AUDIO

Henning Schulzrinne
Dept. of Computer Science
Columbia University
Spring 2015
Key objectives

• How do humans generate and process sound?
• How does digital sound work?
• How fast do I have to sample audio?
• How can we represent time domain signals in the frequency domain? Why?
• How do audio codecs work?
• How do we measure their quality?
• What is the impact of networks (packet loss) on audio quality?
Human speech
Human speech

- voiced sounds: vocal cords vibrate (e.g., A4 [above middle C] = 440 Hz
  - vowels (a, e, i, o, u, ...)
  - determines pitch
- unvoiced sounds:
  - fricatives (f, s)
  - plosives (p, d)
- filtered by vocal tract
- changes slowly (10 to 100 ms)
- air volume → loudness (dB)
Human hearing
Human hearing
Human hearing & age
Digital sound
Analog-to-digital conversion

- Sample value of digital signal at $f_s$ (8 – 96 kHz)
- Digitize into $2^B$ discrete values (8-24)
Sample & hold

Voltage sampled and held to allow quantization.

quantization noise

Sampling Period
(1/sampling rate)

Mark Handley
Direct-Stream Digital

Delta-Sigma coding
How fast to sample?

• Harry Nyquist (1928) & Claude Shannon (1949)
  • no loss of information $\rightarrow$ sampling frequency $\geq 2 \times$ maximum signal frequency

• More recent: compressed sensing
  • works for sparse signals in some space
## Audio coding

<table>
<thead>
<tr>
<th>application</th>
<th>frequency</th>
<th>sampling</th>
<th>quantization</th>
</tr>
</thead>
<tbody>
<tr>
<td>telephone</td>
<td>300-3,400 Hz</td>
<td>8 kHz</td>
<td>12-13</td>
</tr>
<tr>
<td>wide-band</td>
<td>50-7,000 Hz</td>
<td>16 kHz</td>
<td>14-15</td>
</tr>
<tr>
<td>high quality</td>
<td>30-15,000 Hz</td>
<td>32 kHz</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>20-20,000 Hz</td>
<td>44.1 kHz</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>10-22,000 Hz</td>
<td>48 kHz</td>
<td>≤ 24</td>
</tr>
</tbody>
</table>

- **CD**: 24 bit, 44.1/48 kHz
- **DAT**: 24 bit, 44.1/48 kHz
Complete A/D

Diagram:
- Analog Signal
- Low-pass filter
- Filtered Analog Signal
- Sample and Hold
- Quantizer
- Sample Clock
- Digitized Codewords
Aliasing distortion
Quantization

- CDs: 16 bit → lots of bits
- Professional audio: 24 bits (or more)
- 8-bit linear has poor quality (noise)
- Ear has logarithmic sensitivity → “companding”
  - used for Dolby tape decks
  - quantization noise ∼ signal level
Quantization noise

High frequency noise introduced
Fourier transform

- Fourier transform: time series $\rightarrow$ series of frequencies
  - complex frequencies: amplitude & phases
- Inverse Fourier transform: frequencies (amplitude & phase) $\rightarrow$ time series
- Note: also works for other basis functions
Fourier series

• Express periodic function as sum of sines and cosines of different amplitudes
  • iff band-limited, finite sum
• Time domain \rightarrow frequency domain
  • no information loss
    • and no compression
  • but for periodic (or time limited) signals
• http://www.westga.edu/~jhasbun/osp/Fourier.htm
Fourier series of a periodic function

\[ f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} \left[ a_n \cos(nx) + b_n \sin(nx) \right] \]

\[ a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(nx) \, dx \]

\[ b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(nx) \, dx \]

-cos

discrete

time,

continuous
discrete
frequencies
Fourier transform

forward transform (time $x$, real frequency $k$)

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi ikx} \, dk$$

inverse transform

$$f(x) = \int_{-\infty}^{\infty} F(k) e^{2\pi ikx} \, dk$$

continuous time, continuous frequencies

$$e^{xi} = \cos(x) + i\sin(x)$$
Discrete Fourier transform

• For sampled functions, continuous FT not very useful → DFT

\[ F_n = \sum_{k=0}^{N-1} f_k e^{-2\pi i nk/N} \]

\[ f_k = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{2\pi i nk/N} \]

complex numbers → complex coefficients
DFT example

- Interpreting a DFT can be slightly difficult, because the DFT of real data includes complex numbers.
  - The magnitude of the complex number for a DFT component is the power at that frequency.
  - The phase $\theta$ of the waveform can be determined from the relative values of the real and imaginary coefficients.
  - Also both positive and “negative” frequencies show up.
DFT example

Sampled data:
\[ f(x) = 2 \sin(x) + \sin(3x) \]

DFT: Real Components

DFT: Imaginary Components

DFT: Magnitude

“Negative” Frequencies
DFT example

Sampled data: Square wave

DFT: Real Components

DFT: Imaginary Components

DFT: Magnitude
Fast Fourier Transform (FFT)

• Discrete Fourier Transform would normally require $O(n^2)$ time to process for $n$ samples:

\[ F_n = \sum_{k=0}^{N-1} f_k e^{-2\pi i nk/N} \]

• Don’t usually calculate it this way in practice.
  • Fast Fourier Transform takes $O(n \log(n))$ time.
  • Most common algorithm is the Cooley-Tukey Algorithm.
Fourier Cosine Transform

• Split function into odd and even parts:

\[ f(x) = \frac{1}{2}[f(x) + f(-x)] + \frac{1}{2}[f(x) - f(-x)] = E(x) + O(x) \]

• Re-express FT:

\[ F(k) = \int E(x) \cos(2\pi kx) dx - i \int O(x) \sin(2\pi kx) dx \]

• Only real numbers from an even function \( \rightarrow \) DFT becomes DCT
DCT (for JPEG)

\[
f_j = \sum_{n=0}^{N-1} x_n \cos \left[ \frac{\pi}{N} j \left( n + \frac{1}{2} \right) \right]
\]

other versions exist (e.g., for MP3, with overlap)
Why do we use DCT for multimedia?

• For audio:
  • Human ear has different dynamic range for different frequencies.
  • Transform to from time domain to frequency domain, and quantize different frequencies differently.

• For images and video:
  • Human eye is less sensitive to fine detail.
  • Transform from spatial domain to frequency domain, and quantize high frequencies more coarsely (or not at all)
  • Has the effect of slightly blurring the image - may not be perceptible if done right.
Why use DCT/DFT?

• Some tasks easier in frequency domain
  • e.g., graphic equalizer, convolution
• Human hearing is logarithmic in frequency (→ octaves)
• Masking effects (see MP3)
Example: DCT for image

```
PICTURE MATRIX
40 24 15 19 28 24 19 15
38 34 35 35 31 28 27 29
40 47 49 40 33 29 32 43
42 49 50 39 34 30 32 46
40 47 46 35 31 32 35 43
38 43 42 31 27 27 28 33
39 33 25 17 14 15 19 26
29 16 6 1 -4 0 7 18
```

```
DCT COEFFICIENTS
239 32 27 -12 3 -5 3 1
34 -3 -19 6 3 0 -1 1
-70 2 8 23 9 6 -1 -1
5 0 -6 11 -2 0 -1 1
-17 -3 6 6 3 -1 0 0
2 4 2 2 1 -2 0 1
-3 0 0 -1 -1 -1 0 0
1 -1 3 1 0 0 0 0
```
μ-law encoding
μ-law encoding

Input Signal (linear encoding)

Output Signal (8-bit μ-law encoding)

Small amplitude input signal coded with more code points than it would be with 8-bit linear
Companding

Wikipedia
\( \mu \)-law & A-law

\[ F(x) = \frac{\text{sgn}(x) \ln(1 + \mu |x|)}{\ln(1 + \mu)} \quad 0 \leq |x| \leq 1 \]

\( \mu \) is 255 in US/Japan

A-law

\[ F(x) = \frac{A|x|}{\ln(1 + A)} \quad 0 \leq |x| < \frac{1}{A} \]
\[ F(x) = \frac{\text{sgn}(x) \ln(1 + A|x|)}{\ln(1 + A)} \quad \frac{1}{A} \leq |x| \leq 1 \]

\( A = 87.7 \) in Europe
Differential codec
(Adaptive) Differential Pulse Code Modulation
ADPCM

• Makes a simple prediction of the next sample, based on weighted previous $n$ samples.
• For G.721, previous 8 weighted samples are added to make the prediction.
• Lossy coding of the difference between the actual sample and the prediction.
  • Difference is quantized into 4 bits $\Rightarrow$ 32Kb/s sent.
  • Quantization levels are adaptive, based on the content of the audio.
  • Receiver runs same prediction algorithm and adaptive quantization levels to reconstruct speech.
Model-based coding

• PCM, DPCM and ADPCM directly code the received audio signal.
• An alternative approach is to build a parameterized model of the sound source (i.e., human voice).
• For each time slice (e.g., 20ms):
  • Analyze the audio signal to determine how the signal was produced.
  • Determine the model parameters that fit.
  • Send the model parameters.
  • At the receiver, synthesize the voice from the model and received parameters.
Speech formation

Voiced sounds: series of pulses of air as larynx opens and closes. Basic tone then shaped by changing resonance of vocal tract.

Unvoiced sounds: larynx held open, turbulent noise made in mouth.
Linear predictive codec

- Earliest low-rate codec (1960s)
- LPC10 at 2.4 kb/s
  - sampling rate 8 kHz
  - frame length 180 samples (22.5 ms)
  - linear predictive filter (10 coefficients = 42 bits)
  - pitch and voicing (7 bits)
  - gain information (5 bits)
Linear predictive codec

Infinite impulse response (IIR) vs. finite impulse response
Code Excited Linear Prediction (CELP)

• Goal is to efficiently encode the residue signal, improving speech quality over LPC, but without increasing the bit rate too much.
• CELP codecs use a codebook of typical residue values. (→ vector quantization)
• Analyzer compares residue to codebook values.
  • Chooses value which is closest.
  • Sends that value.
  • Receiver looks up the code in its codebook, retrieves the residue, and uses this to excite the LPC formant filter.
• Problem is that codebook would require different residue values for every possible voice pitch.
  • Codebook search would be slow, and code would require a lot of bits to send.
    • One solution is to have two codebooks.
    • One fixed by codec designers, just large enough to represent one pitch period of residue.
    • One dynamically filled in with copies of the previous residue delayed by various amounts (delay provides the pitch)
• CELP algorithm using these techniques can provide pretty good quality at 4.8Kb/s.
Enhanced LPC usage

- **GSM (Groupe Speciale Mobile)**
  - Residual Pulse Excited LPC
  - 13 kb/s

- **LD-CELP**
  - Low-delay Code-Excited Linear Prediction (G.728)
  - 16 kb/s

- **CS-ACELP**
  - Conjugate Structure Algebraic CELP (G.729)
  - 8 kb/s

- **MP-MLQ**
  - Multi-Pulse Maximum Likelihood Quantization (G.723.1)
  - 6.3 kb/s
Distortion metrics

- error (noise) \( r(n) = x(n) - y(n) \)
- variances \( \sigma_x^2, \sigma_y^2, \sigma_r^2 \)
- power for signal with pdf \( p(x) \) and range \(-V \ldots +V\)

\[ \sigma_x^2 = \int_{-V}^{+V} (x - \bar{x})^2 p(x) \, dx \]

- SNR = 6.02N – 1.73 for uniform quantizer with N bits
Distortion measures

• SNR *not* a good measure of perceptual quality
• ➣ segmental SNR: time-averaged blocks (say, 16 ms)
• frequency weighting

• *subjective measures*:
  • A-B preference
  • subjective SNR: comparison with additive noise
  • MOS (mean opinion score of 1-5), DRT, DAM, . . .
Quality metrics

- speech vs. music
- communication vs. toll quality

<table>
<thead>
<tr>
<th>score</th>
<th>MOS</th>
<th>DMOS</th>
<th>understanding</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>excellent</td>
<td>inaudible</td>
<td>no effort</td>
</tr>
<tr>
<td>4</td>
<td>good, toll quality</td>
<td>audible, not annoying</td>
<td>no appreciable effort</td>
</tr>
<tr>
<td>3</td>
<td>fair</td>
<td>slightly annoying</td>
<td>moderate effort</td>
</tr>
<tr>
<td>2</td>
<td>poor</td>
<td>annoying</td>
<td>considerable effort</td>
</tr>
<tr>
<td>1</td>
<td>bad</td>
<td>very annoying</td>
<td>no meaning</td>
</tr>
</tbody>
</table>
Subjective quality metrics

- Test phrases (ITU P.800)
  - You will have to be very quiet.
  - There was nothing to be seen.
  - They worshipped wooden idols.
  - I want a minute with the inspector.
  - Did he need any money?

- Diagnostic rhyme test (DRT)
  - 96 pairs like dune vs. tune
  - 90% right → toll quality
Objective quality metrics

- approximate human perception of noise and other distortions
- distortion due to encoding and packet loss (gaps, interpolation of decoder)
- examples: PSQM (P.861), PESQ (P.862), MNB, EMBSD – compare reference signal to distorted signal
- either generate MOS scores or distance metrics
- much cheaper than subjective tests
- only for telephone-quality audio so far
Objective vs. subjective quality
## Common narrowband audio codecs

<table>
<thead>
<tr>
<th>Codec</th>
<th>rate (kb/s)</th>
<th>delay (ms)</th>
<th>multi-rate</th>
<th>embedded</th>
<th>VBR</th>
<th>bit-robust/PLC</th>
<th>remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>iLBC</td>
<td>15.2</td>
<td>20/30</td>
<td></td>
<td></td>
<td></td>
<td>--/X</td>
<td>quality higher than G.729A no licensing</td>
</tr>
<tr>
<td></td>
<td>13.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speex</td>
<td>2.15--2</td>
<td>30</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>--/X</td>
<td>no licensing</td>
</tr>
<tr>
<td></td>
<td>4.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMR-NB</td>
<td>4.75--1</td>
<td>20</td>
<td></td>
<td>X</td>
<td></td>
<td>X/X</td>
<td>3G wireless</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G.729</td>
<td>8</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td>X/X</td>
<td>TDMA wireless</td>
</tr>
<tr>
<td>GSM-FR</td>
<td>13</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td>X/X</td>
<td>GSM wireless (Cingular)</td>
</tr>
<tr>
<td>GSM-EFR</td>
<td>12.2</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td>X/X</td>
<td>2.5G</td>
</tr>
<tr>
<td>G.728</td>
<td>16/12.8</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td>X/X</td>
<td>H.320 (ISDN videconferencing)</td>
</tr>
<tr>
<td>G.723.1</td>
<td>5.3/6.3</td>
<td>37.5</td>
<td></td>
<td></td>
<td></td>
<td>X/--</td>
<td>H.323, videoconferences</td>
</tr>
</tbody>
</table>
# Common wideband audio codecs

<table>
<thead>
<tr>
<th>Codec</th>
<th>rate (kb/s)</th>
<th>delay (ms)</th>
<th>multi-rate</th>
<th>embedded</th>
<th>VBR</th>
<th>bit-robust/PLC</th>
<th>remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speex</td>
<td>4—44.4</td>
<td>34</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>--/X</td>
<td>no licensing</td>
</tr>
<tr>
<td>AMR-WB</td>
<td>6.6—23.85</td>
<td>20</td>
<td>X</td>
<td></td>
<td></td>
<td>X/X</td>
<td>3G wireless</td>
</tr>
<tr>
<td>G.722</td>
<td>48, 56, 64</td>
<td>0.12—5 (1.5)</td>
<td></td>
<td></td>
<td>X/--</td>
<td></td>
<td>2 sub-bands now dated</td>
</tr>
</tbody>
</table>

http://www.voiceage.com/listeningroom.php
MOS vs. packet loss

![Graph showing MOS vs. packet loss for different encoding schemes.]

- G.711 Bernoulli (10ms)
- G.711 Bursty (10ms)
- G.729 Bursty (p_c=30%, 20ms)
iLBC – MOS behavior with packet loss

The tests were performed by Dynstat, Inc., an independent test laboratory.
Score system range: 1 = bad, 2 = poor, 3 = fair, 4 = good, 5 = excellent

Courtesy of GLOBAL IP SOUND
Recent audio codecs

- iLBC: optimized for high packet loss rates (frames encoded independently)
- AMR-NB
  - 3G wireless codec
  - 4.75-12.2 kb/s
  - 20 ms coding delay
Opus audio codex (RFC 6716)

- interactive speech & (stereo) music
- 6 kb/s … 510 kb/s (music)
- frame size: 2.5 ms … 60 ms
- Linear prediction + MDCT
- SILK
  - Developed by Skype
  - Based on Linear Prediction
  - Efficient for voice
  - Up to 8 kHz audio bandwidth

- CELT
  - Developed by Xiph.Org
  - Based on MDCT
  - Good for universal audio/music

SILK decoder

```
| Quantization Indices | Excitation Generator | Long-Term Predictor | LTP Synthesis Filter | LPC Excitation | Short-Term Predictor | LPC Synthesis Filter | Quantized Output |
```

Audio traffic models

• talkspurt: *typically*, constant bit rate:
  • one packet every 20...100 ms ➞ mean: 1.67 s

• silence period: usually none
  • (maybe transmit background noise value) ➞ 1.34 s

• ➞ for telephone conversation, both roughly *exponentially distributed*

• double talk for “hand-off”

• may vary between conversations
  • ➞ only in aggregate
Sound localization

- Human ear uses 3 metrics for stereo localization:
  - intensity
  - time of arrival (TOA) – 7 µs
  - direction filtering and spectral shaping by outer ear

- For shorter wavelengths (4 – 20 kHz), head casts an acoustical shadow giving rise to a lower sound level at the ear farthest from the sound sources

- At long wavelength (20 Hz - 1 KHz) the, head is very small compared to wavelengths
  - In this case localization is based on perceived Interaural Time Differences (ITD)
Audio samples

• Opus: [http://opus-codec.org/examples/](http://opus-codec.org/examples/)
  • both narrowband and wideband