1

Signals, Systems, and the Fourier Transform

1.0 INTRODUCTION

Most signals can be classified into three broad groups. One group, which consists of *analog* or *continuous-space* signals, is continuous in both space* and amplitude. In practice, a majority of signals falls into this group. Examples of analog signals include image, seismic, radar, and speech signals. Signals in the second group, *discrete-space* signals, are discrete in space and continuous in amplitude. A common way to generate discrete-space signals is by sampling analog signals. Signals in the third group, *digital* or *discrete* signals, are discrete in both space and amplitude. One way in which digital signals are created is by amplitude quantization of discrete-space signals. Discrete-space signals and digital signals are also referred to as *sequences*.

Digital systems and computers use only digital signals, which are discrete in both space and amplitude. The development of signal processing concepts based on digital signals, however, requires a detailed treatment of amplitude quantization, which is extremely difficult and tedious. Many useful insights would be lost in such a treatment because of its mathematical complexity. For this reason, most digital signal processing concepts have been developed based on discrete-space signals. Experience shows that theories based on discrete-space signals are often applicable to digital signals.

A system maps an input signal to an output signal. A major element in studying signal processing is the analysis, design, and implementation of a system that transforms an input signal to a more desirable output signal for a given application. When developing theoretical results about systems, we often impose

*Although we refer to "space," an analog signal can instead have a variable in time, as in the case of speech processing.

the constraints of linearity and shift invariance. Although these constraints are very restrictive, the theoretical results thus obtained apply in practice at least approximately to many systems. We will discuss signals and systems in Sections 1.1 and 1.2, respectively.

The Fourier transform representation of signals and systems plays a central role in both one-dimensional (1-D) and two-dimensional (2-D) signal processing. In Sections 1.3 and 1.4, the Fourier transform representation including some aspects that are specific to image processing applications is discussed. In Section 1.5, we discuss digital processing of analog signals. Many of the theoretical results, such as the 2-D sampling theorem summarized in that section, can be derived from the Fourier transform results.

Many of the theoretical results discussed in this chapter can be viewed as straightforward extensions of the one-dimensional case. Some, however, are unique to two-dimensional signal processing. Very naturally, we will place considerably more emphasis on these. We will now begin our journey with the discussion of signals.

1.1 SIGNALS

The signals we consider are discrete-space signals. A 2-D discrete-space signal (sequence) will be denoted by a function whose two arguments are integers. For example, $x(n_1, n_2)$ represents a sequence which is defined for all integer values of n_1 and n_2. Note that $x(n_1, n_2)$ for a noninteger n_1 or n_2 is not zero, but is undefined. The notation $x(n_1, n_2)$ may refer either to the discrete-space function x or to the value of the function x at a specific (n_1, n_2). The distinction between these two will be evident from the context.

An example of a 2-D sequence $x(n_1, n_2)$ is sketched in Figure 1.1. In the figure, the height at (n_1, n_2) represents the amplitude at (n_1, n_2). It is often tedious to sketch a 2-D sequence in the three-dimensional (3-D) perspective plot as shown

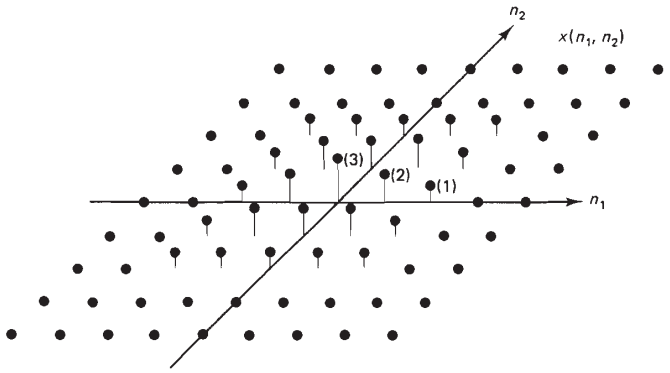


Figure 1.1 2-D sequence $x(n_1, n_2)$.

in Figure 1.1. An alternate way to sketch the 2-D sequence in Figure 1.1 is shown in Figure 1.2. In this figure, open circles represent amplitudes of 0 and filled-in circles represent nonzero amplitudes, with the values in parentheses representing the amplitudes. For example, $x(3, 0)$ is 0 and $x(1, 1)$ is 2.

Many sequences we use have amplitudes of 0 or 1 for large regions of (n_1, n_2). In such instances, the open circles and parentheses will be eliminated for convenience. If there is neither an open circle nor a filled-in circle at a particular (n_1, n_2), then the sequence has zero amplitude at that point. If there is a filled-in circle with no amplitude specification at a particular (n_1, n_2), then the sequence has an amplitude of 1 at that point. Figure 1.3 shows the result when this additional simplification is made to the sequence in Figure 1.2.

1.1.1 Examples of Sequences

Certain sequences and classes of sequences play a particularly important role in 2-D signal processing. These are impulses, step sequences, exponential sequences, separable sequences, and periodic sequences.

Impulses. The impulse or unit sample sequence, denoted by $\delta(n_1, n_2)$, is defined as

$$\delta(n_1, n_2) = \begin{cases} 1, & n_1 = n_2 = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{1.1}$$

The sequence $\delta(n_1, n_2)$, sketched in Figure 1.4, plays a role similar to the impulse $\delta(n)$ in 1-D signal processing.

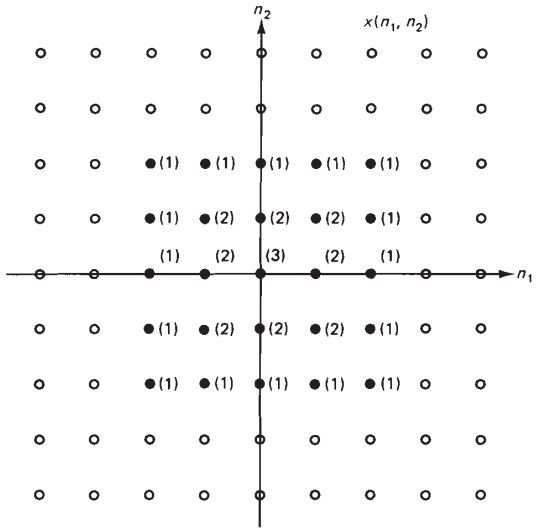


Figure 1.2 Alternate way to sketch the 2-D sequence in Figure 1.1. Open circles represent amplitudes of zero, and filled-in circles represent nonzero amplitudes, with values in parentheses representing the amplitude.

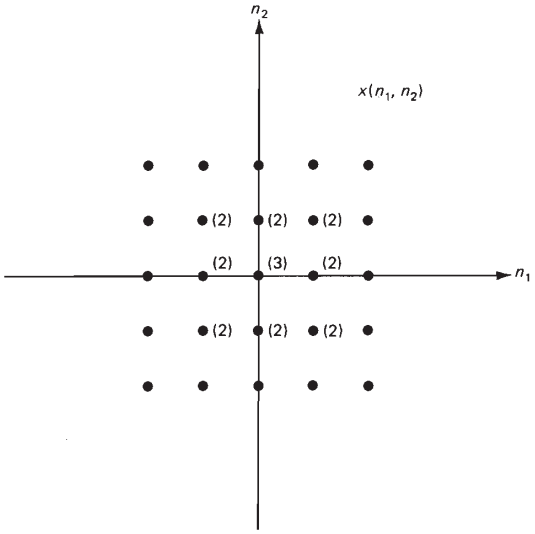


Figure 1.3 Sequence in Figure 1.2 sketched with some simplification. Open circles have been eliminated and filled-in circles with amplitude of 1 have no amplitude specifications.

Any sequence $x(n_1, n_2)$ can be represented as a linear combination of shifted impulses as follows:

$$\begin{aligned} x(n_1,n_2) = {} & \cdots + x(-1,-1)\,\delta(n_1+1,n_2+1) + x(0,-1)\,\delta(n_1,n_2+1) \\ & + x(1,-1)\,\delta(n_1-1,n_2+1) + \cdots + x(-1,0)\,\delta(n_1+1,n_2) \\ & + x(0,0)\,\delta(n_1,n_2) + x(1,0)\,\delta(n_1-1,n_2) \\ & + \cdots + x(-1,1)\,\delta(n_1+1,n_2-1) \\ & + x(0,1)\,\delta(n_1,n_2-1) + x(1,1)\,\delta(n_1-1,n_2-1) + \cdots \\ = {} & \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1,k_2)\,\delta(n_1-k_1,n_2-k_2). \end{aligned} \tag{1.2}$$

The representation of $x(n_1, n_2)$ by (1.2) is very useful in system analysis.

Line impulses constitute a class of impulses which do not have any counterparts in 1-D. An example of a line impulse is the 2-D sequence $\delta_T(n_1)$, which is sketched in Figure 1.5 and is defined as

$$x(n_1, n_2) = \delta_T(n_1) = \begin{cases} 1, & n_1 = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{1.3}$$

Other examples include $\delta_T(n_2)$ and $\delta_T(n_1 - n_2)$, which are defined similarly to $\delta_T(n_1)$. The subscript T in $\delta_T(n_1)$ indicates that $\delta_T(n_1)$ is a 2-D sequence. This notation is used to avoid confusion in cases where the 2-D sequence is a function of only one variable. For example, without the subscript T, $\delta_T(n_1)$ might be

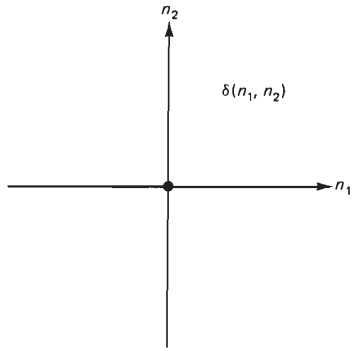


Figure 1.4 Impulse $\delta(n_1, n_2)$.

confused with the 1-D impulse $\delta(n_1)$. For clarity, then, the subscript T will be used whenever a 2-D sequence is a function of one variable. The sequence $x_T(n_1)$ is thus a 2-D sequence, while $x(n_1)$ is a 1-D sequence.

Step sequences. The unit step sequence, denoted by $u(n_1, n_2)$, is defined as

$$u(n_1, n_2) = \begin{cases} 1, & n_1, n_2 \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{1.4}$$

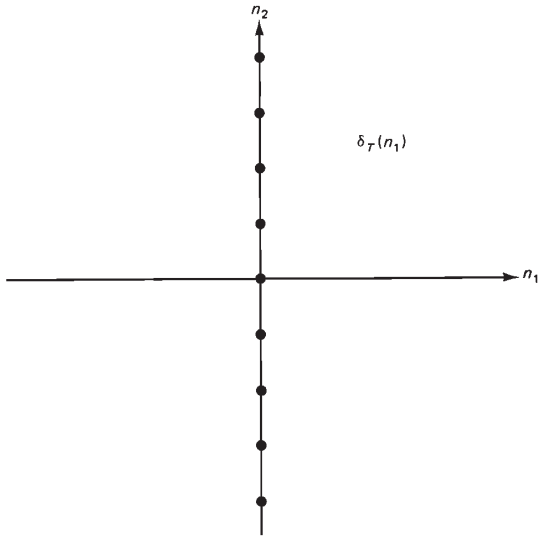


Figure 1.5 Line impulse $\delta_T(n_1)$.

The sequence $u(n_1, n_2)$, which is sketched in Figure 1.6, is related to $\delta(n_1, n_2)$ as

$$u(n_1, n_2) = \sum_{k_1=-\infty}^{n_1} \sum_{k_2=-\infty}^{n_2} \delta(k_1, k_2) \tag{1.5a}$$

or

$$\delta(n_1, n_2) = u(n_1, n_2) - u(n_1 - 1, n_2) - u(n_1, n_2 - 1) + u(n_1 - 1, n_2 - 1). \tag{1.5b}$$

Some step sequences have no counterparts in 1-D. An example is the 2-D sequence $u_T(n_1)$, which is sketched in Figure 1.7 and is defined as

$$x(n_1, n_2) = u_T(n_1) = \begin{cases} 1, & n_1 \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{1.6}$$

Other examples include $u_T(n_2)$ and $u_T(n_1 - n_2)$, which are defined similarly to $u_T(n_1)$.

Exponential sequences. Exponential sequences of the type $x(n_1, n_2) = A\alpha^{n_1}\beta^{n_2}$ are important for system analysis. As we shall see later, sequences of this class are eigenfunctions of linear shift-invariant (LSI) systems.

Separable sequences. A 2-D sequence $x(n_1, n_2)$ is said to be a separable sequence if it can be expressed as

$$x(n_1, n_2) = f(n_1)g(n_2) \tag{1.7}$$

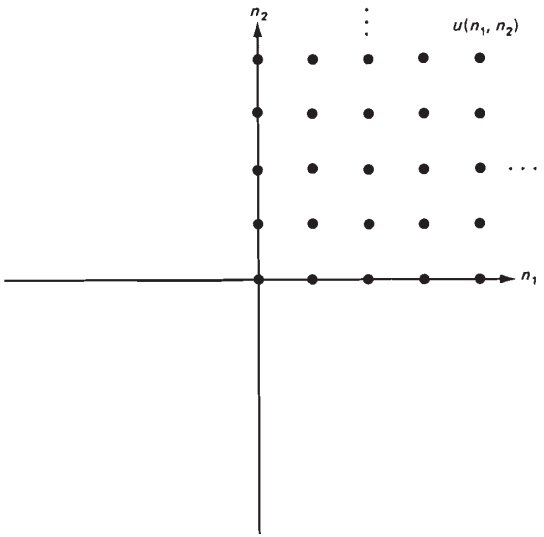


Figure 1.6 Unit step sequence $u(n_1, n_2)$.

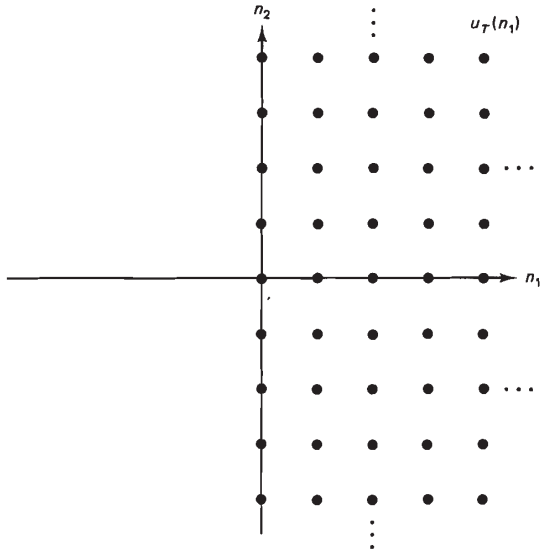


Figure 1.7 Step sequence $u_T(n_1)$.

where $f(n_1)$ is a function of only n_1 and $g(n_2)$ is a function of only n_2. Although it is possible to view $f(n_1)$ and $g(n_2)$ as 2-D sequences, it is more convenient to consider them to be 1-D sequences. For that reason, we use the notations $f(n_1)$ and $g(n_2)$ rather than $f_T(n_1)$ and $g_T(n_2)$.

The impulse $\delta(n_1, n_2)$ is a separable sequence since $\delta(n_1, n_2)$ can be expressed as

$$\delta(n_1, n_2) = \delta(n_1)\,\delta(n_2) \tag{1.8}$$

where $\delta(n_1)$ and $\delta(n_2)$ are 1-D impulses. The unit step sequence $u(n_1, n_2)$ is also a separable sequence since $u(n_1, n_2)$ can be expressed as

$$u(n_1, n_2) = u(n_1)u(n_2) \tag{1.9}$$

where $u(n_1)$ and $u(n_2)$ are 1-D unit step sequences. Another example of a separable sequence is $a^{n_1}b^{n_2} + b^{n_1+n_2}$, which can be written as $(a^{n_1} + b^{n_1})b^{n_2}$.

Separable sequences form a very special class of 2-D sequences. A typical 2-D sequence is not a separable sequence. As an illustration, consider a sequence $x(n_1, n_2)$ which is zero outside $0 \leq n_1 \leq N_1 - 1$ and $0 \leq n_2 \leq N_2 - 1$. A general sequence $x(n_1, n_2)$ of this type has N_1N_2 degrees of freedom. If $x(n_1, n_2)$ is a separable sequence, $x(n_1, n_2)$ is completely specified by some $f(n_1)$ which is zero outside $0 \leq n_1 \leq N_1 - 1$ and some $g(n_2)$ which is zero outside $0 \leq n_2 \leq N_2 - 1$, and consequently has only $N_1 + N_2 - 1$ degrees of freedom.

Despite the fact that separable sequences constitute a very special class of 2-D sequences, they play an important role in 2-D signal processing. In those cases where the results that apply to 1-D sequences do not extend to general 2-D sequences in a straightforward manner, they often do for separable 2-D sequences.

In addition, the separability of the sequence can be exploited in order to reduce computation in various contexts, such as digital filtering and computation of the discrete Fourier transform. This will be discussed further in later sections.

Periodic sequences. A sequence $x(n_1, n_2)$ is said to be periodic with a period of $N_1 \times N_2$ if $x(n_1, n_2)$ satisfies the following condition:

$$x(n_1, n_2) = x(n_1 + N_1, n_2) = x(n_1, n_2 + N_2) \quad \text{for all } (n_1, n_2) \qquad (1.10)$$

where N_1 and N_2 are positive integers. For example, $\cos(\pi n_1 + (\pi/2)n_2)$ is a periodic sequence with a period of 2×4, since $\cos(\pi n_1 + (\pi/2)n_2) = \cos(\pi(n_1 + 2) + (\pi/2)n_2) = \cos(\pi n_1 + (\pi/2)(n_2 + 4))$ for all (n_1, n_2). The sequence $\cos(n_1 + n_2)$ is not periodic, however, since $\cos(n_1 + n_2)$ cannot be expressed as $\cos((n_1 + N_1) + n_2) = \cos(n_1 + (n_2 + N_2))$ for all (n_1, n_2) for any nonzero integers N_1 and N_2. A periodic sequence is often denoted by adding a "~" (tilde), for example, $\tilde{x}(n_1, n_2)$, to distinguish it from an aperiodic sequence.

Equation (1.10) is not the most general representation of a 2-D periodic sequence. As an illustration, consider the sequence $x(n_1, n_2)$ shown in Figure 1.8. Even though $x(n_1, n_2)$ can be considered a periodic sequence with a period of 3×2 it cannot be represented as such a sequence by using (1.10). Specifically, $x(n_1, n_2) \neq x(n_1 + 3, n_2)$ for all (n_1, n_2). It is possible to generalize (1.10) to incorporate cases such as that in Figure 1.8. However, in this text we will use (1.10) to define a periodic sequence, since it is sufficient for our purposes, and sequences such as that in Figure 1.8 can be represented by (1.10) by increasing N_1 and/or N_2. For example, the sequence in Figure 1.8 is periodic with a period of 6×2 using (1.10).

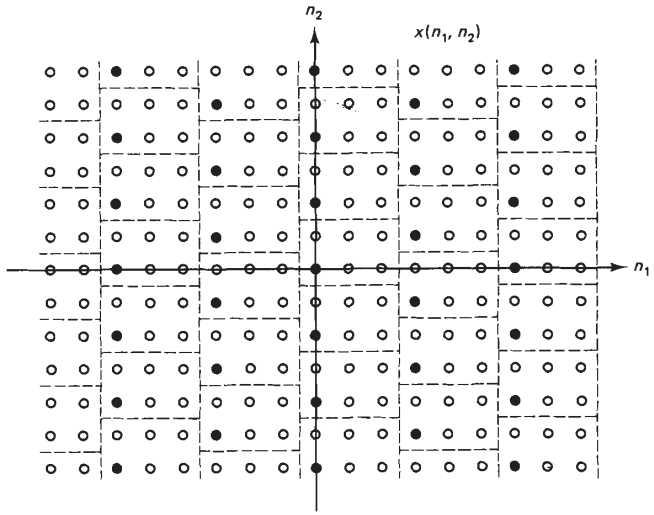
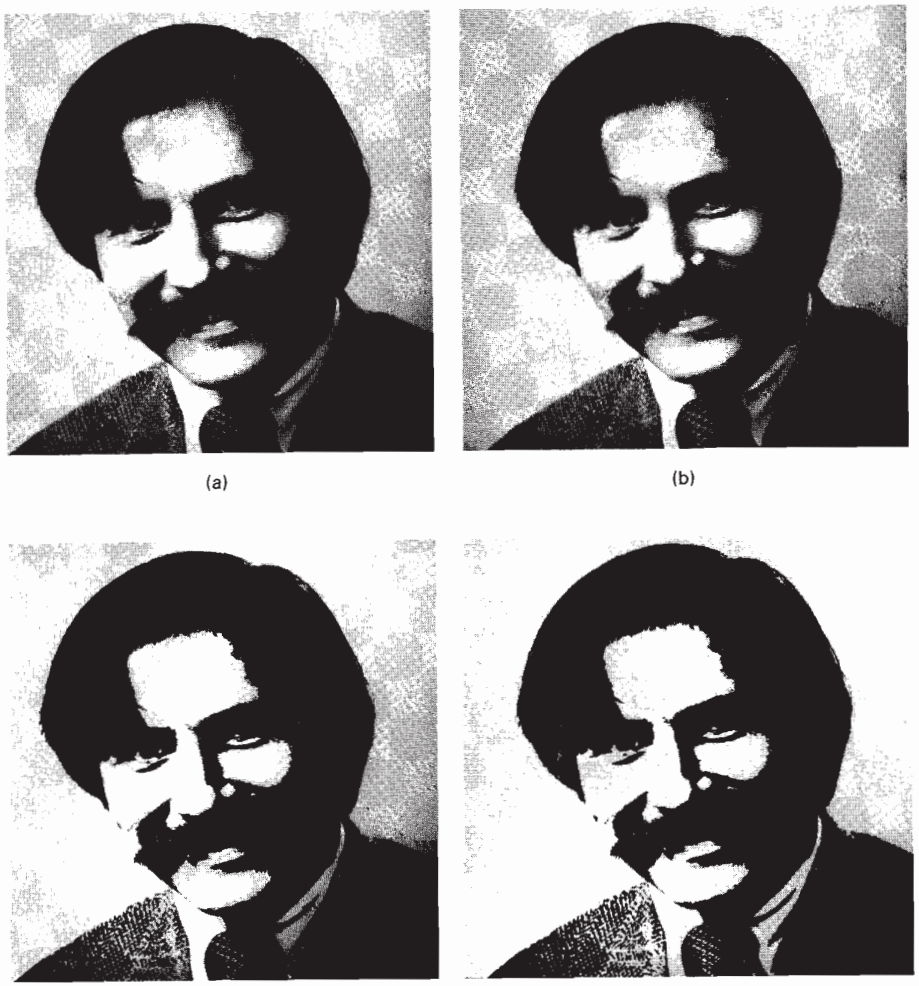


Figure 1.8 Periodic sequence with a period of 6×2.

1.1.2 Digital Images

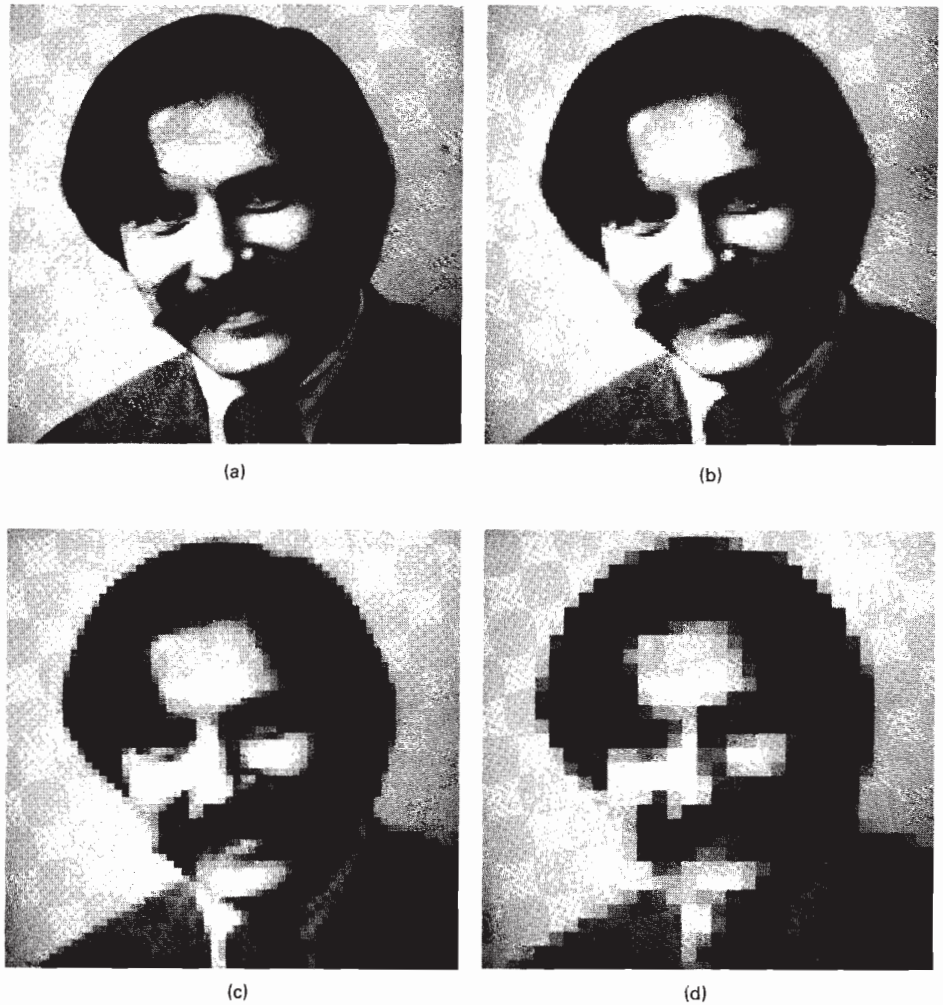
Many examples of sequences used in this book are digital images. A digital image, which can be denoted by $x(n_1, n_2)$, is typically obtained by sampling an analog image, for instance, an image on film. The amplitude of a digital image is often quantized to 256 levels (which can be represented by eight bits). Each level is commonly denoted by an integer, with 0 corresponding to the darkest level and 255 to the brightest. Each point (n_1, n_2) is called a pixel or pel (picture element). A digital image $x(n_1, n_2)$ of 512×512 pixels with each pixel represented by eight bits is shown in Figure 1.9. As we reduce the number of amplitude quantization levels, the signal-dependent quantization noise begins to appear as false contours. This is shown in Figure 1.10, where the image in Figure 1.9 is displayed with 64 levels (six bits), 16 levels (four bits), 4 levels (two bits), and 2 levels (one bit) of amplitude quantization. As we reduce the number of pixels in a digital image, the spatial resolution is decreased and the details in the image begin to disappear. This is shown in Figure 1.11, where the image in Figure 1.9 is displayed at a spatial resolution of 256×256 pixels, 128×128 pixels, 64×64 pixels, and 32×32 pixels. A digital image of 512×512 pixels has a spatial resolution similar to that seen in a television frame. To have a spatial resolution similar to that of an image on 35-mm film, we need a spatial resolution of 1024×1024 pixels in the digital image.

Figure 1.9 Digital image of 512×512 pixels quantized at 8 bits/pixel.



(a) (b) (c) (d)

Figure 1.10 Image in Figure 1.9 with amplitude quantization at (a) 6 bits/pixel, (b) 4 bits/pixel, (c) 2 bits/pixel, and (d) 1 bit/pixel.



(a) (b) (c) (d)

Figure 1.11 Image in Figure 1.9 with spatial resolution of (a) 256 × 256 pixels, (b) 128 × 128 pixels, (c) 64 × 64 pixels, and (d) 32 × 32 pixels.

1.2 SYSTEMS

1.2.1 Linear Systems and Shift-Invariant Systems

An input-output relationship is called a system if there is a unique output for any given input. A system T that relates an input $x(n_1, n_2)$ to an output $y(n_1, n_2)$ is represented by

$$y(n_1, n_2) = T[x(n_1, n_2)]. \tag{1.11}$$

This definition of a system is very broad. Without any restrictions, characterizing a system requires a complete input-output relationship. Knowing the output of a system to one set of inputs does not generally allow us to determine the output of the system to any other set of inputs. Two types of restriction which greatly simplify the characterization and analysis of a system are linearity and shift invariance. In practice, fortunately, many systems can be approximated to be linear and shift invariant.

The linearity of a system T is defined as

$$\text{Linearity} \iff T[ax_1(n_1, n_2) + bx_2(n_1, n_2)] = ay_1(n_1, n_2) + by_2(n_1, n_2) \tag{1.12}$$

where $T[x_1(n_1, n_2)] = y_1(n_1, n_2)$, $T[x_2(n_1, n_2)] = y_2(n_1, n_2)$, a and b are any scalar constants, and $A \iff B$ means that A implies B and B implies A. The condition in (1.12) is called the *principle of superposition*. To illustrate this concept, a linear system and a nonlinear system are shown in Figure 1.12. The linearity of the system in Figure 1.12(a) and the nonlinearity of the system in Figure 1.12(b) can be easily verified by using (1.12).

The shift invariance (SI) or space invariance of a system is defined as

$$\text{Shift invariance} \iff T[x(n_1 - m_1, n_2 - m_2)] = y(n_1 - m_1, n_2 - m_2) \tag{1.13}$$

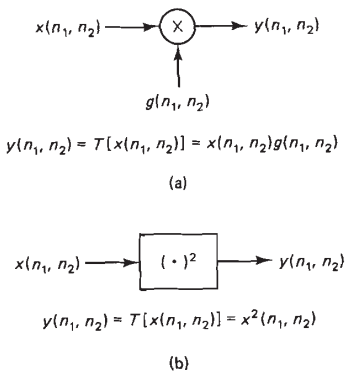


Figure 1.12 (a) Example of a linear shift-variant system; (b) example of a nonlinear shift-invariant system.

where $y(n_1, n_2) = T[x(n_1, n_2)]$ and m_1 and m_2 are any integers. The system in Figure 1.12(a) is not shift invariant since $T[x(n_1 - m_1, n_2 - m_2)] = x(n_1 - m_1, n_2 - m_2)g(n_1, n_2)$ and $y(n_1 - m_1, n_2 - m_2) = x(n_1 - m_1, n_2 - m_2)g(n_1 - m_1, n_2 - m_2)$. The system in Figure 1.12(b), however, is shift invariant, since $T[x(n_1 - m_1, n_2 - m_2)] = x^2(n_1 - m_1, n_2 - m_2)$ and $y(n_1 - m_1, n_2 - m_2) = x^2(n_1 - m_1, n_2 - m_2)$.

Consider a linear system T. Using (1.2) and (1.12), we can express the output $y(n_1, n_2)$ for an input $x(n_1, n_2)$ as

$$\begin{aligned} y(n_1, n_2) &= T[x(n_1, n_2)] = T\left[\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} x(k_1, k_2)\delta(n_1 - k_1, n_2 - k_2)\right] \\ &= \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} x(k_1, k_2)T[\delta(n_1 - k_1, n_2 - k_2)]. \end{aligned} \tag{1.14}$$

From (1.14), a linear system can be completely characterized by the response of the system to the impulse $\delta(n_1, n_2)$ and its shifts $\delta(n_1 - k_1, n_2 - k_2)$. If we know $T[\delta(n_1 - k_1, n_2 - k_2)]$ for all integer values of k_1 and k_2, the output of the linear system to any input $x(n_1, n_2)$ can be obtained from (1.14). For a nonlinear system, knowledge of $T[\delta(n_1 - k_1, n_2 - k_2)]$ for all integer values of k_1 and k_2 does not tell us the output of the system when the input $x(n_1, n_2)$ is $2\delta(n_1, n_2)$, $\delta(n_1, n_2) + \delta(n_1 - 1, n_2)$, or many other sequences.

System characterization is further simplified if we impose the additional restriction of shift invariance. Suppose we denote the response of a system T to an input $\delta(n_1, n_2)$ by $h(n_1, n_2)$;

$$h(n_1, n_2) = T[\delta(n_1, n_2)]. \tag{1.15}$$

From (1.13) and (1.15),

$$h(n_1 - k_1, n_2 - k_2) = T[\delta(n_1 - k_1, n_2 - k_2)] \tag{1.16}$$

for a shift-invariant system T. For a linear and shift-invariant (LSI) system, then, from (1.14) and (1.16), the input-output relation is given by

$$y(n_1, n_2) = T[x(n_1, n_2)] = \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} x(k_1, k_2)h(n_1 - k_1, n_2 - k_2). \tag{1.17}$$

Equation (1.17) states that an LSI system is completely characterized by the impulse response $h(n_1, n_2)$. Specifically, for an LSI system, knowledge of $h(n_1, n_2)$ alone allows us to determine the output of the system to any input from (1.17). Equation (1.17) is referred to as *convolution*, and is denoted by the convolution operator "$*$" as follows:

For an LSI system,

$$\begin{aligned} y(n_1, n_2) &= x(n_1, n_2) * h(n_1, n_2) \\ &= \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} x(k_1, k_2)h(n_1 - k_1, n_2 - k_2). \end{aligned} \tag{1.18}$$

Note that the impulse response $h(n_1, n_2)$, which plays such an important role for an LSI system, loses its significance for a nonlinear or shift-variant system. Note also that an LSI system can be completely characterized by the system response to one of many other input sequences. The choice of $\delta(n_1, n_2)$ as the input in characterizing an LSI system is the simplest, both conceptually and in practice.

1.2.2 Convolution

The convolution operator in (1.18) has a number of properties that are straightforward extensions of 1-D results. Some of the more important are listed below.

Commutativity

$$x(n_1, n_2) * y(n_1, n_2) = y(n_1, n_2) * x(n_1, n_2) \tag{1.19}$$

Associativity

$$(x(n_1, n_2) * y(n_1, n_2)) * z(n_1, n_2) = x(n_1, n_2) * (y(n_1, n_2) * z(n_1, n_2)) \tag{1.20}$$

Distributivity

$$x(n_1, n_2) * (y(n_1, n_2) + z(n_1, n_2)) = (x(n_1, n_2) * y(n_1, n_2)) + (x(n_1, n_2) * z(n_1, n_2)) \tag{1.21}$$

Convolution with Shifted Impulse

$$x(n_1, n_2) * \delta(n_1 - m_1, n_2 - m_2) = x(n_1 - m_1, n_2 - m_2) \tag{1.22}$$

The commutativity property states that the output of an LSI system is not affected when the input and the impulse response interchange roles. The associativity property states that a cascade of two LSI systems with impulse responses $h_1(n_1, n_2)$ and $h_2(n_1, n_2)$ has the same input-output relationship as one LSI system with impulse response $h_1(n_1, n_2) * h_2(n_1, n_2)$. The distributivity property states that a parallel combination of two LSI systems with impulse responses $h_1(n_1, n_2)$ and $h_2(n_1, n_2)$ has the same input-output relationship as one LSI system with impulse response given by $h_1(n_1, n_2) + h_2(n_1, n_2)$. In a special case of (1.22), when $m_1 = m_2 = 0$, we see that the impulse response of an identity system is $\delta(n_1, n_2)$.

The convolution of two sequences $x(n_1, n_2)$ and $h(n_1, n_2)$ can be obtained by explicitly evaluating (1.18). It is often simpler and more instructive, however, to evaluate (1.18) graphically. Specifically, the convolution sum in (1.18) can be interpreted as multiplying two sequences $x(k_1, k_2)$ and $h(n_1 - k_1, n_2 - k_2)$, which are functions of the variables k_1 and k_2, and summing the product over all integer values of k_1 and k_2. The output, which is a function of n_1 and n_2, is the result of convolving $x(n_1, n_2)$ and $h(n_1, n_2)$. To illustrate, consider the two sequences $x(n_1, n_2)$ and $h(n_1, n_2)$, shown in Figures 1.13(a) and (b). From $x(n_1, n_2)$ and $h(n_1, n_2)$, $x(k_1, k_2)$ and $h(n_1 - k_1, n_2 - k_2)$ as functions of k_1 and k_2 can be obtained, as shown in Figures 1.13(c)–(f). Note that $g(k_1 - n_1, k_2 - n_2)$ is

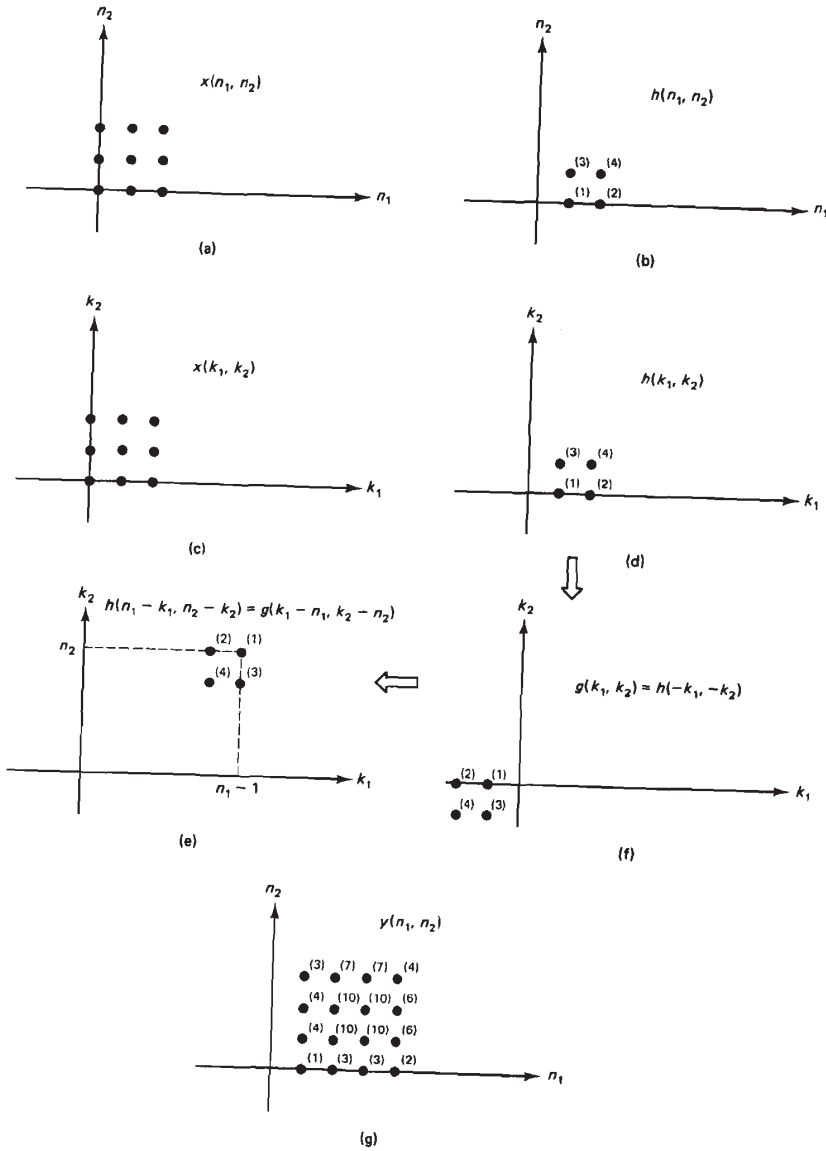


Figure 1.13 Example of convolving two sequences.

$g(k_1, k_2)$ shifted in the positive k_1 and k_2 directions by n_1 and n_2 points, respectively. Figures 1.13(d)–(f) show how to obtain $h(n_1 - k_1, n_2 - k_2)$ as a function of k_1 and k_2 from $h(n_1, n_2)$ in three steps. It is useful to remember how to obtain $h(n_1 - k_1, n_2 - k_2)$ directly from $h(n_1, n_2)$. One simple way is to first change the variables n_1 and n_2 to k_1 and k_2, flip the sequence with respect to the origin, and then shift the result in the positive k_1 and k_2 directions by n_1 and n_2 points, respectively. Once $x(k_1, k_2)$ and $h(n_1 - k_1, n_2 - k_2)$ are obtained, they can be multiplied and summed over k_1 and k_2 to produce the output at each different value of (n_1, n_2). The result is shown in Figure 1.13(g).

An LSI system is said to be ***separable***, if its impulse response $h(n_1, n_2)$ is a separable sequence. For a separable system, it is possible to reduce the number of arithmetic operations required to compute the convolution sum. For large amounts of data, as typically found in images, the computational reduction can be considerable. To illustrate this, consider an input sequence $x(n_1, n_2)$ of $N \times N$ points and an impulse response $h(n_1, n_2)$ of $M \times M$ points:

$$\begin{aligned} &x(n_1, n_2) = 0 \quad \text{outside} \quad 0 \le n_1 \le N - 1, \qquad 0 \le n_2 \le N - 1 \\ \text{and} \quad &h(n_1, n_2) = 0 \quad \text{outside} \quad 0 \le n_1 \le M - 1, \qquad 0 \le n_2 \le M - 1 \end{aligned} \tag{1.23}$$

where $N >> M$ in typical cases. The regions of (n_1, n_2) where $x(n_1, n_2)$ and $h(n_1, n_2)$ can have nonzero amplitudes are shown in Figures 1.14(a) and (b). The output of the system, $y(n_1, n_2)$, can be expressed as

$$\begin{aligned} y(n_1, n_2) &= x(n_1, n_2) * h(n_1, n_2) \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2). \end{aligned} \tag{1.24}$$

The region of (n_1, n_2) where $y(n_1, n_2)$ has nonzero amplitude is shown in Figure 1.14(c). If (1.24) is used directly to compute $y(n_1, n_2)$, approximately $(N + M - 1)^2 M^2$ arithmetic operations (one arithmetic operation = one multiplication and one addition) are required since the number of nonzero output points is $(N + M - 1)^2$ and computing each output point requires approximately M^2 arithmetic operations. If $h(n_1, n_2)$ is a separable sequence, it can be expressed as

$$\begin{aligned} &h(n_1, n_2) = h_1(n_1) h_2(n_2) \\ &h_1(n_1) = 0 \quad \text{outside} \quad 0 \le n_1 \le M - 1 \\ &h_2(n_2) = 0 \quad \text{outside} \quad 0 \le n_2 \le M - 1. \end{aligned} \tag{1.25}$$

From (1.24) and (1.25),

$$\begin{aligned} y(n_1, n_2) &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h_1(n_1 - k_1) h_2(n_2 - k_2) \\ &= \sum_{k_1=-\infty}^{\infty} h_1(n_1 - k_1) \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h_2(n_2 - k_2). \end{aligned} \tag{1.26}$$

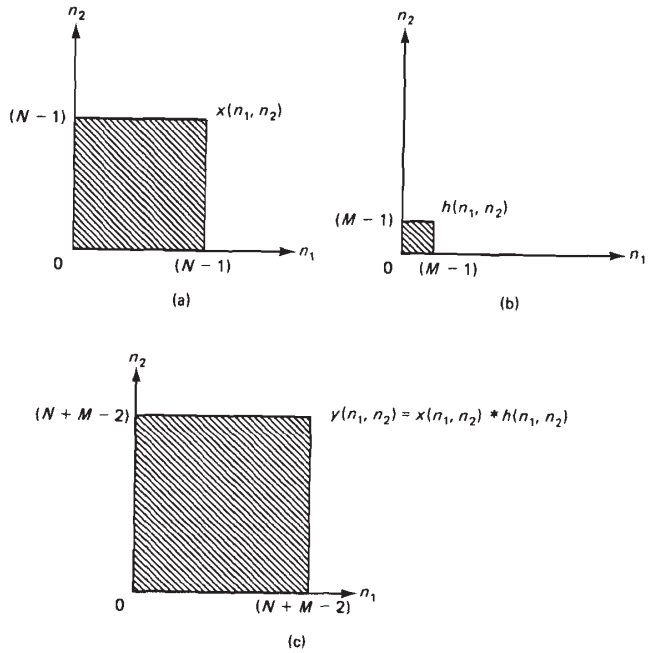


Figure 1.14 Regions of (n_1, n_2) where $x(n_1, n_2)$, $h(n_1, n_2)$, and $y(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2)$ can have nonzero amplitude.

For a fixed k_1, $\sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h_2(n_2 - k_2)$ in (1.26) is a 1-D convolution of $x(k_1, n_2)$ and $h_2(n_2)$. For example, using the notation

$$f(k_1, n_2) = \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h_2(n_2 - k_2), \tag{1.27}$$

$f(0, n_2)$ is the result of 1-D convolution of $x(0, n_2)$ with $h_2(n_2)$, as shown in Figure 1.15. Since there are N different values of k_1 for which $x(k_1, k_2)$ is nonzero, computing $f(k_1, n_2)$ requires N 1-D convolutions and therefore requires approximately $NM(N + M - 1)$ arithmetic operations. Once $f(k_1, n_2)$ is computed, $y(n_1, n_2)$ can be computed from (1.26) and (1.27) by

$$y(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} h_1(n_1 - k_1) f(k_1, n_2). \tag{1.28}$$

From (1.28), for a fixed n_2, $y(n_1, n_2)$ is a 1-D convolution of $h_1(n_1)$ and $f(n_1, n_2)$. For example, $y(n_1, 1)$ is the result of a 1-D convolution of $f(n_1, 1)$ and $h_1(n_1)$, as shown in Figure 1.15, where $f(n_1, n_2)$ is obtained from $f(k_1, n_2)$ by a simple

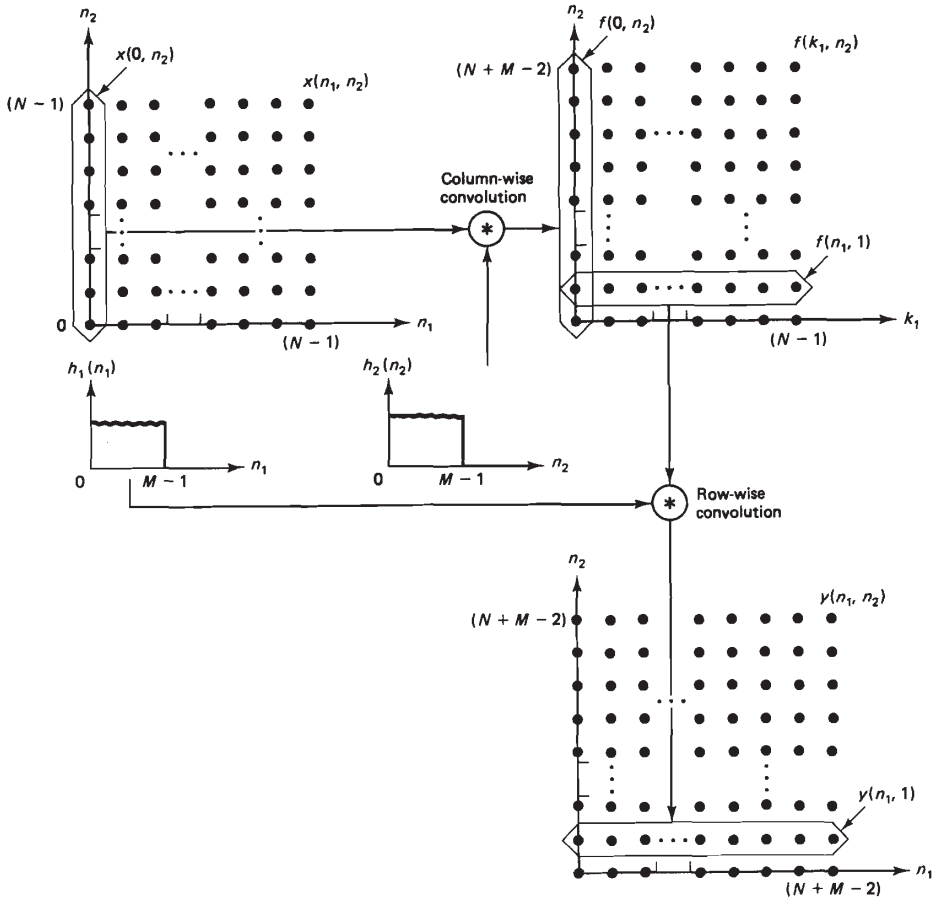


Figure 1.15 Convolution of $x(n_1, n_2)$ with a separable sequence $h(n_1, n_2)$.

change of variables. Since there are $N + M - 1$ different values of n_2, computing $y(n_1, n_2)$ from $f(k_1, n_2)$ requires $N + M - 1$ 1-D convolutions thus approximately $M(N + M - 1)^2$ arithmetic operations. Computing $y(n_1, n_2)$ from (1.27) and (1.28), exploiting the separability of $h(n_1, n_2)$, requires approximately $NM(N + M - 1) + M(N + M - 1)^2$ arithmetic operations. This can be a considerable computational saving over $(N + M - 1)^2M^2$. If we assume $N >> M$, exploiting the separability of $h(n_1, n_2)$ reduces the number of arithmetic operations by approximately a factor of $M/2$.

As an example, consider $x(n_1, n_2)$ and $h(n_1, n_2)$, shown in Figures 1.16(a) and (b). The sequence $h(n_1, n_2)$ can be expressed as $h_1(n_1)h_2(n_2)$, where $h_1(n_1)$ and

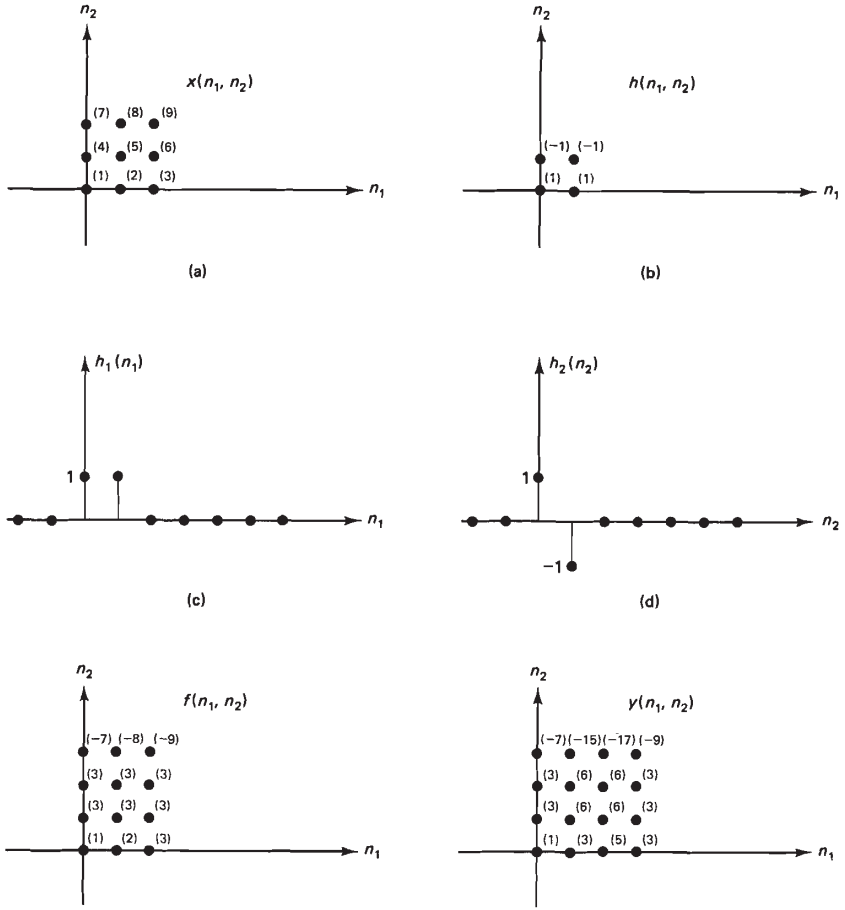


Figure 1.16 Example of convolving $x(n_1, n_2)$ with a separable sequence $h(n_1, n_2)$.

$h_2(n_2)$ are shown in Figures 1.16(c) and (d), respectively. The sequences $f(n_1, n_2)$ and $y(n_1, n_2)$ are shown in Figures 1.16(e) and (f).

In the above discussion, we performed a 1-D convolution first for each column of $x(n_1, n_2)$ with $h_2(n_2)$ and then a 1-D convolution for each row of $f(n_1, n_2)$ with $h_1(n_1)$. By changing the order of the two summations in (1.26) and following the same procedure, it is simple to show that $y(n_1, n_2)$ can be computed by performing a 1-D convolution first for each row of $x(n_1, n_2)$ with $h_1(n_1)$ and then a 1-D convolution for each column of the result with $h_2(n_2)$. In the above discussion, we have assumed that $x(n_1, n_2)$ and $h(n_1, n_2)$ are $N_1 \times N_2$-point and $M_1 \times M_2$-

point sequences respectively with $N_1 = N_2$ and $M_1 = M_2$. We note that the results discussed above can be generalized straightforwardly to the case when $N_1 \neq N_2$ and $M_1 \neq M_2$.

1.2.3 Stable Systems and Special Support Systems

For practical reasons, it is often appropriate to impose additional constraints on the class of systems we consider. Stable systems and special support systems have such constraints.

Stable systems. A system is considered stable in the bounded-input-bounded-output (BIBO) sense if and only if a bounded input always leads to a bounded output. Stability is often a desirable constraint to impose, since an unstable system can generate an unbounded output, which can cause system overload or other difficulties. From this definition and (1.18), it can be shown that a necessary and sufficient condition for an LSI system to be stable is that its impulse response $h(n_1, n_2)$ be absolutely summable:

$$\text{Stability of an LSI system} \iff \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |h(n_1, n_2)| < \infty. \qquad (1.29)$$

Although (1.29) is a straightforward extension of 1-D results, 2-D systems differ greatly from 1-D systems when a system's stability is tested. This will be discussed further in Section 2.3. Because of (1.29), an absolutely summable sequence is defined to be a *stable sequence*. Using this definition, a necessary and sufficient condition for an LSI system to be stable is that its impulse response be a stable sequence.

Special support systems. A 1-D system is said to be causal if and only if the current output $y(n)$ does not depend on any future values of the input, for example, $x(n+1), x(n+2), x(n+3), \ldots$. Using this definition, we can show that a necessary and sufficient condition for a 1-D LSI system to be causal is that its impulse response $h(n)$ be zero for $n < 0$. Causality is often a desirable constraint to impose in designing 1-D systems. A noncausal system would require delay, which is undesirable in such applications as real time speech processing. In typical 2-D signal processing applications such as image processing, the causality constraint may not be necessary. At any given time, a complete frame of an image may be available for processing, and it may be processed from left to right, from top to bottom, or in any direction one chooses. Although the notion of causality may not be useful in 2-D signal processing, it is useful to extend the notion that a 1-D causal LSI system has an impulse response $h(n)$ whose nonzero values lie in a particular region. A 2-D LSI system whose impulse response $h(n_1, n_2)$ has all its nonzero values in a particular region is called a special support system.

A 2-D LSI system is said to be a *quadrant support system* when its impulse response $h(n_1, n_2)$ is a *quadrant support sequence*. A quadrant support sequence, or a quadrant sequence for short, is one which has all its nonzero values in one quadrant. An example of a first-quadrant support sequence is the unit step sequence $u(n_1, n_2)$.

A 2-D LSI system is said to be a *wedge support system* when its impulse response $h(n_1, n_2)$ is a *wedge support sequence*. Consider two lines emanating from the origin. If all the nonzero values in a sequence lie in the region bounded by these two lines, and the angle between the two lines is less than 180°, the sequence is called a wedge support sequence, or a wedge sequence for short. An example of a wedge support sequence $x(n_1, n_2)$ is shown in Figure 1.17.

Quadrant support sequences and wedge support sequences are closely related. A quadrant support sequence is always a wedge support sequence. In addition, it can be shown that any wedge support sequence can always be mapped to a first-quadrant support sequence by a linear mapping of variables without affecting its stability. To illustrate this, consider the wedge support sequence $x(n_1, n_2)$ shown in Figure 1.17. Suppose we obtain a new sequence $y(n_1, n_2)$ from $x(n_1, n_2)$ by the following linear mapping of variables:

$$y(n_1, n_2) = x(m_1, m_2)|_{m_1 = l_1 n_1 + l_2 n_2, m_2 = l_3 n_1 + l_4 n_2} \qquad (1.30)$$

where the integers l_1, l_2, l_3, and l_4 are chosen to be 1, 0, -1 and 1 respectively. The sequence $y(n_1, n_2)$ obtained by using (1.30) is shown in Figure 1.18, and is clearly a first-quadrant support sequence. In addition, the stability of $x(n_1, n_2)$ is equivalent to the stability of $y(n_1, n_2)$, since

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |x(n_1, n_2)| = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |y(n_1, n_2)|.$$

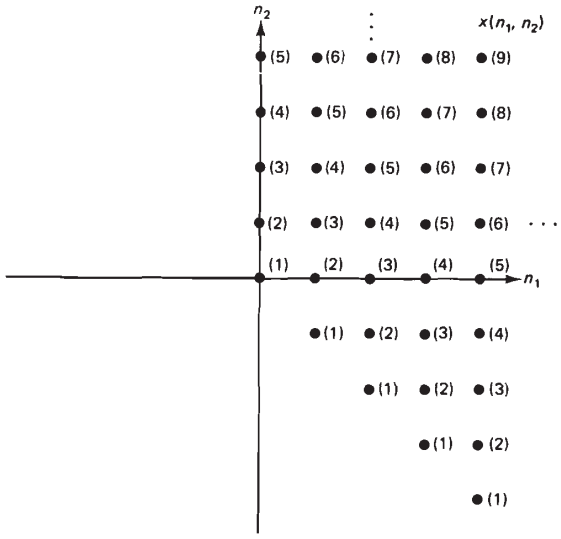


Figure 1.17 Example of a wedge support sequence.

The notion that a wedge support sequence can always be transformed to a first-quadrant support sequence by a simple linear mapping of variables without affecting its stability is very useful in studying the stability of a 2-D system. As we will discuss in Chapter 2, our primary concern in testing the stability of a 2-D system will be limited to a class of systems known as *recursively computable systems*. To test the stability of a recursively computable system, we need to test the stability of a wedge support sequence $h'(n_1, n_2)$. To accomplish this, we will transform $h'(n_1, n_2)$ to a first-quadrant support sequence $h''(n_1, n_2)$ by an appropriate linear mapping of variables and then check the stability of $h''(n_1, n_2)$. This approach exploits the fact that it is much easier to develop stability theorems for first-quadrant support sequences than for wedge support sequences. This will be discussed further in Section 2.3.

1.3 THE FOURIER TRANSFORM

1.3.1 The Fourier Transform Pair

It is a remarkable fact that any stable sequence $x(n_1, n_2)$ can be obtained by appropriately combining complex exponentials of the form $X(\omega_1, \omega_2)e^{j\omega_1 n_1}e^{j\omega_2 n_2}$. The function $X(\omega_1, \omega_2)$, which represents the amplitude associated with the complex exponential $e^{j\omega_1 n_1}e^{j\omega_2 n_2}$, can be obtained from $x(n_1, n_2)$. The relationships between $x(n_1, n_2)$ and $X(\omega_1, \omega_2)$ are given by

Discrete-Space Fourier Transform Pair

$$X(\omega_1, \omega_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2)e^{-j\omega_1 n_1}e^{-j\omega_2 n_2} \tag{1.31a}$$

$$x(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} X(\omega_1, \omega_2)e^{j\omega_1 n_1}e^{j\omega_2 n_2}\, d\omega_1\, d\omega_2 \tag{1.31b}$$

Equation (1.31a) shows how the amplitude $X(\omega_1, \omega_2)$ associated with the exponential $e^{j\omega_1 n_1}e^{j\omega_2 n_2}$ can be determined from $x(n_1, n_2)$. The function $X(\omega_1, \omega_2)$ is called the *discrete-space Fourier transform*, or *Fourier transform* for short, of $x(n_1, n_2)$. Equation (1.31b) shows how complex exponentials $X(\omega_1, \omega_2)e^{j\omega_1 n_1}e^{j\omega_2 n_2}$ are specifically combined to form $x(n_1, n_2)$. The sequence $x(n_1, n_2)$ is called the *inverse discrete-space Fourier transform* or *inverse Fourier transform* of $X(\omega_1, \omega_2)$. The consistency of (1.31a) and (1.31b) can be easily shown by combining them.

From (1.31), it can be seen that $X(\omega_1, \omega_2)$ is in general complex, even though $x(n_1, n_2)$ may be real. It is often convenient to express $X(\omega_1, \omega_2)$ in terms of its magnitude $|X(\omega_1, \omega_2)|$ and phase $\theta_x(\omega_1, \omega_2)$ or in terms of its real part $X_R(\omega_1, \omega_2)$ and imaginary part $X_I(\omega_1, \omega_2)$ as

$$X(\omega_1, \omega_2) = |X(\omega_1, \omega_2)|e^{j\theta_x(\omega_1,\omega_2)} = X_R(\omega_1, \omega_2) + jX_I(\omega_1, \omega_2). \tag{1.32}$$

From (1.31), it can also be seen that $X(\omega_1, \omega_2)$ is a function of continuous variables ω_1 and ω_2, although $x(n_1, n_2)$ is a function of discrete variables n_1 and n_2. In

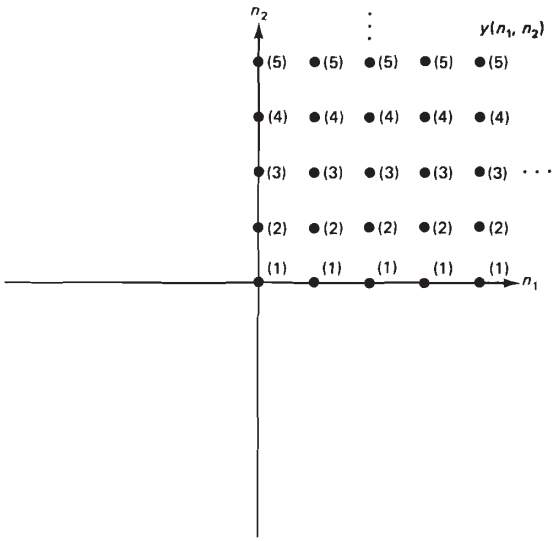


Figure 1.18 First-quadrant support sequence obtained from the wedge support sequence in Figure 1.17 by linear mapping of variables.

addition, $X(\omega_1, \omega_2)$ is always periodic with a period of $2\pi \times 2\pi$; that is, $X(\omega_1, \omega_2) = X(\omega_1 + 2\pi, \omega_2) = X(\omega_1, \omega_2 + 2\pi)$ for all ω_1 and ω_2. We can also show that the Fourier transform converges uniformly for stable sequences. The Fourier transform of $x(n_1, n_2)$ is said to *converge uniformly* when $X(\omega_1, \omega_2)$ is finite and

$$\lim_{N_1\to\infty} \lim_{N_2\to\infty} \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} x(n_1, n_2)e^{-j\omega_1 n_1}e^{-j\omega_2 n_2} = X(\omega_1, \omega_2) \quad \text{for all } \omega_1 \text{ and } \omega_2. \tag{1.33}$$

When the Fourier transform of $x(n_1, n_2)$ converges uniformly, $X(\omega_1, \omega_2)$ is an analytic function and is infinitely differentiable with respect to ω_1 and ω_2.

A sequence $x(n_1, n_2)$ is said to be an eigenfunction of a system T if $T[x(n_1, n_2)] = kx(n_1, n_2)$ for some scalar k. Suppose we use a complex exponential $e^{j\omega_1 n_1}e^{j\omega_2 n_2}$ as an input $x(n_1, n_2)$ to an LSI system with impulse response $h(n_1, n_2)$. The output of the system $y(n_1, n_2)$ can be obtained as

$$\begin{aligned} y(n_1, n_2) &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h(k_1, k_2)x(n_1 - k_1, n_2 - k_2) \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h(k_1, k_2)e^{j\omega_1(n_1-k_1)}e^{j\omega_2(n_2-k_2)} \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h(k_1, k_2)e^{-j\omega_1 k_1}e^{-j\omega_2 k_2}e^{j\omega_1 n_1}e^{j\omega_2 n_2} \\ &= H(\omega_1, \omega_2)e^{j\omega_1 n_1}e^{j\omega_2 n_2}. \end{aligned} \tag{1.34}$$

From (1.34), $e^{j\omega_1 n_1}e^{j\omega_2 n_2}$ is an eigenfunction of any LSI system for which $H(\omega_1, \omega_2)$ is well defined and $H(\omega_1, \omega_2)$ is the Fourier transform of $h(n_1, n_2)$. The function $H(\omega_1, \omega_2)$ is called the *frequency response* of the LSI system. The fact that $H(\omega_1, \omega_2)$ is called the eigenfunction of an LSI system and that $H(\omega_1, \omega_2)$ is the scaling factor by which $e^{j\omega_1 n_1}e^{j\omega_2 n_2}$ is multiplied when it is an input to the LSI system simplifies system analysis for a sinusoidal input. For example, the output of an LSI system with frequency response $H(\omega_1, \omega_2)$ when the input is $\cos(\omega_1' n_1 + \omega_2' n_2)$ can be obtained as follows:

$$\begin{aligned} T[\cos(\omega_1' n_1 + \omega_2' n_2)] &= T\left[\frac{e^{j\omega_1' n_1}e^{j\omega_2' n_2}}{2} + \frac{e^{-j\omega_1' n_1}e^{-j\omega_2' n_2}}{2}\right] \\ &= \tfrac{1}{2}T[e^{j\omega_1' n_1}e^{j\omega_2' n_2}] + \tfrac{1}{2}T[e^{-j\omega_1' n_1}e^{-j\omega_2' n_2}] \qquad (1.35) \\ &= \tfrac{1}{2}H(\omega_1', \omega_2')e^{j\omega_1' n_1}e^{j\omega_2' n_2} + \tfrac{1}{2}H(-\omega_1', -\omega_2')e^{-j\omega_1' n_1}e^{-j\omega_2' n_2}. \end{aligned}$$

1.3.2 Properties

We can derive a number of useful properties from the Fourier transform pair in (1.31). Some of the more important properties, often useful in practice, are listed in Table 1.1. Most are essentially straightforward extensions of 1-D Fourier transform properties. The only exception is Property 4, which applies to separable sequences. If a 2-D sequence $x(n_1, n_2)$ can be written as $x_1(n_1)x_2(n_2)$, then its Fourier transform, $X(\omega_1, \omega_2)$, is given by $X_1(\omega_1)X_2(\omega_2)$, where $X_1(\omega_1)$ and $X_2(\omega_2)$ represent the 1-D Fourier transforms of $x_1(n_1)$ and $x_2(n_2)$, respectively. This property follows directly from the Fourier transform pair of (1.31). Note that this property is quite different from Property 3, the multiplication property. In the multiplication property, both $x(n_1, n_2)$ and $y(n_1, n_2)$ are 2-D sequences. In Property 4, $x_1(n_1)$ and $x_2(n_2)$ are 1-D sequences, and their product $x_1(n_1)x_2(n_2)$ forms a 2-D sequence.

1.3.3 Examples

Example 1

We wish to determine $H(\omega_1, \omega_2)$ for the sequence $h(n_1, n_2)$ shown in Figure 1.19(a). From (1.31),

$$\begin{aligned} H(\omega_1, \omega_2) &= \sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} h(n_1, n_2)e^{-j\omega_1 n_1}e^{-j\omega_2 n_2} \\ &= \tfrac{1}{3} + \tfrac{1}{6}e^{-j\omega_1} + \tfrac{1}{6}e^{-j\omega_2} + \tfrac{1}{6}e^{j\omega_1} + \tfrac{1}{6}e^{j\omega_2} \\ &= \tfrac{1}{3} + \tfrac{1}{3}\cos\omega_1 + \tfrac{1}{3}\cos\omega_2. \end{aligned}$$

The function $H(\omega_1, \omega_2)$ for this example is real and its magnitude is sketched in Figure 1.19(b). If $H(\omega_1, \omega_2)$ in Figure 1.19(b) is the frequency response of an LSI system, the system corresponds to a lowpass filter. The function $|H(\omega_1, \omega_2)|$ shows smaller values in frequency regions away from the origin. A lowpass filter applied to an

TABLE 1.1 PROPERTIES OF THE FOURIER TRANSFORM

$x(n_1, n_2) \longleftrightarrow X(\omega_1, \omega_2)$
$y(n_1, n_2) \longleftrightarrow Y(\omega_1, \omega_2)$

Property 1. *Linearity*
$ax(n_1, n_2) + by(n_1, n_2) \longleftrightarrow aX(\omega_1, \omega_2) + bY(\omega_1, \omega_2)$

Property 2. *Convolution*
$x(n_1, n_2) * y(n_1, n_2) \longleftrightarrow X(\omega_1, \omega_2)Y(\omega_1, \omega_2)$

Property 3. *Multiplication*
$x(n_1, n_2)y(n_1, n_2) \longleftrightarrow X(\omega_1, \omega_2) \circledast Y(\omega_1, \omega_2)$

$$= \frac{1}{(2\pi)^2}\int_{\theta_1=-\pi}^{\pi}\int_{\theta_2=-\pi}^{\pi} X(\theta_1, \theta_2)Y(\omega_1 - \theta_1, \omega_2 - \theta_2)\,d\theta_1\,d\theta_2$$

Property 4. *Separable Sequence*
$x(n_1, n_2) = x_1(n_1)x_2(n_2) \longleftrightarrow X(\omega_1, \omega_2) = X_1(\omega_1)X_2(\omega_2)$

Property 5. *Shift of a Sequence and a Fourier Transform*
(a) $x(n_1 - m_1, n_2 - m_2) \longleftrightarrow X(\omega_1, \omega_2)e^{-j\omega_1 m_1}e^{-j\omega_2 m_2}$
(b) $e^{j\nu_1 n_1}e^{j\nu_2 n_2}x(n_1, n_2) \longleftrightarrow X(\omega_1 - \nu_1, \omega_2 - \nu_2)$

Property 6. *Differentiation*
(a) $-jn_1x(n_1, n_2) \longleftrightarrow \dfrac{\partial X(\omega_1, \omega_2)}{\partial\omega_1}$
(b) $-jn_2x(n_1, n_2) \longleftrightarrow \dfrac{\partial X(\omega_1, \omega_2)}{\partial\omega_2}$

Property 7. *Initial Value and DC Value Theorem*
(a) $x(0, 0) = \dfrac{1}{(2\pi)^2}\displaystyle\int_{\omega_1=-\pi}^{\pi}\int_{\omega_2=-\pi}^{\pi} X(\omega_1, \omega_2)\,d\omega_1\,d\omega_2$

(b) $X(0, 0) = \displaystyle\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} x(n_1, n_2)$

Property 8. *Parseval's Theorem*
(a) $\displaystyle\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} x(n_1, n_2)y^*(n_1, n_2)$

$$= \frac{1}{(2\pi)^2}\int_{\omega_1=-\pi}^{\pi}\int_{\omega_2=-\pi}^{\pi} X(\omega_1, \omega_2)Y^*(\omega_1, \omega_2)\,d\omega_1\,d\omega_2$$

(b) $\displaystyle\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} |x(n_1, n_2)|^2 = \frac{1}{(2\pi)^2}\int_{\omega_1=-\pi}^{\pi}\int_{\omega_2=-\pi}^{\pi} |X(\omega_1, \omega_2)|^2\,d\omega_1\,d\omega_2$

Property 9. *Symmetry Properties*
(a) $x(-n_1, n_2) \longleftrightarrow X(-\omega_1, \omega_2)$
(b) $x(n_1, -n_2) \longleftrightarrow X(\omega_1, -\omega_2)$
(c) $x(-n_1, -n_2) \longleftrightarrow X(-\omega_1, -\omega_2)$
(d) $x^*(n_1, n_2) \longleftrightarrow X^*(-\omega_1, -\omega_2)$
(e) $x(n_1, n_2)$: real $\longleftrightarrow X(\omega_1, \omega_2) = X^*(-\omega_1, -\omega_2)$
$X_R(\omega_1, \omega_2)$, $|X(\omega_1, \omega_2)|$: even (symmetric with respect to the origin)
$X_I(\omega_1, \omega_2)$, $\theta_x(\omega_1, \omega_2)$: odd (antisymmetric with respect to the origin)
(f) $x(n_1, n_2)$: real and even $\longleftrightarrow X(\omega_1, \omega_2)$: real and even
(g) $x(n_1, n_2)$: real and odd $\longleftrightarrow X(\omega_1, \omega_2)$: pure imaginary and odd

Property 10. *Uniform Convergence*
For a stable $x(n_1, n_2)$, the Fourier transform of $x(n_1, n_2)$ uniformly converges.

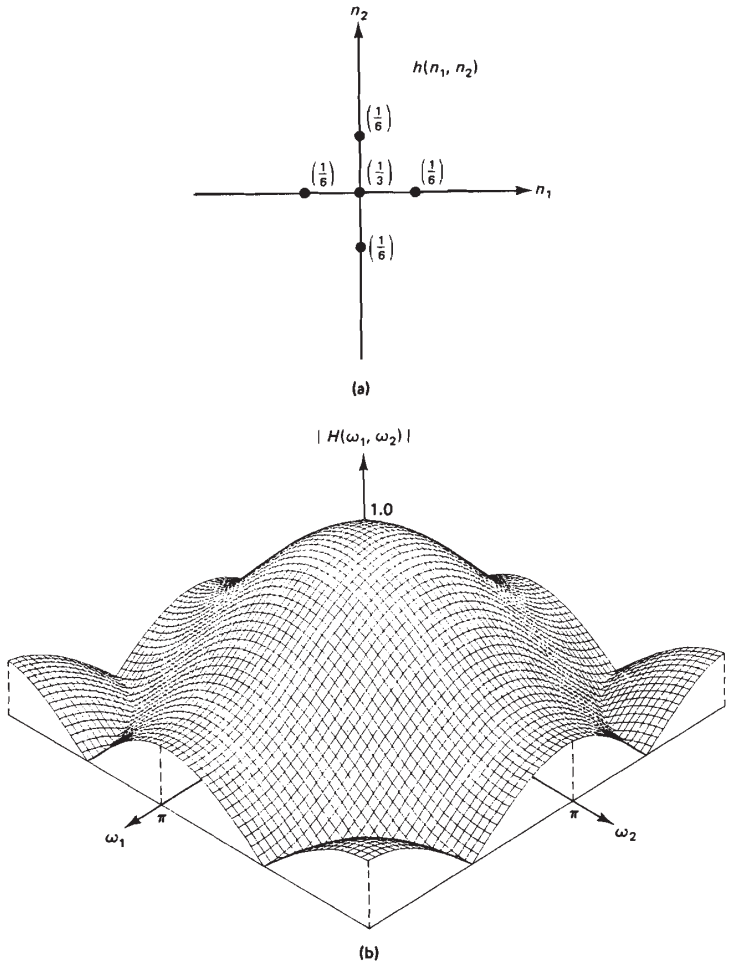


Figure 1.19 (a) 2-D sequence $h(n_1, n_2)$; (b) Fourier transform magnitude $|H(\omega_1, \omega_2)|$ of $h(n_1, n_2)$ in (a).

image blurs the image. The function $H(\omega_1, \omega_2)$ is 1 at $\omega_1 = \omega_2 = 0$, and therefore the average intensity of an image is not affected by the filter. A bright image will remain bright and a dark image will remain dark after processing with the filter. Figure 1.20(a) shows an image of 256 × 256 pixels. Figure 1.20(b) shows the image obtained by processing the image in Figure 1.20(a) with a lowpass filter whose impulse response is given by $h(n_1, n_2)$ in this example.

Figure 1.20 (a) Image of 256 × 256 pixels; (b) image processed by filtering the image in (a) with a lowpass filter whose impulse response is given by $h(n_1, n_2)$ in Figure 1.19 (a).

Example 2

We wish to determine $H(\omega_1, \omega_2)$ for the sequence $h(n_1, n_2)$ shown in Figure 1.21(a). We can use (1.31) to determine $H(\omega_1, \omega_2)$, as in Example 1. Alternatively, we can use Property 4 in Table 1.1. The sequence $h(n_1, n_2)$ can be expressed as $h_1(n_1)h_2(n_2)$, where one possible choice of $h_1(n_1)$ and $h_2(n_2)$ is shown in Figure 1.21(b). Computing the 1-D Fourier transforms $H_1(\omega_1)$ and $H_2(\omega_2)$ and using Property 4 in Table 1.1, we have

$$H(\omega_1, \omega_2) = H_1(\omega_1)H_2(\omega_2) = (3 - 2\cos\omega_1)(3 - 2\cos\omega_2).$$

The function $H(\omega_1, \omega_2)$ is again real, and its magnitude is sketched in Figure 1.21(c). A system whose frequency response is given by the $H(\omega_1, \omega_2)$ above is a highpass filter. The function $|H(\omega_1, \omega_2)|$ has smaller values in frequency regions near the origin. A highpass filter applied to an image tends to accentuate image details or local contrast, and the processed image appears sharper. Figure 1.22(a) shows an original image of 256 × 256 pixels and Figure 1.22(b) shows the highpass filtered image using $h(n_1, n_2)$ in this example. When an image is processed, for instance by highpass filtering, the pixel intensities may no longer be integers between 0 and 255. They may be negative, noninteger, or above 255. In such instances, we typically add a bias and then scale and quantize the processed image so that all the pixel intensities are integers between 0 and 255. It is common practice to choose the bias and scaling factors such that the minimum intensity is mapped to 0 and the maximum intensity is mapped to 255.

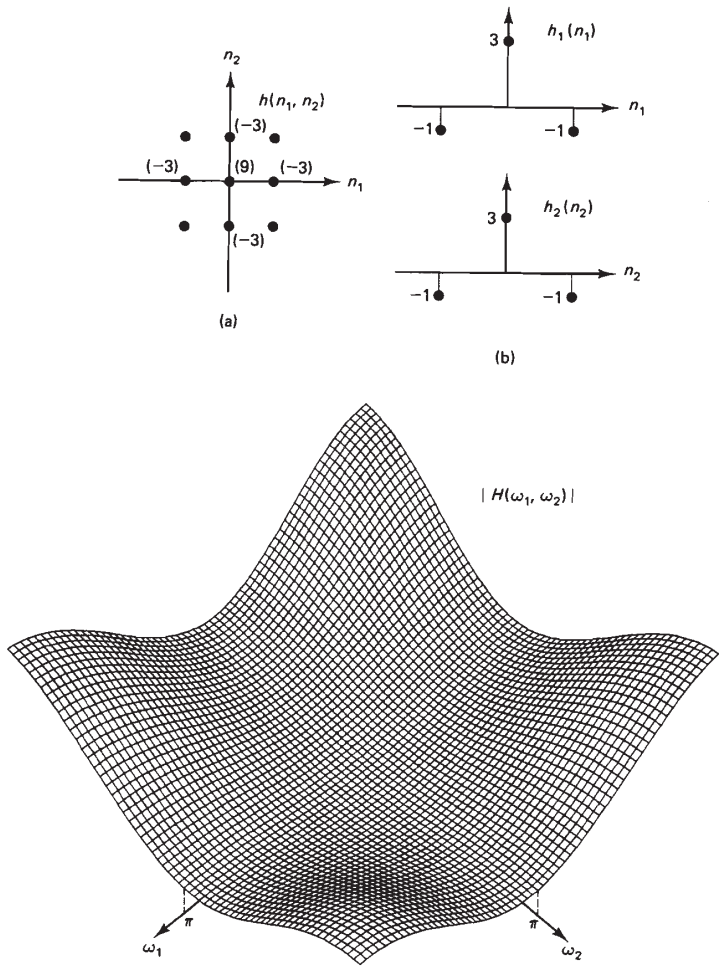
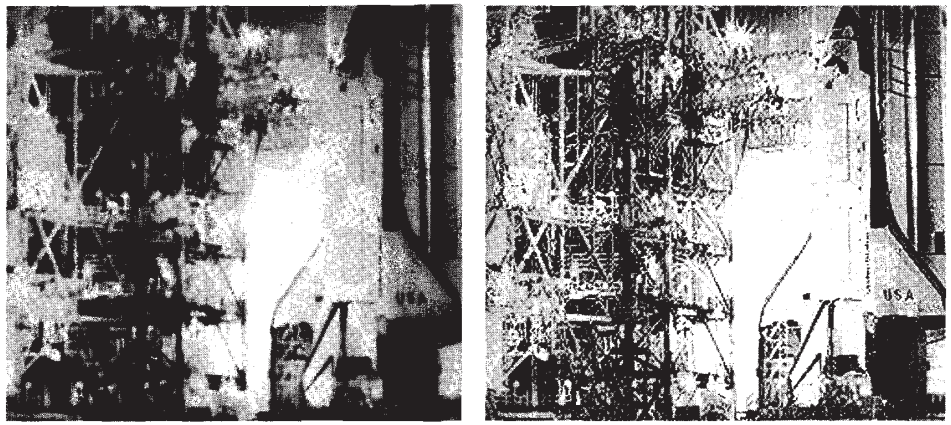


Figure 1.21 (a) 2-D sequence $h(n_1, n_2)$; (b) possible choice of $h_1(n_1)$ and $h_2(n_2)$ where $h(n_1, n_2) = h_1(n_1)h_2(n_2)$; (c) Fourier transform magnitude $|H(\omega_1, \omega_2)|$ of $h(n_1, n_2)$ in (a).

Example 3

We wish to determine $h(n_1, n_2)$ for the Fourier transform $H(\omega_1, \omega_2)$ shown in Figure 1.23. The function $H(\omega_1, \omega_2)$ is given by

$$H(\omega_1, \omega_2) = \begin{cases} 1, & |\omega_1| \le a \quad \text{and} \quad |\omega_2| \le b \text{ (shaded region)} \\ 0, & a < |\omega_1| \le \pi \quad \text{or} \quad b < |\omega_2| \le \pi \text{ (unshaded region).} \end{cases}$$



(a) (b)

Figure 1.22 (a) Image of 256 × 256 pixels; (b) image obtained from filtering the image in (a) with a highpass filter whose impulse response is given by $h(n_1, n_2)$ in Figure 1.21(a).

Since $H(\omega_1, \omega_2)$ is always periodic with a period of 2π along each of the two variables ω_1 and ω_2, $H(\omega_1, \omega_2)$ is shown only for $|\omega_1| \le \pi$ and $|\omega_2| \le \pi$. The function $H(\omega_1, \omega_2)$ can be expressed as $H_1(\omega_1)H_2(\omega_2)$, where one possible choice of $H_1(\omega_1)$ and $H_2(\omega_2)$ is also shown in Figure 1.23. When $H(\omega_1, \omega_2)$ above is the frequency response of a 2-D LSI system, the system is called a *separable ideal lowpass filter*. Computing the 1-D inverse Fourier transforms of $H_1(\omega_1)$ and $H_2(\omega_2)$ and using Property 4 in Table 1.1, we obtain

$$h(n_1, n_2) = h_1(n_1)h_2(n_2) = \frac{\sin an_1}{\pi n_1} \frac{\sin bn_2}{\pi n_2}$$

Example 4

We wish to determine $h(n_1, n_2)$ for the Fourier transform $H(\omega_1, \omega_2)$ shown in Figure 1.24. The function $H(\omega_1, \omega_2)$ is given by

$$H(\omega_1, \omega_2) = \begin{cases} 1, & \sqrt{\omega_1^2 + \omega_2^2} \le \omega_C \quad \text{(shaded region)} \\ 0, & \omega_c < \sqrt{\omega_1^2 + \omega_2^2} \quad \text{and} \quad |\omega_1|, |\omega_2| \le \pi \text{ (unshaded region).} \end{cases}$$

When $H(\omega_1, \omega_2)$ above is the frequency response of a 2-D LSI system, the system is called a *circularly symmetric ideal lowpass filter*, or an *ideal lowpass filter* for short. The inverse Fourier transform of $H(\omega_1, \omega_2)$ in this example requires a fair amount of algebra (see Problem 1.24). The result is

$$h(n_1, n_2) = \frac{\omega_C}{2\pi\sqrt{n_1^2 + n_2^2}} J_1(\omega_C \sqrt{n_1^2 + n_2^2}) \tag{1.36}$$

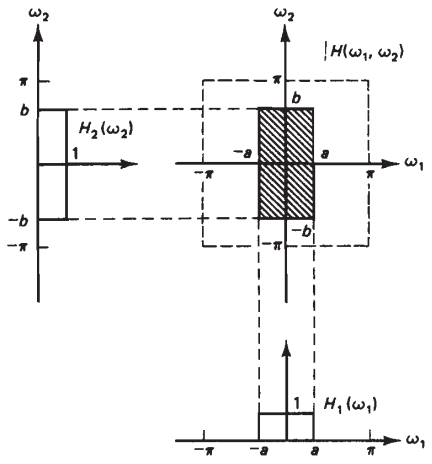


Figure 1.23 Separable Fourier transform $H(\omega_1, \omega_2)$ and one possible choice of $H_1(\omega_1)$ and $H_2(\omega_2)$ such that $H(\omega_1, \omega_2) = H_1(\omega_1)H_2(\omega_2)$. The function $H(\omega_1, \omega_2)$ is 1 in the shaded region and 0 in the unshaded region.

where $J_1(\cdot)$ represents the Bessel function of the first kind and the first order and can be expanded in series form as

$$J_1(x) = \frac{x}{2} - \frac{x^3}{2^3 1!\, 2!} + \frac{x^5}{2^5 2!\, 3!} - \frac{x^7}{2^7 3!\, 4!} + \frac{x^9}{2^9 4!\, 5!} - \cdots . \tag{1.37}$$

This example shows that 2-D Fourier transform or inverse Fourier transform operations can become much more algebraically complex than 1-D Fourier transform or inverse Fourier transform operations, despite the fact that the 2-D Fourier transform pair and many 2-D Fourier transform properties are straightforward extensions of 1-D results. From (1.36), we observe that the impulse response of a 2-D circularly symmetric ideal lowpass filter is also circularly symmetric, that is, it is a function of $n_1^2 + n_2^2$. This is a special case of a more general result. Specifically, if $H(\omega_1, \omega_2)$ is a function of $\omega_1^2 + \omega_2^2$ in the region $\sqrt{\omega_1^2 + \omega_2^2} \leq \pi$ and is a constant outside the region, then the corresponding $h(n_1, n_2)$ is a function of $n_1^2 + n_2^2$. Note, however, that circular symmetry of $h(n_1, n_2)$ does not imply circular symmetry of $H(\omega_1, \omega_2)$. The function $J_1(x)/x$ is sketched in Figure 1.25. The sequence $h(n_1, n_2)$ in (1.36) is sketched in Figure 1.26 for the case $\omega_C = 0.4\pi$.

The impulse responses $h(n_1, n_2)$ obtained from the separable and circularly symmetric ideal lowpass filters in Examples 3 and 4 above are not absolutely summable, and their Fourier transforms do not converge uniformly to $H(\omega_1, \omega_2)$ used to obtain $h(n_1, n_2)$. This is evident from the observation that the two $H(\omega_1, \omega_2)$ contain discontinuities and are not analytic functions. Nevertheless, we will regard them as valid Fourier transform pairs, since they play an important role in digital filtering and the Fourier transforms of the two $h(n_1, n_2)$ converge to $H(\omega_1, \omega_2)$ in the mean square sense.*

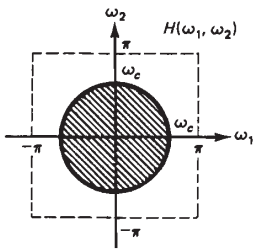


Figure 1.24 Frequency response of a circularly symmetric ideal lowpass filter.

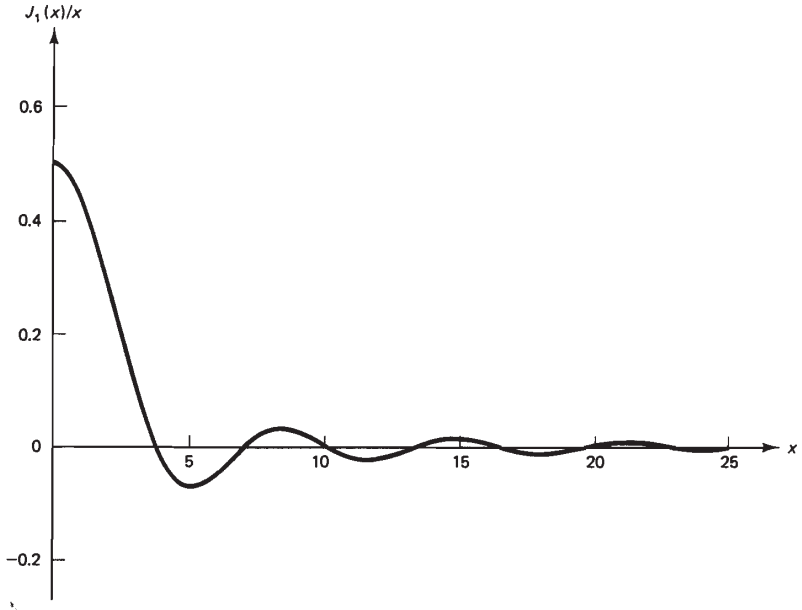


Figure 1.25 Sketch of $\frac{J_1(x)}{x}$, where $J_1(x)$ is the Bessel function of the first kind and first order.

1.4 ADDITIONAL PROPERTIES OF THE FOURIER TRANSFORM

1.4.1 Signal Synthesis and Reconstruction from Phase or Magnitude

The Fourier transform of a sequence is in general complex-valued, and the unique representation of a sequence in the Fourier transform domain requires both the

*The Fourier transform of $h(n_1, n_2)$ is said to converge to $H(\omega_1, \omega_2)$ in the mean square sense when

$$\lim_{N_1 \to \infty} \lim_{N_2 \to \infty} \int_{\omega_1 = -\pi}^{\pi} \int_{\omega_2 = -\pi}^{\pi} \left| \sum_{n_1 = -N_1}^{N_1} \sum_{n_2 = -N_2}^{N_2} h(n_1, n_2) e^{-j\omega_1 n_1} e^{-j\omega_2 n_2} - H(\omega_1, \omega_2) \right|^2 d\omega_1\, d\omega_2 = 0.$$

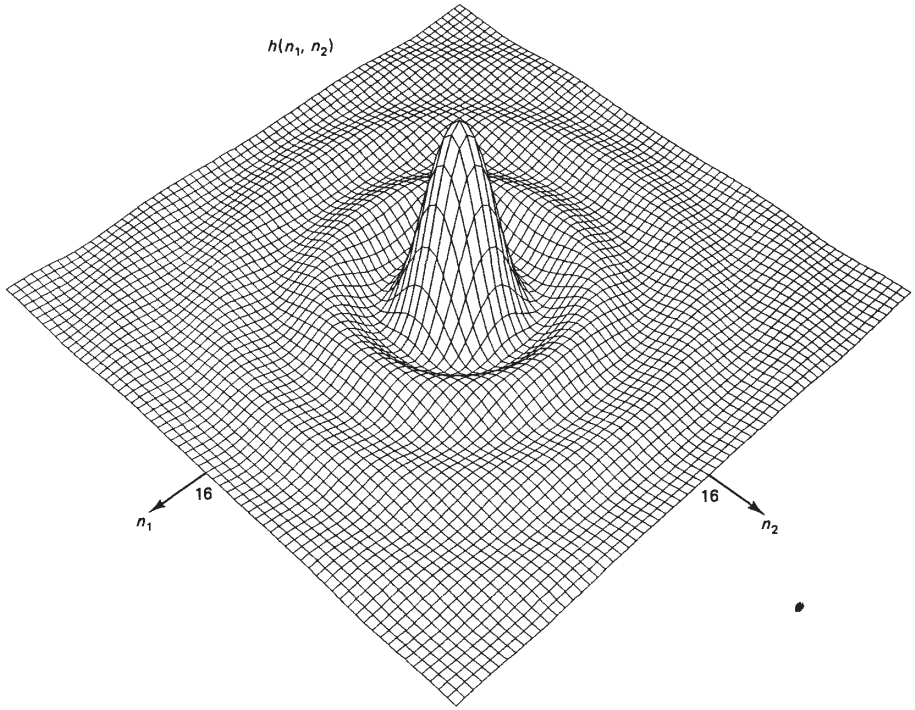


Figure 1.26 Impulse response of a circularly symmetric ideal lowpass filter with $\omega_c = 0.4\pi$ in Equation (1.36). The value at the origin, $h(0, 0)$, is 0.126.

phase and magnitude of the Fourier transform. In various contexts, however, it is often desirable to synthesize or reconstruct a signal from only partial Fourier domain information [Saxton; Ramachandran and Srinivasan]. In this section, we discuss the problem of signal synthesis and reconstruction from the Fourier transform phase alone or from the Fourier transform magnitude alone.

Consider a 2-D sequence $x(n_1, n_2)$ with Fourier transform $X(\omega_1, \omega_2)$ so that

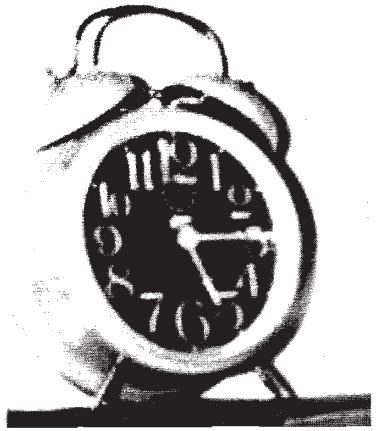
$$X(\omega_1, \omega_2) = F[x(n_1, n_2)] = |X(\omega_1, \omega_2)|e^{j\theta_x(\omega_1, \omega_2)}. \tag{1.38}$$

It has been observed that a straightforward signal synthesis from the Fourier transform phase $\theta_x(\omega_1, \omega_2)$ alone often captures most of the intelligibility of the original signal $x(n_1, n_2)$. A straightforward synthesis from the Fourier transform magnitude $|X(\omega_1, \omega_2)|$ alone, however, does not generally capture the original signal's intelligibility. To illustrate this, we synthesize the phase-only signal $x_p(n_1, n_2)$ and the magnitude-only signal $x_m(n_1, n_2)$ by

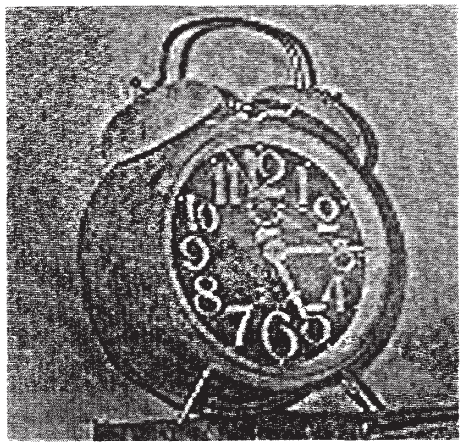
$$x_p(n_1, n_2) = F^{-1}[1e^{j\theta_x(\omega_1, \omega_2)}] \tag{1.39}$$

$$x_m(n_1, n_2) = F^{-1}[|X(\omega_1, \omega_2)|e^{j0}] \tag{1.40}$$

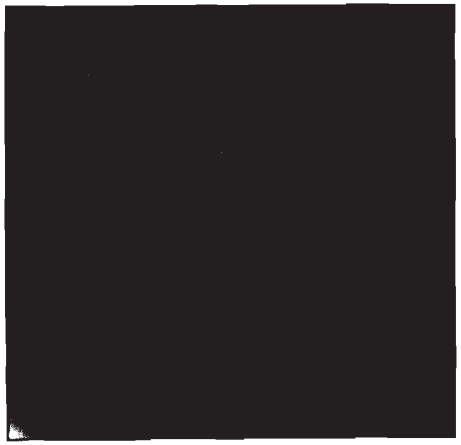
where $F^{-1}[\cdot]$ represents the inverse Fourier transform operation. In phase-only signal synthesis, the correct phase is combined with an arbitrary constant magnitude. In the magnitude-only signal synthesis, the correct magnitude is combined with an arbitrary constant phase. In this synthesis, $x_p(n_1, n_2)$ often preserves the intelligibility of $x(n_1, n_2)$, while $x_m(n_1, n_2)$ does not. An example of this is shown in Figure 1.27. Figure 1.27(a) shows an original image $x(n_1, n_2)$, and Figures 1.27(b) and (c) show $x_p(n_1, n_2)$ and $x_m(n_1, n_2)$, respectively.



(a)



(b)



(c)

Figure 1.27 Example of phase-only and magnitude-only synthesis. (a) Original image of 128 × 128 pixels; (b) result of phase-only synthesis; (c) result of magnitude-only synthesis.

An experiment which more dramatically illustrates the observation that phase-only signal synthesis captures more of the signal intelligibility than magnitude-only synthesis can be performed as follows. Consider two images $x(n_1, n_2)$ and $y(n_1, n_2)$. From these two images, we synthesize two other images $f(n_1, n_2)$ and $g(n_1, n_2)$ by

$$f(n_1, n_2) = F^{-1}[|Y(\omega_1, \omega_2)|e^{j\theta_x(\omega_1,\omega_2)}] \tag{1.41}$$

$$g(n_1, n_2) = F^{-1}[|X(\omega_1, \omega_2)|e^{j\theta_y(\omega_1,\omega_2)}]. \tag{1.42}$$

In this experiment, $f(n_1, n_2)$ captures the intelligibility of $x(n_1, n_2)$, while $g(n_1, n_2)$ captures the intelligibility of $y(n_1, n_2)$. An example is shown in Figure 1.28. Figures 1.28(a) and (b) show the two images $x(n_1, n_2)$ and $y(n_1, n_2)$ and Figures 1.28(c) and (d) show the two images $f(n_1, n_2)$ and $g(n_1, n_2)$.

The high intelligibility of phase-only synthesis raises the possibility of exactly reconstructing a signal $x(n_1, n_2)$ from its Fourier transform phase $\theta_x(\omega_1, \omega_2)$. This is known as the *magnitude-retrieval* problem. In fact, it has been shown [Hayes] that a sequence $x(n_1, n_2)$ is uniquely specified within a scale factor if $x(n_1, n_2)$ is real and has finite extent, and if its Fourier transform cannot be factored as a product of lower-order polynomials in $e^{j\omega_1}$ and $e^{j\omega_2}$. Typical images $x(n_1, n_2)$ are real and have finite regions of support. In addition, the fundamental theorem of algebra does not apply to 2-D polynomials, and their Fourier transforms cannot generally be factored as products of lower-order polynomials in $e^{j\omega_1}$ and $e^{j\omega_2}$. Typical images, then, are uniquely specified within a scale factor by the Fourier transform phase alone.

Two approaches to reconstructing a sequence from its Fourier transform phase alone have been considered. The first approach leads to a closed-form solution and the second to an iterative procedure. In the first approach, $\tan \theta_x(\omega_1, \omega_2)$ is expressed as

$$\tan \theta_x(\omega_1, \omega_2) = \frac{X_I(\omega_1, \omega_2)}{X_R(\omega_1, \omega_2)} = -\frac{\sum\limits_{(n_1,n_2)\in R_x}\sum x(n_1, n_2) \sin(\omega_1 n_1 + \omega_2 n_2)}{\sum\limits_{(n_1,n_2)\in R_x}\sum x(n_1, n_2) \cos(\omega_1 n_1 + \omega_2 n_2)} \tag{1.43}$$

where R_x is the region of support of $x(n_1, n_2)$. Rewriting (1.43), we have

$$\sum_{(n_1,n_2)\in R_x}\sum x(n_1, n_2) \cos(\omega_1 n_1 + \omega_2 n_2) \tan \theta_x(\omega_1, \omega_2)$$

$$= -\sum_{(n_1,n_2)\in R_x}\sum x(n_1, n_2) \sin(\omega_1 n_1 + \omega_2 n_2). \tag{1.44}$$

Equation (1.44) is a linear equation for the unknown values in $x(n_1, n_2)$ for each frequency (ω_1, ω_2). If there are N^2 unknown values in $x(n_1, n_2)$, we can obtain a set of N^2 linear equations for $x(n_1, n_2)$ by sampling (ω_1, ω_2) at N^2 points. If the frequencies are sampled at distinctly different points, noting that $\theta_x(\omega_1, \omega_2)$ is an odd function and is periodic with a period of $2\pi \times 2\pi$, the solution to the set of N^2 linear equations can be shown to be $kx(n_1, n_2)$, where k is an arbitrary real scaling factor. An example of signal reconstruction from phase using (1.44) is shown in Figure 1.29. Figure 1.29(a) shows an image of 12 × 12 pixels, and Figure

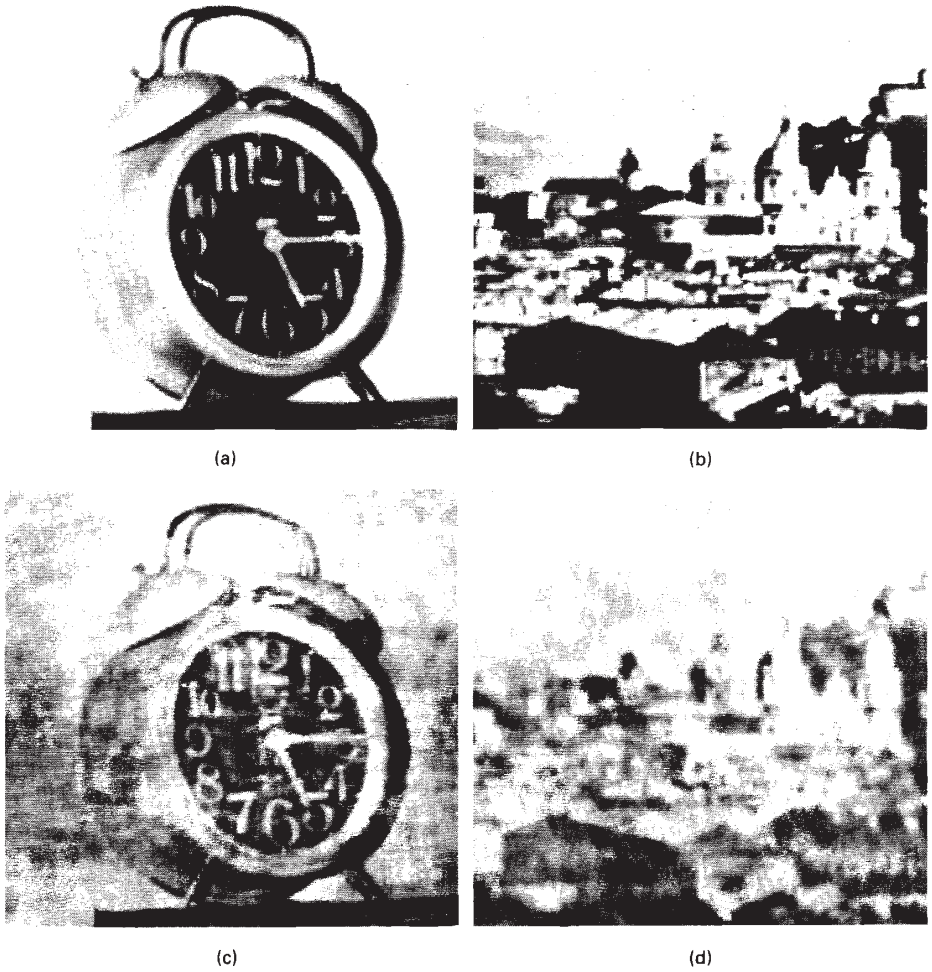


Figure 1.28 Example of image synthesis from the Fourier transform phase of one image and the Fourier transform magnitude of another image. (a) Original image $x(n_1, n_2)$ of 128 × 128 pixels; (b) original image $y(n_1, n_2)$ of 128 × 128 pixels; (c) result of synthesis from $\theta_x(\omega_1, \omega_2)$ and $|Y(\omega_1, \omega_2)|$; (d) result of synthesis from $\theta_y(\omega_1, \omega_2)$ and $|X(\omega_1, \omega_2)|$.

1.29(b) shows the reconstruction. The scaling factor of the reconstructed sequence in the figure is chosen such that the reconstruction will match the original sequence.

The reconstruction algorithm discussed above is reasonable for a small size image, but is not practical for an image of typical size. For example, reconstructing an image of 512 × 512 pixels using (1.44) requires the solution of approximately

(a) (b)

Figure 1.29 Example of phase-only reconstruction by a closed-form algorithm. (a) Original image of 12 × 12 pixels. One pixel in the image is a large square block; (b) phase-only reconstruction of the image in (a) by solving a set of linear equations in (1.44).

a quarter of a million linear equations. An alternate approach is to recognize that the solution to the phase-only reconstruction problem must satisfy constraints in both the spatial and frequency domains. Specifically, the solution must be real, must be zero outside the known region of support, and must have a nonfactorable Fourier transform. In addition, the phase of the Fourier transform of the solution must be the same as the $\theta_x(\omega_1, \omega_2)$ given. A useful approach to solving such a problem is an iterative procedure, in which we impose the spatial and frequency domain constraints separately in each domain. An iterative procedure for the phase-only reconstruction is shown in Figure 1.30. In the procedure, we begin with an initial estimate of the signal. This can be any real sequence with the same region of support as $x(n_1, n_2)$. We next compute its Fourier transform. We then replace the Fourier transform with the given $\theta_x(\omega_1, \omega_2)$. The Fourier transform magnitude is not affected. We then compute the inverse Fourier transform of the modified Fourier transform. Due to the modification in the Fourier transform domain, the sequence is no longer zero outside the known region of support of $x(n_1, n_2)$. We now impose the spatial domain constraint by setting the sequence to zero outside the known region of support. The resulting sequence is a new estimate of the solution. This completes one iteration in the iterative procedure. When the initial estimate of the sequence chosen is real, the constraint that the solution is real will automatically be satisfied. The above algorithm can be shown to converge to the desired solution [Tom, et al.]. An example of signal reconstruction from phase using the iterative procedure in Figure 1.30 is shown in Figure 1.31. Figure 1.31(a) shows an original image of 128 × 128 pixels. Figures 1.31(b), (c), and (d) show the results of the iterative procedure after one iteration [phase-

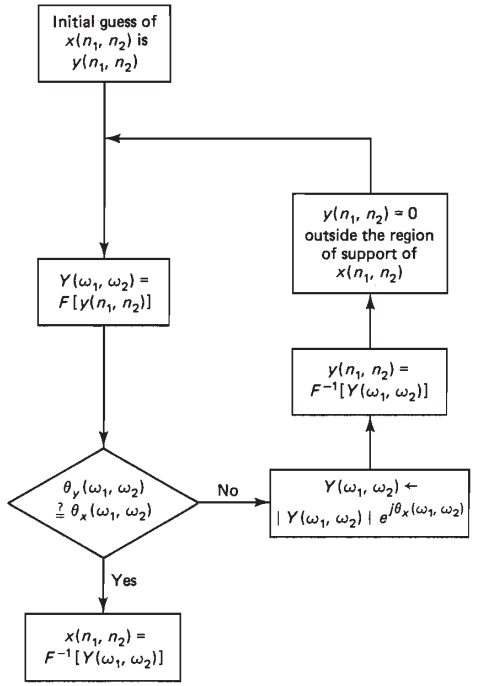
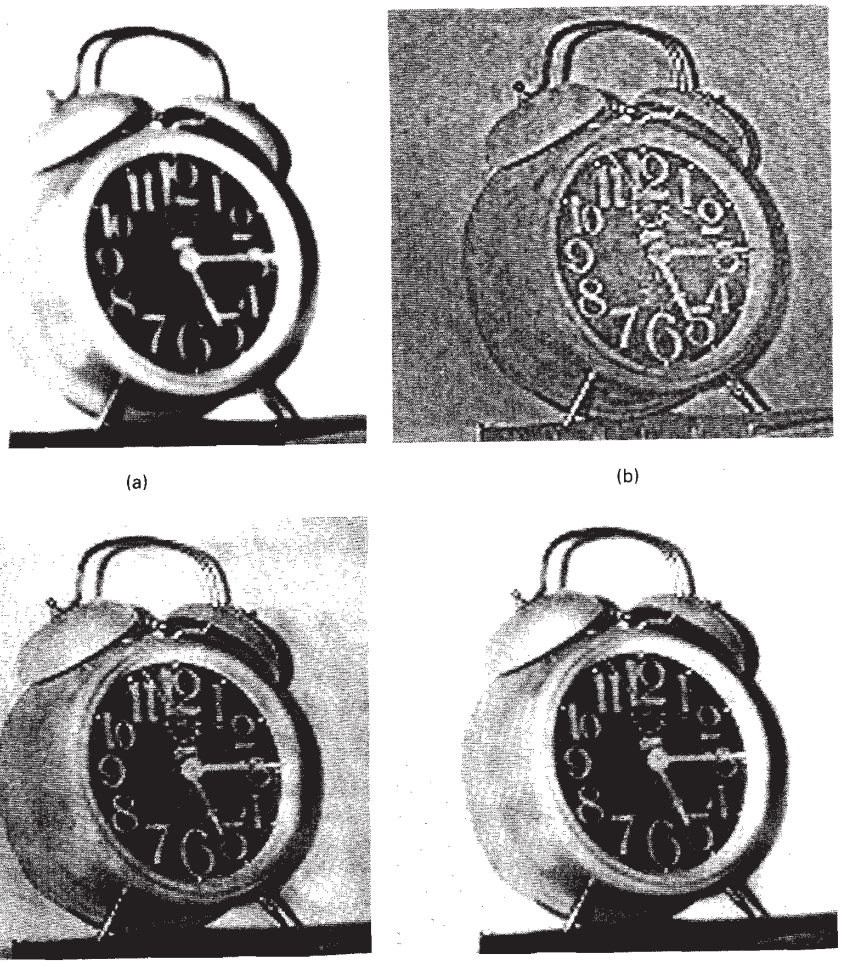


Figure 1.30 Iterative procedure for phase-only reconstruction of $x(n_1, n_2)$ from its phase $\theta_x(\omega_1, \omega_2)$.

only synthesis of (1.39)], 10 iterations, and 50 iterations. The initial estimate used is $\delta(n_1, n_2)$.

Although the magnitude-only synthesis of (1.40) does not capture the intelligibility of typical signals, almost all typical images are also uniquely specified by the Fourier transform magnitude. Specifically, if $x(n_1, n_2)$ is real, has finite extent, and has a nonfactorable Fourier transform, then $x(n_1, n_2)$ is uniquely specified by its Fourier transform magnitude $|X(\omega_1, \omega_2)|$ within a sign factor, translation, and rotation by 180 degrees [Bruck and Sodin, Hayes]. This raises the possibility of exactly reconstructing $x(n_1, n_2)$ from $|X(\omega_1, \omega_2)|$ within a sign factor, translation and rotation by 180 degrees. This is known in the literature as the *phase-retrieval* problem, and has many more potential applications than the phase-only reconstruction problem. Unfortunately, none of the algorithms developed to date are as straightforward or well-behaved as the algorithms developed for the phase-only reconstruction problem. It is possible to derive a closed-form algorithm or a set of linear equations that can be used in solving for $x(n_1, n_2)$ from $|X(\omega_1, \omega_2)|$, but their derivation is quite involved [Izraelevitz and Lim, Lane, et al.]. In addition, the closed-form solution is not practical for an image of reasonable size due to the large number of linear equations that must be solved. It is also possible to derive



(a) (b) (c) (d)

Figure 1.31 Example of phase-only reconstruction by an iterative algorithm. (a) Original image of 128 × 128 pixels; (b) result of phase-only reconstruction of the image in (a) after one iteration of the iterative procedure in Figure 1.30. Since the initial estimate used is $\delta(n_1, n_2)$, this is the same as the phase-only synthesis of (1.39); (c) result after 10 iterations; (d) result after 50 iterations.

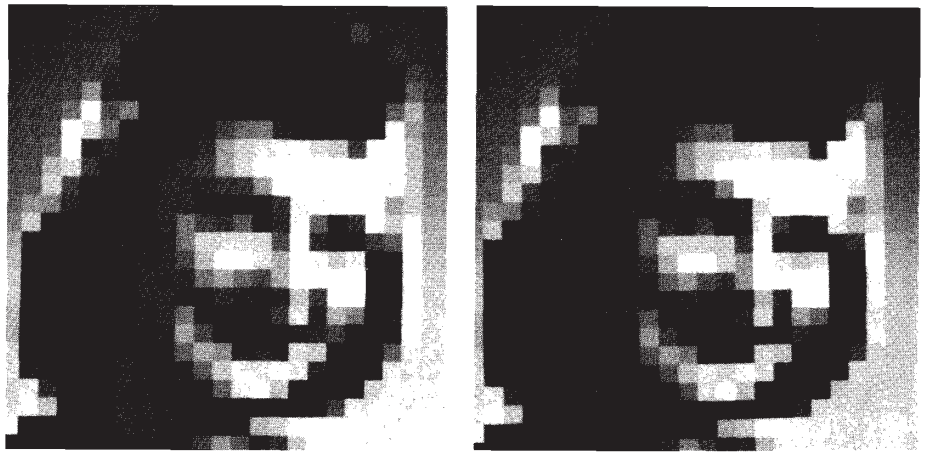
an iterative procedure similar to that in Figure 1.30, which was developed for the phase-only reconstruction. The only modification required is to replace the Fourier transform magnitude with the given $|X(\omega_1, \omega_2)|$ rather than to replace the Fourier transform phase with the given $\theta_x(\omega_1, \omega_2)$ when the frequency domain constraints

are imposed. The algorithm has been observed to converge to the desired solution when the initial estimate used is quite accurate or the signal $x(n_1, n_2)$ has a special characteristic such as a triangular region of support. The magnitude-only reconstruction problem specifies $x(n_1, n_2)$ within a sign factor, translation, and rotation by 180°, and, therefore, more than one solution is possible. Imposing an initial estimate sufficiently close to a possible solution or imposing additional constraints such as a triangular region of support appear to prevent the iterative procedure from wandering around from one possible solution to another. In general, however, the algorithm does not converge to the desired solution. Figure 1.32 shows an example of signal reconstruction from the magnitude using a closed-form algorithm [Izraelevitz and Lim]. Figures 1.32(a) and (b) show the original and the reconstruction respectively. Developing a practical procedure that can be used to reconstruct $x(n_1, n_2)$ from $|X(\omega_1, \omega_2)|$ remains a problem for further research.

In addition to the phase-only and magnitude-only signal synthesis and reconstruction problems discussed above, a variety of results on the synthesis and reconstruction of a signal from other partial Fourier transform information—for instance, one bit of Fourier transform phase or signed Fourier transform magnitude—have been reported [Oppenheim, et al. (1983)].

1.4.2 The Fourier Transform of Typical Images

The Fourier transforms of typical images have been observed to have most of their energy concentrated in a small region in the frequency domain, near the origin

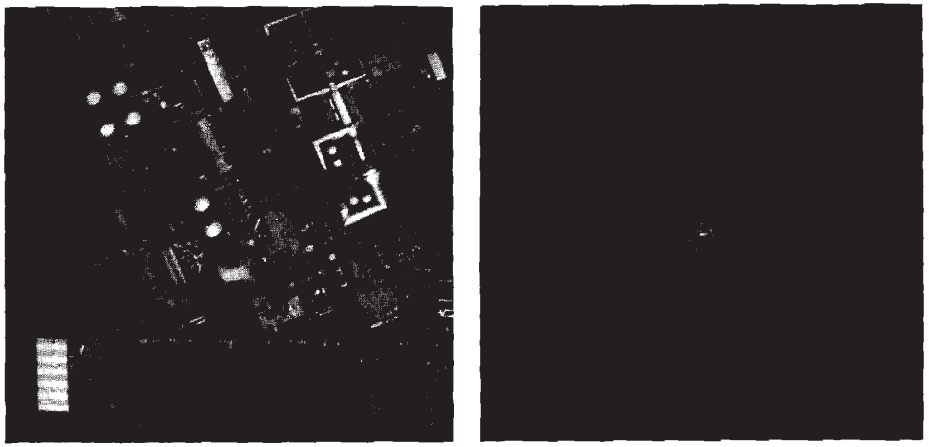


(a) (b)

Figure 1.32 Example of magnitude-only reconstruction by a closed-form algorithm. (a) Original image of 24 × 24 pixels; (b) result of magnitude-only reconstruction of the image in (a) using a closed-form algorithm. After [Izraelevitz and Lim.]

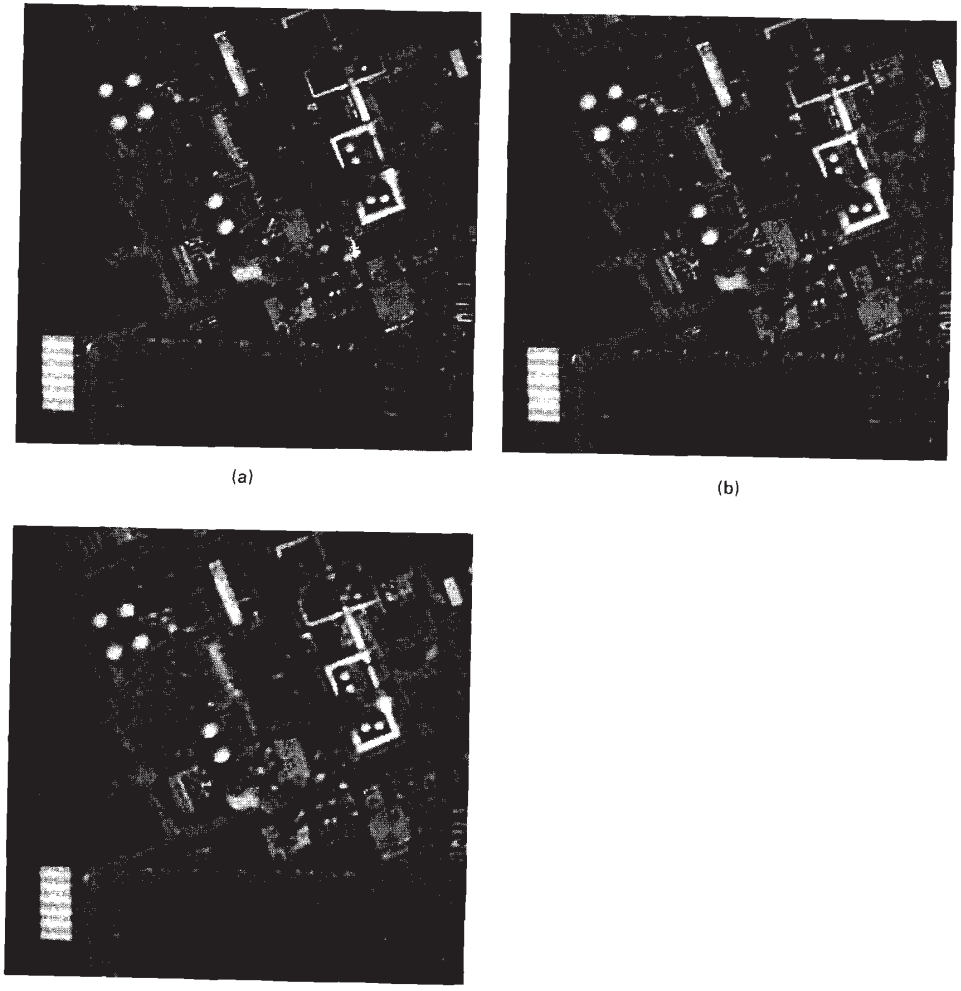
and along the ω_1 and ω_2 axes. One reason for the energy concentration near the origin is that images typically have large regions where the intensities change slowly. Furthermore, sharp discontinuities such as edges contribute to low-frequency as well as high-frequency components. The energy concentration along the ω_1 and ω_2 axes is in part due to a rectangular window used to obtain a finite-extent image. The rectangular window creates artificial sharp discontinuities at the four boundaries. Discontinuities at the top and bottom of the image contribute energy along the ω_2 axis and discontinuities at the two sides contribute energy along the ω_1 axis. Figure 1.33 illustrates this property. Figure 1.33(a) shows an original image of 512 × 512 pixels, and Figure 1.33(b) shows $|X(\omega_1, \omega_2)|^{1/4}$ of the image in Figure 1.33(a). The operation $(\cdot)^{1/4}$ has the effect of compressing large amplitudes while expanding small amplitudes, and therefore shows $|X(\omega_1, \omega_2)|$ more clearly for higher-frequency regions. In this particular example, energy concentration along approximately diagonal directions is also visible. This is because of the many sharp discontinuities in the image along approximately diagonal directions. This example shows that most of the energy is concentrated in a small region in the frequency plane.

Since most of the signal energy is concentrated in a small frequency region, an image can be reconstructed without significant loss of quality and intelligibility from a small fraction of the transform coefficients. Figure 1.34 shows images that were obtained by inverse Fourier transforming the Fourier transform of the image in Figure 1.33(a) after setting most of the Fourier transform coefficients to zero. The percentages of the Fourier transform coefficients that have been preserved in



(a) (b)

Figure 1.33 Example of the Fourier transform magnitude of an image. (a) Original image $x(n_1, n_2)$ of 512 × 512 pixels; (b) $|X(\omega_1, \omega_2)|^{1/4}$, scaled such that the smallest value maps to the darkest level and the largest value maps to the brightest level. The operation $(\cdot)^{1/4}$ has the effect of compressing large amplitudes while expanding small amplitudes, and therefore shows $|X(\omega_1, \omega_2)|$ more clearly for higher-frequency regions.



(a) (b)

(c)

Figure 1.34 Illustration of energy concentration in the Fourier transform domain for a typical image. (a) Image obtained by preserving 12.4% of Fourier transform coefficients of the image in Figure 1.33(a). All other coefficients are set to 0. (b) Same as (a) with 10% of Fourier transform coefficients preserved; (c) same as (a) with 4.8% of Fourier transform coefficients preserved.

Figures 1.34(a), (b), and (c) are 12.4%, 10%, and 4.8%, respectively. The frequency region that was preserved in each of the three cases has the shape (shaded region) shown in Figure 1.35.

The notion that an image with good quality and intelligibility can be reconstructed from a small fraction of transform coefficients for some transforms, for instance the Fourier transform, is the basis of a class of image coding systems known collectively as *transform coding techniques*. One objective of image coding is to represent an image with as few bits as possible while preserving a certain level of image quality and intelligibility. Reduction of transmission channel or storage requirements is a typical application of image coding. In transform coding, the transform coefficients of an image rather than its intensities are coded. Since only a small fraction of the transform coefficients need to be coded in typical applications, the bit rate required in transform coding is often significantly lower than image coding techniques that attempt to code image intensities. The topic of image coding is discussed in Chapter 10.

1.4.3 The Projection-Slice Theorem

Another property of the Fourier transform is the projection-slice theorem, which is the mathematical basis of computed tomography (CT). Computed tomography has a number of applications, including the medical application of reconstructing cross sections of a human body from x-ray images. The impact of computed tomography on medicine requires no elaboration.

Consider a 2-D analog function $f_c(t_1, t_2)$ where t_1 and t_2 are continuous variables. The subscript c denotes that the signal is a function of a continuous variable or variables. The analog Fourier transform $F_c(\Omega_1, \Omega_2)$ is related to $f_c(t_1, t_2)$ by

$$F_c(\Omega_1, \Omega_2) = \int_{t_1=-\infty}^{\infty} \int_{t_2=-\infty}^{\infty} f_c(t_1, t_2) e^{-j\Omega_1 t_1} e^{-j\Omega_2 t_2} \, dt_1 \, dt_2 \tag{1.45a}$$

$$f_c(t_1, t_2) = \frac{1}{(2\pi)^2} \int_{\Omega_1=-\infty}^{\infty} \int_{\Omega_2=-\infty}^{\infty} F_c(\Omega_1, \Omega_2) e^{j\Omega_1 t_1} e^{j\Omega_2 t_2} \, d\Omega_1 \, d\Omega_2. \tag{1.45b}$$

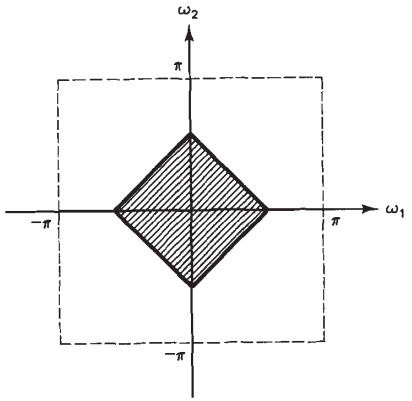


Figure 1.35 Shape of the frequency region where Fourier transform coefficients are preserved in obtaining the images in Figure 1.34.

Let us integrate $f_c(t_1, t_2)$ along the parallel rays shown in Figure 1.36. The angle that the rays make with the t_2-axis is denoted by θ. The result of the integration at a given θ is a 1-D function, and we denote it by $p_\theta(t)$. In this figure, $p_\theta(0)$ is the result of integrating $f_c(t_1, t_2)$ along the ray passing through the origin. The function $p_\theta(t)$, which is called the *projection* of $f_c(t_1, t_2)$ at angle θ or *Radon transform* of $f_c(t_1, t_2)$, can be expressed in terms of $f_c(t_1, t_2)$ by

$$p_\theta(t) = \int_{u=-\infty}^{\infty} f_c(t_1, t_2) \Big|_{t_1 = t\cos\theta - u\sin\theta, t_2 = t\sin\theta + u\cos\theta} \, du. \tag{1.46}$$

Equation (1.46) arises naturally from the analysis of an x-ray image. Consider a 2-D object (a slice of a 3-D object, for example) through which we radiate a monoenergetic x-ray beam, as shown in Figure 1.36. On the basis of the Lambert-Beer law, which describes the attenuation of the x-ray beam as it passes through an object, and of a model of a typical film used to record the output x-ray beam, the image recorded on film can be modeled by $p_\theta(t)$ in (1.46), where $f_c(t_1, t_2)$ is the attenuation coefficient of the 2-D object as a function of two spatial variables t_1 and t_2. The function $f_c(t_1, t_2)$ depends on the material that composes the 2-D object at the spatial position (t_1, t_2). To the extent that the attenuation coefficients of different types of material such as human tissue and bone differ, $f_c(t_1, t_2)$ can be used to determine the types of material. Reconstructing $f_c(t_1, t_2)$ from the recorded $p_\theta(t)$ is, therefore, of considerable interest.

Consider the 1-D analog Fourier transform of $p_\theta(t)$ with respect to the variable t and denote it by $P_\theta(\Omega)$, so that

$$P_\theta(\Omega) = \int_{t=-\infty}^{\infty} p_\theta(t) e^{-j\Omega t} \, dt. \tag{1.47}$$

It can be shown (see Problem 1.33) that there is a simple relationship between $P_\theta(\Omega)$ and $F_c(\Omega_1, \Omega_2)$, given by

$$P_\theta(\Omega) = F_c(\Omega_1, \Omega_2) \Big|_{\Omega_1 = \Omega\cos\theta, \Omega_2 = \Omega\sin\theta} = F_c(\Omega\cos\theta, \Omega\sin\theta). \tag{1.48}$$

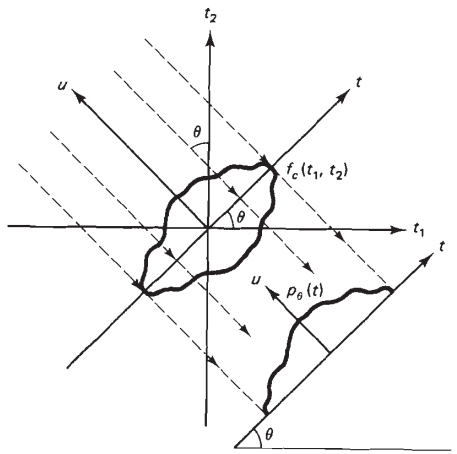


Figure 1.36 Projection of $f_c(t_1, t_2)$ at angle θ.

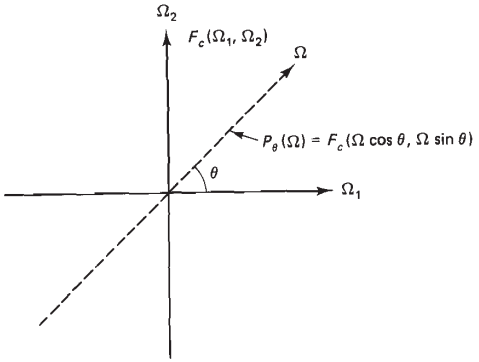


Figure 1.37 Projection slice theorem. $P_\theta(\Omega)$ is the 2-D Fourier transform $F_c(\Omega_1, \Omega_2)$ evaluated along the dotted line.

Expressed graphically, (1.48) states that the 1-D Fourier transform of the projection $p_\theta(t)$ is $F_c(\Omega_1, \Omega_2)$ evaluated along the slice that passes through the origin and makes an angle of θ with the Ω_1 axis, as shown in Figure 1.37. The relationship in (1.48) is called the *projection-slice theorem*.

The projection-slice theorem of (1.48) can be used in developing methods to reconstruct the 2-D function $f_c(t_1, t_2)$ from its projections $p_\theta(t)$. One method is to compute the inverse Fourier transform of $F_c(\Omega_1, \Omega_2)$ obtained from $p_\theta(t)$. Specifically, if we compute the 1-D Fourier transform of $p_\theta(t)$ with respect to t for all $0 \le \theta < \pi$, we will have complete information on $F_c(\Omega_1, \Omega_2)$. In practice, of course, $p_\theta(t)$ cannot be measured for all possible angles $0 \le \theta < \pi$, so $F_c(\Omega_1, \Omega_2)$ must be estimated by interpolating known slices of $F_c(\Omega_1, \Omega_2)$.

Another reconstruction method, known as the *filtered back-projection method*, is more popular in practice and can be derived from (1.45b) and (1.48). It can be shown [Kak] that

$$f_c(t_1, t_2) = \frac{1}{2\pi} \int_{\theta=0}^{\pi} q_\theta(t)\big|_{t=t_1\cos\theta + t_2 \sin\theta} d\theta = \frac{1}{2\pi} \int_{\theta=0}^{\pi} q_\theta(t_1 \cos\theta + t_2 \sin\theta)\, d\theta \qquad (1.49)$$

where $q_\theta(t)$ is related to $p_\theta(t)$ by

$$q_\theta(t) = p_\theta(t) * h(t) = \int_{\tau=-\infty}^{\infty} p_\theta(\tau) h(t - \tau)\, d\tau. \qquad (1.50)$$

The function $h(t)$, which can be viewed as the impulse response of a filter, is given by

$$h(t) = \frac{1}{2\pi} \int_{\Omega=-\Omega_c}^{\Omega_c} |\Omega| e^{j\Omega t}\, d\Omega \qquad (1.51)$$

where Ω_c is the frequency above which the energy in any projection $p_\theta(t)$ can be assumed to be zero. From (1.49) and (1.50), we can see that one method of reconstructing $f_c(t_1, t_2)$ from $p_\theta(t)$ is to first compute $q_\theta(t)$ by filtering (convolving) $p_\theta(t)$ with $h(t)$ and then to determine $f_c(t_1, t_2)$ from $q_\theta(t)$ by using (1.49). The

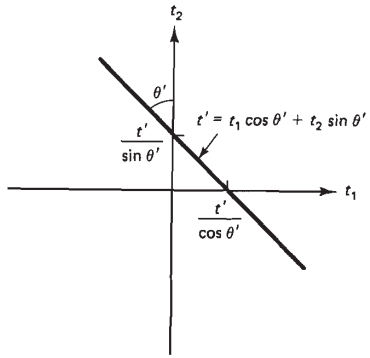


Figure 1.38 Values of (t_1, t_2) for which $f_c(t_1, t_2)$ is affected by $q_{\theta'}(t')$ in the filtered back-projection reconstruction method. They can be described by $t' = t_1 \cos\theta' + t_2 \sin\theta'$.

process of determining $f_c(t_1, t_2)$ from $q_\theta(t)$ using (1.49) can be viewed as a back-projection. Consider a particular θ and t, say θ' and t'. From (1.49), the values of (t_1, t_2) for which $f_c(t_1, t_2)$ is affected by $q_{\theta'}(t')$ are given by $t' = t_1 \cos\theta' + t_2 \sin\theta'$. These values are shown by the straight line in Figure 1.38. Furthermore, the contribution that $q_{\theta'}(t')$ makes to $f_c(t_1, t_2)$ is equal at all points along this line. In essence, $q_{\theta'}(t')$ is back-projected in the (t_1, t_2) domain. This back-projection takes place for all values of t' and is integrated over all values of θ'. Since $q_\theta(t)$ is a filtered version of $p_\theta(t)$, this technique is called the filtered back-projection method. In practice, $p_\theta(t)$ is not available for all values of θ. As a result, $q_\theta(t)$ must be interpolated from the known slices of $q_\theta(t)$.

In addition to the interpolation involved in both the direct Fourier transform method and the filtered back-projection method, a number of practical issues arise in reconstructing $f_c(t_1, t_2)$ from $p_\theta(t)$. For example, the Fourier transform, inverse Fourier transform, filtering, and integration require a discretization of the problem, which raises a variety of important issues, including sampling and aliasing. In practice, the measured function $p_\theta(t)$ may be only an approximate projection of $f_c(t_1, t_2)$. In addition, the measured data may not have been obtained from parallel-beam projection, but instead from fan-beam projection, in which case a different set of equations governs. More details on these and other theoretical and practical issues can be found in [Scudder, Kak]. We will close this section with an example in which a cross section of a human head was reconstructed from its x-ray projections. Figure 1.39 shows the reconstruction by the back-projection method.

1.5 DIGITAL PROCESSING OF ANALOG SIGNALS

Most signals that occur in practice are analog. In this section, we discuss digital processing of analog signals. Since the issues that arise in digital processing of analog signals are essentially the same in both the 1-D and 2-D cases, we will briefly summarize the 2-D results.

Consider an analog 2-D signal $x_c(t_1, t_2)$. We'll denote its analog Fourier

Figure 1.39 Cross section of a human head reconstructed from its projections by the filtered back-projection method. Courtesy of Tamas Sandor.

transform by $X_c(\Omega_1, \Omega_2)$. Suppose we obtain a discrete-space signal $x(n_1, n_2)$ by sampling the analog signal $x_c(t_1, t_2)$ with sampling period (T_1, T_2) as follows:

$$x(n_1, n_2) = x_c(t_1, t_2)\big|_{t_1 = n_1 T_1, t_2 = n_2 T_2}. \tag{1.52}$$

Equation (1.52) represents the input-output relationship of an ideal analog-to-digital (A/D) converter. The relationship between $X(\omega_1, \omega_2)$, the discrete-space Fourier transform of $x(n_1, n_2)$, and $X_c(\Omega_1, \Omega_2)$, the continuous-space Fourier transform of $x_c(t_1, t_2)$, is given by

$$X(\omega_1, \omega_2) = \frac{1}{T_1 T_2} \sum_{r_1=-\infty}^{\infty} \sum_{r_2=-\infty}^{\infty} X_c\left(\frac{\omega_1 - 2\pi r_1}{T_1}, \frac{\omega_2 - 2\pi r_2}{T_2}\right). \tag{1.53}$$

Two examples of $X_c(\Omega_1, \Omega_2)$ and $X(\omega_1, \omega_2)$ are shown in Figure 1.40. Figure 1.40(a) shows a case in which $1/T_1 > \Omega_c'/\pi$ and $1/T_2 > \Omega_c''/\pi$, where Ω_c' and Ω_c'' are the cutoff frequencies of $X_c(\Omega_1, \Omega_2)$, as shown in the figure. Figure 1.40(b) shows a case in which $1/T_1 < \Omega_c'/\pi$ and $1/T_2 < \Omega_c''/\pi$. From the figure, when $1/T_1 > \Omega_c'/\pi$ and $1/T_2 > \Omega_c''/\pi$, $x_c(t_1, t_2)$ can be recovered from $x(n_1, n_2)$. Otherwise, $x_c(t_1, t_2)$ cannot be exactly recovered from $x(n_1, n_2)$ without additional information on $x_c(t_1, t_2)$. This is the 2-D sampling theorem, and is a straightforward extension of the 1-D result.

An ideal digital-to-analog (D/A) converter recovers $x_c(t_1, t_2)$ from $x(n_1, n_2)$ when the sampling frequencies $1/T_1$ and $1/T_2$ are high enough to satisfy the requirements of the sampling theorem. The output of the ideal D/A converter, $y_c(t_1, t_2)$, is given by

$$y_c(t_1, t_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2) \frac{\sin \frac{\pi}{T_1}(t_1 - n_1 T_1)}{\frac{\pi}{T_1}(t_1 - n_1 T_1)} \frac{\sin \frac{\pi}{T_2}(t_2 - n_2 T_2)}{\frac{\pi}{T_2}(t_2 - n_2 T_2)}. \tag{1.54}$$

The function $y_c(t_1, t_2)$ is identical to $x_c(t_1, t_2)$ when the sampling frequencies used in the ideal A/D converter are sufficiently high. Otherwise, $y_c(t_1, t_2)$ is an aliased version of $x_c(t_1, t_2)$. Equation (1.54) is a straightforward extension of the 1-D result.

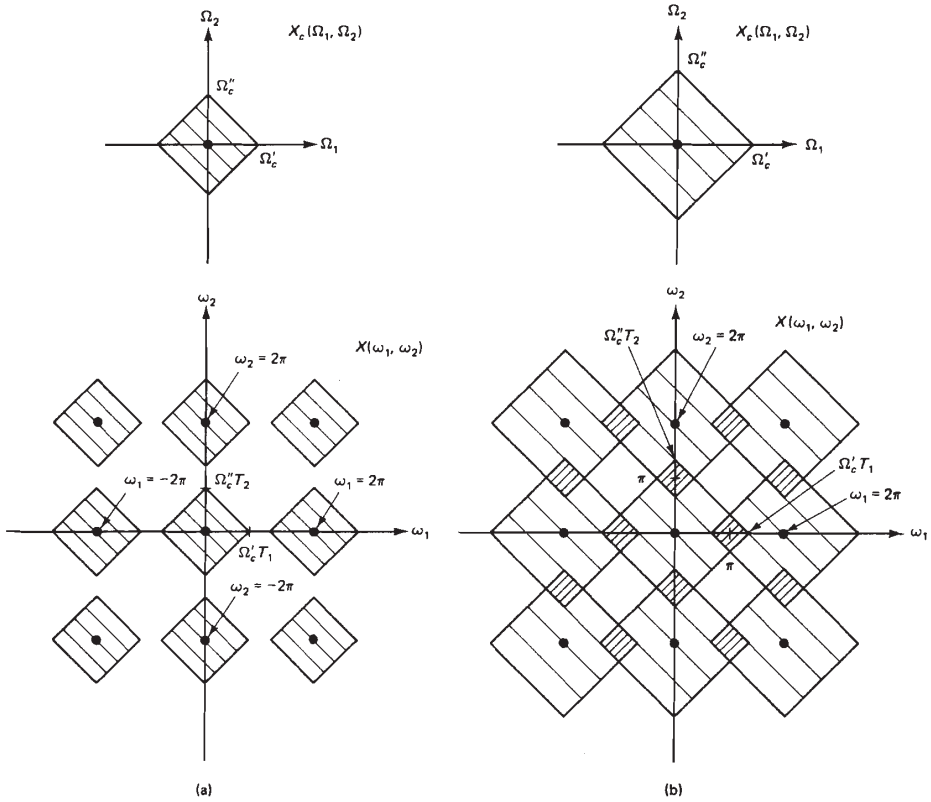


Figure 1.40 Example that illustrates the relationship between $X_c(\Omega_1, \Omega_2)$ and $X(\omega_1, \omega_2)$ given by (1.53). (a) No aliasing; (b) aliasing. Aliased regions are shown shaded.

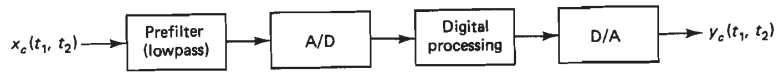


Figure 1.41 Digital processing of analog signals.

(a)

(b)

Figure 1.42 (a) Image of 128 × 128 pixels with little aliasing due to an effective antialiasing filter; (b) image of 128 × 128 pixels with noticeable aliasing.

An analog signal can often be processed by digital processing techniques using the A/D and D/A converters discussed above. The digital processing of analog signals can, in general, be represented by the system in Figure 1.41. The analog lowpass filter limits the bandwidth of the analog signal to reduce the effect of aliasing. In digital image processing, the analog prefiltering operation is often performed by a lens and the scanning aperture used in converting an optical image to an electrical signal. The importance of the antialiasing filter is illustrated in Figure 1.42. Figure 1.42(a) shows an image of 128 × 128 pixels with little aliasing due to an effective antialiasing filter used. Figure 1.42(b) shows an image of 128 × 128 pixels with noticeable aliasing.

The A/D converter of (1.52) is based on sampling on the Cartesian grid. The analog signal can also be sampled on a different type of grid. Sampling on a hexagonal grid is discussed in Problem 1.35.

REFERENCES

In this text, we have assumed that the reader is familiar with fundamentals of 1-D digital signal processing. For a comprehensive treatment of 1-D digital signal processing concepts, see [Oppenheim and Schafer (1975); Rabiner and Gold; Lim and Oppenheim; Oppenheim and Schafer (1989)].

For different viewpoints or more detailed treatment of some topics in 2-D digital signal processing, see [Huang; Huang; Dudgeon and Mersereau]. For collections of selected papers on 2-D digital signal processing, see [Mitra and Ekstrom; IEEE].

For a more detailed treatment of the Fourier transform theory, see [Papoulis]. For processing data obtained from sampling on any regular periodic lattice including the rectangular lattice and hexagonal lattice, see [Mersereau; Mersereau and Speake].

Y. M. Bruck and L. G. Sodin, On the ambiguity of the image reconstruction problem, *Opt. Commun.*, September 1979, pp. 304–308.

D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

M. H. Hayes, The reconstruction of a multidimensional sequence from the phase or magnitude of its Fourier transform, *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP-30, April 1982, pp. 140–154.

T. S. Huang, ed., *Two-Dimensional Digital Signal Processing I*, in "Topics in Applied Physics," Vol. 42. Berlin: Springer-Verlag, 1981.

T. S. Huang, ed.,*Two-Dimensional Digital Signal Processing II*, in "Topics in Applied Physics," Vol. 43. Berlin: Springer-Verlag, 1981.

IEEE, ASSP Society's MDSP Committee, editor, *Selected Papers in Multidimensional Digital Signal Processing*, IEEE Press, New York, 1986.

D. Izraelevitz and J. S. Lim, A new direct algorithm for image reconstruction from Fourier

transform magnitude. *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP-35, April 1987, pp. 511–519.

A. C. Kak, "Image Reconstruction from Projections," in *Digital Image Processing Techniques*, edited by M. Ekstrom. Orlando, FL: Academic Press, 1984, Chapter 4.

R. G. Lane, W. R. Fright, and R. H. T. Bates, Direct phase retrieval, *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP-35, April 1987, pp. 520–525.

J. S. Lim and A. V. Oppenheim, ed., *Advanced Topics in Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1988.

R. M. Mersereau, "The processing of hexagonally sampled two-dimensional signals," *Proc. IEEE*, Vol. 67, May 1979, pp. 930–949.

R. M. Mersereau and T. C. Speake, The processing of periodically sampled multidimensional signals, *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP-31, February 1983, pp. 188–194.

S. K. Mitra and M. P. Ekstrom, eds. *Two-Dimensional Digital Signal Processing*. Stroudsburg, PA: Dowden, Hutchinson and Ross, 1978.

A. V. Oppenheim, J. S. Lim, and S. R. Curtis, Signal synthesis and reconstruction from partial Fourier domain information, *J. Opt. Soc. Amer.*, Vol. 73, November 1983, pp. 1413–1420.

A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1975.

A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

A. Papoulis, *The Fourier Integral and Its Applications*. New York: McGraw-Hill, 1962.

L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1975.

G. N. Ramachandran and R. Srinivasan, *Fourier Methods in Crystallography*. New York: Wiley-Interscience, 1978.

W. O. Saxton, *Computer Techniques for Image Processing in Electron Microscopy*. New York: Academic Press, 1970.

H. J. Scudder, Introduction to computer aided tomography, *Proc. IEEE*, Vol. 66, June 1978, pp. 628–637.

V. T. Tom, T. F. Quatieri, M. H. Hayes, and J. H. McClellan, Convergence of iterative nonexpansive signal reconstruction algorithms, *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP-29, October 1981, pp. 1052–1058.

PROBLEMS

1.1. Sketch the following sequences:
- **(a)** $\delta(n_1 + 2, n_2 - 3) + 2\delta(n_1, -n_2 + 2)$
- **(b)** $\delta_T(n_1)u(n_1, n_2)$
- **(c)** $(\frac{1}{2})^{n_2}\,\delta_T(n_1 + n_2)u(-n_1 + 1, -n_2)$

1.2. Consider a sequence $x(n_1, n_2)$ sketched below:

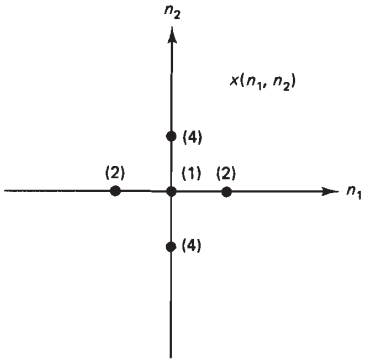


Figure P1.2

Express $x(n_1, n_2)$ as a linear combination of $\delta(n_1, n_2)$ and its shifts.

1.3. We have defined a sequence $x(n_1, n_2)$ to be periodic with a period of $N_1 \times N_2$ if

$$x(n_1, n_2) = x(n_1 + N_1, n_2) = x(n_1, n_2 + N_2) \quad \text{for all } (n_1, n_2). \tag{1}$$

More generally defined, the condition is

$$x(n_1, n_2) = x(n_1 + N_{11}, n_2 + N_{12}) = x(n_1 + N_{21}, n_2 + N_{22}) \text{ for all } (n_1, n_2) \tag{2}$$

with the number of points in a period given by

$$|N_{11}N_{22} - N_{12}N_{21}|.$$

- **(a)** Show that the condition in (2) reduces to the condition in (1) with a proper choice of N_{11}, N_{12}, N_{21}, and N_{22}.
- **(b)** Consider the periodic sequence $x(n_1, n_2)$, which was shown in Figure 1.8. If we use (1), the minimum choices of N_1 and N_2 are 6 and 2, respectively, and the number of points in one period is 12. If we use (2), N_{11}, N_{12}, N_{21}, and N_{22} can be chosen such that $|N_{11}N_{22} - N_{12}N_{21}| = 6$. Determine one such set of N_{11}, N_{12}, N_{21}, and N_{22}.
- **(c)** Show that any sequence that satisfies the condition in (2) will also satisfy the condition in (1) as long as N_1 and N_2 are chosen appropriately. This result shows that (1) can be used in representing any periodic sequence that can be represented by (2), although the number of points in one period may be much larger when (1) rather than (2) is used.

1.4. For each of the following systems, determine whether or not the system is (1) linear, (2) shift invariant, and (3) stable.
- **(a)** $y(n_1, n_2) = T[x(n_1, n_2)] = e^{3x(n_1,n_2)}$
- **(b)** $y(n_1, n_2) = T[x(n_1, n_2)] = \sum_{k_1=n_1-3}^{n_1+6} x(k_1, n_2)$
- **(c)** $y(n_1, n_2) = T[x(n_1, n_2)] = \sum_{k_2=0}^{n_2} x(n_1, k_2)$

1.5. The median filter is used in a number of signal processing applications, including image processing. When a median filter is applied to an image, a window slides along the image, and the median intensity value of the pixels (picture elements) within the window replaces the intensity of the pixel being processed. For example, when the

pixel values within a window are 5, 6, 35, 10, and 5, and the pixel being processed has a value of 35, its value is changed to 6, the median of the five values. Answer each of the following questions. In your answer, use a 2-D median filter of size 3×3, with the center of the window corresponding to the pixel being processed.

(a) Is a median filter linear, shift invariant, and/or stable?

(b) Using an example, illustrate that a median filter tends to preserve sharp discontinuities, such as steps.

(c) Using an example, illustrate that a median filter is capable of eliminating impulsive values without seriously affecting the value of the pixel near those with the impulsive values. A pixel has an impulsive value when its value is significantly different from its neighborhood pixel values.

1.6. Consider a system T. When the input to the system is the unit step sequence $u(n_1, n_2)$, the response of the system is $s(n_1, n_2)$ as shown below.

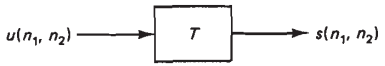


Figure P1.6

For each of the following three cases, determine the class of inputs for which we can determine the output of the system in terms of $s(n_1, n_2)$. For each input in the class, express the output in terms of $s(n_1, n_2)$.

(a) T is linear, but not shift invariant.

(b) T is shift invariant, but not linear.

(c) T is linear and shift invariant.

1.7. Compute $x(n_1, n_2) * h(n_1, n_2)$ for each of the following two problems.

(a)

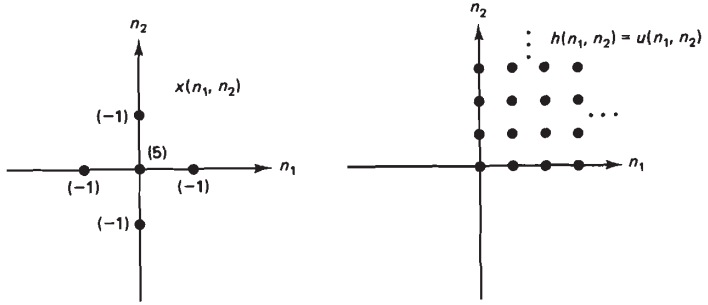


Figure P1.7

(b) $x(n_1, n_2) = (\frac{1}{2})^{n_1}(\frac{1}{3})^{n_2}u(n_1, n_2)$
$h(n_1, n_2) = u(n_1 - 1, n_2 - 2)$

1.8. Convolve the sequence $x(n_1, n_2) = a^{n_1}b^{n_2}u(n_1, n_2)$ with the sequence shown in the figure below. Assume $|a| < 1$. The filled-in circles represent samples with amplitude of 1. The vertical lines at $n_1 = -1, 0, 1$ extend to ∞, and the horizontal line at $n_2 = 0$ extends to $-\infty$ and $+\infty$.

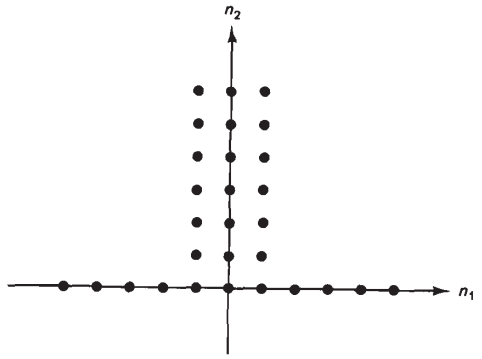


Figure P1.8

1.9. Consider a 2-D LSI system whose impulse response $h(n_1, n_2)$ has first-quadrant support. When the input $x(n_1, n_2)$ to the system is given by

$$x(n_1, n_2) = 2^{n_1n_2}u(n_1, n_2),$$

some portion of the output $y(n_1, n_2)$ has been observed. Suppose the observed portion of $y(n_1, n_2)$ is as shown in the following figure.

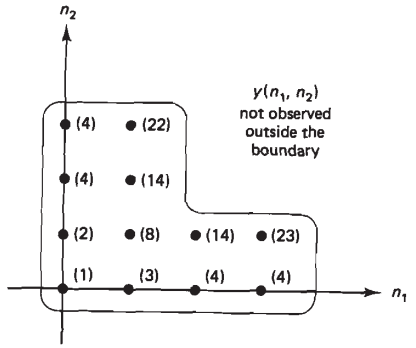


Figure P1.9

Determine $h(1, 1)$, the impulse response $h(n_1, n_2)$ evaluated at $n_1 = n_2 = 1$.

1.10. If the input $x(n_1, n_2)$ to an LSI system is periodic with a period of $N_1 \times N_2$, is the output $y(n_1, n_2)$ of the system periodic? If so, determine the periodicity of $y(n_1, n_2)$.

1.11. Consider the following system in which $x(n_1, n_2)$ represents an input sequence and $h_i(n_1, n_2)$ for $i = 1, 2, 3, 4, 5$ represents the impulse response of an LSI system.

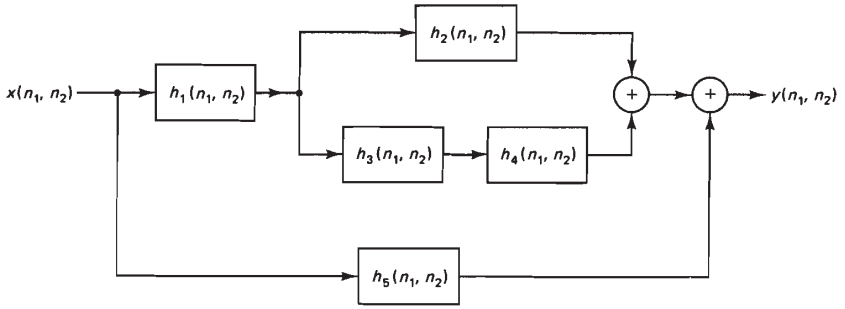


Figure P1.11

Without affecting the input-output relationship, the above system can be simplified as

$$x(n_1, n_2) \rightarrow \boxed{h(n_1, n_2)} \rightarrow y(n_1, n_2).$$

Express $h(n_1, n_2)$ in terms of $h_1(n_1, n_2)$, $h_2(n_1, n_2)$, $h_3(n_1, n_2)$, $h_4(n_1, n_2)$ and $h_5(n_1, n_2)$.

1.12. Let $h(n_1, n_2, n_3)$ be a separable, finite-extent, 3-D sequence of $M \times M \times M$ points which can be expressed as

$$h(n_1, n_2, n_3) = a(n_1)\, b(n_2)\, c(n_3).$$

(a) The Fourier transform of $h(n_1, n_2, n_3)$, $H(\omega_1, \omega_2, \omega_3)$, is defined by

$$H(\omega_1, \omega_2, \omega_3) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \sum_{n_3=-\infty}^{\infty} h(n_1, n_2, n_3) e^{-j\omega_1 n_1} e^{-j\omega_2 n_2} e^{-j\omega_3 n_3}.$$

Show that $H(\omega_1, \omega_2, \omega_3)$ is a separable function that can be expressed in the form of $A(\omega_1)B(\omega_2)C(\omega_3)$.

(b) We wish to filter an input sequence $x(n_1, n_2, n_3)$ of $N \times N \times N$ points using an LSI system with impulse response $h(n_1, n_2, n_3)$ as given above. Develop a computationally efficient way to compute the output $y(n_1, n_2, n_3)$.

(c) How does your method compare to direct evaluation of the convolution sum for each output point when $N = 512$ and $M = 10$?

1.13. An LSI system can be specified by its impulse response $h(n_1, n_2)$. An LSI system can also be specified by its unit step response $s(n_1, n_2)$, the response of the system when the input is the unit step sequence $u(n_1, n_2)$.

(a) Express $y(n_1, n_2)$, the output of an LSI system, in terms of the input $x(n_1, n_2)$ and the unit step response $s(n_1, n_2)$.

(b) In determining the output $y(n_1, n_2)$ of an LSI system, which of the two methods requires less computation: your result in (a), or convolving $x(n_1, n_2)$ with $h(n_1, n_2)$?

1.14. Show that an LSI system is stable in the BIBO (bounded-input-bounded-output) sense if and only if the impulse response of the system $h(n_1, n_2)$ is absolutely summable, that is,

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |h(n_1, n_2)| < \infty.$$

1.15. For each of the following cases, determine the region of support of $y(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2)$.

(a) $x(n_1, n_2)$ and $h(n_1, n_2)$ are first-quadrant support sequences.

(b) $x(n_1, n_2)$ and $h(n_1, n_2)$ are second-quadrant support sequences.

(c) $x(n_1, n_2)$ is a first-quadrant support sequence and $h(n_1, n_2)$ is a fourth-quadrant support sequence.

(d) $x(n_1, n_2)$ is a first-quadrant support sequence and $h(n_1, n_2)$ is a third-quadrant support sequence.

1.16. Let $x(n_1, n_2)$ and $y(n_1, n_2)$ denote an input and the corresponding output sequence of a 2-D system. We define a system to be *pseudo-causal* if $y(n_1, n_2)$ does not depend on $x(n_1 - k_1, n_2 - k_2)$ for $k_1 < 0$ or $k_2 < 0$. Show that a necessary and sufficient condition for an LSI system to be pseudo-causal is that its impulse response $h(n_1, n_2)$ must be a first-quadrant support sequence.

1.17. It is known that any wedge support sequence $x(n_1, n_2)$ can be mapped to a first-quadrant support sequence $y(n_1, n_2)$ without affecting its stability by linear mapping of variables:

$$y(n_1, n_2) = x(m_1, m_2)\,|_{m_1 = l_1 n_1 + l_2 n_2, m_2 = l_3 n_1 + l_4 n_2}$$

where l_1, l_2, l_3, and l_4 are integers. Consider the following wedge support sequence $x(n_1, n_2)$.

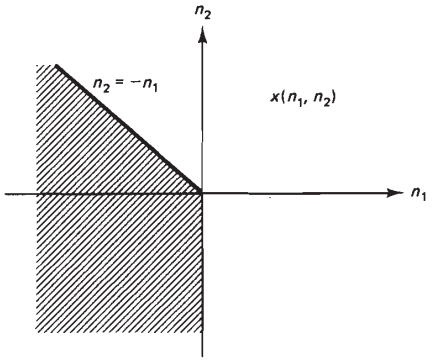


Figure P1.17

The shaded region in the figure is the region of (n_1, n_2) for which $x(n_1, n_2)$ is nonzero. Determine one specific linear mapping of variables that maps $x(n_1, n_2)$ to a first-quadrant support sequence $y(n_1, n_2)$ without affecting its stability.

1.18. Determine if each of the following sequences is an eigenfunction of a general LSI system.

(a) $x(n_1, n_2) = e^{j\omega_2 n_2}$

(b) $x(n_1, n_2) = a^{n_1} + b^{n_2}$

(c) $x(n_1, n_2) = e^{j\omega_1 n_1} e^{j\omega_2 n_2}$

(d) $x(n_1, n_2) = e^{j\omega_1 n_1} e^{j\omega_2 n_2} u(n_1, n_2)$

1.19. Determine the Fourier transform of each of the following sequences.

(a) $\delta(n_1, n_2)$

(b) $a^{2n_1 + n_2} u(n_1, n_2)$, $|a| < 1$

(c) $a^{n_1} b^{n_2}\, \delta_T(4n_1 - n_2) u_T(n_1)$, $|a| < 1$, $|b| < 1$

(d) $n_1 (\frac{1}{2})^{n_1} (\frac{1}{2})^{n_2} u(n_1, n_2)$

1.20. The Fourier transform $X(\omega_1, \omega_2)$ of the sequence $x(n_1, n_2)$ is given by

$$X(\omega_1, \omega_2) = 3 + 2\cos\omega_1 + j4\sin\omega_2 + 8e^{-j\omega_1}e^{-j\omega_2}.$$

Determine $x(n_1, n_2)$.

1.21. Consider the following sequence:

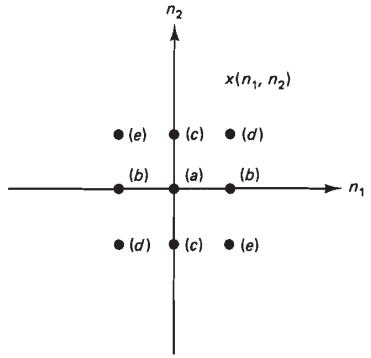


Figure P1.21

The coefficients a, b, c, d, and e are real.

(a) What can you say about $X(\omega_1, \omega_2)$ without explicitly computing $X(\omega_1, \omega_2)$?

(b) Determine $X(0, 0)$.

(c) Determine $X(\omega_1, \omega_2)$.

1.22. Using the Fourier transform pair of Equation (1.31), show that

$$\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}|x(n_1, n_2)|^2 = \frac{1}{(2\pi)^2}\int_{\omega_1=-\pi}^{\pi}\int_{\omega_2=-\pi}^{\pi}|X(\omega_1, \omega_2)|^2\, d\omega_1\, d\omega_2.$$

1.23. We wish to design a highpass filter with impulse response $h(n_1, n_2)$ and frequency response $H(\omega_1, \omega_2)$.

(a) Determine one $h(n_1, n_2)$ which can be viewed as a highpass filter and which has the following property:

$$H(\omega_1, \omega_2) = \begin{cases} 0, & \omega_1 = \omega_2 = 0 \\ 1, & (\omega_1, \omega_2) = (\pi, 0), (\pi, \pi), (0, \pi), (-\pi, \pi), (-\pi, 0), \\ & (-\pi, -\pi), (0, -\pi) \text{ and } (\pi, -\pi). \end{cases}$$

(b) For your answer in (a), determine $H(\omega_1, \omega_2)$. Demonstrate that it can be viewed as a highpass filter by evaluating $H(\omega_1, \omega_2)$ at a reasonable number of values of (ω_1, ω_2).

(c) Let $x(n_1, n_2)$ represent the intensity of a digital image. The amplitude of $x(n_1, n_2)$ is real and nonnegative. We process $x(n_1, n_2)$ with the highpass filter designed in (a) and denote the resulting output by $y(n_1, n_2)$. Determine

$$\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} y(n_1, n_2).$$

(d) From your answer to (c), discuss how $y(n_1, n_2)$ will appear on a display device that sets all negative amplitudes of $y(n_1, n_2)$ to 0 (the darkest level) before it displays $y(n_1, n_2)$.

1.24. The impulse response of a circularly symmetric ideal lowpass filter with cutoff frequency of ω_c is given by

$$h(n_1, n_2) = \frac{\omega_c}{2\pi\sqrt{n_1^2 + n_2^2}} J_1(\omega_c\sqrt{n_1^2 + n_2^2})$$

where $J_1(\cdot)$ is the Bessel function of the first kind and first order. In this problem, we derive this result.

(a) The frequency response of the filter is given by

$$H(\omega_1, \omega_2) = \begin{cases} 1, & \omega_1^2 + \omega_2^2 \le \omega_c^2 \\ 0, & \text{otherwise.} \end{cases}$$

The sequence $h(n_1, n_2)$ is then given by

$$h(n_1, n_2) = \frac{1}{(2\pi)^2}\int_{(\omega_1,\omega_2)\in[\omega_1^2+\omega_2^2\le\omega_c^2]}\int 1e^{j\omega_1 n_1}e^{j\omega_2 n_2}\, d\omega_1\, d\omega_2.$$

We now make the following change of variables:

$$r\cos\theta = \omega_1$$

$$r\sin\theta = \omega_2.$$

Show that $h(n_1, n_2)$ can be expressed as

$$h(n_1, n_2) = \frac{1}{(2\pi)^2}\int_{r=0}^{\omega_c} r\, dr \int_{\theta=a}^{a+2\pi} e^{jr(n_1\cos\theta + n_2\sin\theta)}\, d\theta \tag{1}$$

for any real constant a.

(b) We next make the following change of variables:

$$n_1 = n\cos\phi$$

$$n_2 = n\sin\phi$$

Note that $n^2 = n_1^2 + n_2^2$. Show that $h(n_1, n_2)$ can be written as

$$h(n_1, n_2) = \frac{1}{(2\pi)^2}\int_{r=0}^{\omega_c} rf(r)\, dr \tag{2}$$

where

$$f(r) = \int_{\theta=a}^{a+2\pi} e^{jrn\cos(\theta-\phi)}\, d\theta. \tag{3}$$

(c) It is known that

$$J_0(x) = \frac{1}{2\pi}\int_{\theta=0}^{2\pi}\cos(x\sin\theta)\, d\theta = \frac{1}{2\pi}\int_{\theta=0}^{2\pi}\cos(x\cos\theta)\, d\theta \tag{4}$$

where $J_0(x)$ is the Bessel function of the first kind, zeroth order. From (3) and (4) with $a = \phi$, show that

$$f(r) = 2\pi J_0(r\sqrt{n_1^2 + n_2^2}). \tag{5}$$

(d) It is known that

$$xJ_1(x)\Big|_{x=a}^{b} = \int_{x=a}^{b} xJ_0(x)\, dx \tag{6}$$

where $J_1(x)$ is the Bessel function of the first kind, first order. From (2), (5), and (6), show that

$$h(n_1, n_2) = \frac{\omega_c}{2\pi\sqrt{n_1^2 + n_2^2}} J_1(\omega_c\sqrt{n_1^2 + n_2^2}).$$

This is the desired result.

1.25. Determine the impulse response of each of the following two filters. You may use the results of Problem 1.24.

(a) Circularly symmetric ideal highpass filter:

$$H(\omega_1, \omega_2) = \begin{cases} 0, & \omega_1^2 + \omega_2^2 \le \omega_c^2 \text{ (unshaded region)} \\ 1, & \text{otherwise (shaded region)} \end{cases}$$

(b) Circularly symmetric ideal bandpass filter:

$$H(\omega_1, \omega_2) = \begin{cases} 1, & R_1^2 \le \omega_1^2 + \omega_2^2 \le R_2^2 \text{ (shaded region)} \\ 0, & \text{otherwise (unshaded region)} \end{cases}$$

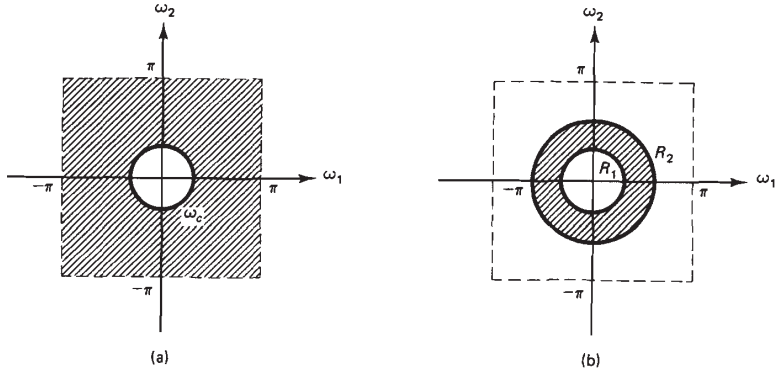


Figure P1.25

1.26. It is well known that circular symmetry of $X(\omega_1, \omega_2)$ implies circular symmetry of $x(n_1, n_2)$. However, circular symmetry of $x(n_1, n_2)$ does not imply circular symmetry of $X(\omega_1, \omega_2)$. To show the latter, determine a circularly symmetric sequence $x(n_1, n_2)$ that is a function of $n_1^2 + n_2^2$ with the property that $X(\omega_1, \omega_2)$ cannot be expressed as a function of $\omega_1^2 + \omega_2^2$ for $\omega_1^2 + \omega_2^2 \le \pi^2$.

1.27. Evaluate the following expression:

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \frac{\left[J_1\left(\frac{1}{2}\sqrt{n_1^2 + n_2^2}\right)\right]^2}{n_1^2 + n_2^2}$$

where $J_1(\cdot)$ is the Bessel function of the first kind and first order.

1.28. Let $f(x, y)$ denote a 2-D analog function that is circularly symmetric and can therefore be expressed as

$$f(x, y) = g(r)|_{r=\sqrt{x^2+y^2}}.$$

Let $F(\Omega_x, \Omega_y)$ denote the 2-D analog Fourier transform of $f(x, y)$. For a circularly symmetric $f(x, y)$, $F(\Omega_x, \Omega_y)$ is also circularly symmetric and can therefore be expressed as

$$F(\Omega_x, \Omega_y) = G(\rho)|_{\rho=\sqrt{\Omega_x^2+\Omega_y^2}}.$$

Note that the analog case is in sharp contrast with the discrete-space case, in which circular symmetry of a sequence $x(n_1, n_2)$ does not imply circular symmetry of its Fourier transform $X(\omega_1, \omega_2)$. The relationship between $g(r)$ and $G(\rho)$ is called the *zeroth-order Hankel transform* pair and is given by

$$G(\rho) = 2\pi \int_{r=0}^{\infty} rg(r)J_0(r\rho)\, dr$$

and

$$g(r) = \frac{1}{2\pi} \int_{\rho=0}^{\infty} \rho G(\rho)J_0(r\rho)\, d\rho,$$

where $J_0(\cdot)$ is the Bessel function of the first kind and zeroth order. Determine the Fourier transform of $f(x, y)$ when $f(x, y)$ is given by

$$f(x, y) = \begin{cases} 1, & \sqrt{x^2 + y^2} \le 2 \\ 0, & \text{otherwise.} \end{cases}$$

Note that

$$xJ_1(x)|_{x=a}^{b} = \int_{x=a}^{b} xJ_0(x)\, dx$$

where $J_1(x)$ is the Bessel function of the first kind and first order.

1.29. Cosine transforms are used in many signal processing applications. Let $x(n_1, n_2)$ be a real, finite-extent sequence which is zero outside $0 \le n_1 \le N_1 - 1$, $0 \le n_2 \le N_2 - 1$. One of the possible definitions of the cosine transform $C_x(\omega_1, \omega_2)$ is

$$C_x(\omega_1, \omega_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) \cos \omega_1 n_1 \cos \omega_2 n_2.$$

(a) Express $C_x(\omega_1, \omega_2)$ in terms of $X(\omega_1, \omega_2)$, the Fourier transform of $x(n_1, n_2)$.

(b) Derive the inverse cosine transform relationship; that is, express $x(n_1, n_2)$ in terms of $C_x(\omega_1, \omega_2)$.

1.30. In reconstructing an image from its Fourier transform phase, we have used an iterative algorithm, shown in Figure 1.30. The method of imposing constraints separately in each domain in an iterative manner in order to obtain a solution that satisfies all the required constraints is useful in a variety of applications. One such application is the band-limited extrapolation of a signal. As an example of a band-limited extrapolation problem, consider $x(n_1, n_2)$, which has been measured only for $0 \le n_1 \le N - 1$, $0 \le n_2 \le N - 1$. From prior information, however, we know that $x(n_1, n_2)$ is band-limited and that its Fourier transform $X(\omega_1, \omega_2)$ satisfies $X(\omega_1, \omega_2) = 0$ for $\sqrt{\omega_1^2 + \omega_2^2} \ge \omega_c$. Develop an iterative algorithm that may be used for determining $x(n_1, n_2)$ for all (n_1, n_2). You do not have to show that your algorithm converges to a desired solution. However, using $N = 1$, $x(0, 0) = 1$, and $\omega_c = \frac{\pi}{2}$, carry out a few iterations of your algorithm and illustrate that it behaves reasonably for at least this particular case.

1.31. Let $x(n_1, n_2)$ represent the intensity of a digital image. Noting that $|X(\omega_1, \omega_2)|$ decreases rapidly as the frequency increases, we assume that an accurate model of $|X(\omega_1, \omega_2)|$ is

$$|X(\omega_1, \omega_2)| = \begin{cases} Ae^{-2\sqrt{\omega_1^2+\omega_2^2}}, & \sqrt{\omega_1^2 + \omega_2^2} \leq \pi \\ 0, & \text{otherwise.} \end{cases}$$

Suppose we reconstruct $y(n_1, n_2)$ by retaining only a fraction of the frequency components of $x(n_1, n_2)$. Specifically,

$$Y(\omega_1, \omega_2) = \begin{cases} X(\omega_1, \omega_2), & \sqrt{\omega_1^2 + \omega_2^2} \leq \dfrac{\pi}{10} \\ 0, & \text{otherwise.} \end{cases}$$

The fraction of the frequency components retained is $\dfrac{\pi(\pi/10)^2}{4\pi^2}$, or approximately 1%. By evaluating the quantity

$$\frac{\sum_{n1=-\infty}^{\infty} \sum_{n2=-\infty}^{\infty} (y(n_1, n_2) - x(n_1, n_2))^2}{\sum_{n1=-\infty}^{\infty} \sum_{n2=-\infty}^{\infty} x^2(n_1, n_2)},$$

discuss the amount of distortion in the signal caused by discarding 99% of the frequency components.

1.32. For a typical image, most of the energy has been observed to be concentrated in the low-frequency regions. Give an example of an image for which this observation may not be valid.

1.33. In this problem, we derive the projection-slice theorem, which is the basis for computed tomography. Let $f(t_1, t_2)$ denote an analog 2-D signal with Fourier transform $F(\Omega_1, \Omega_2)$.

(a) We integrate $f(t_1, t_2)$ along the t_2 variable and denote the result by $p_0(t_1)$; that is,

$$p_0(t_1) = \int_{t2=-\infty}^{\infty} f(t_1, t_2)\, dt_2.$$

Express $P_0(\Omega)$ in terms of $F(\Omega_1, \Omega_2)$, where $P_0(\Omega)$ is the 1-D Fourier transform of $p_0(t_1)$ given by

$$P_0(\Omega) = \int_{t1=-\infty}^{\infty} p_0(t_1) e^{-j\Omega t_1}\, dt_1.$$

(b) We integrate $f(t_1, t_2)$ along the t_1 variable and denote the result by $p_{\pi/2}(t_2)$; that is,

$$p_{\pi/2}(t_2) = \int_{t1=-\infty}^{\infty} f(t_1, t_2)\, dt_1.$$

Express $P_{\pi/2}(\Omega)$ in terms of $F(\Omega_1, \Omega_2)$, where $P_{\pi/2}(\Omega)$ is the 1-D Fourier transform of $p_{\pi/2}(t_2)$ given by

$$P_{\pi/2}(\Omega) = \int_{t2=-\infty}^{\infty} p_{\pi/2}(t_2) e^{-j\Omega t_2}\, dt_2.$$

(c) Suppose we obtain $a(t, u)$ from $f(t_1, t_2)$ by the coordinate rotation given by

$$a(t, u) = f(t_1, t_2)|_{t_1 = t\cos\theta - u\sin\theta, t_2 = t\sin\theta + u\cos\theta}$$

where θ is the angle shown in Figure P1.33(a). In addition, we obtain $B(\Omega_1', \Omega_2')$ from $F(\Omega_1, \Omega_2)$ by coordinate rotation given by

$$B(\Omega_1', \Omega_2') = F(\Omega_1, \Omega_2)|_{\Omega_1 = \Omega_1'\cos\theta - \Omega_2'\sin\theta, \Omega_2 = \Omega_1'\sin\theta + \Omega_2'\cos\theta}$$

where θ is the angle shown in Figure P1.33(b).

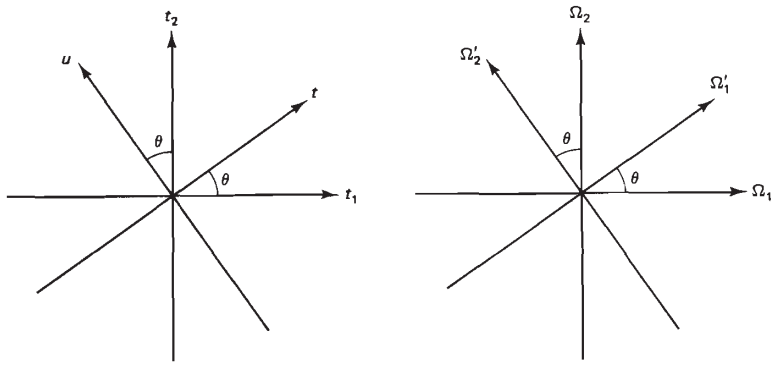


Figure P1.33

Show that $B(\Omega_1', \Omega_2') = A(\Omega_1', \Omega_2')$ where

$$A(\Omega_1', \Omega_2') = \int_{t=-\infty}^{\infty} \int_{u=-\infty}^{\infty} a(t, u) e^{-j\Omega_1' t} e^{-j\Omega_2' u}\, dt\, du.$$

The result states that when $f(t_1, t_2)$ is rotated by an angle θ with respect to the origin in the (t_1, t_2) plane, its Fourier transform $F(\Omega_1, \Omega_2)$ rotates by the same angle in the same direction with respect to the origin in the (Ω_1, Ω_2) plane. This is a property of the 2-D analog Fourier transform.

(d) Suppose we integrate $f(t_1, t_2)$ along the u variable where the u variable axis is shown in Figure P1.33(a). Let the result of integration be denoted by $p_\theta(t)$. The function $p_\theta(t)$ is called the projection of $f(t_1, t_2)$ at angle θ. Using the results of (a) and (c) or the results of (b) and (c), discuss how $P_\theta(\Omega)$ can be simply related to $F(\Omega_1, \Omega_2)$, where

$$P_\theta(\Omega) = \int_{t=-\infty}^{\infty} p_\theta(t) e^{-j\Omega t}\, dt.$$

The relationship between $P_\theta(\Omega)$ and $F(\Omega_1, \Omega_2)$ is the projection-slice theorem.

1.34. Consider an analog 2-D signal $s_c(t_1, t_2)$ degraded by additive noise $w_c(t_1, t_2)$. The degraded observation $y_c(t_1, t_2)$ is given by

$$y_c(t_1, t_2) = s_c(t_1, t_2) + w_c(t_1, t_2).$$

Suppose the spectra of $s_c(t_1, t_2)$ and $w_c(t_1, t_2)$ are nonzero only over the shaded regions shown in the following figure.

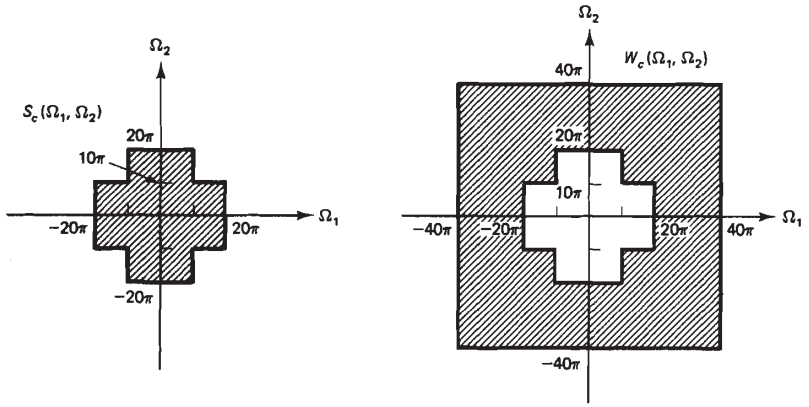


Figure P1.34

We wish to filter the additive noise $w_c(t_1, t_2)$ by digital filtering, using the following system:

$$y_c(t_1, t_2) \rightarrow \boxed{\text{Ideal A/D}} \xrightarrow{y(n_1,n_2)} \boxed{H(\omega_1, \omega_2)} \xrightarrow{\hat{s}(n_1,n_2)} \boxed{\text{Ideal D/A}} \xrightarrow{\hat{s}_c(t_1,t_2)}$$

$$y(n_1, n_2) = y_c(t_1, t_2)|_{t_1 = n_1T_1, t_2 = n_2T_2}$$

$$\hat{s}_c(t_1, t_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \hat{s}(n_1, n_2) \frac{\sin \frac{\pi}{T_1}(t_1 - n_1T_1) \sin \frac{\pi}{T_2}(t_2 - n_2T_2)}{\frac{\pi}{T_1}(t_1 - n_1T_1) \frac{\pi}{T_2}(t_2 - n_2T_2)}$$

Assuming that it is possible to have any desired $H(\omega_1, \omega_2)$, determine the maximum T_1 and T_2 for which $\hat{s}_c(t_1, t_2)$ can be made to equal $s_c(t_1, t_2)$.

1.35. In Section 1.5, we discussed the results for the ideal A/D and D/A converters when the analog signal is sampled on a rectangular grid. In this problem, we derive the corresponding results when the analog signal is sampled on a hexagonal grid. Let $x_c(t_1, t_2)$ and $X_c(\Omega_1, \Omega_2)$ denote an analog signal and its analog Fourier transform. Let $x(n_1, n_2)$ and $X(\omega_1, \omega_2)$ denote a sequence and its Fourier transform. An ideal A/D converter converts $x_c(t_1, t_2)$ to $x(n_1, n_2)$ by

$$x(n_1, n_2) = \begin{cases} x_c(t_1, t_2)|_{t_1 = n_1T_1, t_2 = n_2T_2}, & \text{if both } n_1 \text{ and } n_2 \text{ are even, or} \\ & \text{both } n_1 \text{ and } n_2 \text{ are odd} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The sampling periods T_1 and T_2 are related by $T_2 = \frac{\sqrt{3}}{3} T_1$. The sampling grid used in (1) is shown in the following figure.

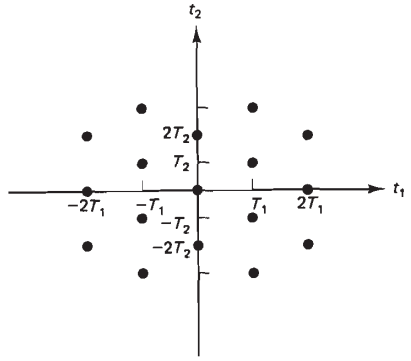


Figure P1.35a

We wish to derive the relationship between $X(\omega_1, \omega_2)$ and $X_c(\Omega_1, \Omega_2)$. It is convenient to represent (1) by the system shown below:

$$x_c(t_1, t_2) \longrightarrow \otimes \xrightarrow{x_s(t_1, t_2)} \boxed{\text{G}} \longrightarrow x(n_1, n_2)$$
$$\uparrow$$
$$p_c(t_1, t_2)$$

The function $p_c(t_1, t_2)$ is a periodic train of impulses given by

$$p_c(t_1, t_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \delta(t_1 - 2n_1T_1, t_2 - 2n_2T_2) + \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \delta(t_1 - 2n_1T_1 - T_1, t_2 - 2n_2T_2 - T_2) \quad (2)$$

where $\delta(t_1, t_2)$ is a dirac-delta function. The system G converts an analog signal $x_s(t_1, t_2)$ to a sequence $x(n_1, n_2)$ by measuring the area under each impulse and using it as the amplitude of the sequence $x(n_1, n_2)$.

(a) Sketch an example of $x_c(t_1, t_2)$, $x_s(t_1, t_2)$ and $x(n_1, n_2)$. Note, from (1), that $x(n_1, n_2)$ is zero for even n_1 and odd n_2 or odd n_1 and even n_2.

(b) Determine $P_c(\Omega_1, \Omega_2)$. Note that the Fourier transform of

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \delta(t_1 - n_1T, t_2 - n_2T)$$

is given by

$$\left(\frac{2\pi}{T}\right)^2 \sum_{r_1=-\infty}^{\infty} \sum_{r_2=-\infty}^{\infty} \delta\left(\Omega_1 - \frac{2\pi r_1}{T}, \Omega_2 - \frac{2\pi r_2}{T}\right).$$

(c) Express $X_s(\Omega_1, \Omega_2)$ in terms of $X_c(\Omega_1, \Omega_2)$.

(d) Express $X(\omega_1, \omega_2)$ in terms of $X_s(\Omega_1, \Omega_2)$.