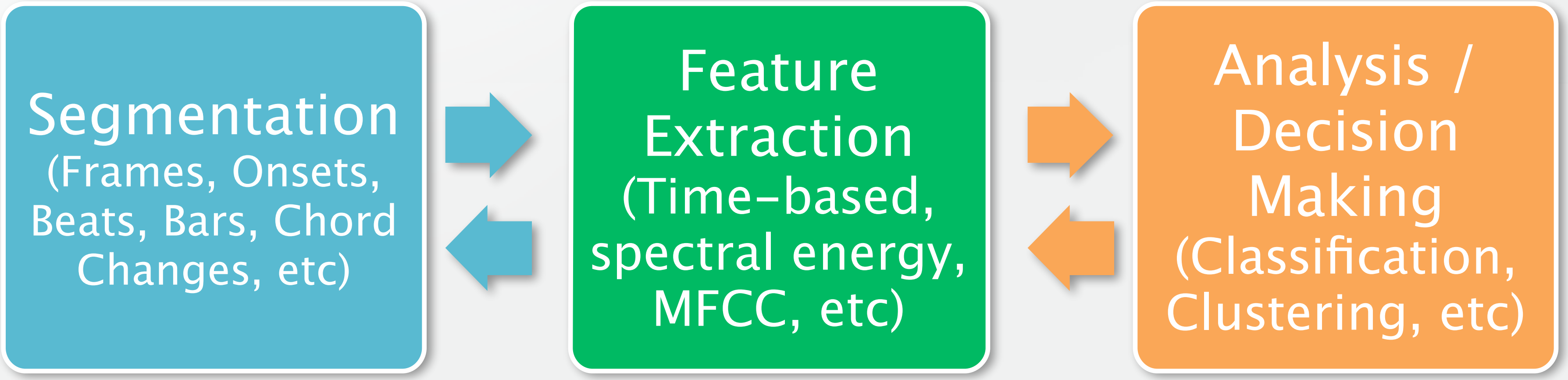
CCRMA MIR Workshop 2014
Signal Analysis & Feature Extraction

Leigh M. Smith
Humtap Inc.
leigh@humtap.com

Basic system overview

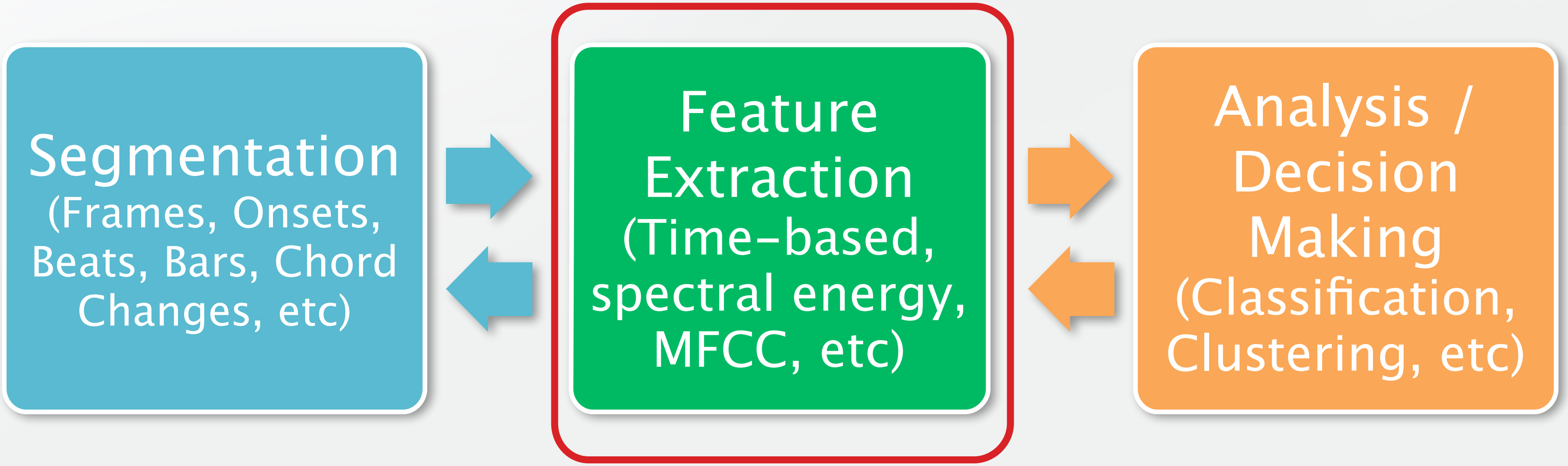


Segmentation
(Frames, Onsets, Beats, Bars, Chord Changes, etc)

Feature Extraction
(Time-based, spectral energy, MFCC, etc)

Analysis / Decision Making
(Classification, Clustering, etc)

Basic system overview



Segmentation
(Frames, Onsets, Beats, Bars, Chord Changes, etc)

Feature Extraction
(Time-based, spectral energy, MFCC, etc)

Analysis / Decision Making
(Classification, Clustering, etc)

Outline

- Signal Analysis and Feature Extraction
- Feature-vector Design
  - Time-domain Features
  - Windowed Feature Extraction
  - Frequency-domain Features
  - Spatial-domain Features
  - Other Feature Domains

Introductions, Context

- Leigh Smith
  - University of Western Australia – Comp Sci Dept.
  - Universiteit van Amsterdam – EmCAP project.
  - IRCAM – Quaero project.
  - iZotope Inc., Humtap Inc.
  - CCRMA MIR Workshop 2011 – 2013
- leigh@humtap.com
- http://www.leighsmith.com/Research

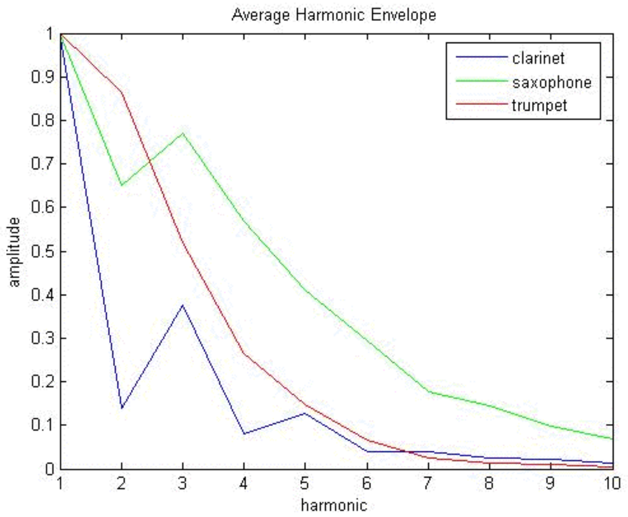
Signal Analysis and Feature Extraction for MIR Applications

- What do we want to do?
  - Match, search, index, transcribe, source-sep, ...
- What do we need to know to do it?
  - Basic feature set
  - Higher-level features
  - Feature data post-processing
  - Application integration
- MIR application design
  - How does the metadata fit in?
- Feature vector design for applications

Typical Audio Source Processing Stages

- Input processing
  - Streaming, decompression, reformatting
- Signal segmentation, windowing in time/freq
  - window size, share, overlap
- 1st-pass windowed feature extraction
  - Basic time-, freq-domain features
- 2nd-pass feature processing
  - Feature massaging, smoothing, pruning
  - 2nd-pass features (tempo, segmentation)
- Post-processing, data output
  - Many options

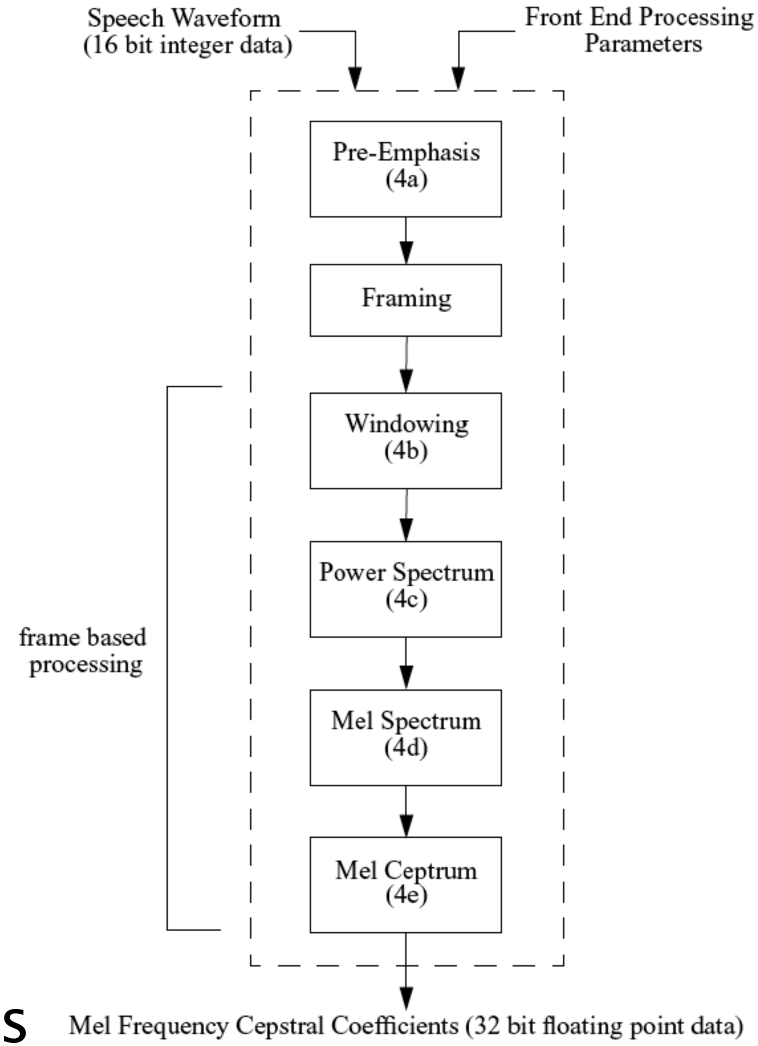
Feature Vectors and Indexing



- **Feature** = derived (numerical) parameter
- **Feature vector** = list of features for a single point/window in time, or average for an entire selection
- **Feature table** = list of feature vectors for several time slices (not always used/stored)

Signal Analysis

- Time-domain Audio Analysis
  - Windowed RMS Envelope Extraction
  - Beat Detection and Rhythm Analysis
  - Time-based signal segmentation
- Frequency-domain Analysis
  - Pitch Detection Techniques
  - Spectral Analysis and Interpretation
  - Spectral Peaks and Tracking
  - Other Spectral Measures
- Cross-domain or combined analysis
  - e.g. Wavelets.



Feature-vector Design

- Application Requirements
  - Labeling, segmentation, etc.
  - Derive feature vector from the app requirements
- Kinds/Domains of Features
  - Time-domain
    - Simple features, onset detection
    - Rhythm, segmentation
  - Frequency-domain
    - Spectrum, spectral statistics
    - Pitch, chroma, key

(See e.g: http://www.create.ucsb.edu/~stp/PostScript/PopeHolmKouznetsov_icmc2.pdf)

Example Features

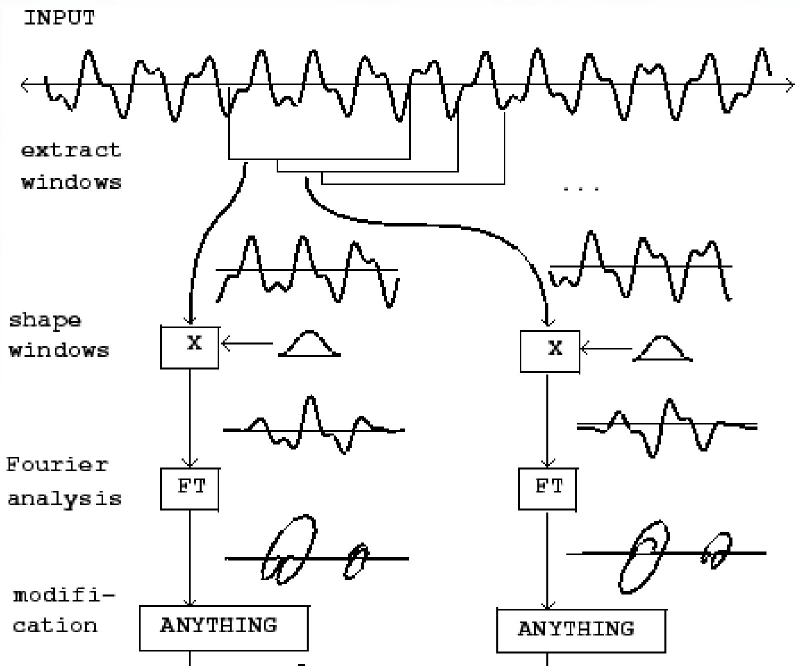
- Features:
  - Time-domain, low-level
    - Windowed RMS amplitude
  - Time-domain, high-level
    - Tempo, beat structure, segmentation
  - Frequency-domain, low-level
    - Pitch, spectrum, spectral peaks
  - Frequency-domain, high-level
    - Peak track birth/death statistics, instrument ID
  - Many other possibilities (see below)

Feature Extraction and Signal Analysis

- Multi-step process:
  - Read input.
  - Apply window or frame extraction.
  - Derive several low-level features.
  - Map, derive next-level features
    - Possible heuristics determine which next-level features are relevant
  - Prune data when appropriate.
- Goal: reduce signal to the smallest set of numbers describing or matching human perception.

Time Sequences, Windowing

- Read audio input.
- Vector multiply by window function.
- Perform analysis.
- Step to next window.
- Hop size normally diff. to window size (overlap).
- Window features
  - Main lobe width, side lobe level, side lobe slope



Time-domain Features

- RMS, Peak
  - LPF/HPF RMS
    - e.g., F < 200 Hz, F > 2000 Hz
- Attack Time/Slope
- Zero-crossing rate (time & freq. domain)
- Temporal Centroid
- Higher-level statistics
  - Mean/variance
  - Variance of sliding windows
  - Spacing of peaks/troughs
  - Many other options

$$RMS = \sqrt{\frac{1}{N}\sum_{n}^{N} x_n^2}$$

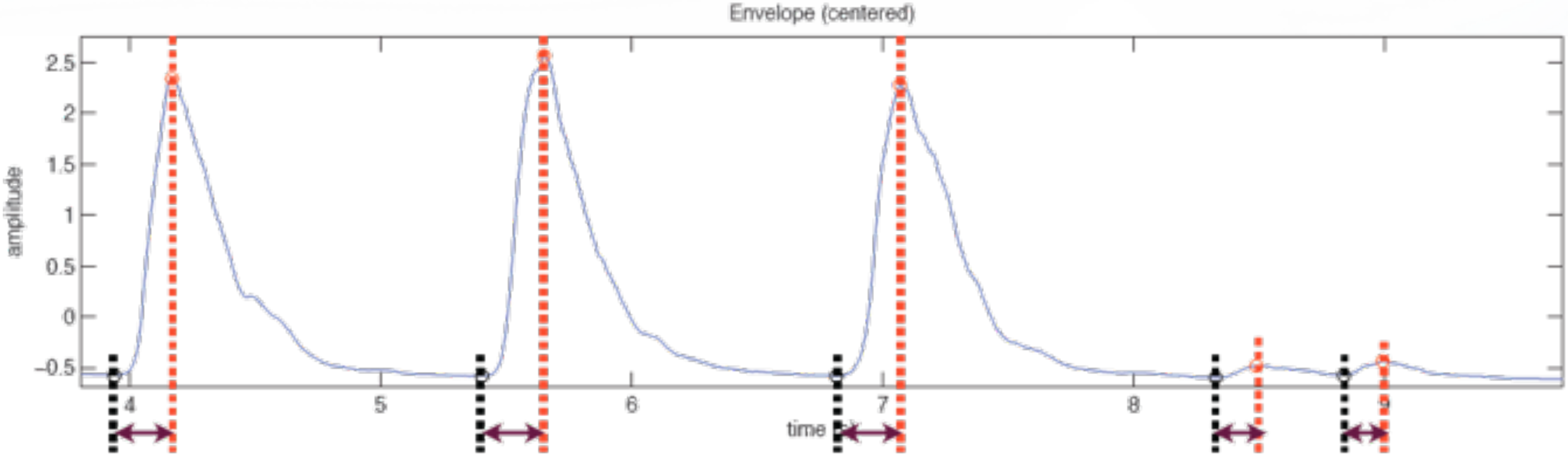
Optional Time-domain Steps

- Pre-filter to get low-freq. and high-freq. RMS values.
- Process stereo channels to get M/S (sum/difference) signals.
- Noise detection.
- Silence detection.

Windowed RMS Envelope Extraction

- pseudo-code for envelope extraction:
  - Outer loop for windows
  - Inner loop to run window and compute RMS value
  - Silence threshold (noise gate)
  - Note-on trigger (peak detector)
  - Suitable sounds: piano sample, drum loop

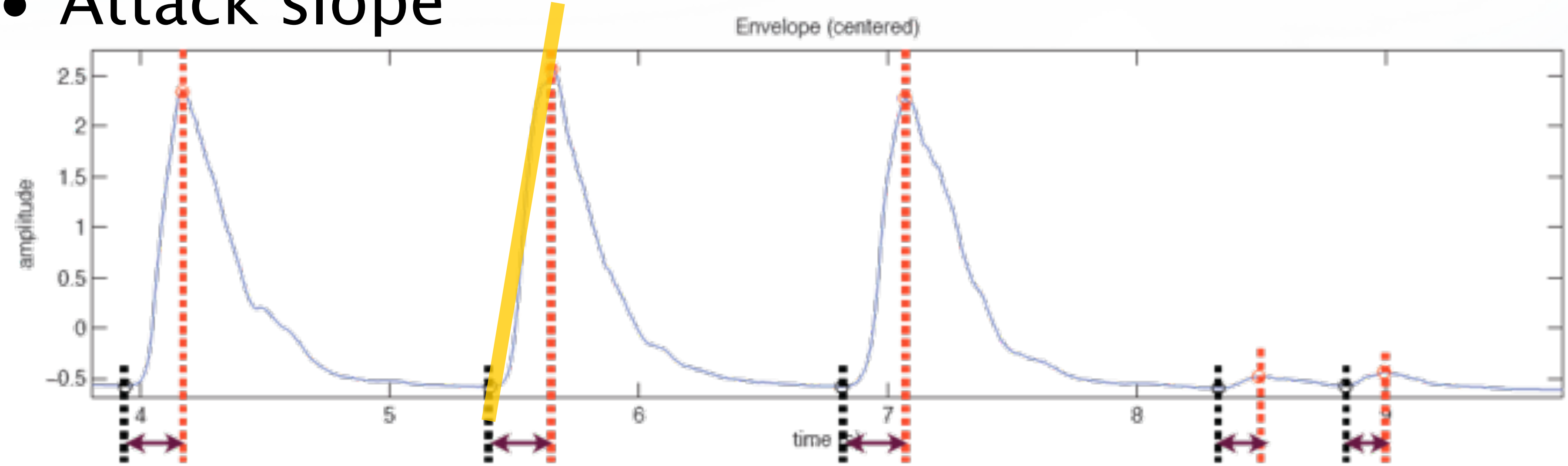
Temporal Information: Attack Features



Picture courtesy: Olivier Lartillot

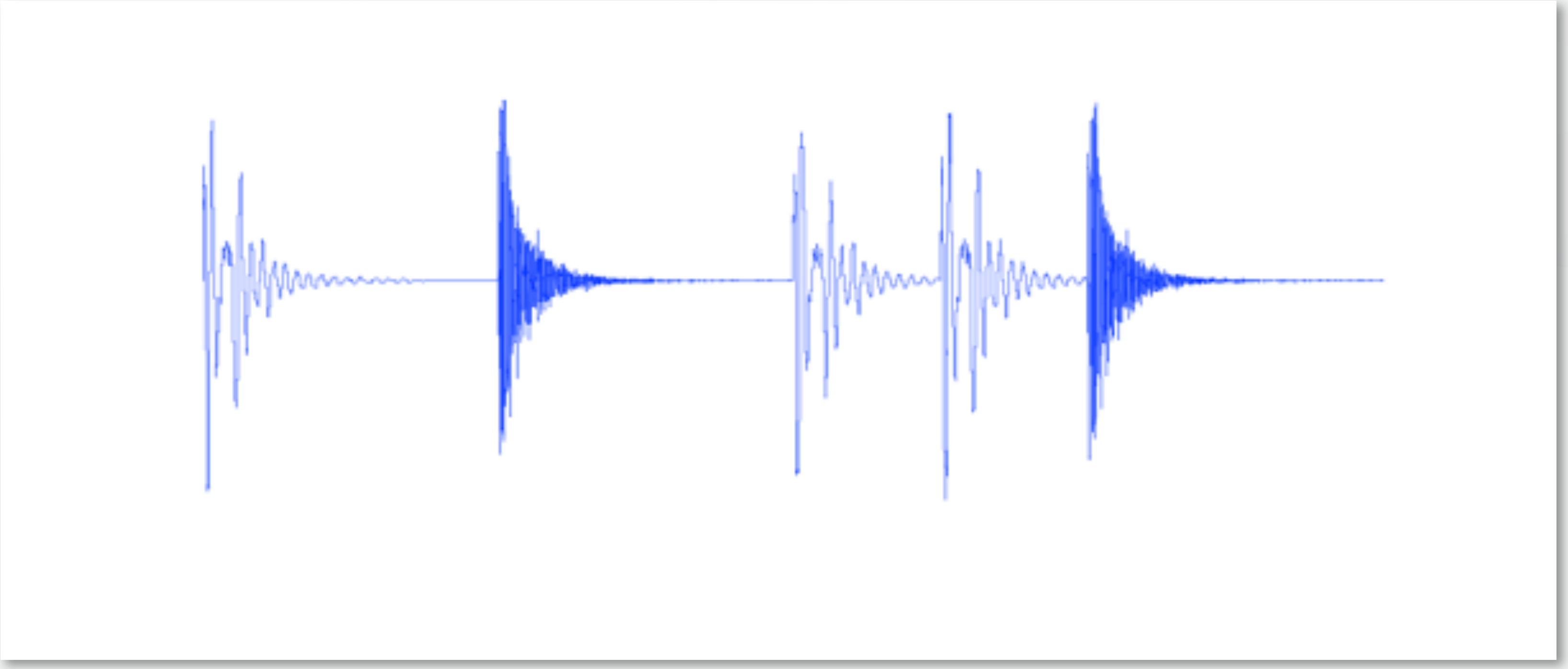
Temporal Information: Attack Features

- Rise time or Attack time – time interval between the onset and instant of maximal amplitude
- Attack slope

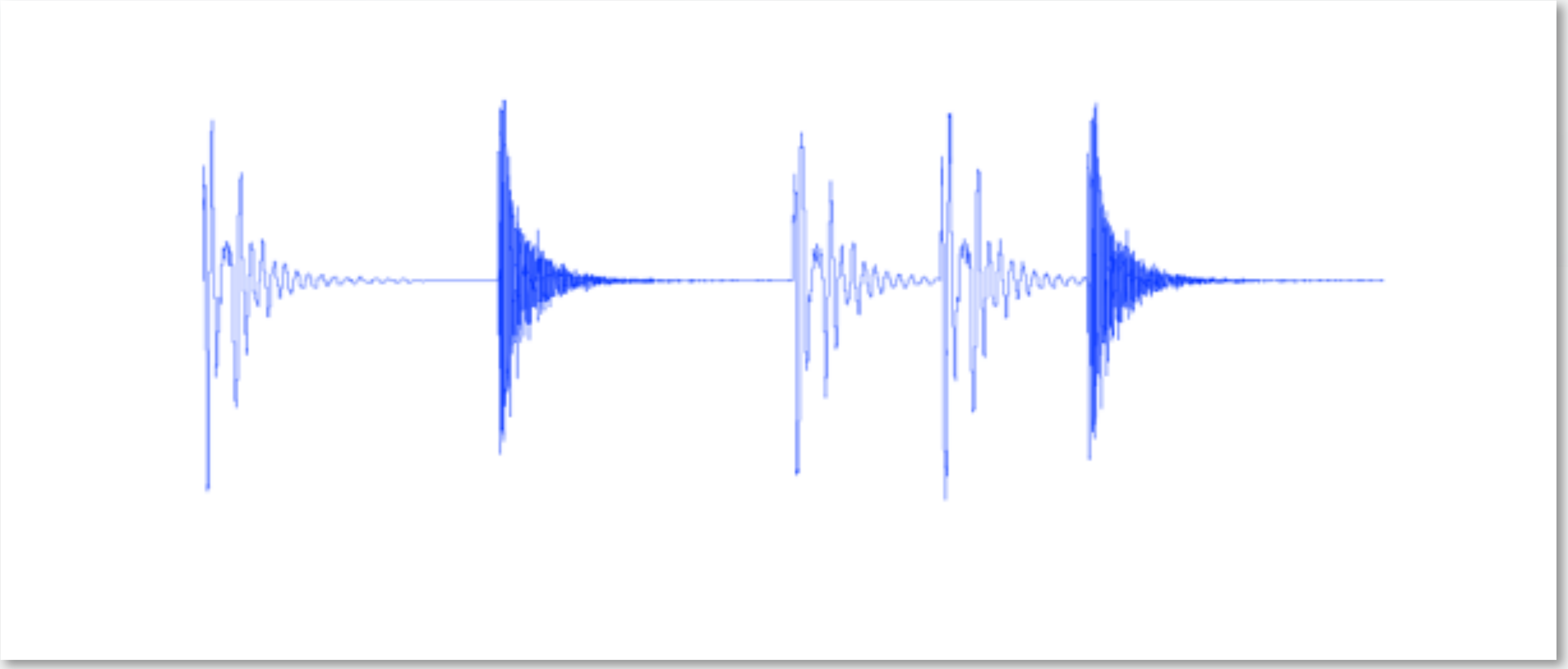


Picture courtesy: Olivier Lartillot

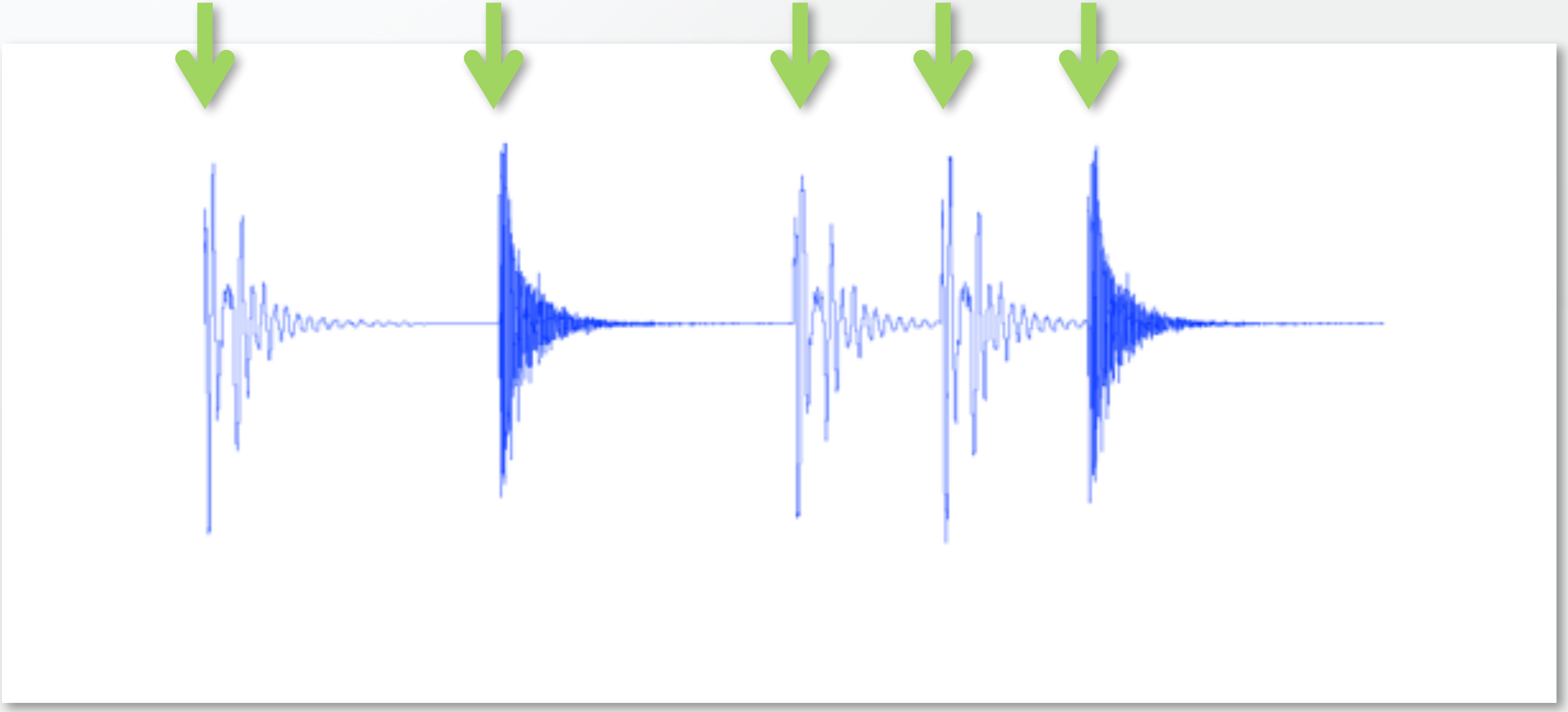
Onset segmentation→Analysis frame



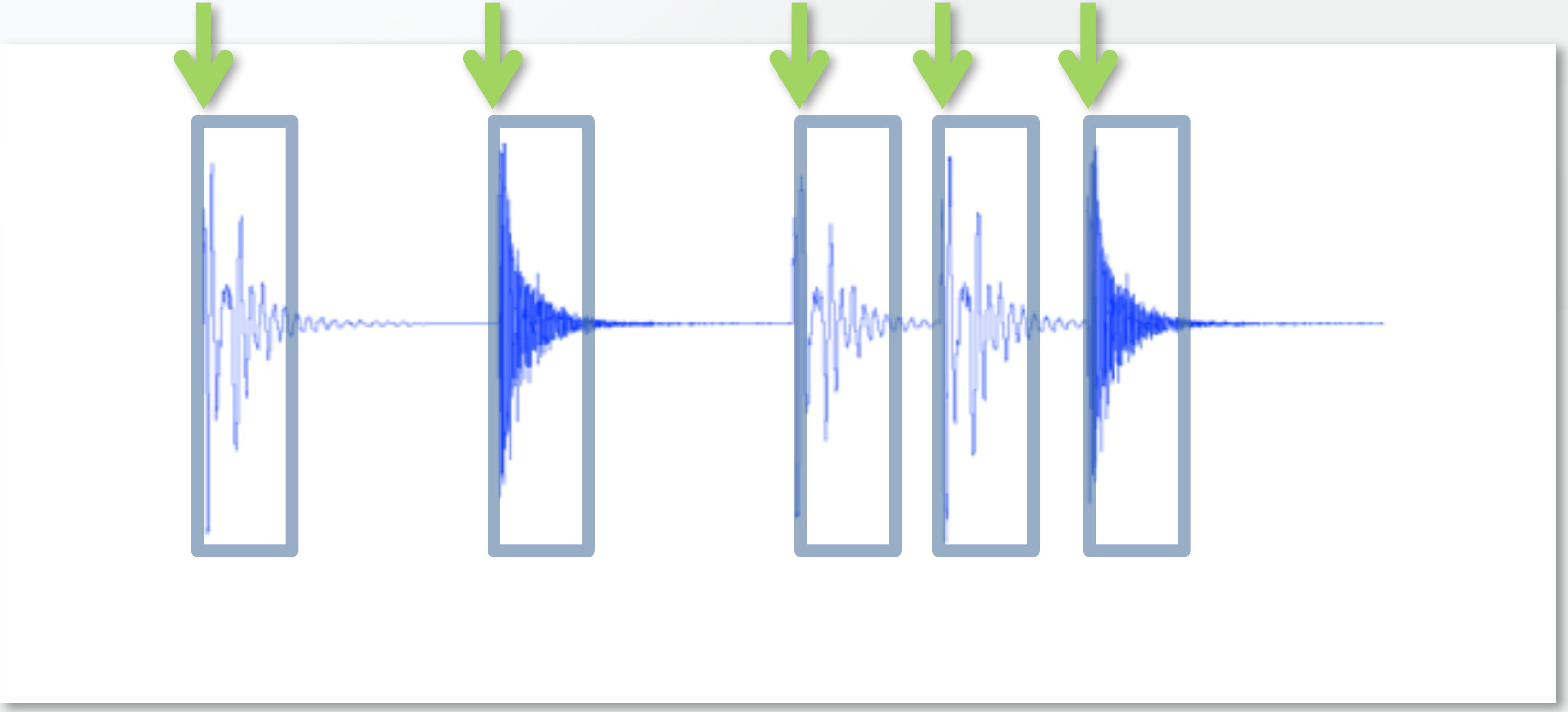
Onset segmentation→Analysis frame



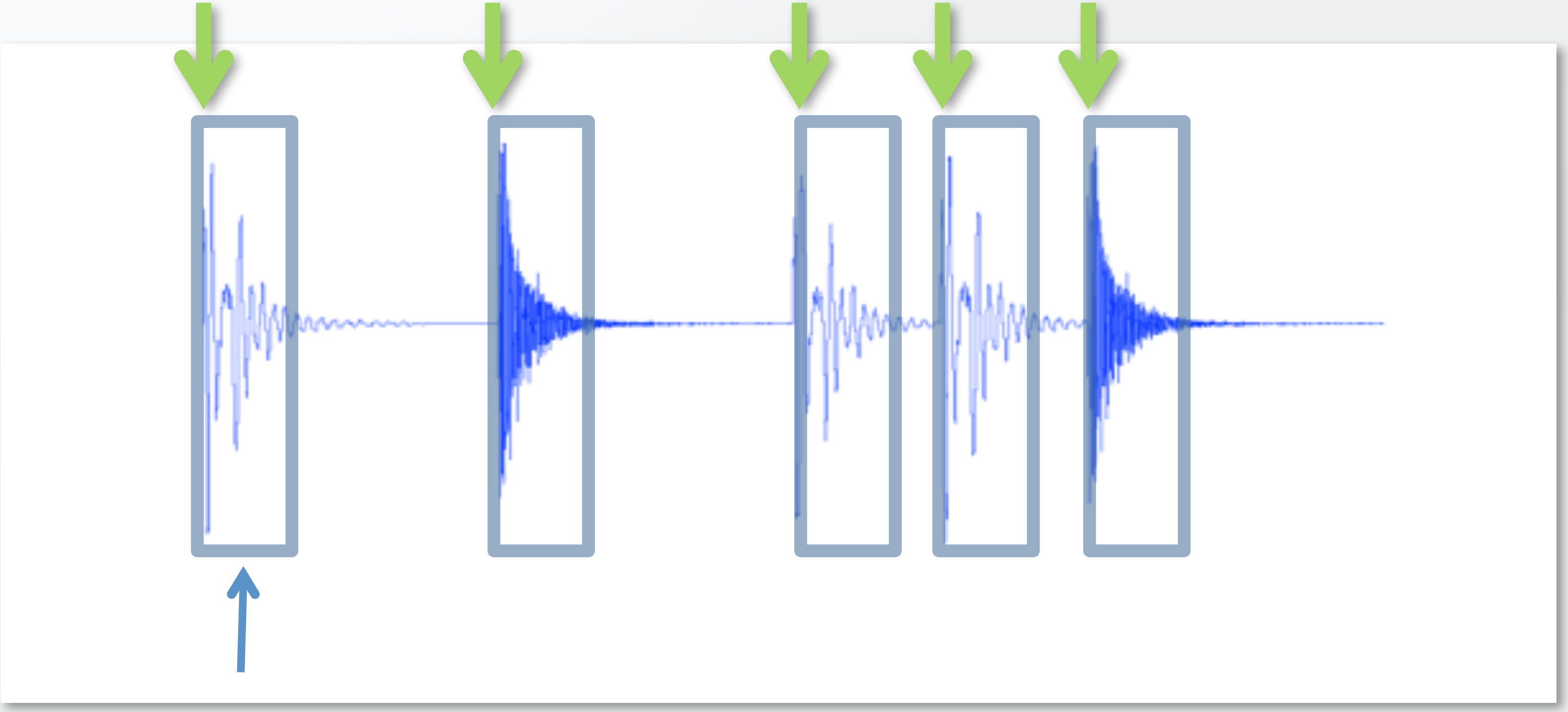
Onset segmentation→Analysis frame



Onset segmentation→Analysis frame

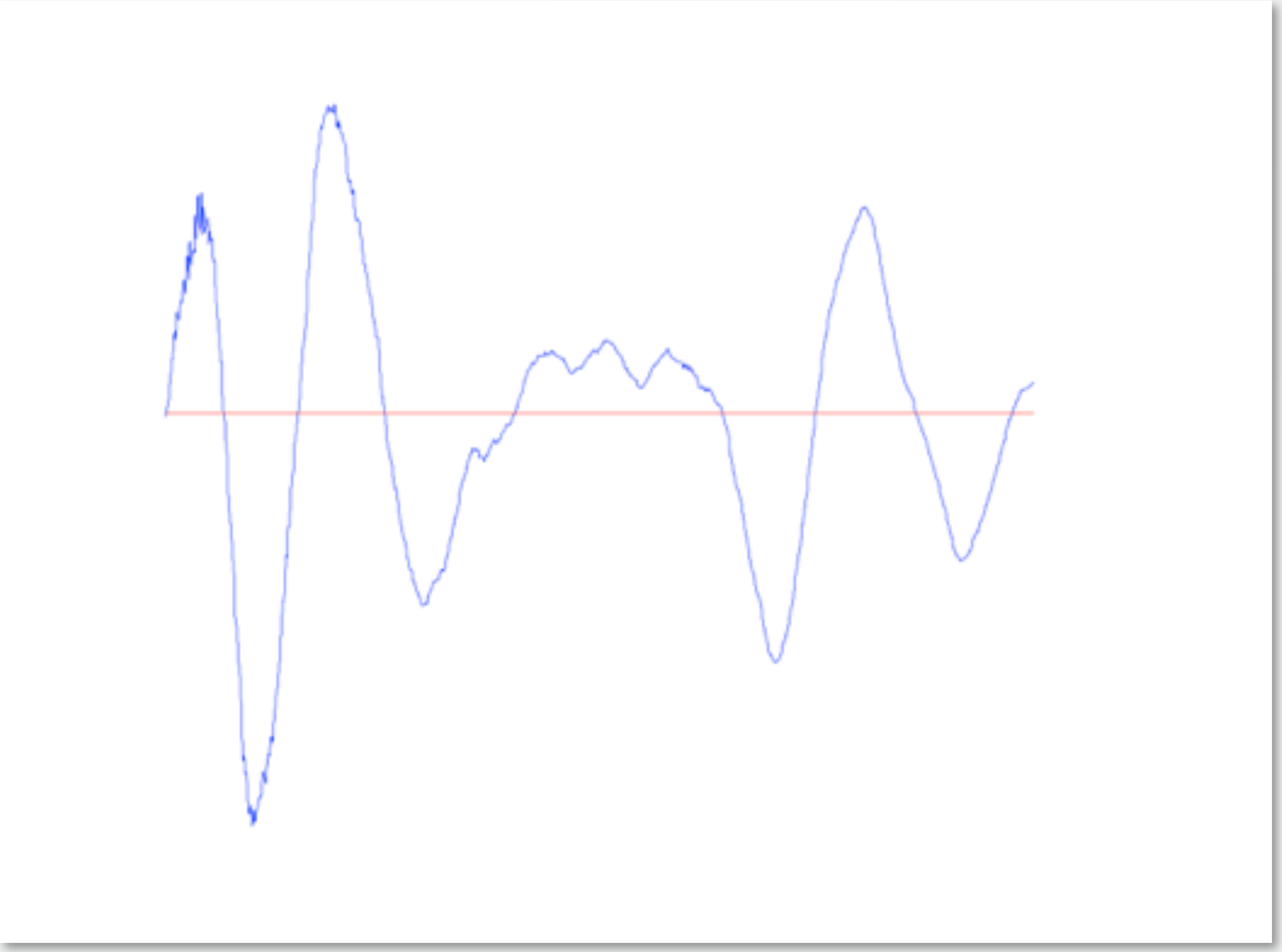


Onset segmentation→Analysis frame

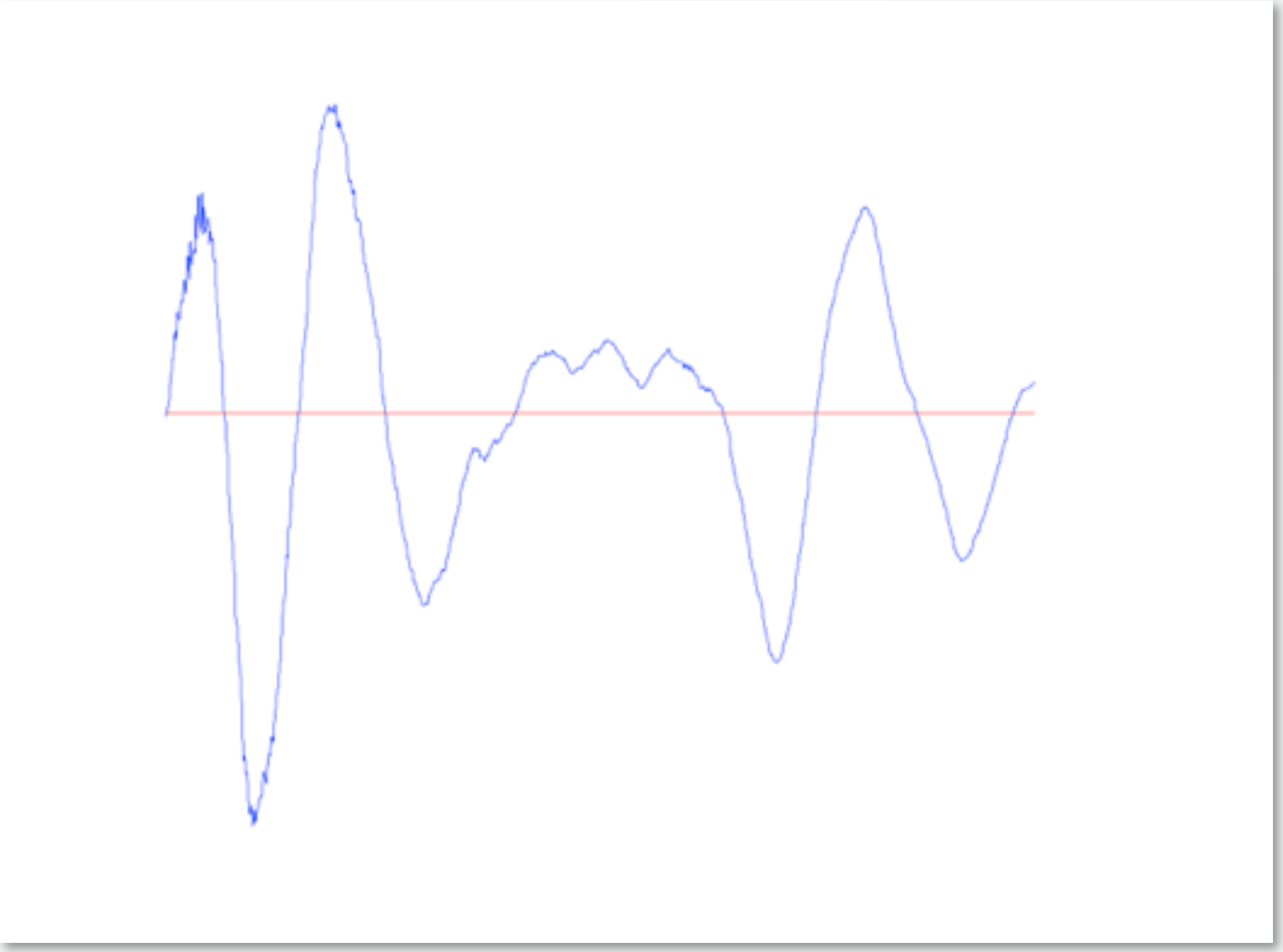


Frame 1

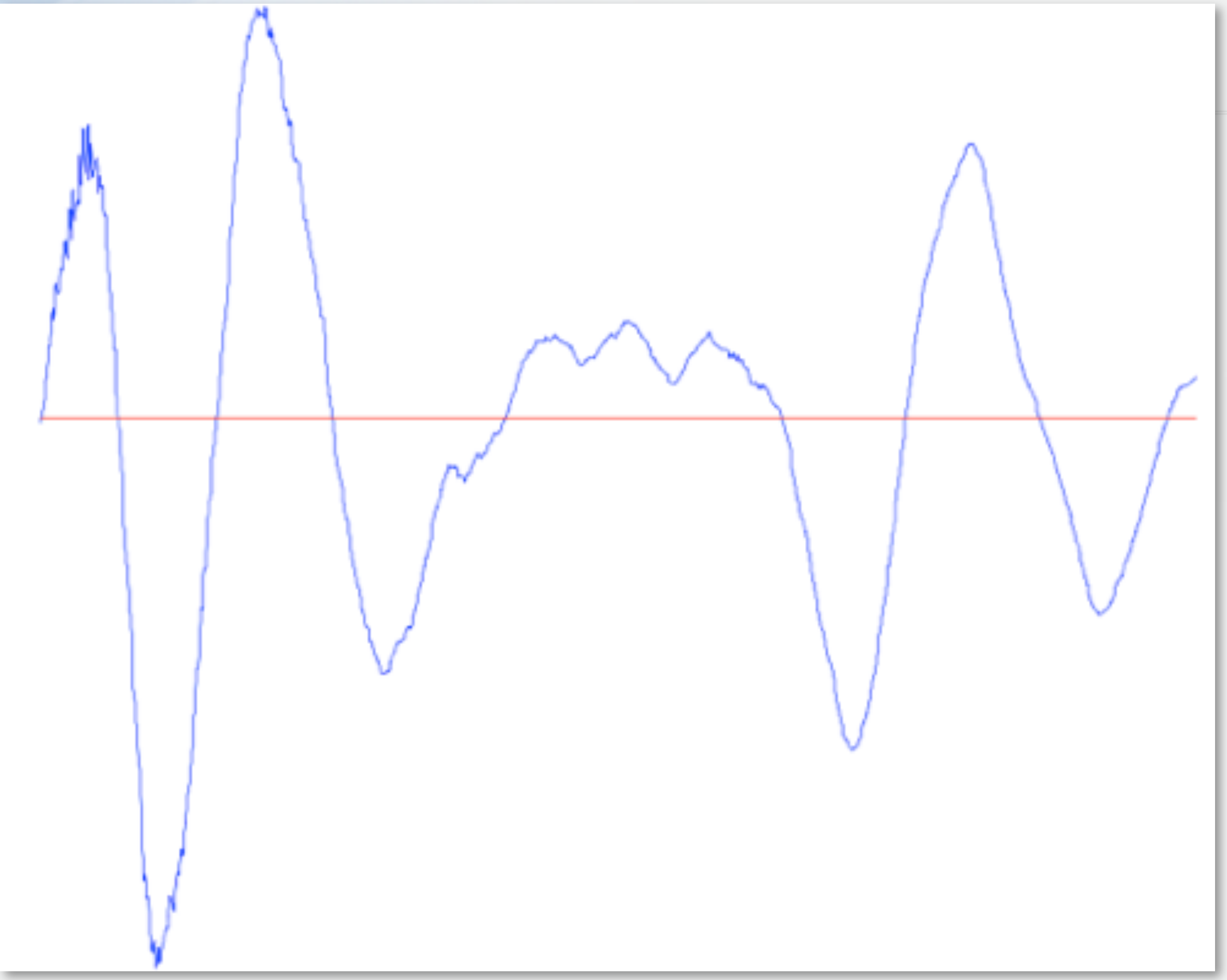
Frame 1



Frame 1

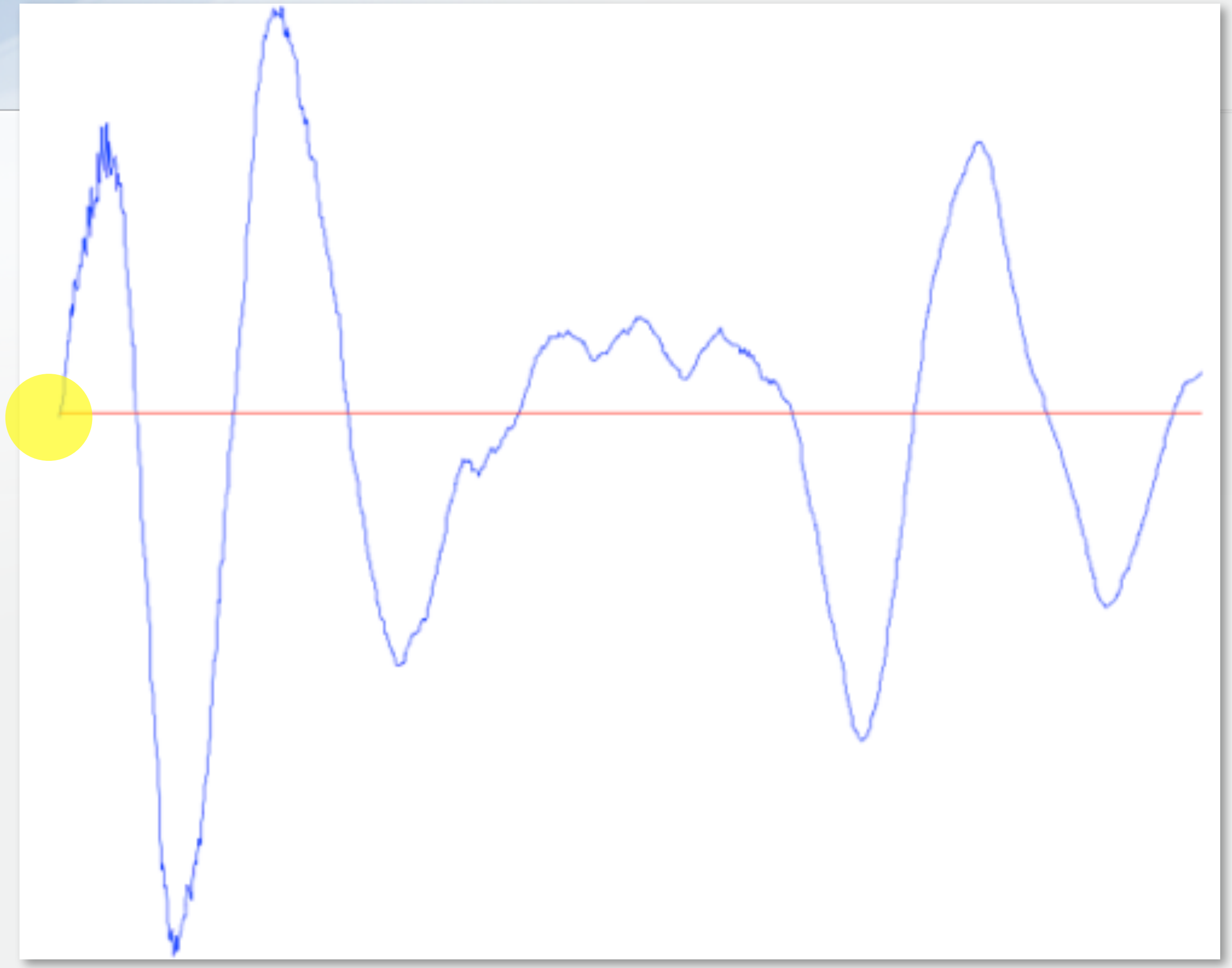


Zero crossing rate



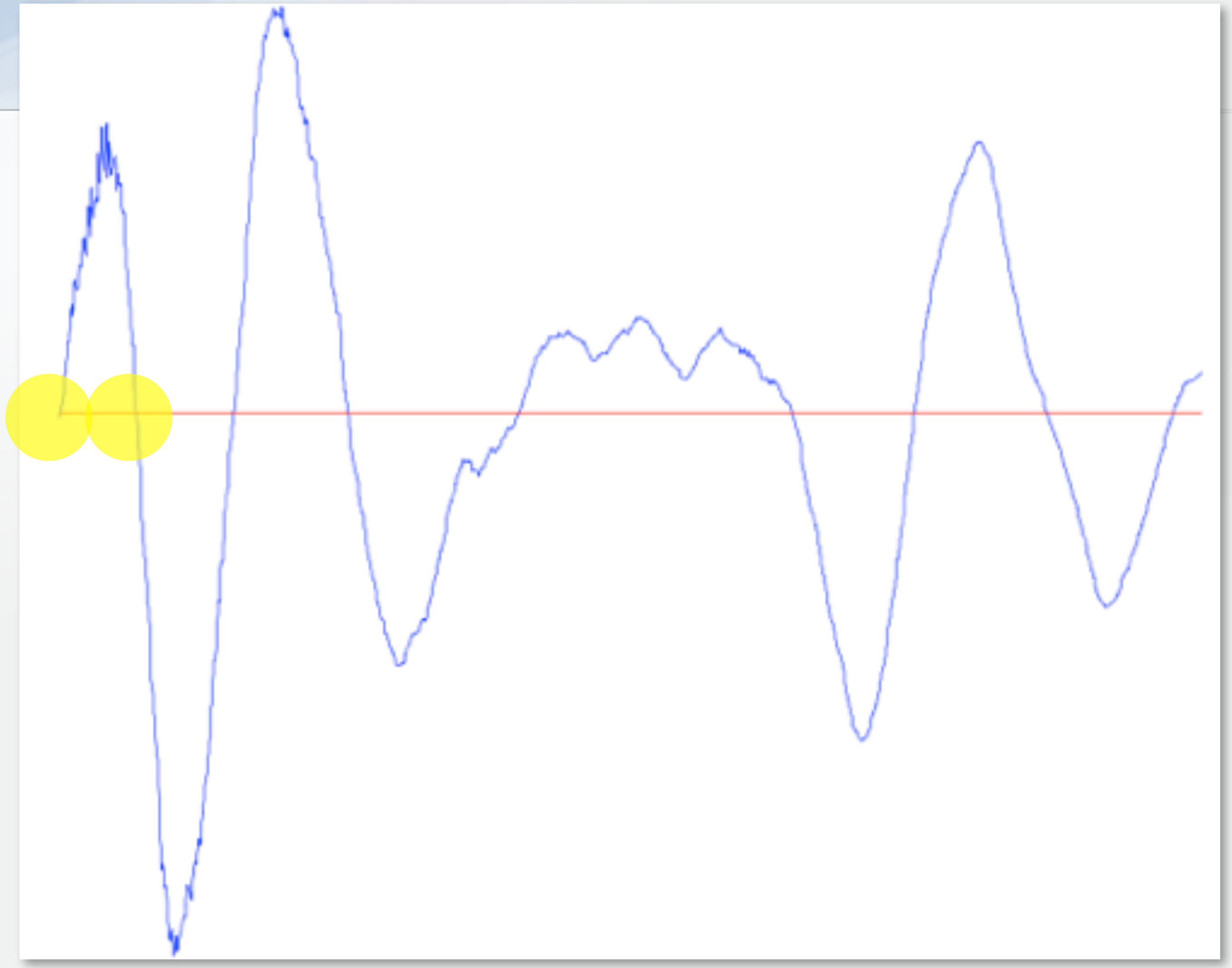
Frame 1

Zero crossing rate



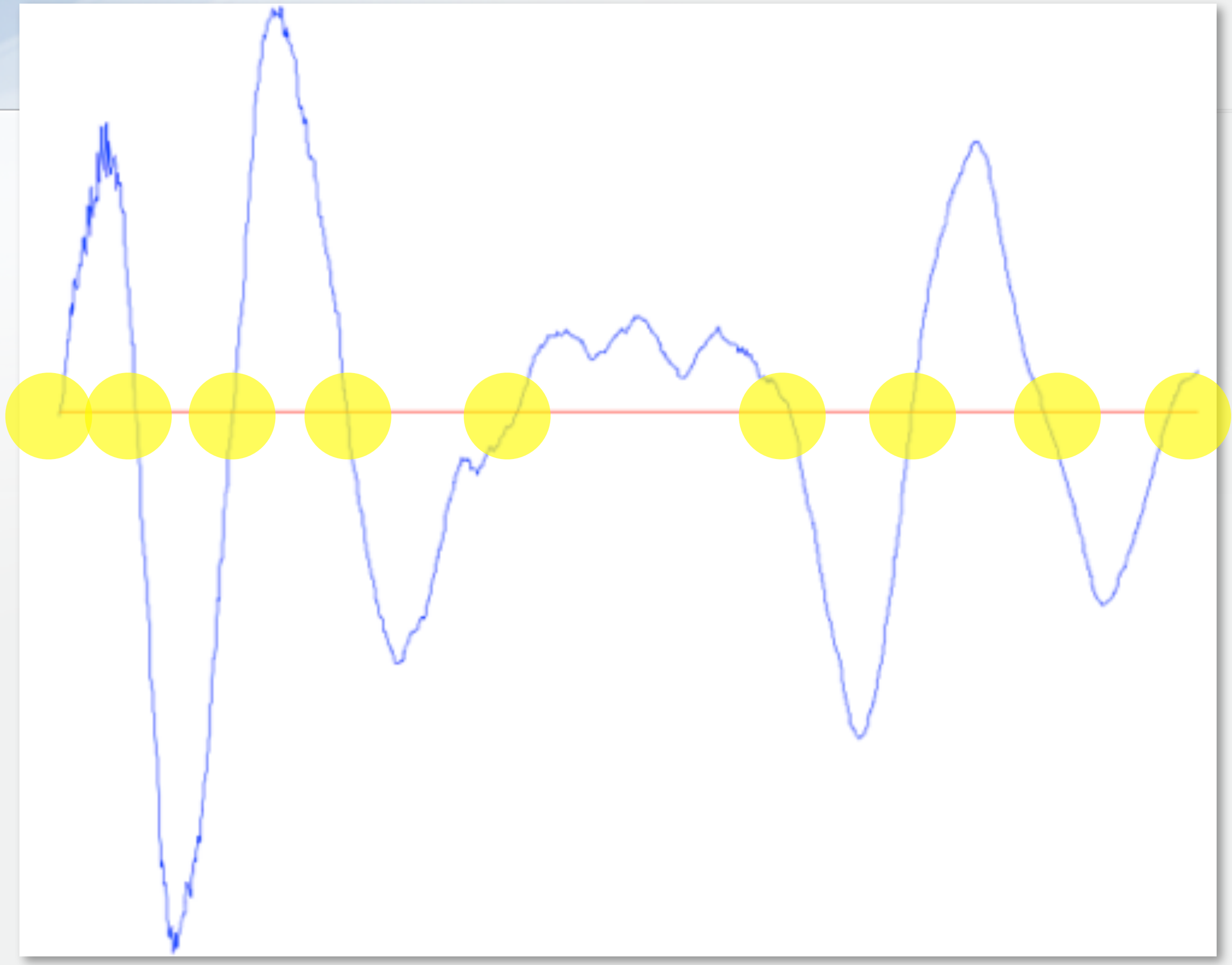
Frame 1

Zero crossing rate



Frame 1

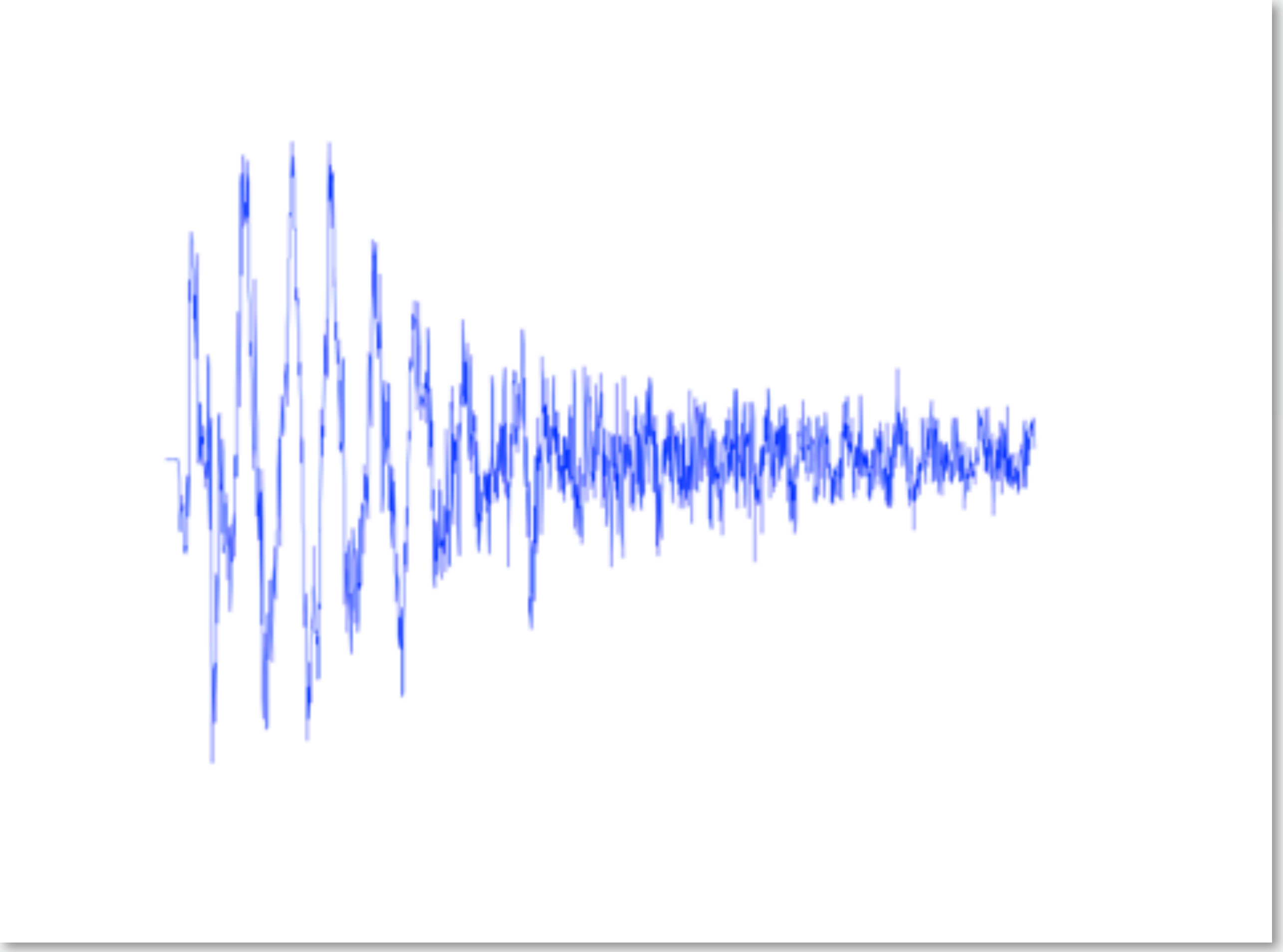
Zero crossing rate



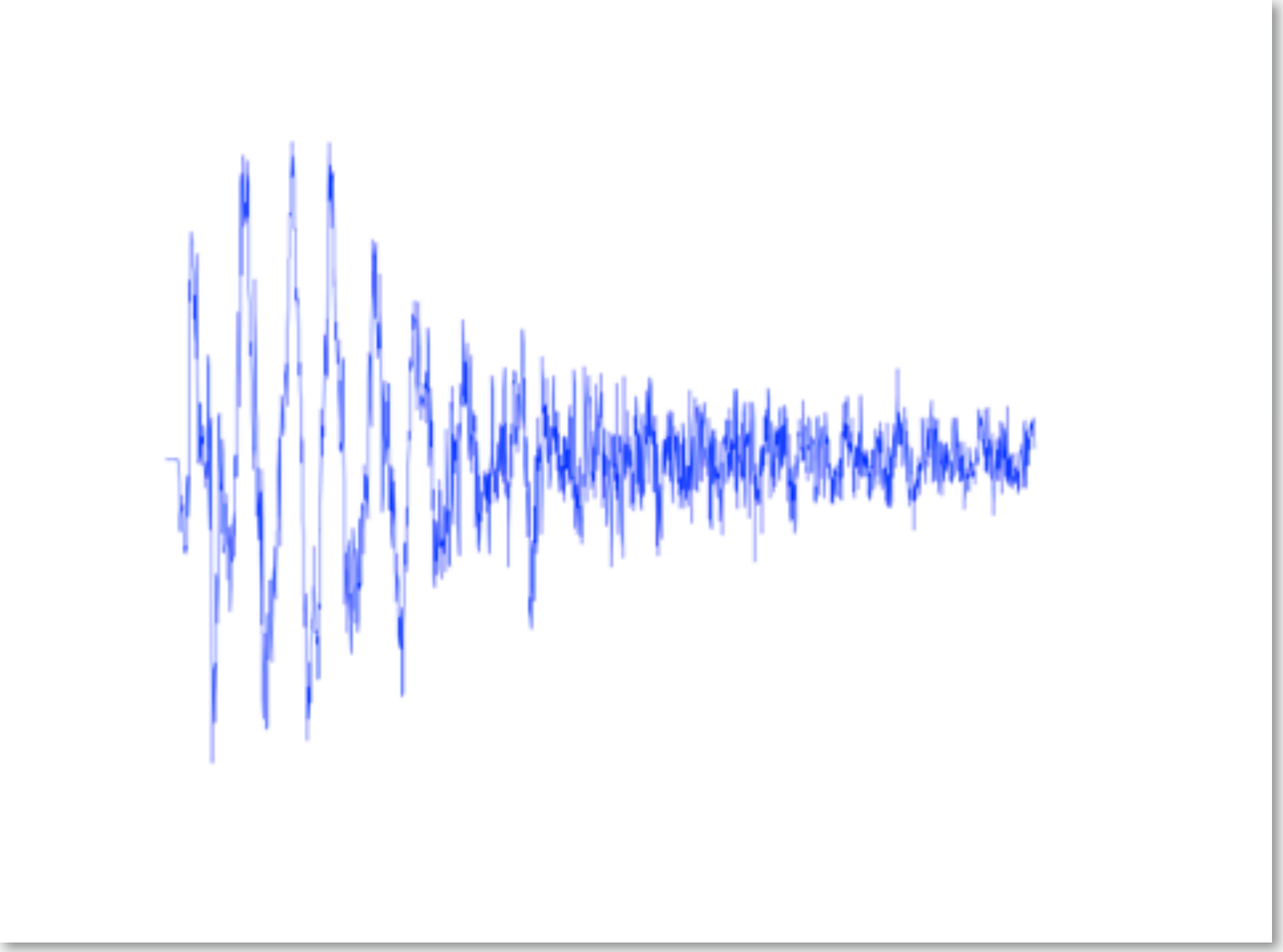
Frame 1

Zero crossing rate = 9

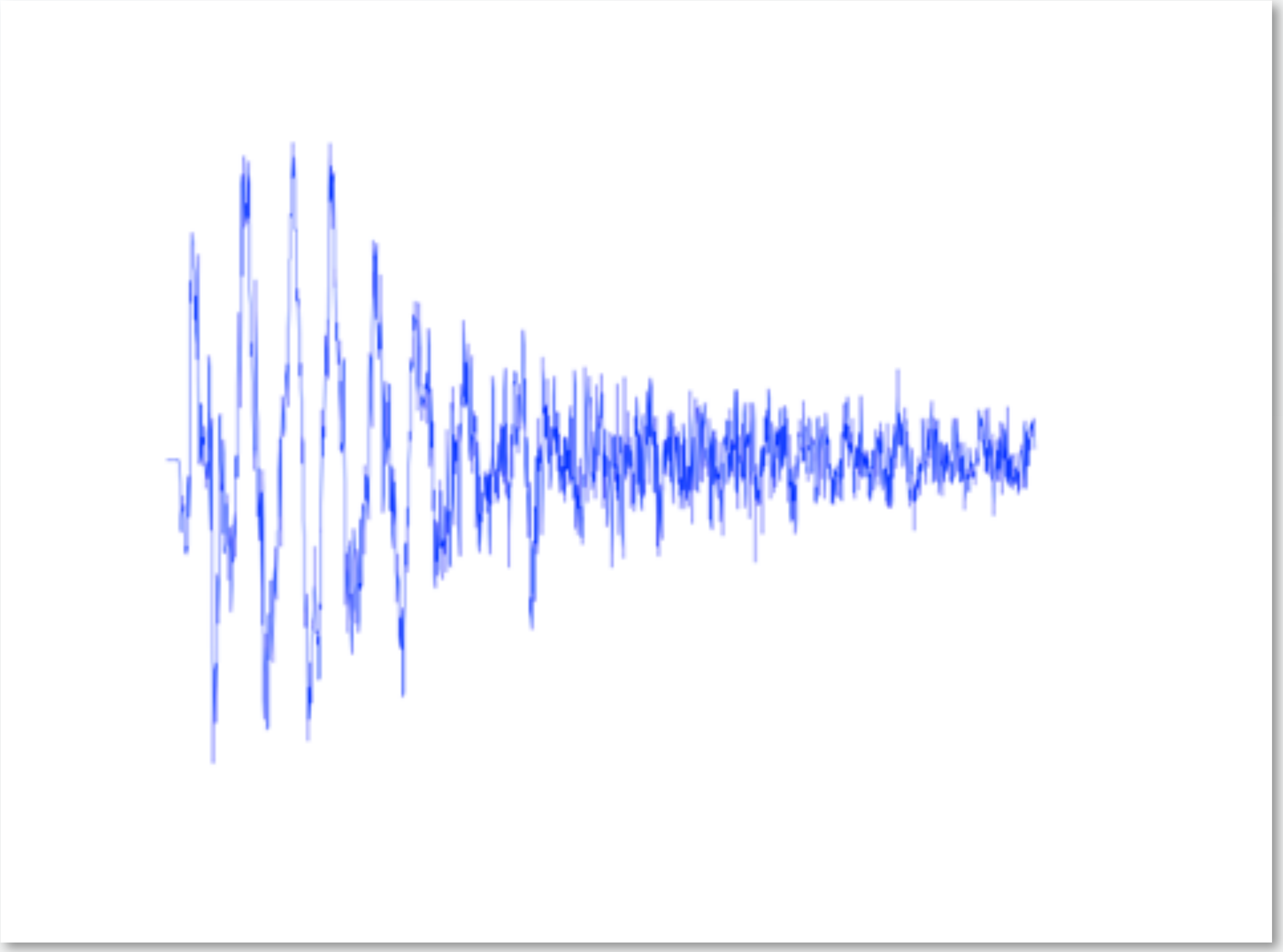
Frame 2



Frame 2



Frame 2



Zero crossing rate = 423

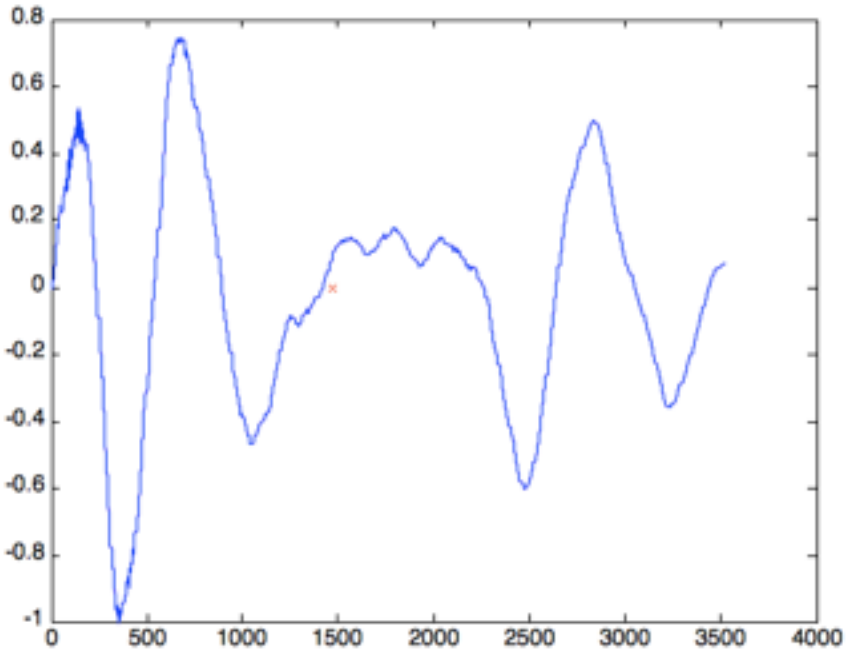
Features : SimpleLoop.wav

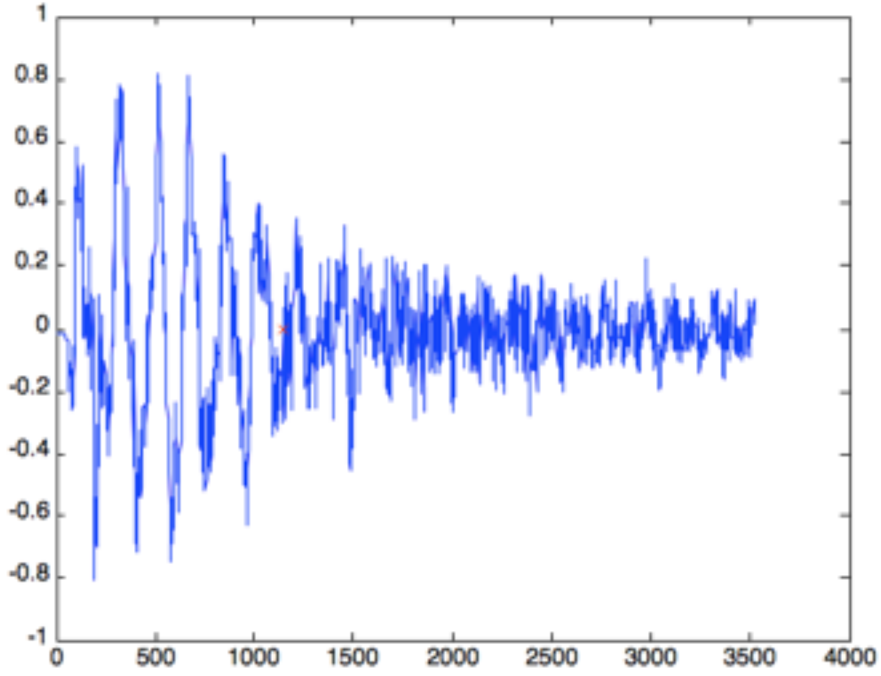
Frame	ZCR
1	9
2	423
3	22
4	28
5	390

Warning: example results only - not actual results from audio analysis…

Temporal Centroid

$$C_t = \frac{\sum_t tE(t)}{\sum_t E(t)}$$

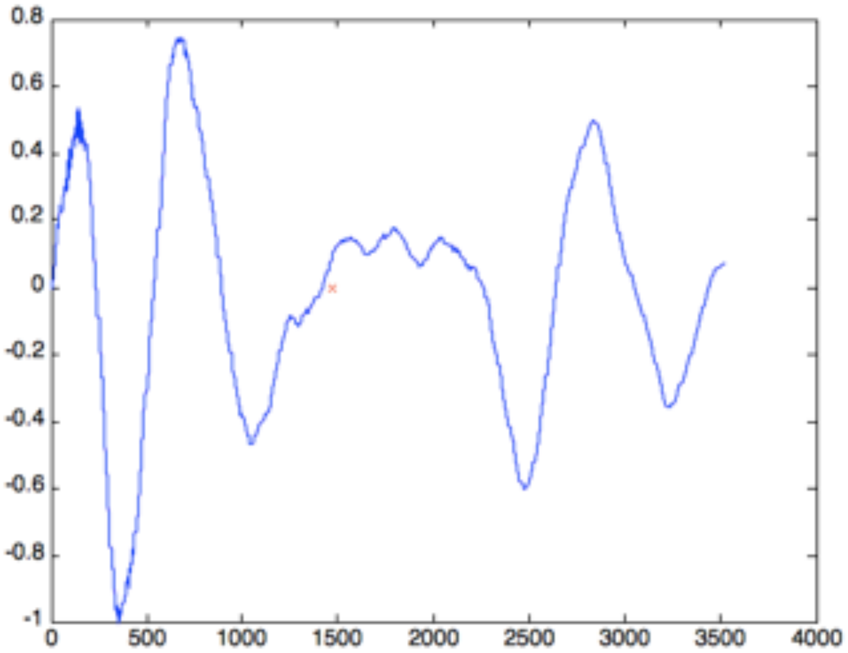


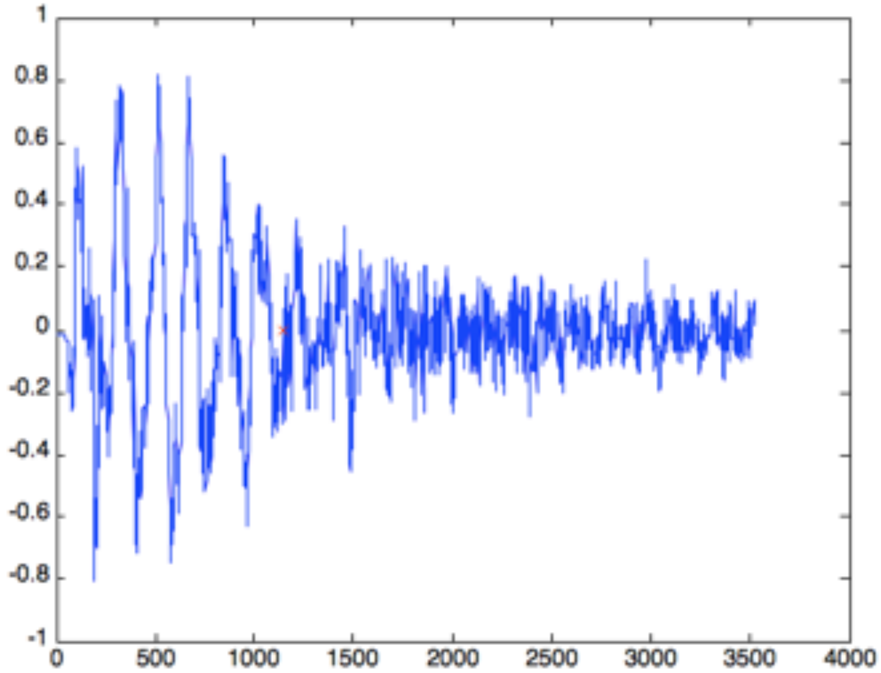


Temporal Centroid

- “Balancing point” of event energy ⇒ short vs. long.

$$C_t = \frac{\sum_t tE(t)}{\sum_t E(t)}$$

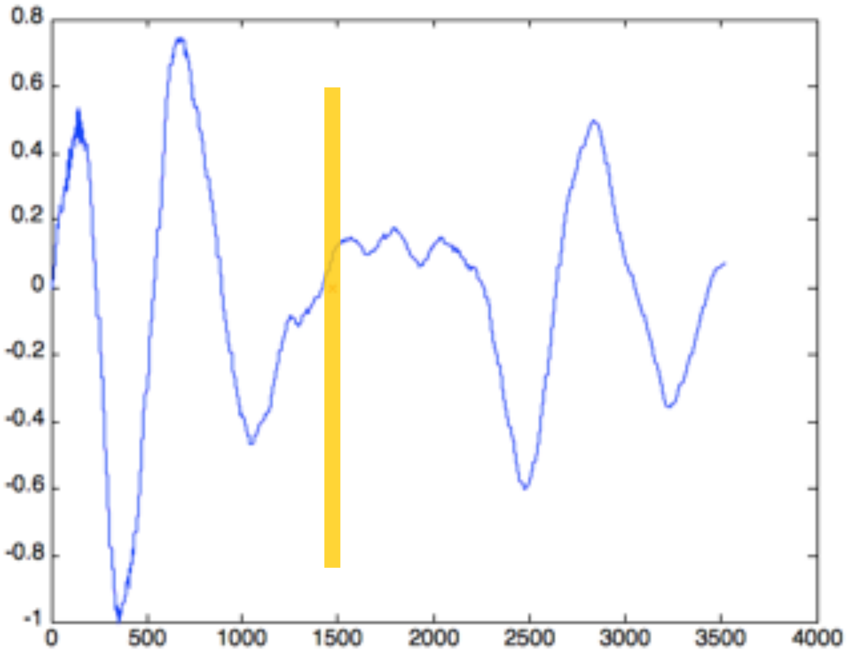


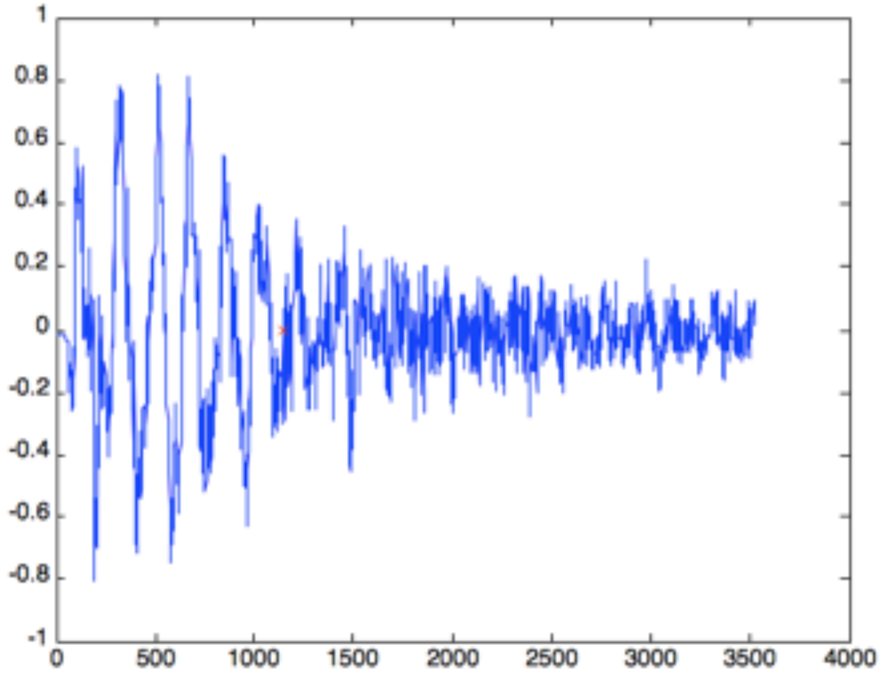


Temporal Centroid

- “Balancing point” of event energy ⇒ short vs. long.

$$C_t = \frac{\sum_t tE(t)}{\sum_t E(t)}$$

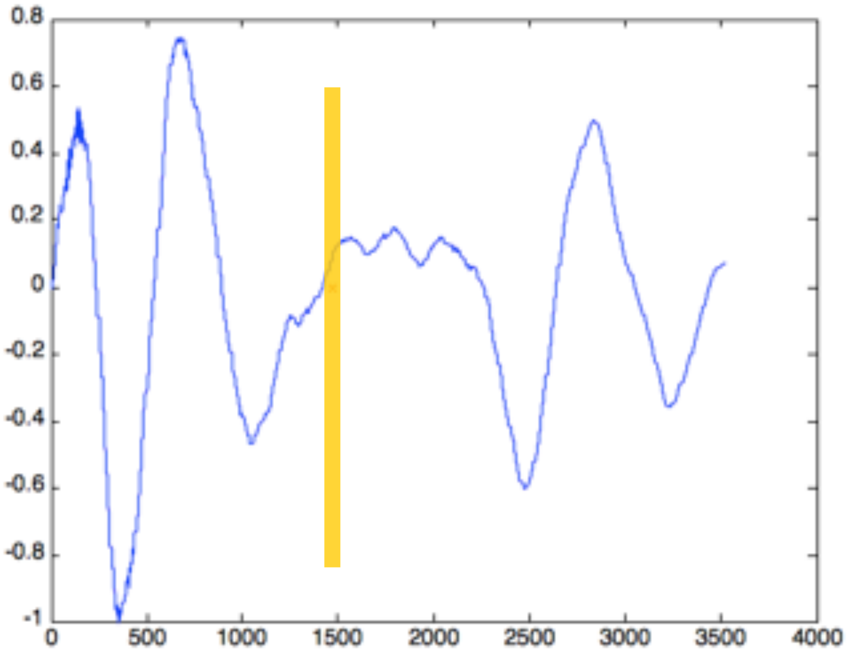


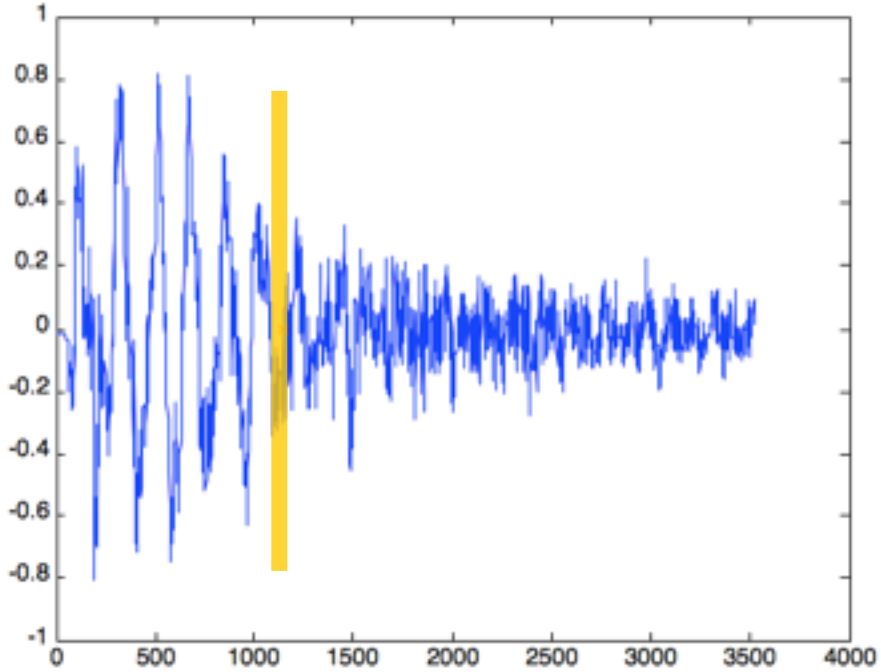


Temporal Centroid

- “Balancing point” of event energy ⇒ short vs. long.

$$C_t = \frac{\sum_t tE(t)}{\sum_t E(t)}$$





To Be Continued…

- Frequency Domain features.

Frequency-domain Features

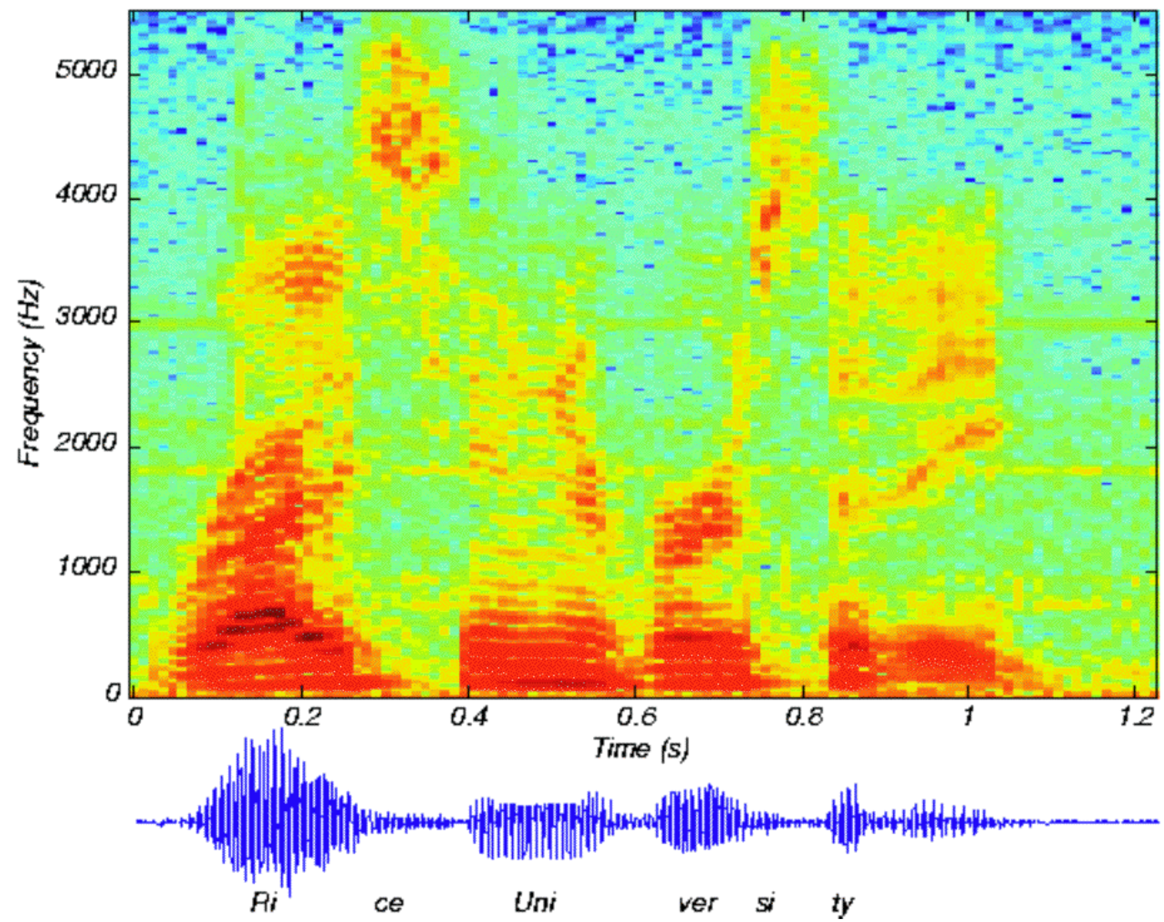
- Spectrum, Spectral bins
  - Window/hop sizes
  - Improving spectral data: phase unwrapping, time realignment
- Spectral measures (statistical moments)
- MFCCs
- Peak-picking and peak-tracking
- Pitch-estimation and pitch-tracking

Frequency-domain Analysis

- Short-time Fourier transform
  - Configuration options and trade-offs
  - Interpretation/weighting of spectral bins (perceptual scales)
- Other frequency-domain techniques
  - Filter banks
  - Linear prediction
  - Filter matching
  - Multiresolution techniques (i.e. Wavelets)
- Many options!

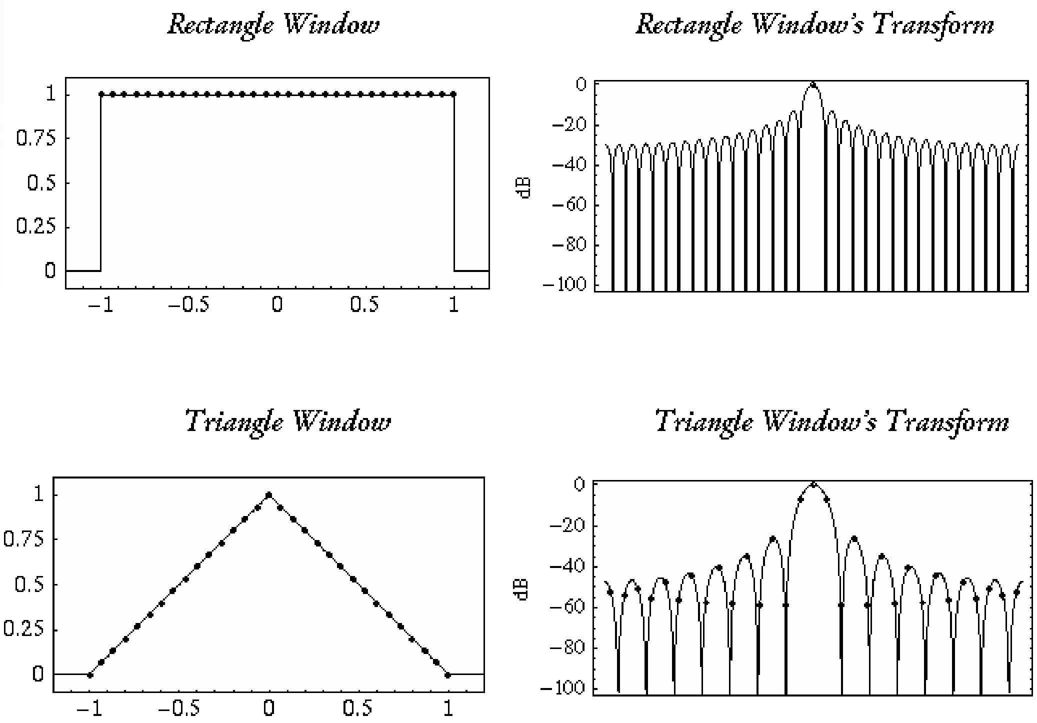
Example Speech Spectrogram

- Kinds of spectral plots
- Short term spectral energy representation.
- Features can be derived from 2D spectral representation.

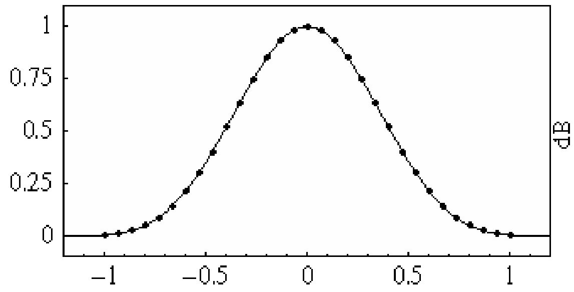


Windows and their Spectra

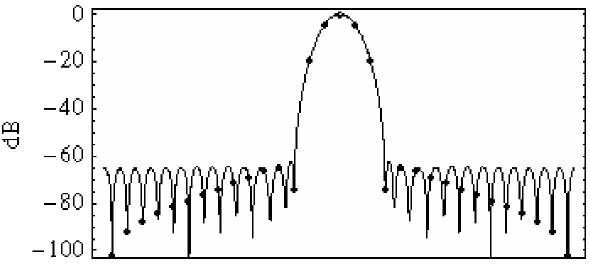
- Trade-offs between window characteristics
- Different windows for different analysis domains



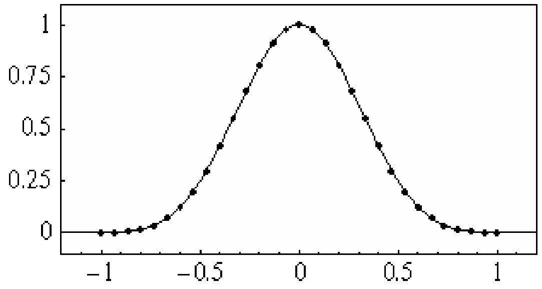
Advanced Windows for Spectral Analysis


Blackman Window

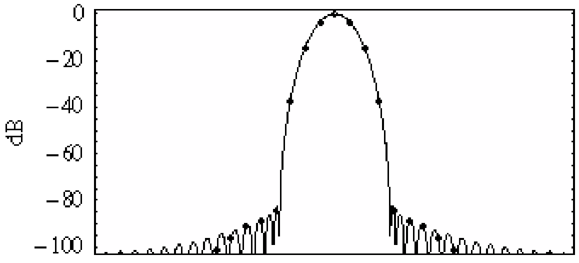
$a_0 = 0.427$, $a_1 = 0.497$, $a_2 = 0.077$


Blackman Window's Transform

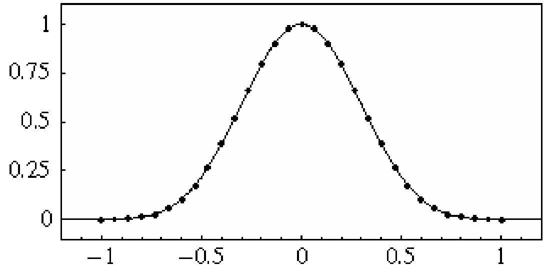
$a_0 = 0.427$, $a_1 = 0.497$, $a_2 = 0.077$


Kaiser Window

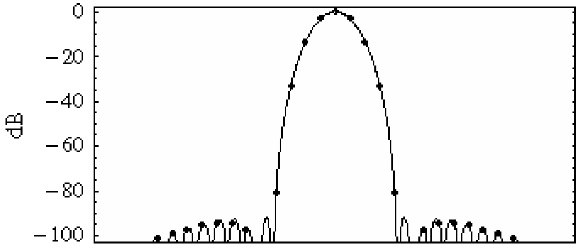
(α = 3.5)


Kaiser Window's Transform

(α = 3.5)

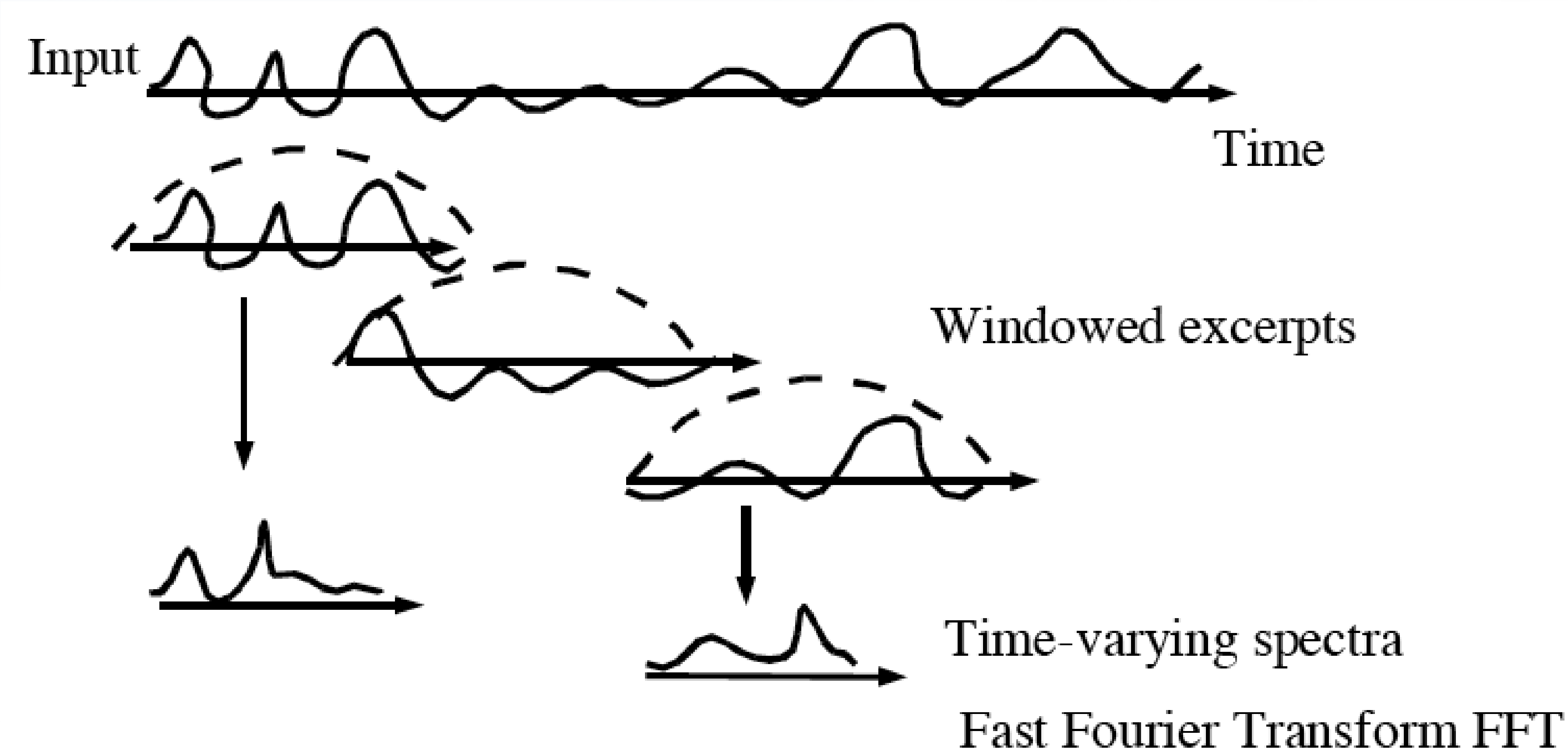

Blackman-Harris Window

(4-term , -96 dB)

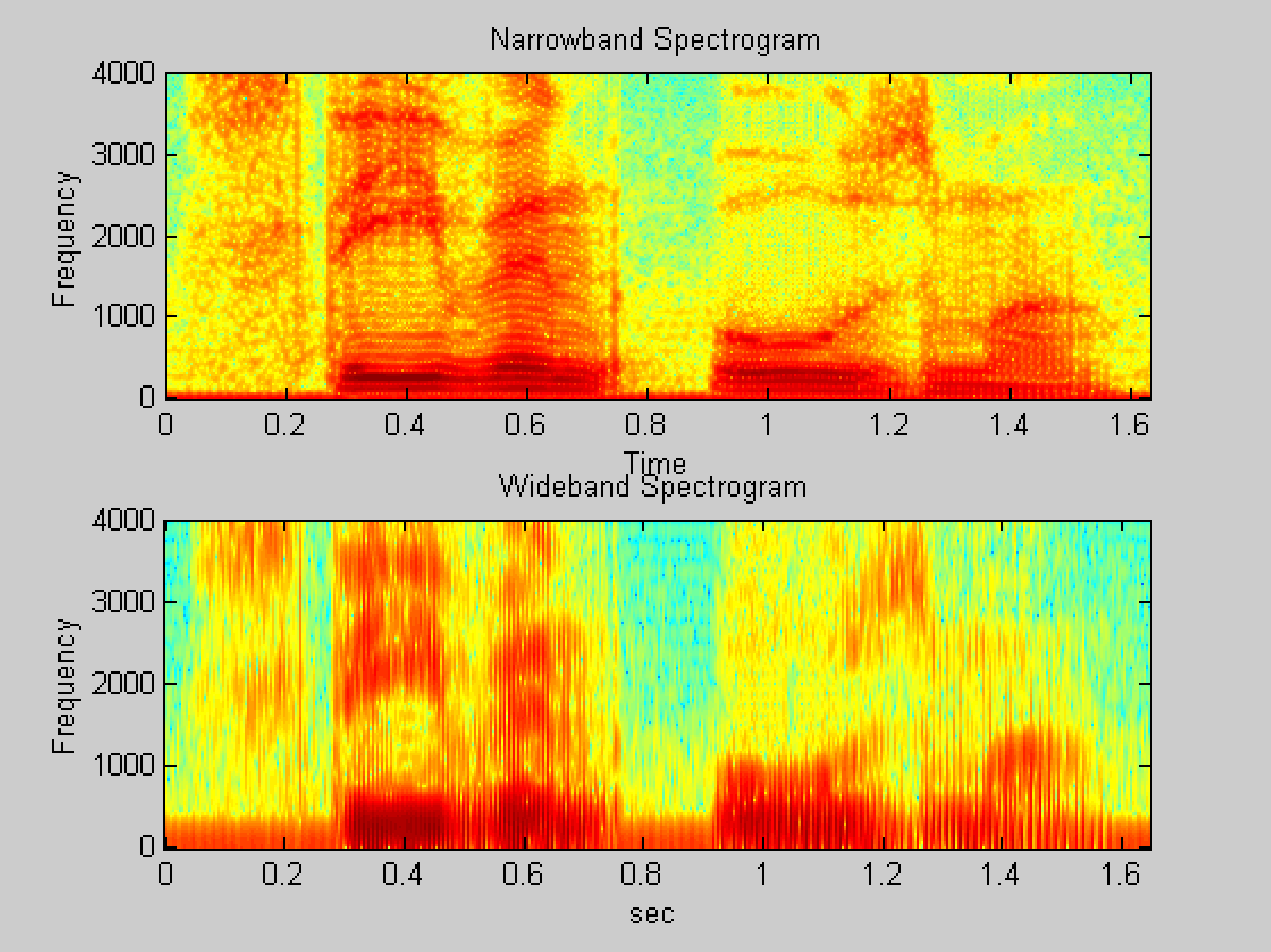

Blackman-Harris Window's Transform

(4-term , -96 dB)

Windowing and the STFT

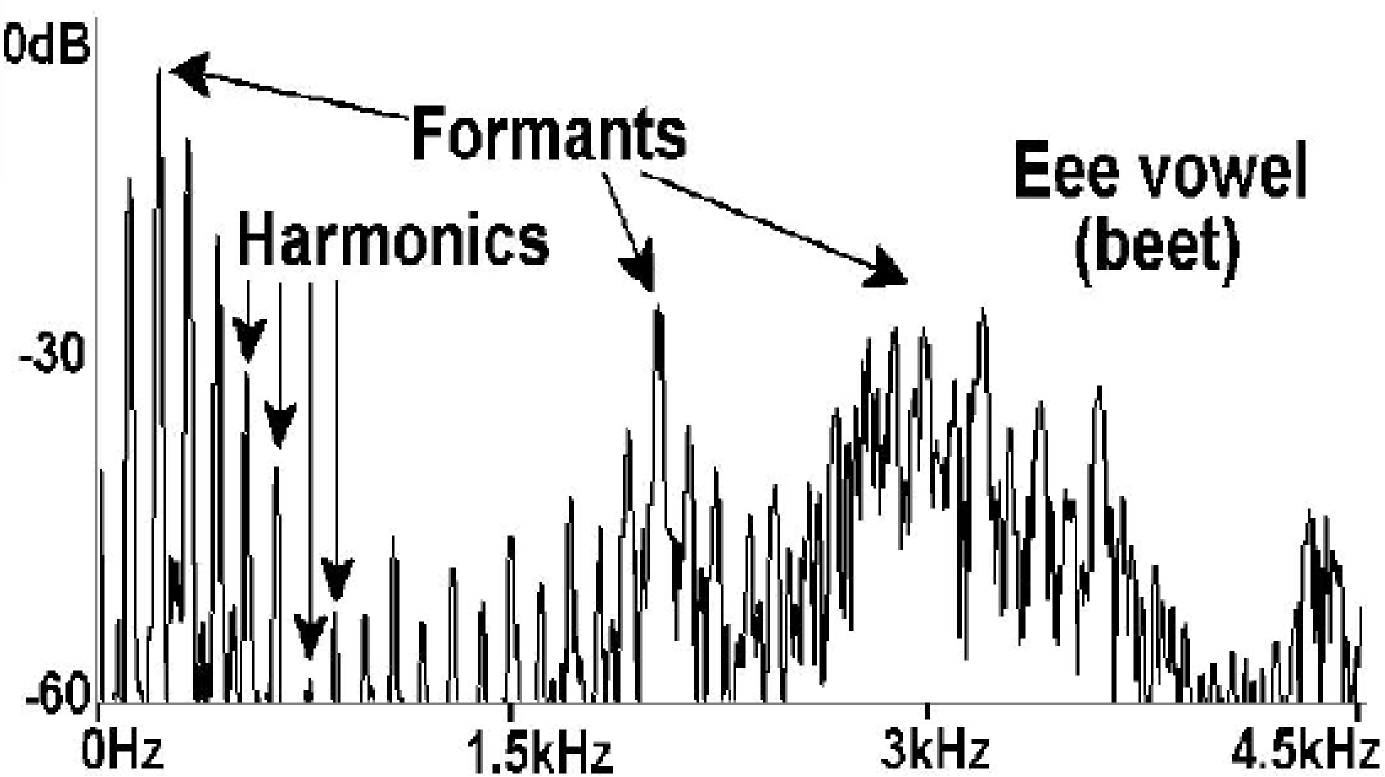


The Pitch/Time Trade-off

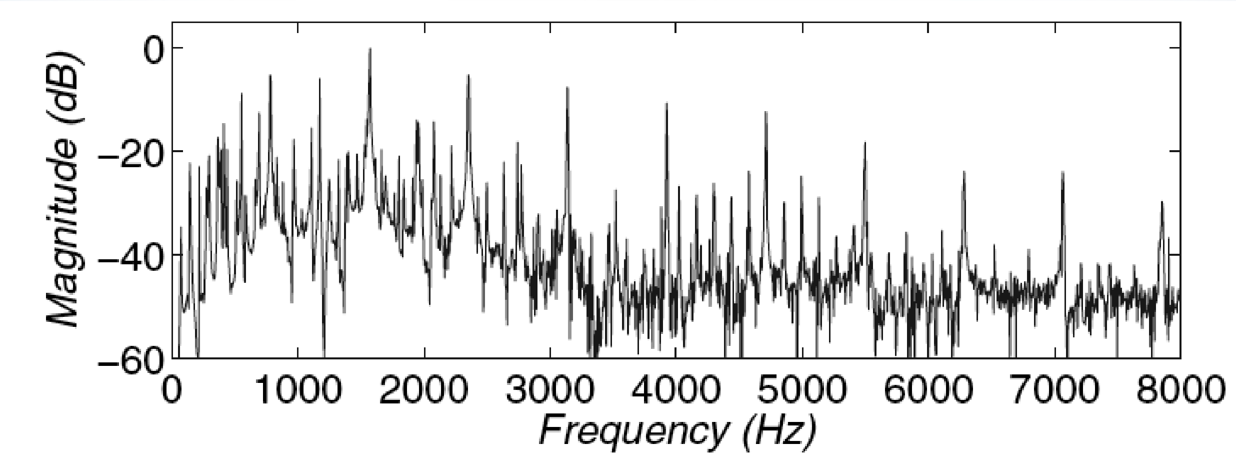


Harmonics and Formants

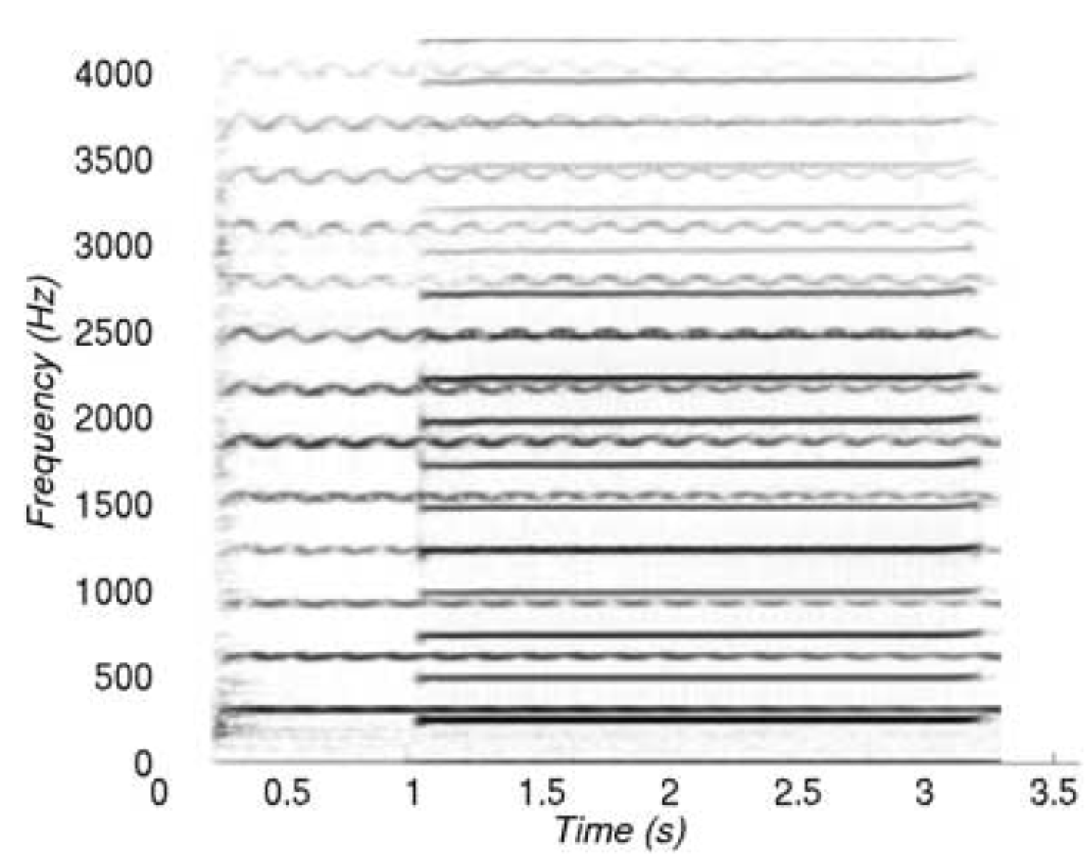
- Source/Filter – instrument resonances



Composite Spectra



- How to disambiguate?
- Track birth/death statistics
- Vibrato (see figure)
- Statistical techniques



Spectral Analysis and Interpretation

- Spectral data extraction
  - Base frequency
  - Overtone spectrum
  - Formants, resonances, regions
  - Instrument signatures
- Spectral statistics
  - Peak, mean, average, centroid, slope, etc.
  - Spectral variety, etc.

Spectral Features

- Spectral Centroid
- Spectral Bandwidth/Spread
- Spectral Skewness
- Spectral Kurtosis
- Spectral Tilt/Slope
- Spectral Roll-Off
- Spectral Flatness Measure

Spectral Features

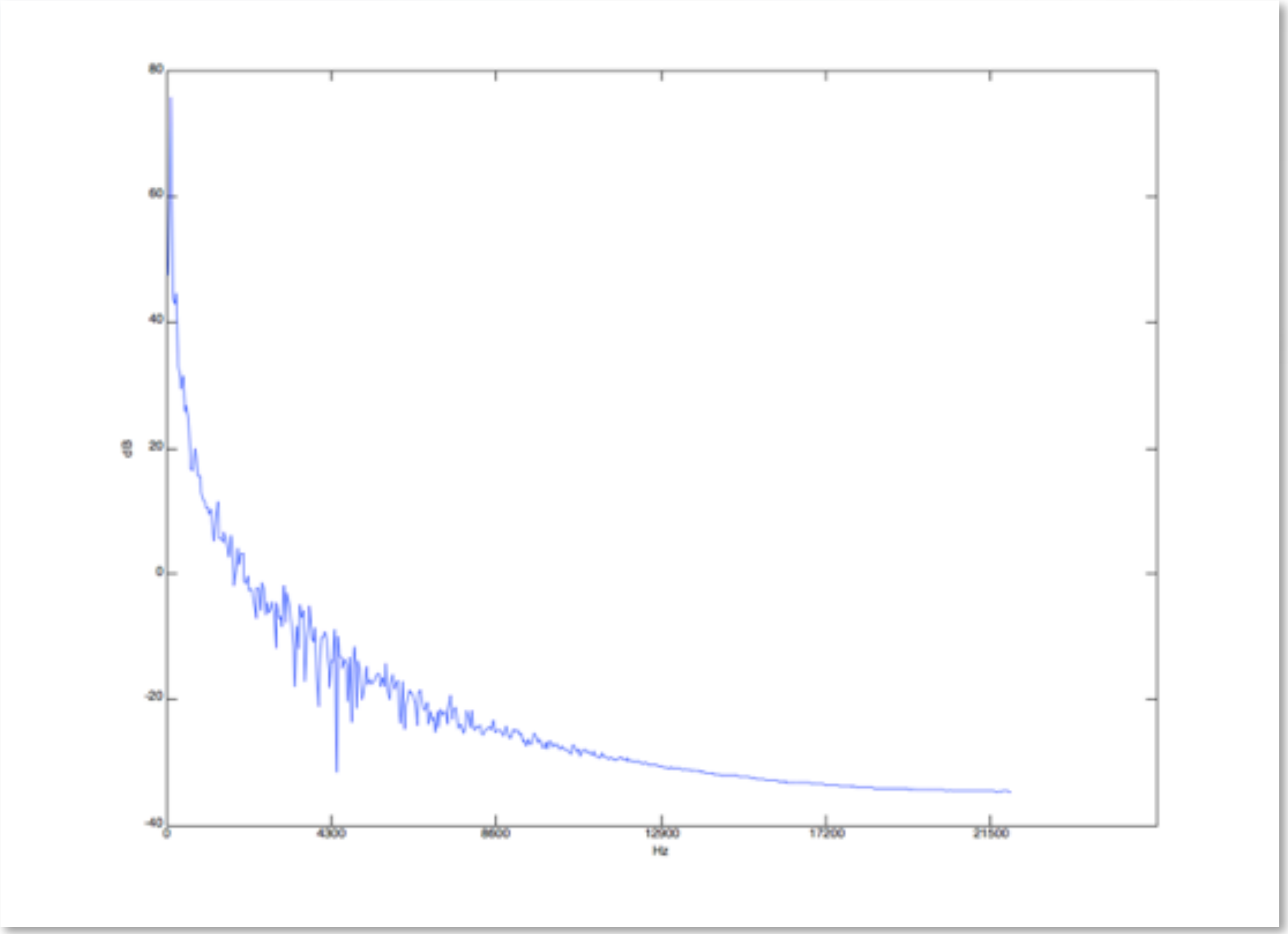
- Spectral Centroid
- Spectral Bandwidth/Spread
- Spectral Skewness
- Spectral Kurtosis
- Spectral Tilt/Slope
- Spectral Roll-Off
- Spectral Flatness Measure

Spectral Features

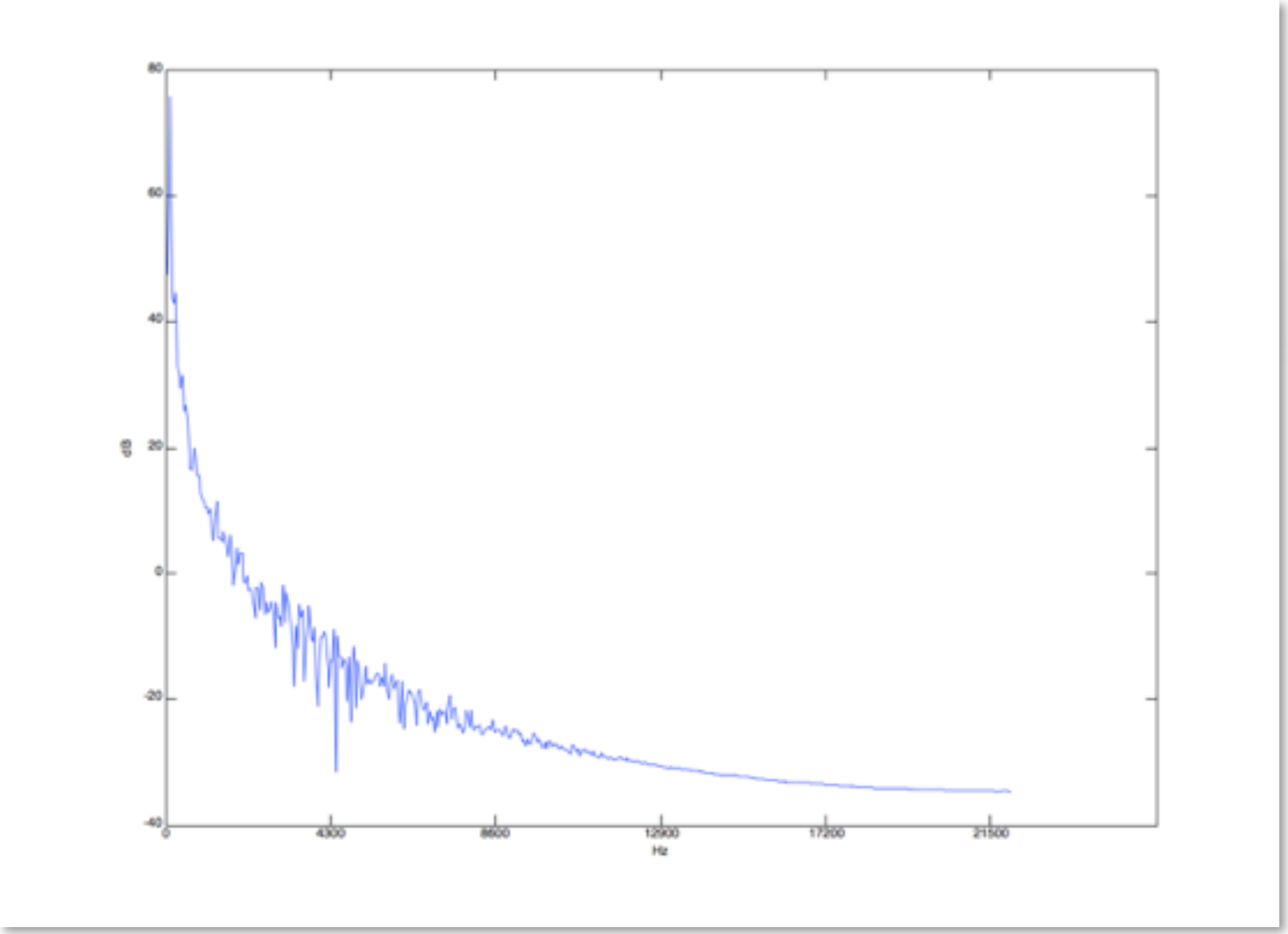
- Spectral Centroid
- Spectral Bandwidth/Spread
- Spectral Skewness
- Spectral Kurtosis
- Spectral Tilt/Slope
- Spectral Roll-Off
- Spectral Flatness Measure

Spectral moments

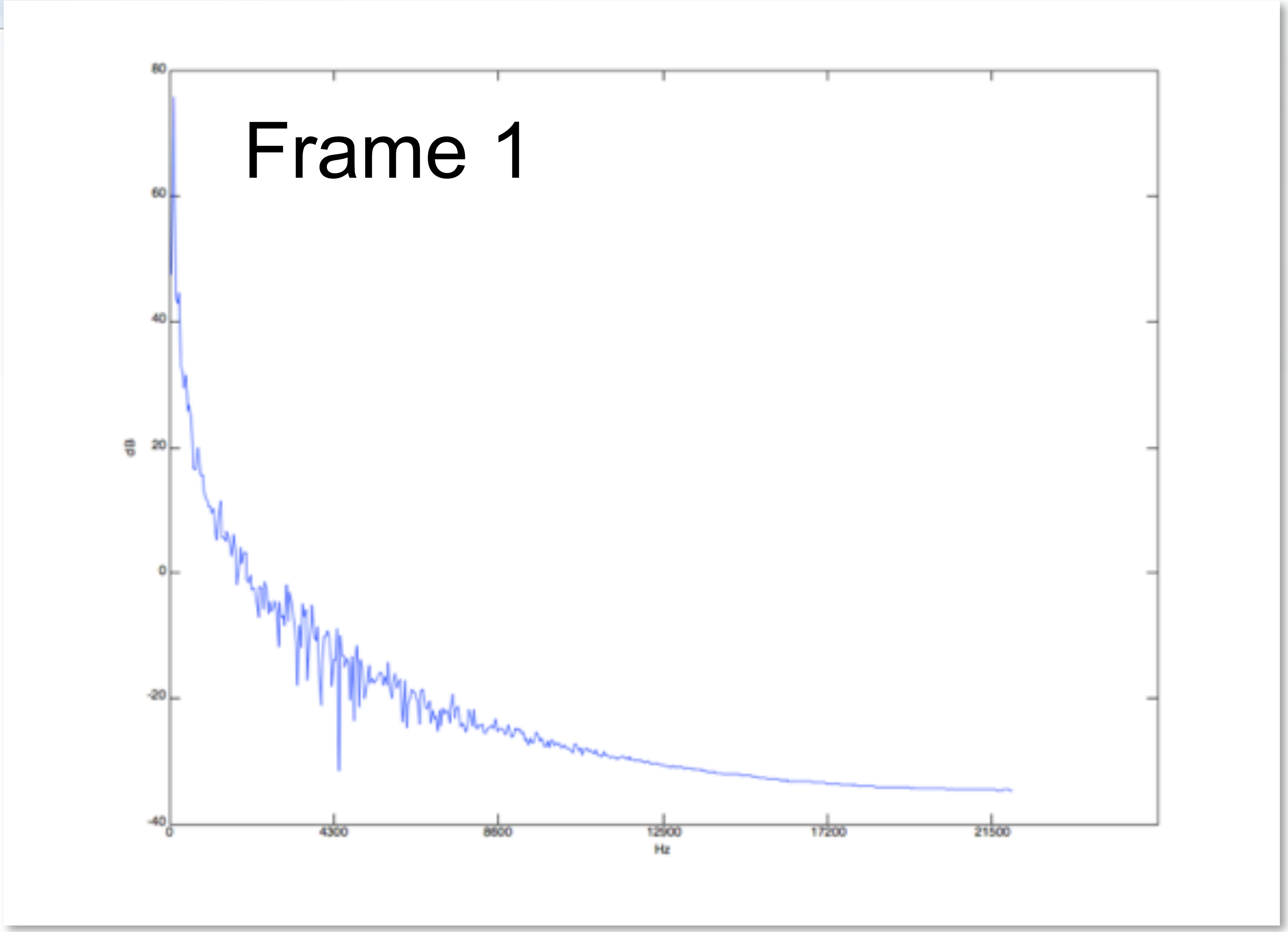
FFT of single window (aka “frame”)



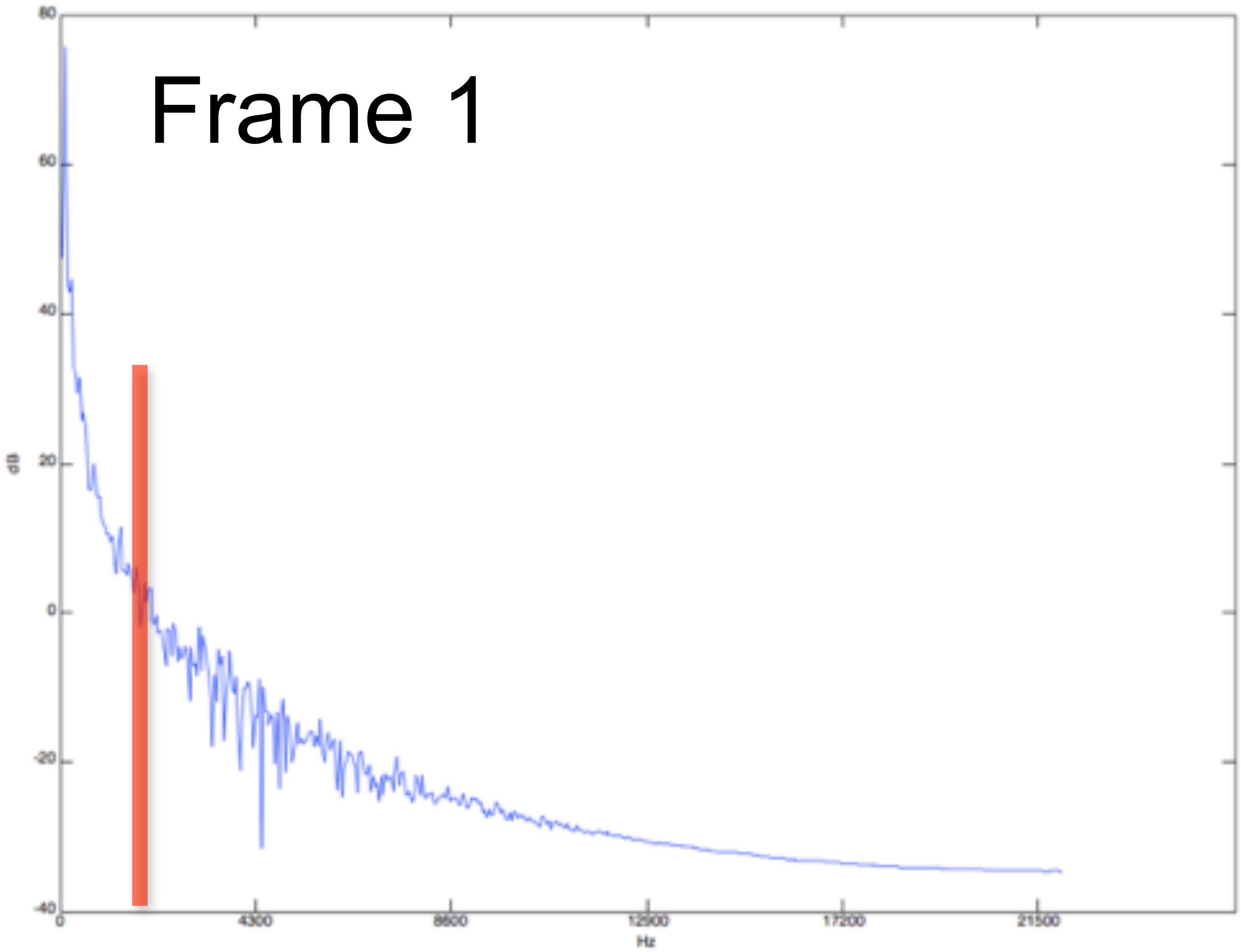
FFT of single window (aka “frame”)



Spectral Moments Demonstration

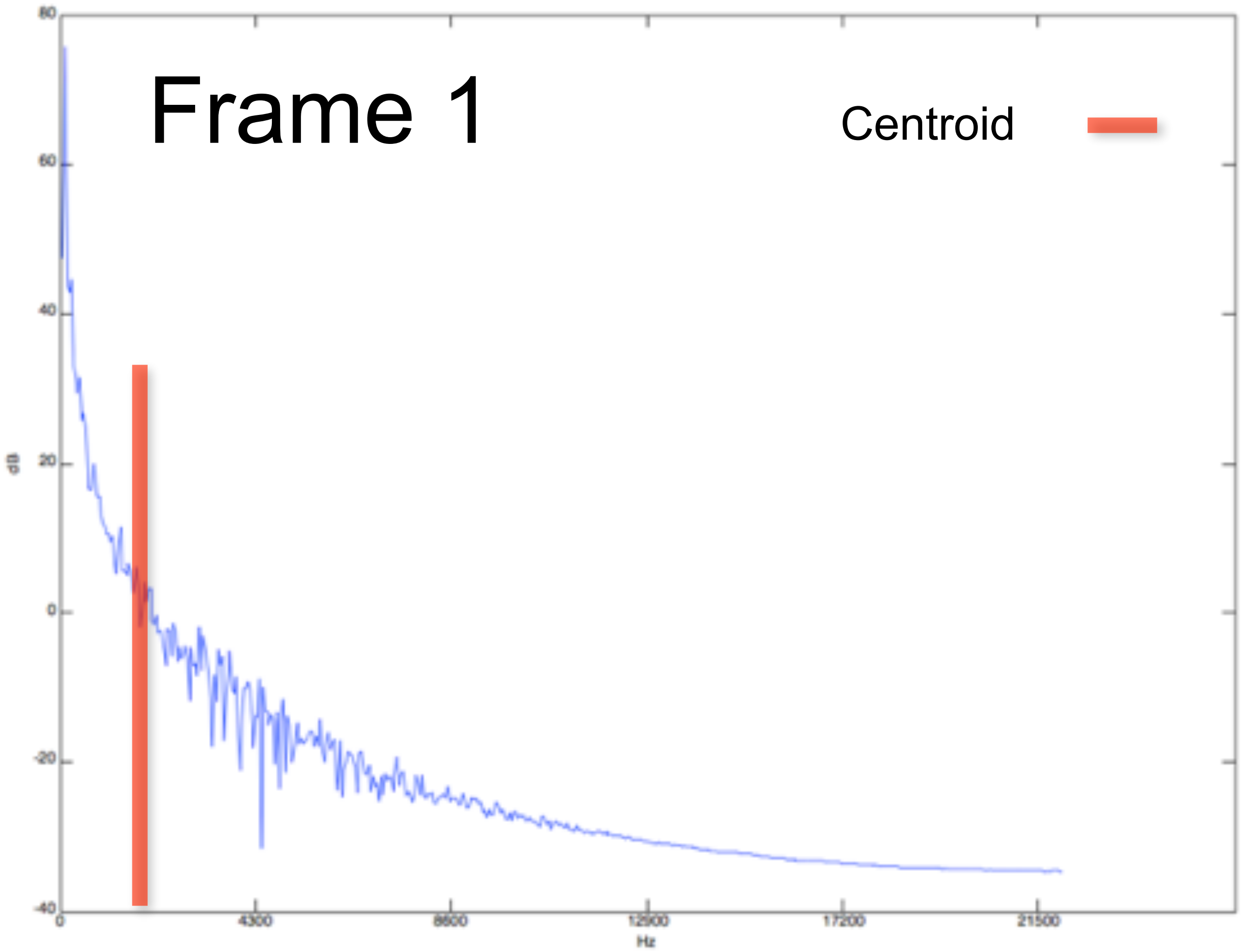


Spectral Moments Demonstration

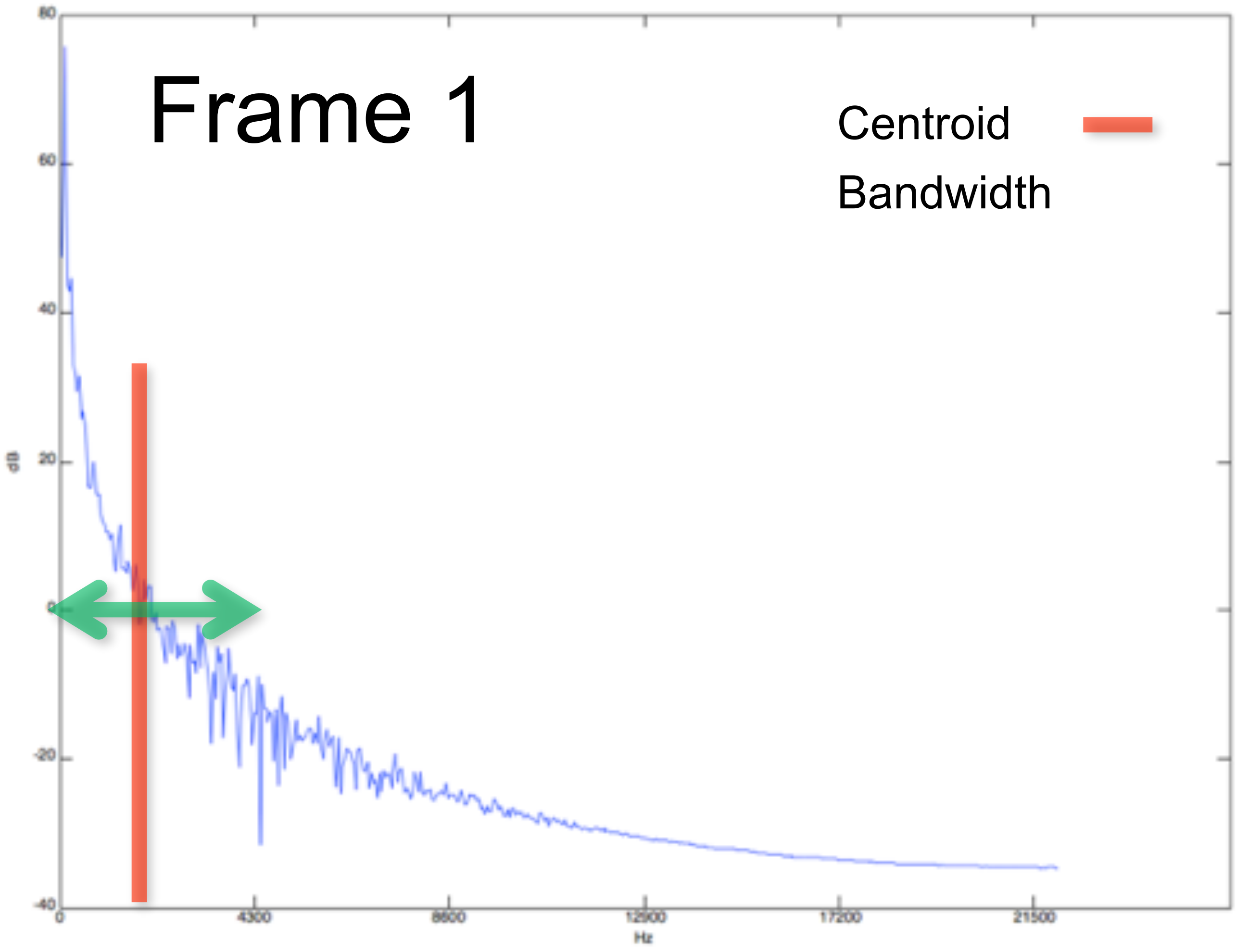


Frame 1

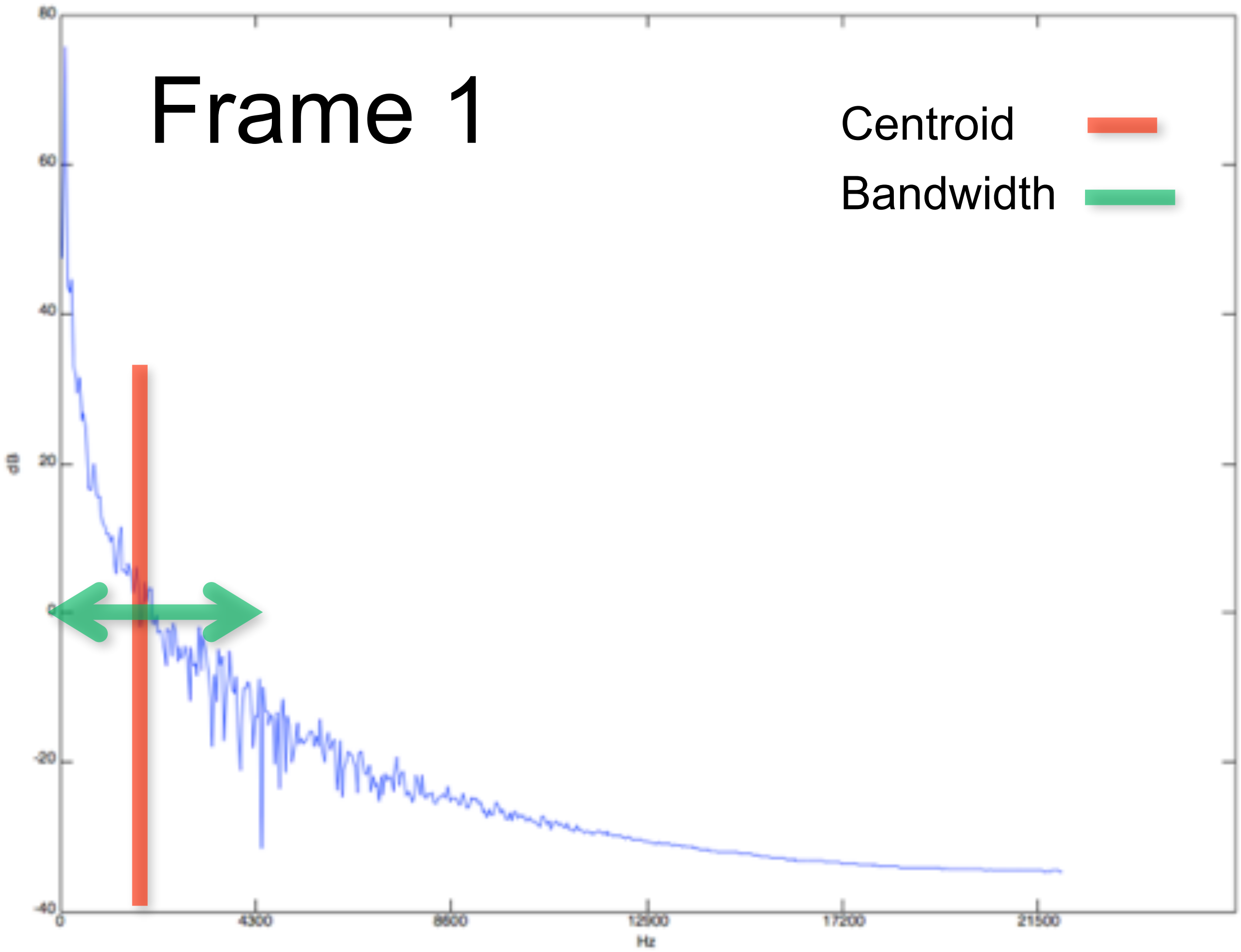
Spectral Moments Demonstration



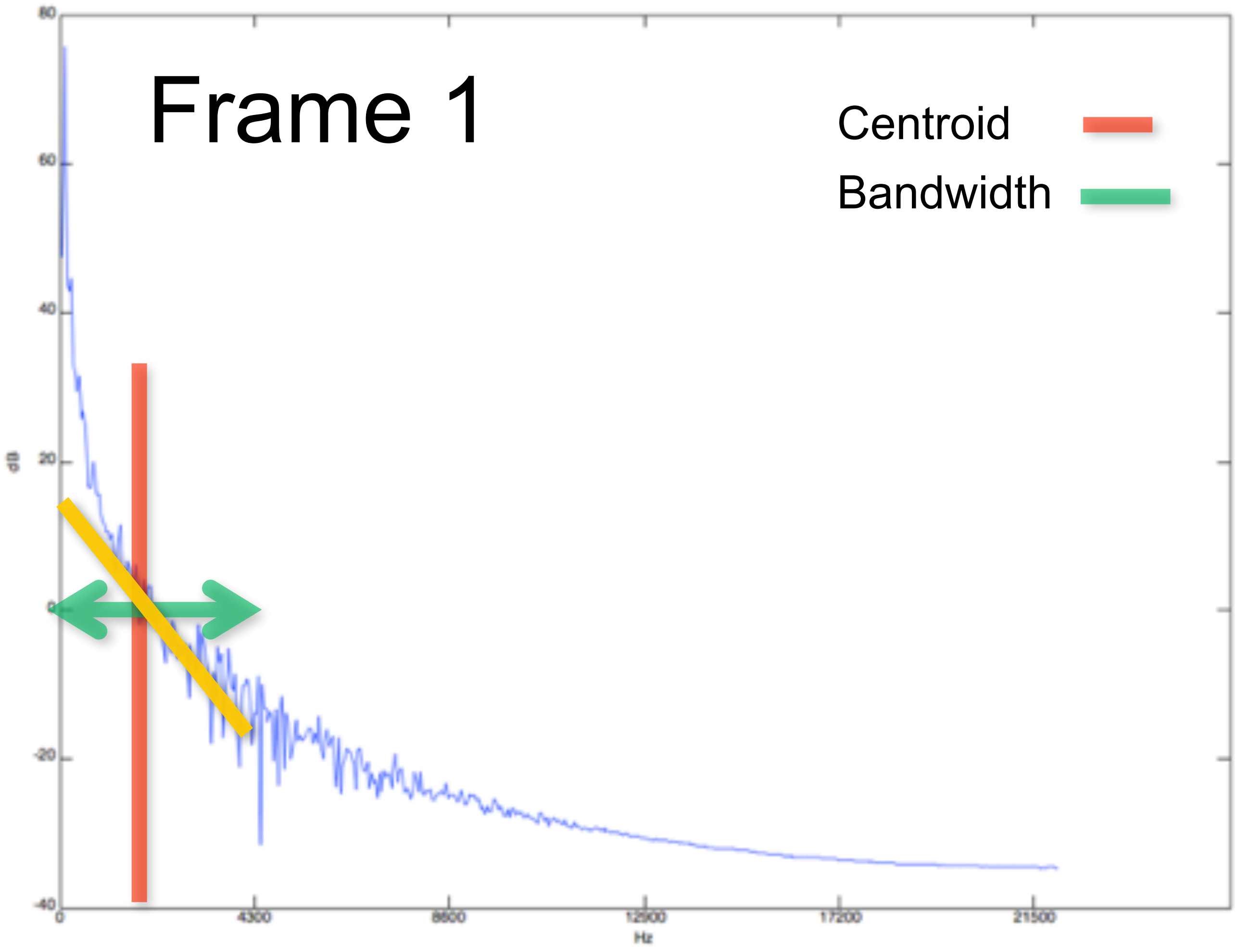
Spectral Moments Demonstration



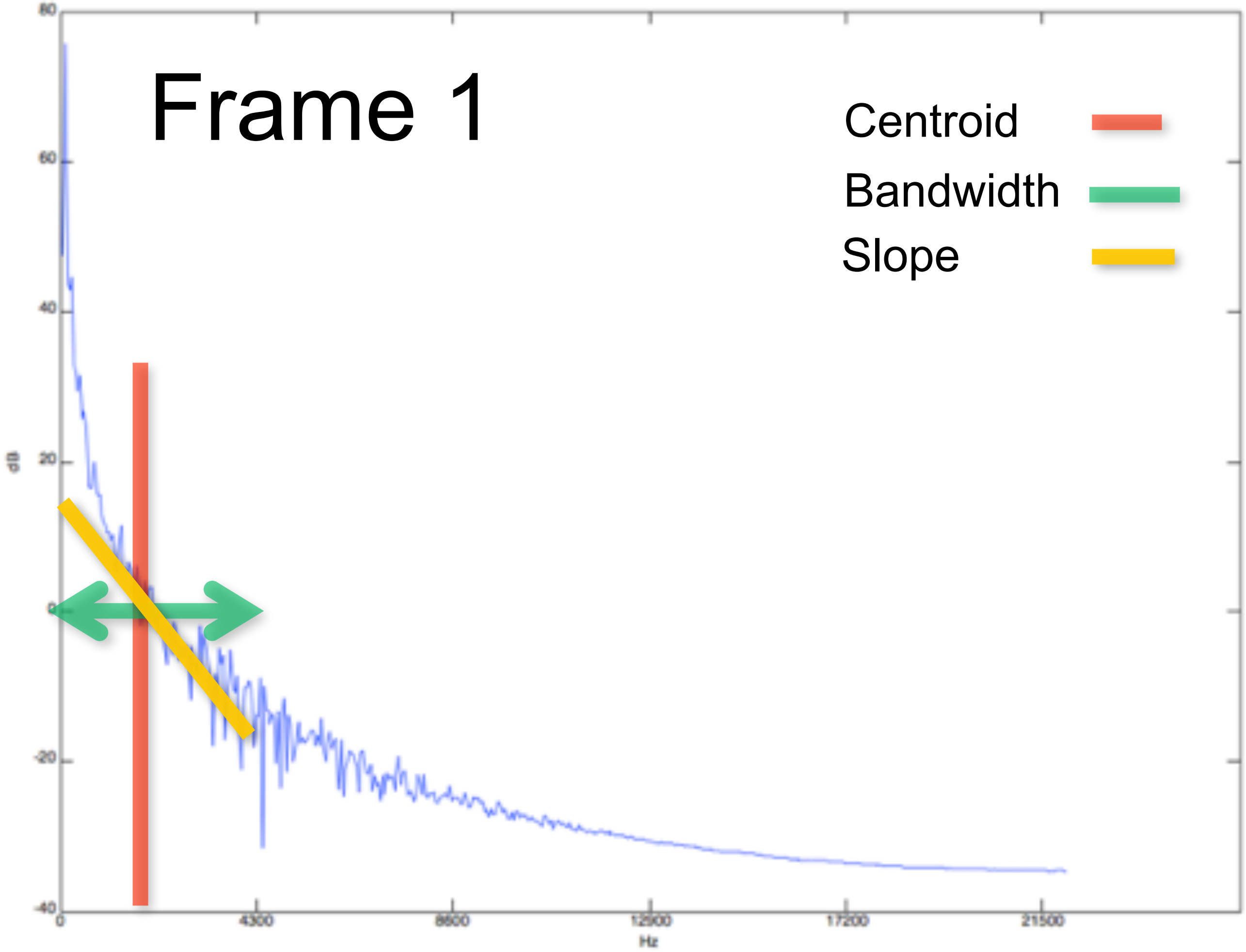
Spectral Moments Demonstration



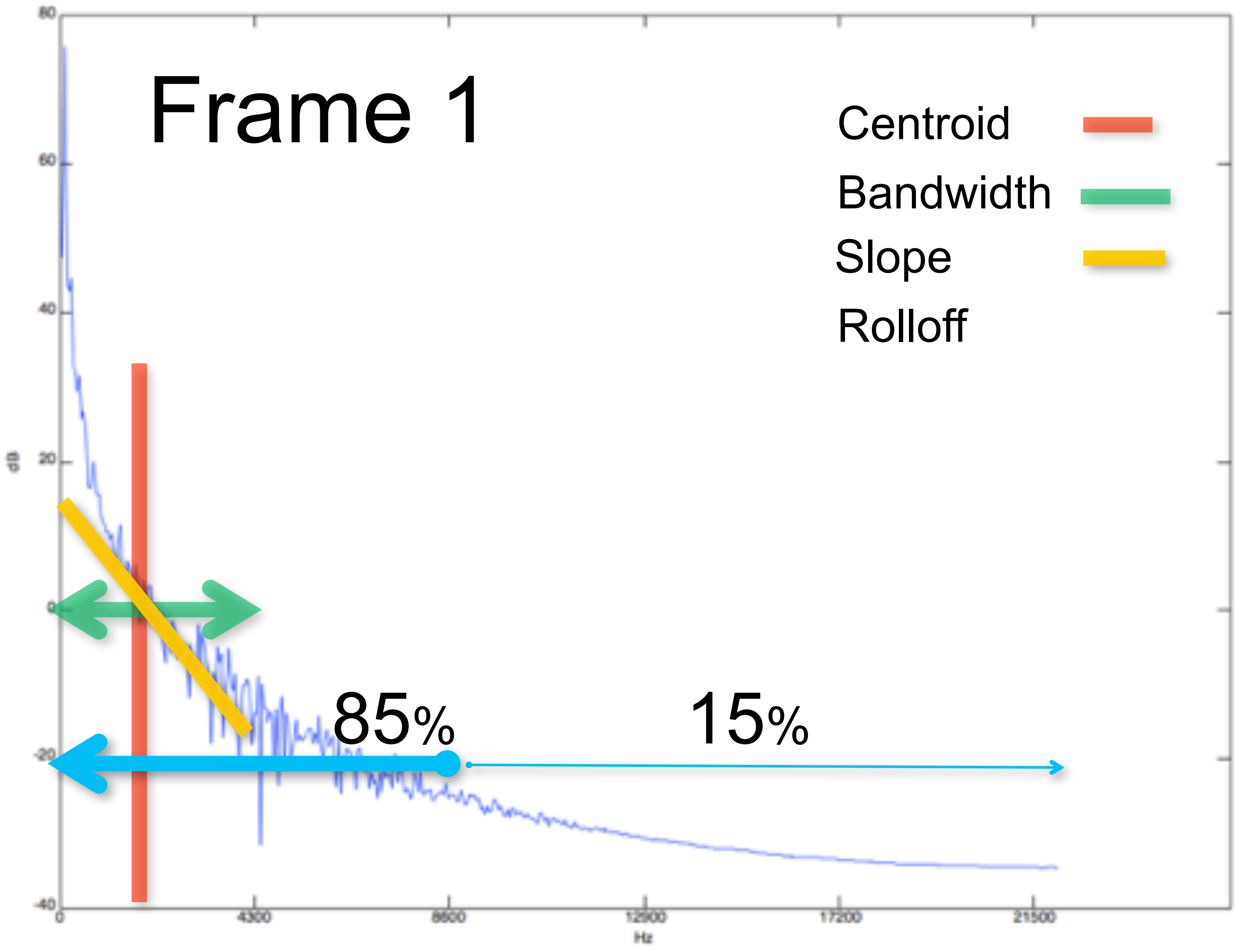
Spectral Moments Demonstration



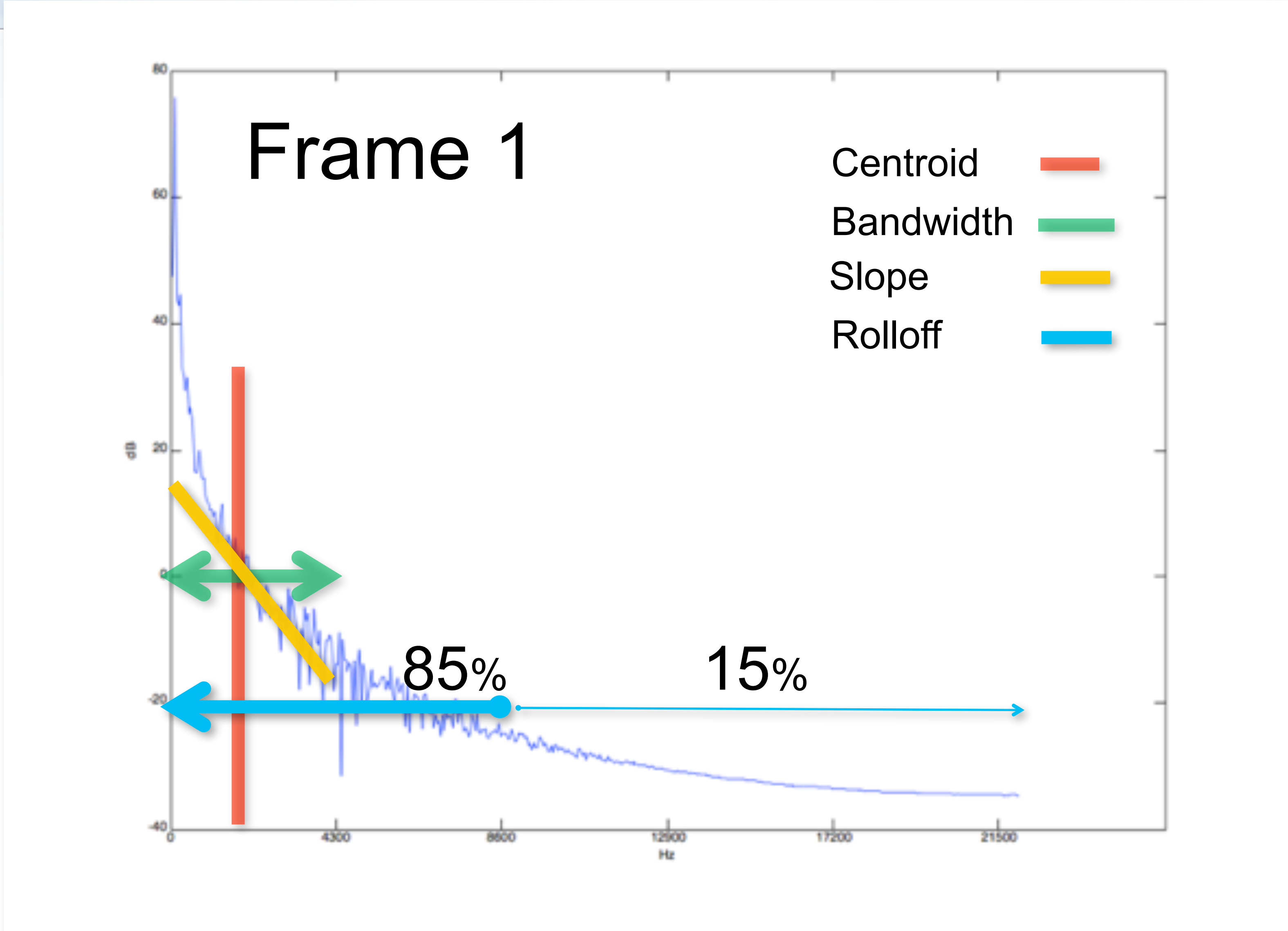
Spectral Moments Demonstration

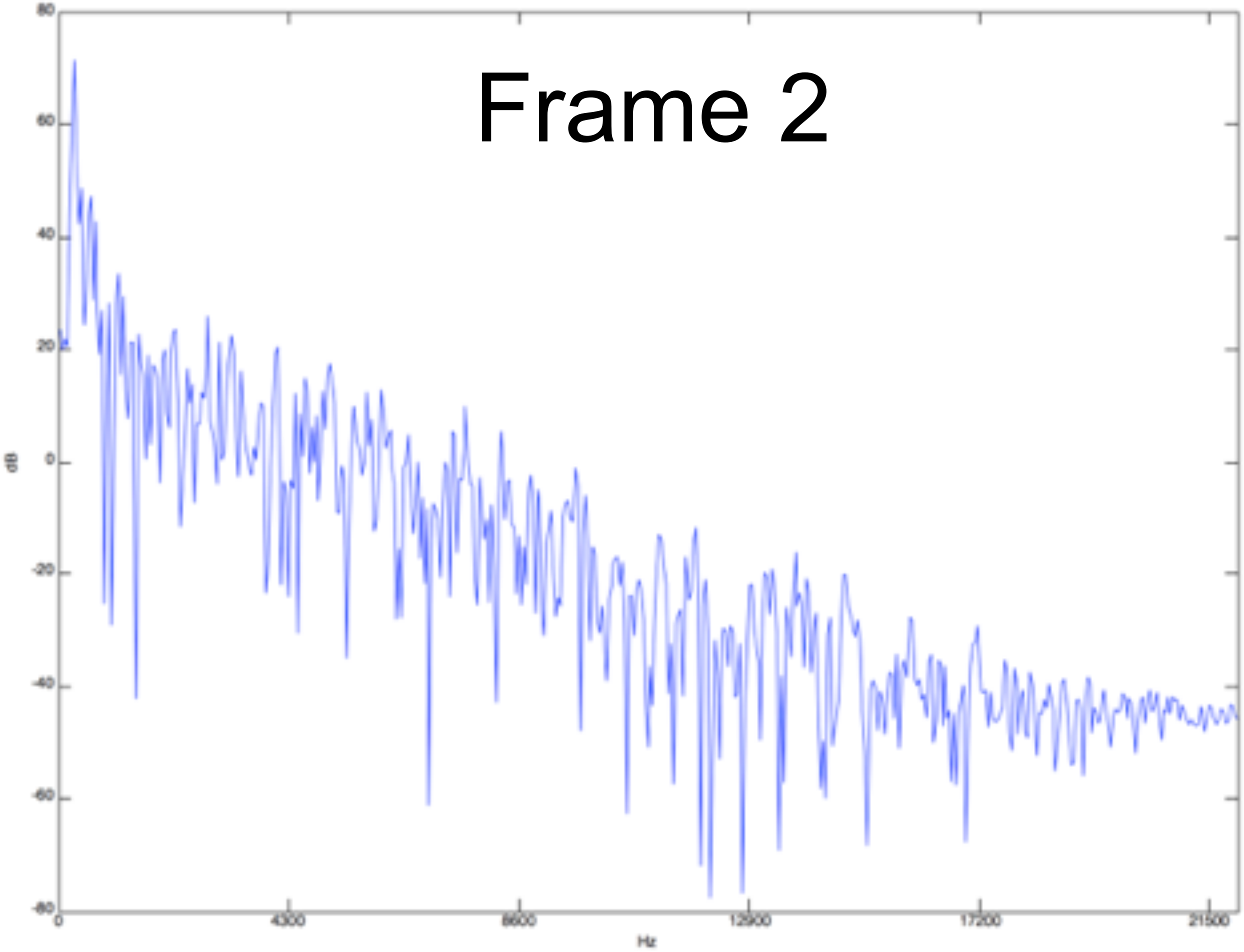


Spectral Moments Demonstration

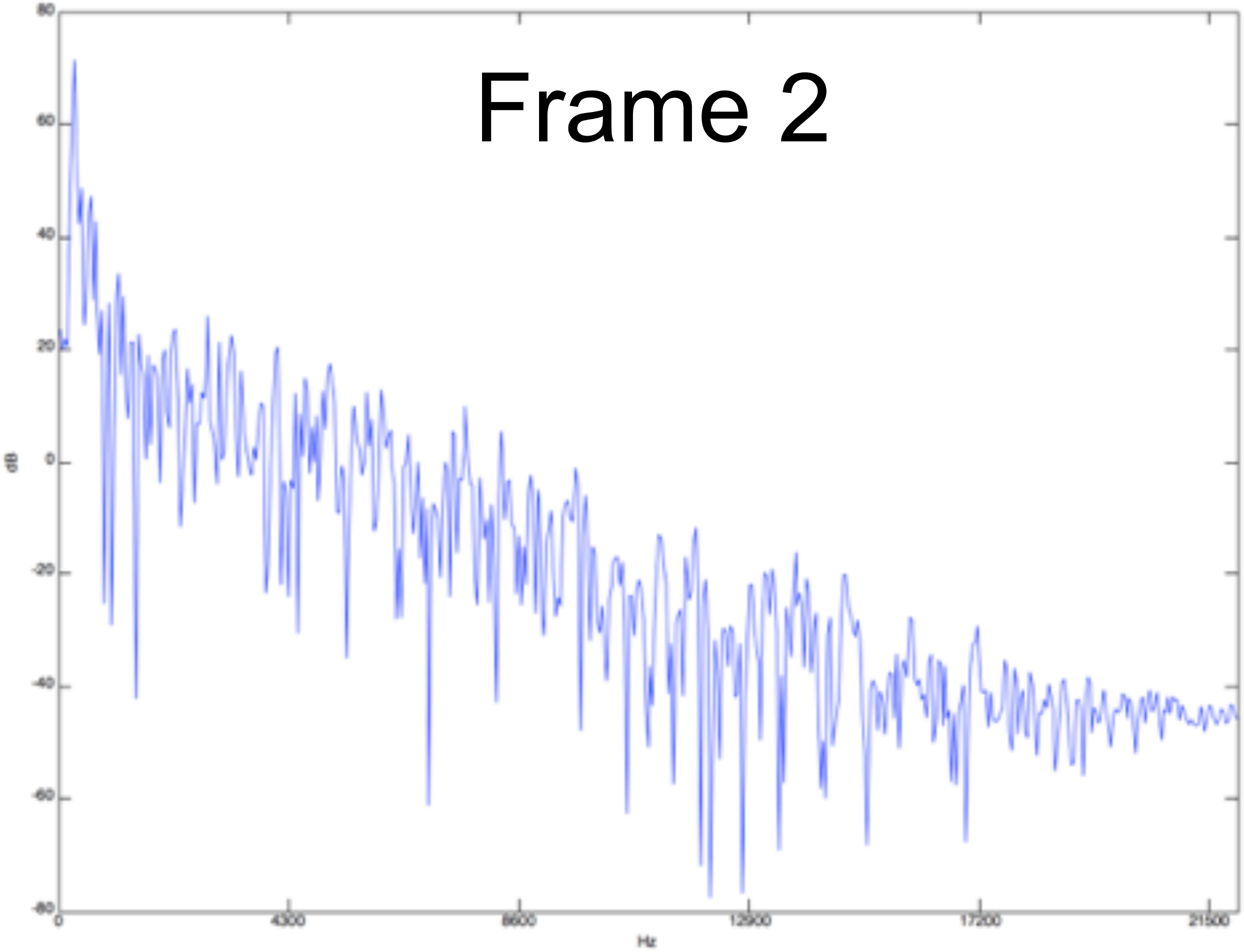


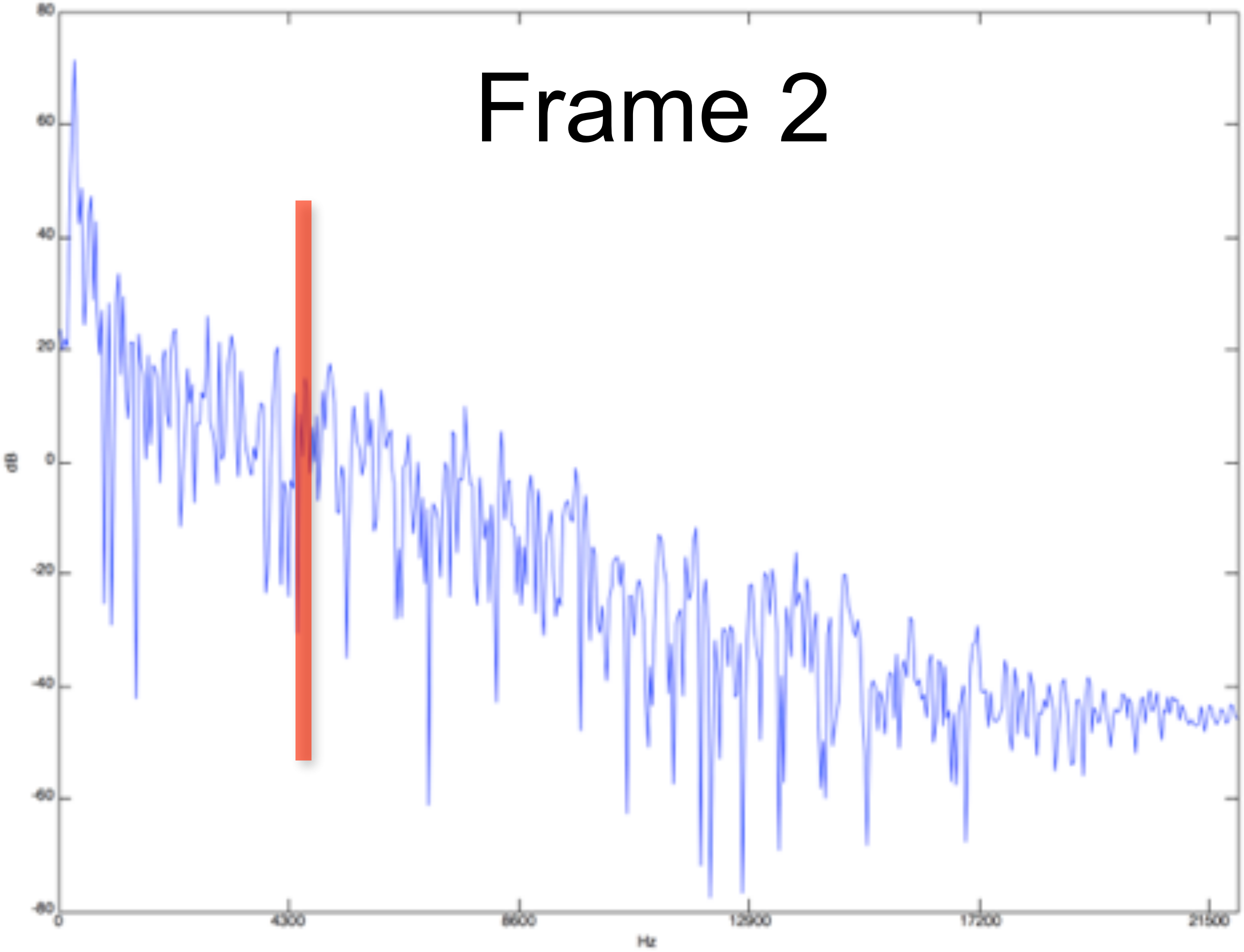
Spectral Moments Demonstration

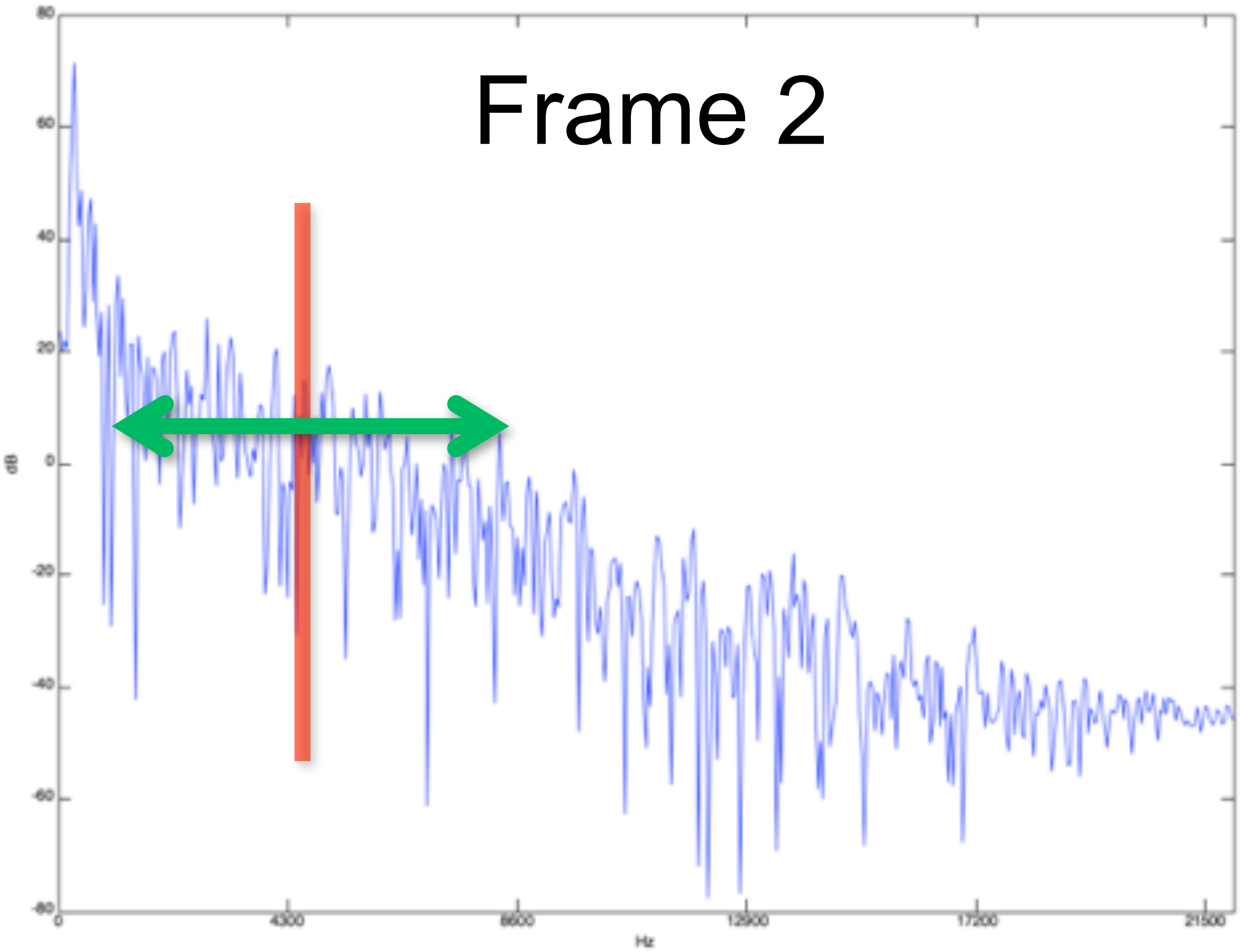


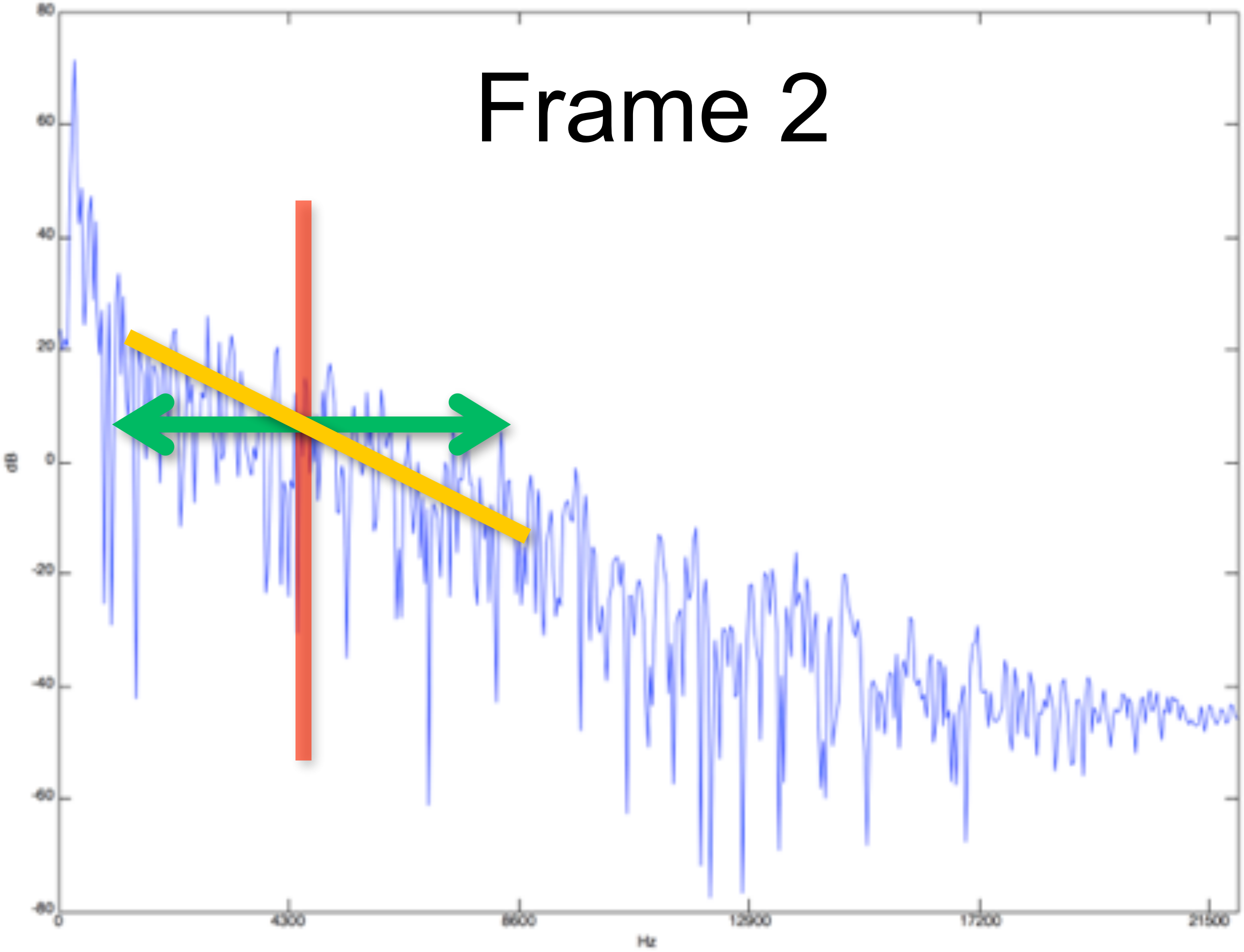

Frame 2
80
60
40
20
0
-20
-40
-60
-80
dB
0
4300
8600
12900
17200
21500
Hz

Frame 2




Frame 2
80
60
40
20
0
-20
-40
-60
-80
dB
0
4300
8600
12900
17200
21500
Hz


Frame 2
80
60
40
20
0
-20
-40
-60
-80
dB
0
4300
8600
12900
17200
21500
Hz


Frame 2
80
60
40
20
0
-20
-40
-60
-80
dB
0
4300
8600
12900
17200
21500
Hz

Spectral Moments

$$\tilde{X}(k) = \frac{|X(k)|}{\sum_k |X(k)|}$$

- 1st – Spectral Centroid
  - Dull vs. Brightness

$$SC = \frac{\sum_{k=0}^{N/2} f_k |X(k)|^2}{\sum_{k=0}^{N/2} |X(k)|^2} = \quad C_f = \frac{k|X(k)|}{\sum_k |X(k)|}$$

- 2nd – Bandwidth/Spread
  - Noisy vs. “peaky” (resonant)

$$S_f^2 = \sum_k (k - C_f)^2 \tilde{X}(k)$$

- 3rd – Skew
  - assymmetry of spectrum (high vs. low)

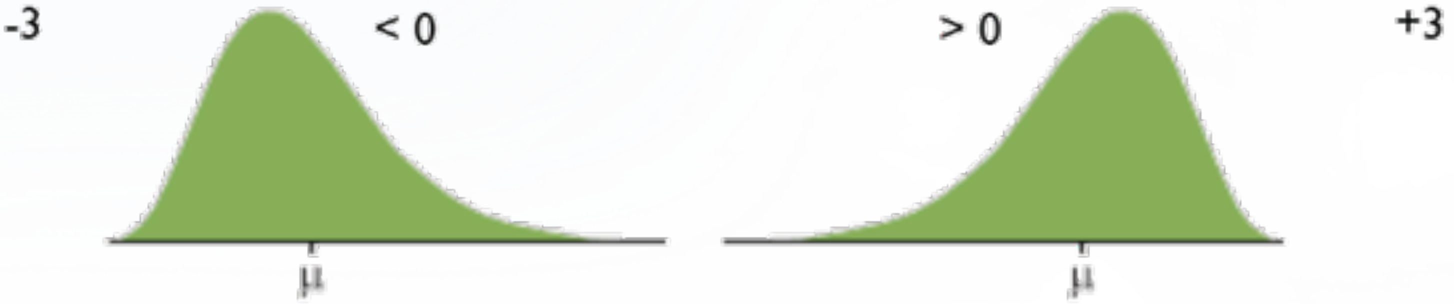
$$\gamma_1 = \frac{\sum_k (k - C_f)^3 \tilde{X}(k)}{S_f^3}$$

- 4th – Kurtosis
  - Equal spectral energy vs. narrow.

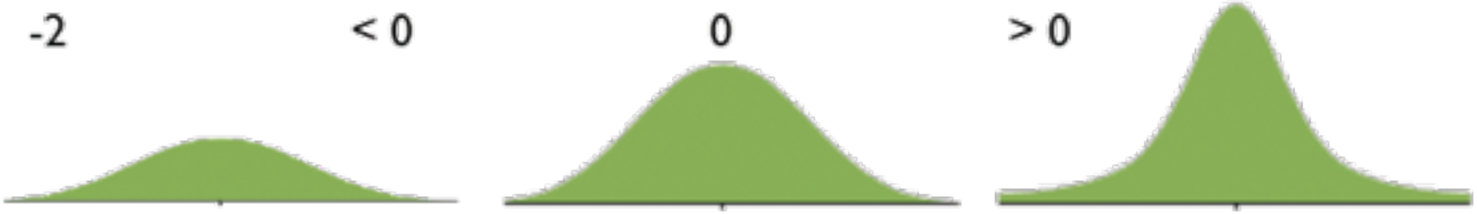
$$\gamma_2 = \frac{\sum_k (k - C_f)^4 \tilde{X}(k)}{S_f^4}$$

Higher Spectral Moments

- Skewness



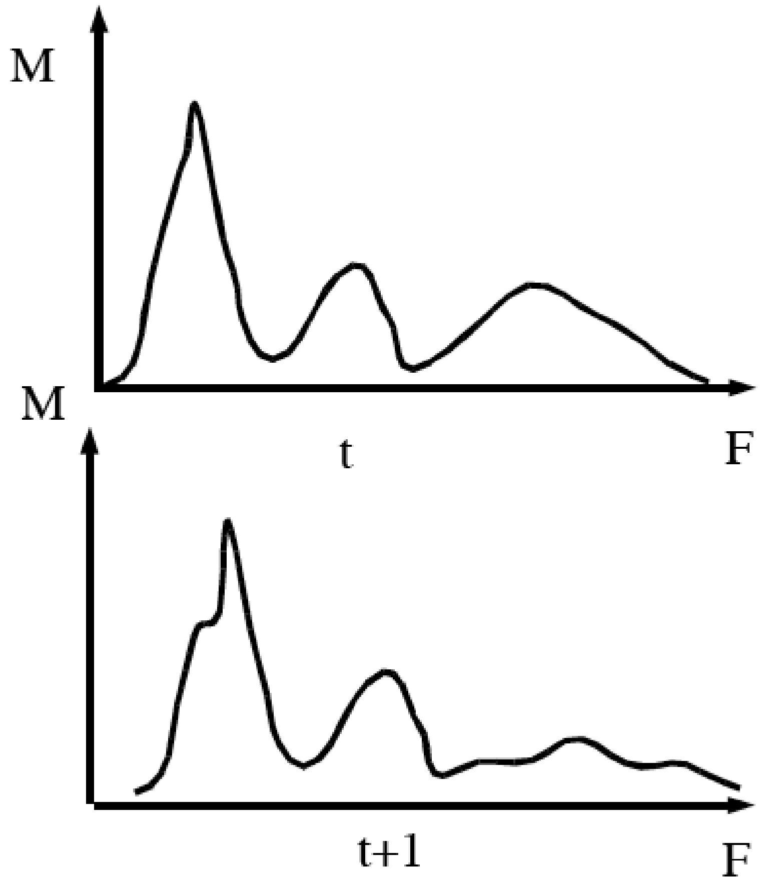
- Kurtosis



http://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox/userguide1.1

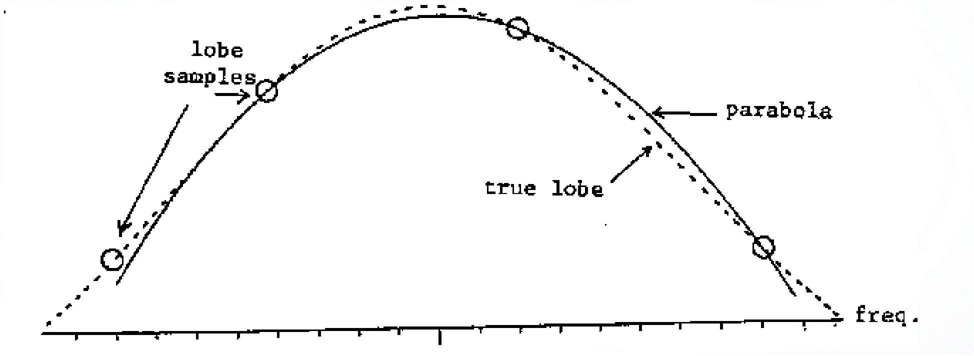
Spectra as Time-varying

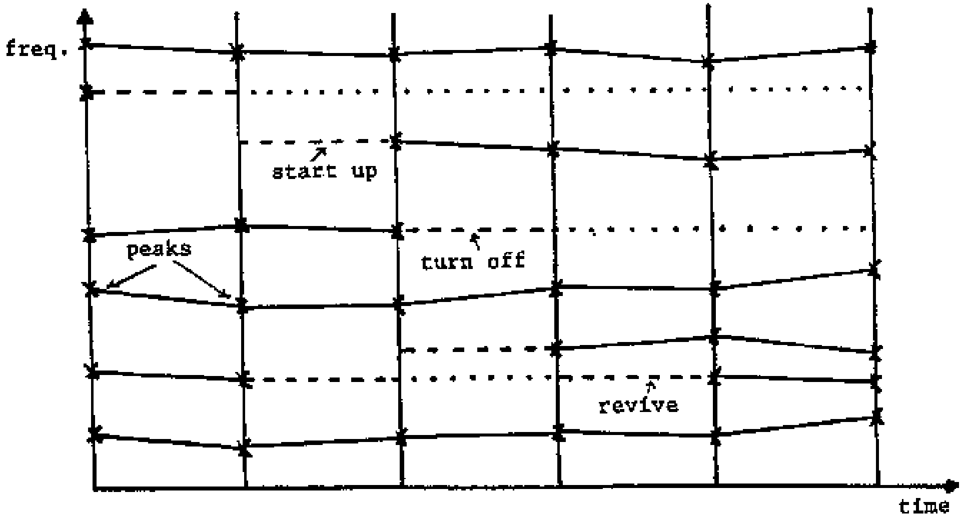
- Track peaks/regions between frames (requires thresholds of change)
- Model the dynamicity (e.g., formant trajectory, vibrato extraction)



Spectral Peaks and Tracking

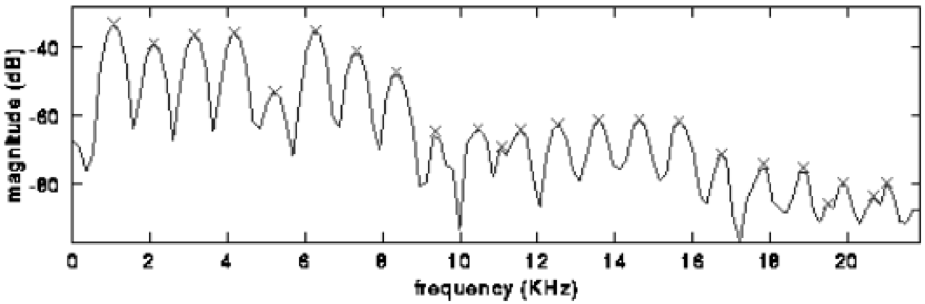
- Peak finding
  - (via autocorrelation)
- Peak discrimination
- Peak continuation:
  - tracks and guides
- Derived statistics
- Problem cases

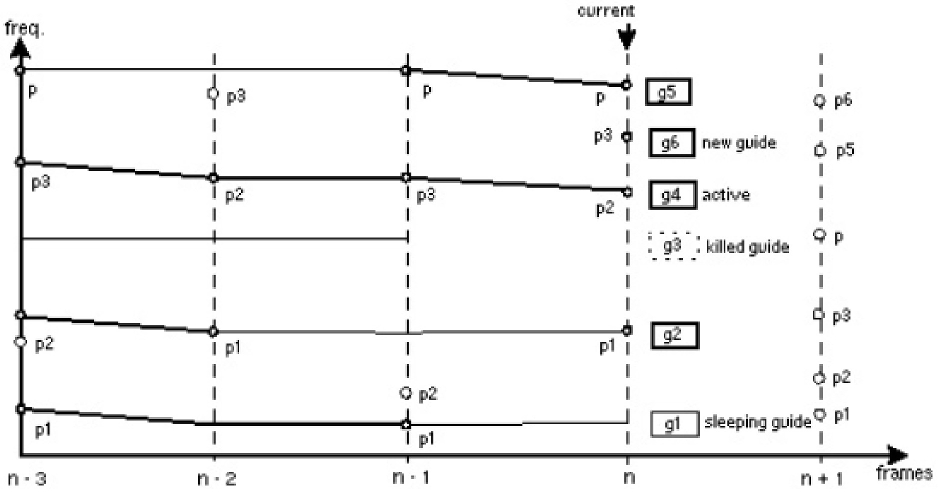




Peaks and Tracks

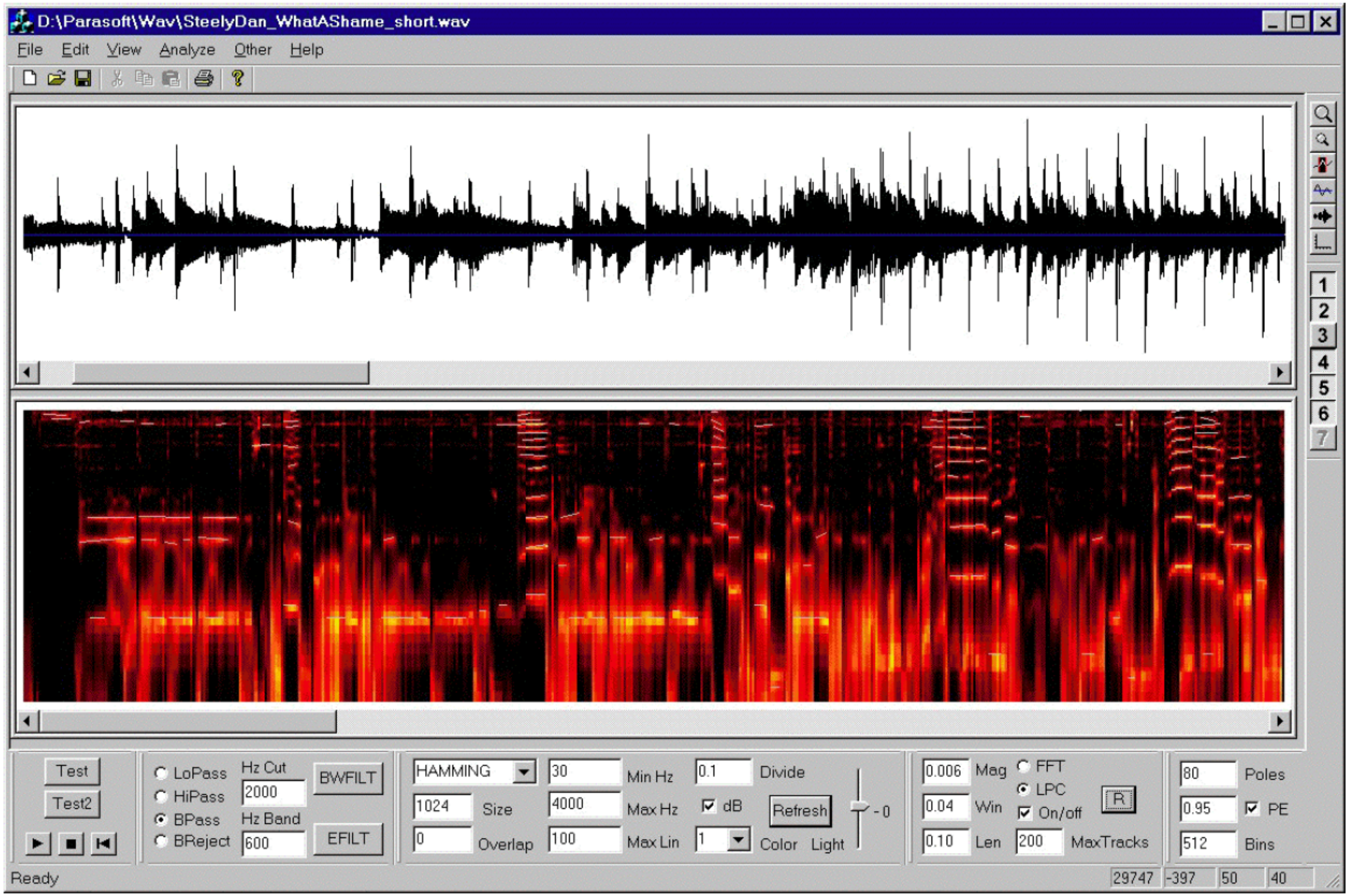
- Peak–finding
  - Thresholds, distances, heuristics
- Peak–continuation
  - Inter–frame distances and guides
  - Dropped frames and stretching
  - Track birth/death criteria





Spectral Peak-Tracking Example

Courtesy Stephen T. Pope



D:\Parasoft\Wav\SteelyDan_WhatAShame_short.wav

○ LoPass
○ HiPass
● BPass
○ BReject

Hz Cut: 2000
Hz Band: 600

HAMMING
Size: 1024
Overlap: 0

Min Hz: 30
Max Hz: 4000
Max Lin: 100

Divide: 0.1
☑ dB
1 Color Light
-0

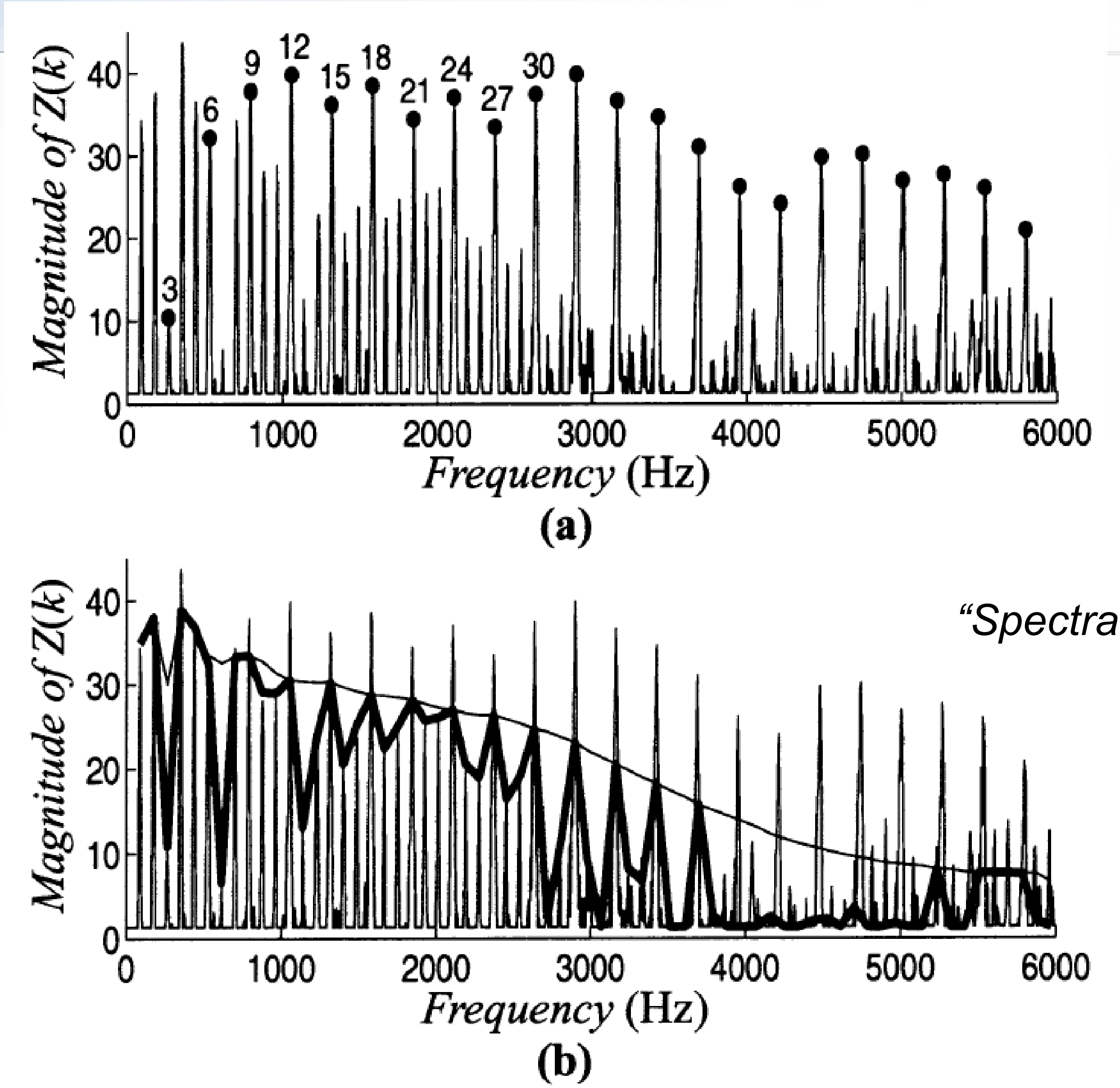
Mag: 0.006
Win: 0.04
Len: 0.10

○ FFT
● LPC
☑ On/off
MaxTracks: 200

Poles: 80
☑ PE: 0.95
Bins: 512

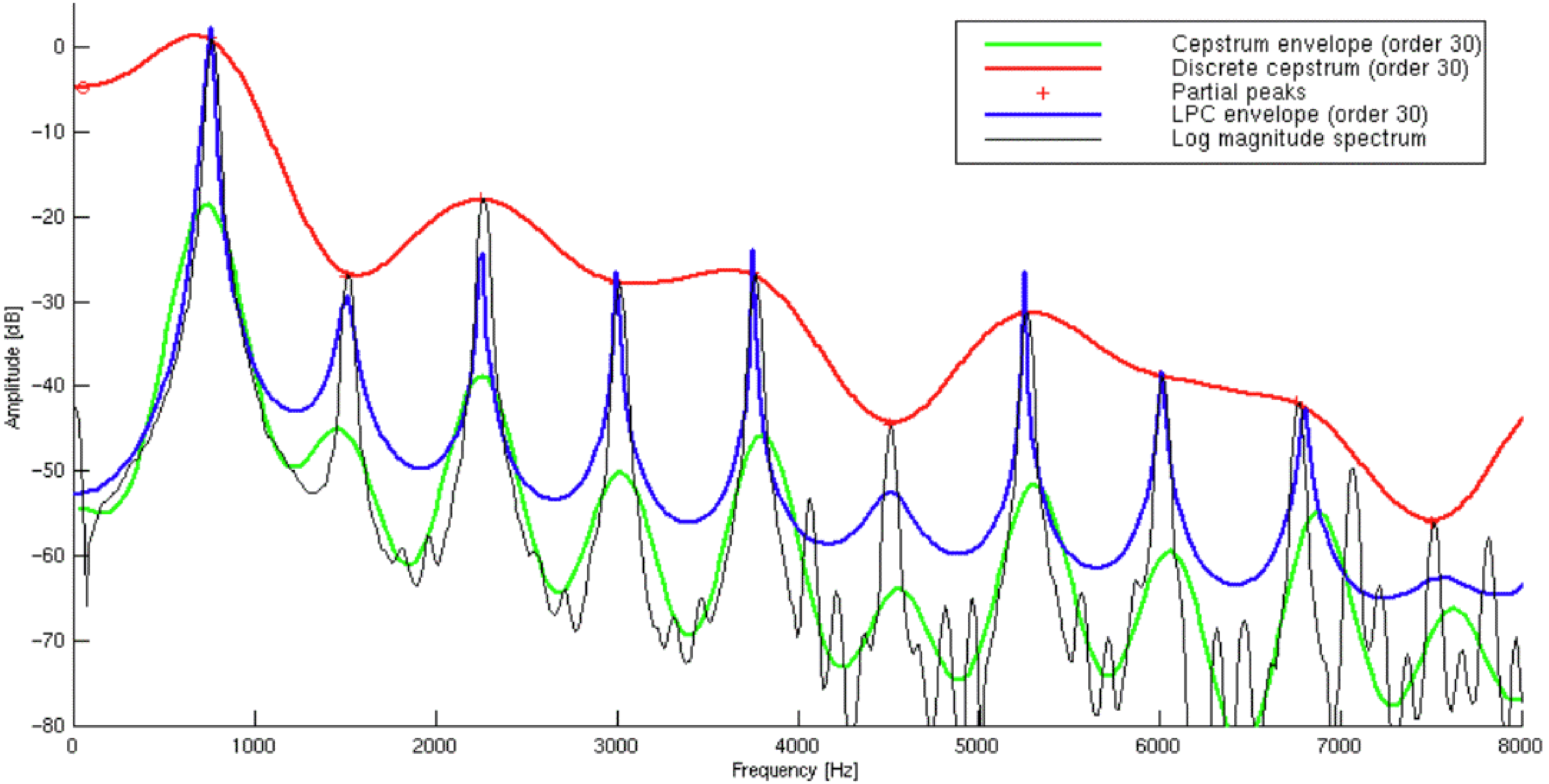
Ready 29747 -397 50 40

Spectral Smoothness Measure



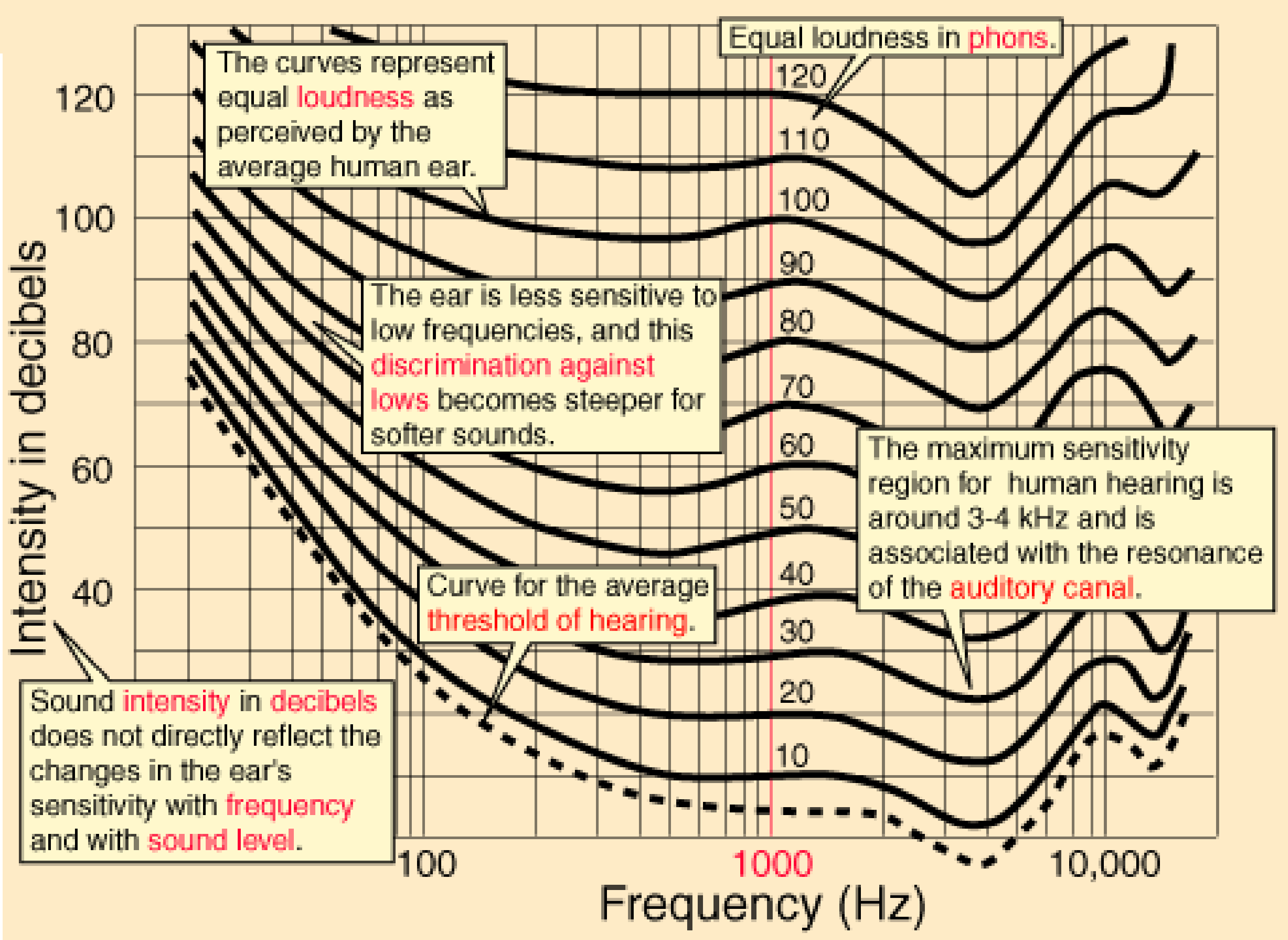
“Spectral Envelope”

Smoothed Spectrum Types



Equal-loudness Curves

- Fletcher-Munson vs. Robinson-Dadson



Equal loudness in phons.

The curves represent equal loudness as perceived by the average human ear.

The ear is less sensitive to low frequencies, and this discrimination against lows becomes steeper for softer sounds.

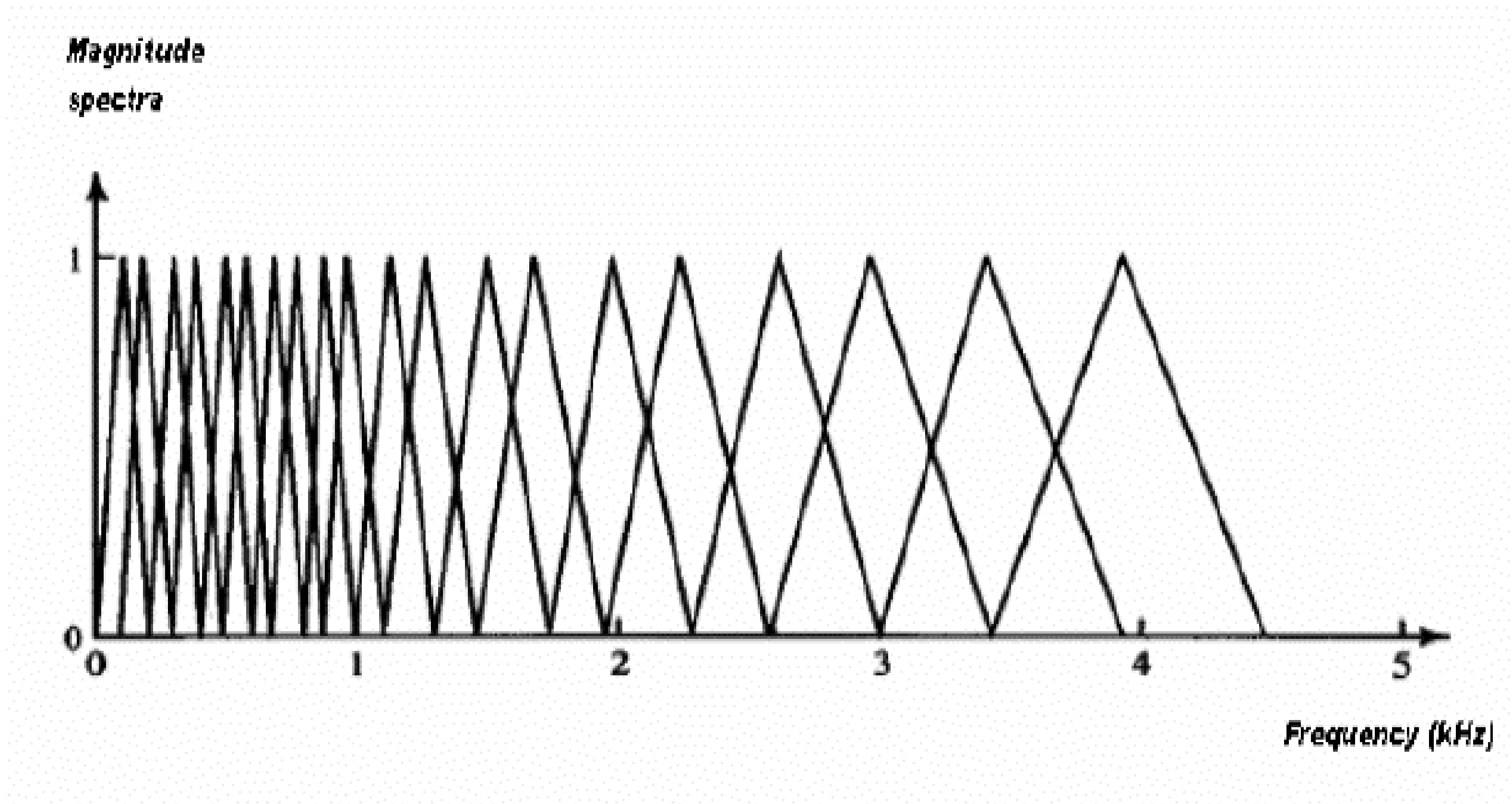
The maximum sensitivity region for human hearing is around 3-4 kHz and is associated with the resonance of the auditory canal.

Curve for the average threshold of hearing.

Sound intensity in decibels does not directly reflect the changes in the ear's sensitivity with frequency and with sound level.

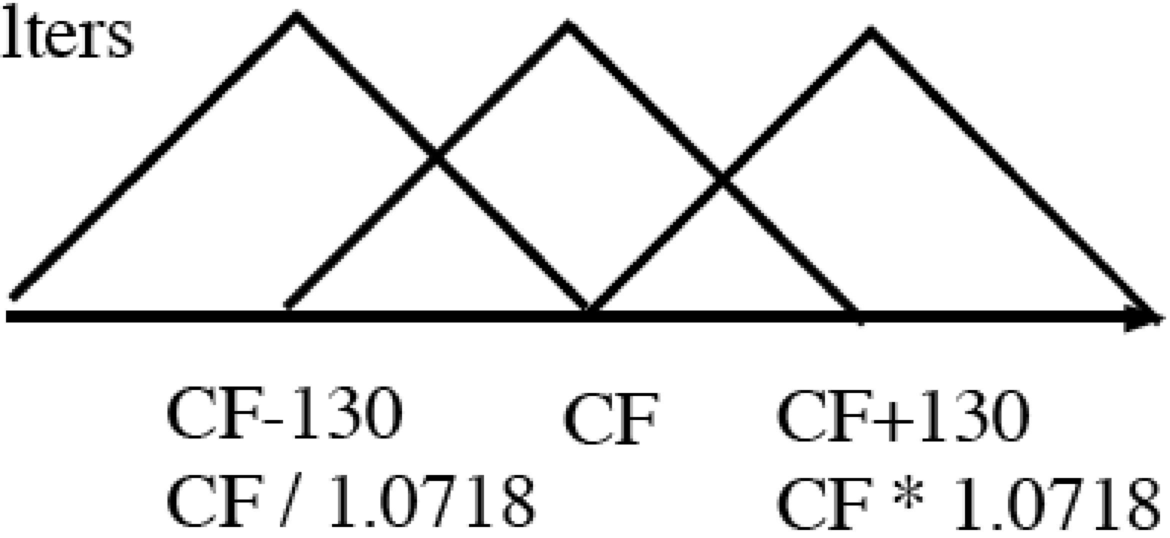
Frequency Regions and Scaling

- Mel-warped frequency bands



Mel Scale and Coefficients

Mel-scale
13 linearly-spaced filters
27 log-spaced filters



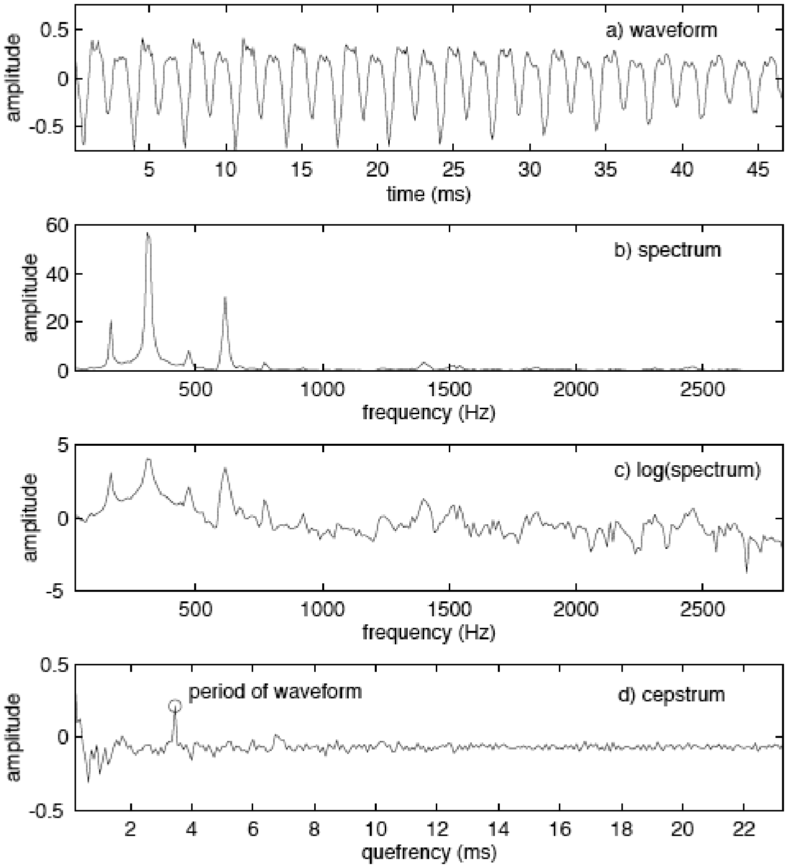
Mel-Freq Cepstral Coefficients

- Steps:
  - Signal
  - FT
  - Log magnitude
  - Phase unwrapping
  - FT (or DCT)
- Interpretations
  - “ceps” = “Spec”trum of spectrum
  - “Quefrency”
  - Mel-scale filters

- Represents spectrum in a small set of coeff’s.
- Instead of AC, use FFT or DCT of PDS
- Leads to interesting statistics (1st deriv=DeltaMFCC, variance) of higher-level spectral properties, see next section

MFCC Analysis

- Analogy
  - Start with log spectrum of mixed complex tones: several sets of related partial peaks
  - Take, e.g., the autocorr. of the FFT PDS
  - Warped frequencies of peaks correspond to fundamental frequencies of overtone series



Comparison With LPC (by Andrianakis & White)

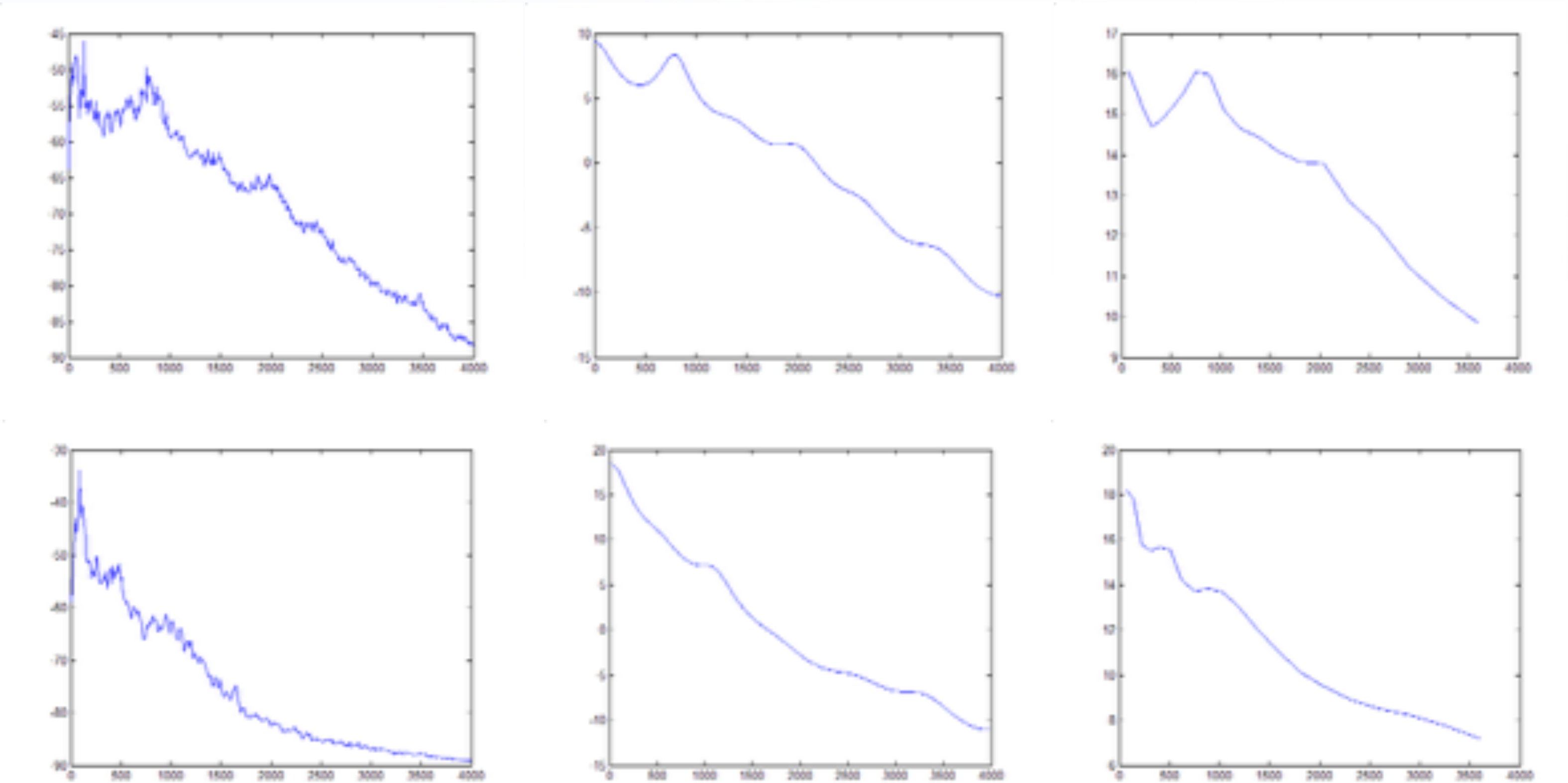
FFT PSD

13-pole LPC Spectrum

20 Mel Spectrum

Train

Car



Frequency [Hz]

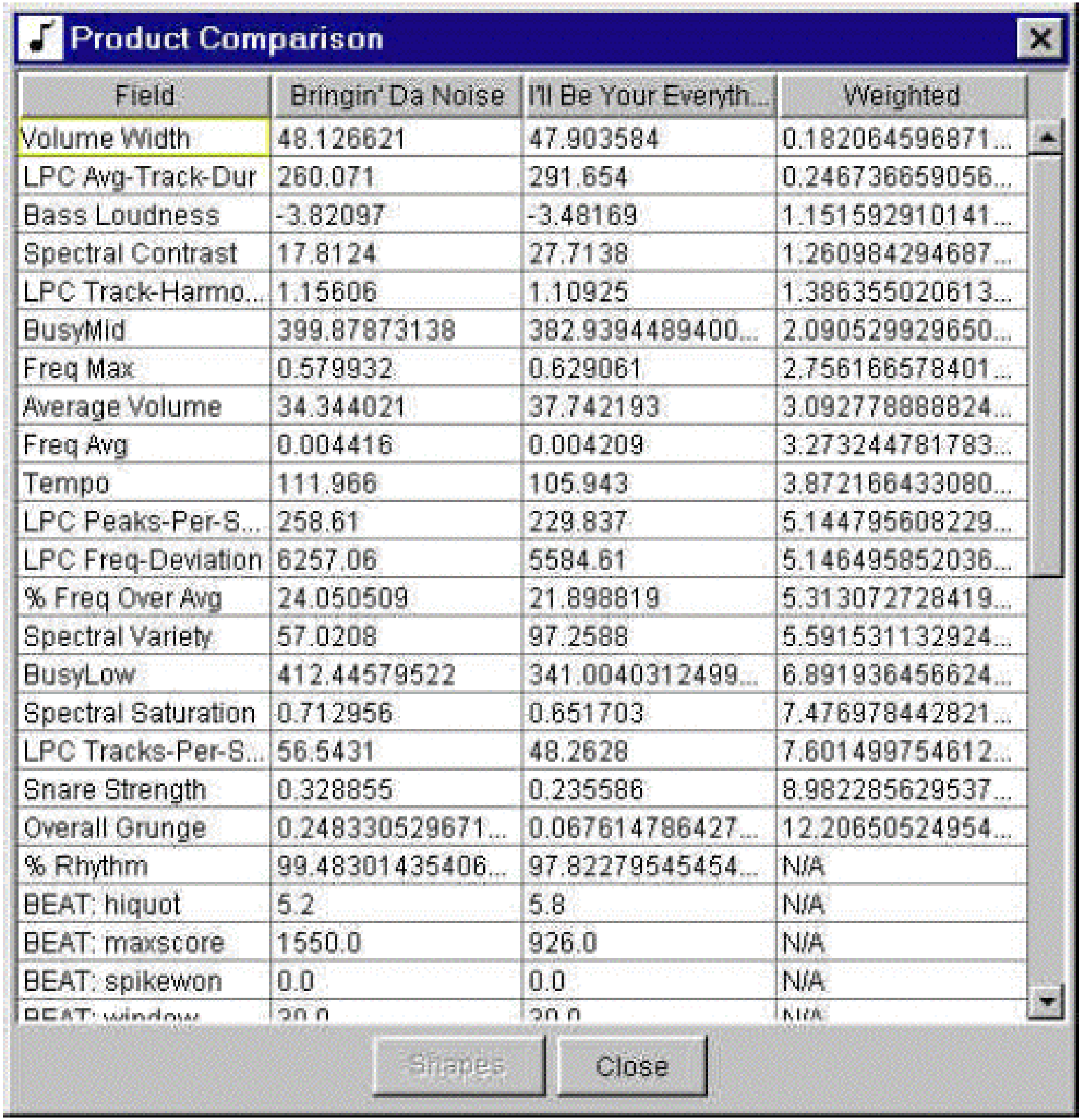
Spatial-domain Features

- M/S Encoding (stereo sum & difference)
- Surround-sound processing
  - L/R vs C
  - L/R vs Ls/Rs
- Frequency-dependent spatial separation
- Higher-dimensional sources
- Stem tracks

Other Feature Domains

- Other time-domain features
  - Beats, beat histograms (tomorrow)
- Other frequency-domain features
  - Fluctuation patterns
- Other time-frequency transforms
  - Filter banks
- Wavelets
  - Trades off temporal & spectral resolution
- Linear Predictive Coding
  - polynomial representation

Feature Vector Examples



Field	Bringin' Da Noise	I'll Be Your Everyth...	Weighted
Volume Width	48.126621	47.903584	0.182064596871...
LPC Avg-Track-Dur	260.071	291.654	0.246736659056...
Bass Loudness	-3.82097	-3.48169	1.151592910141...
Spectral Contrast	17.8124	27.7138	1.260984294687...
LPC Track-Harmo...	1.15606	1.10925	1.386355020613...
BusyMid	399.87873138	382.9394489400...	2.090529929650...
Freq Max	0.579932	0.629061	2.756166578401...
Average Volume	34.344021	37.742193	3.092778888824...
Freq Avg	0.004416	0.004209	3.273244781783...
Tempo	111.966	105.943	3.872166433080...
LPC Peaks-Per-S...	258.61	229.837	5.144795608229...
LPC Freq-Deviation	6257.06	5584.61	5.146495852036...
% Freq Over Avg	24.050509	21.898819	5.313072728419...
Spectral Variety	57.0208	97.2588	5.591531132924...
BusyLow	412.44579522	341.0040312499...	6.891936456624...
Spectral Saturation	0.712956	0.651703	7.476978442821...
LPC Tracks-Per-S...	56.5431	48.2628	7.601499754612...
Snare Strength	0.328855	0.235586	8.982285629537...
Overall Grunge	0.248330529671...	0.067614786427...	12.20650524954...
% Rhythm	99.48301435406...	97.82279545454...	N/A
BEAT: hiquot	5.2	5.8	N/A
BEAT: maxscore	1550.0	926.0	N/A
BEAT: spikewon	0.0	0.0	N/A
BEAT: window	20.0	20.0	N/A

Example: FMAK3 Feature Table

```
class FeatureTable {                    // FeatureTable is a root object (no parents)

public:                                 // Data members (instance variables)
          float mTimeStamp;             // When do I start?
          float mTimeDur;               // How long a time-span do I represent?
                                        // Time-domain features
          unsigned int mRMSWindowSize;  // Size of RMS window
          FeatureDatum mRMS;            // Rectangular-windowed RMS amplitude
          FeatureDatum mPeak;           // Max sample amplitude
          FeatureDatum mLPRMS;          // RMS amplitude of LP-filtered signal
          FeatureDatum mHPRMS;          // RMS amplitude of HP-filtered signal
          size_t mZeroCrossings;        // Count of zero crossings
          FeatureDatum mDynamicRange;   // RMS dynamic range of sub-windows
          FeatureDatum mPeakIndex;      // RMS peak sub-window index
          FeatureDatum mTempo;          // RMS/FWT instantaneous tempo estimate
          FeatureDatum mTimeSignature;  // Time signature guess
          FeatureDatum mBassPitch;      // Bass pitch guess in Hz
          unsigned int mBassNote;       // Bass note (MIDI key number) guess
          FeatureDatum mBassDynamicity; // Bass note dynamicity (size of histogram)
                                        // Spatial features
          FeatureDatum mStereoWidth;    // L/R difference
          FeatureDatum mSurroundDepth;  // Front/Surround difference
          FeatureDatum mCenterDistinction; // Center vs. L/R sum difference
```

Example: FMAK3 Feature Table, cont’d

```
                              // Frequency-domain features
unsigned int mFFTWindowSize;  // Size of FFT window
FtVector mSpectrum;           // Hanning windowed FFT data (1024 points, or NULL)
FtVector mReducedSpectrum;    // 1-octave FFT data (10-12 points)
FtVector mBandSpectrum;       // 2.5-octave FFT data (4 points -- spectral bands)
FPartialVector mSpectralPeaks;// List of major spectral peak indeces
FPartialVector mSpectralTracks; // List of tracked peak frequencies
FeatureDatum mSpectralCentroid; // Spectral centroid measure
FeatureDatum mSpectralSlope;  // Spectral slope measure
FeatureDatum mSpectralVariety;// Inter-frame spectral variety measure
                              // Hi-frequency properties
FeatureDatum HiFreqBalance;   // Relative HF level
FeatureDatum HiFreqVariety;   // HF inter-frame spectral variety
FeatureDatum HiFreqCorrelation;// Correlation between HF and audio-band tracks
FeatureDatum mSTrackBirths;   // Spectral peak track births and deaths
                              // LPC features
unsigned int mLPCWindowSize;  // Size of LPC window
FPartialVector mLPCFormants;  // List of LPC formant peaks
FPartialVector mLPCTracks;    // List of tracked LPC formants
FeatureDatum mLPCResidual;    // LPC residual level (noisiness)
FeatureDatum mLPCPitch;       // Pitch estimate
FeatureDatum mLTrackBirths;   // LPC formant peak track births, deaths
                              // Wavelet-domain (FWT) features
FtVector mWaveletCoeff;       // FWT coefficient or NULL
FtVector mWTNSpectrum;        // Reduced FWT HiFreq noise spectrum
FtVector mWTTracks;           // List of tracked FWT peaks
FeatureDatum mWTNoise;        // FWT noise estimate
```

Review

- Signal analysis processing chains
- Feature vector design from app requirements
- Kinds of audio features
- Basic feature statistics