Chapter 3. Multimedia Data Representations

3.1. Basics of Digital Audio

Digitization of Sound
Introduction to MIDI

- *Reference:* K.C. Pohlmann, "Principles of Digital Audio", 3rd ed., McGraw-Hill, 1995.
- *Reference:* Chapter 3 of Steinmetz and Nahrstedt

3.1.1 Digitization of Sound

Facts about Sound

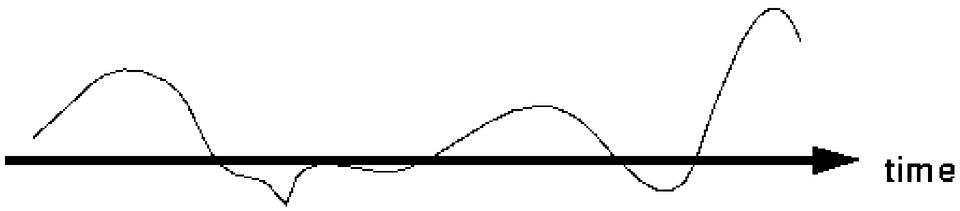
- Sound is a continuous wave that travels through the air.
- The wave is made up of pressure differences. Sound is detected by measuring the pressure level at a location.
- Sound waves have normal wave properties (reflection, refraction, diffraction, etc.).
- Human ears can hear in the range of 16 Hz to about 20 kHz. This changes with age.
  Hence, wavelengths vary from 21.3 m to 1.7 cm.
- The intensity of sound can be measured in terms of Sound Pressure Level (SPL) in decibels (dBs).

  intensity level = 10 log (P / P0) dB,

  where P and P0 are values of acoustic power, and P0 will deliver an intensity of sound at the threshold of hearing, which is 10^{-12} W/m^2 (watts per square meter).

Digitization in General

- Microphones, video cameras produce *analog signals* (continuous-valued voltages)



- To get audio or video into a computer, we must *digitize* it (convert it into a stream of numbers)

  So, we have to understand *discrete sampling* (both time and voltage)

- *Sampling* -- divide the horizontal axis (the time dimension) into discrete pieces. Uniform sampling is ubiquitous.

  Quantization -- divide the vertical axis (signal strength) into pieces. Sometimes, a non-linear function is applied.

  - 8 bit quantization divides the vertical axis into 256 levels. 16 bit gives you 65536 levels.

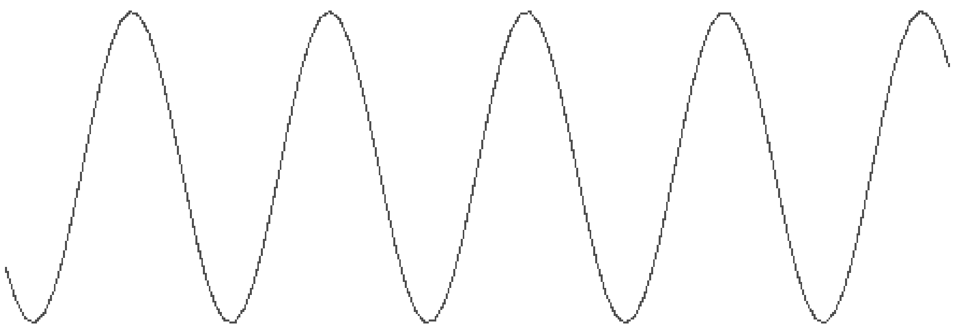
Digitizing Audio

- Questions for producing digital audio (Analog-to-Digital Conversion):

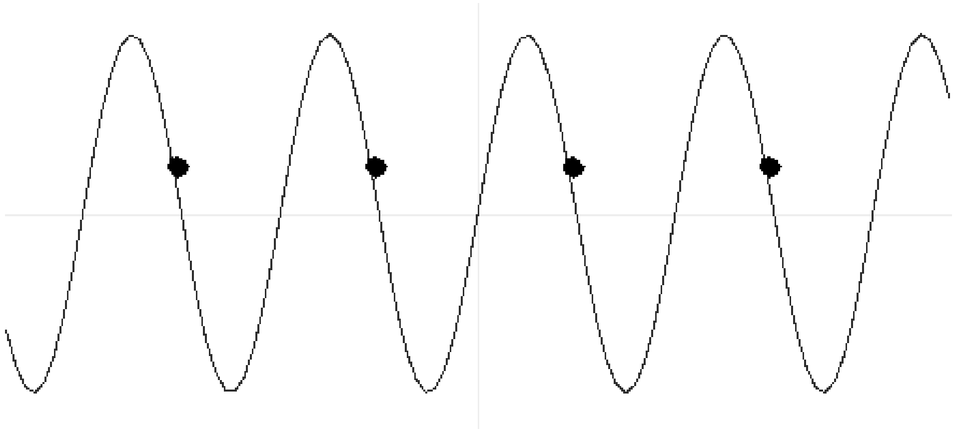
  1. How often do you need to sample the signal?
  2. How good is the signal?
  3. How is audio data formatted?

Nyquist Theorem

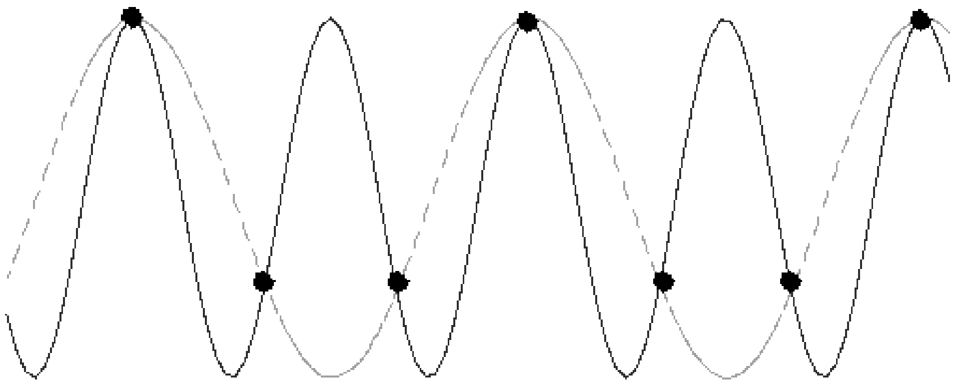
- Suppose we are sampling a sine wave. How often do we need to sample it to figure out its frequency?



- If we sample only once per cycle, we may think the signal is a constant.



- If we sample at another low rate, e.g., 1.5 times per cycle, we may think it's a lower frequency sine wave --> *Alias*



- **Nyquist rate** -- It can be proved that a bandwidth-limited signal can be fully reconstructed from its samples, if the sampling rate is at least twice of the highest frequency in the signal.

Signal to Noise Ratio (SNR)

- In any analog system, some of the voltage is what you want to measure (*signal*), and some of it is random fluctuations (*noise*).

- Ratio of the power of the two is called the *signal to noise ratio* (*SNR*). SNR is a measure of the quality of the signal.

- SNR is usually measured in decibels (*dB*).

$$SNR = 10\log\frac{V_{signal}^2}{V_{noise}^2} = 20\log\frac{V_{signal}}{V_{noise}}$$

Signal to Quantization Noise Ratio (SQNR)

- The precision of the digital audio sample is determined by the number of bits per sample, typically

8 or 16 bits.

The quality of the quantization can be measured by the Signal to Quantization Noise Ratio (SQNR).

- The *quantization error* (or *quantization noise*) is the difference between the actual value of the analog signal at the sampling time and the nearest quantization interval value.

  The largest (worst) quantization error is half of the interval.

- Given N to be the number of bits per sample, the range of the digital signal is - 2 exp (N-1) to 2 exp (N-1).

$$SQNR = 20\log\frac{V_{signal}}{V_{quan-noise}} = 20\log\frac{2^{N-1}}{\frac{1}{2}} = N \times 20\log 2 = 6.02N\ (dB)$$

  In other words, each bit adds about 6 dB of resolution, so 16 bits enable a maximum SQNR = 96 dB.

  (** The above is for the worst case. Assume the input signal is sinusoidal, and the quantization error is statistically independent and its magnitude is uniformly distributed between 0 and half of the interval,
  SQNR = 6.02N + 1.76. [Pohlmann95, p. 37])

Linear and Non-linear Quantization

- Samples are typically stored as raw numbers (*linear format*), or as logarithms (*u-law* (or *A-law* in Europe)).
  - Logarithmic quantization approximates *perceptual non-uniformity*.

Typical Audio Formats

- Popular audio file formats include .au (Unix workstations), .aiff (MAC, SGI), .wav (PC, DEC workstations)

- A simple and widely used audio compression method is Adaptive Delta Pulse Code Modulation (ADPCM). Based on past samples, it predicts the next sample and encodes the difference between the actual value and the predicted value.

Audio Quality vs. Data Rate

Quality	Sample Rate (KHz)	Bits per Sample	Mono/ Stereo	Data Rate (if Uncompressed)	Frequency Band
Telephone	8	8	Mono	8 KBytes/sec	200-3,400 Hz

AM Radio	11.025	8	Mono	11.0 KBytes/sec	
FM Radio	22.050	16	Stereo	88.2 KBytes/sec	
CD	44.1	16	Stereo	176.4 KBytes/sec	20-20,000 Hz
DAT	48	16	Stereo	192.0 KBytes/sec	20-20,000 Hz
DVD Audio	192	24	Stereo	1,152.0 KBytes/sec	20-20,000 Hz

- Telephone uses *u-law* encoding, others use linear. So the dynamic range of digital telephone signals is effectively 13 bits rather than 8 bits.
- CD quality stereo sound --> 10.6 MB / min.

Synthetic Sounds

- FM (Frequency Modulation) Synthesis -- used in low-end Sound Blaster cards, OPL-4 chip

- Wavetable synthesis -- wavetable generated from sound waves of real instruments

  - FM Synthesis is good for creating new sounds. Wavetables can store sounds of existing instruments nicely.
  - The wavetables are stored in memory on the sound card and they can be manipulated by software.
  - To save memory space, a variety of special techniques, such as sample looping, pitch shifting, mathematical interpolation, and polyphonic digital filtering can be applied.

Further Exploration

CD audio file formats

3.1.2 Introduction to MIDI (Musical Instrument Digital Interface)

Definition of MIDI: a protocol that enables computer, synthesizers, keyboards, and other musical device to communicate with each other.

1. Terminologies:

Synthesizer:

- It is a sound generator (various pitch, loudness, tone color).
- A good (musician's) synthesizer often has a microprocessor, keyboard, control panels, memory, etc.

Sequencer:

- It can be a stand-alone unit or a software program for a personal computer. (It used to be a storage server for MIDI data. Nowadays it is more a software *music editor* on the computer.)
- It has one or more MIDI INs and MIDI OUTs.

Track:

- Track in sequencer is used to organize the recordings.
- Tracks can be turned on or off on recording or playing back.

Channel:

- MIDI channels are used to separate information in a MIDI system.
- There are 16 MIDI channels in one cable.
- Channel numbers are coded into each MIDI message.

Timbre:

- The quality of the sound, e.g., flute sound, cello sound, etc.
- Multitimbral -- capable of playing many different sounds at the same time (e.g., piano, brass, drums, etc.)

Pitch:

- musical note that the instrument plays

Voice:

- Voice is the portion of the synthesizer that produces sound.
- Synthesizers can have many (16, 20, 24, 32, 64, etc.) voices.
- Each voice works independently and simultaneously to produce sounds of different timbre and pitch.

Patch:

- the control settings that define a particular timbre.

2. Hardware Aspects of MIDI

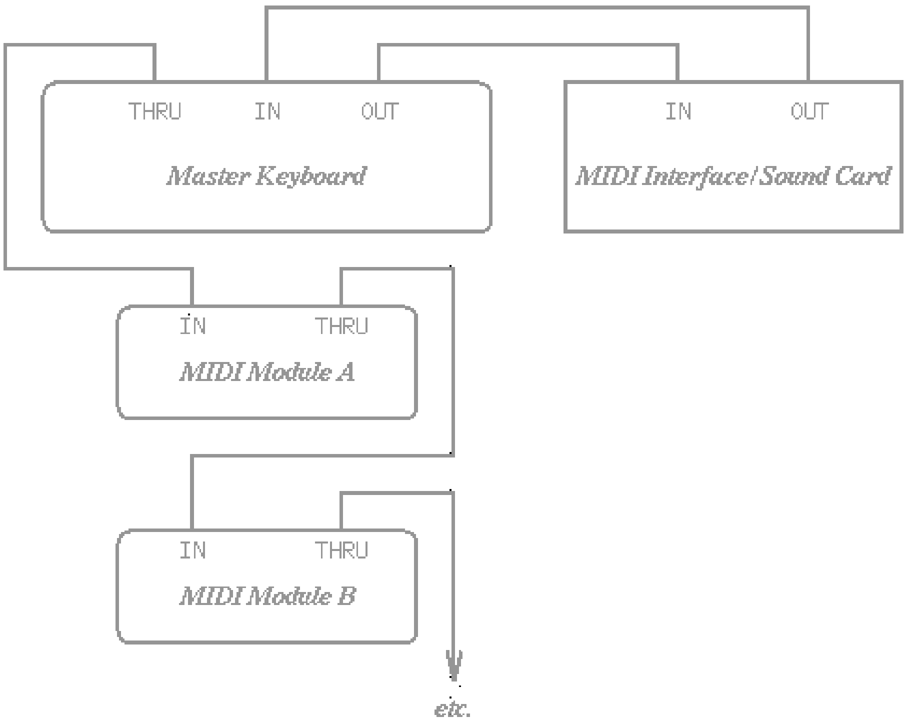
MIDI connectors:

-- three 5-pin ports found on the back of every MIDI unit

- MIDI IN: the connector via which the device receives all MIDI data.
- MIDI OUT: the connector through which the device transmits all the MIDI data it generates itself.
- MIDI THROUGH: the connector by which the device echoes the data receives from MIDI IN.

Note: It is only the MIDI IN data that is echoed by MIDI through. All the data generated by device itself is sent through MIDI OUT.

A Typical MIDI Sequencer Setup:



- MIDI OUT of synthesizer is connected to MIDI IN of sequencer.
- MIDI OUT of sequencer is connected to MIDI IN of synthesizer and "through" to each of the additional sound modules.
- During recording, the keyboard-equipped synthesizer is used to send MIDI message to the sequencer, which records them.
- During play back: messages are send out from the sequencer to the sound modules and the synthesizer which will play back the music.

3. MIDI Messages

-- MIDI messages are used by MIDI devices to communicate with each other.

Structure of MIDI messages:

- MIDI message includes a status byte and up to two data bytes.
- Status byte
  - The most significant bit of status byte is set to 1.
  - The 4 low-order bits identify which channel it belongs to (four bits produce 16 possible channels).
  - The 3 remaining bits identify the message.
- The most significant bit of data byte is set to 0.

Classification of MIDI messages:

```
                                                        ----- voice messages
                   ---- channel messages -----|
                  |                             ----- mode messages
                  |
MIDI messages ----|
                  |                             ---- common messages
                   ----- system messages -----|---- real-time messages
                                                ---- exclusive messages
```

A. Channel messages:

-- messages that are transmitted on individual channels rather that globally to all devices in the MIDI network.

A.1. Channel voice messages:

- Instruct the receiving instrument to assign particular sounds to its voice
- Turn notes on and off
- Alter the sound of the currently active note or notes

Voice Message	Status Byte	Data Byte1	Data Byte2
Note off	&H8x	Key number	Note Off velocity
Note on	&H9x	Key number	Note on velocity
Polyphonic Key Pressure	&HAx	Key number	Amount of pressure
Control Change	&HBx	Controller number	Controller value
Program Change	&HCx	Program number	None
Channel Pressure	&HDx	Pressure value	None
Pitch Bend	&HEx	MSB	LSB

Notes: ‘x’ in status byte hex value stands for a channel number.

Example: a Note On message is followed by two bytes, one to identify the note, and on to specify the velocity.
To play note number 80 with maximum velocity on channel 13, the MIDI device would send these three hexadecimal byte values: &H9C &H50 &H7F

A.2. Channel mode messages: -- Channel mode messages are a special case of the Control Change message (&HBx or 1011nnnn). The difference between a Control message and a Channel Mode message, which share the same status byte value, is in the first data byte. Data byte values 121 through 127 have been reserved in the Control Change message for the channel mode messages.

- Channel mode messages determine how an instrument will process MIDI voice messages.

1st Data Byte	Description	Meaning of 2nd Data Byte
&H79	Reset all controllers	None; set to 0
&H7A	Local control	0 = off; 127 = on
&H7B	All notes off	None; set to 0

&H7C	Omni mode off	None; set to 0
&H7D	Omni mode on	None; set to 0
&H7E	Mono mode on (Poly mode off)	**
&H7F	Poly mode on (Mono mode off)	None; set to 0

** if value = 0 then the number of channels used is determined by the receiver; all other values set a specific number of channels, beginning with the current basic channel.

B. System Messages:

- System messages carry information that is not channel specific, such as timing signal for synchronization, positioning information in pre-recorded MIDI sequences, and detailed setup information for the destination device.

B.1. System real-time messages:

- messages related to synchronization

System Real-Time Message	Status Byte
Timing Clock	&HF8
Start Sequence	&HFA
Continue Sequence	&HFB
Stop Sequence	&HFC
Active Sensing	&HFE
System Reset	&HFF

B.2. System common messages:

- contain the following unrelated messages

System Common Message	Status Byte	Number of Data Bytes
MIDI Timing Code	&HF1	1
Song Position Pointer	&HF2	2
Song Select	&HF3	1
Tune Request	&HF6	None

B.3. System exclusive message:

- (a) Messages related to things that cannot be standardized, (b) addition to the original MIDI specification.
- It is just a stream of bytes, all with their high bits set to 0, bracketed by a pair of system exclusive start and end messages (&HF0 and &HF7).

4. General MIDI

- MIDI + Instrument Patch Map + Percussion Key Map --> a piece of MIDI music sounds the same anywhere it is played
  - Instrument patch map is a standard program list consisting of 128 patch types.
  - Percussion map specifies 47 percussion sounds.
  - Key-based percussion is always transmitted on MIDI channel 10.

- Requirements for General MIDI Compatibility:
  - Support all 16 channels.
  - Each channel can play a different instrument/program (multitimbral).
  - Each channel can play many voices (polyphony).
  - Minimum of 24 fully dynamically allocated voices.

Appendix

A1. General MIDI Instrument Patch Map

A2. General MIDI Percussion Key Map

Further Exploration

Try some good sources for locating internet sound/music materials at

- A tutorial on MIDI and wavetable music synthesis

- YAHOO's Multimedia:Sound Page

Last Updated: ***1/18/2000***