

## 1

## Signal Sampling

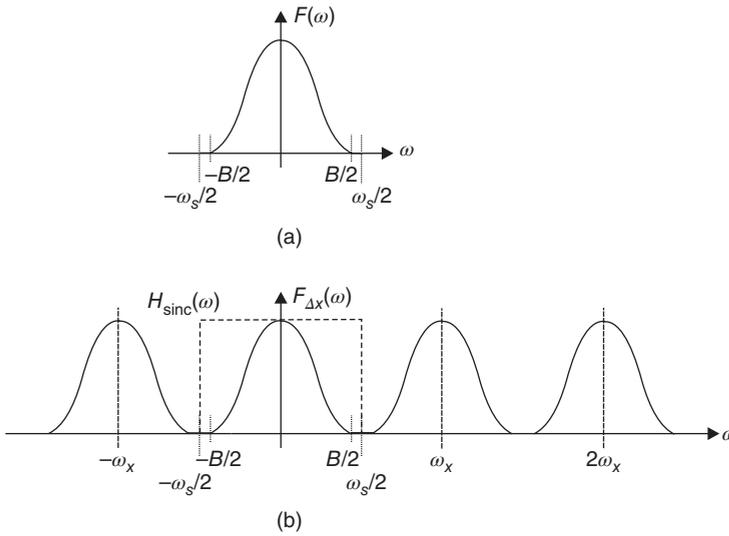
We are living in an analog world that makes it fairly easy to overwhelm our computation system to process the vast information carried by the analog signal. To process the analog signal, it will have to be sampled in a way that the sampled signal can be handled by our computation system. The sampled signal should be able to faithfully represent the analog signal. With this, it is natural to ask: “Is it possible to reconstruct the analog signal from the samples?” Such an important question has been answered by the *sampling theorem* [56]. The sampling theorem considers the signal sequence  $f[k]$  obtained by uniformly sampling an analog function  $f(x)$  with a sampling interval  $\Delta_x$ , such that

$$f[k] = f(x)\delta(x - k\Delta_x) = f(k\Delta_x), \quad \forall k \in \mathbb{Z}, \quad (1.1)$$

where  $\delta(\cdot)$  is a Dirac delta function and  $\mathbb{Z}$  is the set of integers. The sampling theorem tells us when and how to reconstruct the analog signal  $f(x)$  from the sampled signal sequence  $f[k]$ . At the same time, the signal sequence  $f[k]$  to be handled by the computation system is not only a sampled version of  $f(x)$  along  $x$ ; the amplitude of the signal is also “sampled” by a process known as *quantization*. We shall discuss the  $x$  domain (also known as the time domain) sampling process in the next section and the quantization process in Section 1.3. Following the presentation of the sampling theorem, the signal reconstruction problem is alleviated by means of interpolation and/or approximation. Other problems that affect the signal reconstruction accuracy, including quantization, will be discussed in Section 1.3. The quantization problem is an important problem because the quantization process is lossy, which poses tremendous difficulties in the recovery of the analog signal. A number of reconstruction methods for *imperfect signal* will be discussed subsequently.

### 1.1 Sampling and Bandlimited Signal

The readers should have studied Engineering Mathematics in their freshman year; therefore, we shall not discuss the Fourier theorem in detail. Nevertheless, the discrete Fourier transform (DFT) of sampled signal sequence will be introduced in Section 1.2.1 to familiarize the readers with the mathematical notations used in this book. This book also assumes the readers have already acquired the basic knowledge about spectral domain signal processing, and, therefore, this section starts with a formal definition



**Figure 1.1** (a) Spectrum of a bandlimited signal  $f(x)$  with bandwidth  $B$ ; (b) sampled with rate  $\Delta_x = \frac{2\pi}{\omega_s}$  with  $B \leq \omega_x$  can be recovered with a sinc filter with bandwidth  $\omega_s$ .

of bandlimited signal. A signal  $f(x)$  is said to be bandlimited with bandwidth  $B$  if and only if it does not contain any frequency components outside the spectral range of  $-B/2 \leq \omega \leq B/2$ , where  $\omega$  is the angular frequency. An example of bandlimited signal is shown in Figure 1.1, where the  $B$  bandlimited signal  $f(x)$  has its Fourier transform  $F(\omega)$  equal 0 with  $|\omega| > B/2$ .

The sampling theorem tells us the sufficient conditions for the reconstructed signal  $g(x)$  obtained from

$$g(x) = f[k] \otimes h(x) = \sum_{k=-\infty}^{\infty} f(k\Delta_x)h(x - k\Delta_x), \tag{1.2}$$

where  $h(x)$  is the reconstruction function and the sample sequence  $f[k] = f(k\Delta_x)$  with  $k \in \mathbb{Z}$  and  $\Delta_x > 0$  (as discussed in Eq. (1.1)) is lossless, such that  $g(x) = f(x)$ , with  $f(x)$  being bandlimited by  $B$  with sampling frequency  $\omega_x = \frac{2\pi}{\Delta_x} \geq B$ . A formal and also one of the oldest definition of the sampling theorem is given by the following

**Theorem 1.1 Sampling theorem:** Consider a sampled signal  $f[k]$  with samples taken at a  $B$ -bandlimited function  $f(x)$  at sampling period  $\Delta_x$ . The reconstructed signal,

$$g(x) = \sum_{k=-\infty}^{\infty} f[k] \text{sinc} \left( \frac{\pi(x - k\Delta_x)}{\Delta_x} \right) = \sum_{k=-\infty}^{\infty} f[k] \text{sinc} \left( \frac{\omega_x}{2}(x - k\Delta_x) \right), \tag{1.3}$$

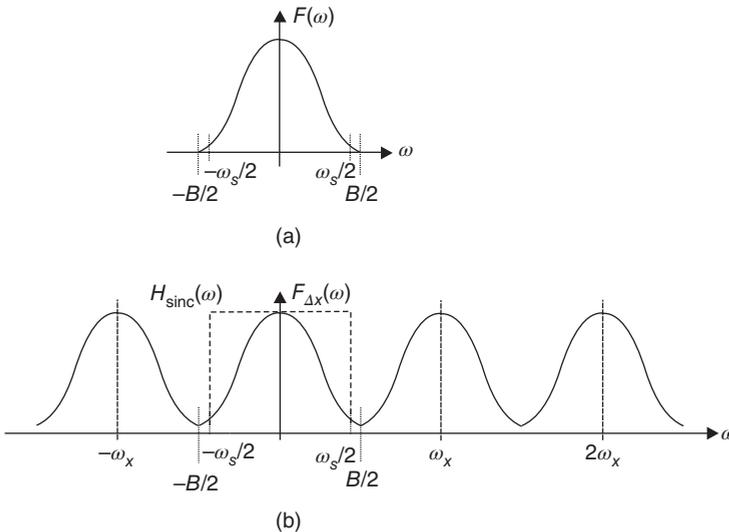
with  $\omega_x = \frac{2\pi}{\Delta_x}$  being the sampling frequency and  $\text{sinc}(a) = \sin(a)/a$  being a sinc function, is an exact reconstruction of  $f(x)$  when  $\omega_s \geq B$ . It should be noted that both  $\omega_x$  and  $B$  are in radian and  $\omega_x = B$  is known as the Nyquist frequency or Nyquist rate.

To understand Eq. (1.3) of the sampling theorem, we can make use of the *discrete time Fourier transform* (DTFT) to examine the reconstructed signal  $g(x)$ .

$$\begin{aligned} G(\omega) &= \sum_k f[k] e^{-2j\omega k} \times \mathcal{F} \left( \text{sinc} \left( \frac{\omega_s}{2} (x - k\Delta_x) \right) \right), \\ &= \frac{H_{\text{sinc}, \Delta_x}(\omega)}{\Delta_x} \sum_{k=-\infty}^{\infty} F(\omega - k\omega_s), \end{aligned} \quad (1.4)$$

where  $H_{\text{sinc}, \Delta_x}(\omega)$  is the DTFT of  $\text{sinc}(\cdot)$  that is a box function of height  $\Delta_x$  in the spectral domain from  $[-\omega_x/2, \omega_x/2]$ , and zero everywhere else, and  $\mathcal{F}$  is the Fourier transform operator. It is vivid from Eq. (1.4) that the spectrum of the sampled signal is a series of duplications of the original analog signal spectrum of  $F(\omega)$  located at spectral locations  $k\omega_x$  with  $k \in \mathbb{Z}$  as shown in Figure 1.1b. Therefore, when the bandwidth of  $f[k]$  is smaller than  $\omega_s$ , the contributions of the duplicated spectral components  $F(\omega - k\omega_x)$  at different  $k$  will not overlap (also known as *aliasing-free*). Otherwise, as shown in Figure 1.2b, when the signal spectrum of  $f(x)$  has a bandwidth wider than  $\omega_s$  as shown in Figure 1.2a, the spectral contributions of the sampled signal spectra at different  $k$  will overlap. As a result, the reconstructed signal obtained by filtering with  $H_{\text{sinc}}(\omega)$  will be a distorted signal  $\hat{F}(\omega)$  (not the same as  $F(\omega)$ ). Such kind of distortion is known as the *aliasing* distortion. This helps to illustrate the *Nyquist frequency* ( $\omega_x = B$ ) as a *sufficient* condition to perfectly reconstruct the analog function  $f(x)$  from its sample sequence  $f[k]$  at a sampling rate  $\Delta_x = \frac{2\pi}{B}$ .

The sampling theorem (Theorem 1.1) stated that a bandlimited signal  $f(x)$  can be sampled at a rate equal to or higher than the Nyquist rate and then reconstructed from its sample sequence without loss by passing the sample sequence  $f[k]$  through a noncausal



**Figure 1.2** (a) Spectrum of a bandlimited signal  $f(x)$  with bandwidth  $B$ ; (b) sampled with rate  $\Delta_x = \frac{2\pi}{\omega_s}$  with  $B > \omega_s$  will suffer from spectrum overlap error, also known as aliasing noise, which makes it difficult to be recovered by a sinc filter with bandwidth  $\omega_s$ .

filter with the impulse response equal to a sinc function. In reality, Eq. (1.3) is of theoretical interest only because the equation is numerically ill conditioned (the range of  $f[k]$  includes both causal and noncausal components). However, it is intuitively clear that the analog function could be closely reconstructed from the sampled sequence using practical reconstruction function (provided that the signal does not change too rapidly and hence bandlimited), and the sampling frequency is relatively high when compared with that of the signal (in that case the sampling frequency is higher than that of the signal bandwidth).

## 1.2 Unitary Transform

The DTFT can be applied to signal sequence with infinite length to represent the signal in frequency domain. For finite length signals, the concept of spectral (Fourier) domain representation is generalized to transform domain representation with unitary transforms. Let us consider a length  $N$  finite duration sequence

$$\mathbf{f} = [ f[0] \quad f[1] \quad \cdots \quad f[N-1] ]^T, \quad (1.5)$$

where  $\mathbf{f}$  can be a vector in either  $\mathbb{R}^{N \times 1}$  or  $\mathbb{C}^{N \times 1}$ . Similarly, consider an invertible matrix  $\mathbf{U}$  that is in either  $\mathbb{R}^{N \times N}$  or  $\mathbb{C}^{N \times N}$ , which is known as the *basis matrix* or *kernel matrix*. A linear transform and the associated inverse transform of  $\mathbf{f}$  by  $\mathbf{U}$  are defined to be

$$\mathbf{F} = \mathbf{U} \cdot \mathbf{f}, \quad (1.6)$$

$$\mathbf{f} = \mathbf{U}^{-1} \cdot \mathbf{F}, \quad (1.7)$$

with  $\mathbf{F} \in \mathbb{R}^{N \times 1}$  or  $\mathbb{C}^{N \times 1}$  being the transform coefficient vector of  $\mathbf{f}$ . In other words, the signal vector  $\mathbf{f}$  is represented by  $\mathbf{F}$  in a domain described by the basis matrix  $\mathbf{U}$ . The transform defined by the set of Eqs. (1.6) and (1.7) is said to be a *unitary transform pair* when  $\mathbf{U} \in \mathbb{R}^{N \times N}$  and

$$\mathbf{U}^{-1} = \mathbf{U}^T \quad \Leftrightarrow \quad \mathbf{U}\mathbf{U}^T = \mathbf{I}. \quad (1.8)$$

In the case of  $\mathbf{U} \in \mathbb{C}^{N \times N}$ , the basis matrix  $\mathbf{U}$  is a unitary transform when it satisfies

$$\mathbf{U}^{-1} = \mathbf{U}^\dagger \quad \Leftrightarrow \quad \mathbf{U}\mathbf{U}^\dagger = \mathbf{I}, \quad (1.9)$$

where the superscript  $\dagger$  denotes the complex conjugate transpose operation and the resulting matrix is known as the Hermitian matrix. The following will present an example of the complex unitary transform, the DFT.

### 1.2.1 Discrete Fourier Transform

The DFT is derived from the DTFT by assuming  $f[n]$  is periodic, which implicitly defines a mapping from  $\mathbb{C}^N$  to  $\mathbb{C}^N$  between  $f[n]$  and  $F[k]$  as

$$f[n] \xrightarrow{F} F[k] = \sum_{n=0}^{N-1} e^{-j2\pi kn/N} f[n], \quad \forall k = 0, \dots, N-1, \quad (1.10)$$

with  $j = \sqrt{-1}$ . The inverse discrete Fourier transform (IDFT) of the sequence  $F[k]$  is given by

$$F[k] \xrightarrow{\mathcal{F}^{-1}} f[n] = \sum_{k=0}^{N-1} e^{\frac{j2\pi kn}{N}} F[k], \quad \forall n = 0, \dots, N-1. \quad (1.11)$$

In the form of unitary transform, the transform kernel of the DFT is given by the  $N \times N$  DFT (Fourier) matrix  $\mathbf{W}_N$ , where the subscript  $N$  indicates the kernel size.

$$\mathbf{W}_N = \left[ e^{\frac{-j2\pi kn}{N}} \right]_{0 \leq k, n < N}. \quad (1.12)$$

If we denote  $W_N^k = e^{\frac{-j2\pi k}{N}}$ , the  $N$ th root of unity, then the Fourier matrix can be expressed as a Vandermonde matrix in  $W$ . As an example, the  $3 \times 3$  Fourier matrix is given by

$$\mathbf{W}_3 = \begin{bmatrix} W_3^0 & W_3^0 & W_3^0 \\ W_3^0 & W_3^1 & W_3^2 \\ W_3^0 & W_3^2 & W_3^4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & W_3^1 & W_3^2 \\ 1 & W_3^2 & W_3^1 \end{bmatrix}. \quad (1.13)$$

Therefore, one can view the computation of  $F[k]$  from  $f[n]$  as a matrix vector product of

$$\mathcal{F}(f) = \mathbf{W}_N \mathbf{f} = \mathbf{F}. \quad (1.14)$$

The IDFT can be easily obtained by multiplication of  $\mathbf{W}_N^{-1}$  to Eq. (1.14). Since the matrix  $\mathbf{W}_N$  is an orthogonal matrix, therefore,  $\mathbf{W}_N^{-1} = \mathbf{W}_N^\dagger$  as given by Eq. (1.9). In image interpolation,  $N$  is usually very large, and an efficient method to compute the DFT is required. In MATLAB, an efficient computation of the DFT is available by means of the *fast Fourier transform* (FFT) command `fft`.

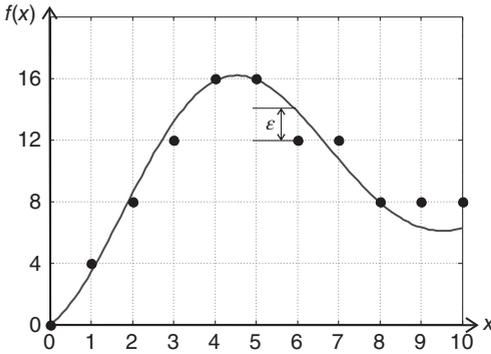
It is vivid that the kernel of the Fourier matrix  $\mathbf{W}_N$  is a function of  $j = \sqrt{-1}$ , which makes this kernel complex. As a result the power of the signal in frequency domain (Fourier domain) given by the power spectrum  $P[u]$  is obtained as the sum of squares of the real and imaginary part of the DFT

$$P[u] = |F[u]|^2 = (R^2(F[u]) + I^2(F[u])), \quad (1.15)$$

which measures the power of individual sinusoidal components contained in the signal.

### 1.3 Quantization

The time domain ( $x$  domain) sampled signal has a continuum of values, as can be observed from the solid line in Figure 1.3. However, the sampled analog signal must be representable in digital form for storage or transmission. Since the number of bits (binary digits) for representing each signal sample is limited, the analog samples must be *quantized* to a finite number of levels before it can be coded in the form of binary numbers. As a result, the quantization process compresses the continuum of analog values to a finite number of discrete values. It is vivid that the quantization process will introduce distortion into the quantized signal when compared with the original



**Figure 1.3** Sampling and quantization of a one-dimensional continuous signal.

analog signal. This kind of distortion is known as *quantization noise*. In simple terms, a scalar quantizer for real signal is a mapping from  $\mathbb{R}$  to a finite set of discrete values on the real number line. The quantized value is chosen to be the closest approximation to the amplitude of the input signal within the finite set. Formally, a scalar quantizer  $Q(\cdot)$  defines the mapping of the input *decision intervals* ( $d_k : k = 0, 1, \dots, L$ ) to output or *reconstruction levels* ( $r_k : k = 0, \dots, L - 1$ ). The quantized signal is given by

$$f_Q(x) = Q(f(x)) = r_k \quad \text{with } d_k \leq f(x) < d_{k+1} \quad \text{for } k = 0, \dots, L - 1. \quad (1.16)$$

Without loss of generality, the decision levels are chosen such that

$$d_0 < d_1 < \dots < d_L. \quad (1.17)$$

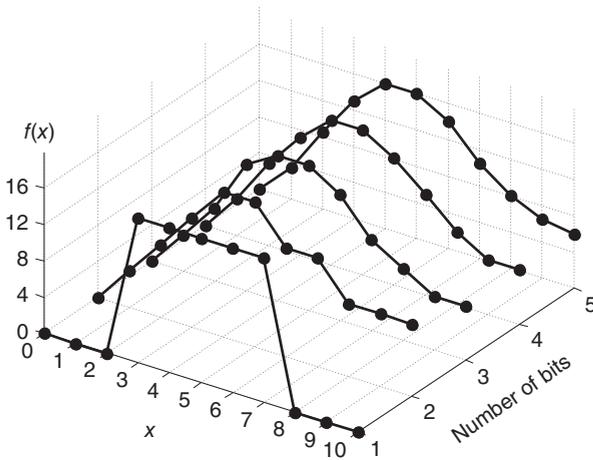
Furthermore,  $d_0$  and  $d_L$  are selected to be the minimum and maximum possible input signals. It should be noted that  $d_0 = -\infty$  and  $d_L = \infty$  are valid and are being chosen for most of the quantizers applied in practice. As a result, the number of bits required to address any one of the output levels is  $\lceil \log_2 L \rceil$  bits with  $\lceil \cdot \rceil$  being the ceiling operator that returns the smallest integer equal to or larger than  $\log_2 L$ . There exist a lot of quantizers (a particular choice of  $d_k$  and  $r_k$ ) that are optimal for different applications. Without loss of generality and limitation in our discussions, we shall focus on uniform quantizer in this book, where the difference between decision levels of the quantizer equals to a constant step size  $\Delta_Q$ .

$$\Delta_Q = d_k - d_{k-1}, \quad \forall k \in \mathbb{Z}^+. \quad (1.18)$$

An example of an analog signal being sampled and quantized is shown in Figure 1.3, where the analog signal plotted in the figure is a damped cosine function.

$$f(x) = 10e^{-x/10} \cos\left(\frac{x}{10}\omega - \theta\right) - \gamma, \quad (1.19)$$

with  $\omega = 2\pi$ ,  $\theta = 3$ , and  $\gamma = -9.9$ . The sampled and quantized signal samples are plotted in Figure 1.3 by black dots together with the analog signal  $f(x)$  by solid line. It can be observed that the sampled signal can faithfully represent the analog signal with quantization error  $\epsilon(x)$  (also known as quantization noise as marked in Figure 1.3 for the case of  $x = 6$ ). The quantization error is highly correlated with the number of bits applied to quantize the signal. Shown in Figure 1.4 is the same signal being sampled with the same

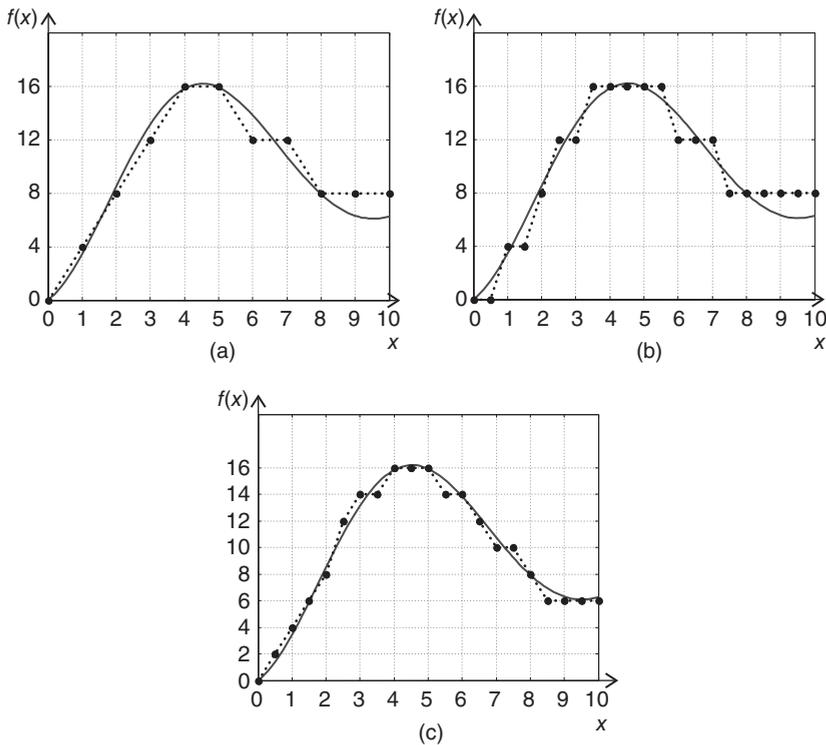


**Figure 1.4** Signal sampling with different quantization bit depths.

sampling rate and quantized with a uniform quantizer under different numbers of bits. It can be observed from Figure 1.4 that the “*quality*” of the sampled and quantized signal sequence improves rapidly with small increases in the number of bits being applied to the uniform quantizer. The higher the *quality*, the better the sampled and quantized signal sequences resemble the analog signal, both visually and also in least squares sense. In fact, the improvement is very efficient in particular for the case of increasing the number of quantization bits from 2 bits to 3 bits. This visual evidence leads us to conjecture that there exists a close relationship between the sampling rate and the quantization bit length that affects the quality of the sampled signal.

### 1.3.1 Quantization and Sampling Interaction

The interaction between sampling and quantization can be revealed by observing the sampling and quantization results shown in Figure 1.5. Shown in Figure 1.5a is the damped cosine function as depicted in Eq. (1.19) being sampled and quantized with a particular rate and bit length. Figure 1.5b shows the same damped cosine function sampled with doubled sampling rate but quantized with the same bit length as that in Figure 1.5a. It can be observed that the quality of the sampled signal does not improve significantly. In other words, the *quality* of the sampled signal sequence cannot be improved by increasing the sampling rate alone. It requires the increment of both the sampling rate and the quantization bit length to improve the *quality* of the sampled signal as shown in Figure 1.5c, where the damped cosine function is sampled with doubled sampling rate and quantized with one more bit when compared with that in Figure 1.5a. In the rest of this book, we shall assume that adequate number of quantization bit length is applied to all the image processing and interpolation problems, such that the quality of different image processing and interpolation problems will be independent of the quantization bit length.

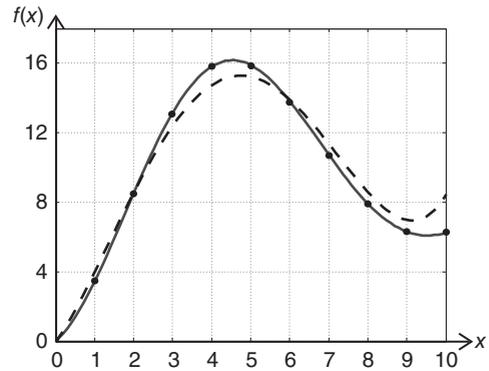


**Figure 1.5** Effects of sampling rate and number of quantization bit on the sampling quality of a one-dimensional continuous signal: (a) a damped cosine function sampled with a particular sampling rate and number of quantization bit, (b) the same function sampled with doubled sampling rate but the same number of quantization bit length, and (c) the same function with the same sampling rate as that in (b) and increased number of quantization bit by one.

## 1.4 Sampled Function Approximation: Fitting and Interpolation

Theoretically the sampled signal can be perfectly reconstructed by Eq. (1.3). However, the filter in Eq. (1.3) is noncausal and thus cannot be used in practice. Furthermore, the samples are obtained with sampling rate that may not satisfy the sampling theorem. Even if the above two conditions are satisfied, the sampled signal will suffer from quantization error. As a result, exact signal reconstruction is difficult, if not impossible. In practice, the signal reconstruction problem is very often reformulated as a *function approximation* problem that extracts a function representation from the given signal samples. The function approximation can be roughly classified into two categories: *interpolation* and *fitting*. The fundamental difference between these two techniques is that the interpolation function passes through all the given signal samples, while the fitting function may not pass all the signal samples. Figure 1.6 is an example illustrating the fundamental difference between interpolation and fitting. The dots lying on the sampling grid are the data points sampled from the original damped cosine function depicted in Eq. (1.19). The solid line is the curve obtained by interpolation, while the dashed line is the result

**Figure 1.6** Reconstructing the analog signal from its samples through *interpolation* (solid line) and *fitting by least squares approximation* (dashed line).



obtained by fitting. It can be observed that the interpolation curve does pass through all data points, while the fitting curve does not. Formally, we can define the interpolation problem as a constrained functional fitting problem, such that the fitting function is constrained to pass through all the given signal samples. There exist a lot of functions that can be applied to the interpolation (function approximation) problem. In particular the one shown in Figure 1.6 is obtained by minimizing the least squares difference between the given samples with a set of predefined basis functions. One of such function is a degree  $n$  polynomial  $P_n(x)$ . The Weierstrass theorem formulated the function approximation problem between  $f(x)$  and  $P_n(x)$  as a least squares problem that minimizes  $\epsilon$  with

$$|f(x) - P_n(x)|_2 \leq \epsilon, \quad x \in [0, N], \quad (1.20)$$

with  $|\cdot|_2$  being the  $\mathcal{L}_2$  norm operator and  $\epsilon > 0$  being a predetermined small quantity also known as the *approximation error*. If the problem is further constrained to satisfy

$$\text{comb}(x, \Delta_x)f(x) = \text{comb}(x, \Delta_x)P_n(x). \quad (1.21)$$

Then the function approximation problem is equivalent to an interpolation problem, where  $P_n(k\Delta_x) = f[k]$ . The comb function  $\text{comb}(x, \Delta_x)$  is given by

$$\text{comb}(x, \Delta_x) = \sum_{k=-\infty}^{\infty} \delta(x - k\Delta_x), \quad (1.22)$$

which is equivalent to a pulse train function. If  $n$  is sufficiently large, a polynomial  $P_n(x)$  can be found to satisfy Eq. (1.21), which in our example is plotted with a solid line in Figure 1.6. On the other hand, when the order of the polynomial is not large enough,  $P_n(x)$  may not be able to satisfy Eq. (1.21), and the best  $P_n(x)$  that satisfies Eq. (1.20) will be similar to that plotted in Figure 1.6 with a dashed line, which does not pass through all the given sample points.

It is vivid that it is impractical to apply polynomial function with a given degree  $n$  to interpolate any given set of signal samples within a predefined interval. As a result, interpolation by polynomial functions is always performed in a piecewise fashion, such as those shown in the following sections, where two frequently used signal reconstruction techniques (which involve the application of two polynomials with degrees 0 and 1) are presented. These two methods are simple and easy to use and are thus adopted in many real-life applications, even though the reconstruction results are far from satisfactory in many cases.

### 1.4.1 Zero-Order Hold (ZOH)

One practical reconstruction method that has been applied in many applications is the zero-order hold (ZOH) method. The ZOH method is also known as the *nearest neighbor* method, where the interpolated signal samples are filled with the same value as that of the signal sample on the left (time or  $x$  domain index smaller than that of the signal location under concern) of it. For theoretical and sometimes practical purposes, it is useful to model the interpolation method by a convolution process. The ZOH interpolation method follows the signal reconstruction Eq. (1.2) with the reconstruction filter  $h(x) = h_0(x)$  given by

$$h_0(x) = \begin{cases} 1, & \text{for } 0 \leq x \leq \Delta_x, \\ 0, & \text{elsewhere.} \end{cases} \quad (1.23)$$

This interpolation kernel is plotted in Figure 1.7a. The filtering can be evaluated between  $f_{\Delta_x}(x)$  and  $h_0(x)$  in time domain through convolution to obtain the reconstructed signal  $g(x)$  as

$$\begin{aligned} g(x) &= h_0(x) \otimes f_{\Delta_x}(x) = \int_0^{\Delta_x} \sum_{k=-\infty}^{\infty} f[k] \delta(x - k\Delta_x - \ell) d\ell \\ &= \sum_{k=-\infty}^{\infty} f[k] (u(x - k\Delta_x) - u(x - (k+1)\Delta_x)), \end{aligned} \quad (1.24)$$

with  $u(\cdot)$  being the step function. The difference between the two-step functions in Eq. (1.24) will lead to a  $g(x)$  that looks like staircase approximation as shown in Figure 1.7b for the sampled damped cosine function  $f(x)$  in Eq. (1.19).

### 1.4.2 First-Order Hold (FOH)

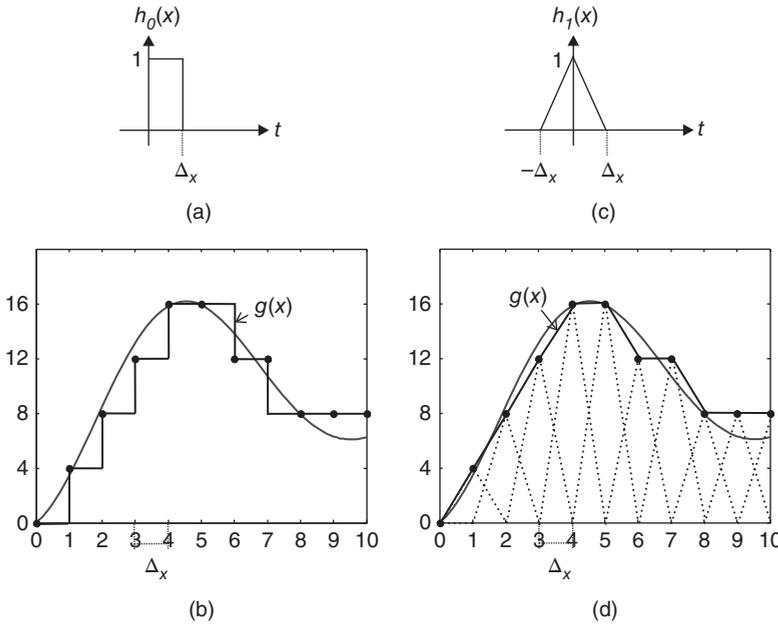
Another frequently used interpolation method constructs an analog signal by connecting adjacent signal samples using a straight line. This is equivalent to interpolate the missing signal as a weighted sum of adjacent signal samples where the weight equals the distance between the missing signal and the two adjacent signal samples. Following Eq. (1.2), the interpolation can be accomplished by passing  $f[k]$  through the first-order hold (FOH) (noncausal) but finite in length filter  $h(x) = h_1(x)$  given by

$$h_1(x) = \text{tri}\left(\frac{x}{\Delta_x}\right) = \begin{cases} 1 - \frac{|x|}{\Delta_x}, & \text{for } |x| \leq \Delta_x, \\ 0, & \text{otherwise,} \end{cases} \quad (1.25)$$

with  $\Delta_x$  being the distance between two signal samples. The interpolation kernel in Eq. (1.25) is equivalent to a triangular function ( $\text{tri}(\cdot)$ ), and its impulse response is plotted in Figure 1.7c. The reconstructed signal  $g(x)$  is thus given by

$$\begin{aligned} g(x) &= h_1(x) \otimes f_{\Delta_x}(x) \\ &= \sum_{k=-\infty}^{\infty} f[k] \text{tri}\left(\frac{x - k\Delta_x}{\Delta_x}\right). \end{aligned} \quad (1.26)$$

The reconstructed signal of the sampled damped cosine function in Eq. (1.19) is shown in Figure 1.7d. It can be observed that the FOH gives much better interpolation results than



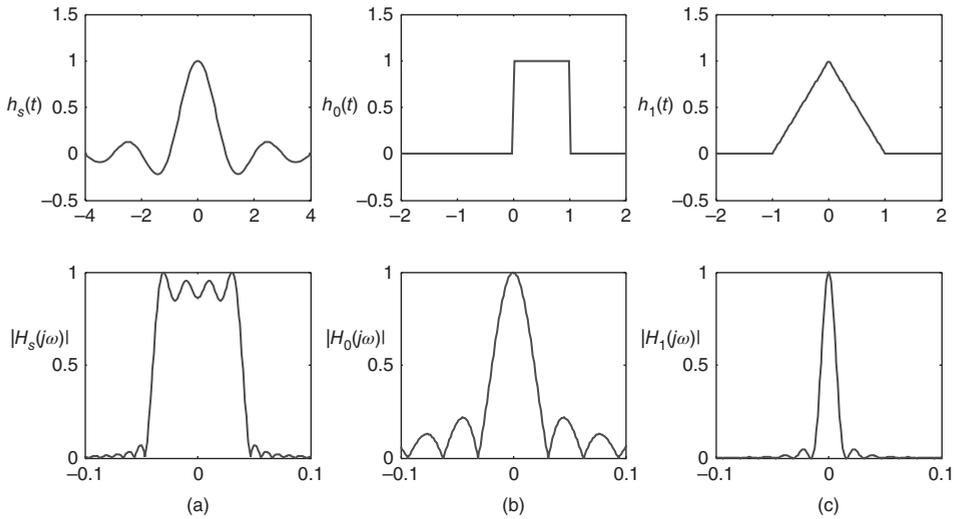
**Figure 1.7** Illustration of (a) impulse response of a zero-order hold (ZOH) filter  $h_0(x)$ ; (b) the sampled signal  $g(x)$  reconstructed using the ZOH filter  $h_0(x)$ ; (c) impulse response of a first-order hold (FOH) filter  $h_1(x)$ ; (d) the reconstructed signal filtered by  $h_1(x)$ .

those obtained by ZOH. If we consider the ZOH and FOH filters in spectral domain, which are given by

$$h_0(x) \stackrel{F}{\leftrightarrow} H_0(j\omega) = \frac{2}{\omega} \sin\left(\frac{\omega\Delta_x}{2}\right) e^{-j\omega\Delta_x/2}, \quad (1.27)$$

$$h_1(x) \stackrel{F}{\leftrightarrow} H_1(j\omega) = \frac{4}{\omega^2\Delta_x} \sin^2\left(\frac{\omega\Delta_x}{2}\right), \quad (1.28)$$

and are plotted in Figure 1.8b,c, respectively, for the case of  $\Delta_x = 1$ , it is vivid that both the time and spectral domain responses of the filter kernels of ZOH and FOH are the approximation to that of the sinc filter with a finite kernel size as shown in Figure 1.8a. The FOH is observed to produce better interpolation result in Figure 1.7d, and at the same time, the FOH kernel (in both time and spectral domains) also achieves better approximation to that of the sinc filter. This observation led us to draw the conclusion that the quality of the interpolated signal not only depends on how well the interpolation filter mimics the sinc filter but also depends on how well the time and spectral responses of the interpolation filter match the time and spectral responses of the analog signal. High-order polynomial interpolation kernel can provide very good mimic to the sinc interpolator. It is, however, the signal reconstructed by higher-order polynomial filter that has a higher-order differentiability. The higher the order of the filter kernel, the faster the decay rate of the filter response in high frequency than that of lower degree filter kernel. As a result, these filters help to minimize the introduction of high frequency interpolation error. But at the same time, they may also remove some high frequency components of the original signal. We shall discuss the two-dimensional (2D) digital



**Figure 1.8** Time and spectral responses for various reconstruction filters: (a) Sinc filter; (b) Zero-order hold; and (c) First-order hold.

image interpolation in a sequel, where it is concluded that the sharpness of the interpolated image is being traded for edge blurring as the degree of the interpolation kernel increases. Disregarding the kind of performance trade-off, it is the purpose of the rest of this book to introduce interpolation kernels and methods that aim at interpolating the sampled signal with the best performance trade-off to reconstruct a signal that closely mimics the analog signal.

### 1.4.3 Digital Interpolation

Instead of reconstructing the analog signal, most of the digital signal applications only interested in *digital interpolation*. The objective of digital interpolation is to obtain a new sequence that is a close approximation to the sampled sequence obtained from sampling the associated analog signal with high sampling rate. Such process is sometimes known as *up-sampling*. The digital interpolation is considered to be computationally efficient, as it avoids the analog function reconstruction problem. As an example, the digital interpolation by a factor of  $r$  through linear filtering can be achieved through a two-stage process, where (i)  $r - 1$  zeros are inserted between every two samples of  $f[k]$  in the first stage and then (ii) filter the zero-inserted signal by a digital filter (the interpolated kernel). Two possible filters (ZOH and FOH) are introduced in previous sections as depicted in Eqs. (1.23) and (1.25). More interpolation filters and their properties will be discussed in subsequent chapters.

## 1.5 Book Organization

In Chapter 2, the fundamental theories of digital and analog images and related mathematical manipulations will be presented, together with a formal definition of the image

interpolation problem. We shall also discuss the application of MATLAB to manipulate digital images in a PC environment. The performance of the interpolation algorithm should be determined by human; however, such quality assessment method is either biased due to small sample size (human observers) or very expensive and slow when the sample size is large (a large group of human observers). Instead, various analytical quality measures are developed, where some are developed specifically to mimic the human visual system to provide limited subjective quality measure. Some of the frequently used quality measures will be discussed in Chapter 3 to assist the performance evaluation of various image interpolation algorithms to be presented in subsequent chapters.

The rest of the book is formally divided into three parts: the traditional nonadaptive image interpolation methods, the model-based image interpolation methods, and the fractal-based interpolation methods. Many other arrangements could be adopted; however, the authors have chosen this framework because they believe that it is the most natural way for the readers to first learn the conventional image interpolation methods and work their way up to the advanced image interpolation methods. Algorithms presented in Chapter 4 are pure nonadaptive interpolation methods, where the image is interpolated with only one assumption, that is, the sampled signal (image) is bandlimited. Chapter 4 derives the traditional nonadaptive linear filtering-based interpolation methods from sampling theorem. Some of the interpolation algorithms are very effective in interpolating texture-rich areas, which will be integrated as part of the model-based image interpolation methods to be presented in later chapters. Chapter 5 is devoted to the discussion of image interpolation in the spectral domain. The basic theoretical performance is the same as the corresponding nonadaptive methods presented in Chapter 4. However, the implementation in frequency domain through block-based transformation allows the change of interpolation kernels (transform basis) on the fly and thus can achieve mixed-basis interpolation with ease. Furthermore, we shall also present iterative interpolation methods, which can be considered as a back propagation algorithm that helps to improve the interpolated image quality.

Chapter 6 extends the image interpolation problem from spatial to spectral domain and then to scale-space. Traditionally wavelet transform can be viewed as conventional transform as those discussed in Chapter 5. However, the multi-resolution decomposition method adopted by wavelet transform allows us to construct a scale-space representation of the image where the transform kernel adapts to the changing scales. Features that are important to the human visual system can be easily located in scale-space. The application of across scale information is the first step to apply model-based image interpolation. The advantage of model-based image interpolation is fully revealed in subsequent chapters.

The model-based image interpolation methods that preserve the structure and edges of the image using explicit edge maps obtained from edge detectors are presented in Chapter 7. The interpolated image quality can be greatly improved; however, their performance is limited by the accuracy of the edge map. Due to the existence of image noise (which includes the quantization noise) and the complexity of the edges in two dimensions, it is difficult to obtain good edge localization and completeness. Edge-directed image interpolation methods that make use of implicit edge information, such as scale-invariant geometric duality in second-order statistics, are presented in Chapter 8.

The iterative interpolation algorithm first discussed in Chapter 5 can be implemented in all model-based algorithms to improve the image interpolation results. This is because

subsequent interpolation loop helps to correct the interpolation error resulted from the previous loop. Similar iterative approach will be applied to the model-based method in Chapter 8.

The fractal image interpolation method presented in Chapter 9 is the ultimate iterative image interpolation method. Instead of modeling the image as composed with edges and filled with texture in between, fractal image coding models the image as self-organized low-resolution fractal images. As a result, interpolation is just as simple as growing the fractal image to the desired resolution. However, fractal image interpolation

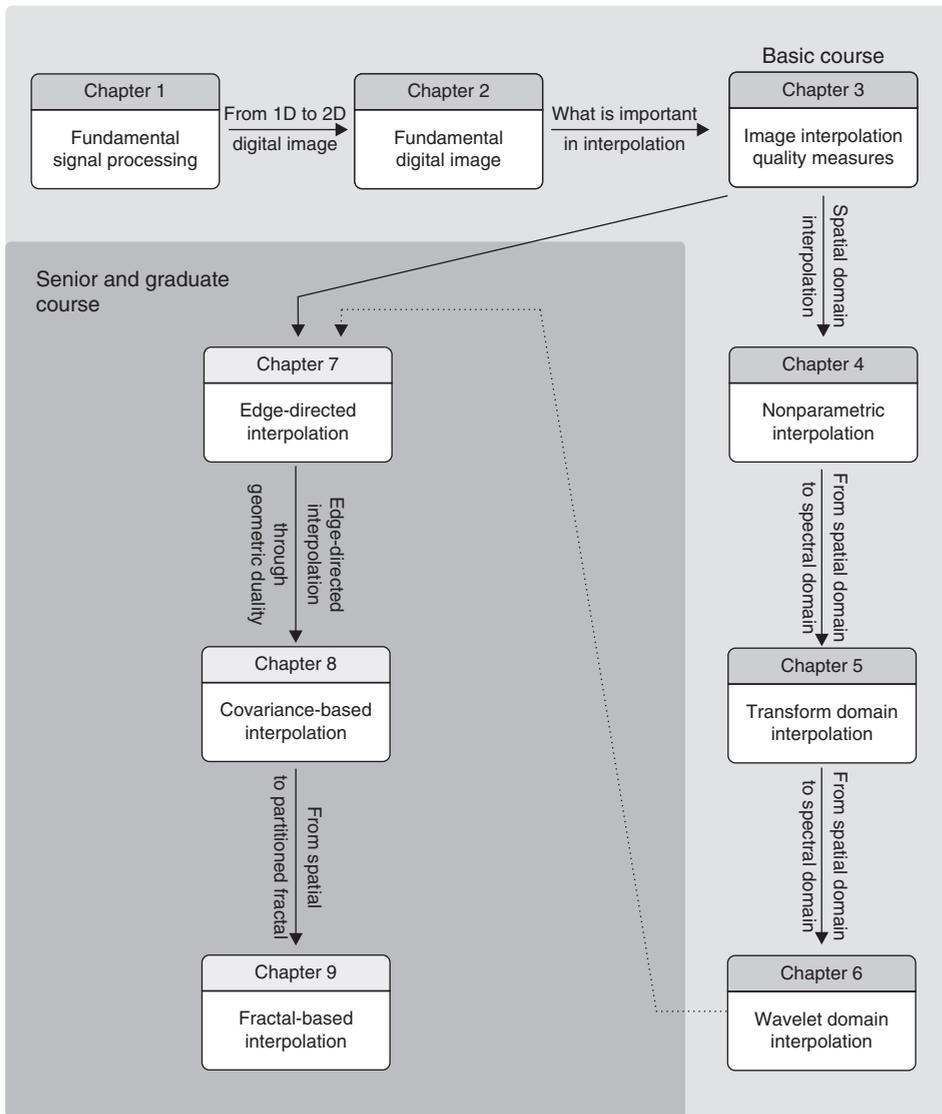


Figure 1.9 Book organization.

is probabilistic and does not guarantee the original image pixel intensities that are preserved in the interpolation process. An appropriate modification of the conventional fractal reconstruction algorithm is required to preserve the low-resolution image pixel values to achieve image interpolation instead of fitting.

This text is intended for use as a senior course in image processing. The above topics can be arranged in many ways in the lectures, depending on the coverage required in the course. Shown in Figure 1.9 is the suggested course material arrangement. A useful approach for undergraduate teaching will cover the materials from Chapters 1 to 6 to provide the understanding of image interpolation from basic signal processing algorithms and interpolated image quality evaluations to nonadaptive image interpolation methods. Chapters 7–9 are advanced topics on image interpolation. In particular, we focus on edge-directed image interpolation methods. A senior undergraduate course can be easily organized from materials presented in Chapters 1–5 and 7. Chapters 6, 8, and 9 involve graduate-level understanding of image processing using high-order statistics, wavelet, and fractal image processing. The materials presented in Chapters 5–9 will form a one-semester graduate-level course, where Chapter 5 will provide the necessary review of the transform domain image processing.

In our experience, students learn more when they are given realistic assignments to carry out. To this end, we would encourage substantial assignments on, for example, the MATLAB implementation of the studied image interpolation methods and tweaking those algorithms to achieve the desired subjective performance. This work should be designed to demonstrate and reinforce the techniques taught. It is important that students actually participate and attend lectures to gain the best from the course.

## 1.6 Exercises

- 1.1 Consider the inverse DFT with  $N = 5$ .
  1. Find the inverse Fourier matrix  $\mathcal{F}^{-1}$ .
  2. Verify that it is the inverse of  $\mathcal{F}$ , that is, show  $\mathcal{F}\mathcal{F}^{-1} = \mathbf{I}$ .
- 1.2 Develop a MATLAB program that accepts  $N + 1$  pairs of data points  $(x_0, y_0), \dots, (x_N, y_N)$  and generates a  $1 \times N$  vector  $[a_N \ a_{N-1} \ a_0]$  such that the polynomial  $P(x) = a_N x^N + a_{N-1} x^{N-1} + \dots + a_0$  will pass through the given data points.
- 1.3 Develop a MATLAB program that accepts  $N$  data points and generates  $2 \times N$  data points by resampling the first-order hold function constructed with the given  $N$  data points.

