are two dominating directions in the Fourier image, one passing vertically and one horizontally through the center.
- These originate from the regular patterns in the background of the original image.

- The phase of the Fourier transform of the same image
- The value of each point determines the phase of the corresponding frequency
- As in the magnitude image, we can identify the vertical and horizontal lines corresponding to the patterns in the original image



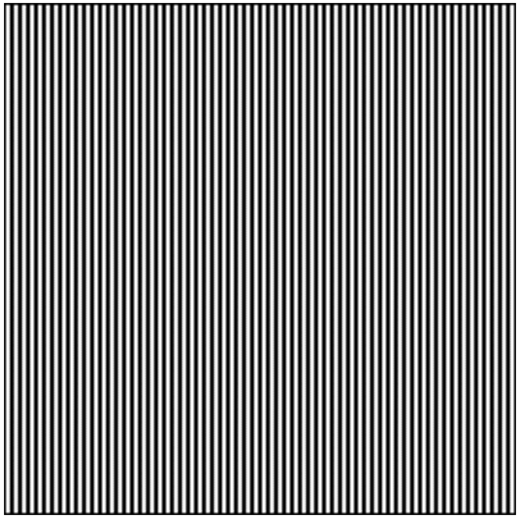
- The phase image does not yield much new information about the structure of the spatial domain image; therefore, we restrict ourselves to displaying only the magnitude of the Fourier Transform

- If we apply the inverse Fourier Transform to the above magnitude image while ignoring the phase we obtain
- Although this image contains the same frequencies (and amount of frequencies) as the original input image, it is corrupted beyond recognition. This shows that the phase information is crucial to reconstruct the correct image in the spatial domain.

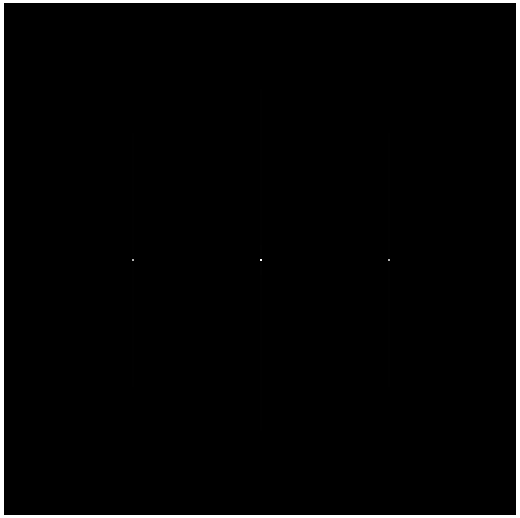


The nature of the transform

- The response of the Fourier Transform to periodic patterns in the spatial domain images can be seen very easily in the following artificial images.



(a)



(b)

- The original image shows 2 pixel wide vertical stripes.
- The magnitude of the Fourier transform of this image is shown in (b)

- If we look carefully, we can see that it contains 3 main values: the DC-value (F(0,0)) and, since the Fourier image is symmetrical to its center, two points corresponding to the frequency of the stripes in the original image.
- Note that the two points lie on a horizontal line through the image center, because the image intensity in the spatial domain changes the most if we go along it horizontally.

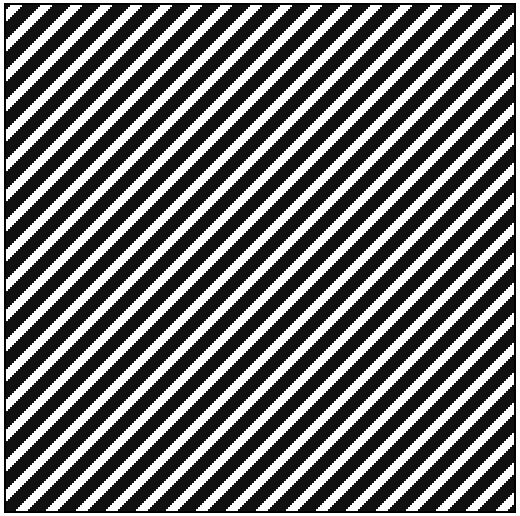
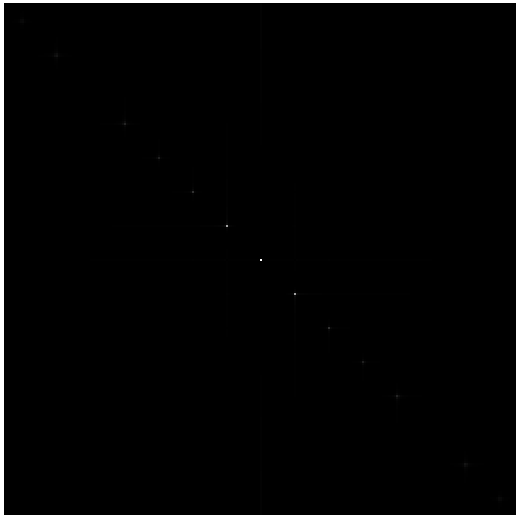
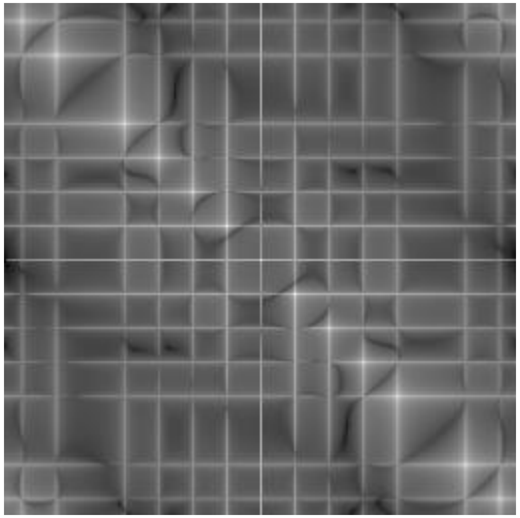
- The distance of the points to the center can be explained as follows: the maximum frequency which can be represented in the spatial domain are two pixel wide stripe pairs (one white, one black).

$$f_{max} = \frac{1}{2 \text{ pixels}}$$

Hence, the two pixel wide stripes in the above image represent

$$f = \frac{1}{4 \text{ pixels}} = \frac{f_{max}}{2}$$

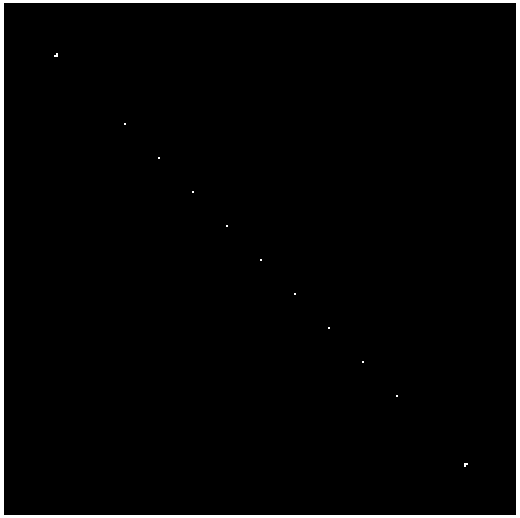
- Thus, the points in the Fourier image are halfway between the center and the edge of the image, *i.e.* the represented frequency is half of the maximum.

- The main components of the transformed image are the DC-value and the two points corresponding to the frequency of the stripes
- However, the logarithmic transform of the Fourier Transform

- It shows that now the image contains many minor frequencies.
- The main reason is that a diagonal can only be approximated by the square pixels of the image, hence, additional frequencies are needed to compose the image.
- The logarithmic scaling makes it difficult to tell the influence of single frequencies in the original image.

- To find the most important frequencies we threshold the original Fourier magnitude image at level 13.
- The resulting Fourier image shows all frequencies whose magnitude is at least 5% of the main peak.

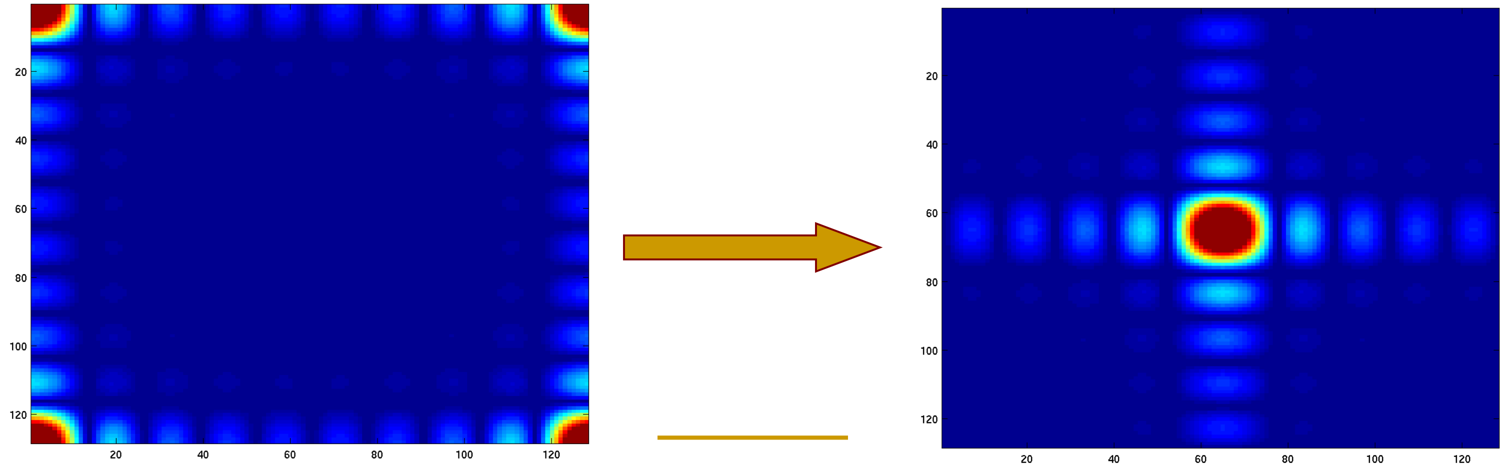


Fourier Transform: shift

- It is common to multiply input image by $(-1)^{x+y}$ prior to computing the FT. This shift the center of the FT to (M/2,N/2).

$$\Im\{f(x,y)\} = F(u,v)$$

$$\Im\{f(x,y)(-1)^{x+y}\} = F(u-M/2, v-N/2)$$



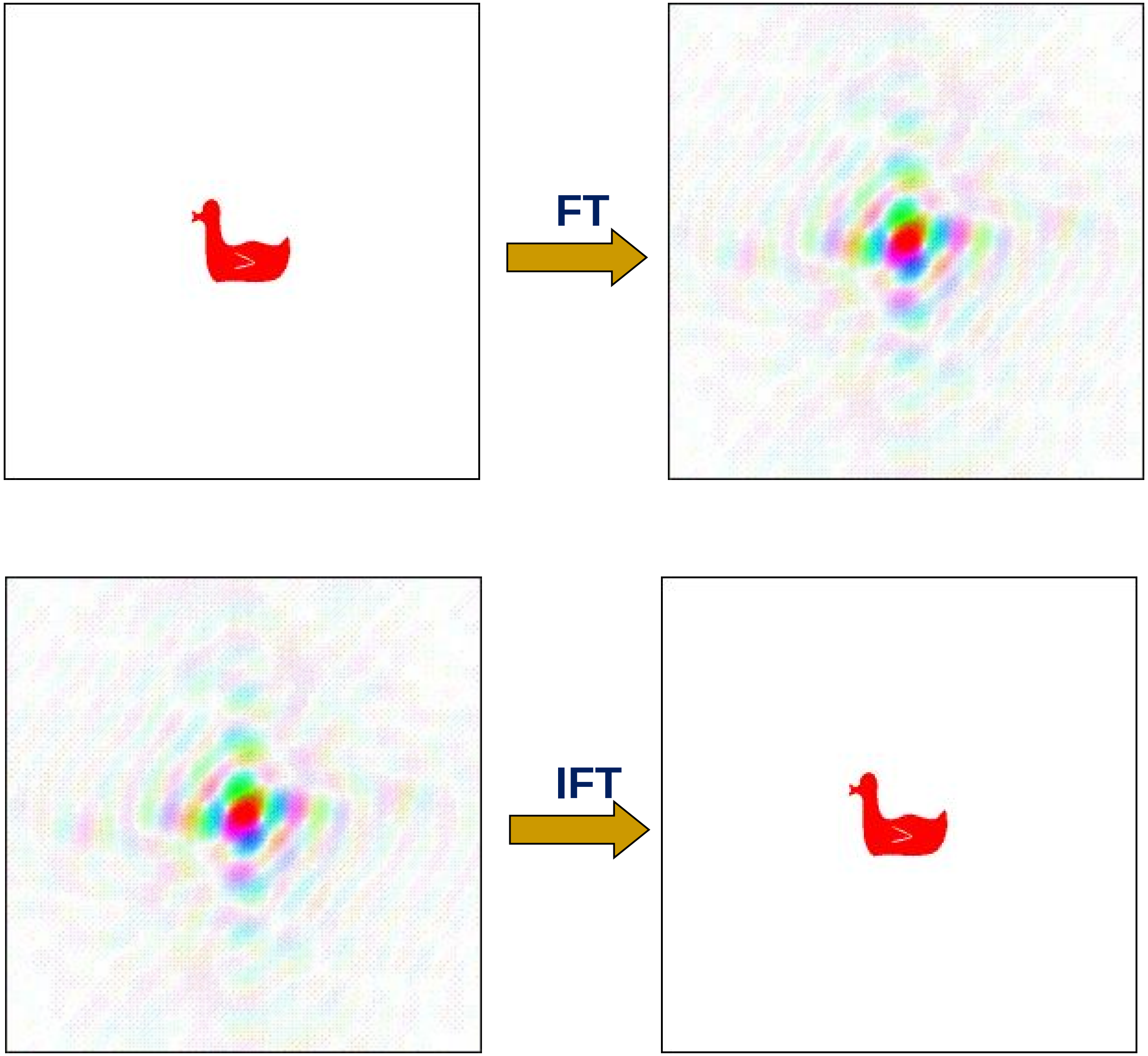
Symmetry of FT

- For real image f(x,y), FT is conjugate (共軛) symmetric:

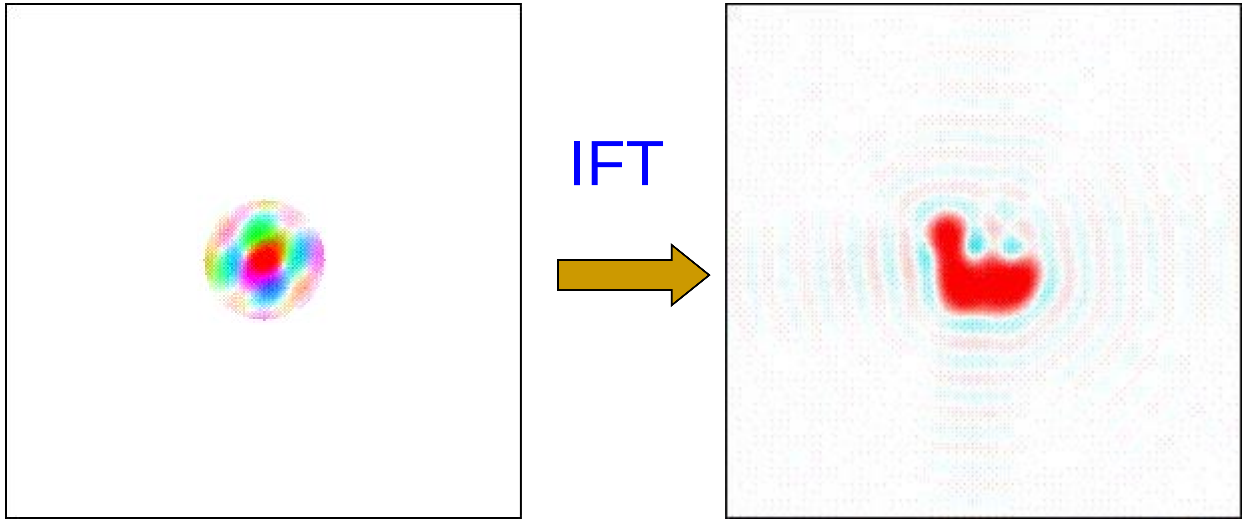
$$F(u,v) = F^*(-u,-v)$$

- The magnitude of FT is symmetric:

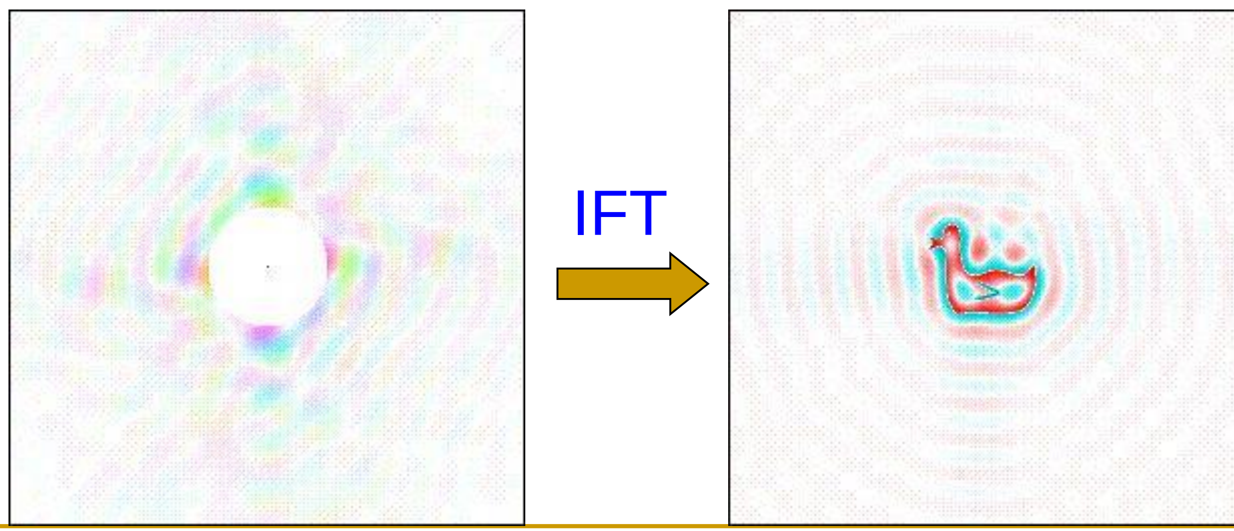
$$|F(u,v)| = |F(-u,-v)|$$

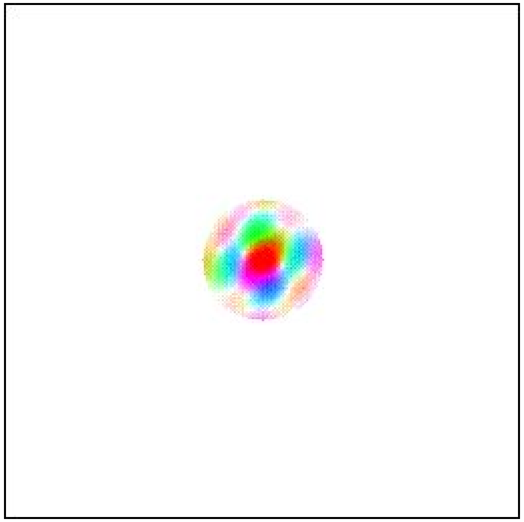

FT
IFT

run a Low Pass Filter



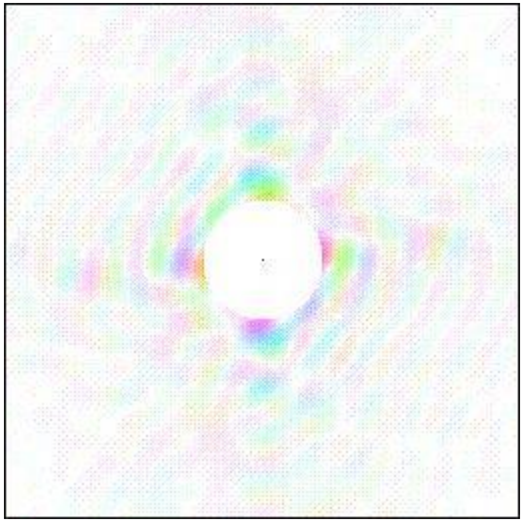
run a High Pass Filter





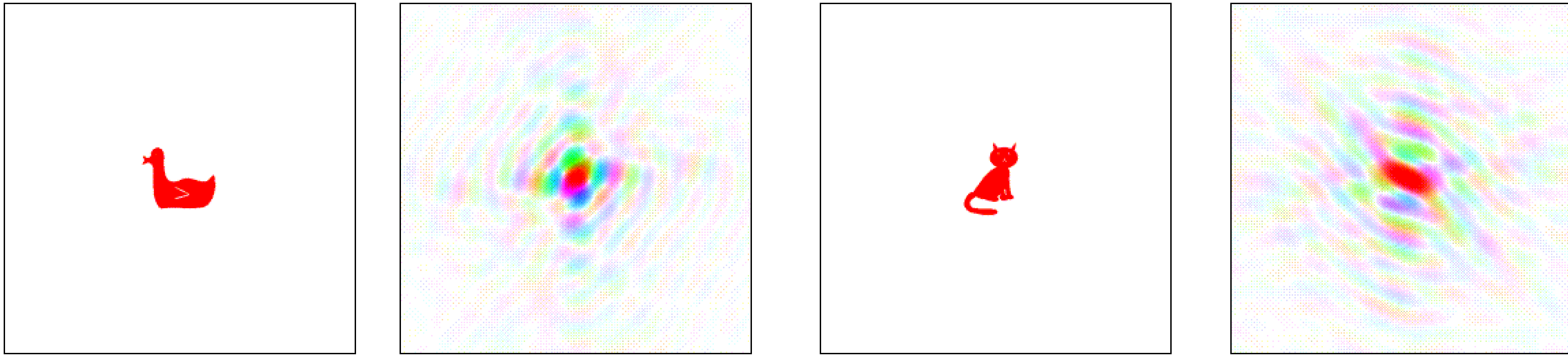
The central part of FT, i.e.

The low frequency components are responsible for the general gray-level appearance of an image.

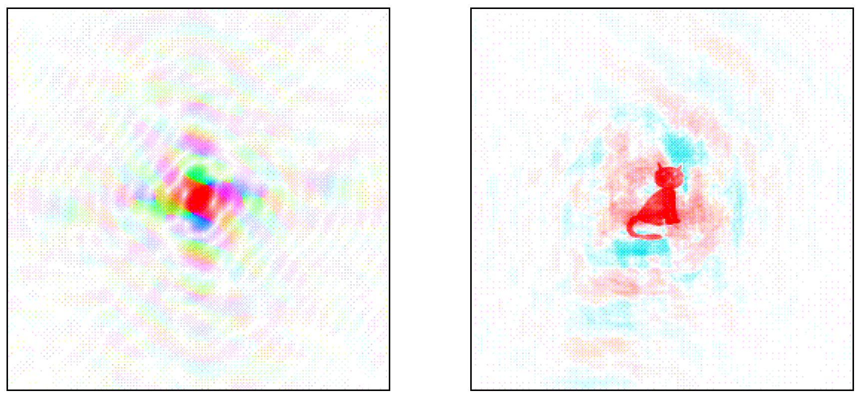


The high frequency components of FT are responsible for the detail information of an image

Magic Tricks



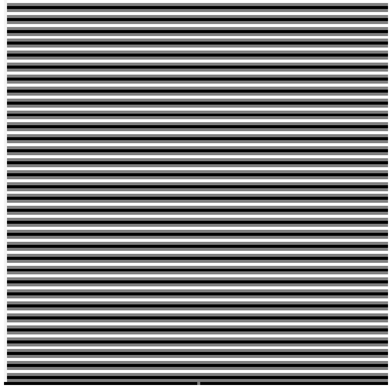
If an image is made that combines the magnitudes of the duck with the phases of the cat you get interesting results:

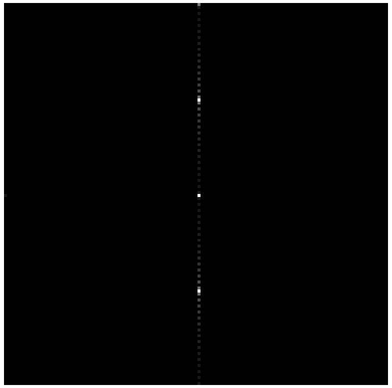


The phases contribute most of the structural information for this plot. Unfortunately FT images we deal with only give magnitude information so much of this information is lost.

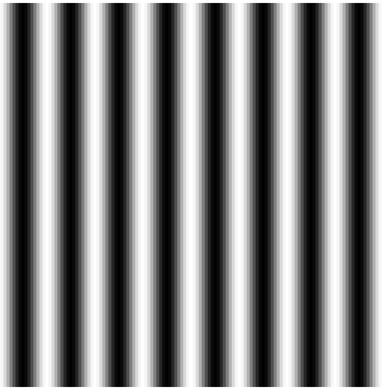
Fourier Transform Images are from: http://www.ysbl.york.ac.uk/~cowtan/fourier/fourier.html

This image exclusively has 32 cycles in the vertical direction.





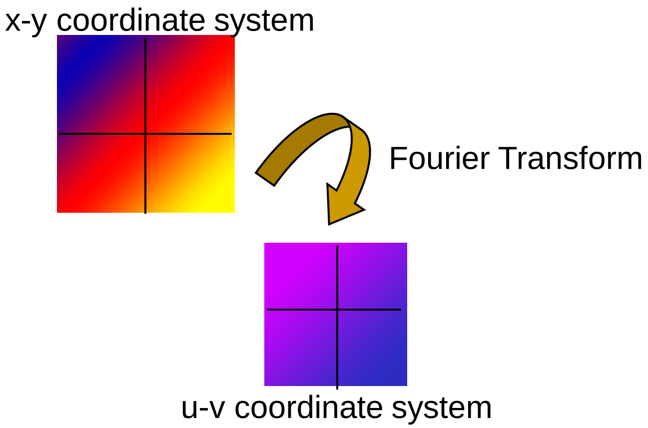
This image exclusively has 8 cycles in the horizontal direction.



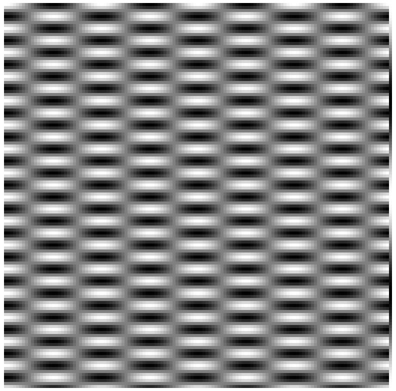
You will notice that the second example is a little more smeared out. This is because the lines are more blurred so more sine waves are required to build it. The transform is weighted so brighter spots indicate sine waves more frequently used.

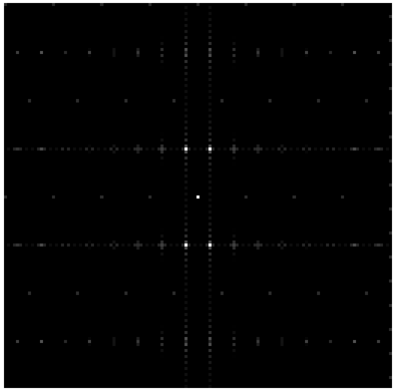
So what is going on here?

The u axis runs from left to right and it represents the horizontal component of the frequency. The v axis runs up and down and it corresponds to vertical components of the frequency.

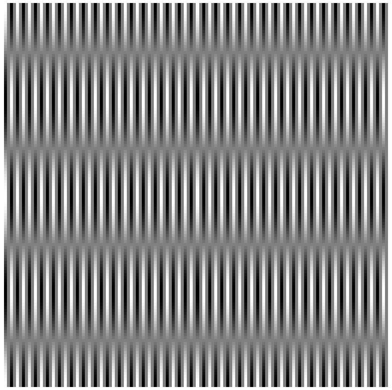


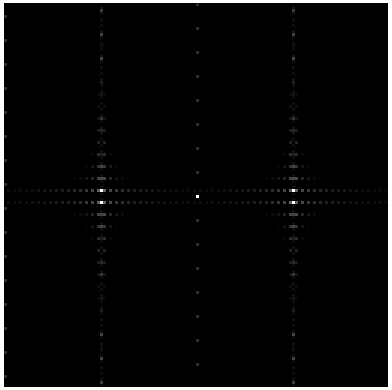
This image exclusively has 4 cycles horizontally and 16 cycles vertically





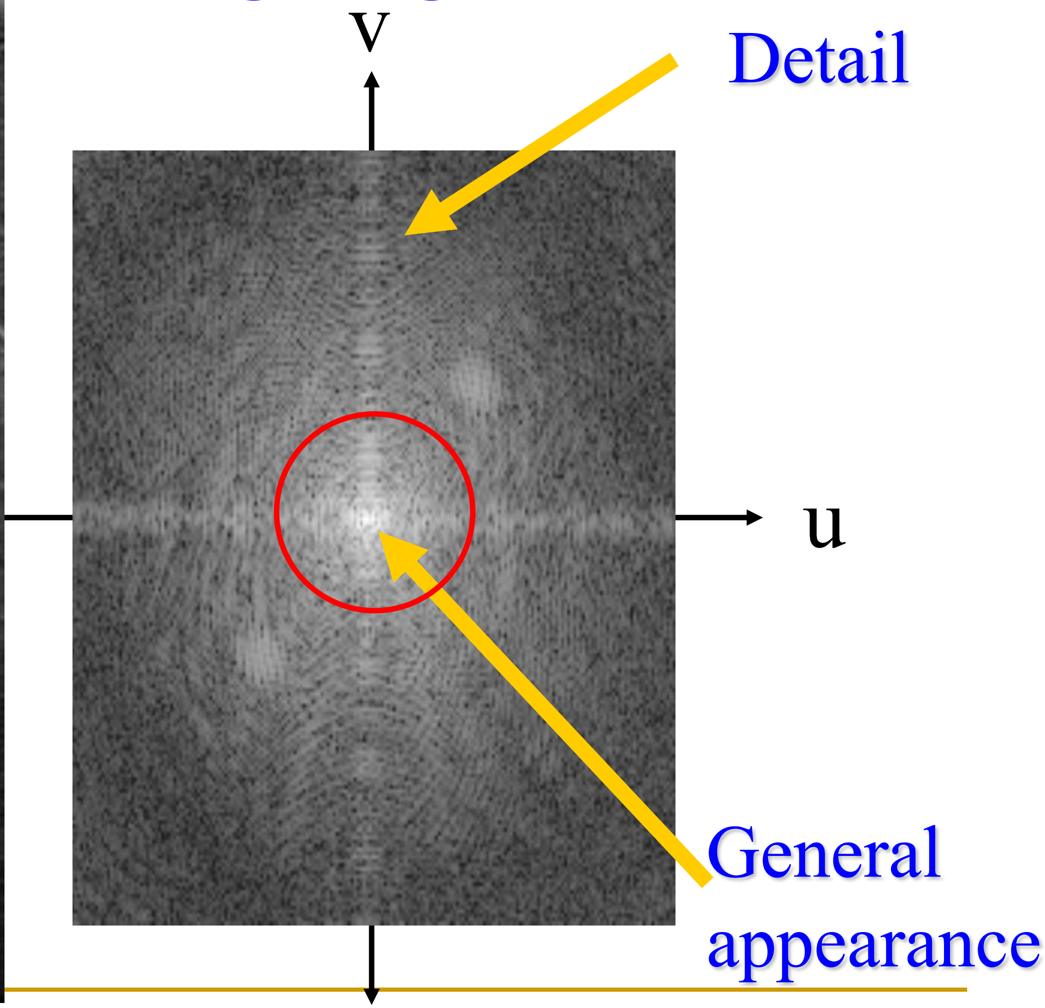
This image exclusively has 32 cycles horizontally and 2 cycles vertically

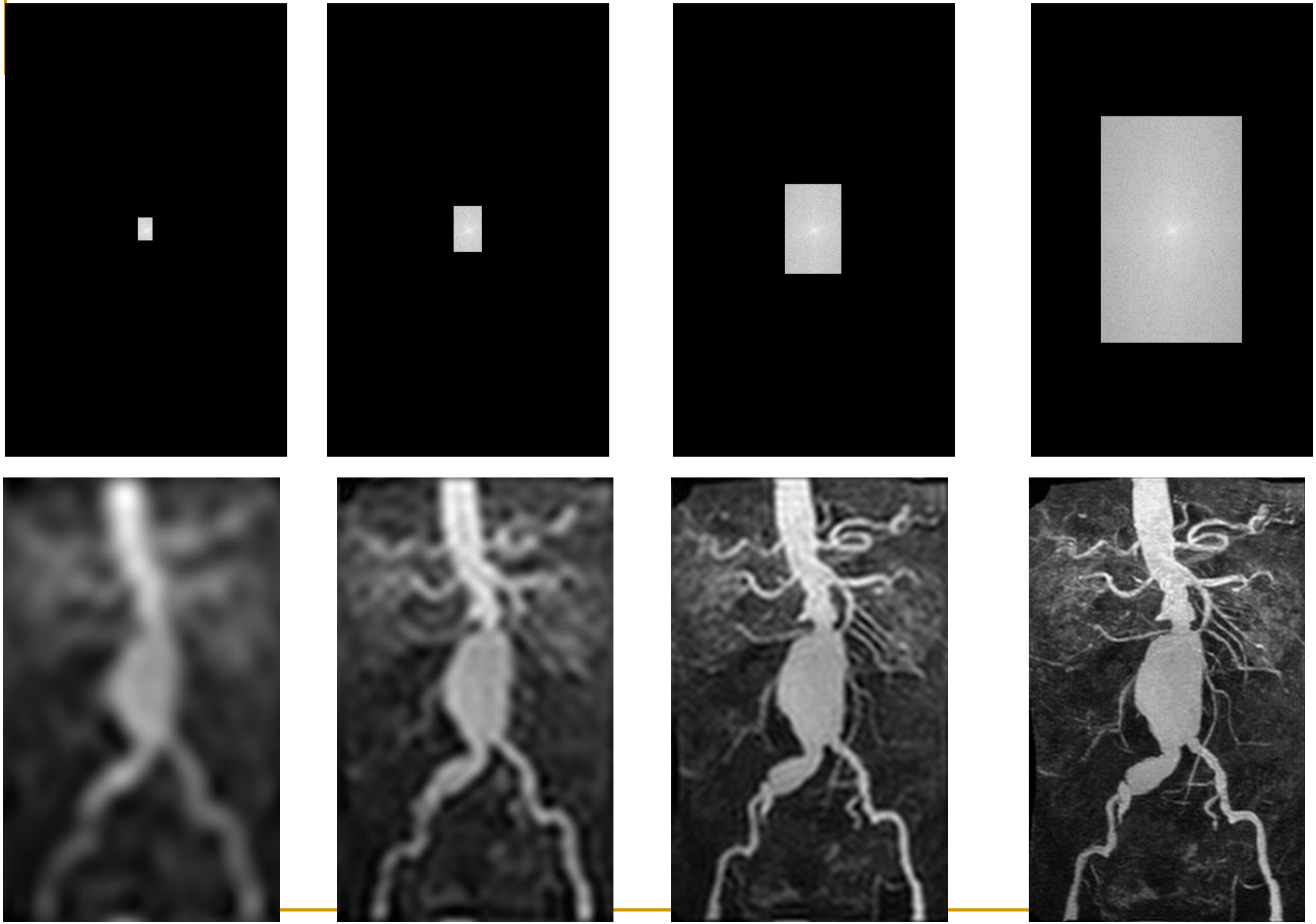




Image

Frequency Domain (log magnitude)





Frequency Domain Filtering

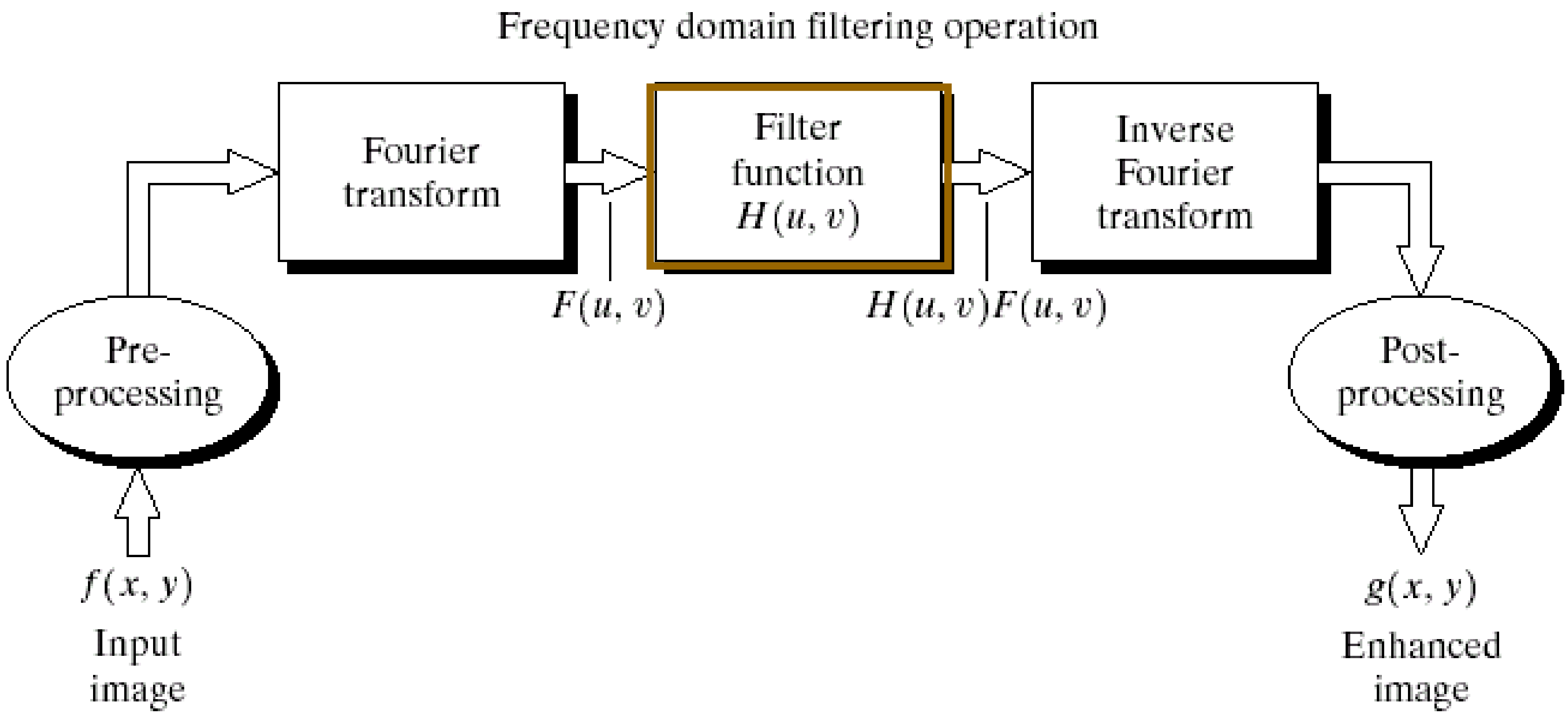


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Frequency Domain Filtering

- Edges and sharp transitions (e.g., noise) in an image contribute significantly to high-frequency content of FT.
- Low frequency contents in the FT are responsible to the general appearance of the image over smooth areas.
- Blurring (smoothing) is achieved by attenuating range of high frequency components of FT.

Convolution in Time Domain

$$g(x,y)=h(x,y)\otimes f(x,y)$$

$$g(x,y)=\sum_{x'=0}^{M-1}\sum_{y'=0}^{M-1}h(x',y')f(x-x',y-y')$$

$$\equiv f(x,y)*h(x,y)$$

- f(x,y) is the input image
- g(x,y) is the filtered
- h(x,y): impulse response

Convolution Theorem

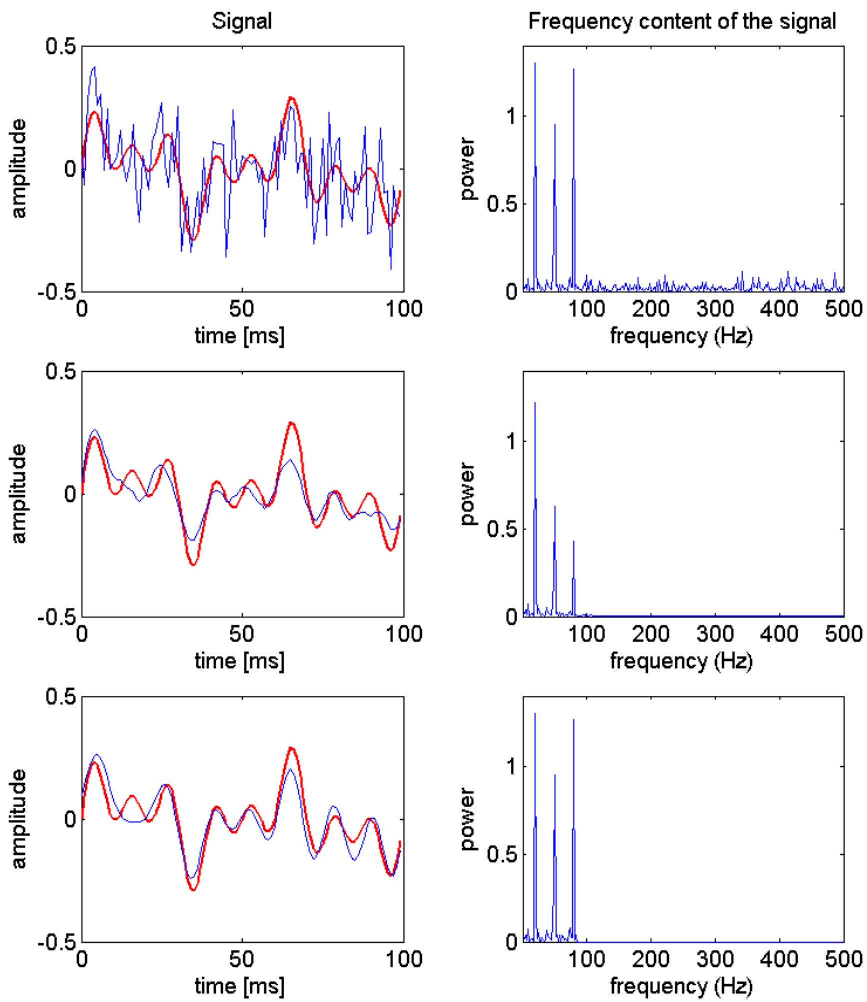
$$G(u,v)=F(u,v)\cdot H(u,v)$$

$$g(x,y)=h(x,y)\otimes f(x,y)$$

Multiplication in Frequency Domain

Convolution in Time Domain

- Filtering in Frequency Domain with H(u,v) is equivalent to filtering in Spatial Domain with h(x,y).



blue line = sum of 3 sinusoids (20, 50, and 80 Hz) + random noise

red line = sum of 3 sinusoids **without** noise

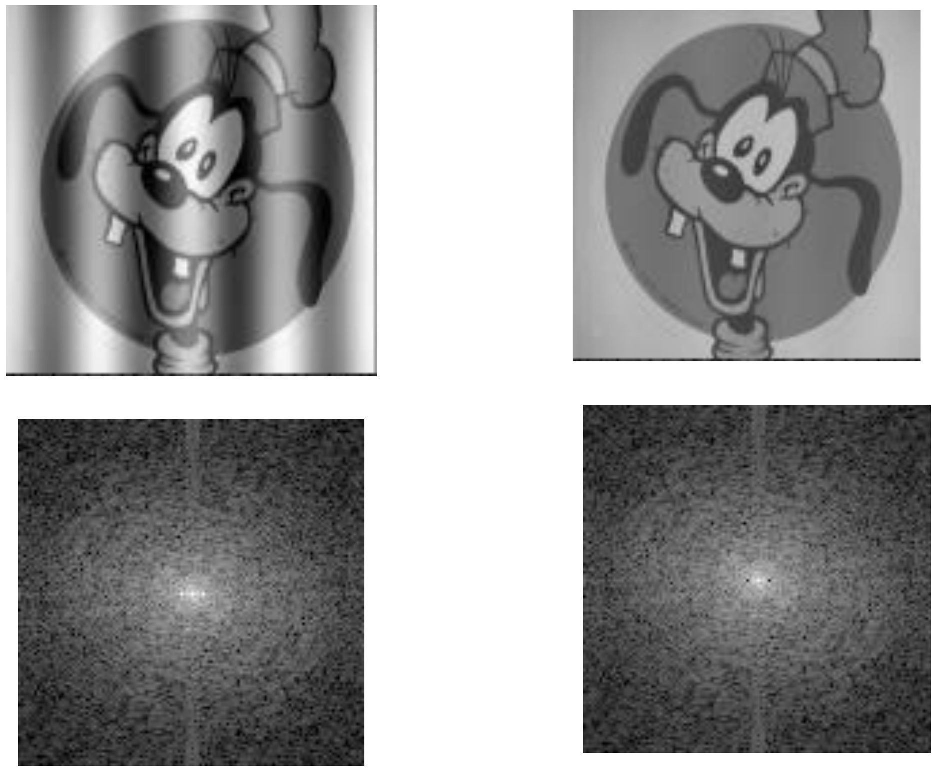
blue line = sum of 3 sinusoids after filtering **in time domain**

1x average [1 1 1 1 1] / 5

blue line = sum of 3 sinusoids after filtering **in frequency domain**

cut-off 90 Hz

A Practical Application



This can be used to eliminate noise without doing an all purpose High Pass Filter that can eliminate detail of the objects being studied!

Fast Fourier Transform

- Separability of Fourier Transform

$$
\begin{aligned}
F(u,v) &= \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp\left[-i2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right] \\
&= \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp\left[-i2\pi\left(\frac{ux}{M}\right)\right]\exp\left[-i2\pi\left(\frac{vy}{N}\right)\right] \\
&= \frac{1}{M}\sum_{x=0}^{M-1}\exp\left[-i2\pi\left(\frac{ux}{M}\right)\right]\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)\exp\left[-i2\pi\left(\frac{vy}{N}\right)\right] \\
&= \frac{1}{M}\sum_{x=0}^{M-1} F(x,v)\exp\left[-i2\pi\left(\frac{ux}{M}\right)\right]
\end{aligned}
$$

$$
F(x,v) = \frac{1}{N}\sum_{y=0}^{N-1} f(x,y)\exp\left[-i2\pi\left(\frac{vy}{N}\right)\right]
$$

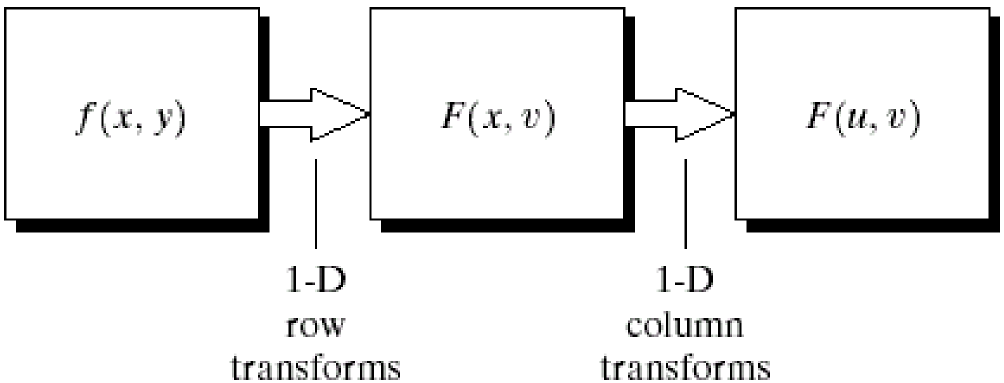


FIGURE 4.35 Computation of the 2-D Fourier transform as a series of 1-D transforms.

Algorithm Complexity

- We can compute the DFT directly using the formula
  - An N point DFT would require N^2 floating point multiplications per output point
  - Since there are N^2 output points , the computational complexity of the DFT is N^4
  - $N^4=4x10^9$ for N=256
  - Bad news! Many hours on a workstation

- The FFT algorithm was developed in the 60's for seismic (地震的) exploration
- Reduced the DFT complexity to $2N^2\log_2 N$
  - $2N^2\log_2 N \sim 10^6$ for N=256
  - A few seconds on a workstation