Computer Audio and Music

Perry R. Cook
Princeton Computer Science
(also Music)

Music/Sound Overview

- Basic Audio storage/playback (sampling)
- Human Audio Perception
- Sound and Music Compression and Representation
- Sound Synthesis
- Music Control and Expression

Waveform Sampling and Playback

- Sample and Hold
  Sample Rate vs. Aliasing
- Quantize
  Word Size vs. Quantization Noise
- Reconstruct: Hold and Smooth (filter)

Waveform Sampling: Quantization

Quantization Introduces Noise

Quantizer

\[ x \rightarrow x + \varepsilon \]

\[ \varepsilon \] (error)
Compression and Representation (Why Bother??)

So Many Bits, So Little Time (Space)
- CD audio rate: $2 \times 2 \times 8 \times 44100 = 1,411,200$ bps
- CD audio storage: $10,584,000$ bytes / minute
- A CD holds only about 70 minutes of audio
- An ISDN line can only carry $128,000$ bps
- Even a cable modem might carry only $1$ Mbps

Security: Best representation removes all recognizable about the original sound

Graphics people get all the bandwidth, cycles, memory

Expression, composition, interaction wanted too!

Views of Sound

- Sound is Perceived: Perception-Based
  Psychoacoustically Motivated Compression

- Sound is Produced: Production-Based
  Physics/Source Model Motivated Compression

- Music(Sound) is Performed/Published/Represented: Event-Based Compression

- Sound is a Waveform / Statistical Distribution / etc.
  (these are not very good ideas in general,
  unless we get lucky (LPC))

Psychoacoustics

Human sound perception:

- Ear: receive 1-D waves
- Cochlea: convert to frequency dependent nerve firings
- Nerves: transmit information to Auditory cortex:
  further refine time & frequency information
- Brain: Higher level cognition, object formation, interpretation

Perceptual Models

Exploit masking, etc., to discard perceptually irrelevant information.

- Example: Quantize soft sounds more accurately, loud sounds less accurately

Benefits: Generic, does not require assumptions about what produced the sound

Drawbacks: Highest compression is difficult to achieve
Production Models

*Build a model of the sound production system, then fit the parameters*

- Example: If signal is speech, then a well parameterized vocal model can yield highest quality and compression ratio

Benefits: *Highest possible compression*

Drawbacks: Signal source(s) must be
- assumed, known, or identified

Audio Compression

*Classical Data Compression View:*

*Take advantage of*

- Redundancy/Correlation
- Statistics (Local/Global)
- Assumptions / Models

*Problem: Much of this doesn’t work directly on sound waveform data*

Transform (Subband) Coders

*Split signal into frequency subbands, then allocate bits to regions adaptively, based on where ear is most sensitive*

Lossless (variable bit rate & comp. ratio)

Lossy (fixed rate and ratio) MP3

Production Models

*Build a parametric model of the production system, then either*

Fit the parameters to a given signal
Use signal processing techniques to extract parameters

**Drive the parameters directly (no encode/decode)**

Examples:
- Rule system to drive speech synthesizer
- MIDI file to drive music synthesizer
**Speech Coders (production)**

Assume speech is produced by a source-filter system (vocal folds/noise + vocal tract tube)

Identify filter, type of source, then code parameters

Takes advantage of slowly varying nature of vocal tract shape and other speech parameters

---

**Future: Multi-Model Parametric Compressors?**

Analysis front end identifies source(s)

Audio is (separated and) sent to optimal model(s)

Benefits:
- High compression
- Other knowledge

Drawbacks:
- We don’t know how to do all this yet

---

**Sound Analysis and Classification**

Cochlear Modeling

Multi-feature analysis (Tzanetakis)

Segmentation, Classification, Annotation, Thumbnails

---

**MIDI and Other ‘Event’ Models**

Musical Instrument Digital Interface

Represents Music as Notes and Events and uses a synthesis engine to “render” it.

An Edit Decision List (EDL) is another example.

A history of source materials, transformations, and processing steps is kept. Operations can be undone or recreated easily. Intermediate non-parametric files are not saved.

Speaking of MIDI and scores, a brief aside on Computing History:
History of Programmable Machines

- First “programmable system” was the early printing process developed in China circa 800 C.E.
  - First “program” was perhaps Chinese translation of Buddhist Canon (the Tipitaka)

History (cont.)

- Gutenberg’s Printing Press (circa 1450)
  - Main Contribution:
    - Just a few “basic instructions” (smaller alphabet size) suffice.

Jacquard’s Loom (circa 1810)

- Punched Cards stored program for weaving patterns.

Wait: Gutenberg 1450
Jacquard 1810

Are we missing something here?????

Nothing happened in 350 years?

Huygen’s Pendulum Clock (circa 1650)

- Main Contribution:
  - Timing, “clock ticks” increase accuracy
History (cont.)

• Musical Machines:
  - Barrel Organs (1500!)
  - Music boxes (between)
  - Player Pianos (c. 1700)

• Main Contributions:
  - Drive cylinder or disk with "pins" (bits!!) which play notes at the right time
  - Change disk -> change song (programmable!)

History (cont.)

• Jacquard's Loom (circa 1810)

• Punched Cards stored program for weaving patterns.

History (cont.)

• Charles Babbage (1822-64):
  - Input -- Punched Cards
  - Hardware -- general-purpose mechanical mathematical system
  - (Analytical Engine) -- never built
  - Could be programmed
    • punched card could say: "Go back 5 punched cards"
    • Instructions could be Executed repeatedly, or in different order.

The Modern Computer

• Basic Idea still the same:
  A machine that can execute certain instructions.

• Machine instructions represented by sequences of 0's and 1's
  (Machine Language)

• Instructions stored in Memory

von Neumann (1945)
Princeton, NJ
Anyway, Event Based Music Representation

*MIDI and Other Scorefiles*
- A Musical Score is a very compact representation of music
- Even the score itself can be compressed further

**Benefits:**
- Highest possible compression
- Encodes “expression”

**Drawbacks:**
- Cannot guarantee the “performance”
- Cannot assure the quality of the sounds
- Cannot make arbitrary sounds (yet)

---

**MIDI**

MIDI Serial Data Transmission
31.25 kbaud, 1 Start, 8 Data, 1 Stop Bit
Asynchronous Bytes

Typical Message: 10010000 00111100 01000000

Meaning:
- 10010000: NoteOn / NoteOff
- 00111100: Velocity=64
- 01000000: Chord=0 (Middle C)

---

**Event Based Representation**

*Enter General MIDI*
- Guarantees a base set of instrument sounds,
- and a means for addressing them,
- but doesn’t guarantee any quality

**Better Yet, Downloadable Sounds**
- Download samples for instruments
- **Benefits:** Does more to guarantee quality
- **Drawbacks:** Samples aren’t reality

---

**Event Based Representation**

*Downloadable Algorithms*
- Specify the algorithm, the synthesis engine runs it, and we just send parameter changes
- Part of “Structured Audio” (MPEG4)

**Benefits:**
- Can upgrade algorithms later
- Can implement scalable synthesis

**Drawbacks:**
- Different algorithm for each class of sounds
  (but can always fall back on samples)
Physical Modeling for Music

- Strings (plucked, struck, bowed)
- Winds (clarinet, flute, brass), voice
- Plates, membranes, bar percussion
- Shakers, scrapers
- The Voice

Physical Modeling: the “Real World”

Sounds Effects (PhOLISE)

Synthesizing Solids

O’Brien, Cook, and Essl

SIGGRAPH 01

Synthesizing Sounds From Physically Based Motion

James F. O’Brien, U.C. Berkeley
Perry R. Cook, Princeton University
Georg Essl, Princeton University

ACM SIGGRAPH 2001

Composition and Creation

Garton “Rough Raga Riffs”
Lansky “mild und leise”

Music for Unprepared Piano
Bargar, Choi, Betts, Cook

Expression and Control

Cook/Morrill Trumpet

Other Controllers

Trueman: BoSSA
### PICOs (musical and “real-world” sonic controllers)

<table>
<thead>
<tr>
<th>K-Frog</th>
<th>J-Mug</th>
<th>P-Pedal</th>
<th>PhilGlas</th>
<th>P-Grinder</th>
<th>T-shoe</th>
<th>T-bourine</th>
<th>Pico Glove</th>
<th>P-Ray’s Cafe</th>
</tr>
</thead>
</table>

**Audio and Computer Music**

**Questions?**