Chapter 5

Sampling and Quantization

Often the domain and the range of an original signal $x(t)$ are modeled as continuous. That is, the time (or spatial) coordinate t is allowed to take on arbitrary real values (perhaps over some interval) and the value $x(t)$ of the signal itself is allowed to take on arbitrary real values (again perhaps within some interval). As mentioned previously in Chapter XX, such signals are called *analog* signals. A continuous model is convenient for some situations, but in other situations it is more convenient to work with *digital* signals — i.e., signals which have a discrete (often finite) domain and range. The process of digitizing the domain is called *sampling* and the process of digitizing the range is called *quantization.*

Most devices we encounter deal with both analog and digital signals. Digital signals are particularly robust to noise, and extremely efficient and versatile means for processing digital signals have been developed. On the other hand, in certain situations analog signals are sometimes more appropriate or even necessary. For example, most underlying physical processes are analog (or at least most conveniently modeled as analog), including the human sensorimotor systems. Hence, analog signals are typically necessary to interface with sensors and actuators. Also, some types of data processing and transmission are most conveniently performed with analog signals. Thus, the conversion of analog signals to digital signals (and vice versa) is an important part of many information processing systems.

In this chapter, we consider some of the fundamental issues and techniques in converting between analog and digital signals. For sampling, three fundamental issues are (i) How are the discrete-time samples obtained from the continuous-time signal?; (ii) How can we reconstruct a continuous-time signal from a discrete set of samples?; and (iii) Under what conditions can we recover the continuous-time signal exactly? For quantization, the three main issues we consider are (i) How many quantization levels should we choose?; (ii) How

[*]©1999-2002 by Sanjeev R. Kulkarni. All rights reserved. Please do not distribute without permission of the author.

[†]Lecture Notes for ELE201 Introduction to Electrical Signals and Systems.

[‡]Thanks to Richard Radke for producing the figures.

should the value of the levels be chosen?; and (iii) How should we map the values of the original signal to one of the quantization levels?

5.1 Sampling a Signal

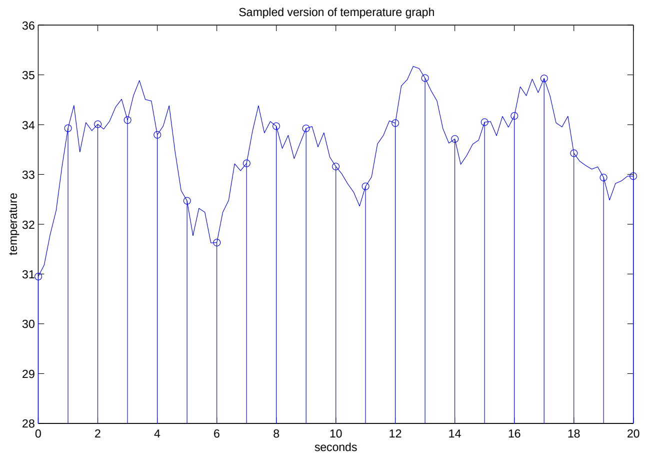


Figure 5.1: Sampling an analog signal.

Figure 5.1 shows an analog signal together with some samples of the signal. The samples shown are equally spaced and simply pick off the value of the underlying analog signal at the appropriate times. If we let T denote the time interval between samples, then the times at which we obtain samples are given by nT where $n = \ldots, -2, -1, 0, 1, 2, \ldots$. Thus, the discrete-time (sampled) signal $x[n]$ is related to the continuous-time signal by

$$x[n] = x(nT).$$

It is often convenient to talk about the sampling frequency f_s. If one sample is taken every T seconds, then the sampling frequency is $f_s = 1/T$ Hz. The sampling frequency could also be stated in terms of radians, denoted by ω_s. Clearly, $\omega_s = 2\pi f_s = 2\pi/T$.

The type of sampling mentioned above is sometimes referred to as “ideal” sampling. In practice, there are usually two non-ideal effects. One effect is that the sensor (or digitizer) obtaining the samples can’t pick off a value at a single time. Instead, some averaging or integration over a small interval occurs, so that the sample actually represents the average value of the analog signal in some interval. This is often modeled as a convolution – namely, we get samples of $y(t) = x(t) * h(t)$, so that the sampled signal is $y[n] = y(nT)$. In this case, $h(t)$ represents the impulse response of the sensor or digitizer. Actually, sometimes this averaging can be desirable. For example, if the original signal $x(t)$ is changing particularly rapidly compared to the sampling frequency or is particularly noisy, then obtaining samples of some averaged signal can actually provide a more useful signal with less variability. The second non-ideal effect is noise. Whether averaged or not, the actual sample value obtained will rarely be the exact value of the underlying analog signal at some time. Noise in the samples is often modeled as adding (usually small) random values to the samples.

Although in real applications there are usually non-ideal effects such as those mentioned above, it is important to consider what can be done in the ideal case for several reasons. The non-ideal effects are often sufficiently small that in many practical situations they can be ignored. Even if they cannot be ignored, the techniques for the ideal case provide insight into how one might deal with the non-ideal effects. For simplicity, we will usually assume that we get ideal, noise-free samples.

5.2 Reconstructing a Signal

In the previous section, we discussed the process of sampling, i.e., obtaining a discrete-time signal from a continuous-time signal. We now consider the reverse problem, namely how to construct a continuous-time signal given discrete-time samples. Suppose we are given a set of samples $x[n]$ that we know came from some continuous-time signal $x(t)$. We also assume we know the sampling rate T, so that we know $x(nT) = x[n]$. How can we recover $x(t)$ for other values of t?

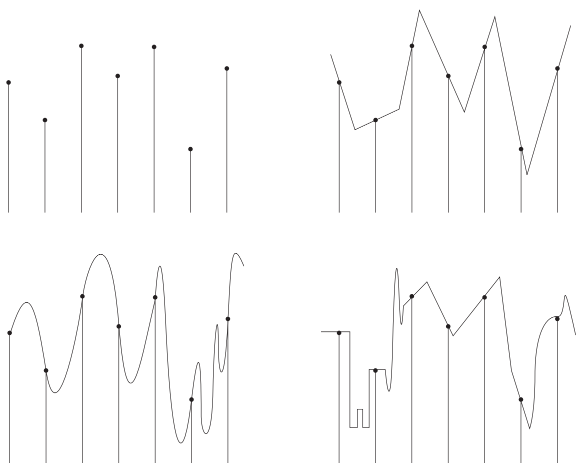


Figure 5.2: Possible continuous-time functions corresponding to samples.

Unless we have some additional knowledge or make some assumptions, this problem clearly has many solutions. Figure 5.2 shows some discrete-time samples and some possible continuous-time functions from which these samples could have been obtained. We have no way to know for sure what the values of the original signal are at times other than nT. However, there are some simple estimates we might make to approximately reconstruct $x(t)$.

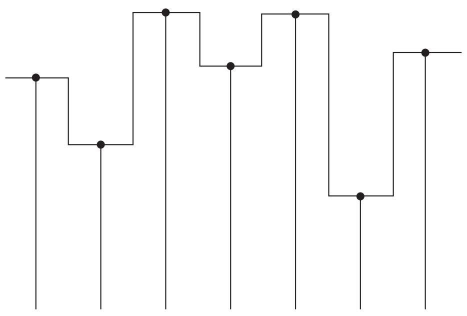


Figure 5.3: Nearest-neighbor reconstruction.

The first estimate one might think of is to just assume that the value at time t is the same as the value of the sample at some time nT that is closest to t. This *nearest-neighbor* interpolation results in a piecewise-constant (staircase-like) reconstruction as shown in Figure 5.3.

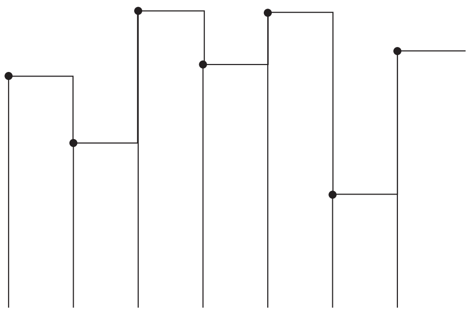


Figure 5.4: Zero-order hold reconstruction.

Actually, instead of nearest-neighbor interpolation, most devices implement a similar type of interpolation known as *zero-order-hold* interpolation shown in Figure 5.4. This is one of the most widely used methods and is easy to implement. As with nearest-neighbor interpolation, this results in a piecewise-constant reconstruction, but the discontinuities are at the sample points instead of between sample points.

Another conceptually simple method is *linear interpolation*, which is also called *first-order-hold* interpolation and is shown in Figure 5.5. With this method, the reconstruction is a continuous function that just connects the sample values with straight lines. Higher order interpolation schemes also exist that pass smoother functions through the samples. The higher the order, the more samples that need to be used to reconstruct a value at each time. One technique that actually uses *all* the samples will be mentioned in the next section.

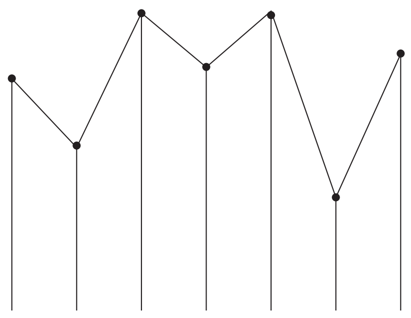


Figure 5.5: First-order hold reconstruction.

These interpolation techniques can be viewed in terms of simple convolution operations. Suppose we replace the discrete-time signal with a continuous-time signal that has delta functions at the sample times nT. The height of (area under) each impulse is the value of the sample at that time. Then the reconstruction from zero-order-hold interpolation can be obtained by convolving this set of impulses with the function

$$h(t) = \begin{cases} 1 & \text{if } 0 \leq t < T \\ 0 & \text{otherwise} \end{cases}$$

In other words, the reconstruction is obtained by passing the discrete-time samples, as a set of impulses, into an appropriate system. In this case, the system has a very simple impulse response and can be easily implemented. Likewise, nearest neighbor and linear interpolation can be viewed as convolving the sample impulses by $\text{rect}(t/T)$ and $\text{tri}(t/T)$, respectively.

One useful property of zero-order-hold that is not satisfied by nearest-neighbor or first-order-hold interpolation is that it is *causal*. This means that the reconstruction at time t only depends on samples obtained *before* time t. Causality can be important in real-time systems when we must (or wish to) reconstruct the continuous-time signal without delays caused by waiting for future sample values. In many applications this is not an issue since either all the data is already stored and available or the delays induced are minimal.

5.3 The Sampling Theorem

In this section we discuss the celebrated Sampling Theorem, also called the Shannon Sampling Theorem or the Shannon-Whitaker-Kotelnikov Sampling Theorem, after the researchers who discovered the result. This result gives conditions under which a signal can be *exactly* reconstructed from its samples. The basic idea is that a signal that changes rapidly will need to be sampled much faster than a signal that changes slowly, but the sampling theorem formalizes this in a clean and elegant way. It is a beautiful example of the power of frequency domain ideas.

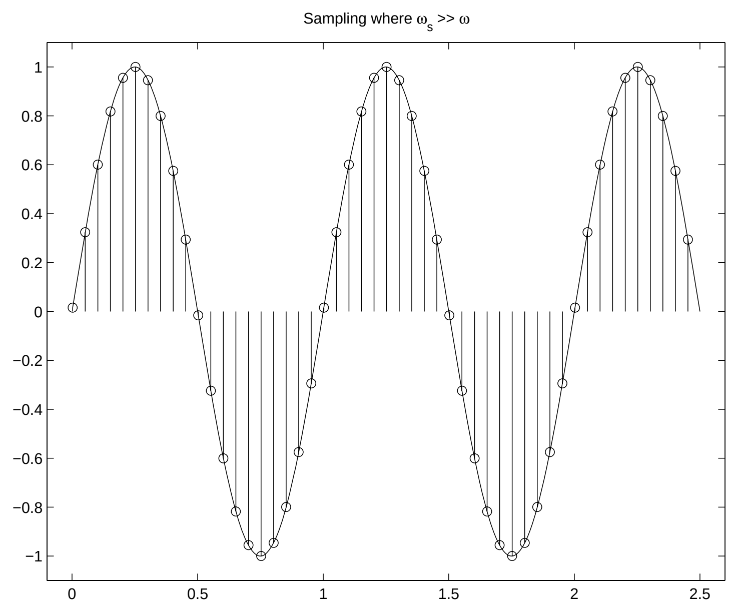


Figure 5.6: Sampling a sinusoid at a high rate.

To begin, consider a sinusoid $x(t) = \sin(\omega t)$ where the frequency ω is not too large. As before, we assume ideal sampling with one sample every T seconds. In this case, the sampling frequency in Hertz is given by $f_S = 1/T$ and in radians by $\omega_s = 2\pi/T$. If we sample at rate fast compared to ω, that is if $\omega_s >> \omega$ then we are in a situation such as that depicted in Figure 5.6. In this case, knowing that we have a sinusoid with ω not too large, it seems clear that we can exactly reconstruct $x(t)$. Roughly, the only unknowns are the frequency, amplitude, and phase, and we have many samples in one period to find these parameters exactly.

On the other hand, if the number of samples is small with respect to the frequency of the sinusoid then we are in a situation such as that depicted in Figure 5.7. In this case, not only does it seem difficult to reconstruct $x(t)$, but the samples actually appear to fall on a sinusoid with a lower frequency than the original. Thus, from the samples we would probably reconstruct the lower frequency sinusoid. This phenomenon is called *aliasing*, since one frequency appears to be a different frequency if the sampling rate is too low. It turns out that if the original signal has frequency f, then we will be able to exactly reconstruct the sinusoid if the sampling frequency satisfies $f_S > 2f$, that is, if we sample at a rate faster than twice the frequency of the sinusoid. If we sample slower than this rate then we will get aliasing, where the alias frequency is given by

$$f_a = |f_s - f|.$$

Actually, we need to be a little bit careful in our thinking. No matter how

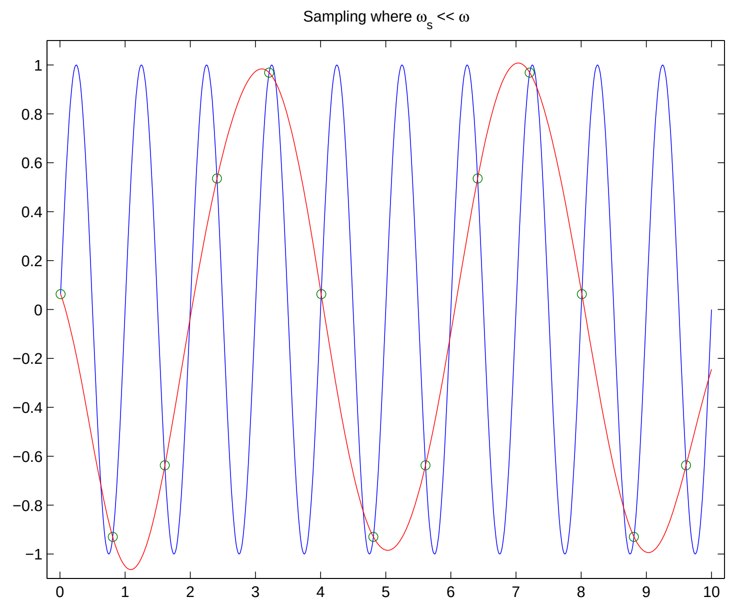


Figure 5.7: Sampling a sinusoid at too slow of a rate.

fast we sample, there are always sinusoids of a sufficiently high frequency for which the sampling rate we happen to be using is too low. So, even in the case of Figure 5.6, where it seems we have no problem recovering the sinusoid, we can't be sure that the true sinusoid is not one of a much higher frequency and we are not sampling fast enough. The way around this problem is to assume from the outset that the sinusoids under consideration will have a frequency no more than some maximum frequency f_B. Then, as long as we sample faster than $2f_B$, we will be able to recover the original sinusoid exactly.

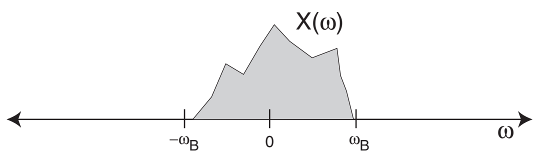


Figure 5.8: The Fourier Transform of a bandlimited signal.

The results for sinusoids would be of limited interest unless we had some way of extending the ideas to more general signals. The natural way to extend the ideas is to use frequency domain analysis and the Fourier transform. A signal $x(t)$ is said to be *bandlimited* if there is some frequency $\omega_B < \infty$ such that

$$X(\omega) = 0 \text{ for } |\omega| > \omega_B$$

That is, a signal is bandlimited if there is some maximum frequency beyond which its Fourier transform is zero. This is shown pictorially in Figure 5.8. This maximum frequency ω_B is the bandlimit of the signal, and can also be expressed in Hertz as

$$f_B = \frac{\omega_B}{2\pi}.$$

If we could recover each sinusoidal component of $x(t)$ then we would be done. To recover a sinusoid at frequency ω we need to sample at a rate at least 2ω. Thus, as long as the various sinusoids can be treated independently, we might guess that for a bandlimited signal we would be able to reconstruct $x(t)$ exactly if we sample at a rate above $2\omega_B$. This is in fact the case, which is a statement of the sampling theorem.

Sampling Theorem A bandlimited signal with maximum frequency ω_B can be perfectly reconstructed from samples if the sampling frequency satisfies $\omega_S > 2\omega_B$ (or equivalently, if $f_S > 2f_B$).

The frequency $2\omega_B$, or $2f_B$ in Hertz, is called the *Nyquist rate*. Thus, the sampling theorem can be rephrased as: a bandlimited signal can be perfectly reconstructed if sampled above its Nyquist rate.

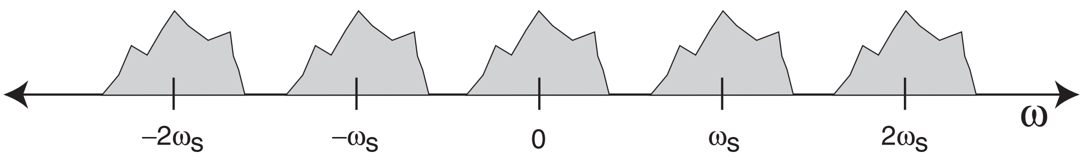


Figure 5.9: The Fourier transformation of a sampled signal consists of shifted replicas of $X(\omega)$.

In addition to just stating that perfect reconstruction can be achieved, formulas for *how* to reconstruct $x(t)$ can also be obtained. These fall out from the proof of the sampling theorem. Rather than give a precise proof and derivation, we describe the basic idea. When a signal is sampled, we can ask what happens in the frequency domain. It turns out that when a signal $x(t)$ is sampled, the Fourier transform of the sampled signal consists of shifted and scaled copies of $X(\omega)$, as shown in Figure 5.9. The spacing between the replicas of $X(\omega)$ depends on the sampling rate, so that the faster we sample, the further apart are the replicas of $X(\omega)$. Specifically, if we take one sample every T seconds then the spacing between the replicas in frequency domain is equal to the sampling frequency $\omega_S = 2\pi/T$. Alternatively, in Hertz, the replicas will be $f_S = 1/T$ apart. The scaling factor is $1/T$, so the replicas are actually $\frac{1}{T}X(\omega)$ rather than $X(\omega)$ itself, but this is not a problem since we can easily undo this scaling.

Thus, we can see that if we sample fast enough, the replicas will be non-overlapping. Then, in principle we could isolate the one un-shifted copy of $X(\omega)$ and take the inverse Fourier transform to recover $x(t)$. To get non-overlapping replicas, we see that for a bandlimited signal $x(t)$ with maximum frequency ω_B, we will need to sample faster than the Nyquist rate $2\omega_B$. In this case, we can get rid of the extra shifted copies of $X(\omega)$ by multiplying by $\text{rect}(\omega/(2\omega_B))$ in frequency domain. Actually, if the sampling frequency is larger than $2\omega_B$, a rect function with even a slightly larger width could also be used, as long as the width of the rect is not so large as to include any parts of other replicas. We will also multiply by T to get $X(\omega)$ instead of $\frac{1}{T}X(\omega)$.

Now, what we have done is to multiply the spectrum of the sampled signal by $\frac{1}{T}\text{rect}(\omega/(2\omega_B)$ in frequency domain to get $X(\omega)$. Then $x(t)$ can be recovered by an inverse Fourier transform. It's also useful to consider what operations are done in the time domain in the process of recovering $x(t)$. Recall that multiplication of two signals in the frequency domain corresponds to a convolution in the time domain – specifically, in this case a convolution of the sampled signal with the sinc function (since the Fourier transform of a sinc function is the rect function).

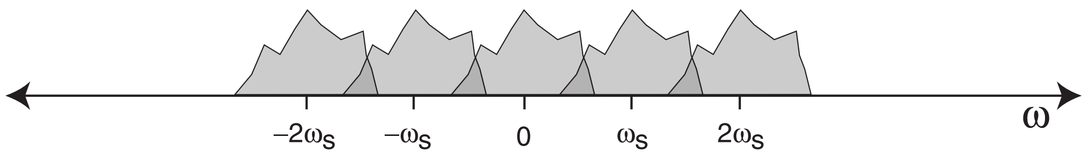


Figure 5.10: Sampling too slowly causes aliasing.

As with pure sinusoids, if a signal is sampled too slowly, then aliasing will occur. High frequency components of the signal will appear as components at some lower frequency. This can also be illustrated nicely in frequency domain. If sampled too slowly, the replicas of $X(\omega)$ overlap and the Fourier transform of the sampled signal is the sum of these overlapping copies of $X(\omega)$. As shown in Figure 5.10, high frequency components alias as lower frequencies and corrupt the unshifted copy of $X(\omega)$. In this figure, the copies are shown individually, with the overlap region simply shaded more darkly. What actually happens is that these copies add together so that we are unable to know what each individual copy looks like. Thus we are unable to recover the original $X(\omega)$, and hence cannot reconstruct the original $x(t)$.

5.4 Quantization

Quantization makes the range of a signal discrete, so that the quantized signal takes on only a discrete, usually finite, set of values. Unlike sampling (where we saw that under suitable conditions exact reconstruction is possible), quantization is generally irreversible and results in loss of information. It therefore introduces distortion into the quantized signal that cannot be eliminated.

One of the basic choices in quantization is the number of discrete quantization levels to use. The fundamental tradeoff in this choice is the resulting signal quality versus the amount of data needed to represent each sample. Figure

5.11 from Chapter XX, repeated here, shows an analog signal and quantized versions for several different numbers of quantization levels. With L levels, we need $N = \log_2 L$ bits to represent the different levels, or conversely, with N bits we can represent $L = 2^N$ levels.

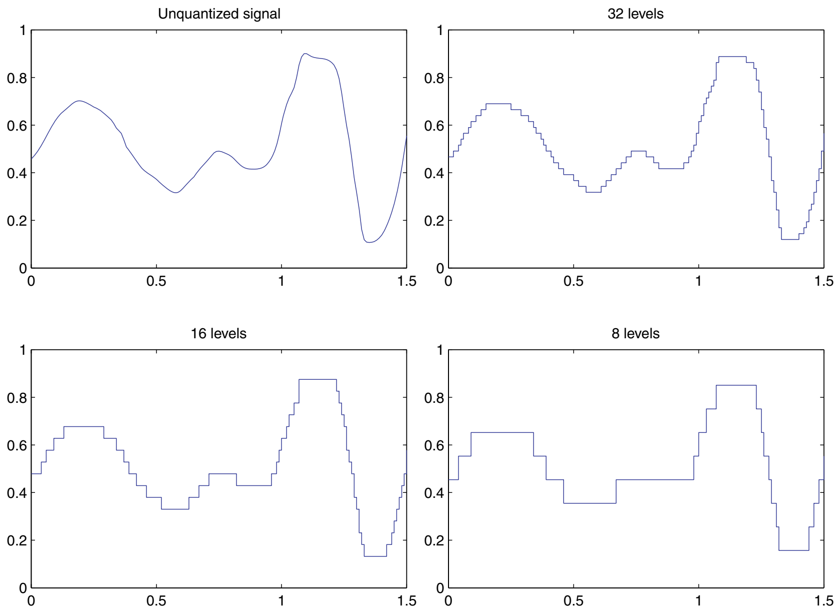


Figure 5.11: Quantized versions of an analog signal.

The simplest type of quantizers are called *zero memory* quantizers in which quantizing a sample is independent of other samples. The signal amplitude is simply represented using some finite number of bits independent of the sample time (or location for images) and independent of the values of neighboring samples. Zero memory quantizers can be represented by a mapping from the amplitude variable x to a discrete set of quantization levels $\{r_1, r_2..., r_L\}$.

The mapping is usually based on simple thresholding or comparison with certain values, t_k. The t_k are called the transition levels (or decision levels), and the r_k are called the reconstruction levels. If the signal amplitude x is between t_k and t_{k+1} then x gets mapped to r_k. This is shown in Figure 5.12.

The simplest zero memory quantizer is the uniform quantizer in which the transition and reconstruction levels are all equally spaced. For example, if the output of an image sensor takes values between 0 and M, and one wants L quantization levels, the uniform quantizer would take

$$r_k = \frac{kM}{L} - \frac{M}{2L}, \quad k = 1, ..., L$$

$$t_k = \frac{M(k-1)}{L}, \quad k = 1, ..., L+1$$

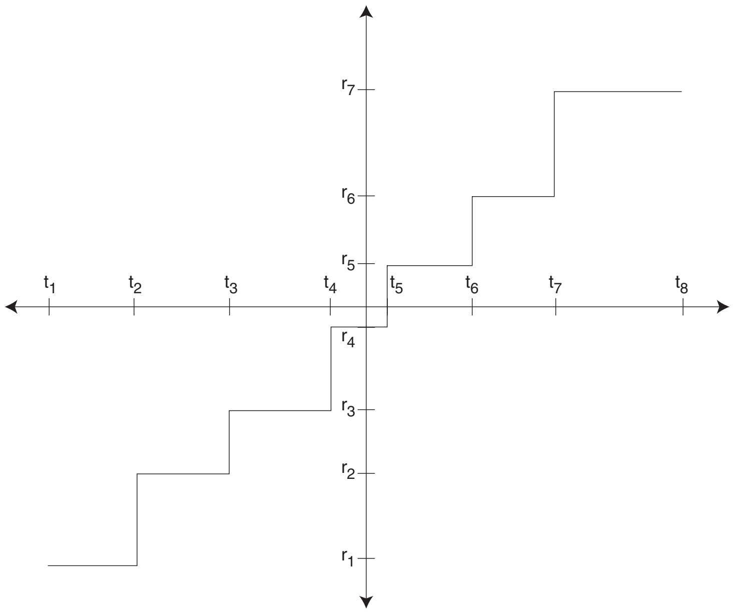


Figure 5.12: A zero-memory quantizer.

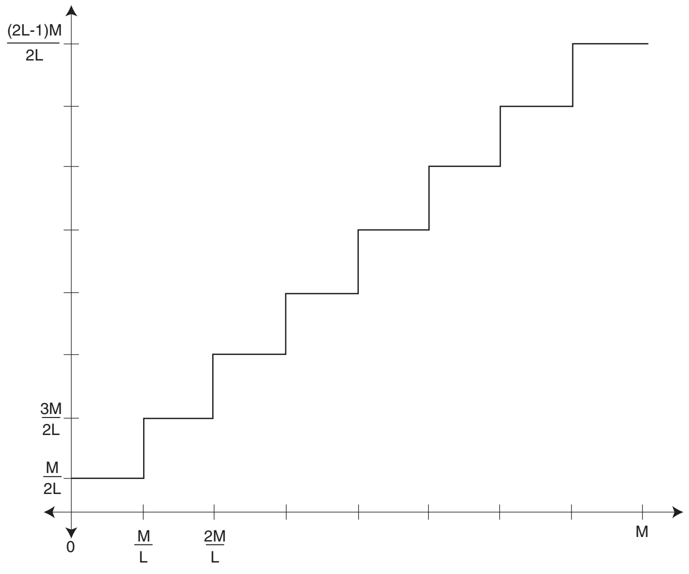


Figure 5.13: A uniform quantizer.

Figure 5.13 shows the reconstruction and transition levels for a uniform quantizer.

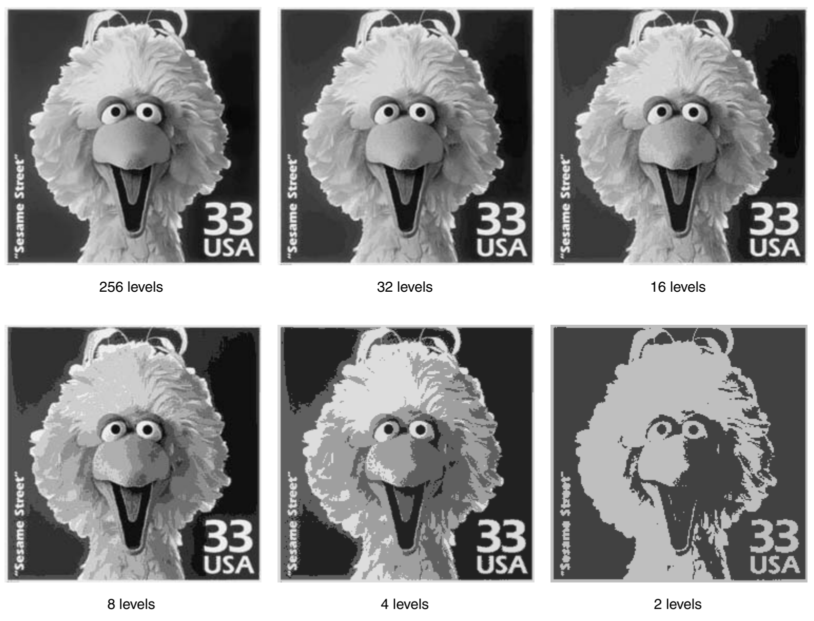


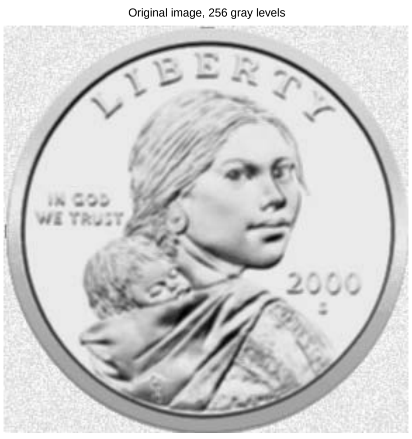
Figure 5.14: Quantizing a gray-level image.

Figure 5.14 show an example of the effects of reducing the number of bits to represent the gray levels in an image using uniform quantization. As fewer quantization levels are used, there is a progressive loss of spatial detail. Furthermore, certain artifacts such as *contouring* (or *false contouring*) begin to appear. These refer to artificial boundaries which become visible due to the large and abrupt intensity changes between consecutive gray levels. Using uniform quantization, images we usually start seeing false contours with 6 or fewer bits/pixel, i.e., about 64 or fewer gray levels. In audio signals, we can often hear distortion/noise in the signal due to quantization with about 8 bits ber sample, or 256 amplitude levels. Of course, these figures depend on the particular image or signal. It turns out that with other quantization methods, as discussed in the following sections, objectionable artifacts can be reduced for a given number of levels.

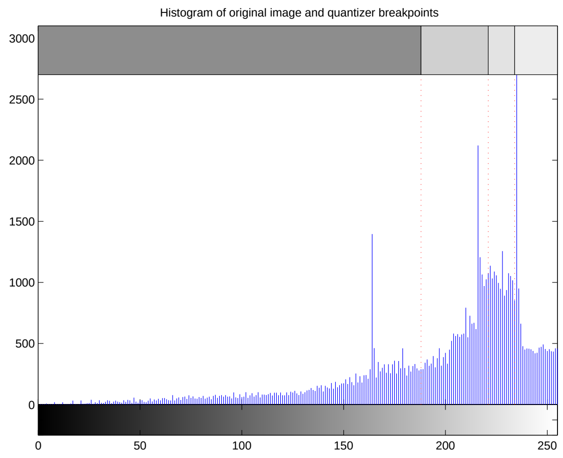
5.5 Nonuniform Quantization

It is natural to expect that the amount of distortion introduced depends on the quantizer, and so one can try to minimize the distortion by choosing a "good" quantizer. For a zero memory quantizer, once we fix the number of levels (or bits per sample), the only choice available in designing a quantizer are the values for the transition and reconstruction levels. With a judicious choice of the quantization levels, however, we may be able to do better than blindly taking the levels uniformly spaced.

For example, if an image never takes on a certain range of gray levels then there is no reason to waste quantization levels in this range of gray levels. One could represent the image more accurately if these quantization levels were instead used to more finely represent gray levels that appeared often in the image.



(a)



(b)

Figure 5.15: (a) Original image. (b) Image histogram.

One way to quantify this idea is to look at the histogram of intensity values of the original signal. This is a plot where the x-axis is the intensity level and the y-axis is the number of samples in the signal that take on that intensity level. The histogram just shows the distribution of intensity levels. Levels that occur often result in large values of the histogram. From the histogram, we can often get a sense of where we should place more quantization levels (places where the histogram is large) and where we should place few levels (where the histogram is small). Figure 5.15b shows the histogram of gray levels for the image of Figure 5.15a. With just 4 levels, by looking at the histogram we might place them at 141,208,227, and 237.

The resulting quantized image is shown in Figure 5.16b which compares very favorably with the image of Figure 5.16a where 4 uniform levels were used.

It turns out that if the distribution is known, then one can formulate the problem of selecting the "optimal" choice of levels (in terms of mean squared error). For the special case that the histogram is flat, for which all gray levels occur equally often, the optimum quantizer turns out to be the uniform quantizer, as one might expect. However, in general the equations for the optimum quantizer are too difficult to solve analytically. One alternative is to solve the equations numerically, and for some common densities, values for the optimal transition and reconstruction levels have been tabulated. Also, various approx-

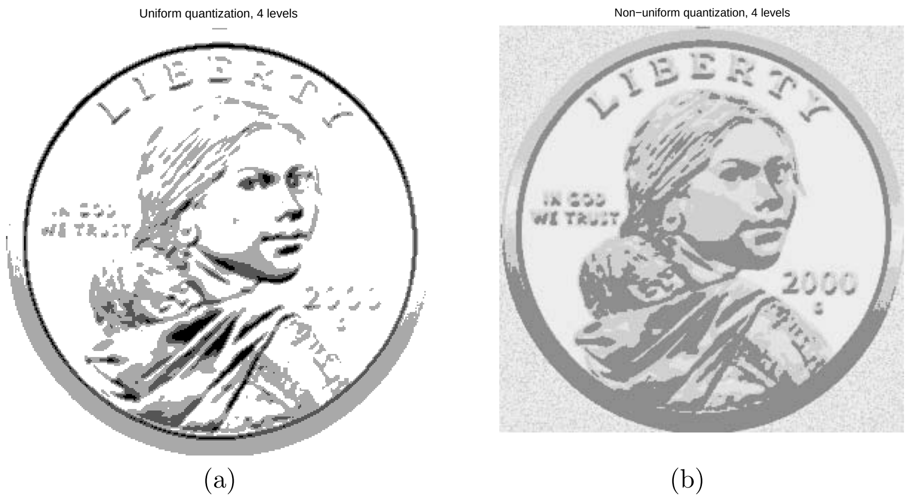

Uniform quantization, 4 levels
Non−uniform quantization, 4 levels
LIBERTY
IN GOD WE TRUST
2000
(a)
(b)

Figure 5.16: (a) Uniformly quantized image. (b) Non-uniformly quantized image

imations and suboptimal (but easy to implement) alternatives to the optimal quantizer have been developed. Although they are suboptimal, these can still give results better than simple uniform quantization.

If quantization levels are chosen to be tailored to a particular image, then when storing the image, one needs to make sure to also store the reconstruction levels so that the image can be displayed properly. This incurs a certain amount of overhead, so that usually one would not choose a different quantizer for each image. Instead, we would choose quantization levels to work for a class of images that we expect to encounter in a given application. If we know something about the images and lighting levels to be expected, then perhaps a non-uniform quantizer may be useful. If not, it may turn out that the most reasonable thing to do is just stick to the uniform quantizer. With increasing capacity and decreasing cost of memory and communication, such issues decrease in importance.

A different rationale for choosing non-uniform quantization levels involves properties of human perception. As discussed in Chapter XX, many human perceptual attributes satisfy Weber's law, which states that we are sensitive to *proportional* changes in the amplitude of a stimulus, rather than in absolute changes. For example, in vision, humans are sensitive to *contrast* not absolute luminance and our contrast sensitivity is proportional to the ambient luminance level. Thus for extremely bright luminance levels there is no reason to have gray levels spaced as close together as for lower luminance levels. Therefore, it makes sense to quantize uniformly with respect to contrast sensitivity, which is logarithmically related to luminance. This provides a rationale for selecting the logarithm of the quantization levels to be equally spaced, rather than the levels themselves being uniform. With this choice there are no claims for even approximating any "optimal" quantizer in general. The justification for ignoring the optimal quantizer is that what is *visually* pleasing does not always correspond to minimizing the mean square error. We will see this again in the next section when we discuss dithering and halftoning.

5.6 Dithering and Halftoning

Dithering As we have seen, coarse quantization often results in the appearance of abrupt discontinuities in the signal. In images, these appear as false contours. One technique used to try to alleviate this problem is called *dithering* or *psuedorandom noise quantization*. The idea is to add a small amount of random noise to the signal *before* quantizing. Sometimes same noise values are stored and then subtracted for display or other purposes, although this is not necessary.

The idea of adding noise intentionally might seem odd. But in the case of dithering, the noise actually serves a useful purpose. Adding noise before quantization has the effect of breaking up false contours by randomly assigning pixels to higher or lower quantization levels. This works because our eyes have limited spatial resolution. By having some pixels in a small neighborhood take on one quantized level and some other pixels take on a different quantized level, the transition between the two levels happens more gradually, as a result of averaging due to limited spatial resolution.

Of course, the noise should not be too large since other features of the signal can also get disrupted by the noise. Usually the noise is small enough to produce changes to only adjacent quantization levels. Even for small amounts of noise, however, there will be some loss in spatial resolution. Nevertheless, the resulting signal is often more desirable. Although dithering in images will introduce some dot-like artifacts, the image is often visually more pleasing since the false contours due to the abrupt transitions are less noticeable.

It is interesting to note that, according to certain quantitative measures, adding noise makes the resulting quantized signal further from the original than quantizing without the noise, as we might expect. For example, one standard measure for the similarity between two signals is the mean squared error. To compute this criterion, one computes the squared difference between corresponding values of two signals, and takes the average of this squared difference. It can be shown that picking the quantization level that is closest to the signal value is the best one can do as far as mean squared error is concerned. Adding noise and then quantizing will only increase the mean squared error, yet we've seen that this can result in a perceptually more pleasing signal. This shows that standard quantitative criteria such as mean square error do not necessarily reflect subjective (perceived) quality.

Halftoning An interesting topic related to dithering is *halftoning*, which refers to techniques used to give a gray scale rendition in images using only black and white pixels. The basic idea is that for an oversampled image, a proper combination of black and white pixels in a neighborhood can give the perception of gray scales due to spatial averaging of the visual system. For example, consider a 4×4 block of pixels. Certain average gray levels for the entire 4×4 block can be achieved even if the 16 pixels comprising the block are allowed to be only black or white. For example, to get an intermediate gray level we need only make sure that 8 pixels are black and 8 are white. Of course, there are a number of specific patterns of 8 black and 8 white pixels. Usually, to allow a gray scale rendition to as fine a spatial resolution as possible, one would avoid using a pattern where all 8 white pixels are grouped together and all 8 black pixels are grouped together. Also, one usually avoids using some fixed pattern for a given gray scale throughout the image since this can often result in the appearance of undesirable but noticeable patterns. A common technique to avoid the appearance of patterns is to select the particular arrangement of black and white pixels for a given gray scale rendition pseudorandomly. Often one repeats a “noise matrix” (also called a halftone matrix or dither matrix). Almost all printed images in books, newspapers, and from laser printers are done using some form of halftoning.