Experimental Design for Machine Learning on Multimedia Data Lecture 8

Prof. Gerald Friedland,
fractor@eecs.berkeley.edu

Website: http://www.icsi.berkeley.edu/~fractor/spring2019/

Today

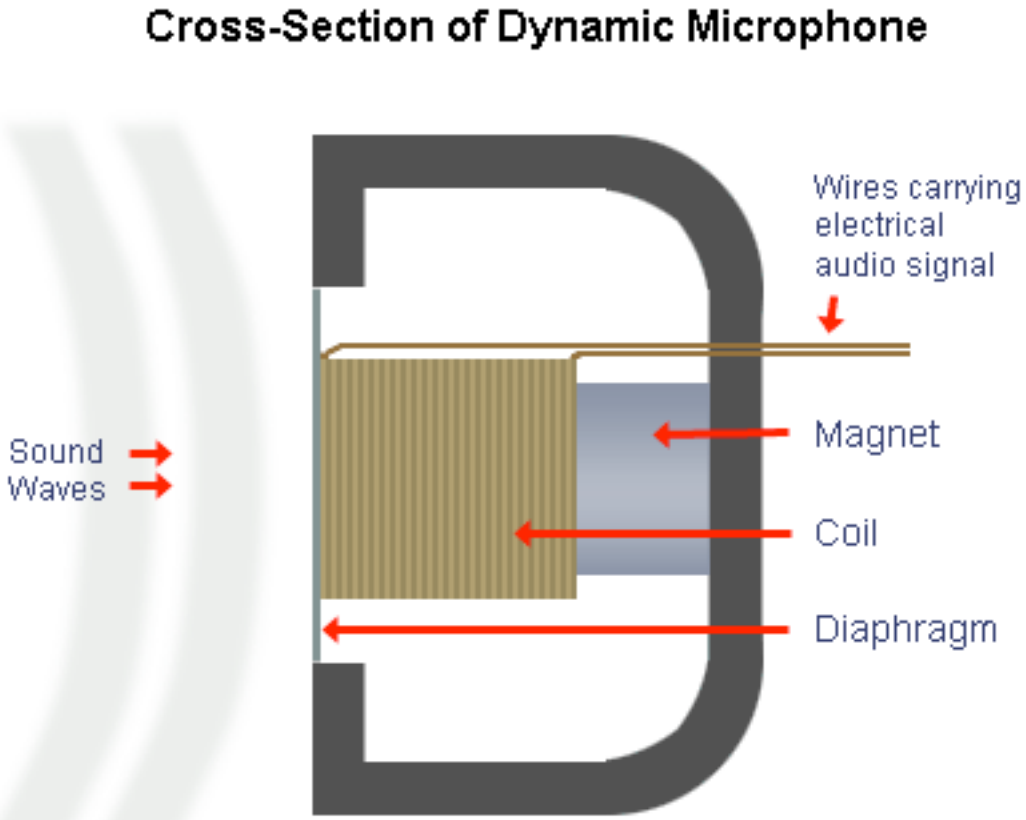
- Answers to Questions
- More on Audio Properties
- Audio Analysis: Features
- Some frameworks and tools to work with sound

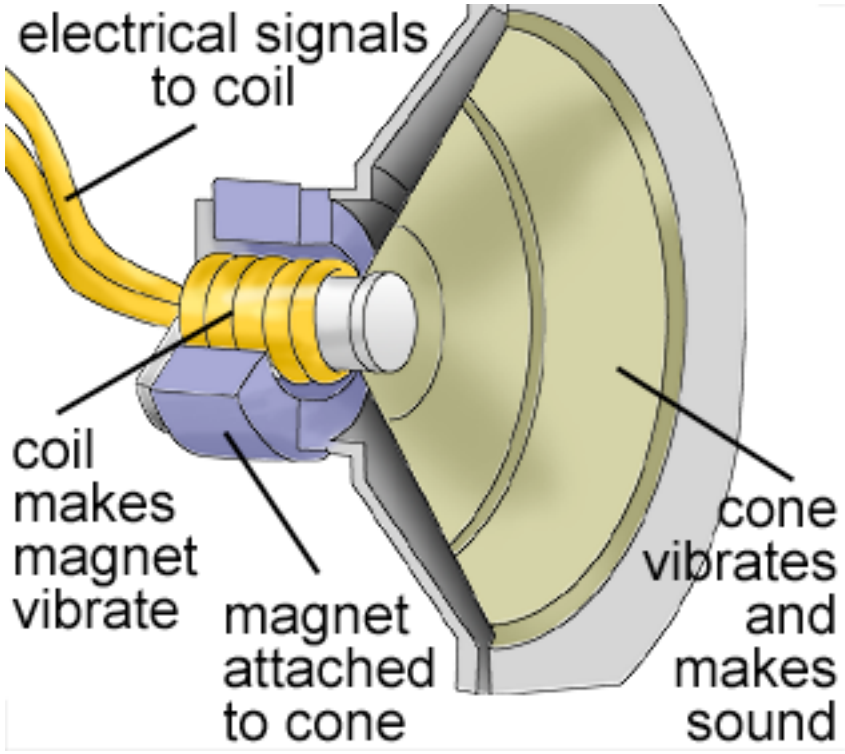
Q&A (2012)

Why is the CD sampling rate 44100Hz and not just 44000?

- Because it is dividable by both 25 video frames per second (PAL) and 30 fps (NTSC)

Loudspeakers vs Microphones

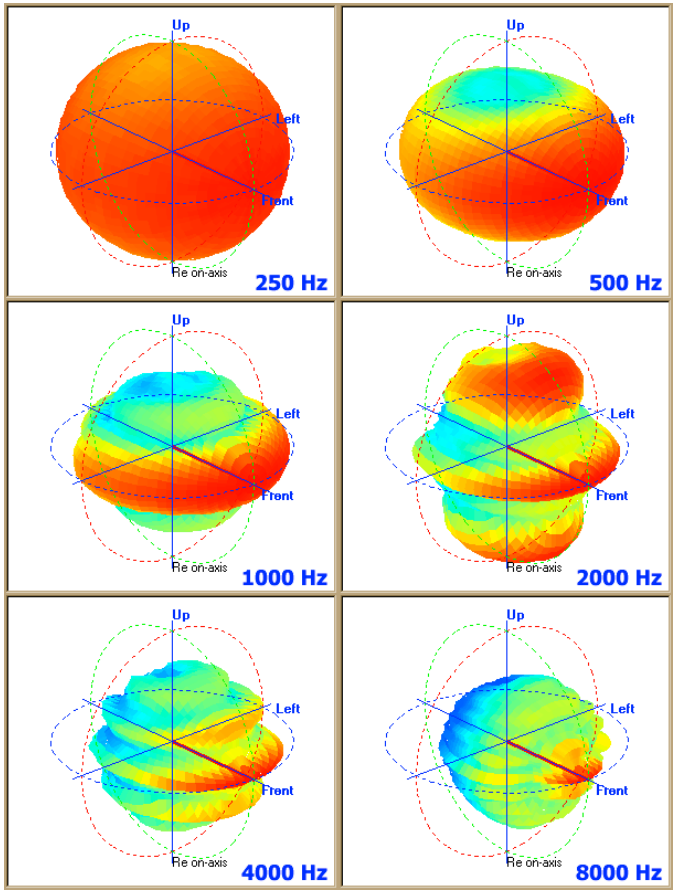




Source: http://www.physics.org

Source: http://www.mediacollege.com/audio/microphones/dynamic.html

Loudspeaker Characteristics



Frequency dependent!
(As are the response patterns of Microphones!)

Useful Filters

- Low-,High-, Band-pass, Graphic Equalizer
- Dynamic Range Compression
- Filter by Example (before Neural Networks): Spectral Subtraction, Wiener Filter
- Handling Multiple Channels

Remember: Fourier Transform

DFT

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

iDFT

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \qquad n = 0, \ldots, N-1.$$

More Information: “Introduction to Multimedia Computing”, Draft Chapter 8

Remember: Fast Fourier Transform

```
// Input: A number N of samples X. N must be an integer power of two.
// s the stride
// Output: The Discrete Fourier Transform of X in the array Y.
FFT(X, N, s)
IF (N==1) THEN
Y[0]:=X{0] // Trivial size-1 case: Sample = DC coefficient
ELSE
Y[0,...,N/2]:=FFT(X,N/2,2*s) // DFT of (X[0],X[2s},X[4s},...)
Y[N/2,...,N-1]:=FFT(X[s,...,N-1],N/2,2*s)
// DFT of (X[s],X[s+2s],X[s+4s],...)
FOR k:=0 TO N/2 // Combine the two halves into one
t:=Y[k]
Y[k]:=t+exp(-2*pi*k/N)*Y[k=N/2]
Y[k+N/2]:=t-exp(-2*pi*k/N)*Y[k=N/2]
FFT := Y
```

More Information: “Introduction to Multimedia Computing”, Draft Chapter 8

Remember: Convolution Theorem

Let f and g be two functions and $f * g$ their convolution. Then:

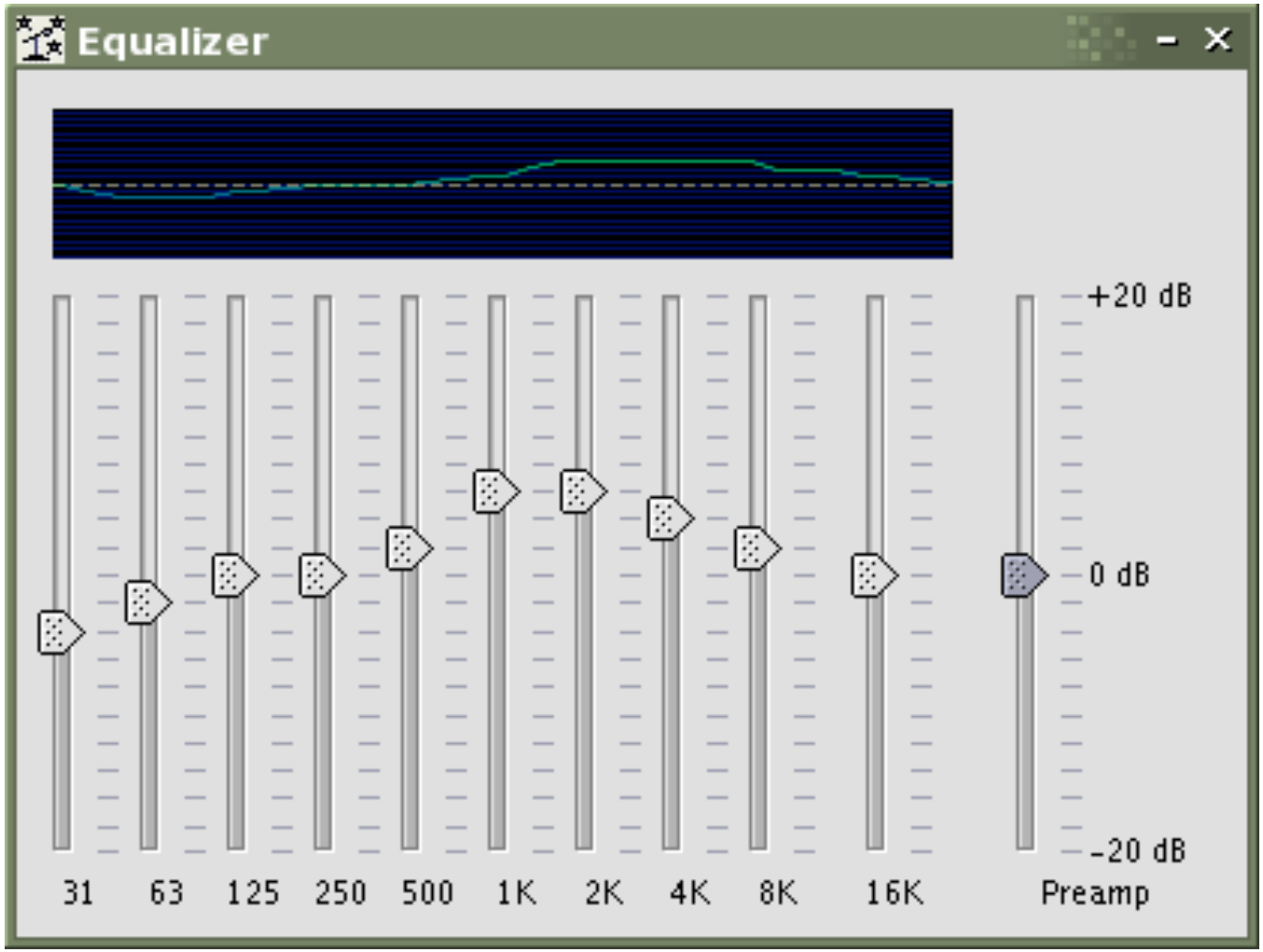
$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}$$

$$\mathcal{F}\{f \cdot g\} = \mathcal{F}\{f\} * \mathcal{F}\{g\}$$

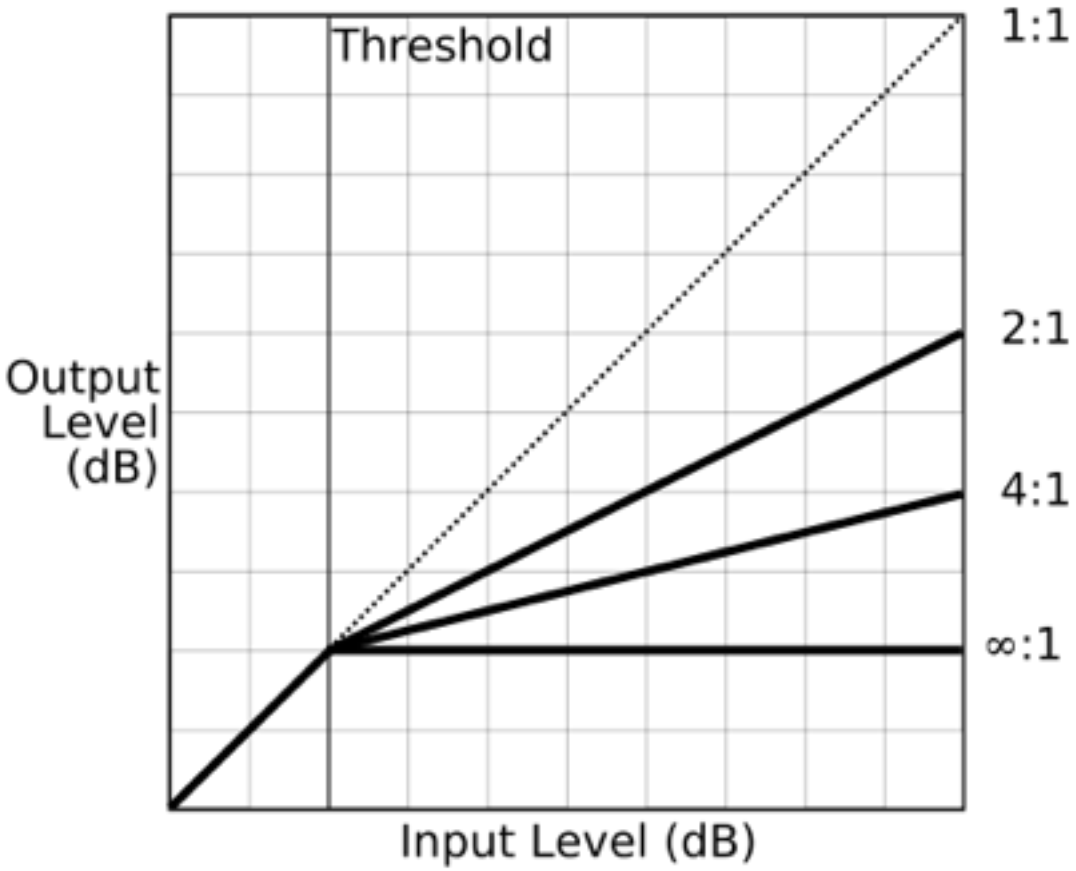
$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$$

More Information: “Introduction to Multimedia Computing”, Draft Chapter 8

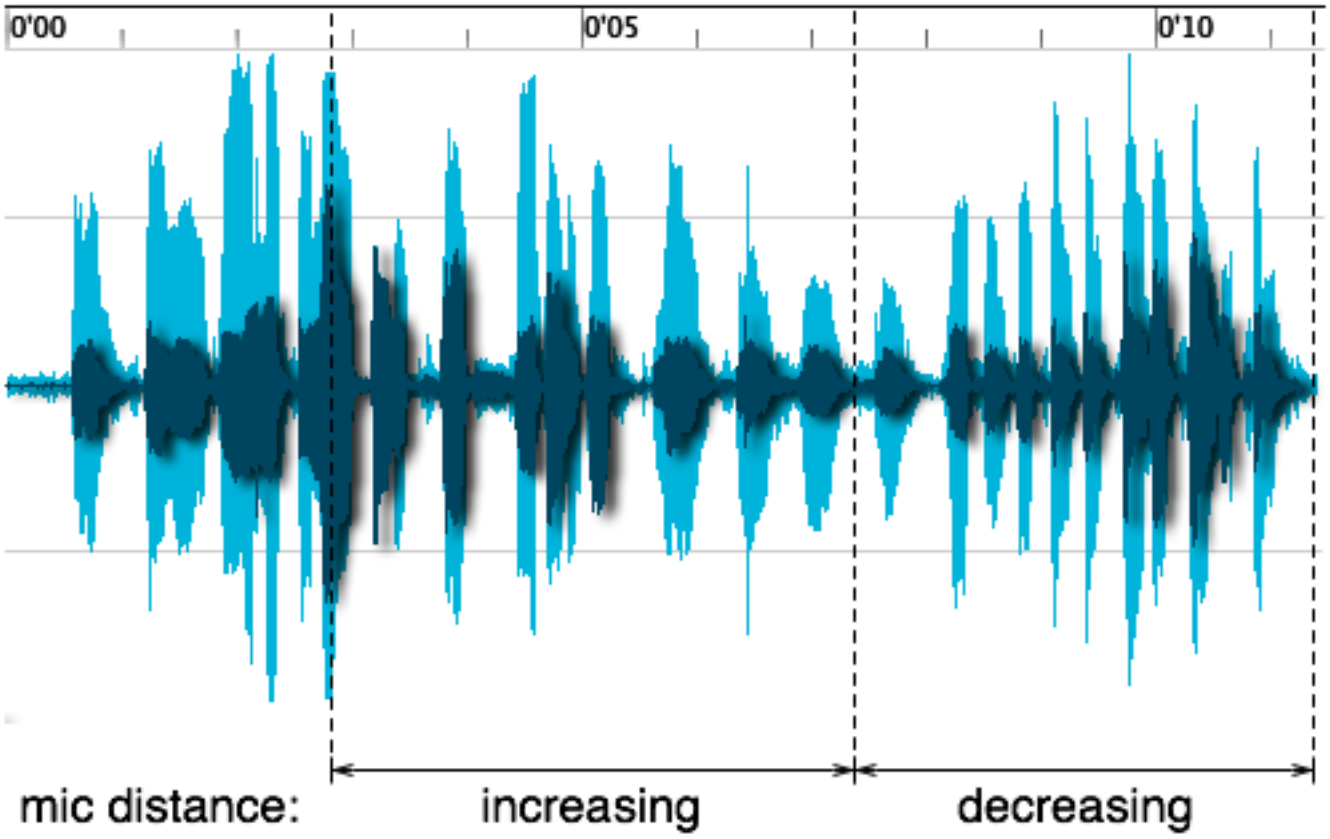
Bandpass Filters



Dynamic Range Compression

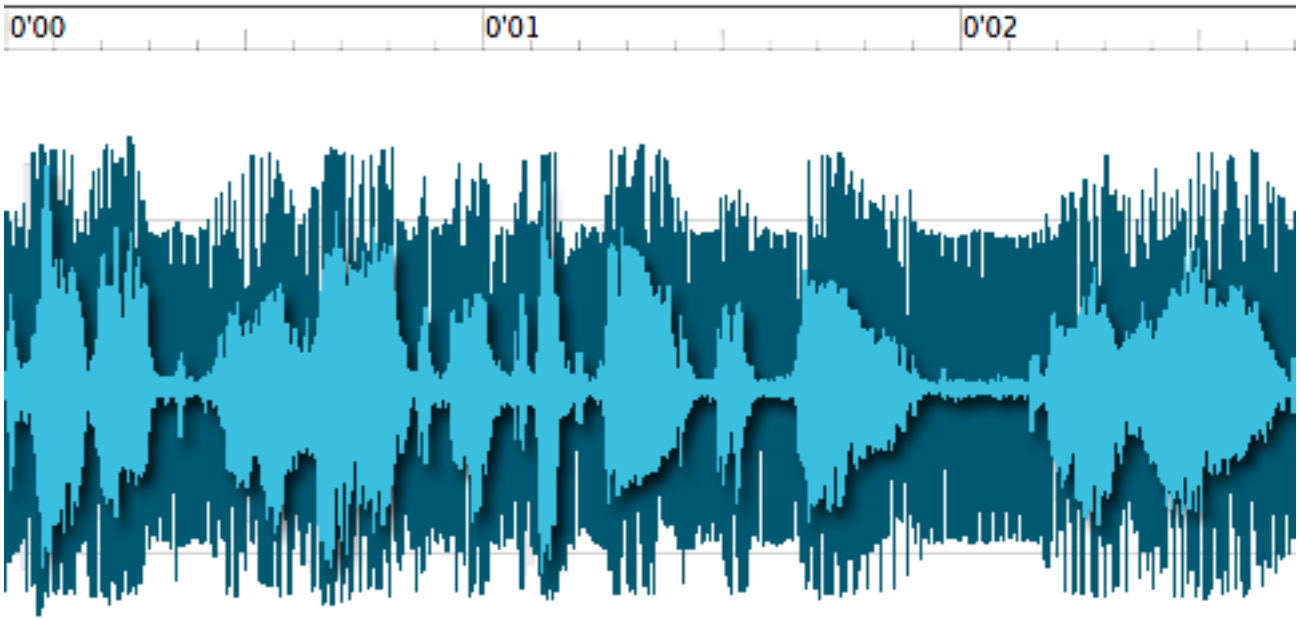


Dynamic Range Compression



Dark blue: Original, Light Blue: Dynamic Range Compressed

Filter by Example: Spectral Subtraction



Dark blue: Original with humming noise, Light blue: Corrected using spectral subtraction of humming.

Filter by Example: Wiener Filter

Assume:

$$\hat{s}(t) = g(t) * [s(t) + n(t)]$$

where:

- s(t) is original signal
- n(t) is noise footprint
- s(t) is estimated signal
- g(t) is Wiener Filter

Filter by Example: Wiener Filter

Define Error:

$$e(t) = s(t+\alpha) - \hat{s}(t)$$

with α being the delay. Thus:

$$e^2(t) = s^2(t+\alpha) - 2s(t+\alpha)\hat{s}(t) + \hat{s}^2(t)$$

(squared error)

Filter by Example: Wiener Filter

Define Error:

$$e(t) = s(t+\alpha) - \hat{s}(t)$$

with α being the delay. Thus:

$$e^2(t) = s^2(t+\alpha) - 2s(t+\alpha)\hat{s}(t) + \hat{s}^2(t)$$

(squared error)

Filter by Example: Wiener Filter

Results in:

$$G(s(t))=|X(s(t))|^2/((|X(s(t))|)^2+(|N(s(t))|)^2)$$

where:

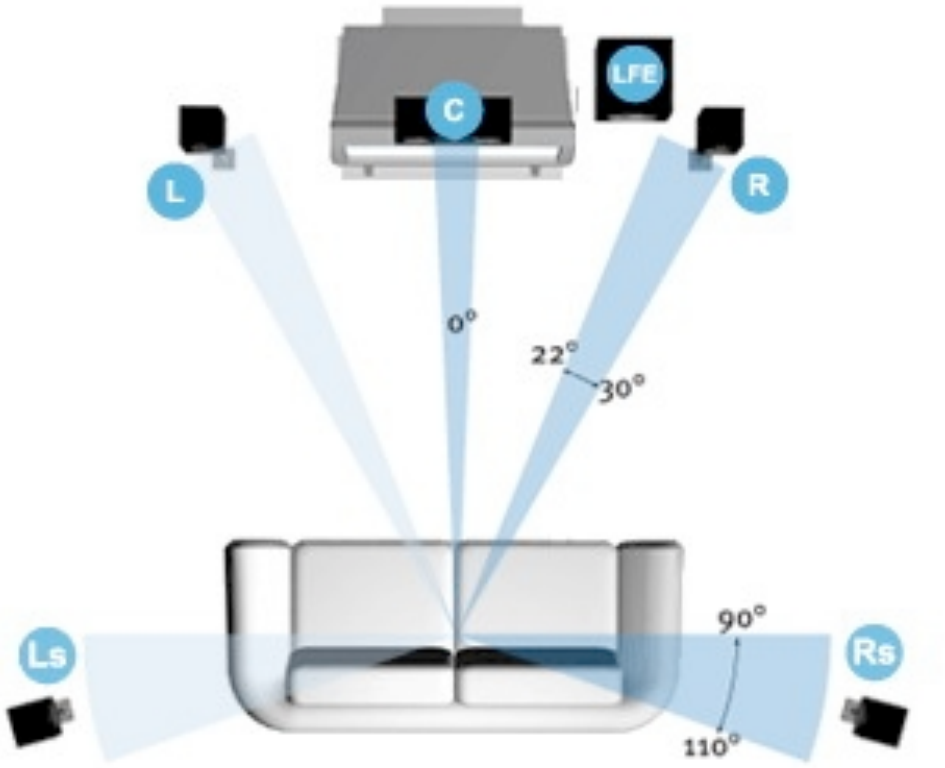
- $X(s(t))$ is the power spectrum of the signal,
- s(t) and $N(s(t))$ is the power spectrum of the noise, and
- $G(s(t))$ is the frequency-space Wiener filter.

Problem: Do we know N(s(t))?

Multichannel Audio

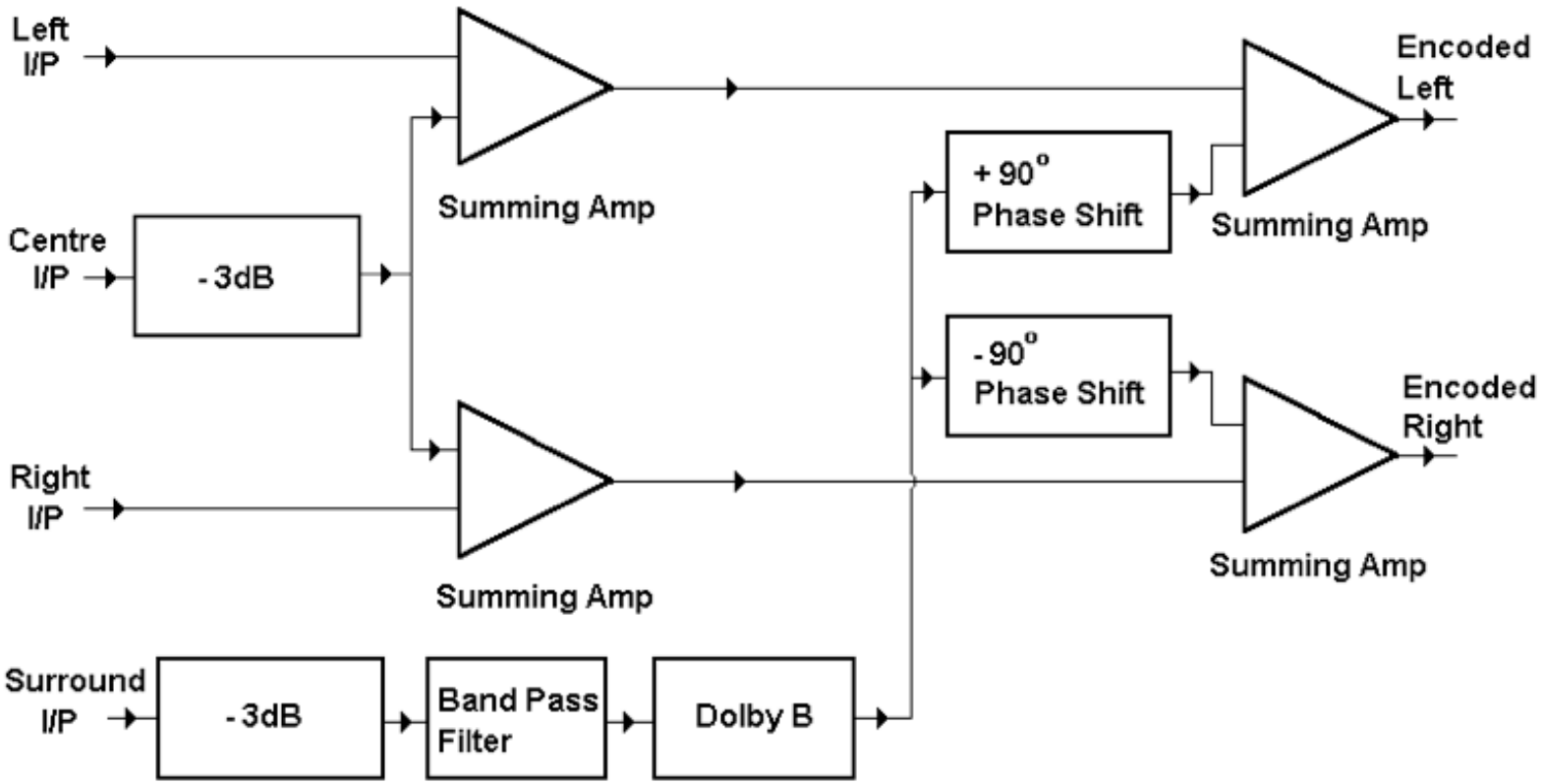
- Microphone Array Recordings popular with Speech Community
- Stereo popular since the 1960s.
- Most cameras have stereo microphones
- Most DVDs are encoded in Dolby Digital 5.1 and in Dolby Surround on Stereo Track

Dolby Digital



Source: http://www.aspecttv.co.uk/surround-sound-setup/

Dolby Surround



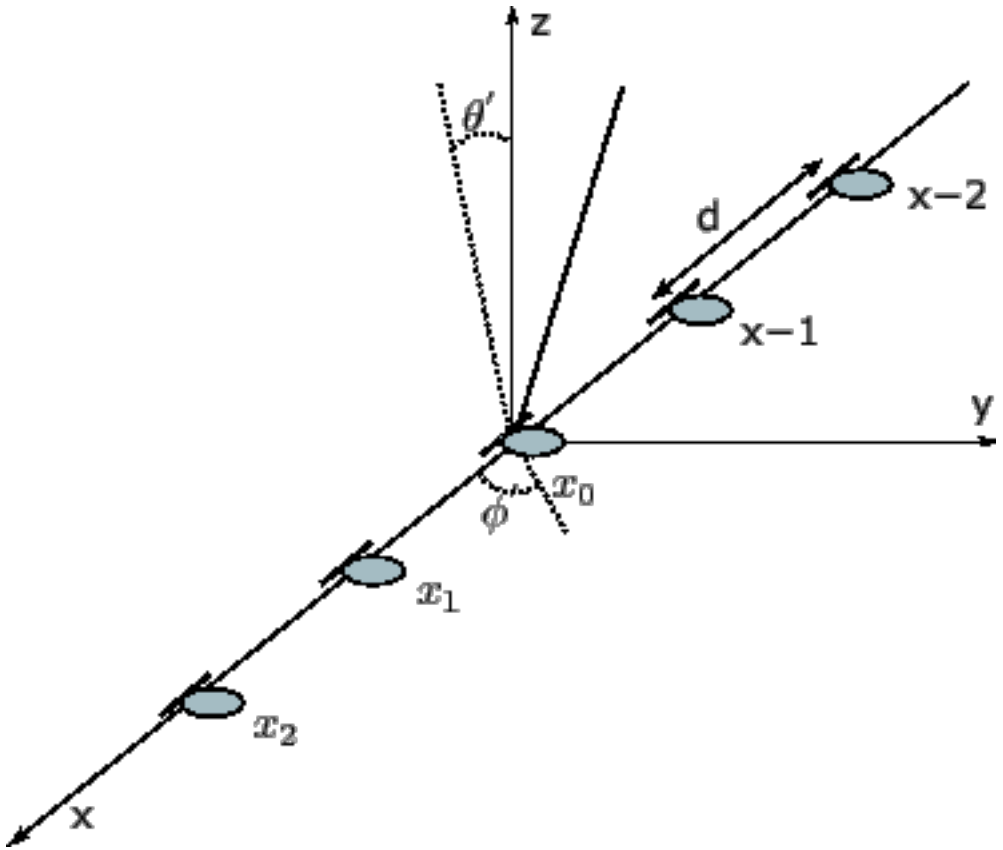
How to Cope with Multiple Channels?

- Treat them individually (different channel=different content)
- Combine them, ideally while reducing noise
- Cross-Infer between channels
- Any combination of the above

Combining Channels

- Beamforming, two major techniques:
- Delay-Sum
- Frequency-Domain Beamformer

Delay-Sum Beamforming



More Information: http://www.xavieranguera.com/phdthesis/

Delay-Sum Beamforming

Output:

- Sum of all channels (lower noise)

- Delay between microphones (directional signal)

Some Basic Audio Features

- Energy (usually rms)
- Zero-Crossing Rate
- Entropy
- Pitch
- Voicedness/Unvoicedness (HNR)
- Long-Term Average Spectrum (LTAS)

Energy

Energy is usually calculated as the root-mean square of a window of samples:

$$x_{\mathrm{rms}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} {x_i}^2} = \sqrt{\frac{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}{n}}.$$

- Most pre-valent basic feature
- Typical filter uses: Squelching
- For analysis: Is there audio at all?

Zero-Crossing Rate

$$SC_{[m]} = \frac{1}{2} \sum_{n=m-N+1}^{m} \left| \operatorname{sgn}(s_{[n]}) - \operatorname{sgn} s_{[n-1]} \right|$$

Used among other things in overlapping speech detection, voiced/unvoiced estimation, music analysis

Entropy

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i).$$

How noisy is the signal? Helps in many cases.

Pitch

Most frequently used method is autocorrelation:

$$R_{[l,m]} = \sum_{n=m-N+1}^{m} S[n]S[n-l]$$

Fails in many situations, computationally intensive.

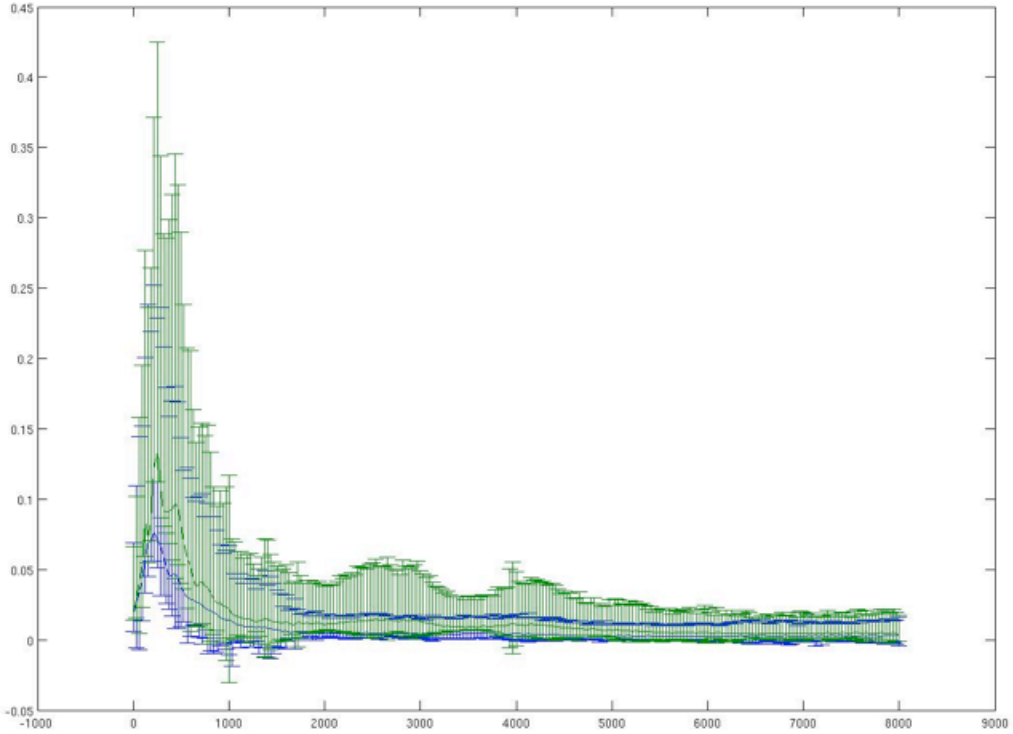
Pitch Estimation is ongoing research!

Harmonic-to-Noise Ratio

$$HNR = \frac{v}{uv}$$

Used as indicator for type of audio (e.g. speech vs music, types of music), speaker age, and as a metric for quality distortion.

Long-Term Average Spectrum



Often used as indicator for type of audio (e.g. speech vs music, types of music).

Basic Audio Features

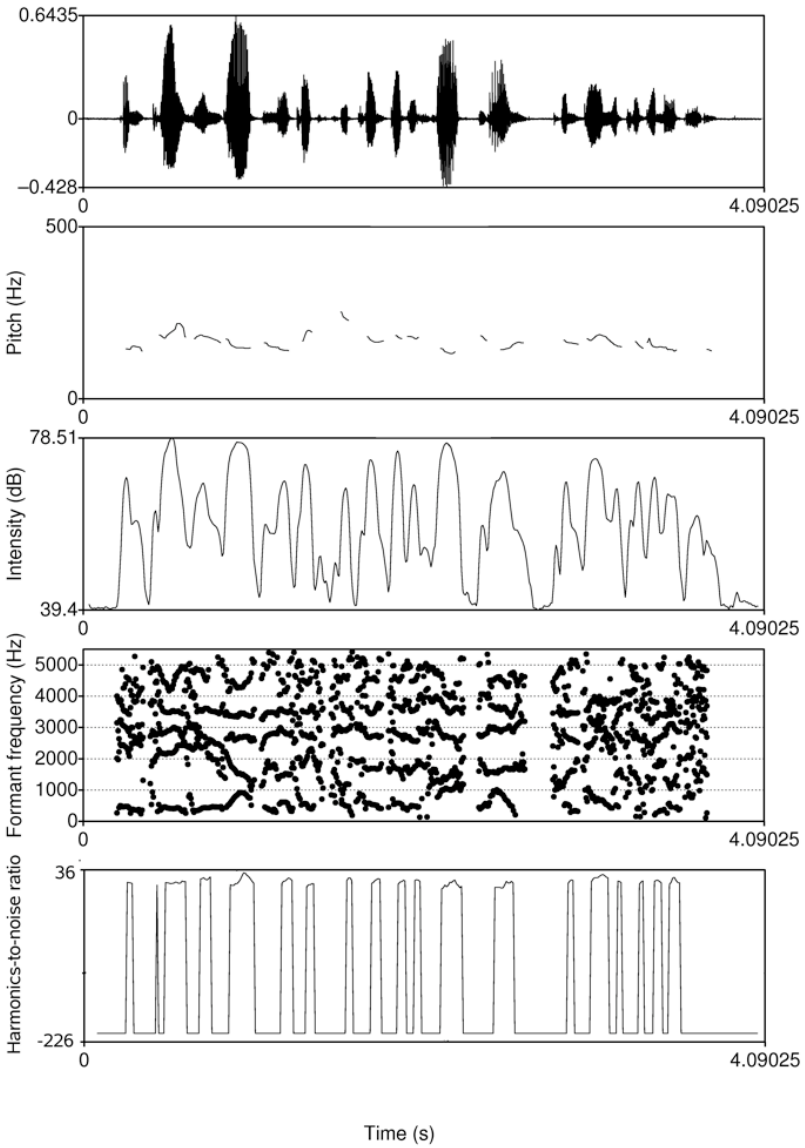
A Speech Signal:

Pitch:

Energy:

Formants:

HNR:



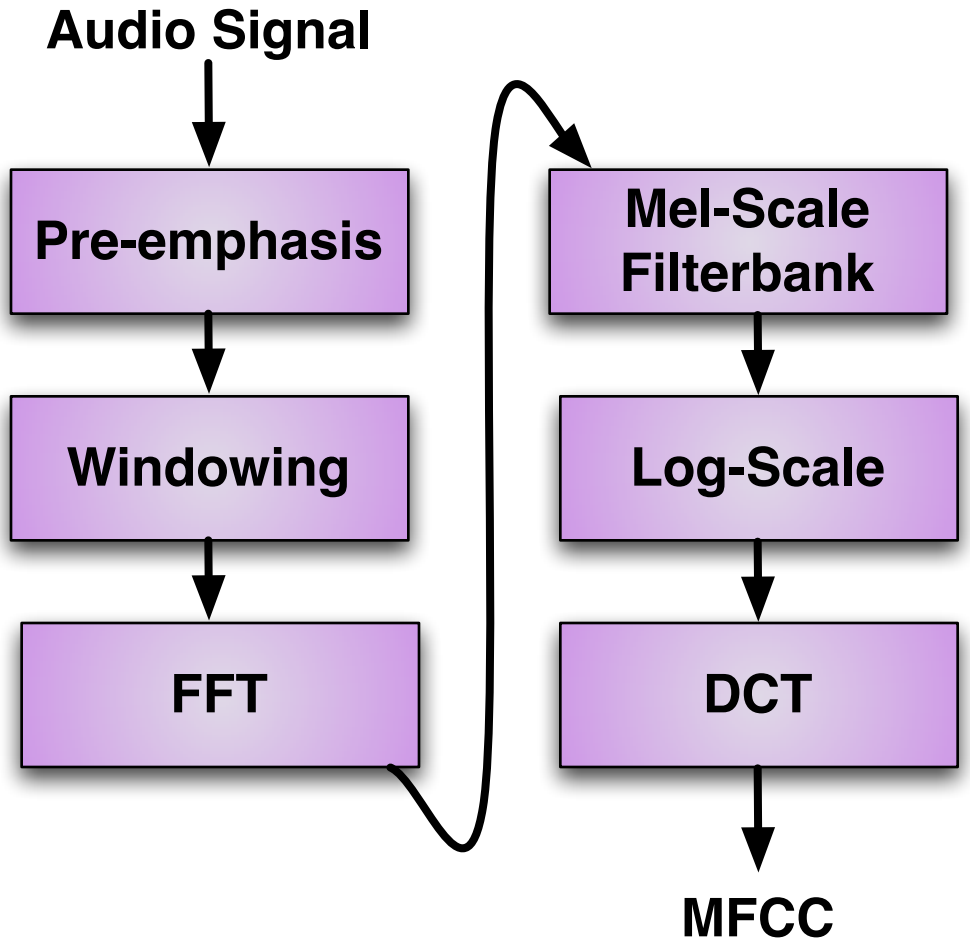
More on Features

- Mel-Frequency-Scaled Coefficients (MFCC)

Other (not explained here):

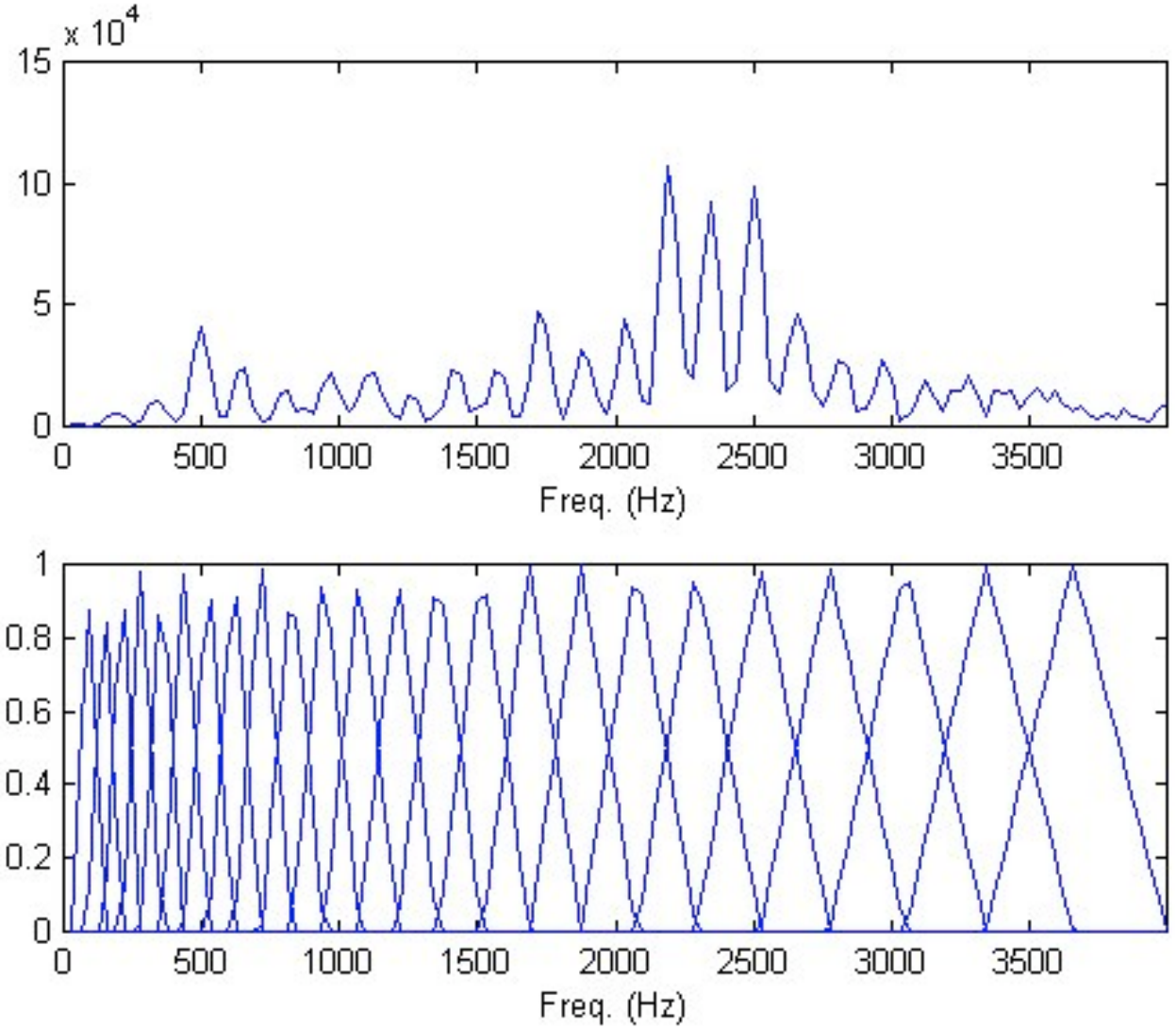
- LPC (Linear Prediction Coefficients)
- PLP (Perceptual Linear Predictive) Features
- RASTA (see Morgan et al)
- MSG (Modulation Spectrogram)

MFCC: Idea



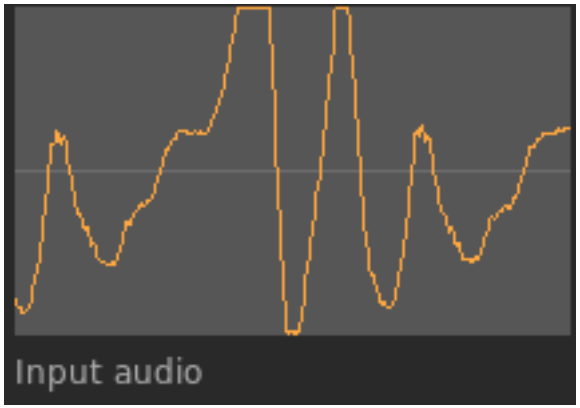
power cepstrum of signal $= \left|\mathbf{F}\left\{\log(|\mathbf{F}\left\{\text{the signal}\right\}|^2)\right\}\right|^2$

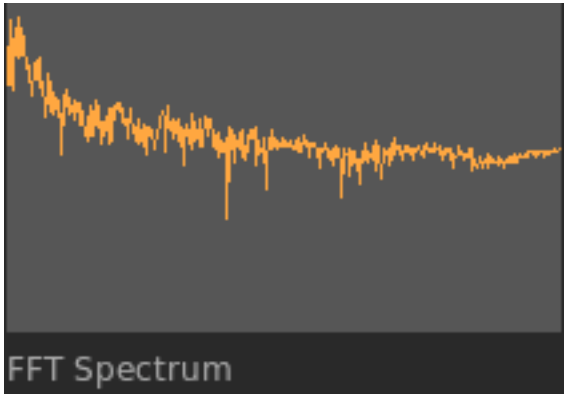
MFCC: Mel Scale

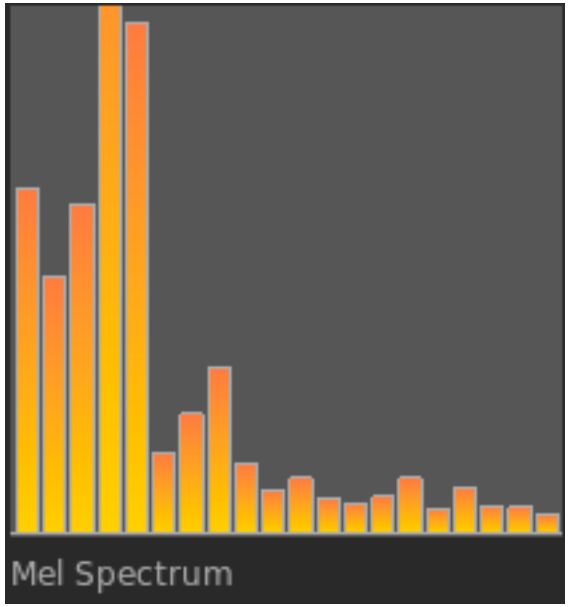


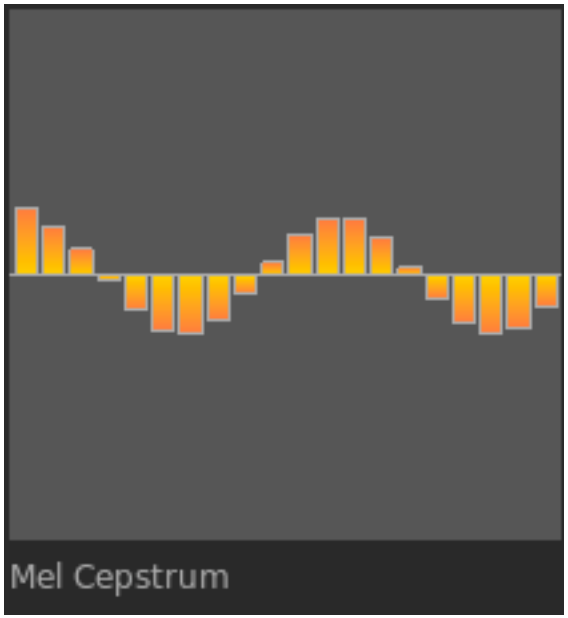

THE UNIVERSITY OF CALIFORNIA
LET THERE BE LIGHT
1868

MFCC: Result


Input audio


FFT Spectrum


Mel Spectrum


Mel Cepstrum

MFCC Variants and Derivates

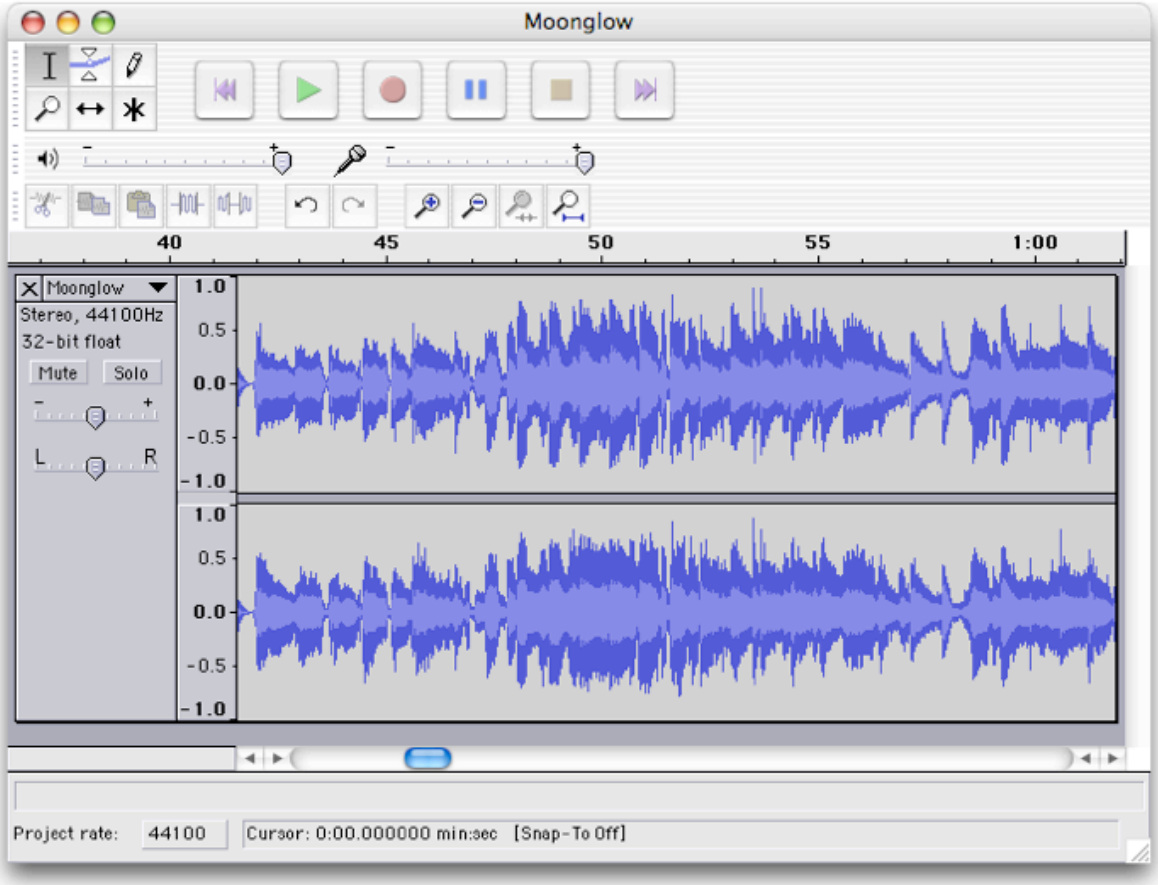
Parameters:

- MFCC12 (often used for ASR)
- MFCC19 (often used in speaker id, diarization)
- “delta”: coefficients subtracted (“first derivative”)
- “deltadelta”: “second derivative”
- Short term: Usually calculated on 10-50ms window

Derivates:

- LFCC (no Mel scale)
- AMFCC (anti Mel scale)

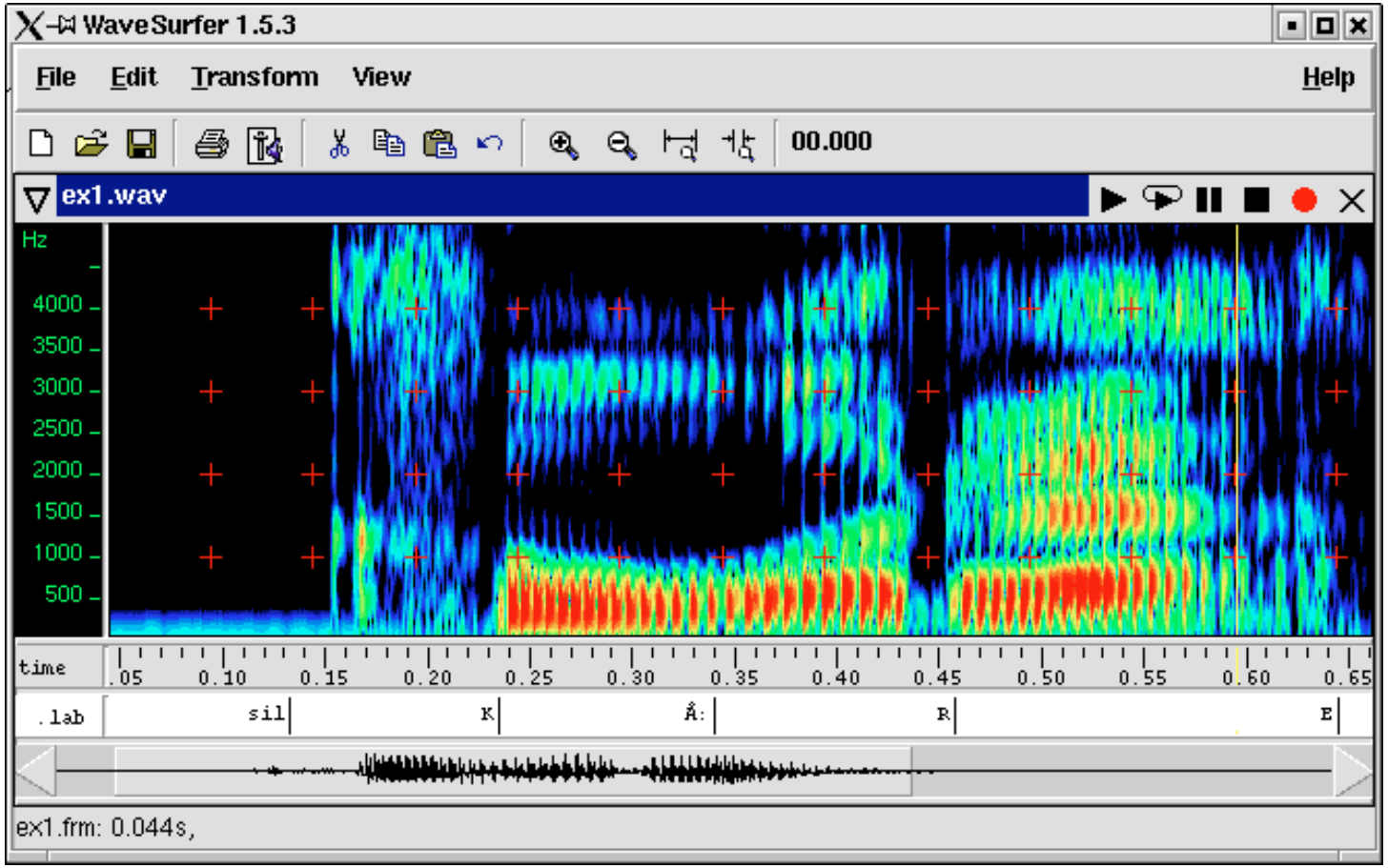
Audacity



Open Source wave editor and filter:

http://audacity.sourceforge.net/

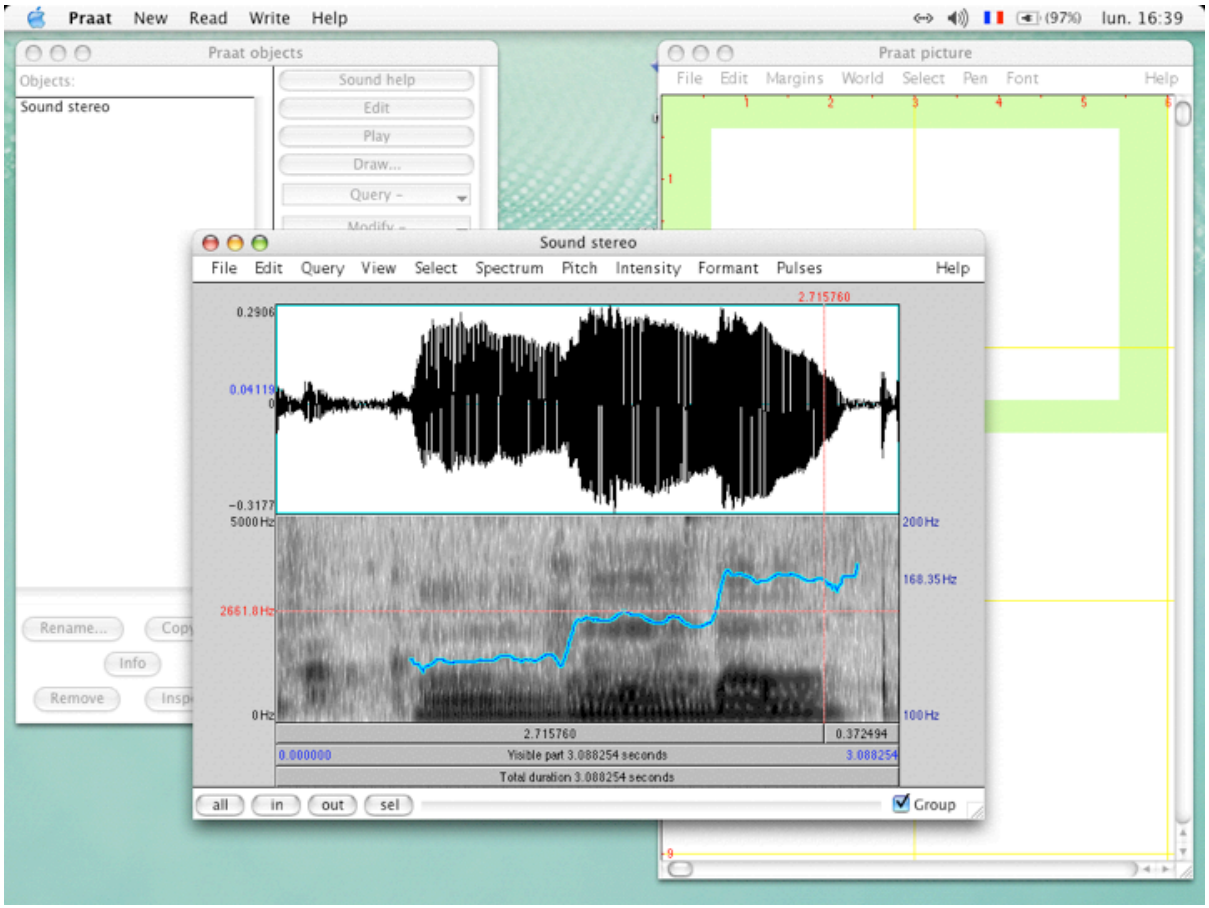
Wavesurfer



Open Source wave editor and filter, specialized for speech:

http://www.speech.kth.se/wavesurfer/

Praat



Open Source speech experimental toolkit

http://www.fon.hum.uva.nl/praat/

Other useful tools

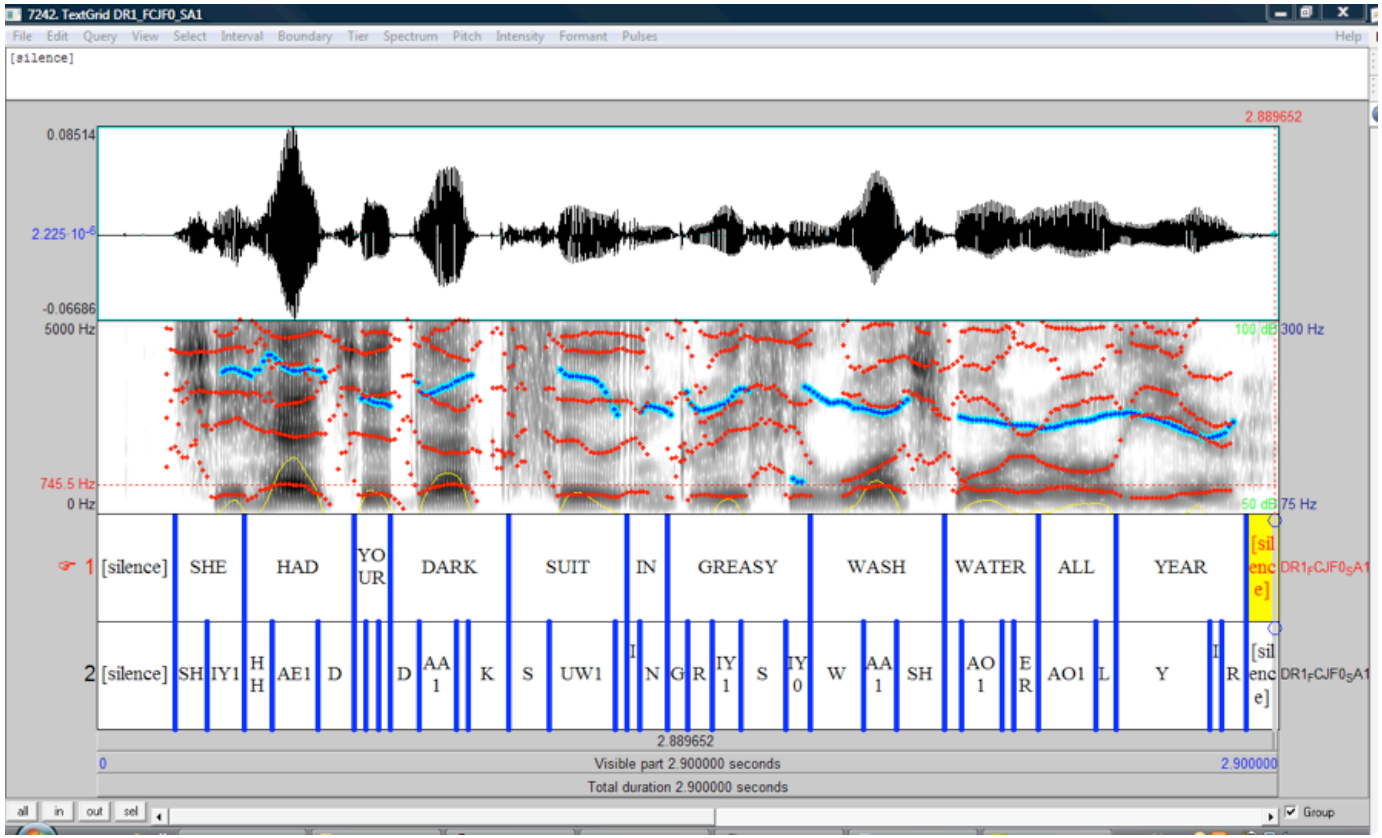
- sox -- command-line based audio filter and converter: http://sox.sourceforge.net/

- ffmpeg -- command-line based audio and video converter http://ffmpeg.org/

- MPlayer (mencoder) -- open source video player and converter: http://www.mplayerhq.hu/

INTERNATIONAL COMPUTER SCIENCE INSTITUTE

More Tools: HTK



Open Source Speech Recognition Toolkit:

http://htk.eng.cam.ac.uk/

More Tools: A nice collection of them

Princeton SoundLab:

http://soundlab.cs.princeton.edu/software/

Next Week

- More on Features
- Some Machine Learning