
An overview of digital audio

Dan Ellis
dpwe@icsi.berkeley.edu
International Computer Science Institute, Berkeley CA

Goal:
Survey techniques, provide discussion framework

Outline:

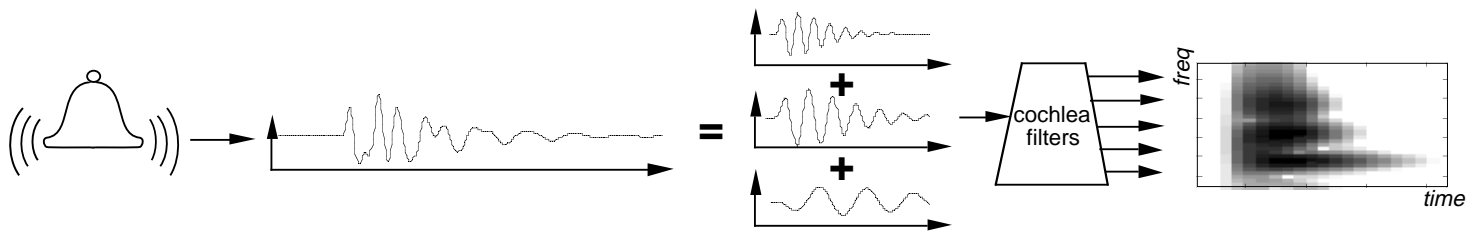
- 1** Sound, hearing & audio processing
- 2** Representation
- 3** Synthesis
- 4** Processing & modification
- 5** Analysis



1

Sound & Hearing

- **Sound= 1-D time-variation of air pressure, $P(t)$**
- **.. decomposed by cochlea into multiple frequency bands**
→ **2-D representation, $I(t,f)$**

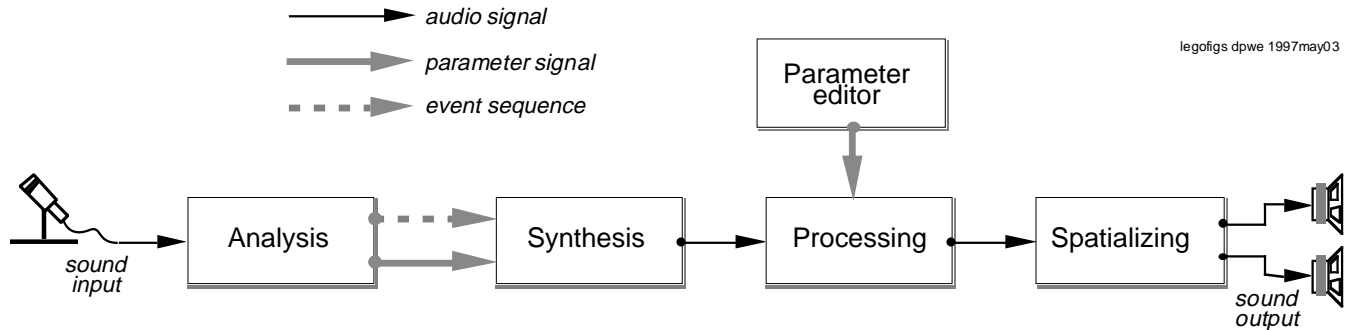


- **Basic sensitivity imposed by cochlea for time, frequency, level, dynamic range**
- **Higher auditory system extracts 'useful info':**
→ **reflects *ecological* constraints**



Audio processing

- **Dataflow diagrams useful for sound signal processing:**



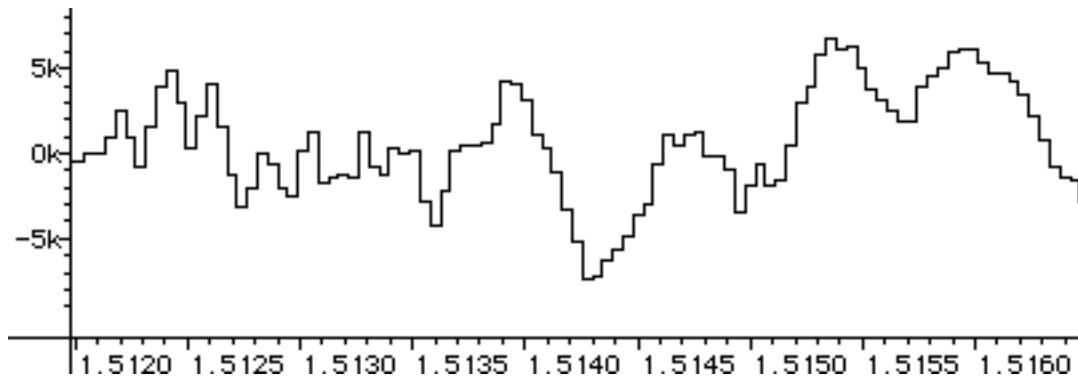
- **Typically several distinct data 'types':**
 - audio signals $a(t)$
 - parameter ('control') signals $K(T)$
 - event sequences $\{\tau_{Ei}\}$



2

Audio representation

- **Sampled waveforms are ubiquitous**
 - represent the 1-D pressure waveform as a sequence of values at regular intervals



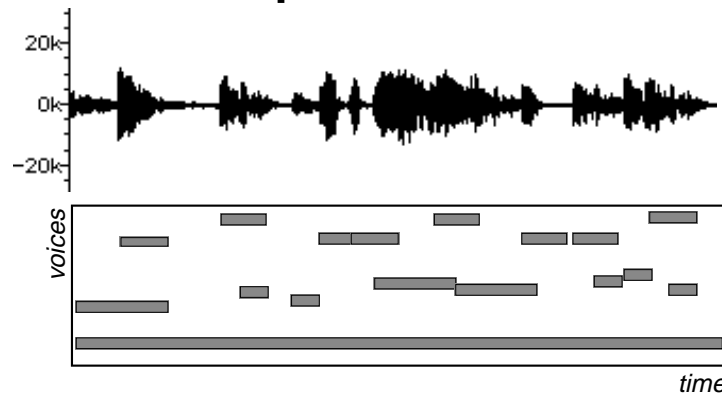
- **Tradeoff between quality and size via:**
 - sampling rate (\rightarrow bandwidth, high frequency)
 - quantization (\rightarrow noise floor)
$$\text{samples/sec} \times \text{bits/sample} = \text{data rate, size}$$



Compressed audio representations

- **Save bits on quantization**
 - variable quantization (mu-law, ADPCM)
 - noise shaping & ‘perceptual coding’
- **Parametric models use stronger constraints**
 - approximate signal as output of a process
 - how to extract/find best parameters?
 - size vs. quality vs. complexity

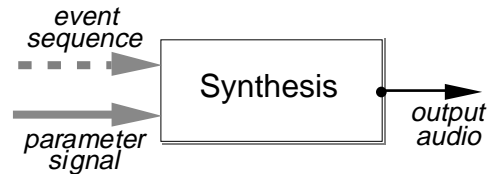
- **Event decomposition**



- encode high-level temporal structure
- e.g. MIDI, MPEG-4
- implies a **synthesis method...**



Synthesis



Creating an audio signal from control inputs

- **Issues:**
 - fidelity / richness
 - flexibility / control ‘knobs’
 - cost in complexity (CPU) & data size (store)
- **Mimic real signal, or just make a new one?**
 - abstract level of correspondence
- **Techniques:**
 - signal models:
 - sampling
 - sinusoid (plus...) models
 - nonlinear algorithms e.g. FM
 - source models:
 - source-filter & LPC
 - waveguide & other physical models



Synthesis 2: Signal models

- **Sampled waveforms**

fidelity: excellent (but.. unnatural repetition?)

controls: very few (level & resampling rate)

cost: simple CPU / lots of store

- enhancements to sampling:

+ looped sections for simple 'sustain'

+ mix 2 or 3 samples for timbre 'space'



- **Sinusoid models**

- exploit harmonic structure of pitched sounds

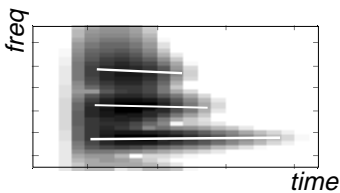
fidelity: good to excellent

controls: pitch and timescale well separated

cost: moderate CPU / large store

- parameter extraction is straightforward

- additional 'noise' residual for non-pitched parts

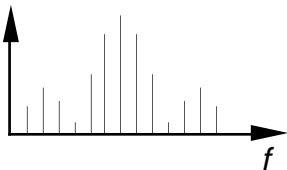


- **Nonlinear models (e.g. FM)**

fidelity: pleasant sounds but limited scope

controls: good range but unpredictable

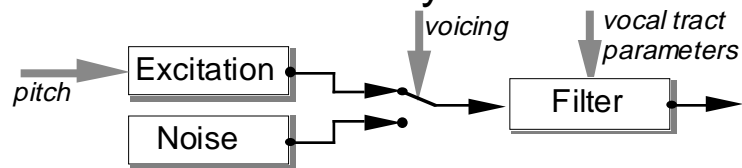
cost: moderate CPU / little store



Synthesis 3: Source models

- **Source-filter models (e.g. LPC)**

- excitation modified by resonances



fidelity: moderate-good for appropriate signals

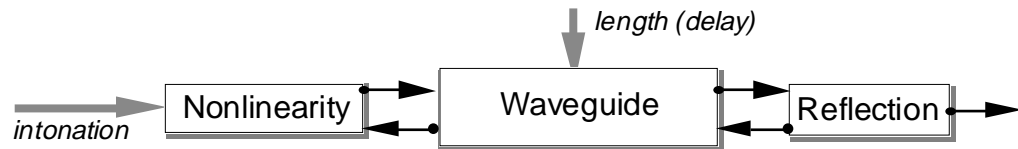
controls: excitation and resonance separate

cost: CPU moderate / storage moderate

- good extraction algorithms available
- works best for speech

- **Physical models (e.g. waveguide)**

- common structure for musical instruments:



fidelity: often startling when it works

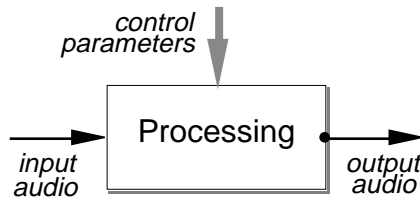
controls: reflect physical source, excellent

cost: CPU moderate / parameter store low

- each model is limited / hard to extend



Processing

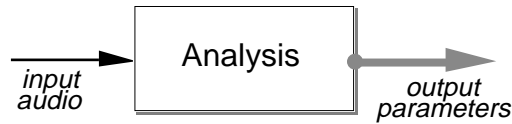


Modifying an audio signal

- **Online:**
 - linear/nonlin. filtering (presence, companding)
 - echo / chorus / flanging
 - reverberation
 - spatial location (azimuth/elevation/range)
- **Event-based**
 - pitch/duration modification (resampling, SOLA, looping, reverse)
 - cross-synthesis (LPC/ Fourier domain)
- **Control inputs from:**
 - explicit interface (sliders, curves)
 - **analysis** extraction from audio streams...



Analysis



Derive control parameters from audio signal

- **Auditory function is hard to model**
 - speech recognition
 - auditory scene analysis
- **.. but a simplistic analysis has uses**
 - pre-linguistic understanding e.g. dogs
- **Audio signal → parameter signal**
 - energy (full band/sub bands/ratios)
 - periodicity/pitch tracking
 - azimuth/triangulation?
- **Audio signal → event sequence**
 - the “clapper”



- **Hearing determines the importance of sound**
 - detectibility
 - relevant aspects
- **Sampled waveforms = core of digital audio**
- **Synthesis algorithms .. tradeoff:**
 - fidelity
 - control flexibility
 - computational cost
 - breadth/range of applicability
 - parameter extraction mechanisms
- **Modifications can be controlled explicitly or by derived parameters**
 - e.g. 'dog hearing'



Spatial location

- **Primary spatial cue is azimuth (from 2 ears)**
 - L-R intensity difference (head shadow) ~ 1 dB
 - L-R envelope delay (path length) ~ 0.1 ms
- **Secondary cues for elevation and range**
 - elevation from L-R *spectrum* & its changes
 - range from level, coloration, direct-to-reverb
- **Synthesizing spatial location**
 - simple pan + delay (freq. dep?) for azimuth
 - sampled HRTFs can incorporate elevation,...
.. depend on individual
- **Delivering spatialized signals**
 - headphones
 - speakers, transaural
 - but: listener location?
dynamic cues?



Speech recognition

- **Major issues:**
 - isolated word or continuous
 - speaker independent, adaptive or individual
 - vocabulary size, (grammar complexity)
 - signal quality / robustness
- **State of the art**
 - moderate perplexity, speaker-independent interactive telephone systems (stock quotes)
 - transcription of TV broadcasts, conversations at 30-40% word error
 - searching alternate Markov model hypotheses is large & slow: ~ real-time on fast CPU
- **Alternatives**
 - fixed small-vocabulary module
 - 'cheap & cheerful' trainable templates

