Representing Audio

cs4: Computer Science Bootcamp

Çetin Kaya Koç
http://koclab.cs.ucsb.edu/teaching/cs4
cetinkoc@ucsb.edu

Audio Data

- Audio data comes in a variety of forms
- However, the number of fundamental ways in which sound can be represented is actually fairly small
- The variety of audio file types is due to the fact that there are quite a few approaches to compressing audio data and a number of different ways of packaging the data

Analog-to-Digital Conversion

- Sound consists of audible variation in air pressure
- Microphones convert variation in air pressure into an "analog" voltage
- To represent sound digitally, we must convert this analog voltage into a series of numbers representing its amplitude
- This process is known as **analog-to-digital conversion**

Analog-to-Digital Conversion

- Audio data consisting of such numbers is said to be in pulse code modulation (PCM) format
- Audio data is often stored in other formats, usually in order to compress it, but it almost always starts off in PCM format
- The air pressure variation, and the corresponding voltage produced by a microphone, is continuous in two-dimensions
- The values vary continuously, and they exist at every point in time

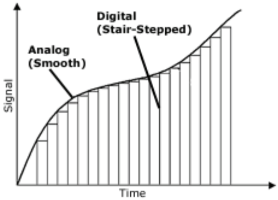
Sampling, Quantization and Resolution

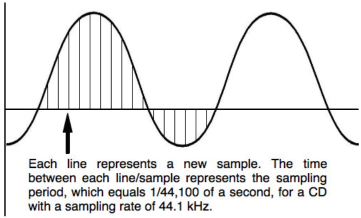
- However, a computer cannot directly represent a continuous signal
- Instead, it must measure the signal at a finite set of discrete times
- This process is known as **sampling**
- Furthremore, it must make use of a finite number of discrete amplitude levels
- This process is known as **quantization**
- The number of levels used in quantization is called **resolution**

Resolution

- The resolution is usually expressed in bits
- A system with a resolution of 8 bits makes use of $2^8 = 256$ levels
- A system with 16 bit resolution makes use of $2^{16} = 65,536$ levels
- The sampling rate and resolution determine the quality of the digital representation of the sound
- "CD-quality" sound has a minimum resolution of 16 bits and a minimum sampling rate of 44,100 samples per second

Resolution

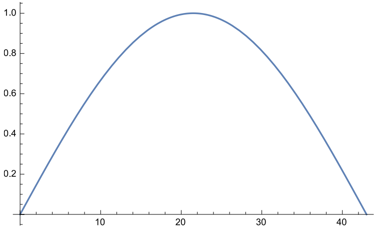


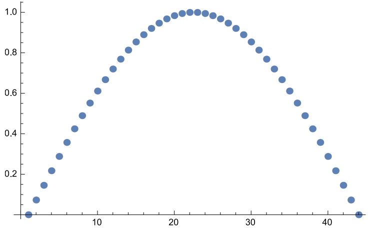


- Therefore, within exactly 1 second, we have 44,100 integers each of which is 2 bytes
- If the audio we are sampling is long 0.001 seconds (1 milliseconds), we will have 44 integers each of which is 2 bytes (between 0 and 65,535)

Sampling Example

- For example, the Sine function between 0 and π can be sampled to 44 values as





- We take these 44 values, which are real numbers between 0 and 1, and quantize them to integers between 0 and 65,535

Sampling Example

- A small integer (close to 0) represents a smaller real number (close to 0), while a large integer (close to 65,535) represents a larger real number (close to 1)
- The list in decimal:

$$\{0, 4783, 9542, 14249, 18880, 23411, 27817, 32074, 36160, 40053, 43733, 47179, 50374, 53299, 55941, 58283, 60315, 62025, 63405, 64445, 65142, 65492, 65492, 65142, 64445, 63405, 62025, 60315, 58283, 55941, 53299, 50374, 47179, 43733, 40053, 36160, 32074, 27817, 23411, 18880, 14249, 9542, 4783, 0\}$$

- The list in hexadecimal:

$$\{0000, 12af, 2546, 37a9, 49c0, 5b73, 6ca9, 7d4a, 8d40, 9c75, aad5, b84b, c4c6, d033, da85, e3ab, eb9b, f249, f7ad, fbbd, fe76, ffd4, ffd4, fe76, fbbd, f7ad, f249, eb9b, e3ab, da85, d033, c4c6, b84b, aad5, 9c75, 8d40, 7d4a, 6ca9, 5b73, 49c0, 37a9, 2546, 12af, 0000\}$$

Channels

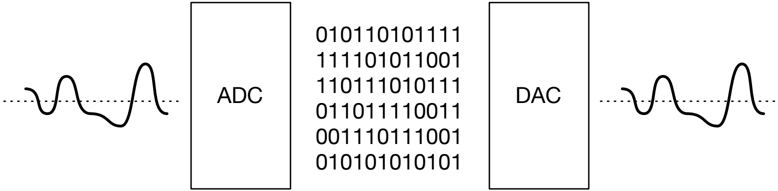
- A single stream of sound, such as that from an ordinary monaural recording, constitutes one channel
- Stereo requires two channels
- Quadriphonic music requires four channels
- Recordings made in professional music studios may have many channels prior to mixing, one for each instrument and singer
- In practice, the most commonly used values are 1 and 2

Raw Audio Data

- Once we have a set of numbers for a given duration of sound, we are essentially done
- We have obtained digital audio: the collection of numbers, each of which shows quantized amplitude at a given time
- The number of numbers shows how much time this sound takes, for example, 22,050 numbers imply we have 0.5 second of audio
- Definition: Herz or Hz $\rightarrow$ The number of samples per second

Raw Audio Data

- These numbers can be stored or transmitted as digital audio
- They can be "played" by converting back to analog



Raw Audio Data

- However, audio data takes up a lot of memory space
- 1 second of CD-quality of music has 44,100 samples
- At 16-bit resolution, this implies 88,200 bytes
- 1 minute: $60 \times 44,100 \times 2 = 5,292,000$ bytes
- 4 minutes: $60 \times 4 \times 44,100 \times 2 = 21,168,000$ bytes
- 1 hour: $60 \times 60 \times 5,292,000 = 317,520,000$ bytes
- Multiply these numbers by 2 if stereo is used

Hi-Res Audio Data

- High-resolution (Hi-res) audio has 24-bit resolution
- It is also sampled at a higher rate: 96,000 or 192,000 Herz*
- Therefore, 1 second of hi-res audio would take $192,000 \times 3 = 576,000$ bytes
- 1 minute: $60 \times 192,000 \times 3 = 34,560,000$ bytes
- 1 hour: $60 \times 60 \times 192,000 \times 3 \approx 2$ billion bytes

- Can you tell the difference?
- Guardian Article, Aug 21, 2014

Compressing Digital Audio Data

- The standard (16-bit & 44,100 Hz) or Hi-res (24-bit & 96,000 or 192,000 Hz) digital audio takes a lot of memory space
- Therefore, compression is required

Compressing Digital Audio Data

- Digital audio has been around for a long time
- Compact Disc (CD) technology was invented in 1970s
- A CD stores about 700m bytes of music
- A DVD can store 4g bytes per layer
- Even today's speed and storage capacity, these are huge numbers

- Compression made digital audio ubiquitous (cheap or free)
- It also bankrupted Tower Records!