2. Image Transforms

Transform theory plays a key role in image processing and will be applied during image enhancement, restoration *etc.* as described later in the course.

Many image processing algorithms are applied in the frequency domain rather than the spatial domain and transformation between the two domains can often permit more useful visualisation of the image content. Transform theory is simply the transformation from one domain to another ie. spatial to frequency domain.

1. Revision - Fourier Transform

We shall consider the Fourier transformation of images which are two dimensional discrete signals as discussed previously. However, as revision we will first consider the fourier transform of 1 dimensional continous signals, then consider 1 dimensional discrete signals and their transforms before considering the transformation of 2D discrete signals (images).

1.1 Introduction to Fourier Transform - Continuous

Consider initially the signal $f(x)$ which is a continuous function of the real variable x. The variable x may represent time or distance, as in the distance in one dimension across an image. $f(x)$ is therefore a time or spatial domain function. This function can be converted to the frequency domain using the Fourier Transform.

$$F(u) = \int_{-\infty}^{\infty} f(x) \exp[-2j\pi ux] dx \qquad (1)$$

where $j = \sqrt{-1}$.

Given $F(u)$ the frequency space function $f(x)$ can be obtained using the inverse Fourier transform.

$$f(x) = \int_{-\infty}^{\infty} F(u) \exp[2j\pi ux] du \qquad (2)$$

Equations 1 and 2 are called the Fourier Transform Pair and exist if $f(x)$ is continuous and $F(u)$ is integrable.

Within image processing we are normally concerned with functions $f(x)$ which are real. The Fourier Transform of a real function is generally complex, *ie*

$$F(u) = R(u) + jI(u)$$

where $R(u)$ and $I(u)$ are respectively the real and imaginary components of $F(u)$. This is often expressed as

$$F(u) = |F(u)| e^{j\phi(u)}$$

where $|F(u)|$ is the magnitude function

$$|F(u)| = \sqrt{R(u)^2 + I(u)^2}$$

This is sometimes called the Fourier Spectrum of $f(x)$, and $\phi(u)$ is the phase angle

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

The square of the spectrum $|F(u)|^2$ is often called the Power Spectrum of $f(x)$ or the spectral density.

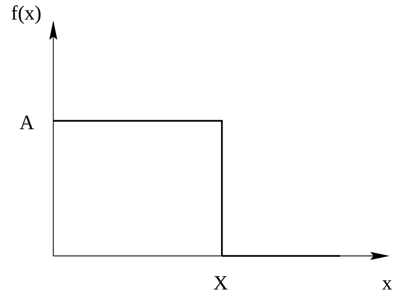
The variable u is often called the frequency variable and this arises from the use of Euler's Formula

$$\exp[-j2\pi ux] = \cos(2\pi ux) - j\sin(2\pi ux)$$

1.1.1 Revision Example

Calculate the Fourier transform of the following signal:-

$$f(x) = \begin{cases} A & \text{if } 0 < x < X \\ 0 & \text{if } x > X \text{ or } x < 0 \end{cases}$$



1.2 Discrete Fourier Transform

1.2.1 Discrete Signals

However, with image processing applications we are normally working with a signal that has been digitised to provide a signal composed of discrete samples which are evenly spaced.

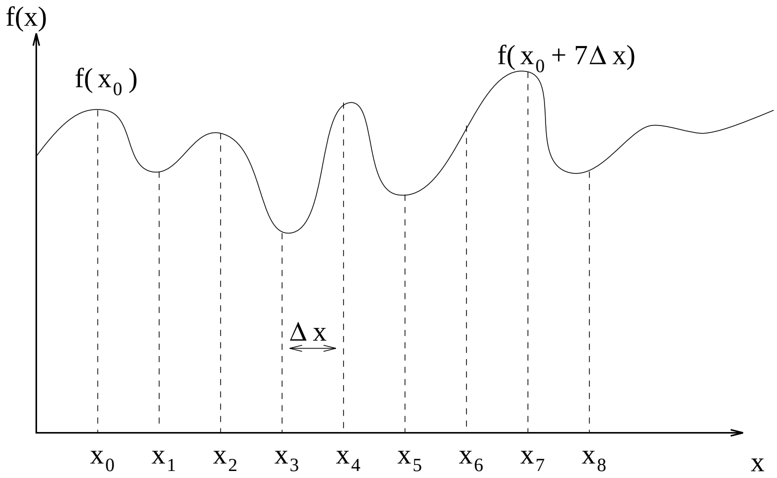


Figure 1: Sampling a continuous function

x now assumes discrete values 0,1,2,.....N-1, and $f(x)$ is now defined as

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

In other words, the sequence {f(0), f(1), f(2),...... f(N-1)} denotes *any* N uniformly spaced samples from a continuous sequence. For any non integer values of x the sequence is undefined.

Two basic simple forms of sample sequences are the unit-sample (or impulse) sequence

$$\delta(x-k) = \begin{cases} 0 & \text{if } x \neq k \\ 1 & \text{if } x = k \end{cases}$$

which consists of a unit sample at location x = k. This is displayed in figure 2 for $\delta(x-4)$. The unit step sequence is denoted as

$$u(x-k) = \begin{cases} 0 & \text{if } x < k \\ 1 & \text{if } x \geq k \end{cases}$$

for a unit step occurring at location x=k. This is displayed in figure 3 for a step occurring at x=3.

In the present discrete case, the unit sample is given by the difference of two unit step sequences one sample apart.

$$\delta(x-k) = u(x-k) - u(x-k-1)$$

Using the above notation any sequence of samples can be expressed as samples of arbitrary amplitude a_k as

$$f(x) = \cdots a_{-2}\delta(x+2) + a_{-1}\delta(x+1) + a_0\delta(x) + a_1\delta(x-1) + \ldots\ldots$$

An example of this is shown in figure 4. This can also be expressed more generally as

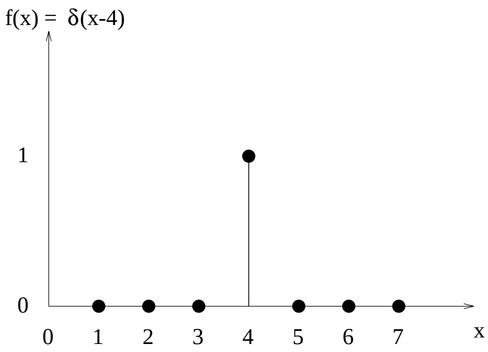


Figure 2: Unit sample sequence

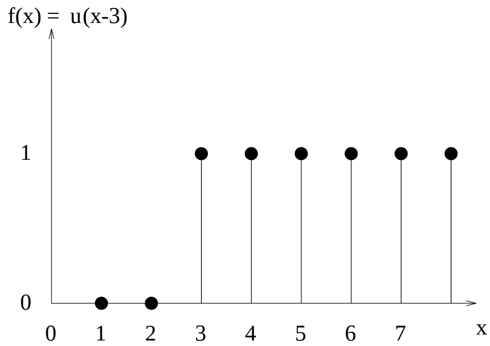


Figure 3: Unit step sequence

$$f(x) = \sum_{k=-\infty}^{\infty} x(k)\delta(x-k)$$

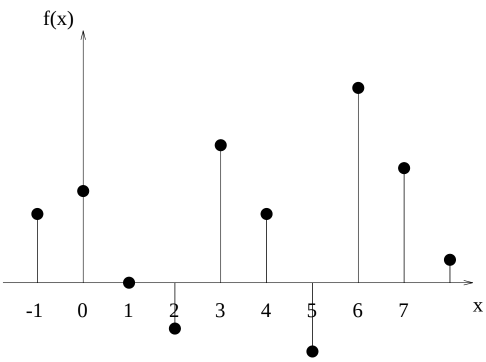


Figure 4: Discrete signal as the sum of weighted unit samples

1.2.2 Discrete 1D Fourier Transform

The *discrete* Fourier Transform is applied in this case, where

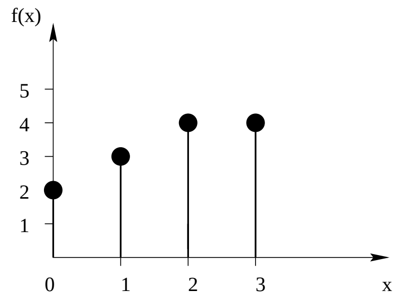
$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\exp\left[-j2\pi\frac{ux}{N}\right]$$

for u = 0, 1, 2, .. N-1 where f(x) contains N uniformly spaced samples. The inverse discrete Fourier transform is given as

$$f(x) = \sum_{u=0}^{N-1} F(u)\exp\left[j2\pi\frac{ux}{N}\right]$$

1.2.3 Example

Calculate the discrete Fourier transform of the signal shown in the figure below.



2. Two Dimensional Signals

Previously we have briefly reviewed one dimensional discrete signal representation. This analysis will now be extended to signals in more than one dimension. We will consider initially the discrete signal $f(x, y)$ where x and y are integers and for non integer values of x and y, f(x,y) is not defined. This can be considered to represent the digital image, where (x,y) gives the pixel location and f(x,y) is the value of this pixel.

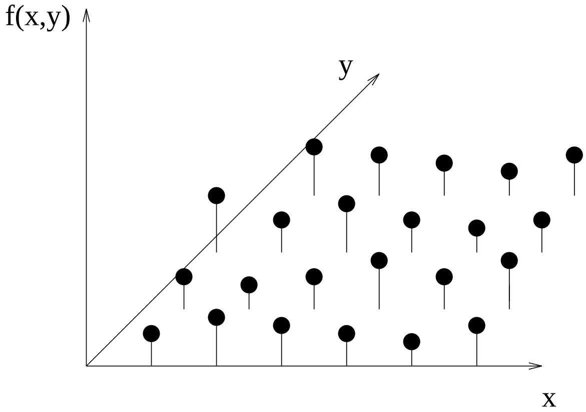


Figure 5: 2D sequence f(x,y)

Figure 5 shows a two dimensional signal which can be considered as the sum of weighted samples in a similar manner to the 1D case.

The unit sample can now be written as

$$\delta(x - k_1, y - k_2) = \begin{cases} 1 & \text{if } x = k_1 \text{ and } y = k_2 \\ 0 & \textit{otherwise} \end{cases}$$

and the unit step as

$$u(x - k_1, y - k_2) = \begin{cases} 1 & \text{if } x \geq k_1 \text{ and } y \geq k_2 \\ 0 & \textit{otherwise} \end{cases}$$

2.1 Separable Sequences

The analysis of 2D signals can often be achieved quite easily if the signal is separable. A 2D sequence is said to be separable if it can be expressed as the product of one dimensional terms, *ie.* a two dimensional sequence f(x,y) is separable if it can be expressed as

$$f(x, y) = f_1(x) f_2(y)$$

where $f_1(x)$ is a function of x only, and $f_2(y)$ is a function of y only. For example the unit sample $\delta(x - k_1, y - k_2)$ is separable since

$$\delta(x - k_1, y - k_2) = \delta(x - k_1)\delta(y - k_2)$$

An example of a sequence which is not separable is

$$f(x, y) = \cos(\omega xy)$$

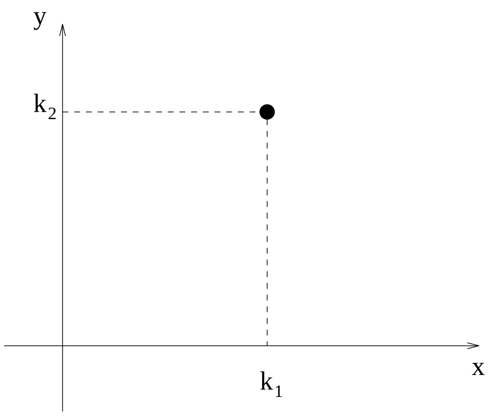


Figure 6: 2D unit sample

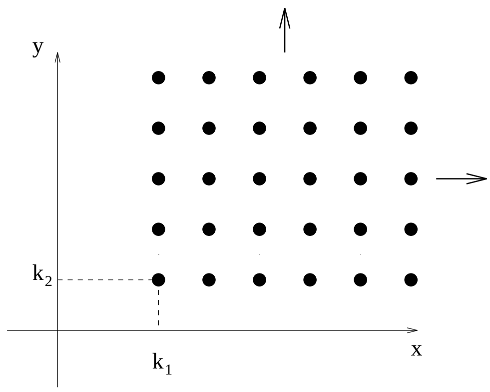


Figure 7: 2D unit step sequence

The separability of sequences can be exploited in order to reduce computation in various contexts, such as in the computation of the discrete Fourier Transform.

2.2 Periodic Sequences

A sequence $f(x, y)$ is said to be periodic with a period $N_1 \times N_2$ if $f(x, y)$ satisfies

$$f(x, y) = f(x + N_1, y) = f(x, N_2 + y) \text{ for } \textit{all } (x, y)$$

where N_1 and N_2 are positive integers.

3. Two Dimensional Discrete Fourier Transform

The two variable discrete Fourier transform can be written

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp\left[-j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \tag{3}$$

for u = 0, 1, 2, ...M-1 and v = 0, 1, 2, ...N-1.

The inverse discrete Fourier transform is

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp\left[j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \tag{4}$$

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

More generally we tend to work with images which are square (ease of use of Fast Fourier Transforms which will be discussed in section 6). If the 2D sequence (image) is sampled as a square array where $N = M$ then the equations become:-

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp\left[-j2\pi \left(\frac{ux + vy}{N} \right) \right] \tag{5}$$

for $u, v = 0, 1, 2, \ldots\ldots, N - 1$ and

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \exp\left[j2\pi \left(\frac{ux + vy}{N} \right) \right] \tag{6}$$

for $x, y = 0, 1, 2, \ldots\ldots, N - 1$

Note the inclusion of the $\frac{1}{N}$ term in both of these equations. In practice, images are typically digitised in square arrays and so we will be most concerned with the Fourier transform pairs of equations 5 and 6. The formulations of equations 3 and 4 are important when stressing the generality of the image size.

3.1 Display of 2D Fourier Transform

The dynamic range of the Fourier spectra is usually much higher than the typical display device, and only the brightest parts of the image are visible. A log normal scaling function, as given by equation 7, is usually applied.

$$D(u, v) = c \log[1 + |F(u, v)|] \tag{7}$$

and D(u,v) is then displayed instead of $|F(u, v)|$. c is a scaling function and the logarithm function performs the desired compression.

For example, figure 8 illustrates the display of the discrete Fourier transform and the scaled transform for the image given. The increase in visible detail is obvious.

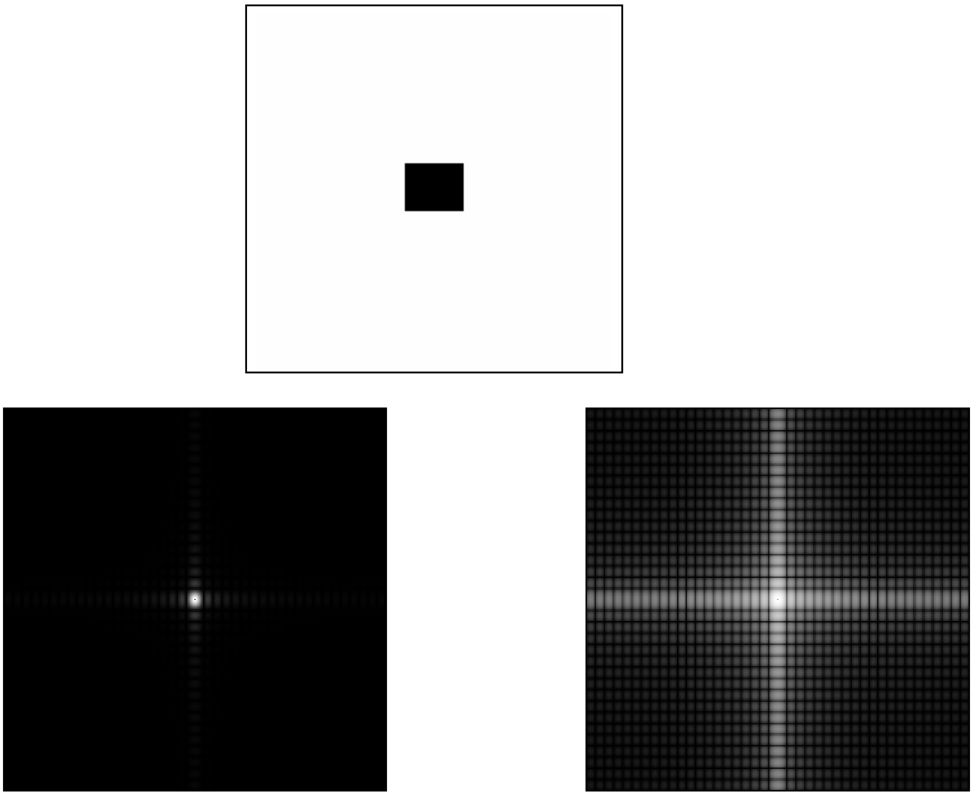


Figure 8: (a) A simple image (b) display of |F(u,v)| (c) display of D(u,v)

The Fourier spectra in this case has been displayed as an intensity function. Figure 9 shows the Fourier spectra which has been log-normalised and plotted directly in figure 9(b) and figure 9(c) then shows this plot displayed as an intensity function.

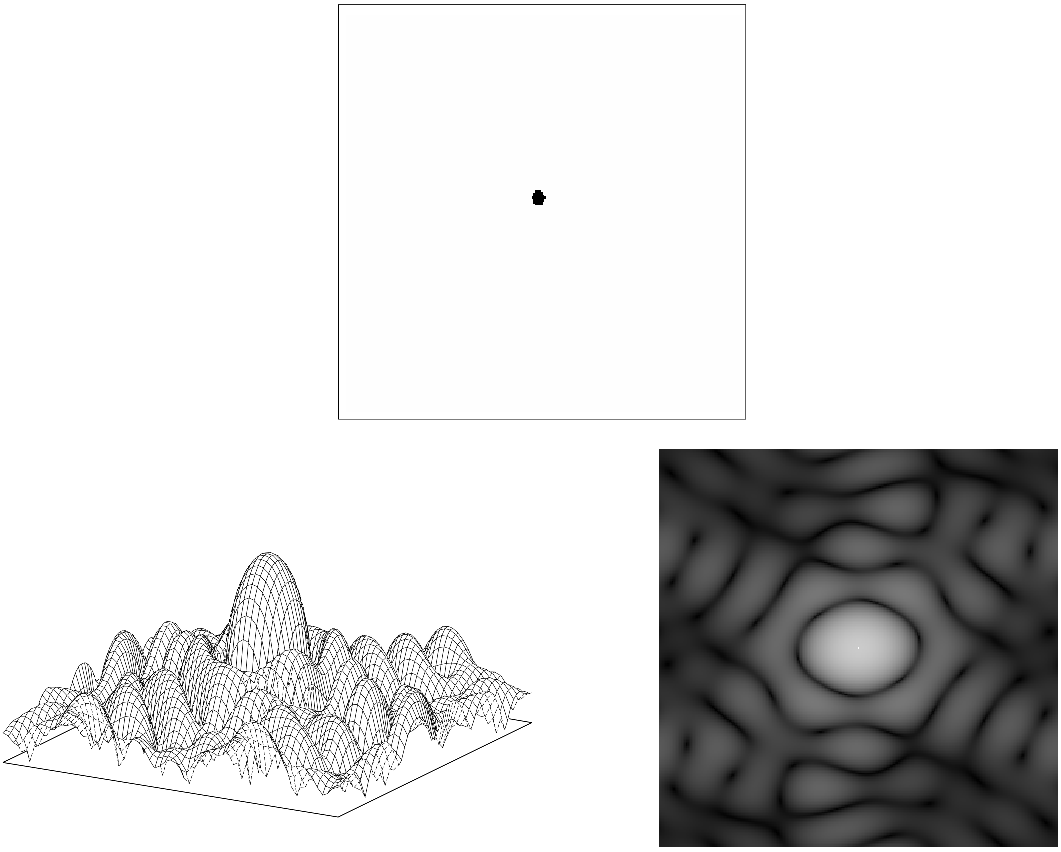


Figure 9: (a) image (b) plot of Fourier spectra (b) Spectra as intensity image

3.2 Examples of Fourier Transforms of Images

Figure 10 illustrates an input image which consists of ripples of approximately a sinusoidal shape and the magnitude response of this image. The magnitude response has a clear peak at the frequency of the ripples. As the ripples run diagonally across the image - the position of the peak of the magnitude response is determined by this directionality.

Figure 11 illustrates a similar input image - but in this case the ripples are larger and hence fewer occur in the image. The magnitude response illustrates this lower frequency - the peak is closer to the centre of the image. The directionality of the ripples has also changed.

Figure 12 shows an image with higher frequency ripples and the resulting magnitude response.

It should be noted that in all cases the Fourier transform is repeated as it is calculated in the angular range from 0 to 180 ° (0 to π radians). Hence the symmetry in the magnitude spectra which illustrates the result for 0 to 360 °.

Figures 13 and 14 display other examples of images and their Fourier Spectra.

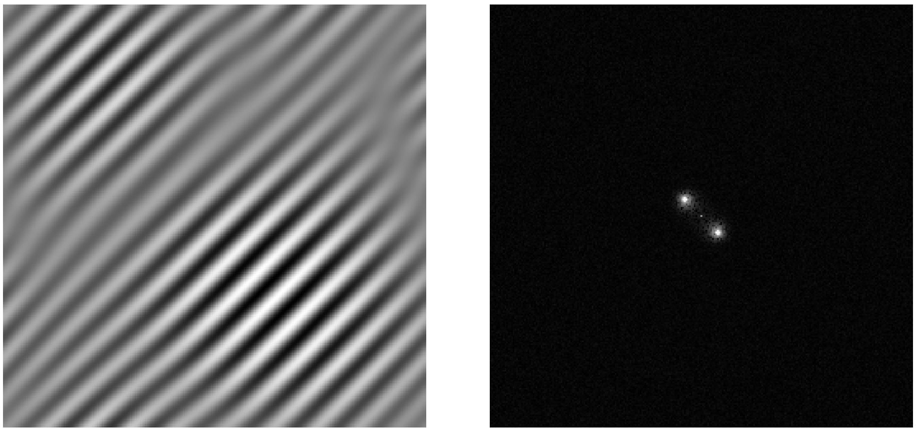


Figure 10: (a) Input Image (b) Magnitude Response from Fourier Transform

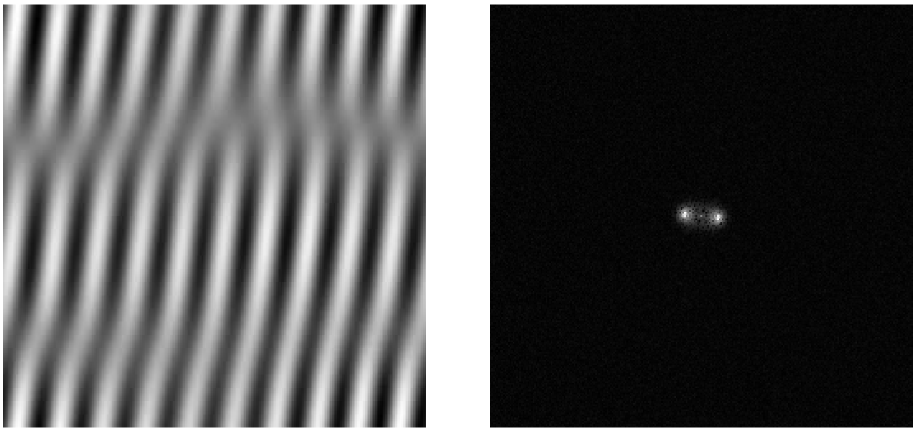


Figure 11: (a) Input Image (b) Magnitude Response from Fourier Transform

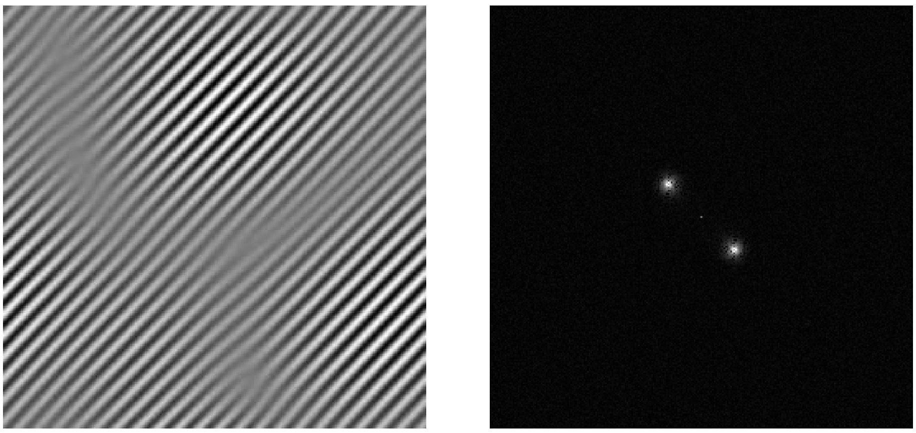


Figure 12: (a) Input Image (b) Magnitude Response from Fourier Transform

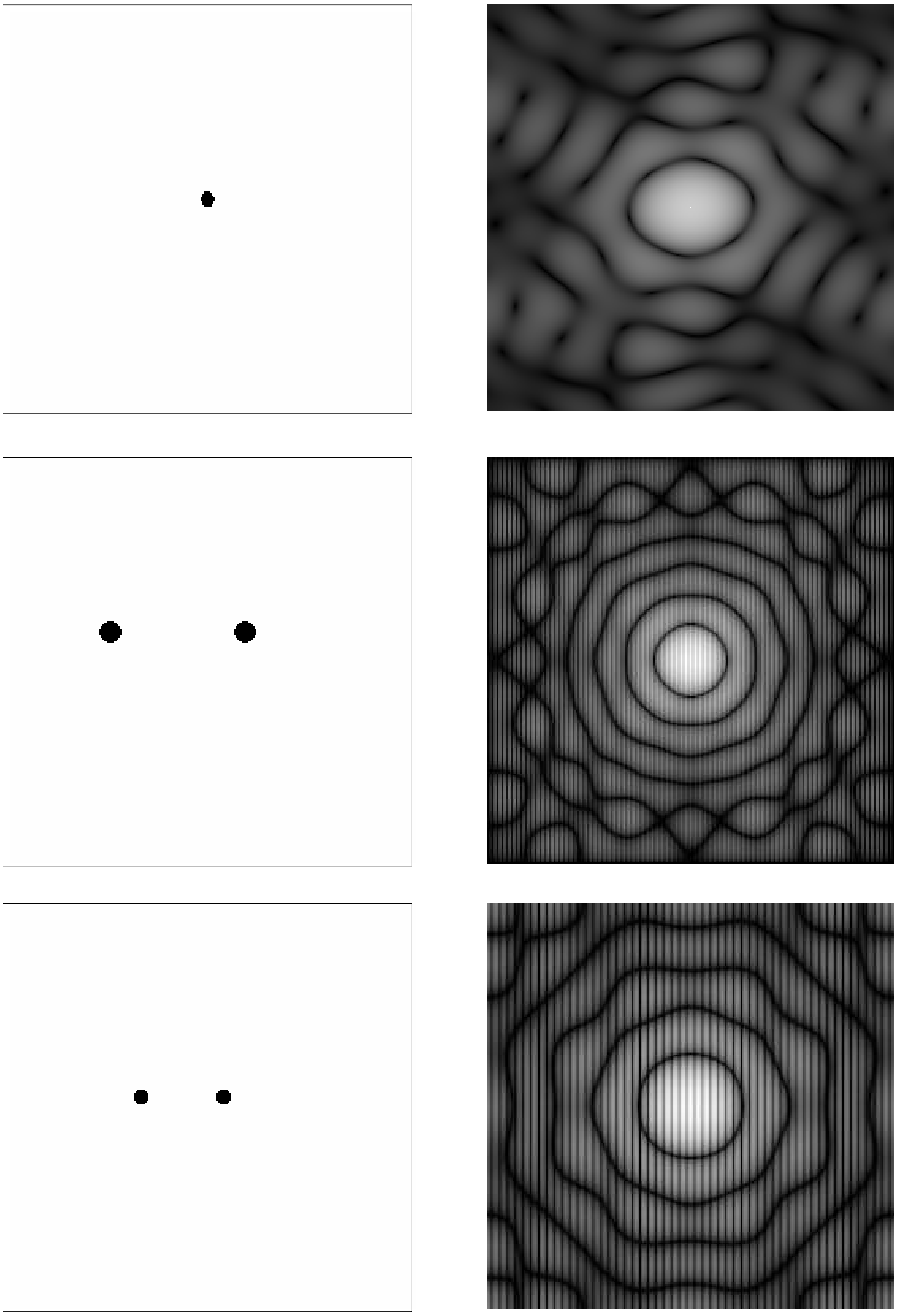


Figure 13: Example Fourier Transforms of Images

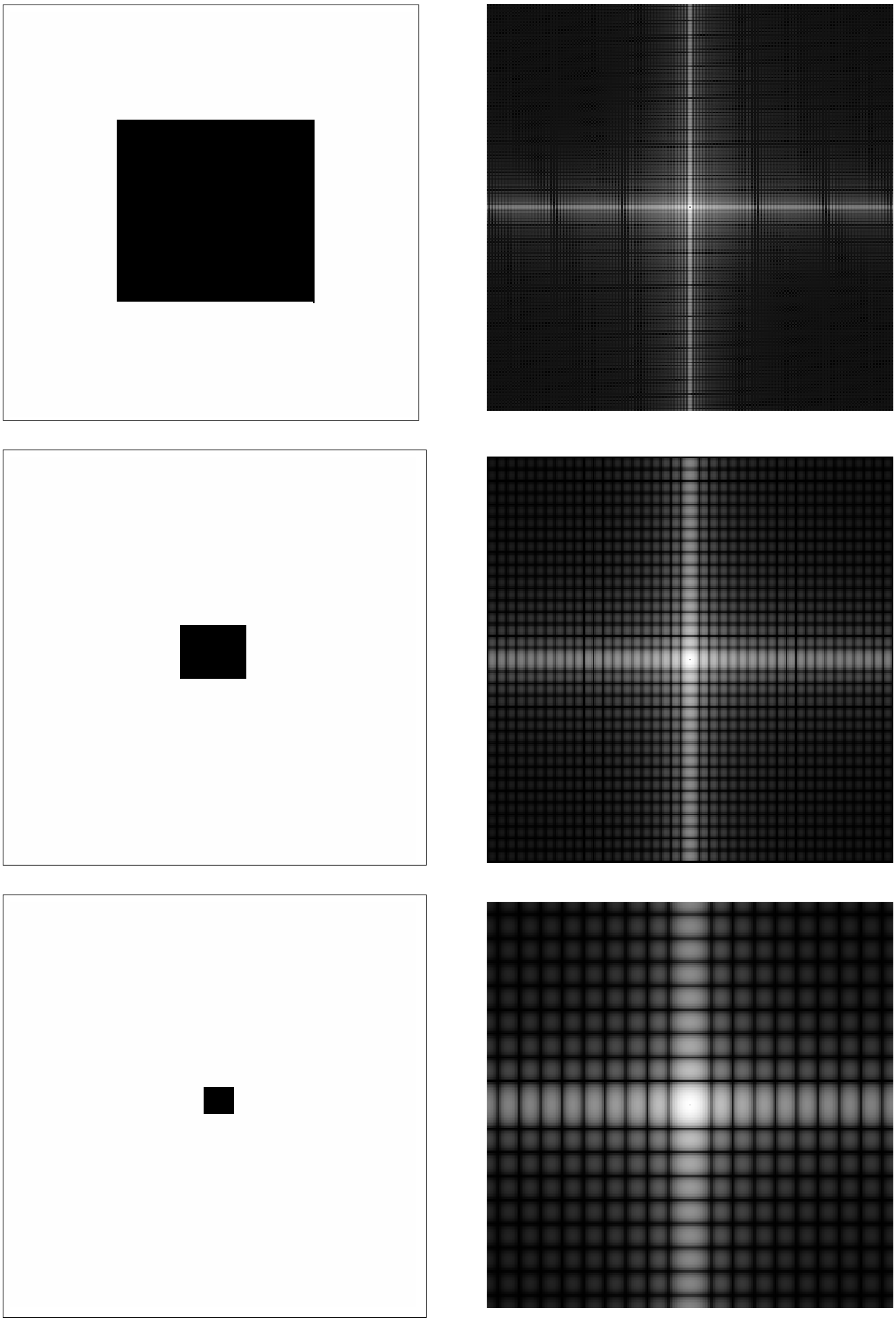


Figure 14: Example Fourier Transforms of Images

4. Properties of Two Dimensional Fourier Transform

4.1 Separability

The property of separable functions can be applied to simplify the calculation of the 2D discrete Fourier Transform, since the exponential function is separable. The discrete Fourier transform can be written as

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\exp\left[-j2\pi\frac{ux}{N}\right]\sum_{y=0}^{N-1} f(x,y)\exp\left[-j2\pi\frac{vy}{N}\right] \tag{8}$$

for u,v = 0, 1, 2, ..., N-1.

And the inverse can then be expressed as

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\exp\left[j2\pi\frac{ux}{N}\right]\sum_{v=0}^{N-1} F(u,v)\exp\left[j2\pi\frac{vy}{N}\right] \tag{9}$$

The principle advantage of the separability property is that $F(u,v)$ or $f(x,y)$ can be obtained in two steps by successive applications of the 1D Fourier transform or its inverse. This can be seen clearly if equation 8 is expressed in the form

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} F(x,v)\exp\left[-j2\pi\frac{ux}{N}\right] \tag{10}$$

where

$$F(x,v) = N\left[\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)\exp\left[-j2\pi\frac{vy}{N}\right]\right] \tag{11}$$

For each value of x the expression inside the brackets of equation 11 is a 1D transform with frequency values $v = 0, 1, 2, \cdots N-1$. Therefore, the 2D function $F(x,v)$ is obtained by taking a transform along each column of $f(x,y)$ and multiplying the result by N. The desired result $F(u,v)$ is then obtained by taking a transform along each row of $F(x,v)$ as shown below.

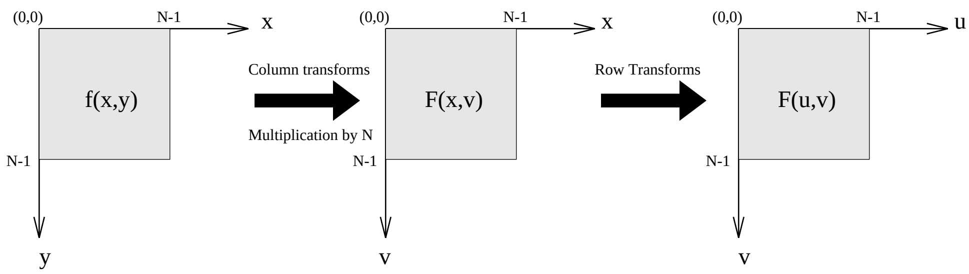


Figure 15: Calculation of 2D Fourier Transform as a series of 1D transforms

The inverse Fourier transform can be separated in a similar manner.

4.2 Translation

The translation properties of the Fourier transform pair are

$$f(x,y)\exp\left[j2\pi\frac{(u_0x+v_0y)}{N}\right] \Leftrightarrow F(u-u_0, v-v_0)$$

$$f(x - x_0, y - y_0) <=> F(u, v) \exp\left[-j2\pi(ux_0 + vy_0)/N\right]$$

where <=> indicates the correspondence between a function and its Fourier transform and vice-versa.

Using this property the origin of the Fourier transform can be moved to the centre of its $N \times N$ frequency square simply by multiplying $f(x, y)$ by $(-1)^{x+y}$.

$$f(x, y)(-1)^{x+y} <=> F(u - N/2, v - N/2) \tag{12}$$

since

$$\exp[j2\pi(u_0 x + v_0 y)/N] = e^{j\pi(x+y)} = (-1)^{x+y}$$

NOTE: The shift does not affect the magnitude of the Fourier transform.

4.2.1 Proof:

It can be shown simply that the above translation property can be used to move the origin of the Fourier transform to the centre of the frequency square.

4.3 Periodicity

The discrete Fourier transform and its inverse are *periodic* with period N, *ie*

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

Although $F(u,v)$ repeats itself for infinitely many values of u and v, only the N values of each variable in any one period are required to obtain $f(x,y)$ from $F(u,v)$. Therefore, only one period of the transform is necessary to specify $F(u,v)$ completely in the frequency domain.

If $f(x,y)$ is real, the Fourier Transform also exhibits conjugate symmetry:

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

or

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

where $F^*(u,v)$ is the complex conjugate of $F(u,v)$.

To interpret the above properties consider the one variable case displayed in figure 16. The periodicity and symmetry properties show that the magnitude of the transform is centred on the origin. Because the discrete fourier transform has been formulated for values of u in the range 0 to N-1, this produces two back to back half periods in this interval. To display one full period it is necessary to move the origin to $u = N/2$. We can achieve this by multiplying $f(x)$ by $(-1)^x$ prior to taking the transform (from translation properties) or by swapping the two halfs of the transform.

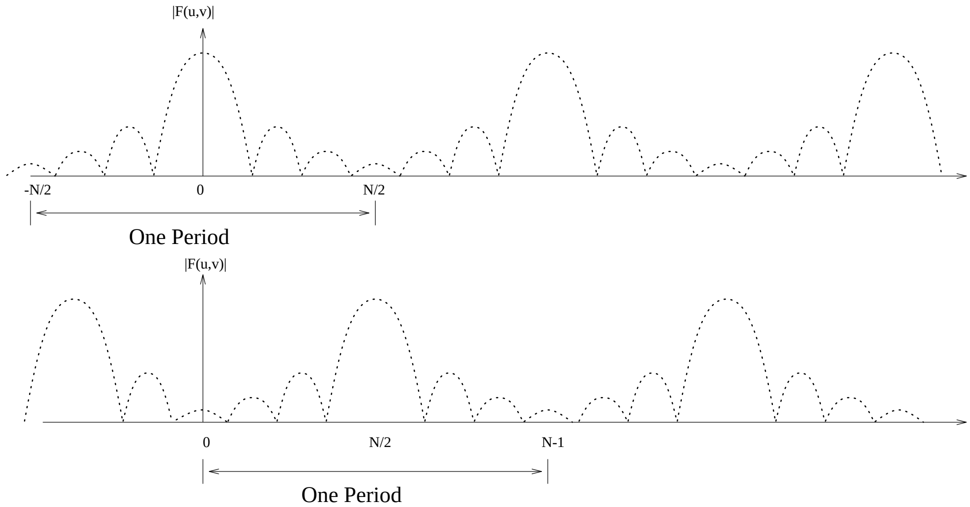


Figure 16: Periodicity properties of Fourier Transform (a)Fourier spectrum (b) shifted spectrum

The same holds for the magnitude of the 2D Fourier transform. The results are easier to interpret if the origin of the transform is shifted to the frequency point $(N/2, N/2)$, as shown in figure 17. This can be achieved by using the centring property of equation 12 or by swapping the four quadrants of the magnitude transform once the transform is calculated, as shown schematically in figure 18.

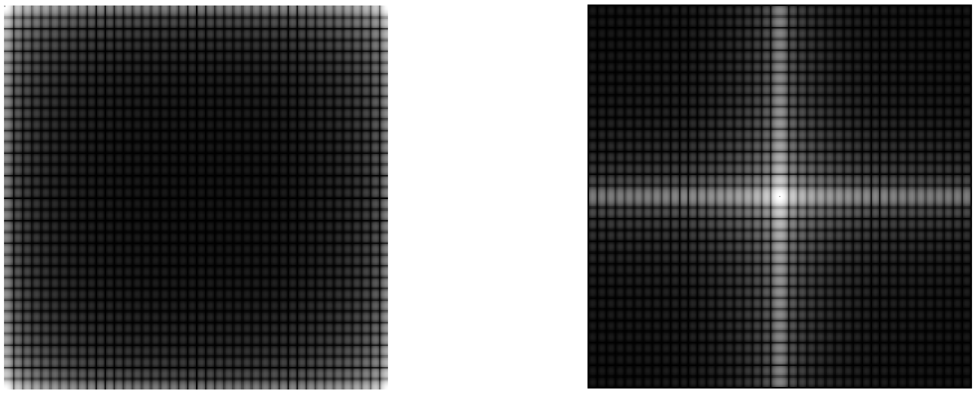


Figure 17: (a) Fourier transform of square without shifting (b) shifted to centre

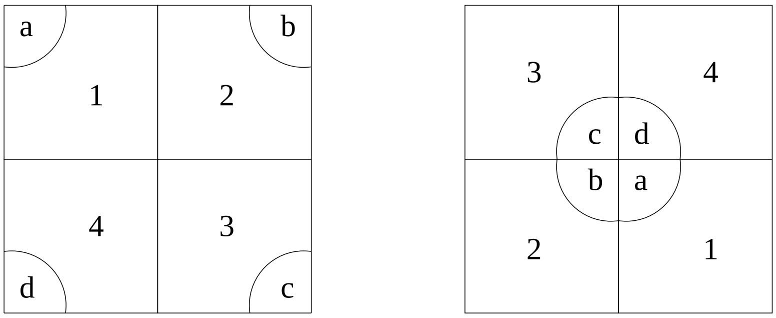


Figure 18: Schematic Representation of 2D Fourier Transforms

4.4 Rotation

This is most easily considered by introducing the polar coordinates

$$x = r\cos\theta \qquad y = r\sin\theta \qquad u = \omega\cos\phi \qquad v = \omega\sin\phi$$

then $f(x, y)$ and $F(u, v)$ become $f(r, \theta)$ and $(F(\omega, \phi)$ respectively. This yields

$$f(r, \theta + \theta_0) <=> F(\omega, \phi + \theta_0)$$

In other words rotating the digital image $f(x, y)$ by an angle θ_0 rotates the Fourier transform $F(u, v)$ by the same angle. This is illustrated in figure 19.

4.5 Distributivity

The Fourier transform is distributive over addition.

$$\Im\left[f_1(x, y) + f_2(x, y)\right] = \Im\left[f_1(x, y)\right] + \Im\left[f_2(x, y)\right]$$

The Fourier transform is NOT distributive over multiplication.

$$\Im\left[f_1(x, y) f_2(x, y)\right] \neq \Im\left[f_1(x, y)\right] \Im\left[f_2(x, y)\right]$$

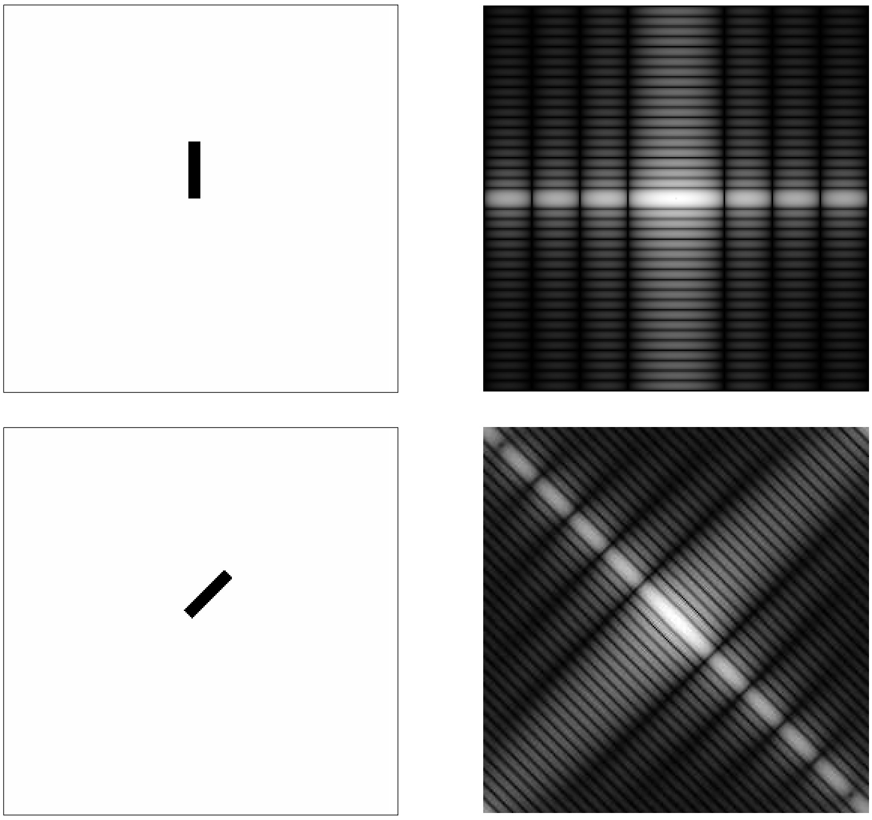


Figure 19: Rotational properties of Fourier Transform (a) simple image (b) spectrum (c) rotated image (d) resulting spectrum

4.6 Scaling

The scaling property of the Fourier transform for scalars *a* and *b* is

$$af(x, y) <=> aF(u, v)$$

$$f(ax, by) <=> \frac{1}{|ab|} F(\frac{u}{a}, \frac{v}{b})$$

This second scaling property is illustrated in figure 14.

5. Convolution and Correlation

You will already have encountered the link between convolution and multiplication in the spatial and frequency domains for 1D signals. This same relationship holds for the discrete 2D case.

5.1 One Dimensional Convolution

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x - \alpha) d\alpha$$

where α is the dummy variable of integration. This is illustrated schematically in figure 20.

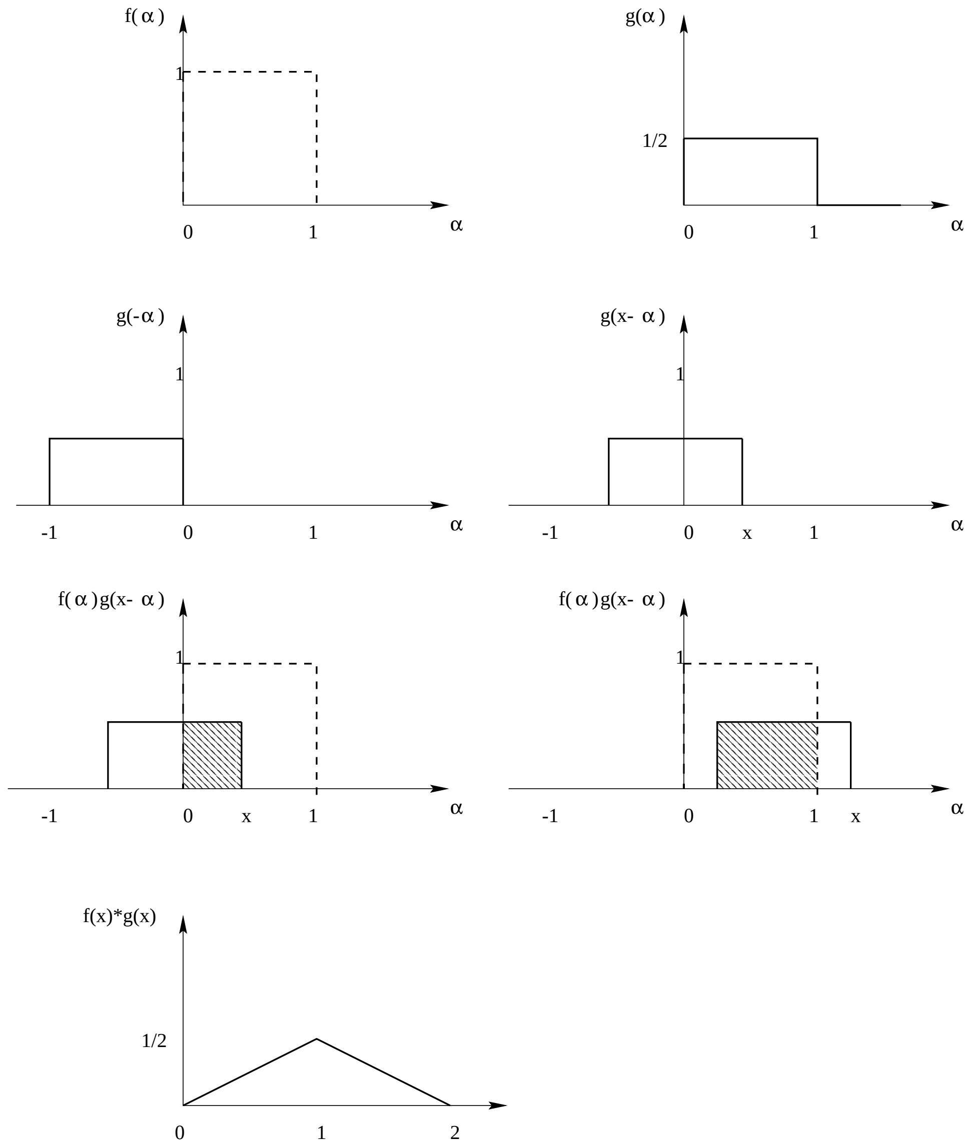


Figure 20: Schematic Illustration of convolution of 1D continuous signals

The importance of convolution in frequency domain analysis is that $f(x) * g(x)$ and $F(u)G(u)$ constitute a Fourier transform pair. In other words if $f(x)$ has a Fourier transform $F(u)$ and $g(x)$ has a Fourier transform $G(u)$ then $f(x) * g(x)$ has the Fourier transform $F(u)G(u)$.

$$f(x) * g(x) <=> F(u)G(u)$$

Therefore convolution in the spatial domain is equivalent to multiplication in the frequency domain and vice-versa

$$f(x)g(x) <=> F(u) * G(u)$$

If $f(x)$ and $g(x)$ are discrete, then the discrete Fourier transform and its inverse are periodic functions. Formulating a discrete convolution theorem to be consistent with periodicity involves assuming that the discrete functions $f(x)$ and $g(x)$ are periodic with the same period M. The resulting convolution is then periodic with the same period. The period M is usually chosen as

$$M \geq A + B - 1$$

where A is the number of samples in $f(x)$ and B is the number of samples in $g(x)$.

The functions $f(x)$ and $g(x)$ are then padded with zeroes to create functions with M samples. The *extended* sequences are then

$$f_e(x) = \begin{cases} f(x) & 0 \leq x \leq A - 1 \\ 0 & A \leq x \leq M - 1 \end{cases}$$

$$g_e(x) = \begin{cases} g(x) & 0 \leq x \leq B - 1 \\ 0 & B \leq x \leq M - 1 \end{cases}$$

The discrete convolution of $f_e(x)$ and $g_e(x)$ is then

$$f_e(x) * g_e(x) = \sum_{m=0}^{M-1} f_e(m) g_e(x - m)$$

for $x = 0, 1, 2, \cdots M - 1$. The convolution function is a discrete periodic array of length M, with the values $x = 0, 1, 2, \ldots.\ M - 1$ describing a full period.

This is illustrated schematically in figure 21.

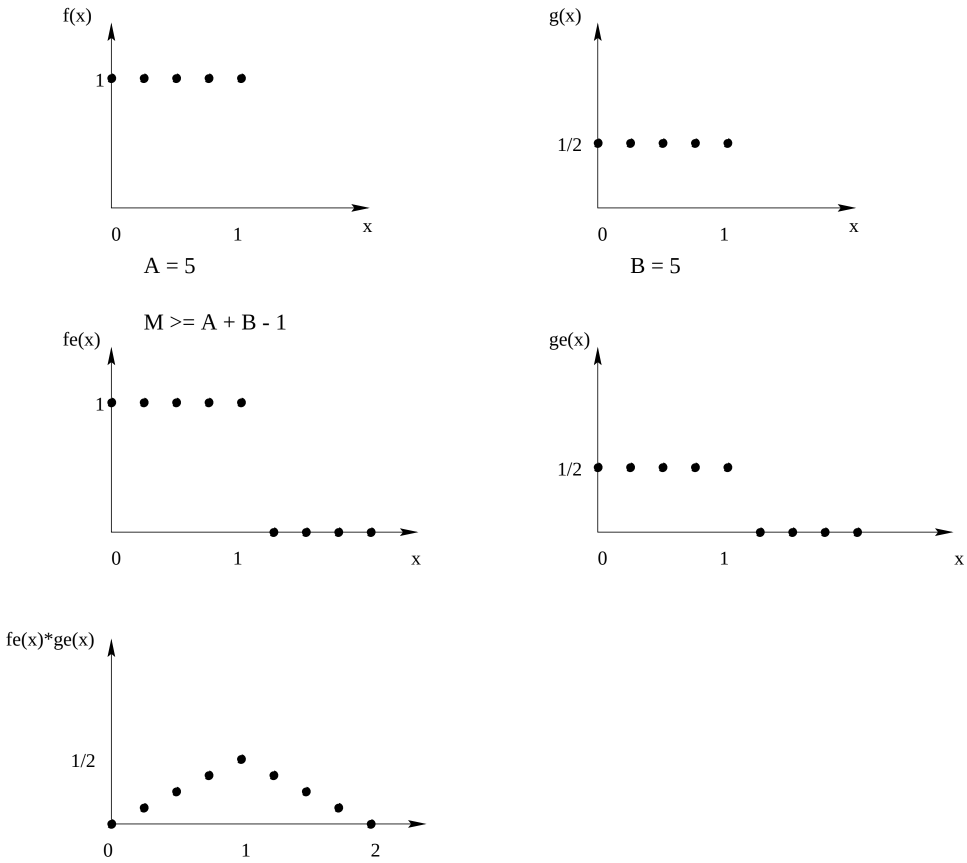


Figure 21: Example of extension of signals required for convolution of discrete signals

5.2 Two Dimensional Convolution - Discrete Case

The convolution theorem again gives

$$f(x, y) * g(x, y) <=> F(u, v)G(u, v)$$

$$f(x, y)g(x, y) <=> F(u, v) * G(u, v)$$

If $f(x, y)$ is a discrete array of size $A \times B$ and $g(x, y)$ is a discrete array of size $c \times D$, then the arrays must be padded so that they are periodic with dimensions $M \times N$, where M and N are chosen to avoid wrap-around error as

$$M \geq A + C - 1$$

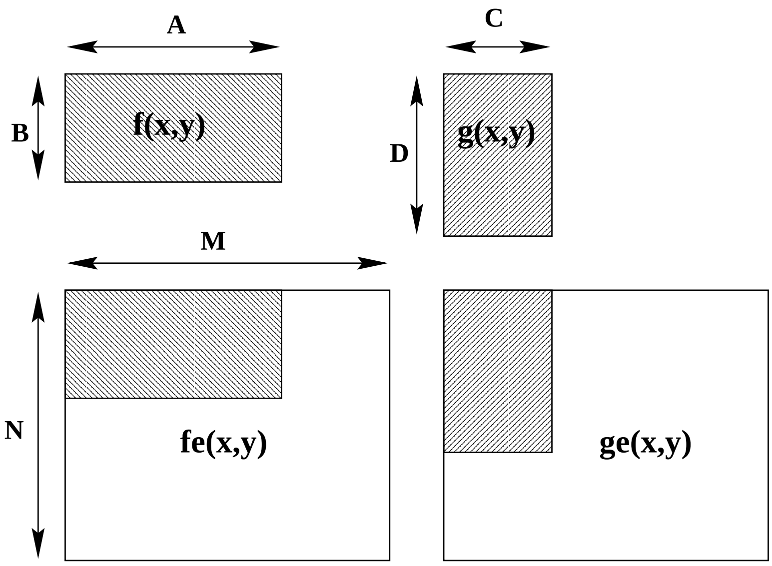
$$N \geq B + D - 1$$

The extended sequences are then formed as

$$f_e(x, y) = \begin{cases} f(x, y) & 0 \leq x \leq A - 1 \ AND \ 0 \leq y \leq B - 1 \\ 0 & A \leq x \leq M - 1 \ OR \ \ B \leq y \leq N - 1 \end{cases}$$

$$g_e(x, y) = \begin{cases} g(x, y) & 0 \leq x \leq C - 1 \ AND \ 0 \leq y \leq D - 1 \\ 0 & C \leq x \leq M - 1 \ OR \ \ D \leq y \leq N - 1 \end{cases}$$

This is illustrated in figure 22



M >= A + C -1
N >= B + D -1

Figure 22: Extension of 2D signals for convolution

The 2D convolution of $f_e(x, y)$ and $g_e(x, y)$ is then defined as

$$f_e(x, y) * g_e(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e(m, n) g_e(x - m, y - n)$$

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

When calculating the convolution by converting to the frequency domain, multiplying and then taking the inverse transform, you must use the extended sequences. Therefore, take the Fourier transforms of $f_e(x, y)$ and $g_e(x, y)$, multiply $F_e(u, v)$ and $G_e(u, v)$ and then take the inverse transform of the result. This method is usually faster than calculating the convolution directly.

5.3 Correlation

The correlation of discrete functions $f(x)$ and $g(x)$ again requires them to be extended as discussed for discrete convolution. If $f(x)$ and $g(x)$ are the same function, equation 13 is usually called the *autocorrelation* function; if $f(x)$ and $g(x)$ are different, the term *cross correlation* is normally used.

$$f_e(x) \circ g_e(x) = \sum_{m=0}^{M-1} f^*_e(m) g_e(x + m) \tag{13}$$

for x = 0, 1, 2, ... M-1 and * denotes the complex conjugate.

This can also be extended to 2D

$$f_e(x, y) \circ g_e(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*_e(m, n) g_e(x + m, y + n)$$

The following *Correlation Theorem* holds

$$f(x, y) \circ g(x, y) <=> F^*(u, v)G(u, v)$$

$$f^*(x, y)g(x, y) <=> F(u, v) \circ G(u, v)$$

6. The Fast Fourier Transform

The Fast Fourier Transform (FFT) provides a method of speeding up the calculation of the discrete Fourier transform by reducing the number of complex multiplications and additions. For a one dimensional signal N^2 complex multiplications and additions are required to calculate the DFT but only $N \log_2 N$ complex calculations are required to calculate the FFT.

6.1 FFT algorithm

To develop the basic FFT algorithm we will look at computing DFTs for N=2, 4 and 8 and how this can be generalised to the case for any N where N is a power of 2. The notation used is

$$F(u) = \sum_{x=0}^{N-1} f(x) {W_N}^{ux} \quad u = 0, 1, 2, \cdots, N-1$$

where

$$W_N = \exp[-j2\pi/N]$$

ux is the exponent of W_N and ${W_N}^{ux}$ is known as the weighting factor, and, ${W_N}^{ux} = \exp[-j2\pi ux/N]$.

6.1.1 Two Point DFT

Consider initially the 2 point DFT

$$F(u) = \sum_{x=0}^{1} f(x) {W_2}^{ux}$$

$$= f(0){W_2}^{0u} + f(1){W_2}^{1u} \quad u = 0, 1$$

$$W_2 = \exp[-j2\pi/2] = -1$$

Therefore

$$F(0) = f(0) \times 1 + f(1) * 1 = f(0) + f(1)$$

$$F(1) = f(0) \times 1 + f(1) * -1 = f(0) - f(1)$$

It can be noted that the DFT has now been formed using only addition and subtraction rather than complex multiplication. This is shown as a Butterfly below

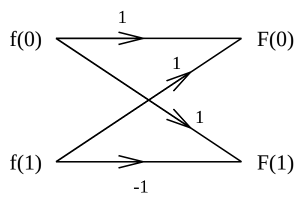


Figure 23: Signal flow graph known as a butterfly, used to determine a 2 point DFT

6.1.2 Four Point DFT

Now consider a 4 point DFT

$$F(u) = \sum_{x=0}^{3} f(x) W_4^{ux} \quad u = 0, 1, 2, 3$$

$$= f(0)W_4^{0u} + f(1)W_4^{1u} + f(2)W_4^{2u} + f(3)W_4^{3u}$$

Consider initially the 1st and 3rd terms

$$f(0)W_4^{0u} + f(2)W_4^{2u}$$

This can be simplified since

$$f_1(n) = f(2n) \qquad n = 0, 1$$

Therefore,

$$f(0)W_4^{0u} + f(2)W_4^{2u} = f_1(0)W_2^{0u} + f_1(1)W_2^{u}$$

which gives the even terms of the original sequence, and the following $F_1(u)$ is the 2 point DFT of the sequence defined as $f_1(n)$.

$$F_1(u) = f_1(0)W_2^{0u} + f_1(1)W_2^{u}$$

Now, consider the 2nd and 4th terms,

$$f(1)W_4^{1}u + f(3)W_4^{3}u = W_4^{u}(f(1)W_4^{0u} + f(3)W_4^{2u})$$

$$= W_4^{u}(f_2(0) + f_2(1)W_2^{u})$$

where $f_2(n) = f(2n+1)$ for n = 0,1, and $f_2(0) + f_2(1)W_2^{u}$ is the 2 point DFT of $F_2(n)$.

Therefore,

$$F(u) = F_1(u) + W_4^{u} F_2(u)$$

where, $F_1(u)$ is the 2 point DFT of the even terms and $F_2(u)$ is the 2 point DFT of the odd terms. W_4^{u} is the *twiddle factor* .

The 4 point DFT has therefore been formed from 2 point DFTs, using the even and odd terms to form 2 point DFTs and then using these. Again, the complexity has been reduced as is shown using the butterfly type structure of figure 24.

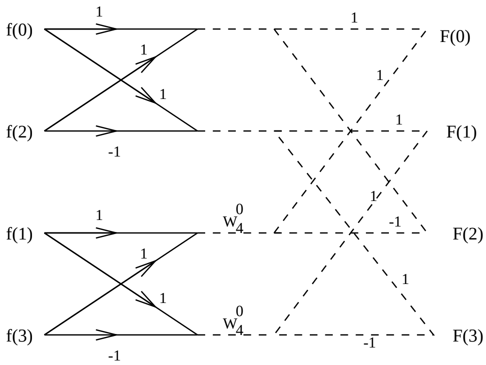


Figure 24: 4 Point DFT structure

6.1.3 Eight Point Case

This can then be extended to the 8 point case as shown in figure 25 below.

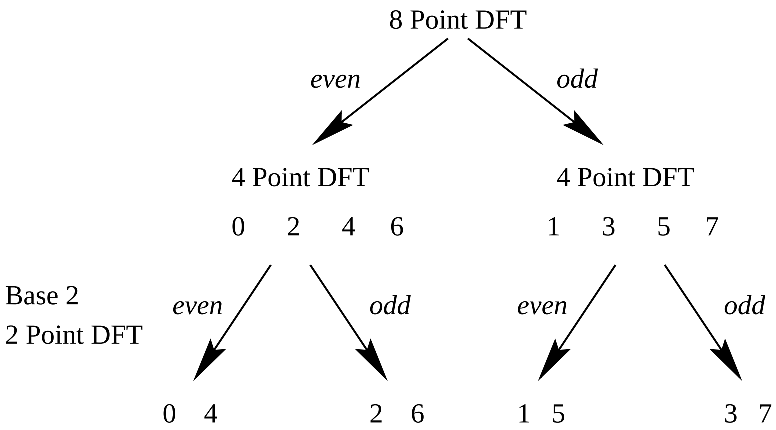


Figure 25: 8 point fft

Likewise this structure can be extended to any length of sequence, provided it contains 2^N samples.

7. Other Separable Image Transforms

The Fourier transform is one class of important transforms which can be expressed in terms of the general relationship

$$T(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) g(x,y,u,v) \tag{14}$$

where $T(u,v)$ is the transform of $f(x,y)$ and $g(x,y,u,v)$ is the *forward transformation kernel*. Similarly the inverse transform is

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u,v) h(x,y,u,v) \tag{15}$$

where $h(x,y,u,v)$ is the *inverse transformation kernel*. The *kernels* depend only on the values of x, y, u and v and not on the values of $f(x,y)$ or $T(u,v)$.

The forward kernel is said to be separable if

$$g(x,y,u,v) = g_1(x,u) g_2(y,v)$$

The kernel is symmetric if g_1 is functionally equal to g_2. In this case

$$g(x,y,u,v) = g_1(x,u) g_1(y,v)$$

A transform with a separable kernel can be computed in two steps, each requiring a 1D transform. The first transform taken is

$$T(x,v) = \sum_{y=0}^{N-1} f(x,y) g_2(y,v)$$

and the next transform is then

$$T(u,v) = \sum_{x=0}^{N\_1} T(x,v) g_1(x,u)$$

This is the method previously described for calculating the 2D Fourier transform.

If the kernel $g(x,y,u,v)$ is separable and symmetric,matrix notation may be used to denote equation 14 as

$$\mathbf{T} = \mathbf{AFA}$$

where $\mathbf{F}$ is the $N \times N$ image matrix, $\mathbf{A}$ is an $N \times N$ transformation matrix with elements $a_{ij} = g_1(i,j)$ and $\mathbf{T}$ is the resulting transform for values of u and v in the range 0 to N-1.

To obtain the inverse transform, we premultiply and postmultiply by an inverse transformation matrix $\mathbf{B}$.

$$\mathbf{BTB} = \mathbf{BAFAB}$$

if $\mathbf{B} = \mathbf{A}^{-1}$

$$\mathbf{F} = \mathbf{BTB}$$

which indicates that the digital image $\mathbf{F}$ can be recovered completely from its transform. If $\mathbf{B}$ is not equal to $\mathbf{A}^{-1}$, the image can be approximated as $\hat{\mathbf{F}}$ from

$$\hat{F} = \mathbf{BAFAB}$$

7.1 Example

Illustrate using the matrix method the calculation of the Fourier Transform of the 2 pixel by 2 pixel 2D image illustrated below.

f(0,0)	f(0,1)
f(1,0)	f(1,1)

Several transforms can be expressed in this form including the Fourier, Walsh, Hadamard, discrete cosine, Haar and Slant transforms.

7.2 Walsh Transform

The forward and inverse Walsh transform kernels are given by

$$g(x, y, u, v) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{[b_i(x)b_{n-1-i}(u) + b_i(y)b_{n-1-i}(v)]}$$

and

$$h(x, y, u, v) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{[b_i(x)b_{n-1-i}(u) + b_i(y)b_{n-1-i}(v)]}$$

where $b_k(z)$ is the k^{th} bit in the binary representation of z *ie* if z = 6 (110 in binary) $b_0(z) = 0, b_1(z) = 1$ and $b_2(z) = 1$.

The formulations for the forward and inverse kernels are identical. The kernels are separable and symmetric.

7.3 Hadamard Transform

The forward and inverse kernels are given by

$$g(x, y, u, v) = \frac{1}{N} (-1)^{\sum_{i=0}^{n-1} [b_i(x)b_i(u) + b_i(y)b_i(v)]}$$

$$h(x, y, u, v) = \frac{1}{N} (-1)^{\sum_{i=0}^{n-1} [b_i(x)b_i(u) + b_i(y)b_i(v)]}$$

7.4 Discrete Cosine Transform

The Discrete Cosine Transform for a 1D discrete signal $f(x)$ is

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{(2x+1)u\pi}{2N}\right]$$

and the inverse transformation is

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left[\frac{(2x+1)u\pi}{2N}\right]$$

where

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u = 1, 2, 3, \ldots N-1 \end{cases}$$

The 2D Transformation is

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

and the inverse transform is

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2x+1)u\pi}{2N}\right]$$

The Discrete cosine transform is often employed during image compression, as will be discussed later in the course.