MM'06 Half Day Tutorial

Computer Audition:
An introduction and research survey

Shlomo Dubnov

Music/UCSD

MM'06, October 23 – 27, 2006,

Santa Barbara, California, USA.

http://music.ucsd.edu/~sdubnov/ComputerAudition.htm

What is Computer Audition?

Computational methods for audio understanding by machine

What is audio understanding?

- Beyond speech
- Beyond target detection or machine monitoring
- No clear denotation, taxonomy. Sound objects are “illusive”, “ambiguous”, “transparent”

This is not a standard pattern recognition or audio engineering task.

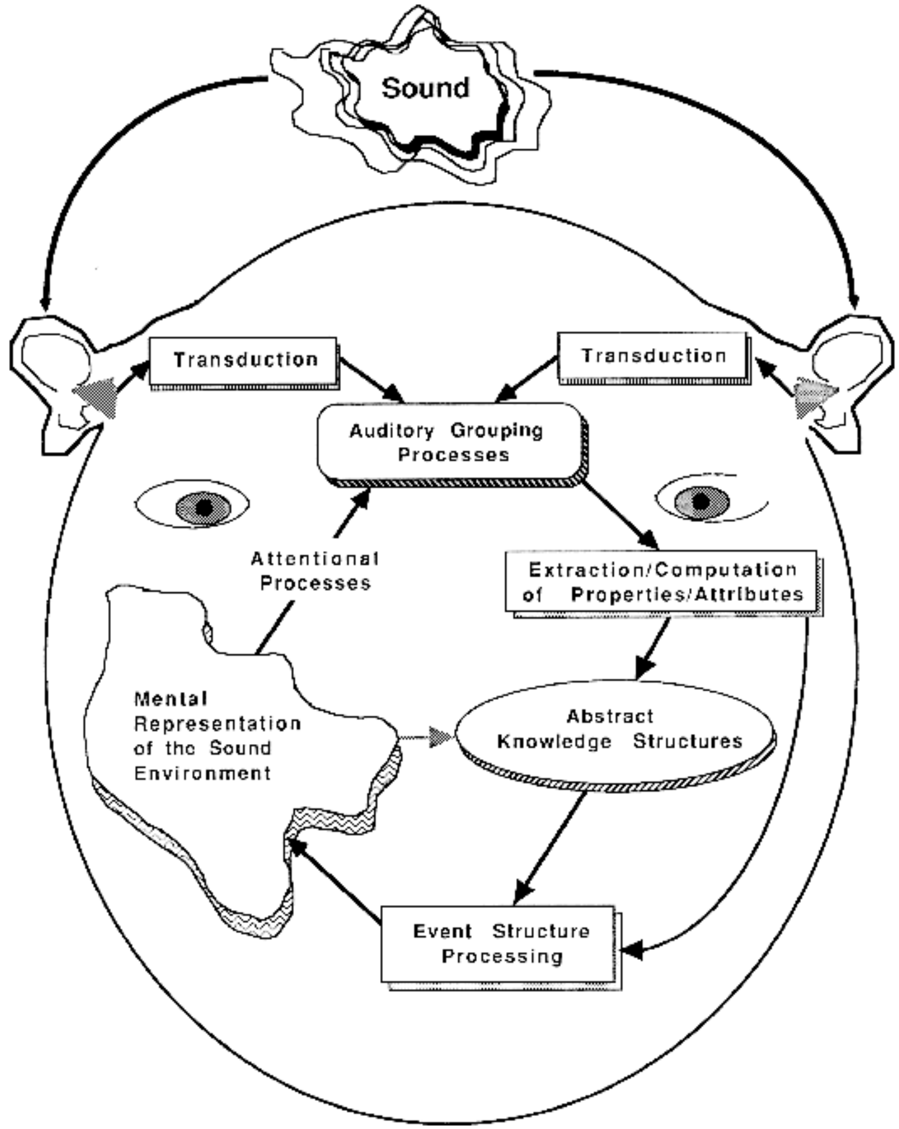
Audio Understanding?

- Music Information Retrieval
- Auditory Scene Analysis
- Computer Generated Music
- Machine Musicanship

Research on auditory and music cognition gives important insight into the problem definition and its mechanisms

Example Questions

- What happened?
- Name that tune?
- What genre do you like?
- Listen to sonic art?
- Is it passionate?



Thinking in Sound

The Cognitive Psychology of Human Audition

Edited by Stephen McAdams and Emmanuel Bigand

Auditory and Music Perception / Cognition research

- Sensory transduction
- perceptual organization processes (auditory grouping, perceptual fusion, stream formation)
- perception of stimulus qualities or attributes (pitch, loudness, height, timbre)
- perceptual categorization and identification of objects, events, and patterns (matching to lexicon)
- memory and attention processes
- musical and auditory knowledge
- mental representation (primal sketch, large scale relations, musical form, narrative)
- grammars of large-scale temporal structures (linguistic ideas applied to music)
- problem solving and reasoning (rarely related to auditory problem solving as might be involved in musical composition, for example)

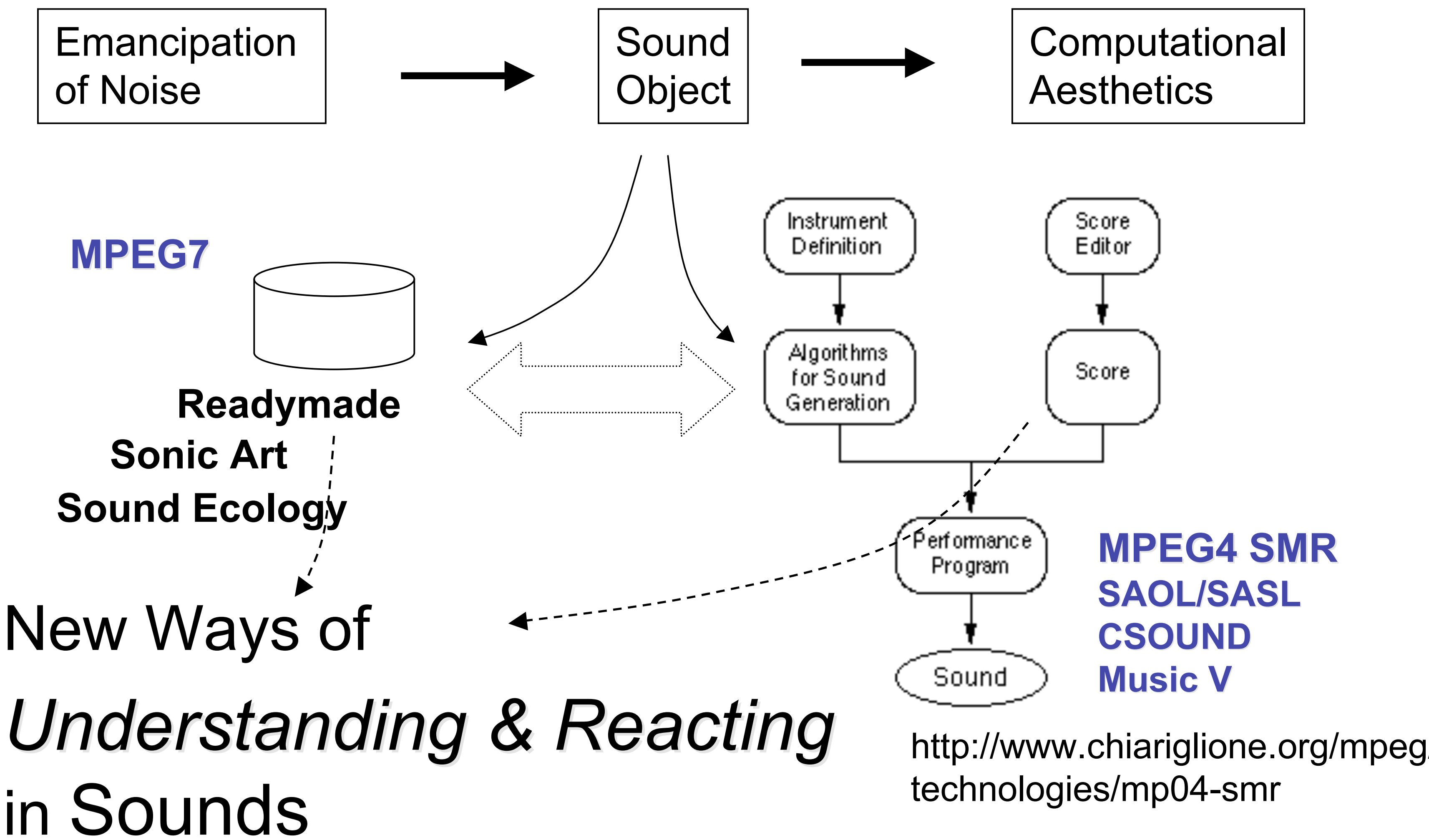
Time scales of music perception

- Pitch (lowest note on the keyboard to highest)
- Timbre (above highest to 20Khz, speech formants)
- Auditory Stream formation (Perceptual fusion):
  - Truax mentions the threshold of approximately 50 milliseconds per event, or 20 per second,
  - Stockhausen mentions 1/16 second threshold
- Phrase, Texture (echoic memory - 0.5 up to 2 sec.)
- Gesture
  - 1) breathing, moderate arm gesture, body sway "phrase" .1 - 1 Hz
  - 2) heartbeat, sucking/chewing, locomotion, "tactus" 1 - 3 Hz
  - 3) speech/lingual motion, hand gesture, digital motion "tatum" 3 - 10 Hz
- Rhythm (tactus range 300-800 msec)
- Short term (working) memory (5 to 9 items stored, uses categorization)
- Perceptual Present
  - Paul Fraisse, Eric. F. Clarke - 7 to 10 sec.
- Long term memory (episodic, semantic, procedural)
  - McAdams “Form bearing dimensions”

Applications

- Recognition of natural, machine, man made or musical sounds, Genre recognition, Query by humming
- Music summarization and thumbnailing, Music annotation
- Audition driven signal processing, synthesis using sound description
- Modeling of musical affect and aesthetics
- Computer aided composition and machine improvisation

Music and Technology



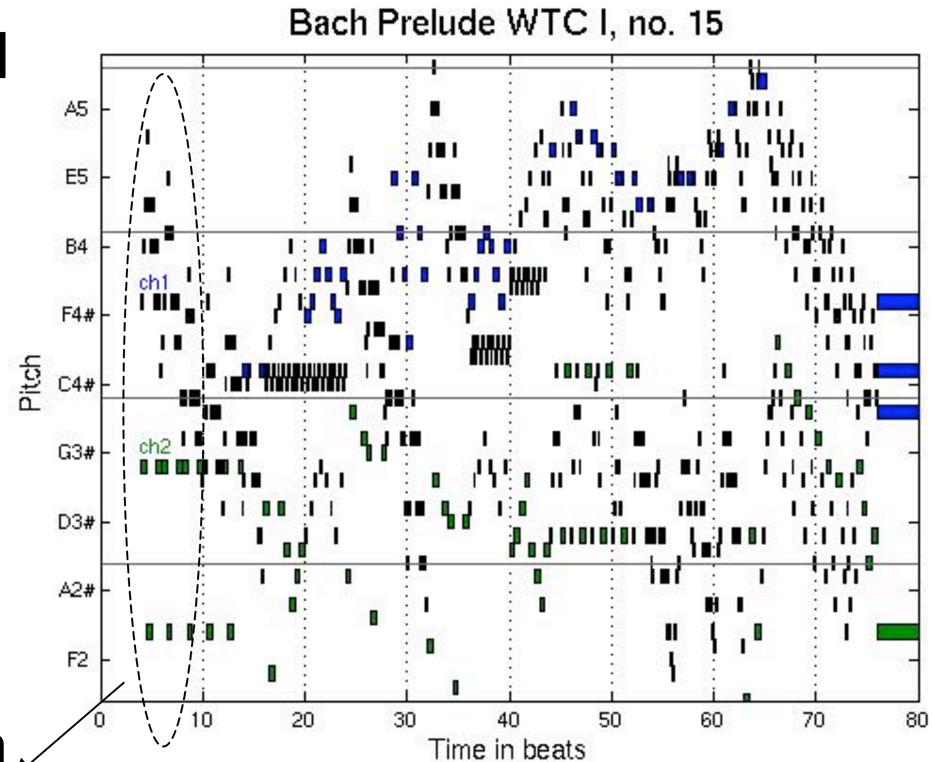
Outline

- Introduction ✓
- Part I : Representation - Signal & Symbolic
- Part II : Alignment and Comparison
- Part III: Audio Semantics
- Conclusion

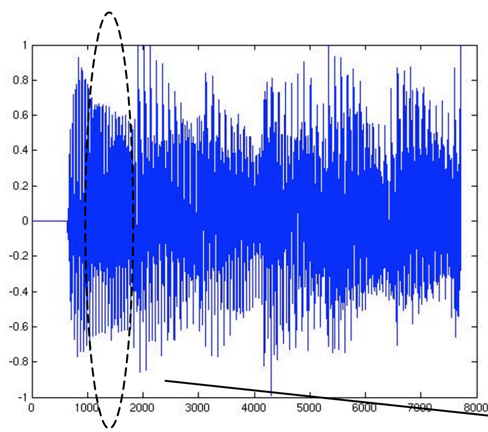
Part I : Representation

- Digital Audio
- Fourier Analysis
- Analysis-Synthesis
  - Non-Parametric: Phase Vocoder, Sinusoidal
  - Parametric: Source-Filter
- Sound Description Files (SDIF)
- Pattern Playback
- Synthesis and MIDI

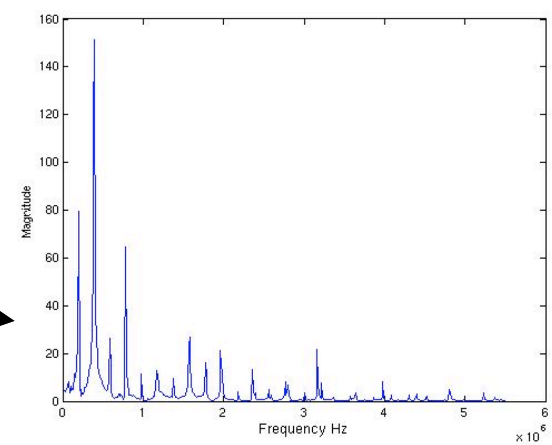
Piano Roll



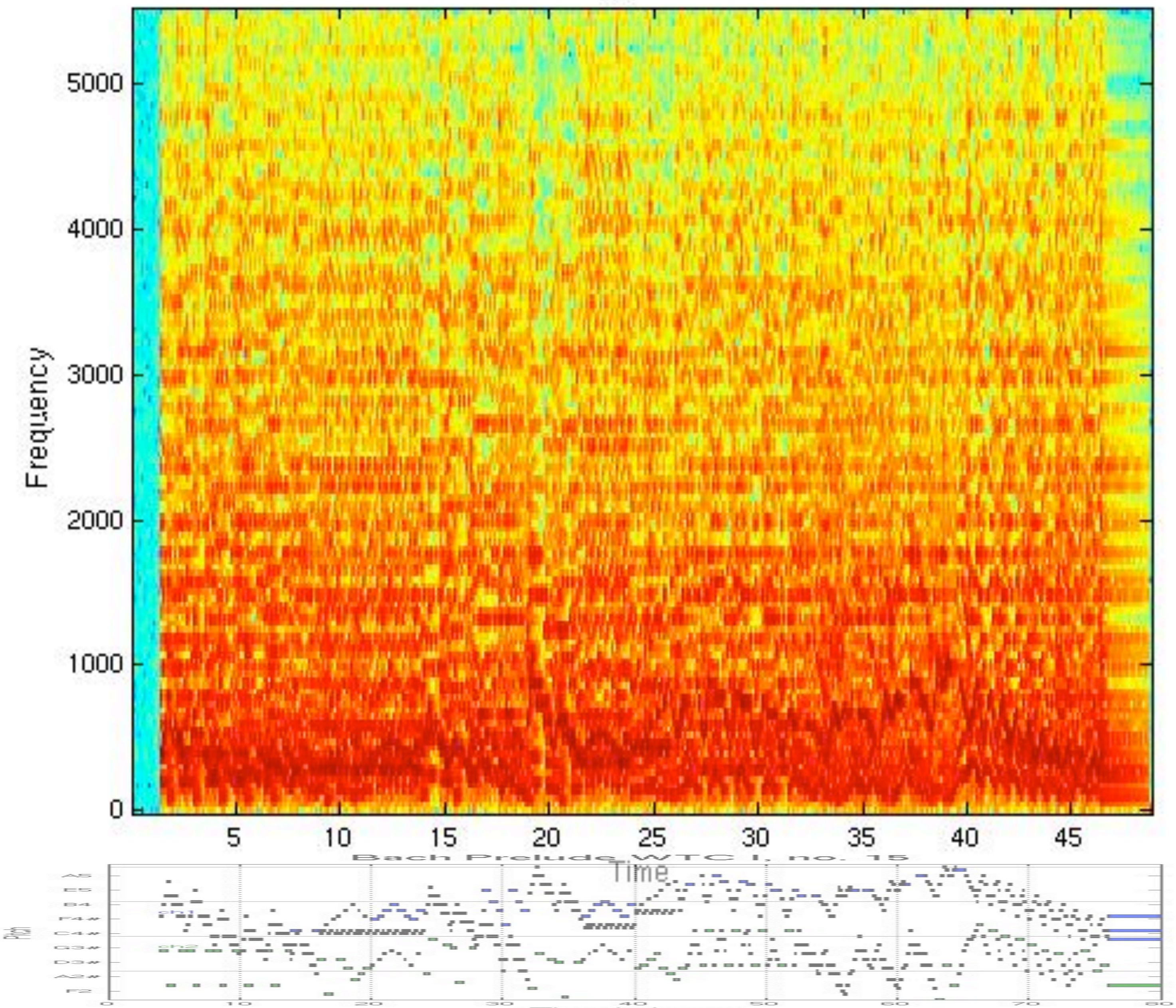
Waveform



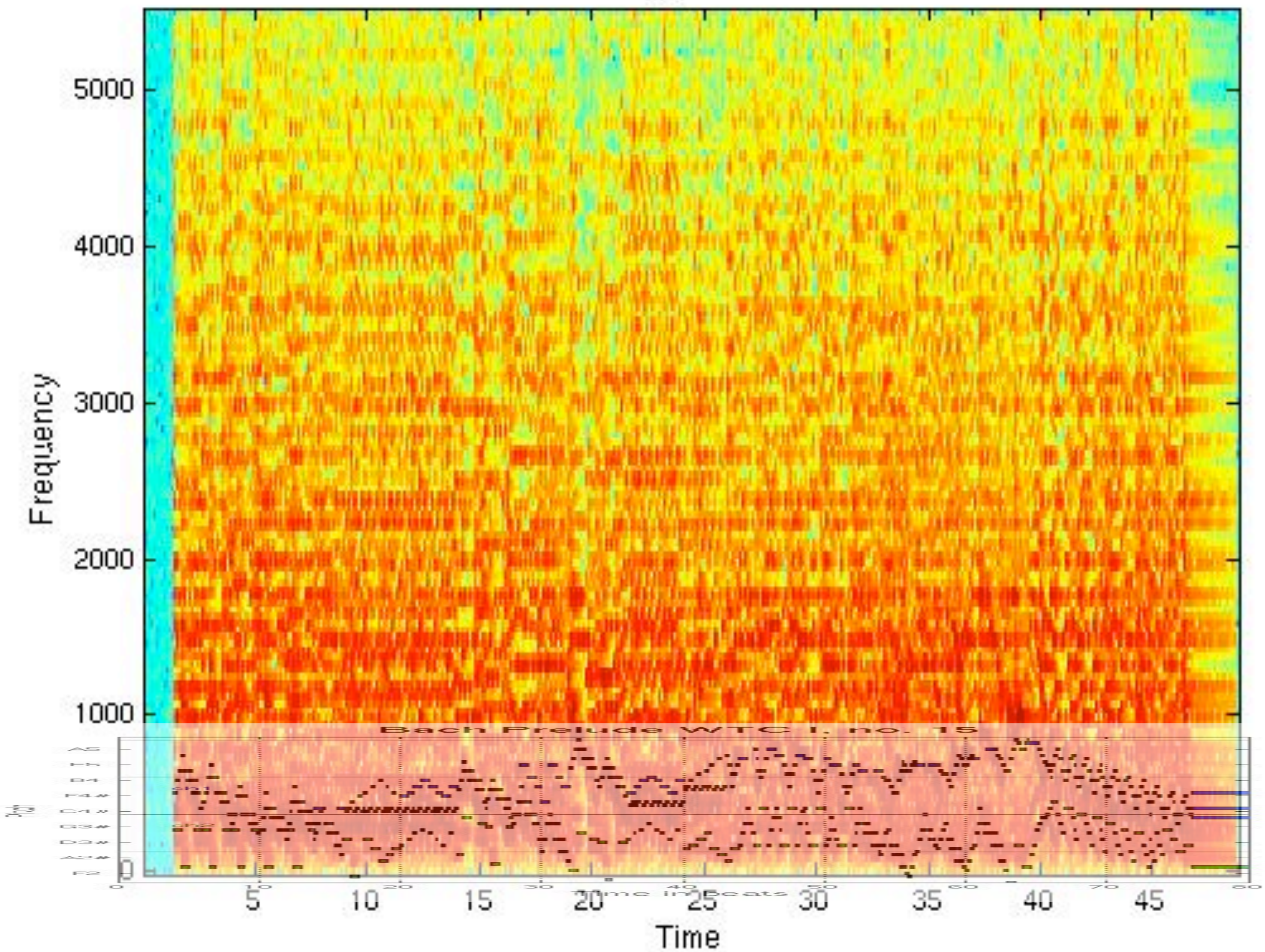
FFT Magnitude



MIDI Rendering of Bach Prelude



MIDI Rendering of Bach Prelude



Digital Audio (cont.)

Reading audio in Matlab

- WAVREAD Read Microsoft WAVE (".wav") sound file.

[Y,FS,NBITS]=WAVREAD(FILE,[N1 N2])

WAVWRITE(Y,FS,NBITS,WAVEFILE)

- Also auread, auwrite
- MP3READ, MP3WRITE

http://www.mathworks.com/matlabcentral/ -> search for mp3read (windows only)

http://labrosa.ee.columbia.edu/matlab/

(Windows and Unix). Requires mpg123, and mp3info

OS X: http://sourceforge.net/projects/mosx-mpg123

http://mp3info.darwinports.com/

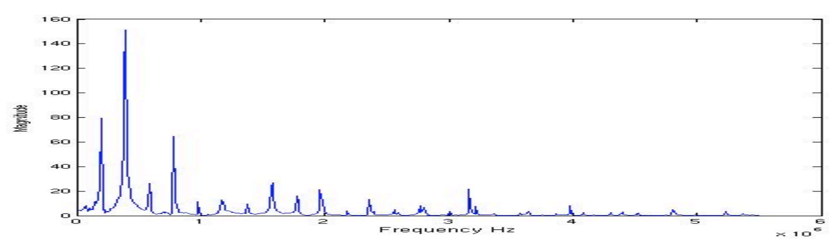
Fourier Analysis

Change or representation $$x(t) \overset{\mathcal{F}}{\longleftrightarrow} X(f)$$

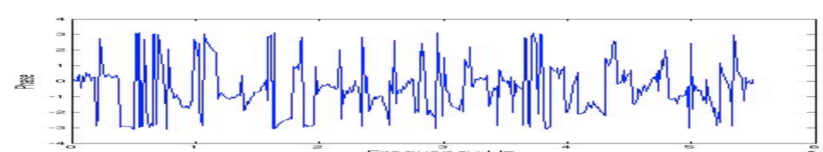
DFT (discrete time and discrete frequency) $$x(n) \overset{\mathcal{DFT}}{\longleftrightarrow} X(k)$$

- –Sound vector of size N
- –Results in N spectral “bins”
- –Freq. resolution Fs/N
- –N/2 Amplitudes |X(k)| and Phases atan=Im(X(k))/Re(X(k))

```
X = fft(x(15101:16100)));
plot([0:499]*fs,abs(X(1:500)))
```



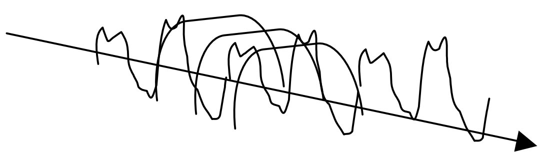

```
plot([0:499]*fs,angle(X(1:500)))
```



Analysis-Synthesis

Short Time Fourier Transform (STFT)

– Sequence of DFT's
– Sliding window in time *w(n)*



$$X(k,\tau) = DFT(w(n-\tau)\cdot x(n))$$

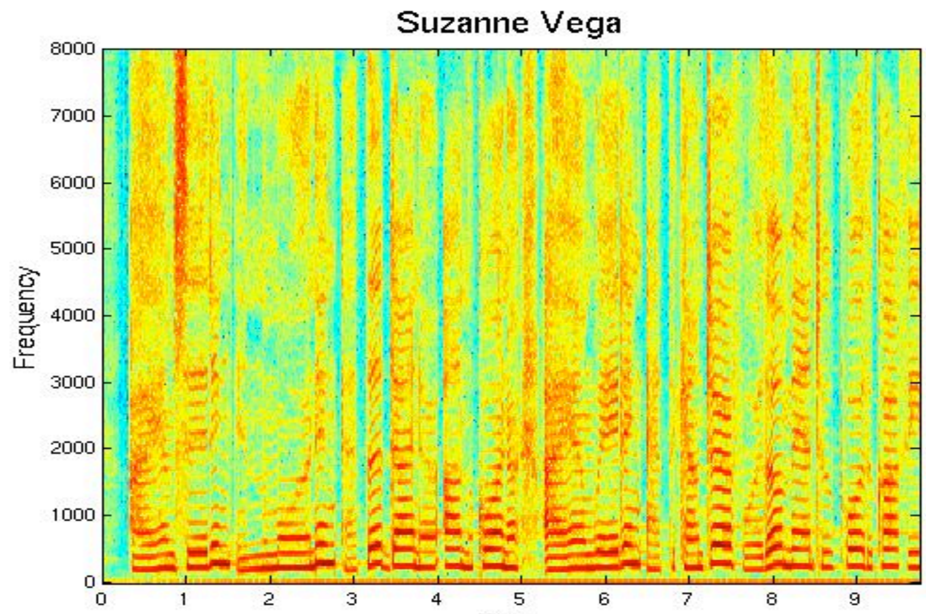
– Localizes signal both in frequency and in time

X = SPECGRAM(x,NFFT,Fs,WINDOW,NOVERLAP)

- Narrow band vs. wide band analysis
- Constant Overlap Add (COLA) conditions for signal reconstruction from STFT

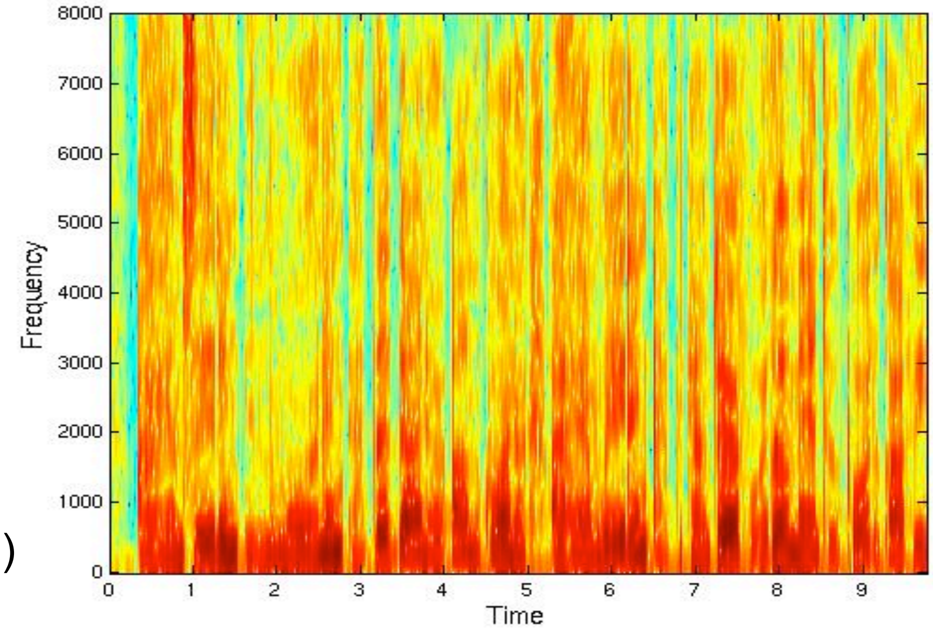
NB

specgram(x,512,fs)



WB

specgram(x,512,fs,hanning(64),32)



THE VODER

- Ten bandpass filters
- Wrist bar switches between "buzz" and "hiss"
- Foot pedal controls the pitch

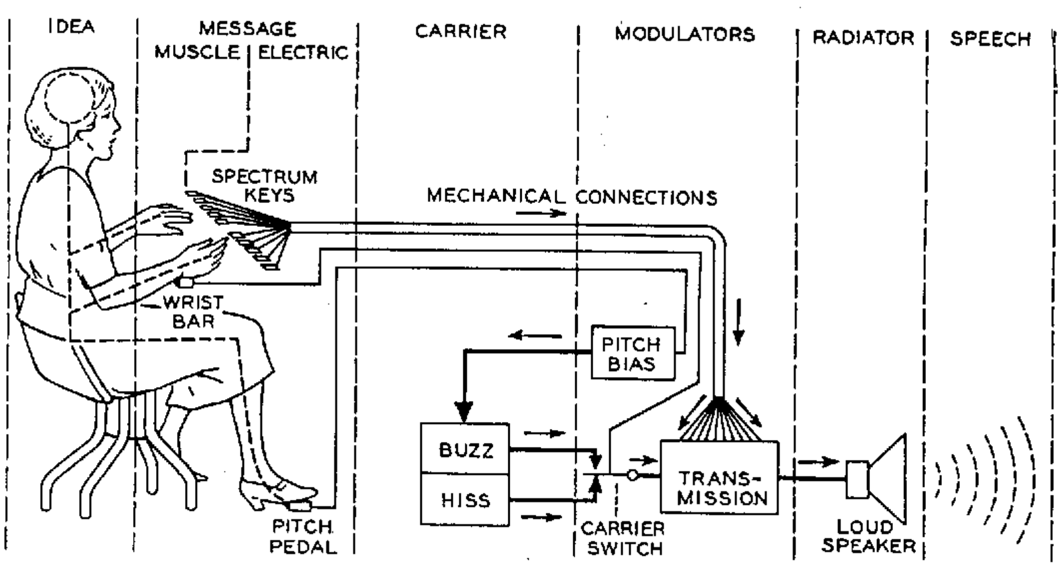
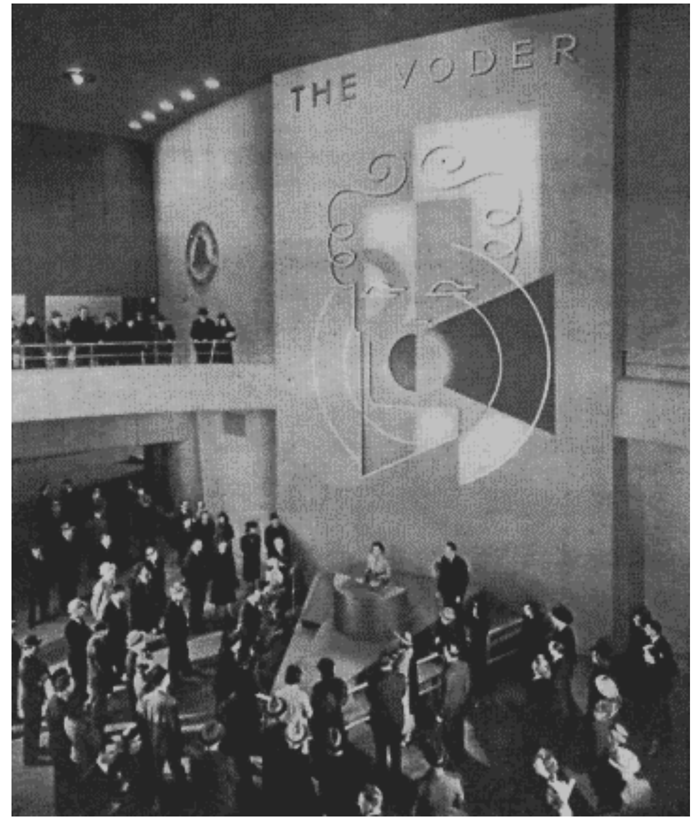


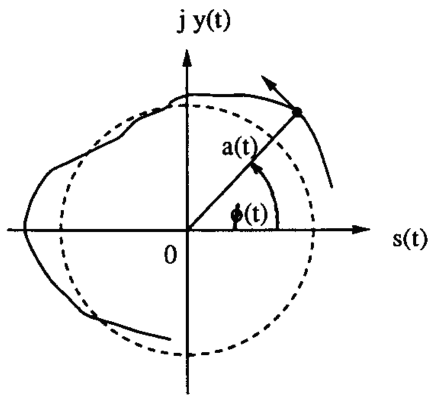
Fig. 8—Schematic circuit of the voder.



The 1939 New York World's Fair

Phase Vocoder

- Based on notion of instantaneous frequency

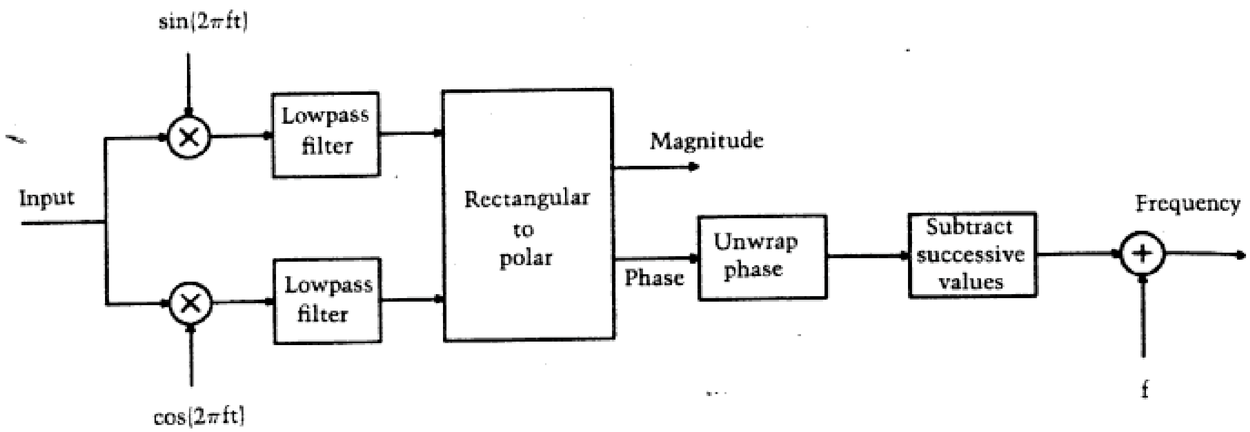


$$f(k,\tau) = \frac{\Delta\phi(k,\tau)}{2\pi\Delta\tau}$$

$$\phi(k,\tau) = angle(X(k,\tau))$$

- Each “bin” must contain a single sinusoid

Phase Vocoder



- Allows timescale modifications:
  - Magnitude is linearly interpolated
  - Preserves phase increment
- OK for time stretching < 10%
  - Otherwise Phasiness, Ringing

Constant Q transform

- bank of filters
- geometrically spaced center frequencies

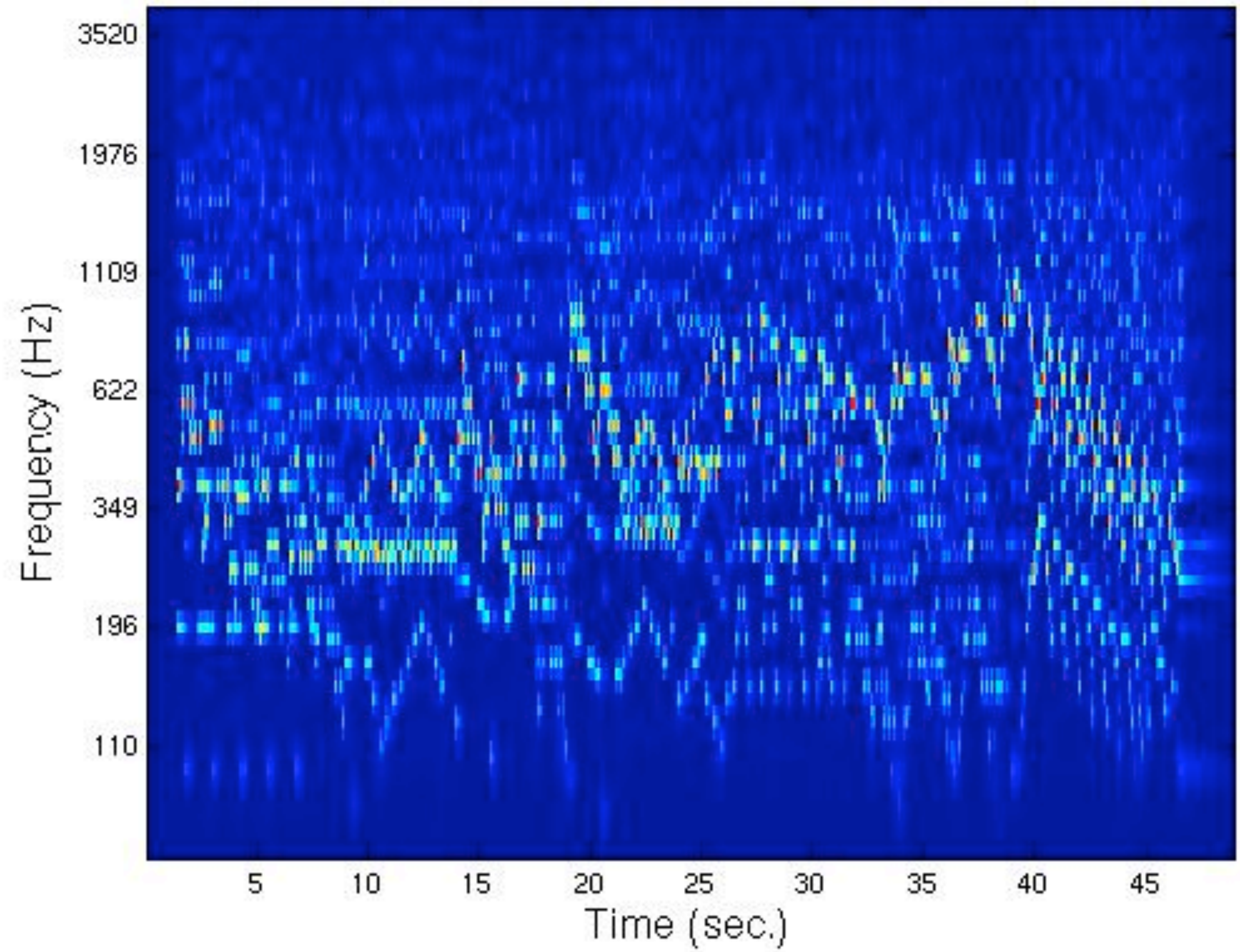
$$f_k = f_0 \cdot 2^{k/b}, \quad k = 0..$$

- bandwidth of the *k*-th filter

$$\Delta_k = f_k (2^{1/b} - 1)$$

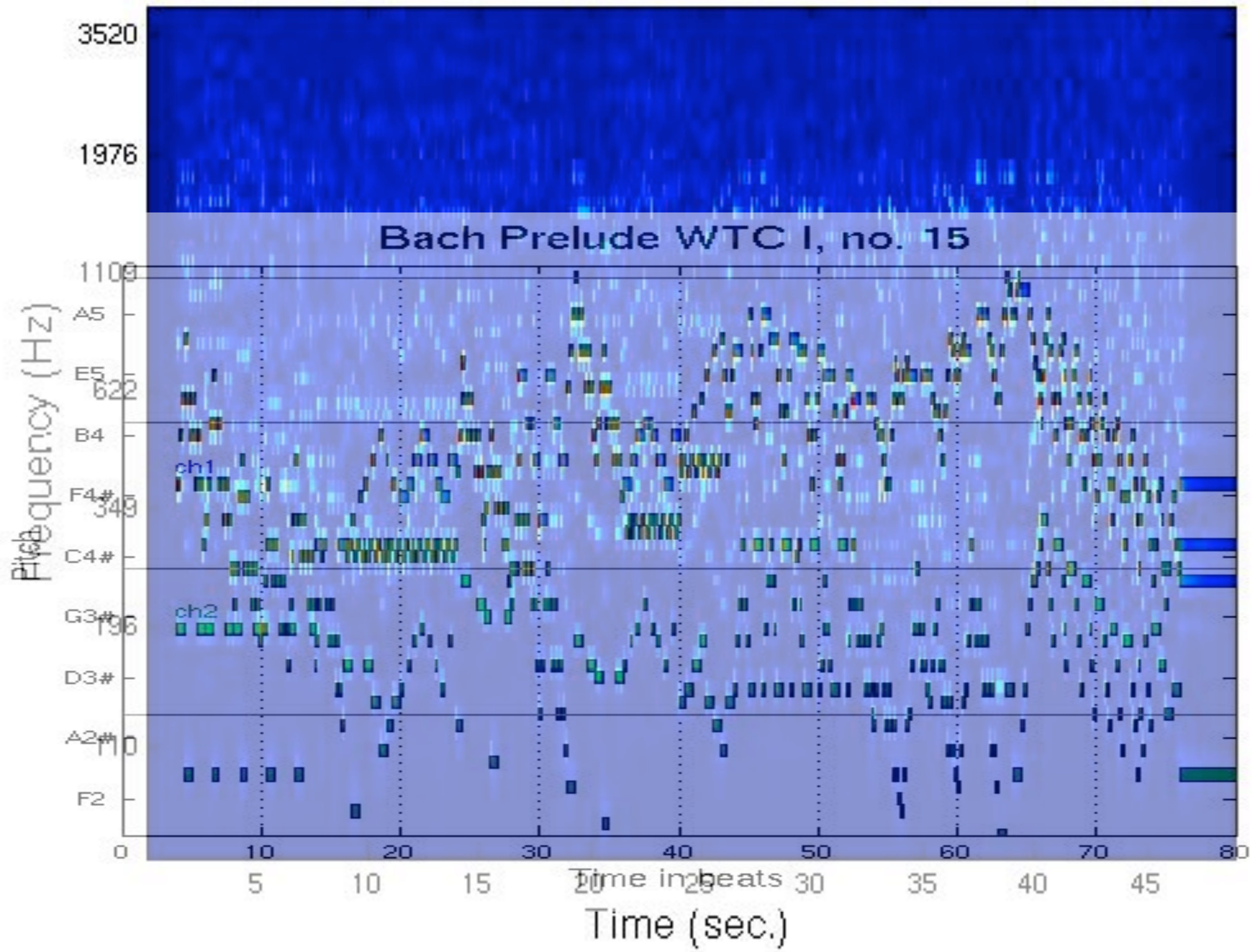
⇨ Constant frequency to resolution ratio $Q = \frac{f_k}{\Delta_k} = \frac{1}{2^{1/b} - 1}$

Constant Q with semitone spacing




```
cqgram(sig,1024,256,midi2hz(60-24),midi2hz(108));
```

Constant Q with semitone spacing

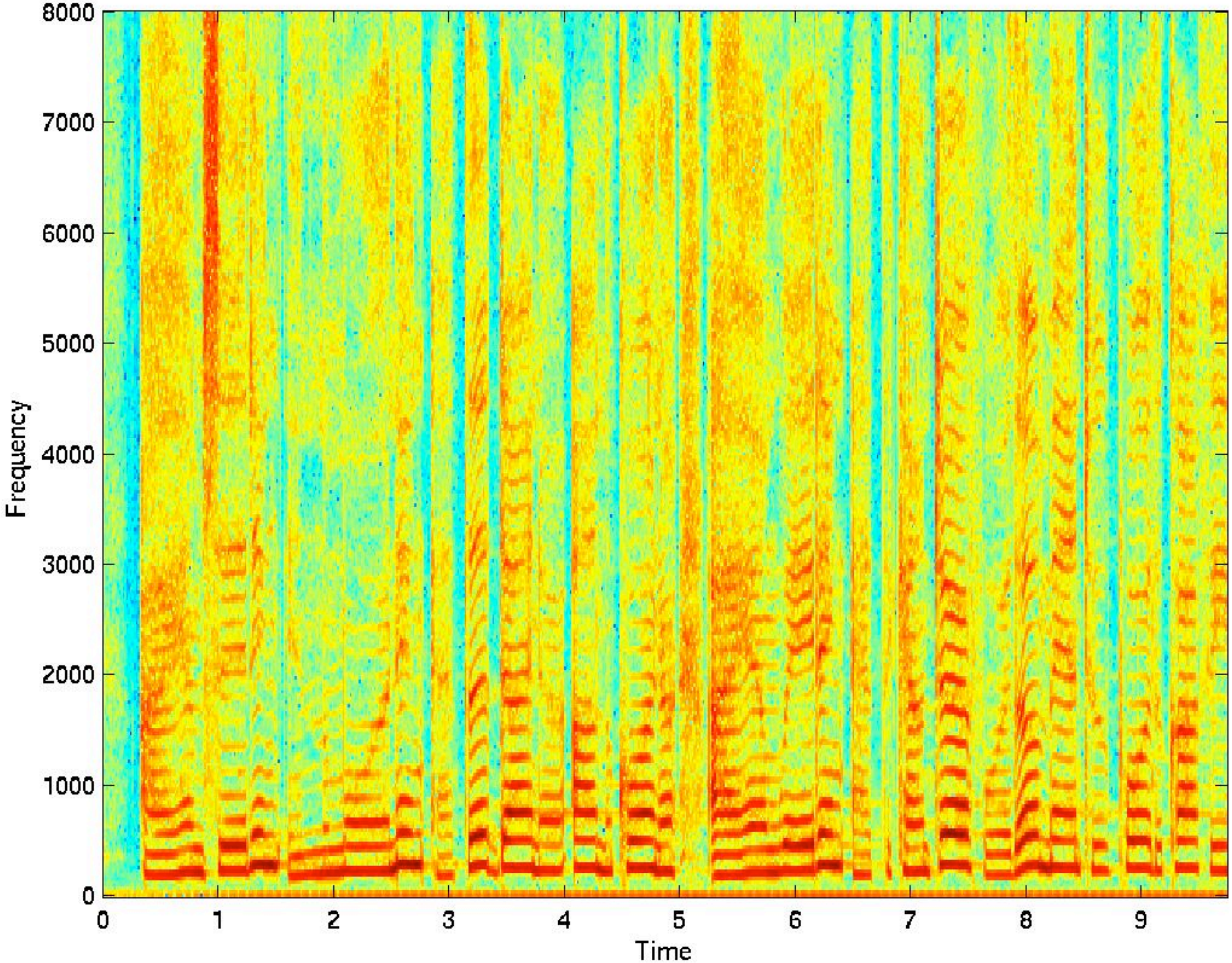


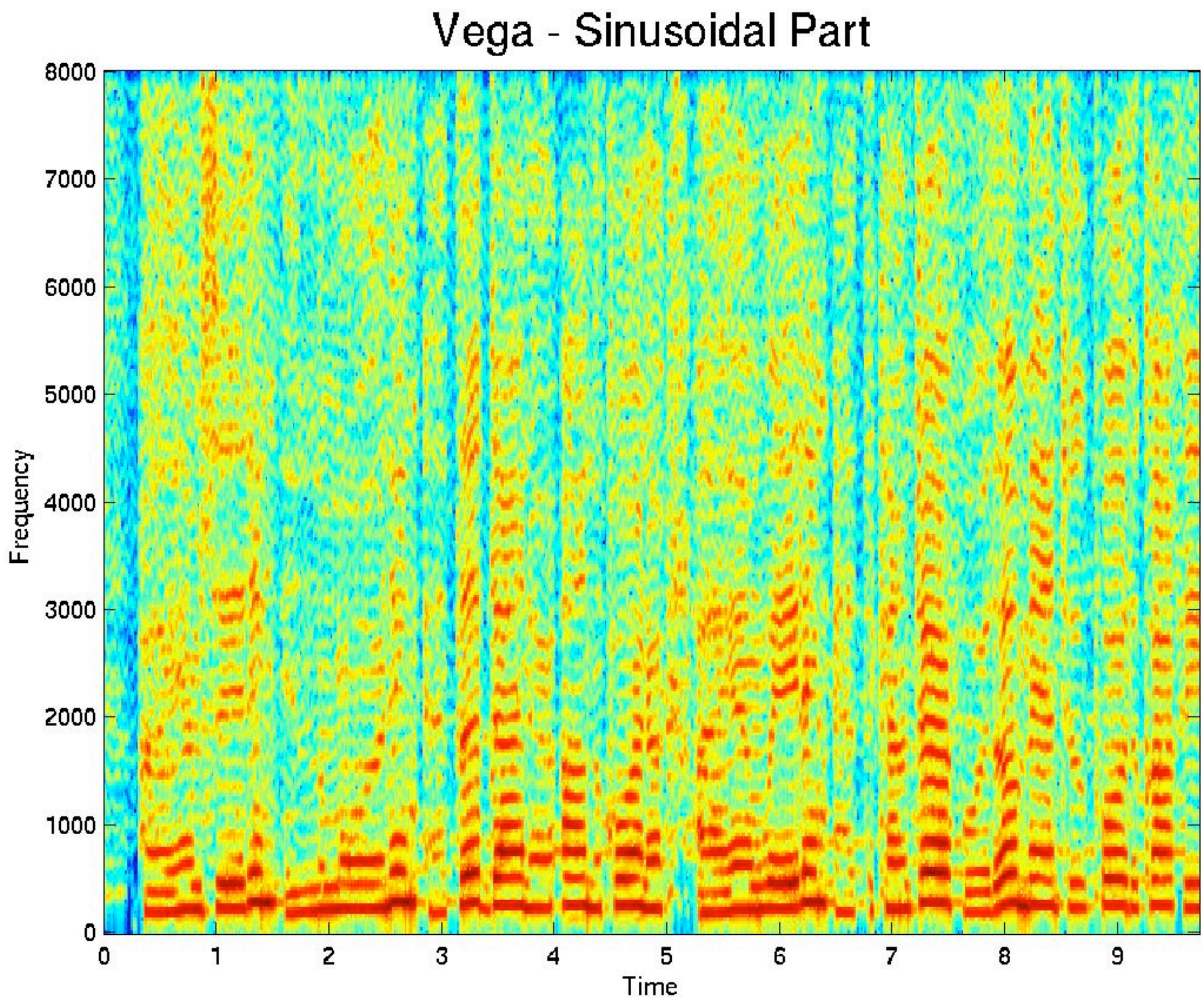
cqgram(sig,1024,256,midi2hz(60-24),midi2hz(108));

Sinusoidal Models

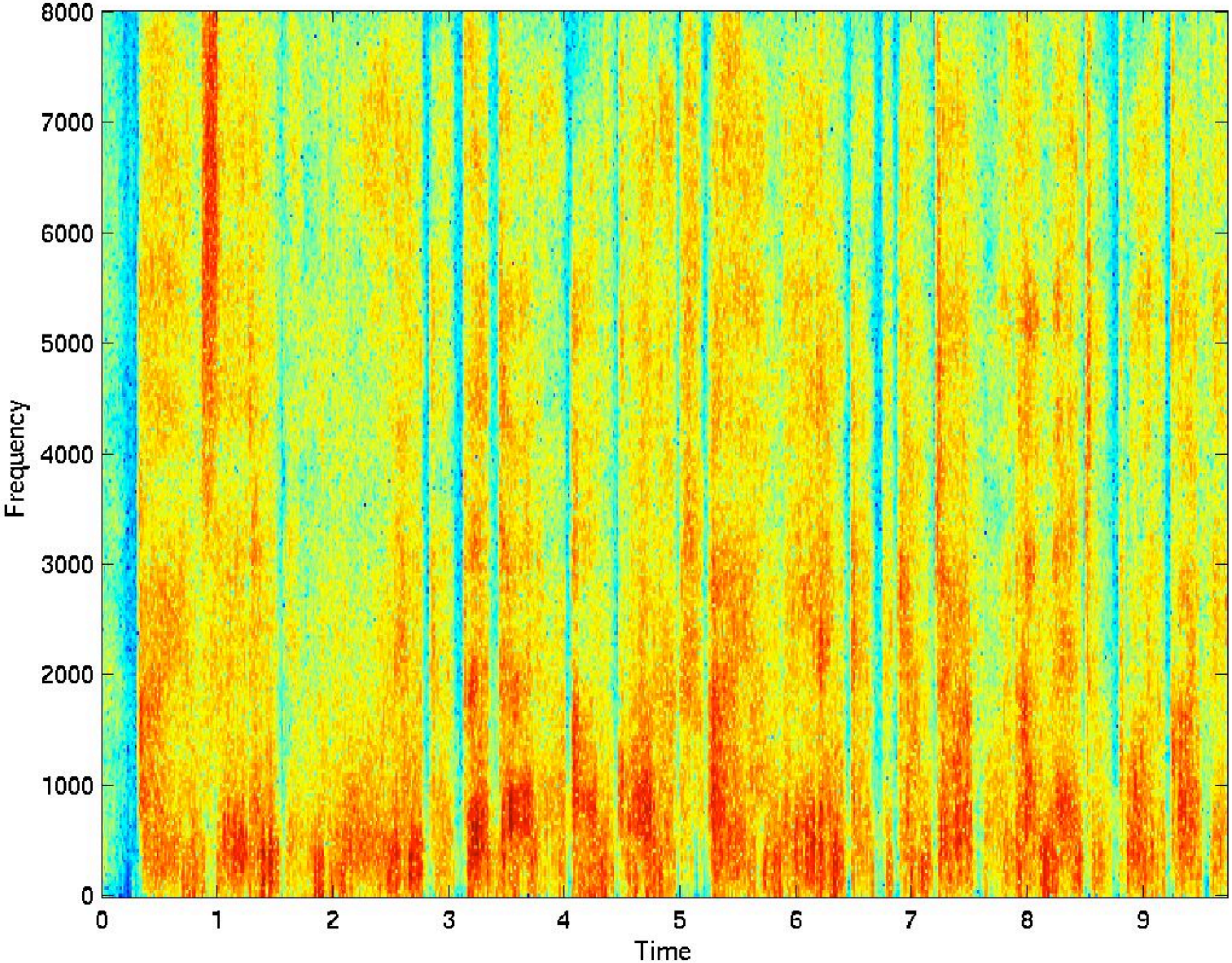
- Explicitly estimate sinusoidal parameters:
  - Amplitude, Frequency, Phase
- Parameters updated every 5-10 msec.(!)
- Separate modeling of noise components
  - STFT, Source-Filter, Bandwidth enhanced sinusoids
- Useful as Sound Descriptors - SDIFF
- Models:
  - McAuley and Quatiery - STC
  - Smith and Serra - PARSHL, Serra - SMS
  - Griffith and Lim - MBE
  - Stylianou - HNM
  - Purnhagen, Meine - HILN
  - Fitz - Loris
  - Dubnov - YASA

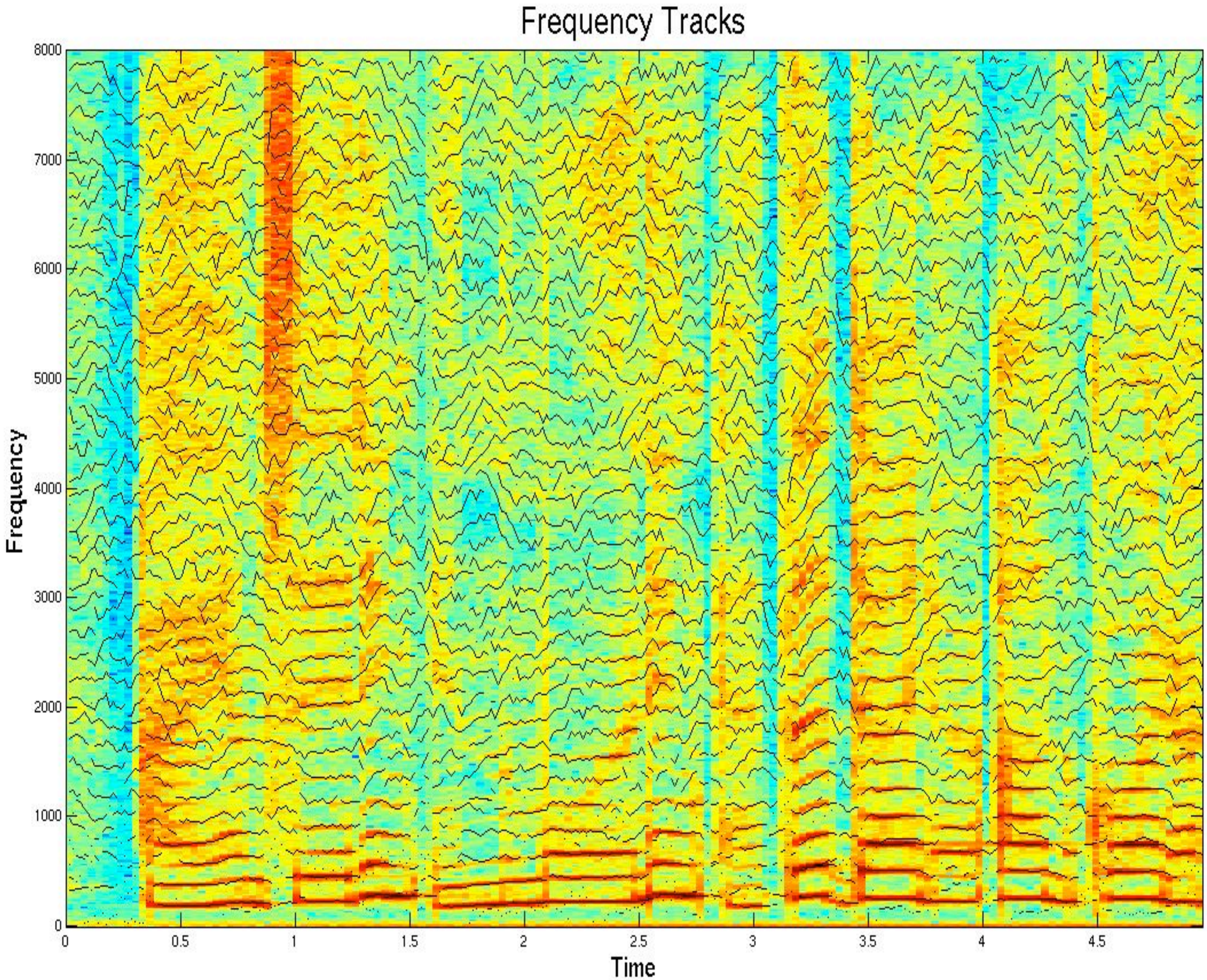
Vega - Original Signal

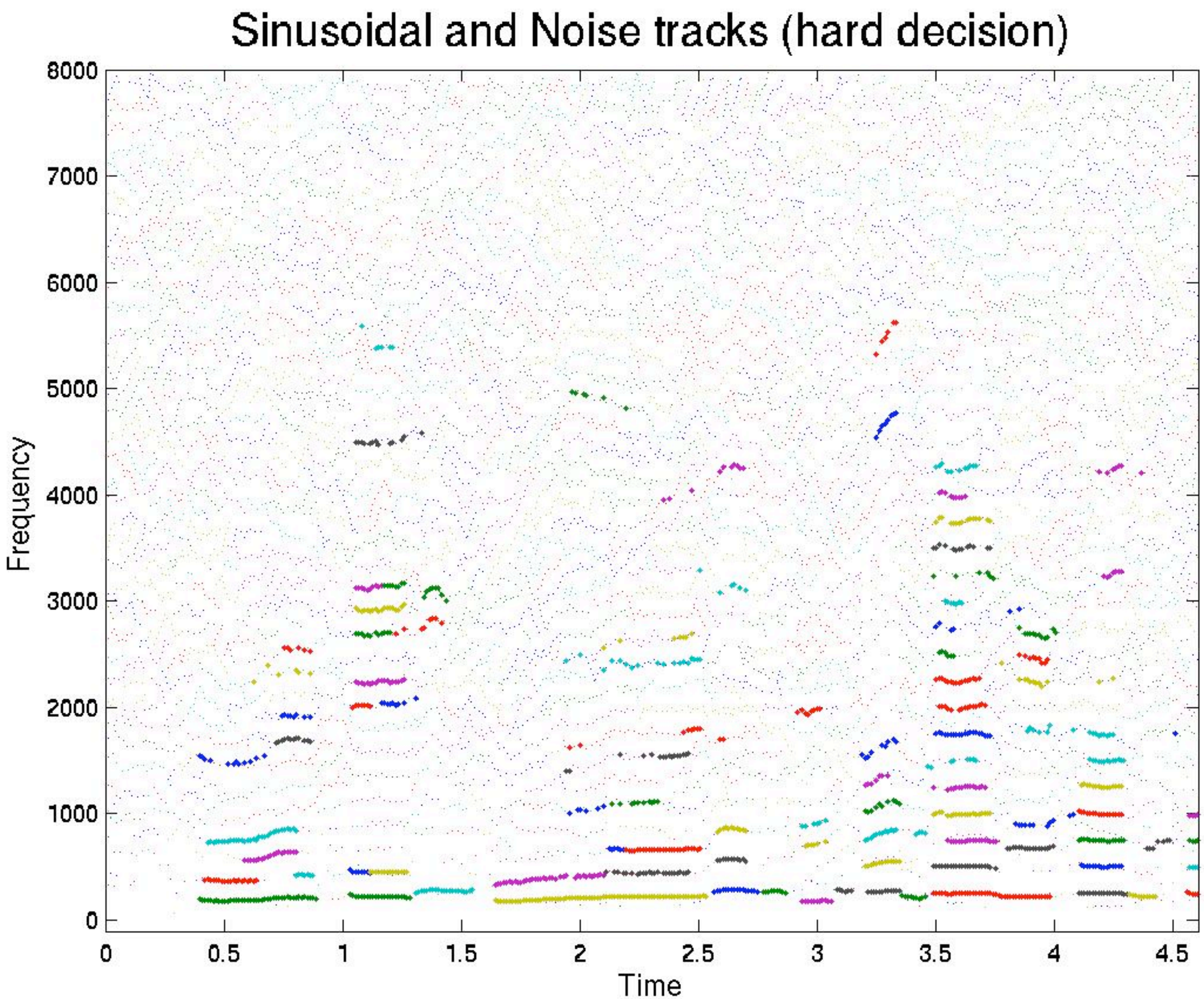



Vega - Sinusoidal Part
Frequency
Time
8000
7000
6000
5000
4000
3000
2000
1000
0
0
1
2
3
4
5
6
7
8
9

Vega - Noise Part




Frequency Tracks
Frequency
Time
8000
7000
6000
5000
4000
3000
2000
1000
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5


Sinusoidal and Noise tracks (hard decision)
Frequency
Time
8000
7000
6000
5000
4000
3000
2000
1000
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

Examples

S. Vega

Original

Sin.

Noise.

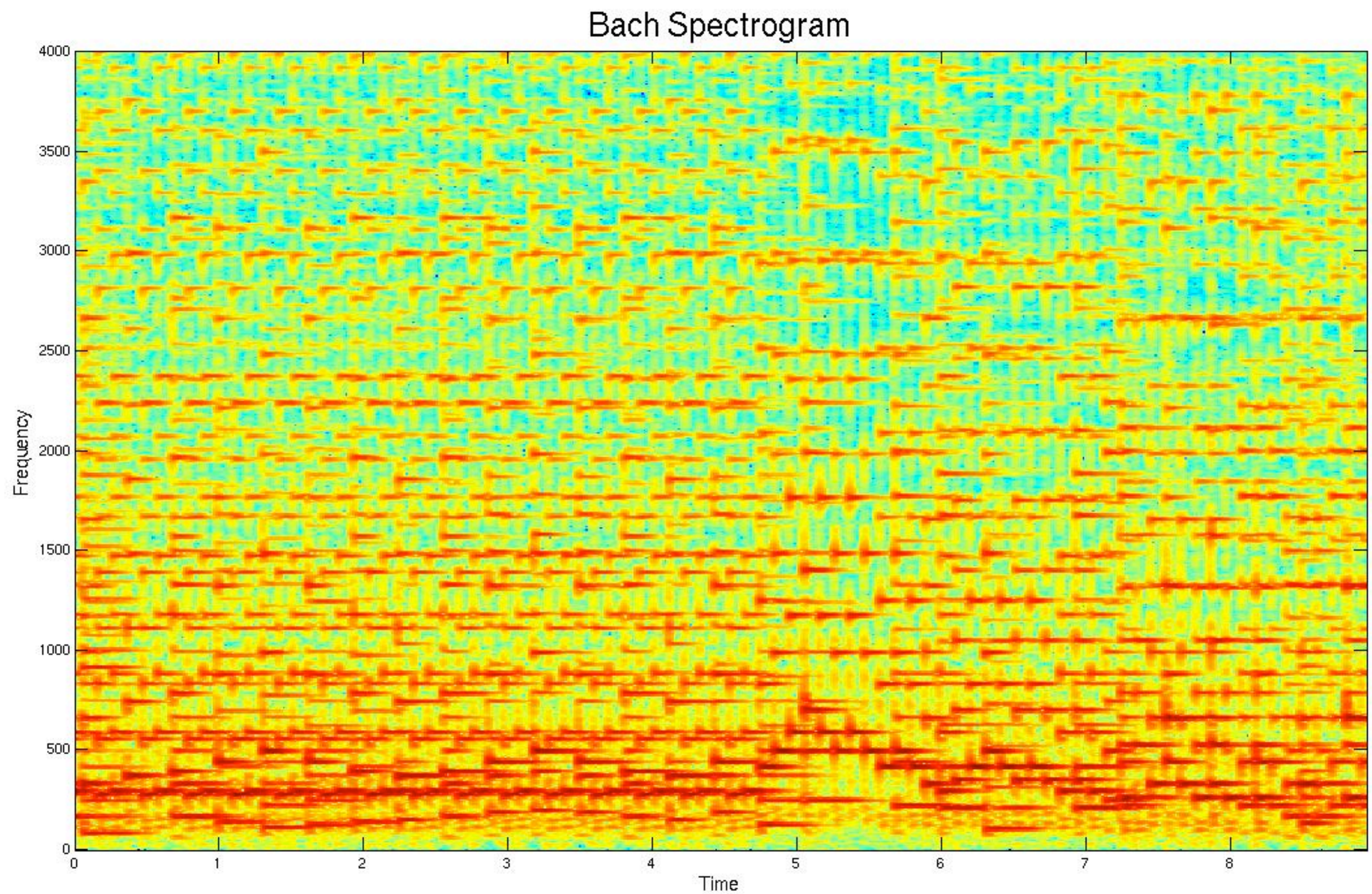
S+N

Cat howl

Original

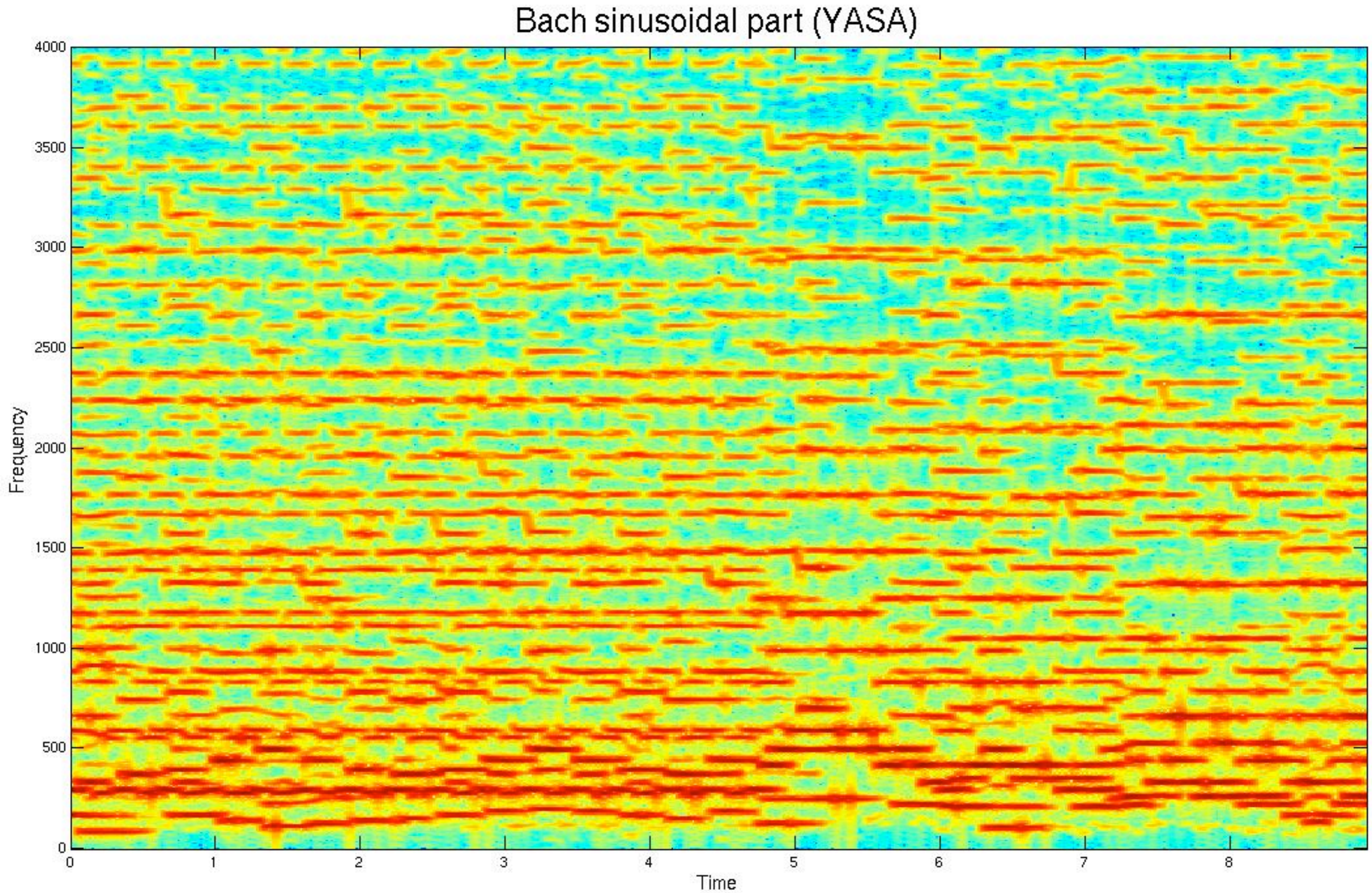
10X pvoc

10X S+N



specgram(z,1024,fs,hanning(1024),1024-128)

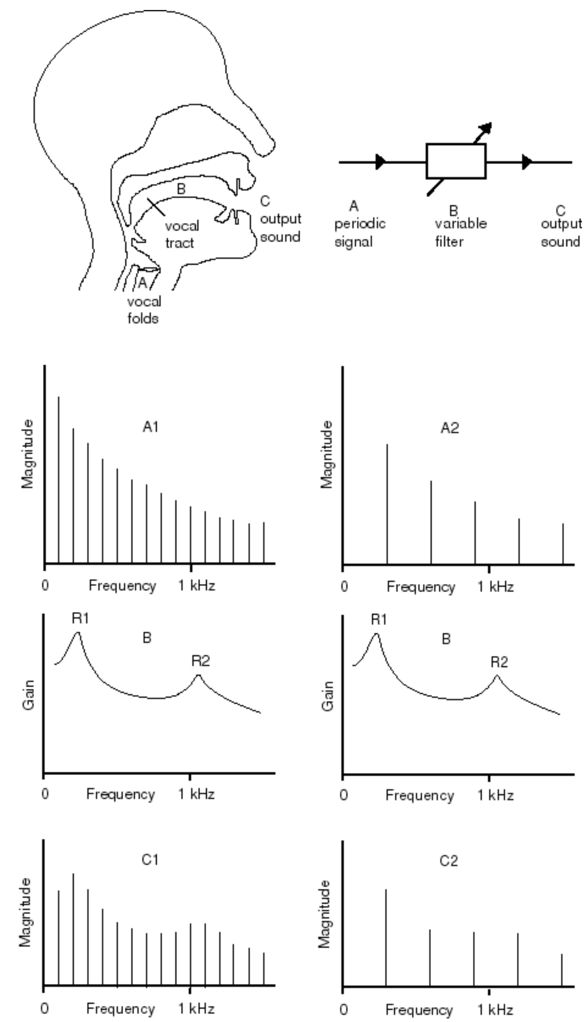
YASA handles polyphonic sounds




```
[f,m,n,T] = yasa(z,1024,4,120,fs);
[F,M,N] = maketracks(f,m,n);
[zs,zn] = synthsntrax(F,M,N,fs,hop);
```

Source-Filter Models

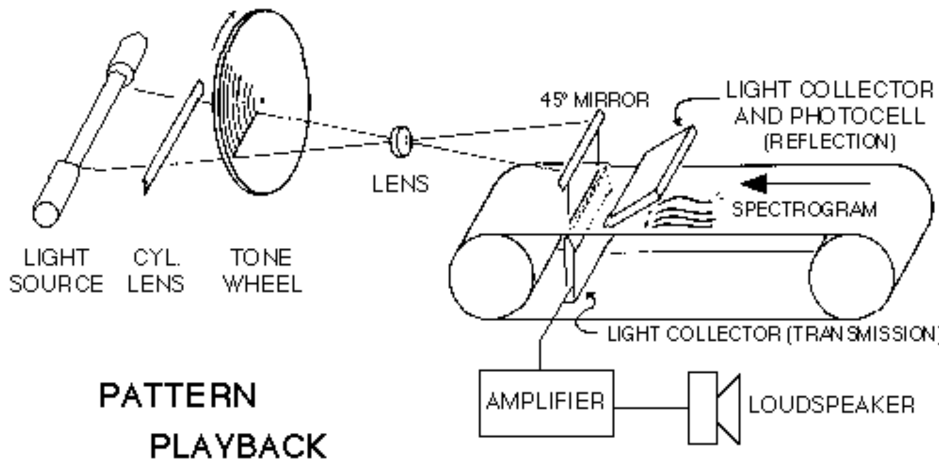
- Assumes sound production mechanism
  - excitation that passes through a filter
- Parameters estimated every ~20 msec.
- Popular in speech processing
  - Source ~ glottal pulses
  - Filter ~ vocal tract
- Requires separate estimation of source parameters (pitch) and filter coefficients (spectral envelope)
- Efficiently estimated by Linear Prediction (LPC)
- Determines speech formants



Example:

- LPC10:
  - 8 kHz sample rate, 180 samples/frame, 44.44 frames/second
  - Order 10 LP analysis:
    - First two coefficients are quantized as log area ratios with five bits each
    - last 8 as reflection coefficients. Number of bits per coefficient decreases with index down to two bits
  - 7 bits used for pitch and voicing decision
  - 5 bits used for gain
  - Total: 54 bits per frame, 2400 bps

Pattern Playback

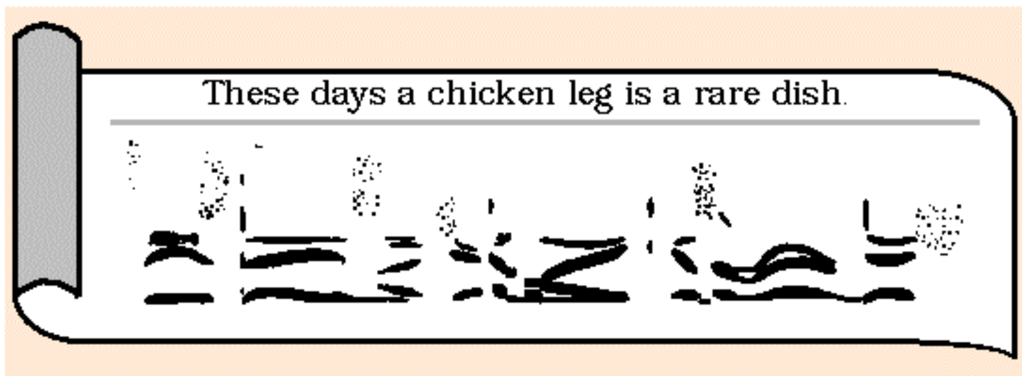


The Pattern Playback is an early talking machine that was built at Haskins Laboratories in the late 1940s

Why Pattern Playback?

- In many cases we analyze some parameters (like spectral magnitude) and ignore others (phase)
- Need a way to re-synthesize sound from a partial representation
- Synthesis using perceptually relevant “patterns”
- Audition and synthesis using same representation
- A way to evaluate performance of computer audition algorithms (and not only “see” the results).
- Today it mostly refers to ways to resynthesize sound from spectral magnitude, cochleagrams and other time-frequency representations.

Why Pattern Playback?



Griffin and Lim,
ASSP-32, No2, 1984

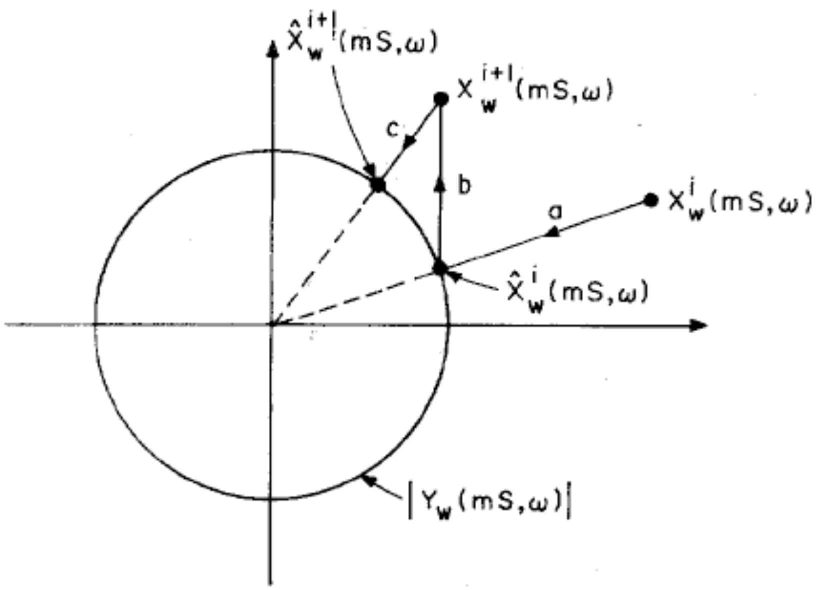


Fig. 7. Successive iterations of LSEE-MSTFTM.

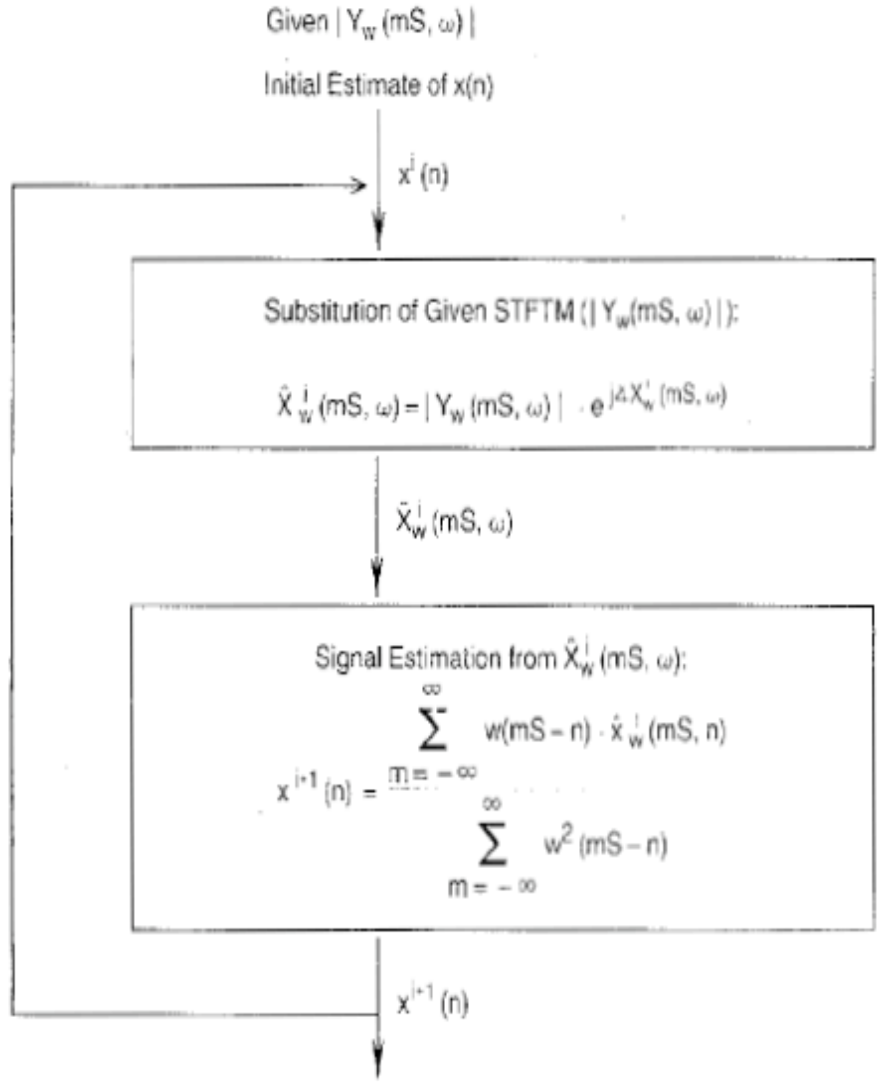


Fig. 1. LSEE-MSTFTM algorithm.

LSEE

Short Time Fourier Transform

$$X(n,w) = \sum_{m=-\infty}^{\infty} x(m)w(n-m)e^{-jwm}$$

Least Squares Signal Estimation From Modified STFT

$$D[X_e(n,w),Y(n,w)] = \sum_{m=-\infty}^{\infty} \frac{1}{2\pi} \int_{-\pi}^{\pi} \left|X_e(m,w) - Y(m,w)\right|^2 dw \quad \Rightarrow \quad x_e(n) = \frac{\sum_{m=-\infty}^{\infty} w(m-n) f_m(n)}{\sum_{m=-\infty}^{\infty} w^2(m-n)}$$

Modified STFTM

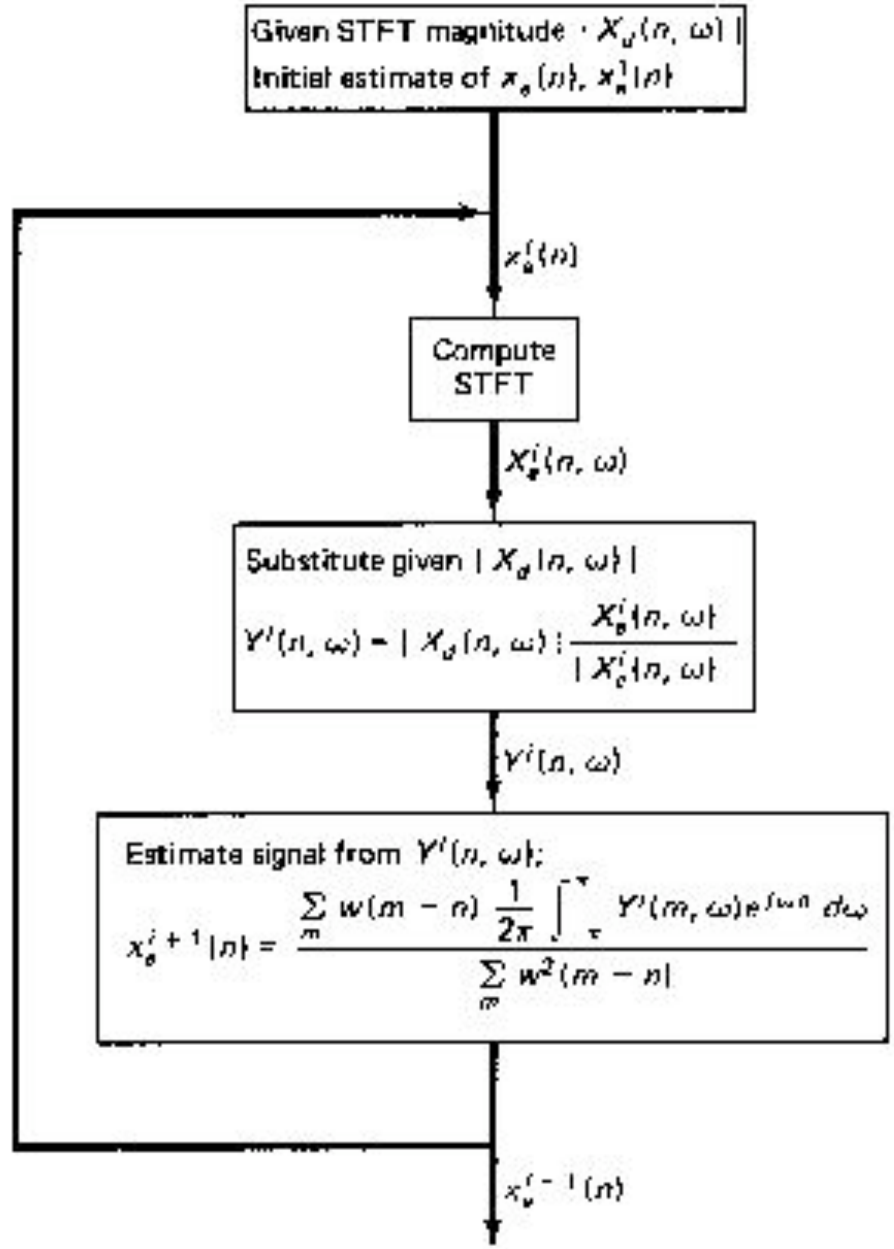
$$D[|X_e(n,w)|,|X_d(n,w)|] = \sum_{m=-\infty}^{\infty} \frac{1}{2\pi} \int_{-\pi}^{\pi} [|X_e(m,w)| - |X_d(m,w)|]^2 \, dw \quad \Rightarrow$$

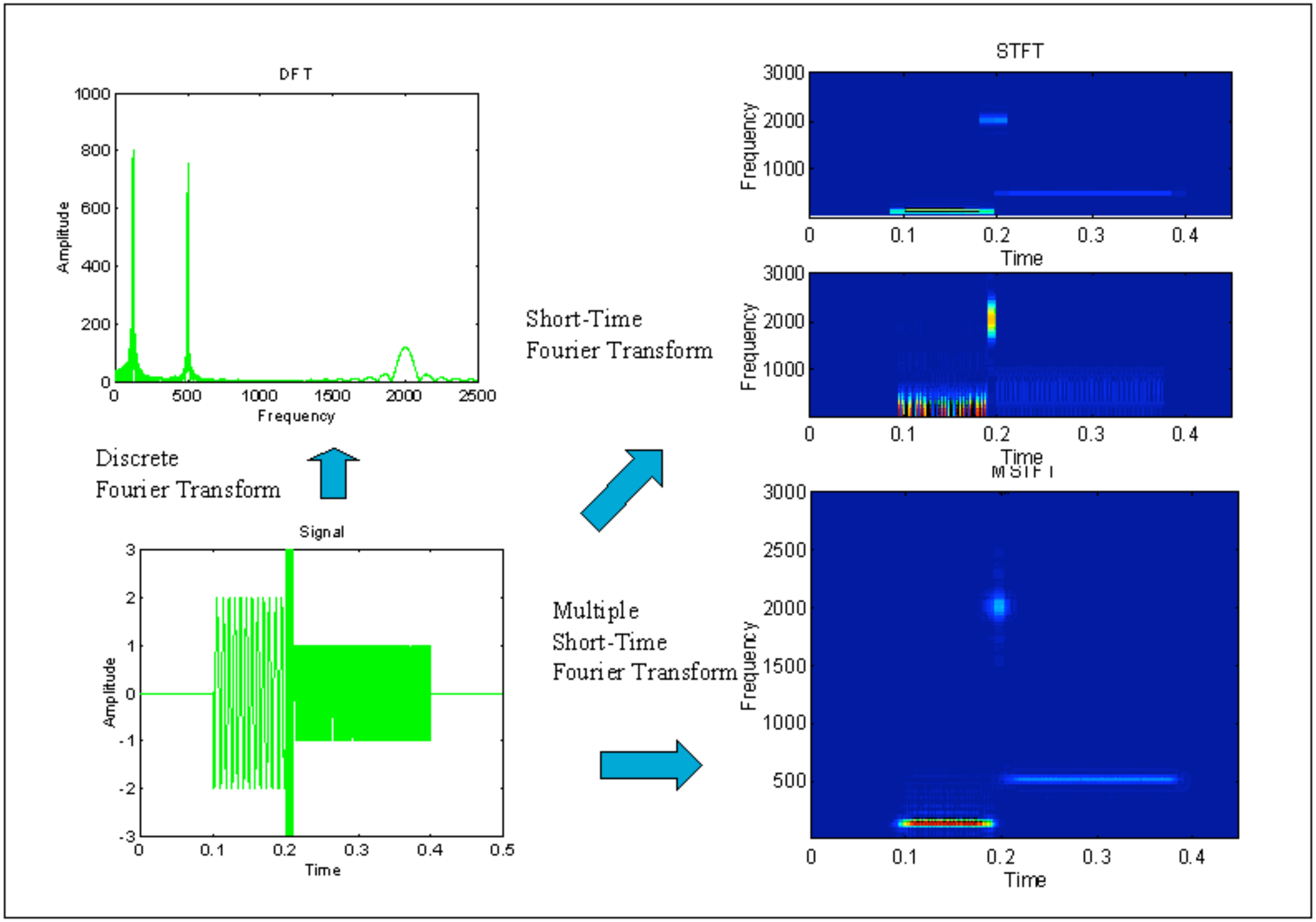
Iterative solution:

Do same thing over and over

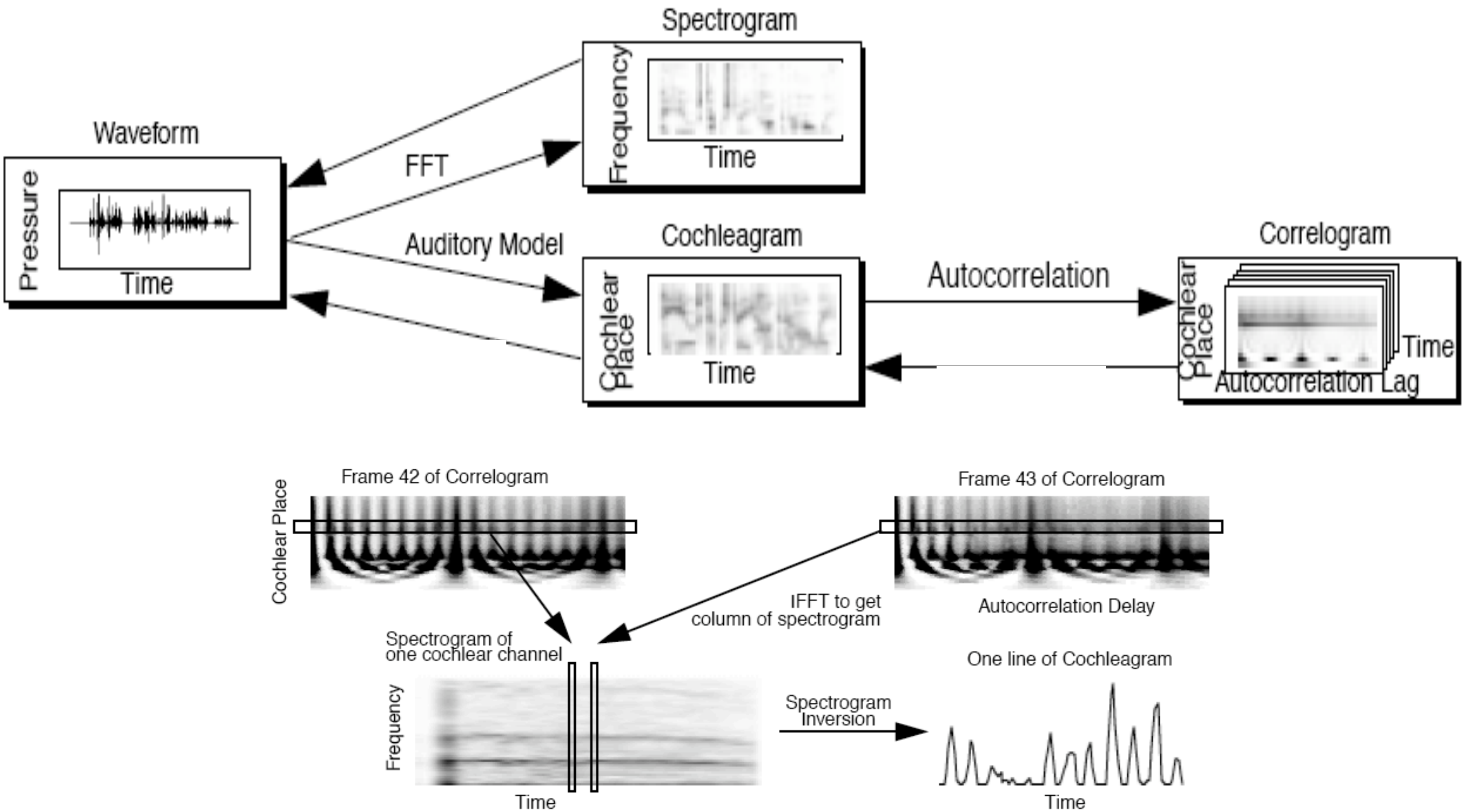
$$Y_w^i(m,w) = \left|X_d(m,w)\right| \frac{X_e^i(m,w)}{\left|X_e^i(m,w)\right|}$$

$$X_e^{i+1}(n) = \frac{\sum_{m=-\infty}^{\infty} w(m-n)\frac{1}{2\pi}\int_{-\pi}^{\pi} Y^i(m,w)e^{jwn}dw}{\sum_{m=-\infty}^{\infty} w^2(m-n)}$$



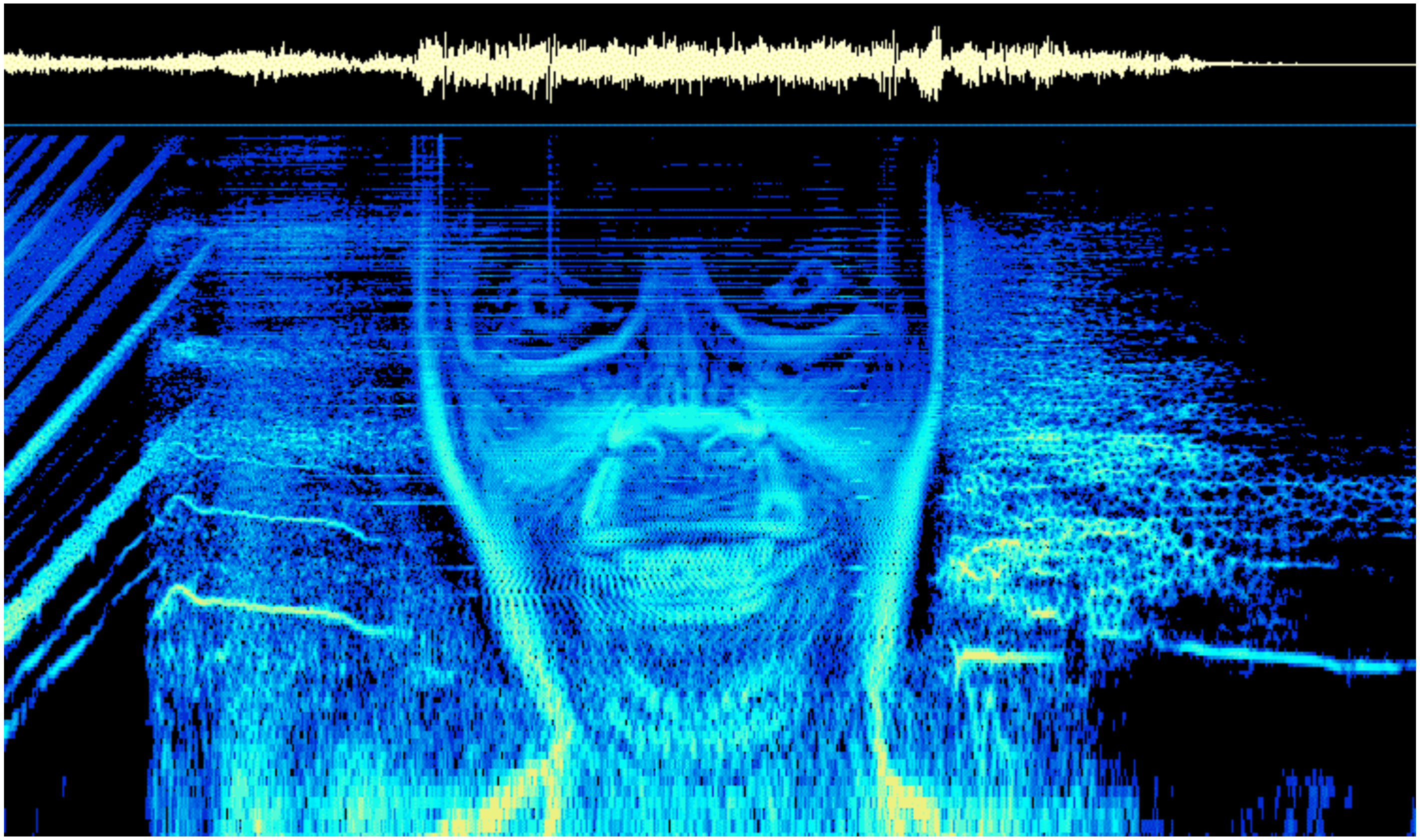

DFT
Amplitude
Frequency
Discrete
Fourier Transform
Signal
Amplitude
Time
Short-Time
Fourier Transform
STFT
Frequency
Time
Frequency
Time
Multiple
Short-Time
Fourier Transform
MSTFT
Frequency
Time

Inversion from Auditory Representations
Malcolm Slaney, IEEE SMC Conference, 1995



Aphex Twin's tracks, #2 (the long formula) on "Windowlicker"

5:27 mark and lasting for about 10 seconds



http://www.bastwood.com/aphex.php

SDIFF

- Sound Description Interchange File Format
- A way to share analysis results
- A way to facilitate synthesis after complex analysis
- Used as a performance tool in computer music

http://www.cnmat.berkeley.edu/SDIF/

http://recherche.ircam.fr/equipes/analyse-synthese/sdif/
includes SDIF Extension for Matlab

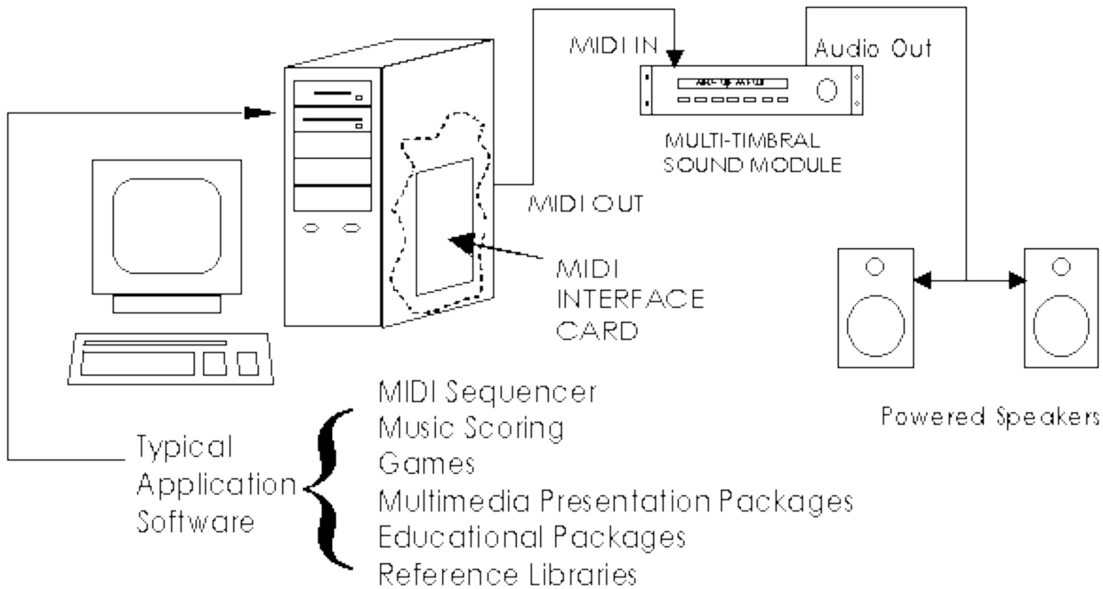
FrameTypeID	char[4]	A unique code indicating what kind of frame this is
FrameDataSize	int32	The size, in bytes, of the frame,.
Data		anything, as long as the size is a multiple of 8 bytes.

Frame Type ID	Frame Type	Columns of Main Matrix
1FQ0	Fundamental Frequency Estimates	Fundamental frequency, confidence
1STF	Discrete Short-Term Fourier Transform	Real & imaginary bin values
1PIC	Picked Spectral Peaks	Freq, Amp, phase, confidence
1TRC	Sinusoidal Tracks	Index, freq, amp, phase
1HRM	Pseudo-harmonic Sinusoidal Tracks	Harmonic partial #, freq, amp, phase
1RES	Resonances	Freq, amp, decay rate, phase
1TDS	Time Domain Samples	Channels of sample data

Matrix type: "1TRC"
Allowed MatrixDataTypes: float32, float64
Rows: Sinusoidal tracks
Columns

- Index (a unique integer >= 1) allowing it to be matched with 1TRC data in other frames.
- Frequency (Hertz).
- Amplitude (linear). Optional; default is 1.0.
- Phase (Radians: must be between 0 and 2*pi). Optional,.

Synthesis and MIDI



- Synthesis
  - Mathematical, signal modeling or sampling methods for generation of sounds
  - Mostly simulate musical instruments
- Musical Instruments Digital Interface (MIDI)
  - standard for communication between synthesizers
  - Music is represented in terms of performance actions: which notes are played, when and how

MIDI Explained

- MIDI message is made up of an eight-bit status byte which is generally followed by one or two data bytes.
- Consists of Channel and System Messages
- Channel Messages: Note On, Note Off, Aftertouch, Pitch Bend, Program Change, and Control Change
- System messages are used for setup and synchronization between synthesizers
- MIDI Files
  – Sequences of MIDI instructions can be stored in a MIDI file
  – Popular today for ring-tones

See also http://www.harmony-central.com/MIDI/Doc/tutorial.html

MIDI in Matlab

Midi Toolbox http://www.jyu.fi/musica/miditoolbox/

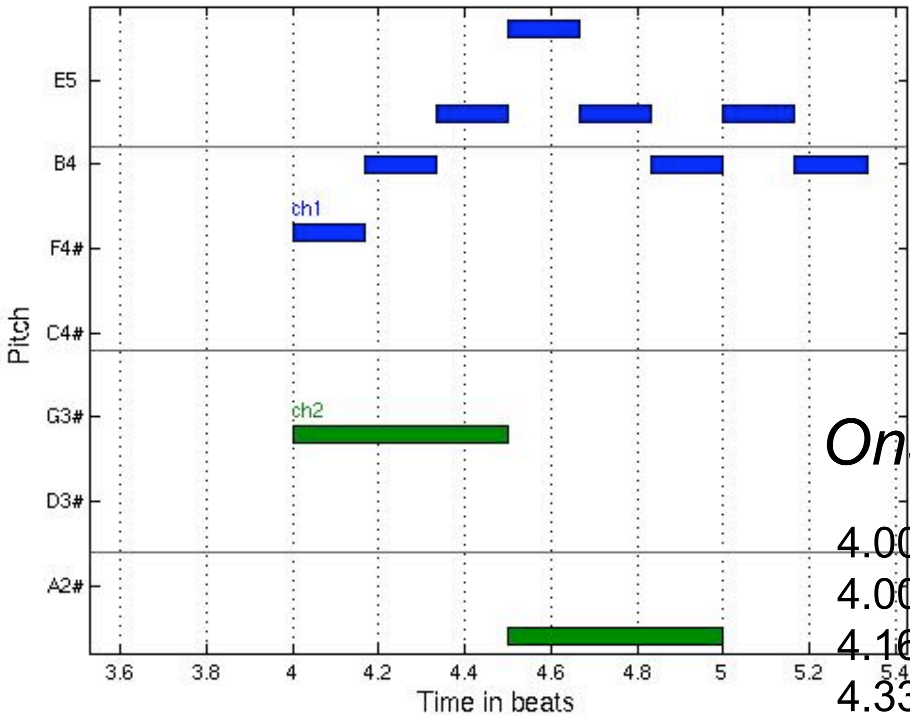
- Reading midi file into a matrix

nmat = readmidi(filename);

writemidi(nmat, filename, <tpq>, <tempo>, <tsig1>, <tsig2>)

- Also contains cognitively inspired analytic techniques for context-dependent musical analysis
  - melodic contour, similarity, key-finding, meter-finding and segmentation

Piano roll and note matrix example




```
nmat = readmidi('wtc1151.mid');
pianoroll(nmat(1:10,:))
```

Onset	*Dur.*	*Chan.*	*Note*	*Vel.*
4.0000	0.1667	1.0000	67.0000	64.0000
4.0000	0.5000	2.0000	55.0000	64.0000
4.1667	0.1667	1.0000	71.0000	64.0000
4.3333	0.1667	1.0000	74.0000	64.0000
4.5000	0.1667	1.0000	79.0000	64.0000
4.5000	0.5000	2.0000	43.0000	64.0000
4.6667	0.1667	1.0000	74.0000	64.0000
4.8333	0.1667	1.0000	71.0000	64.0000
5.0000	0.1667	1.0000	74.0000	64.0000
5.1667	0.1667	1.0000	71.0000	64.0000

To plot only onsets

```
plot(nmat(:,1),nmat(:,4),'x')
```

Why MIDI?

- VERY compact music representation (only few kbps)
- Symbolic representation of musical “content”
  - Intuitive music access and manipulation
- Many interesting questions can be posed about the relations between Audio and MIDI signals
  - Score Transcription from Audio
  - Audio and Score Alignment
  - Score facilitated Audio Processing
  - Analysis of Audio and MIDI contents
- A lot of data
  - Almost all classical music and many popular music are available as MIDI files, such as http://www.classicalarchives.com
- New possibilities using Structured Audio hybrid representations