

Appendix A Digital Representation of Sound

About this appendix

This appendix provides a brief explanation of how sound is represented digitally. An understanding of the basic principles introduced here will be helpful in using Raven.

Digital sampling

Before a continuous, time-varying signal such as sound can be manipulated or analyzed with a digital computer, the signal must be *acquired* or *digitized* by a hardware device called an analog-to-digital (A/D) converter, or *digitizer*. The digitizer repeatedly measures or *samples* the instantaneous voltage amplitude of a continuously varying (analog) input signal at a particular sample rate, typically thousands or tens of thousands of times per second ([Figure A.1](#)). In the case of an audio signal, this time-varying voltage is proportional to the sound pressure at a device such as a microphone. The digital representation of a signal created by the digitizer thus consists of a sequence of numeric values representing the amplitude of the original waveform at discrete, evenly spaced points in time.

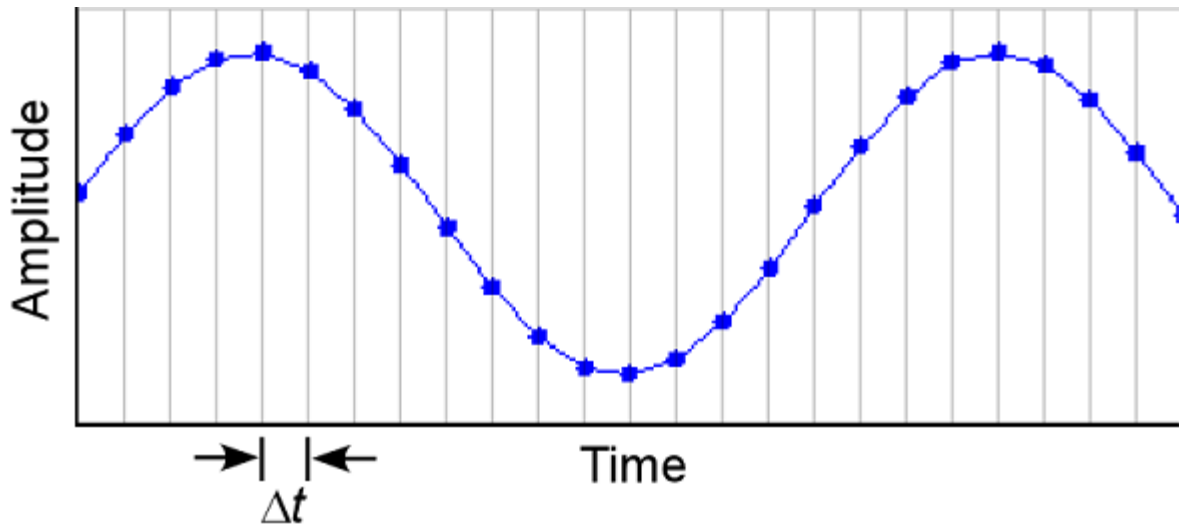


Figure A.1. Sampling to create digital representation of a pure tone signal. The blue sinusoidal curve represents the continuous analog waveform being sampled. Measurements of the instantaneous amplitude of the signal are taken at a sample rate of $1/\Delta t$. The resulting sequence of amplitude values is the digitized signal.

The precision with which the digitized signal represents the continuous signal depends on two parameters of the digitizing process: the rate at which amplitude measurements are made (the *sample rate* or *sampling frequency*), and the number of bits used to represent each amplitude measurement (the *sample size* or *bit depth*).

Sample rate

Raven's Configure Recorder dialog box enables you to choose the sample rate at which a signal is to be digitized. The choices available are determined by the digitizer hardware and the program (called an *audio input plug-in* in Raven) that controls the digitizer; most digitizers have two or more sample rates available. Commercial digital audio applications use sample rates of 44.1 kHz (for audio compact discs) or 48 kHz (for digital audio tape). Once a signal is digitized, its sample rate is fixed.

In order to interpret a sequence of numbers as representing a time-varying signal, one needs to know the sample rate. Thus, when a digitized signal is saved in a file format that is designed for saving sound information (such as AIFF or WAVE), information about the sample rate is saved along with the actual data points comprising the signal.

Aliasing and the Nyquist frequency

The more frequently a signal is sampled, the more precisely the digitized signal represents temporal changes in the amplitude of the original signal. The sample rate that is required to make an acceptable representation of a waveform depends on the signal's frequency. More specifically, the

sample rate must be more than twice as high as the highest frequency contained in the signal. Otherwise, the digitized signal will have frequencies represented in it that were not actually present in the original at all. This appearance of phantom frequencies as an artifact of inadequate sample rate is called *aliasing* (Figure A.2).

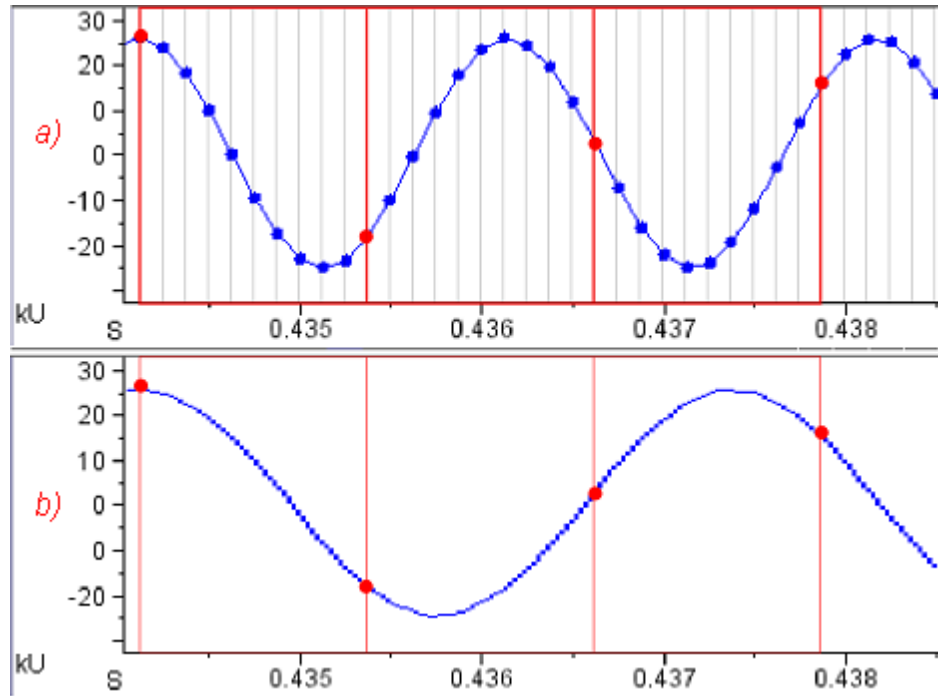


Figure A.2. Aliasing as a result of inadequate sample rate. Vertical lines indicate times at which samples are taken. **(a)** A 500 Hz pure tone sampled at 8000 Hz. The blue sinusoidal curve represents the continuous analog waveform being sampled. There are 16 sample points ($= 8000/500$) in each cycle of the waveform. If the same analog signal were sampled at 800 Hz (red sample points), there would be fewer than two points per cycle, and aliasing would result. **(b)** The aliased waveform that would be represented by sampling the 500 Hz signal at a sample rate of 800 Hz (Nyquist frequency = 400 Hz). Since the original waveform was 100 Hz higher than the Nyquist frequency, the aliased signal is 100 Hz below the Nyquist frequency, or 300 Hz.

The highest frequency that can be represented in a digitized signal without aliasing is called the *Nyquist frequency*, and is equal to half the frequency at which the signal was digitized. The highest frequency shown in a spectrogram or spectrum calculated by Raven is always the Nyquist frequency of the digitized signal. If the only energy above the Nyquist frequency in the analog signal is in the form of low-level, broadband noise, the effect of aliasing is to increase the noise in the digitized signal. However, if the spectrum of the analog signal contains any peaks above the Nyquist frequency, the spectrum of the digitized signal will contain spurious peaks below the Nyquist frequency as a result of aliasing. In

spectrograms, aliasing is recognizable by the appearance of one or more inverted replicates of the real signal, offset in frequency from the original (Figure A.3).

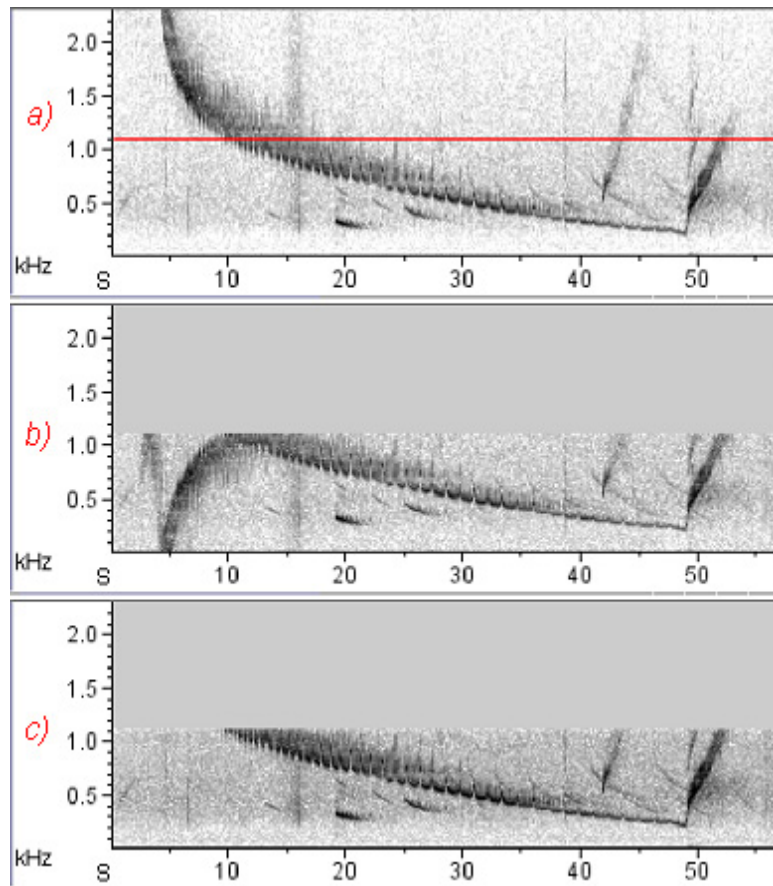


Figure A.3. Appearance of aliasing in spectrogram views. **(a)** Spectrogram of a bearded seal song signal digitized at 11025 Hz. All of the energy in the signal is below the Nyquist frequency (5512.5 Hz); only the lowest 2300 Hz is shown. The red line is at 1103 Hz, one-fifth of the Nyquist frequency. **(b)** The same signal sampled at 2205 Hz (one-fifth of the original rate; Nyquist frequency, 1102.5 Hz) without an anti-aliasing filter. The frequency downswipe in the first ten seconds of the original signal appears in inverted form in this undersampled signal, due to aliasing. **(c)** The same signal as in (b), but this time passed through a low-pass (anti-aliasing) filter with a cutoff of 1100 Hz before being digitized. The downswipe in the first ten seconds of the original signal, which exceeds the Nyquist frequency, does not appear because it was blocked by the filter.

The usual way to prevent aliasing is to pass the analog signal through a low-pass filter (called an *anti-aliasing filter*) before digitizing it, to remove any energy at frequencies greater than the Nyquist frequency. (If the original signal contains no energy at frequencies above the Nyquist

frequency or if it contains only low-level broadband noise, this step is unnecessary.)

Sample size (amplitude resolution)

The precision with which a sample represents the actual amplitude of the waveform at the instant the sample is taken depends on the sample size or number of bits (also called *bit depth*) used in the binary representation of the amplitude value. Some digitizers can take samples of one size only; others allow you to choose (usually through software) between two or more sample sizes. Raven's default audio input device plug-in allows you to choose between 8-bit and 16-bit samples. An 8-bit sample can resolve 256 ($=2^8$) different amplitude values; a 16-bit converter can resolve 65,536 ($=2^{16}$) values. Sound recorded on audio CDs is stored as 16-bit samples. When a sample is taken, the actual value is rounded to the nearest value that can be represented by the number of bits in a sample.

Since the actual analog value of signal amplitude at the time of a sample is usually not precisely equal to one of the discrete values that can be represented exactly by a sample, there is some *digitizing error* inherent in the acquisition process (Figure A.4), which results in *quantization noise* in the digitized signal. The more bits used for each sample, the less quantization noise is contained in the digitized signal. If you listen to a signal digitized with 8-bit samples using high-quality headphones, you can hear the quantization noise as a low-amplitude broadband hiss throughout the recording. Signals digitized with 16-bit samples typically have no detectable hiss. The ratio between the value of the highest amplitude sample that can be represented with a given sample size and the lowest non-zero amplitude is called the *dynamic range* of the signal, and is usually expressed in decibels (dB). The dynamic range corresponds to the ratio in amplitude between the loudest sound that can be recorded and the quantization noise. The dynamic range of a digitized sound is 6 dB/bit.¹

1. The dynamic range of a signal in decibels is equal to $20 \log(A_{max}/A_{min})$, where A_{max} and A_{min} are the maximum and minimum amplitude values in the signal. For a digitized signal, $A_{max}/A_{min} = 2^n$, where n is the number of bits per sample. Since $\log(2^n) = 0.3n$, the dynamic range of a signal is 6 dB/bit.

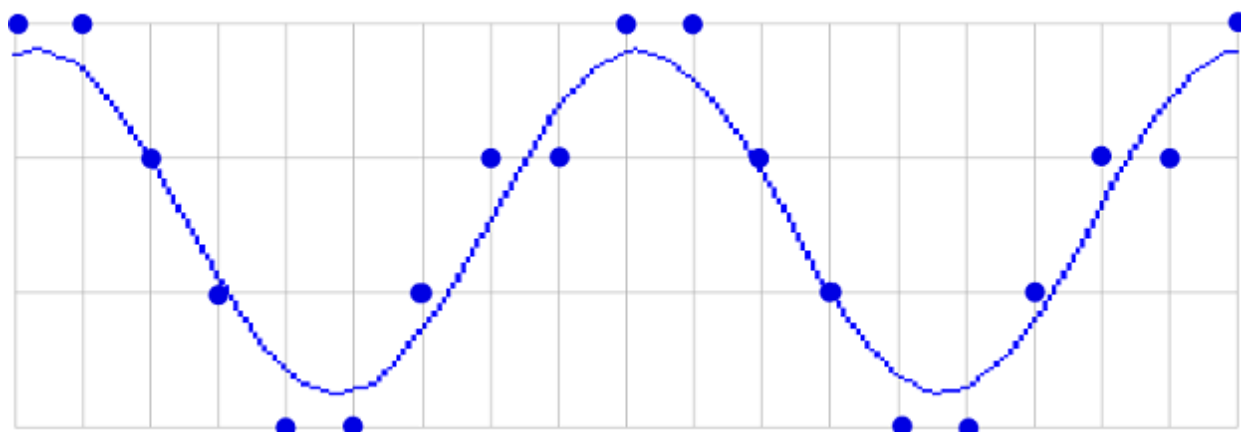


Figure A.4. Digitizing error with a hypothetical 2-bit sample size. 2-bit samples can represent only four different amplitude levels. The blue sinusoidal curve represents the continuous analog waveform being sampled. At each sample time (vertical lines), the actual amplitude levels are rounded to the nearest value that can be represented by a 2-bit sample (horizontal lines). The amplitude values stored for most samples (dots) are slightly different from the true amplitude level of the signal at the time the sample was taken.

Specifying sample sizes when acquiring and saving signals

Raven lets you specify the sample size for a signal when you first acquire it, and again when you save the signal to a file. The set of sample sizes that are available during acquisition is determined by the sound input plug-in that you select (sound acquisition is discussed in [Chapter 4, “Signal Acquisition \(Recording\)”](#)). While Raven is actually working with a signal, samples are always represented by 32-bit floating-point values. When you save a signal with a sample size other than the sample size that the signal had when it was acquired or opened, Raven scales the values to the sample size that you select when you specify the format of the file (saving files is discussed in [Chapter 1, “Getting Started”](#)). For example, if you open a file containing 8-bit samples, and then save the signal with 16-bit samples, each sample value will be multiplied by 2^8 . This scaling ensures that a full-scale value in the original signal is still a full-scale value in the saved signal, even if the sample size differ.



Although you can acquire a signal with 8 bits and then save it with a larger sample size, the saved signal will retain the smaller dynamic range (and audible quantization hiss) of the 8-bit signal. This is because the quantization noise is scaled along with the desired signal when 8-bit signals are scaled to the larger sample size.

Storage requirements

The increased frequency bandwidth obtainable with higher sample rates and the increased dynamic range obtainable with larger samples both come at the expense of the amount of memory required to store a digitized signal. The minimum amount of storage (in bytes) required for a digitized signal is the product of the sample rate (in samples/sec), the sample size (in bytes; one byte equals 8 bits), and the signal duration (seconds). Thus, a 10-second signal sampled at 44.1 kHz with 16-bit (2-byte) precision requires 882,000 bytes ($= 10 \text{ sec} \times 44,100 \text{ samples/sec} \times 2 \text{ bytes/sample}$), or about 861 Kbytes of storage (1 Kbyte = 1024 bytes). The actual amount of storage required for a signal may exceed this minimum, depending on the format in which the samples are stored.

The amount of time that it takes Raven to calculate a spectrogram of a signal depends directly on the number of samples in that signal. Thus, spectrograms take longer to calculate for signals digitized at higher rates. However, the sample size at which a signal is acquired or saved does not affect the speed of spectrogram calculation, because Raven always converts signals to a 16-bit representation for internal operations, even if the signal was initially acquired or saved with a different sample size.