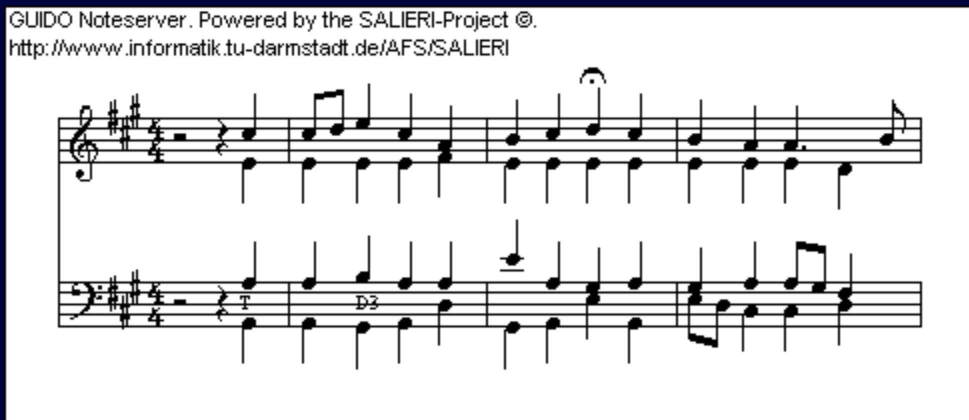
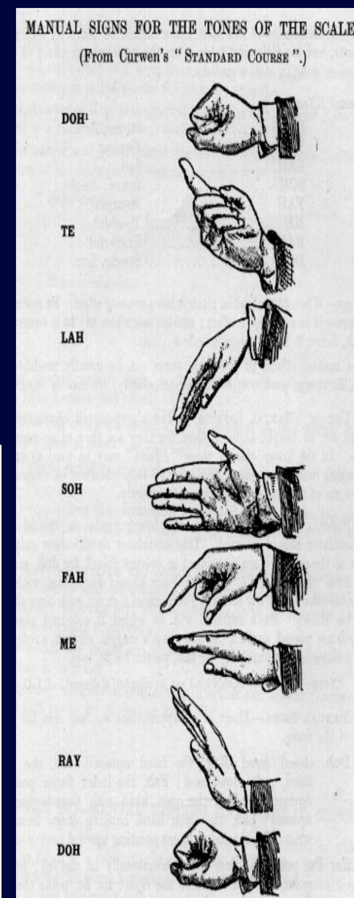
Pitch-based representations, analysis and applications

George Tzanetakis (gtzan@cs.uvic.ca)
Associate Professor
Canada Research Chair (Tier II)
Computer Science Department
(also in Music, ECE)
University of Victoria, Canada

Traditional Music Representations





Pitch content

- Harmony, melody = pitch concepts
- Music Theory — Score = Music
- Bridge to symbolic MIR
- Automatic music transcription
- Non-transcriptive arguments

Split the octave to discrete logarithmically spaced intervals

MIDI

- Musical Instrument Digital Interfaces
  - Hardware interface
  - File Format
- Note events
  - Duration, discrete pitch, "instrument"
- Extensions
  - General MIDI
  - Notation, OMR, continuous pitch

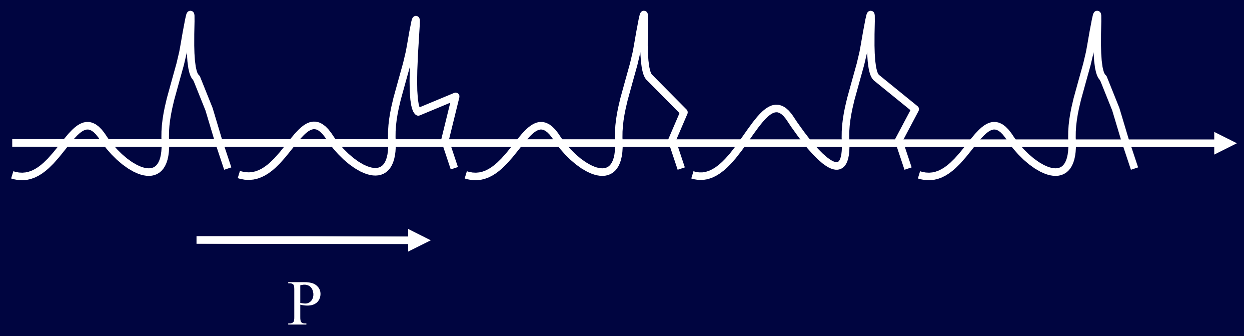
Representations

- Score
  - Discrete, high level abstraction, explicit structure, no performance info
- MIDI
  - Discrete, medium level of abstraction, explicit time but less structure, targeted to keyboard performance
- Audio
  - Continuous, low level abstraction, timing and structure implicit

Psychoacoustics

- Scientific study of sound perception
- Frequently limits of perception
  - Range (20Hz – 20000Hz)
  - Intensity (0dB-120dB)
  - Masking
  - Missing fundamental (2xf, 3xf, 4xf) give humans the impression of 1xf pitch

Pitch Detection



Time-domain
Frequency-domain
Perceptual

Pitch is a PERCEPTUAL attribute correlated but not equivalent to fundamental frequency

Rhythm -> ~20 Hz Pitch

(courtesy of R.Dannenberg – Nyquist)

Pitch Perception I

- Pitch is not just fundamental frequency
- Periodicity or harmonicity or both ?
- Human judgements (adjust sine method)
- 1924 Fletcher – harmonic partials missing fundamental (pitch is still heard)
  - Examples: phone, small radio
- Terhardt (1972), Licklider (1959)
  - duplex theory of pitch (virtual & spectral pitch)

Pitch Perception II

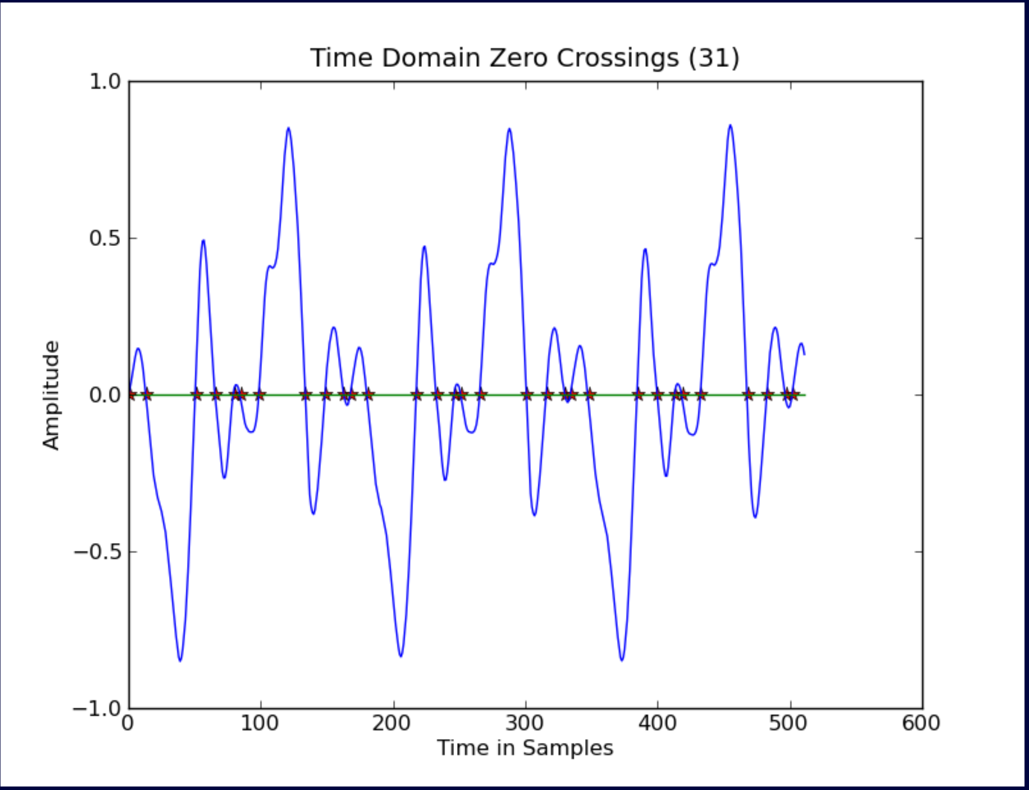
- One perception – two overlapping mechanisms
  - Counting cycles of period < 800Hz
  - Place of excitation along basilar membrane > 1600 Hz

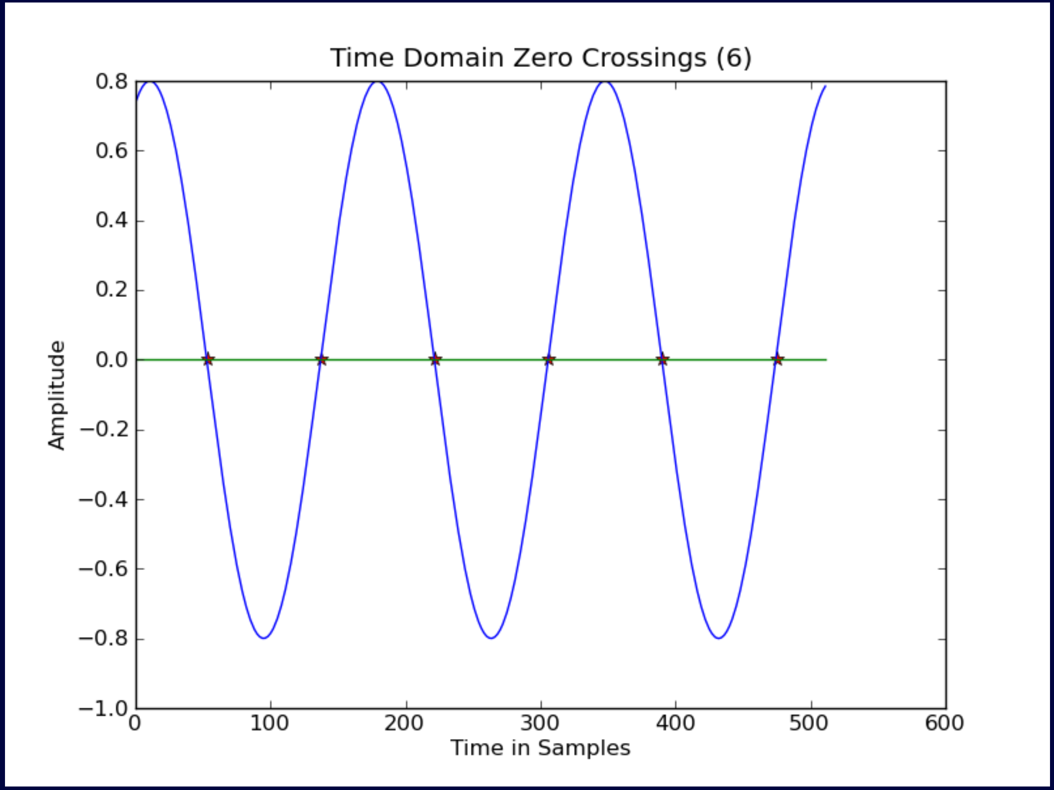
FFT → Pick sinusoids → weighting / masking → candidate generation

most likely pitch ← common divisors

Time Domain

zero-crossings sensitive to noise – needs LPF



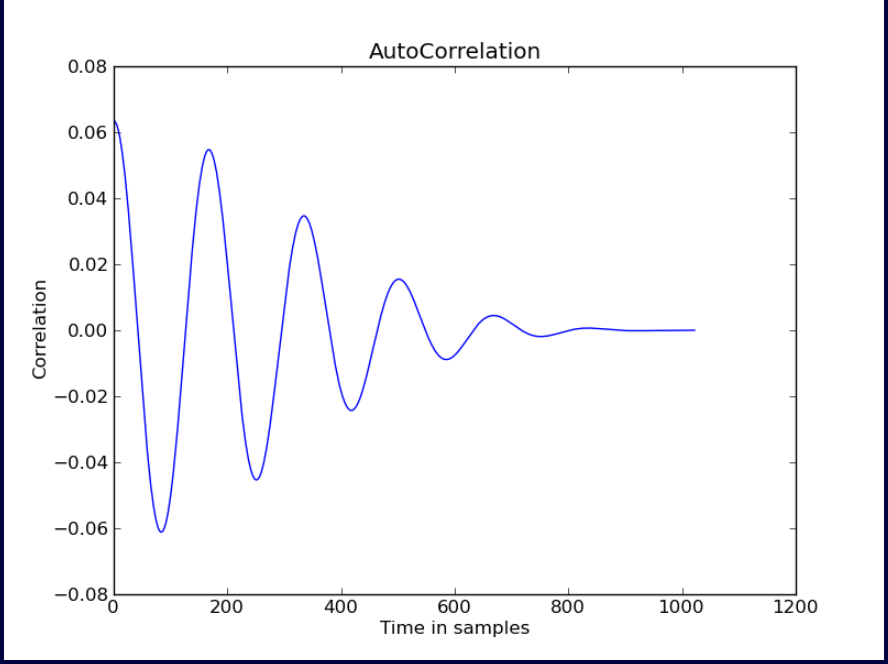


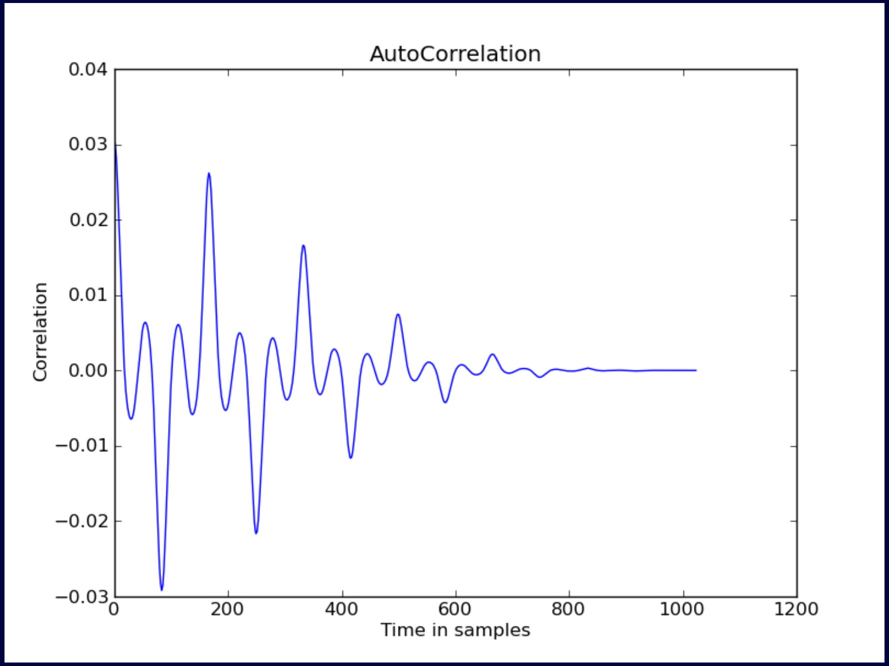
C4 Clarinet Note  C4 Sine Wave 

AutoCorrelation

$$r_x = \sum_{n=0}^{N-1} x(n)\,x(n+l),\ l=0,1,..\,L-1$$

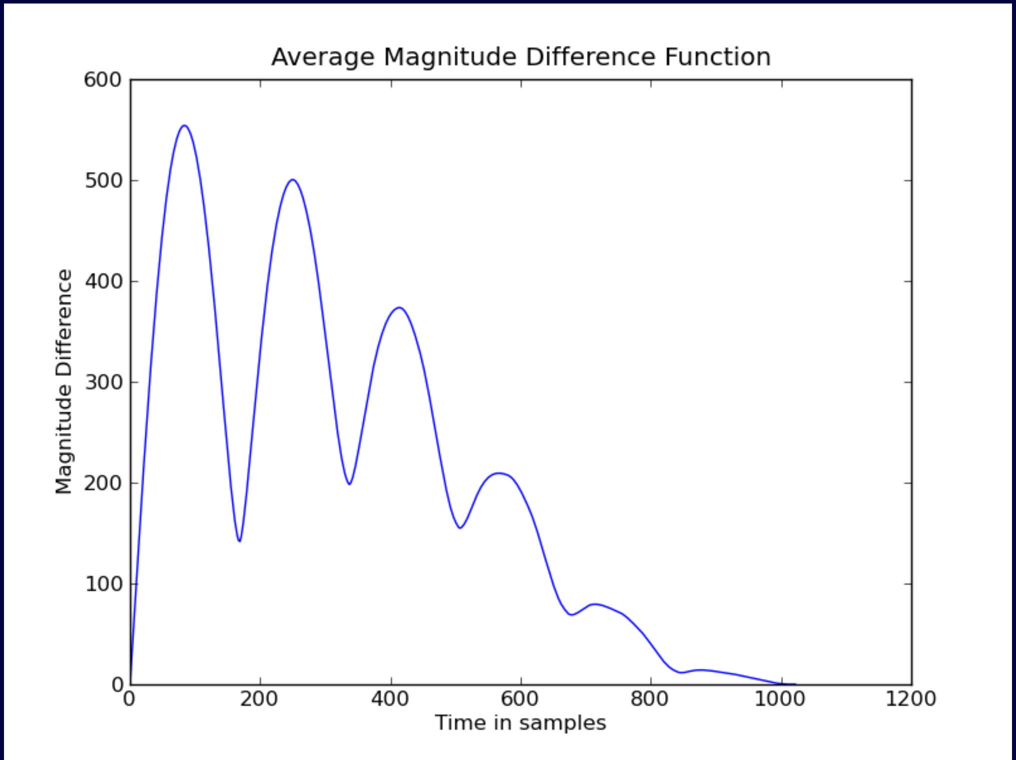
F(f) = FFT(X(t))
S(f) = F(f) F*(f)
R(l) = IFFT(S(f))

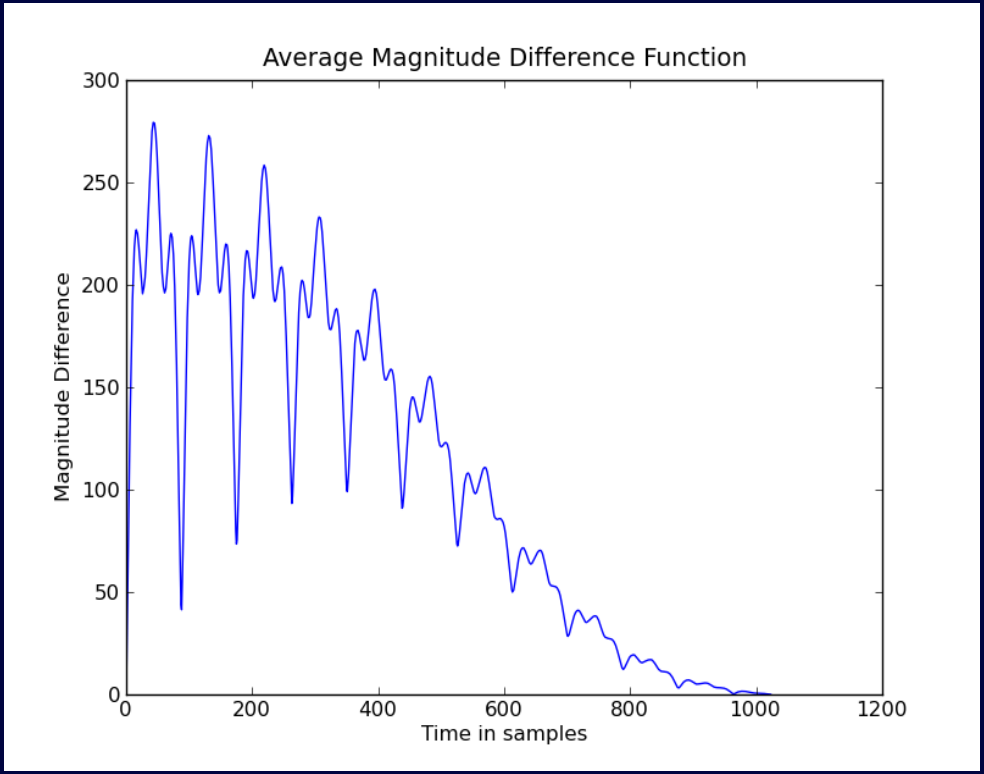




Efficient computation possible for powers of 2 using FFT

Average Magnitude Difference Function



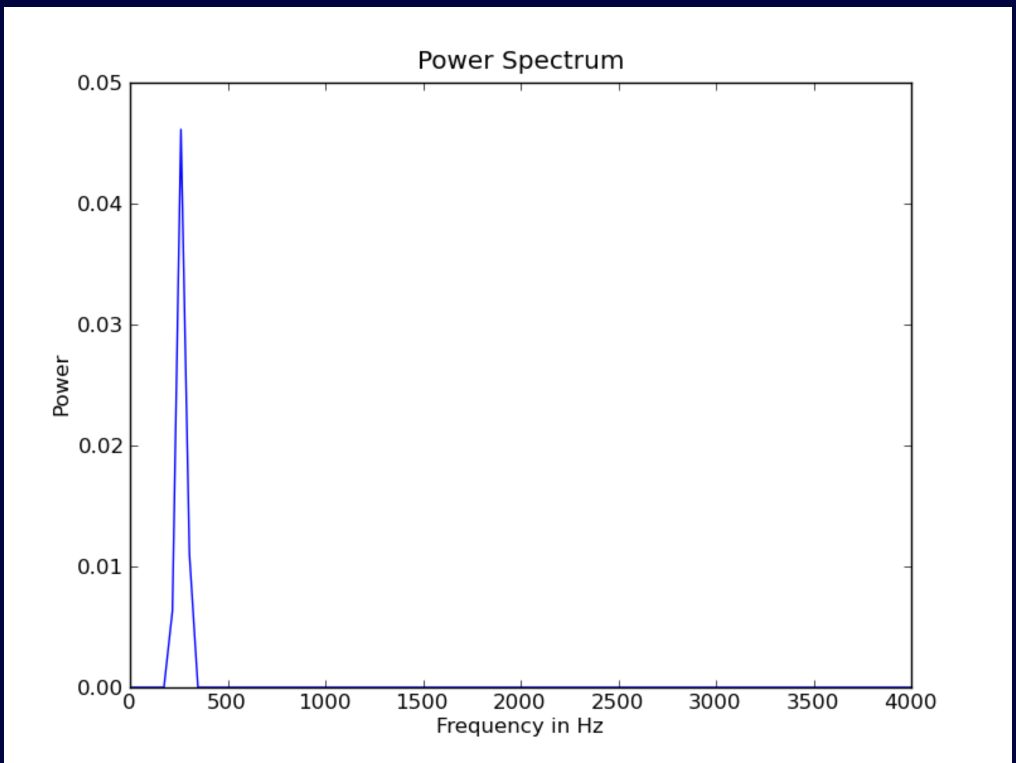


No multiples –
more efficient for fixed point

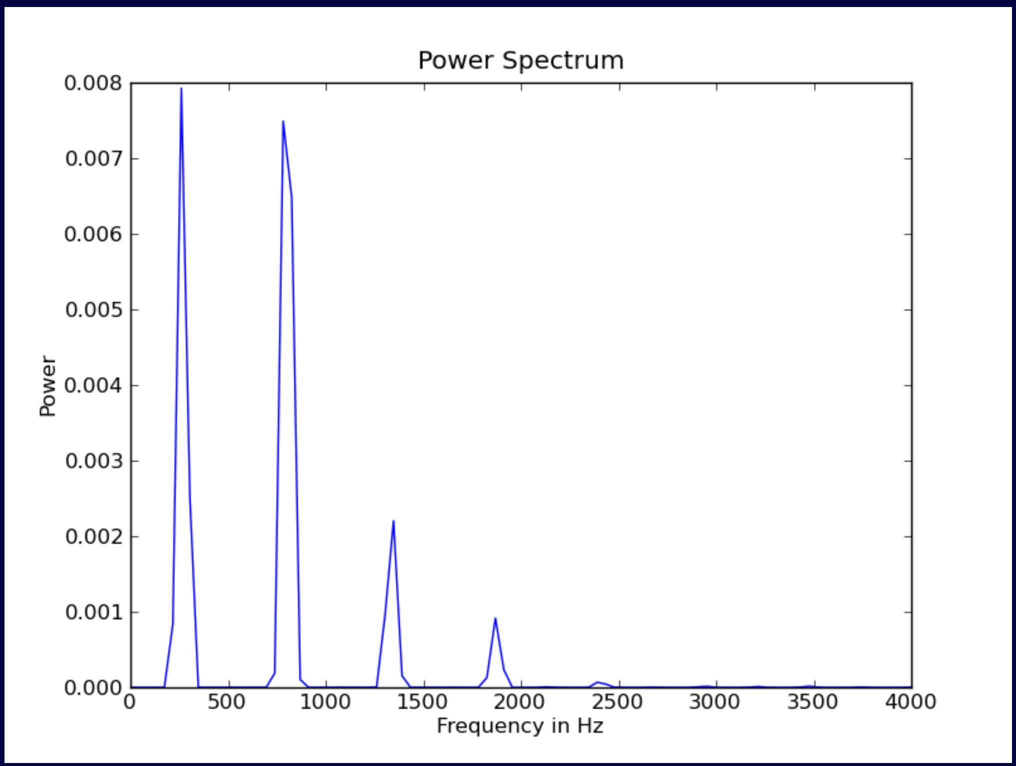
$$AMDF(m) = \sum_{i=q}^{q+N-1} \left| x(i) - x(i+m) \right|^k$$

Frequency Domain

Sine C4



Clarinet C4

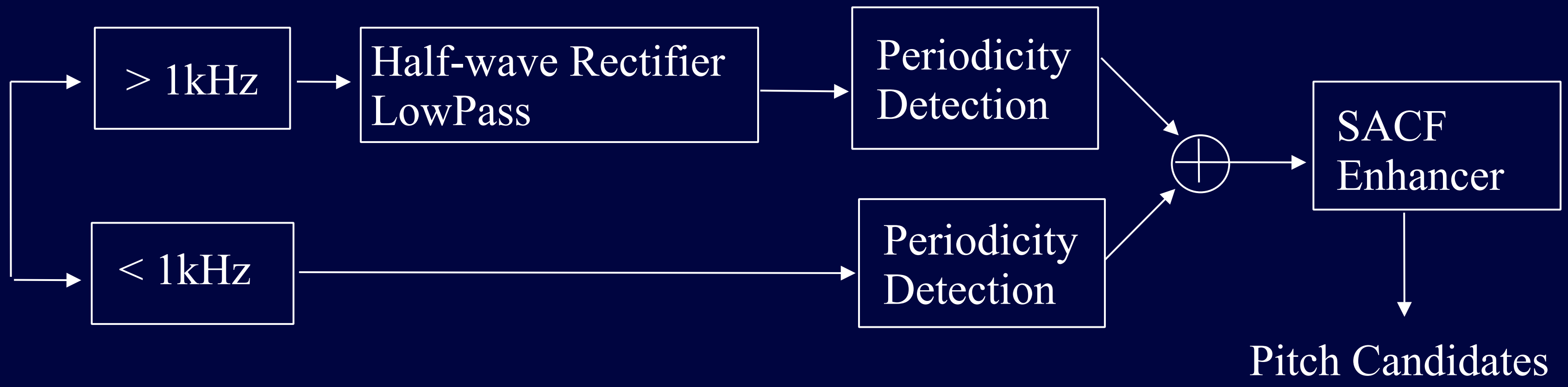


Fundamental frequency (as well as pitch) will correspond to peaks in the Spectrum. The fundamental does not necessarily have the highest amplitude.

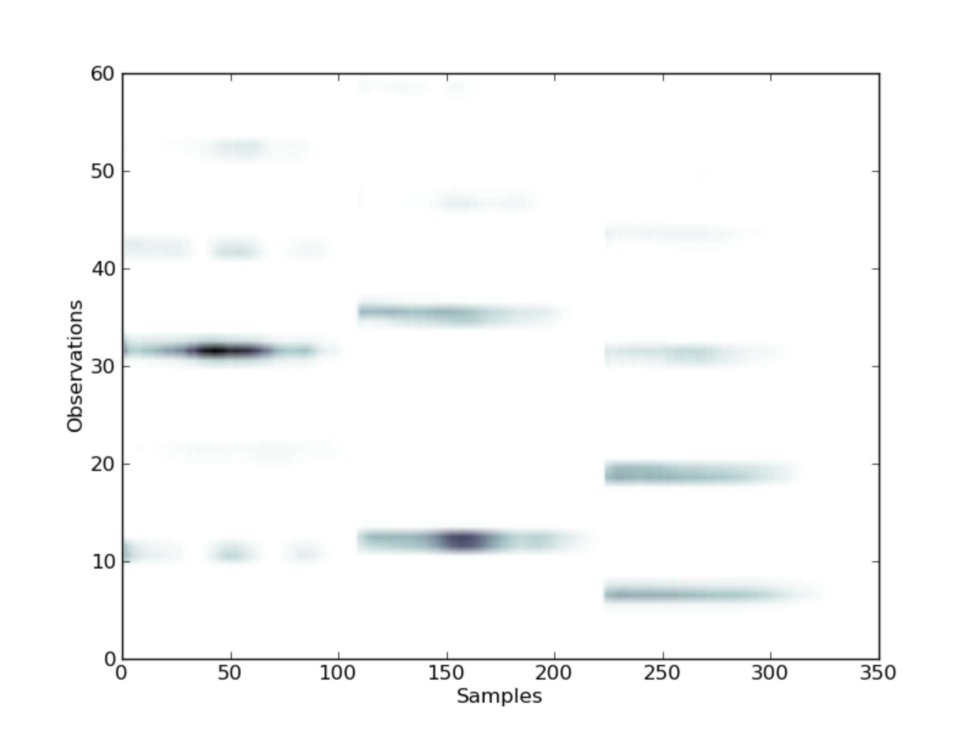
Multiple Pitch Detection

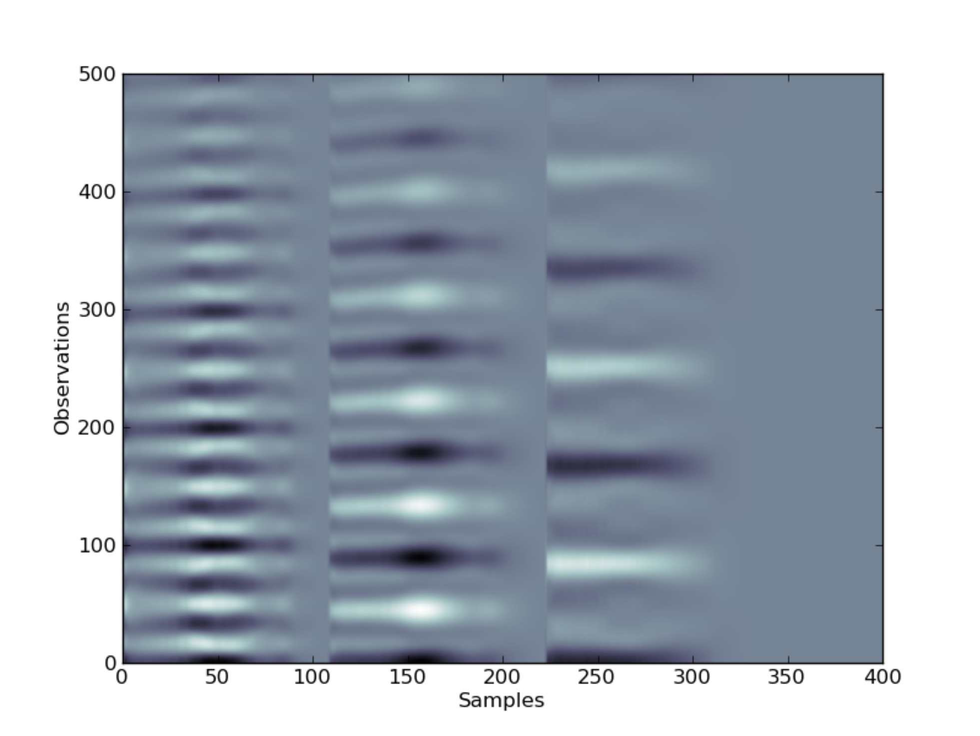
Tolonen and Karjalainen, TSAP00

Tzanetakis et al, ISMIR 01



Plotting over Time

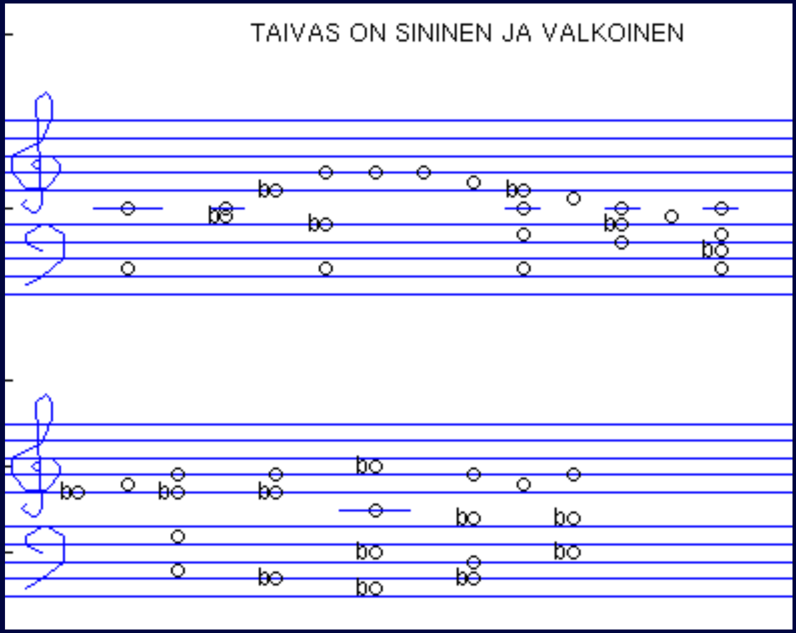




A4, B4, C4

Polyphonic Transcription

Klapuri et al, DAFX 00



Original

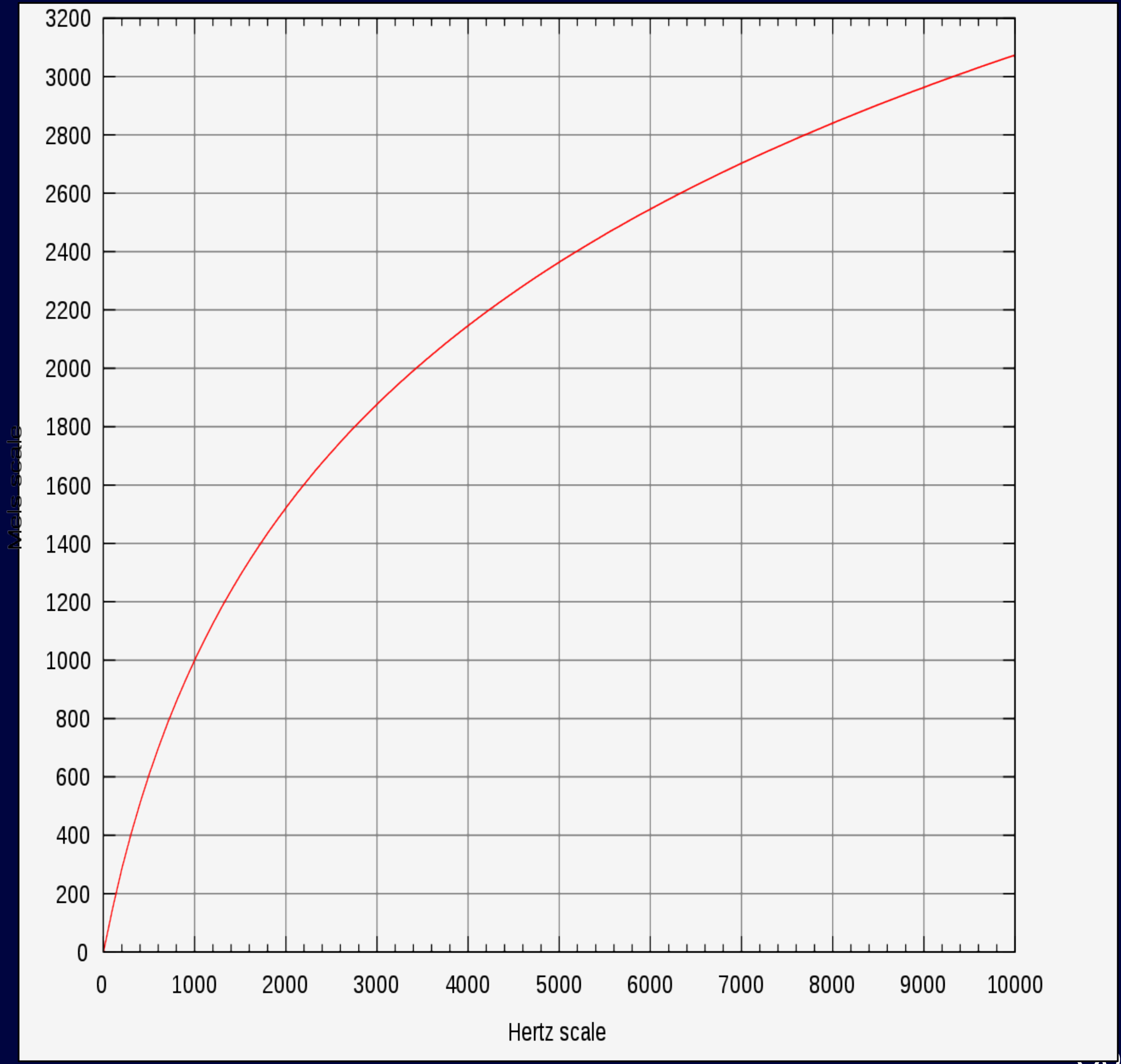
Transcribed

Mixture signal → Noise Suppression → Predominant pitch estimation → Remove detected sound → Estimate # voices iterate → Predominant pitch estimation

Perceptual Pitch Scales

- Perception of frequency
- Various perceptual scales based on JND
  - Typically linear below a break frequency F_b and logarithmic above
- Popular choices for break frequency (1000, 700, 625, 228)
- What the hell is a mel ?

Mel Mapping



3200
3000
2800
2600
2400
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
Mels scale
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Hertz scale

Musical Pitch

- Tuning = different ways of subdividing the octave logarithmically (as ratios) into intervals
- Tension between harmonic ratios, modulation to different keys, regularity, pure fifths (ratio of 1.5 or 3:2)

> Many tuning systems have been explored through history

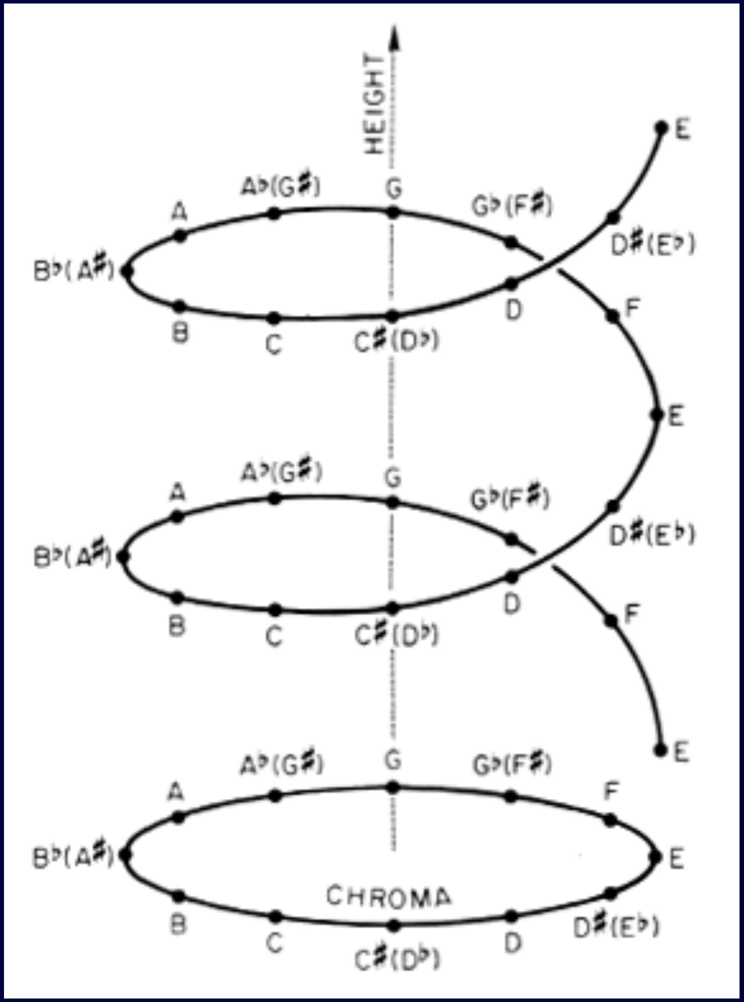
Tuning systems

- Just intonation (1:1, 9:8, 5:4, 4:3, 3:2, 5:3, 15:8, 2:1)
- Pythagorean tuning all notes derives from 3:2 (1:1, 256:243, 9:8,…)
- Equal temperament
  - All notes spaced by logarithmically equal distances (100 cents). Each step is higher by $2^{1/12}$ (1.0594) from previous.

Notation

- A, B, C, D, E, F, G
- Number indicate octave
  - A4 is 440Hz and MIDI number 69
- Do, Re, Mi, Fa, Sol, La, Ti
- MIDI (0-128)
- $m = 69 + 12 \log_2(f/440)$

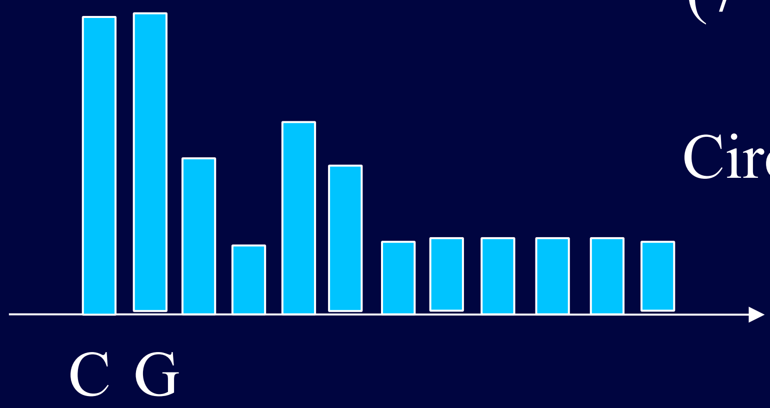
Pitch Helix



Linear pitch (i.e log(frequency) is wrapped around a cylinder – in order to model the octave equivalence.

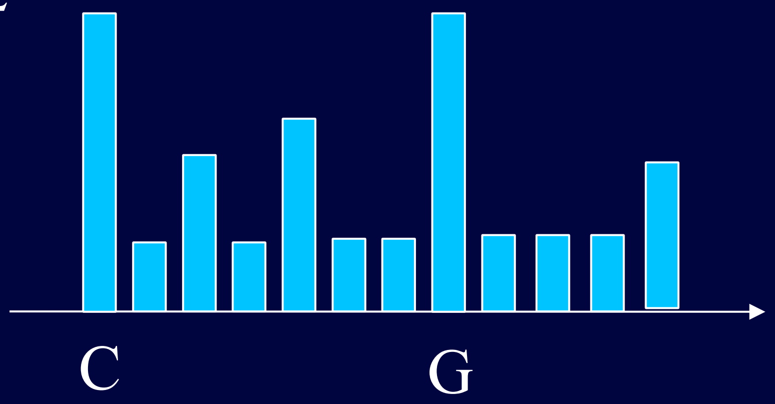
Pitch perception has two dimensions:
Height: naturally organizes pitches from low to high
Chroma: represents the inherent circularity of pitch

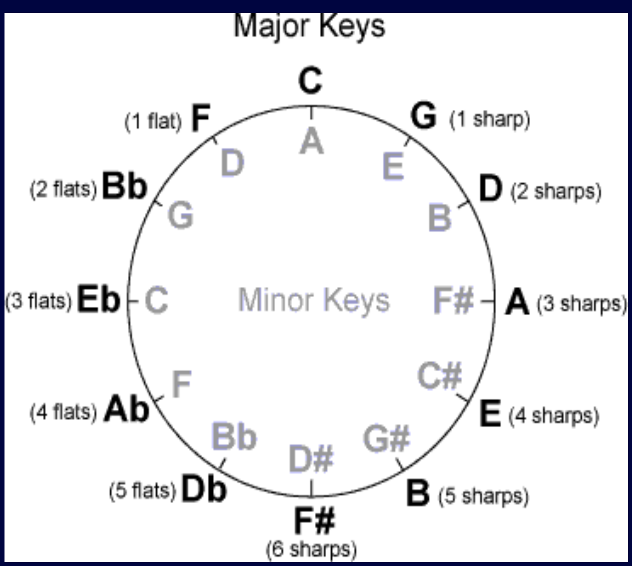
Pitch Histograms

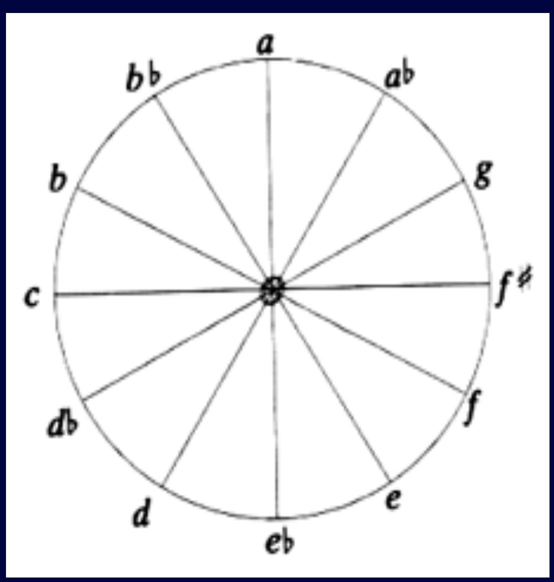


(7 * c) mod 12

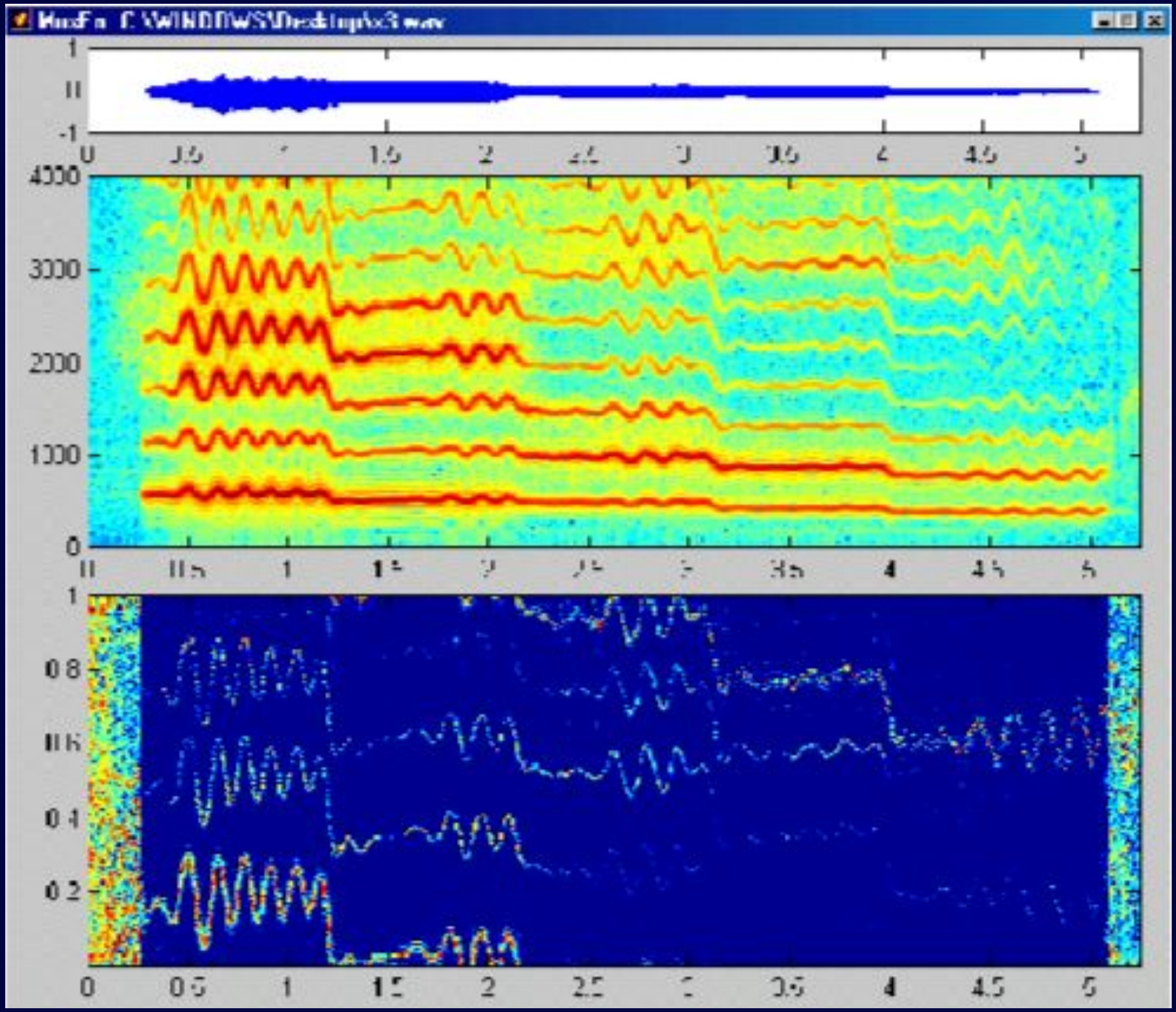
Circle of 5s







Chroma



Calculating Pitch Profiles – Warping

- Calculate FFT of a signal segment
- Map each FFT bin to Hertz
  - 512 time domain samples -> 256 FFT bins @ 22050 Hz. Each bin will be 11025/256 ~= 43 Hz
  - f = k * (srate / fft_size)
- Map each bin (in Hertz) to MIDI:
  - $m = 69 + 12 \log_2(f/440)$

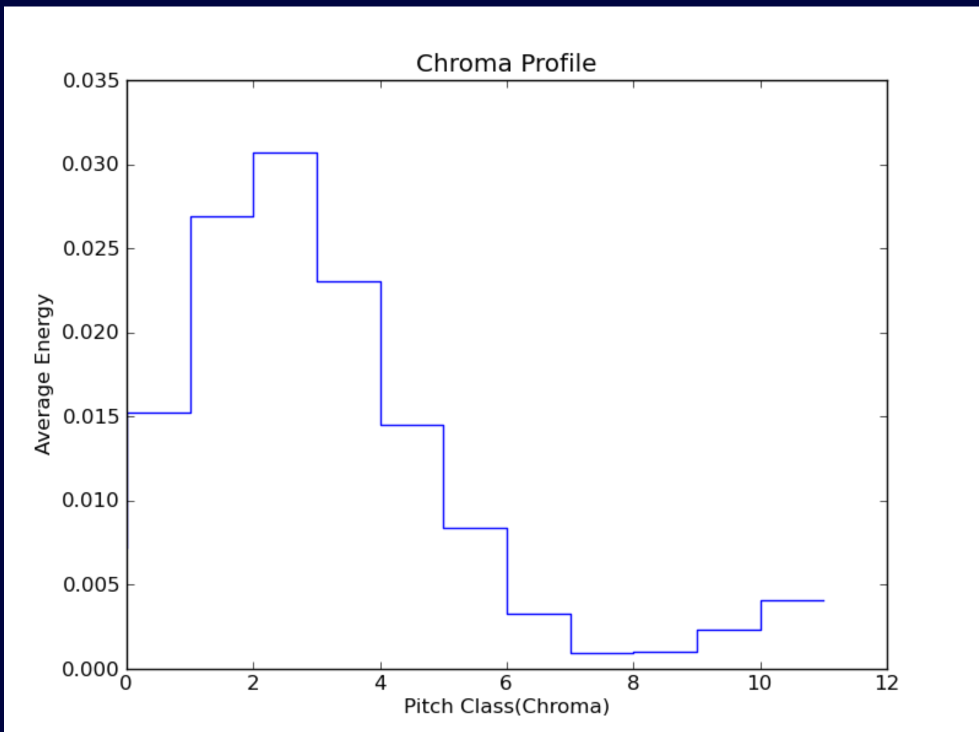
Pitch Histogram

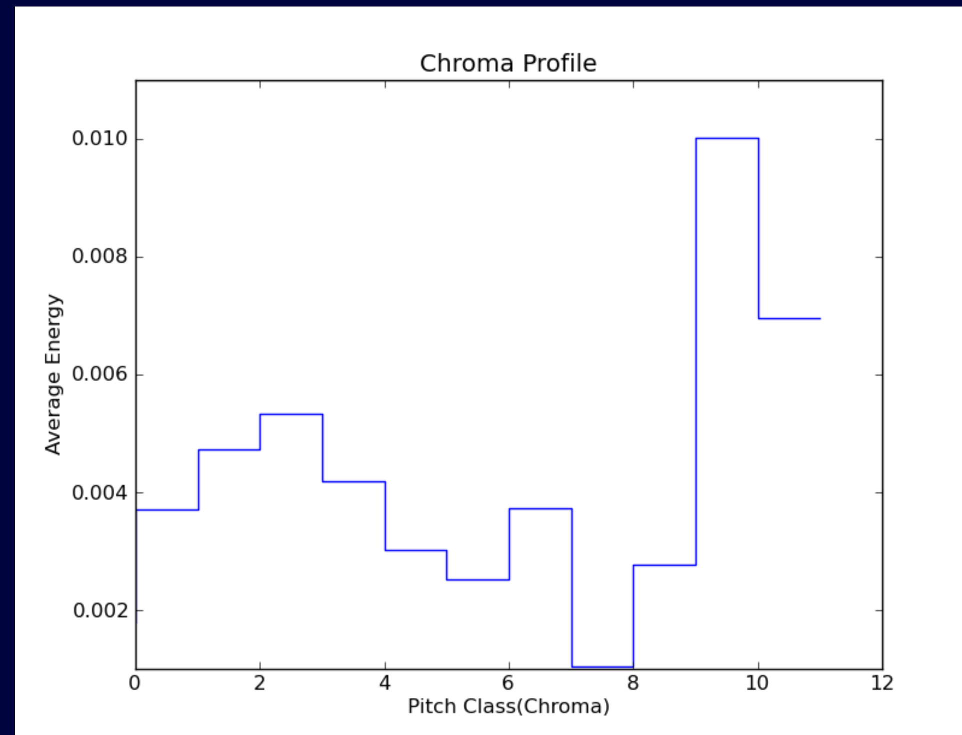
- Average amplitudes of bins mapping to the same MIDI note number

(different averaging shapes can be used)

- If desired fold the resulting histogram, collapsing bins that belong to the same pitch class into one
- Frequently more than 12 bins per octave to account for tuning/performance variations

Chroma Profiles



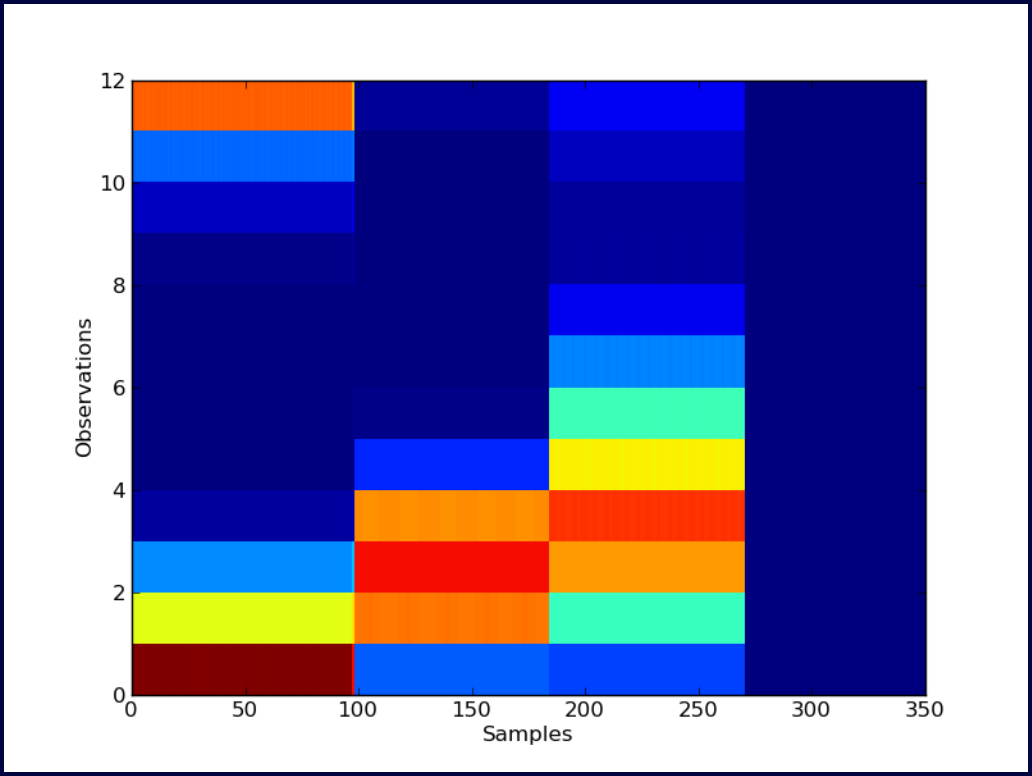


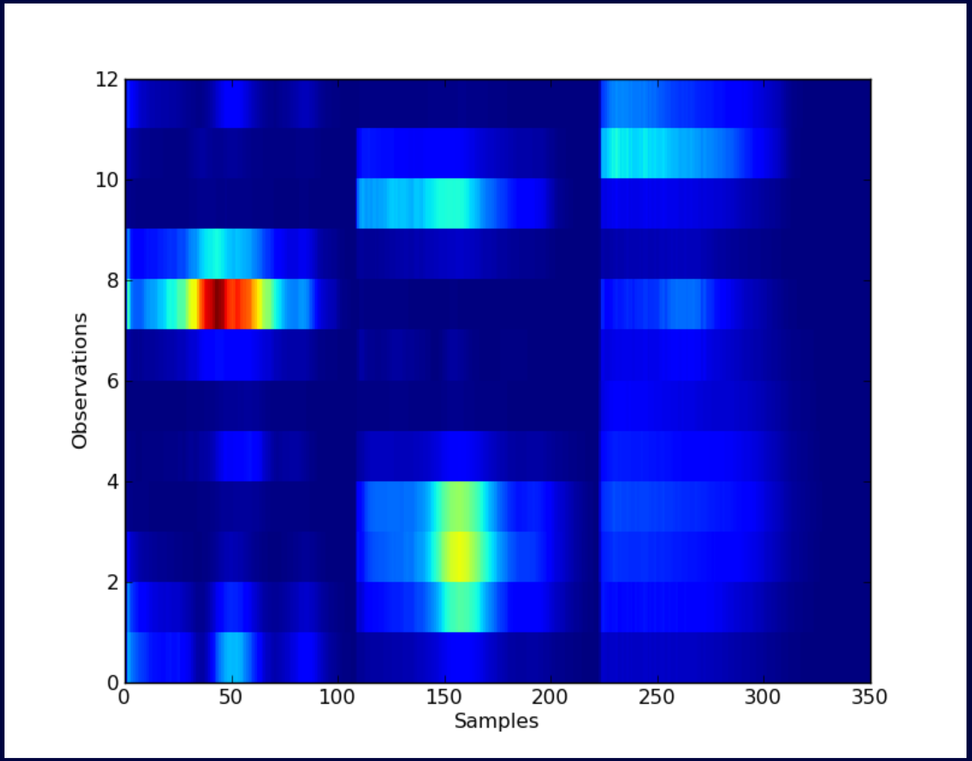
Sine C4

Clarinet C4

0 bin is A and spacing is chromatic

Chromagrams

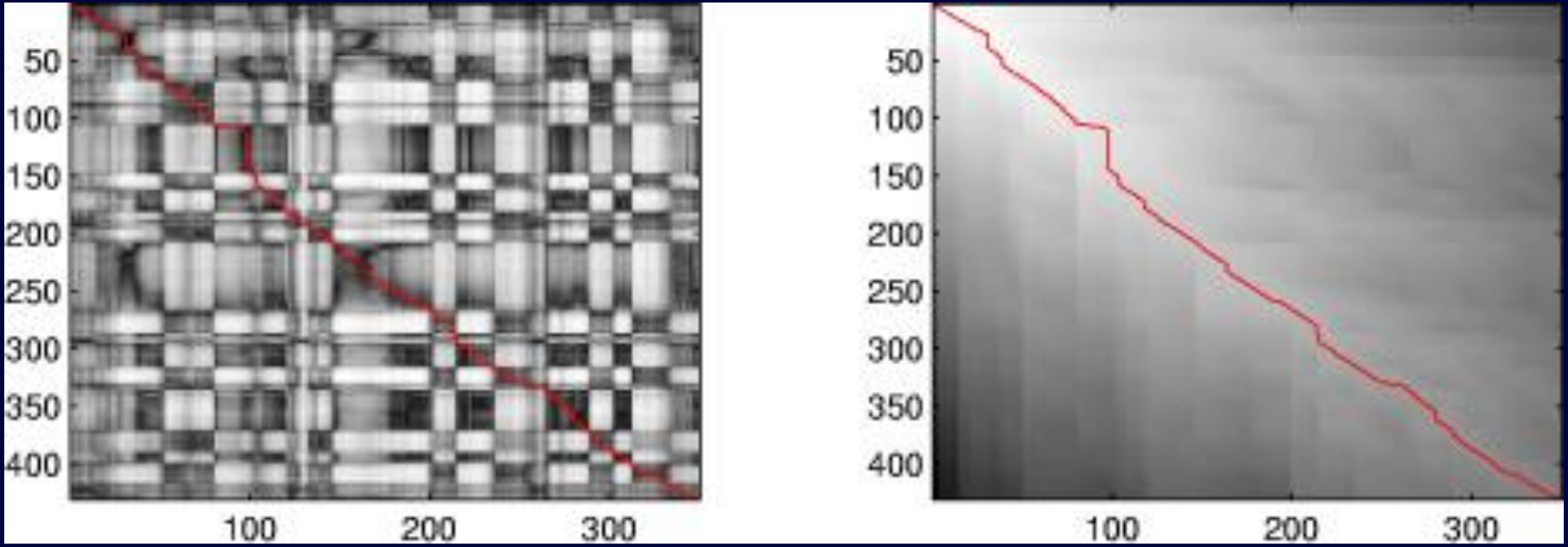




Time Alignment

- ARTHUR (Foote 2000)
- Two sequences of energy contours corresponding to two performances of the same symphony
- We are given two pitch sequences of the same melody sung by different singers
- How can we find if they match ?

Dynamic Time Warping



POLYPHONIC AUDIO AND MIDI ALIGNMENT

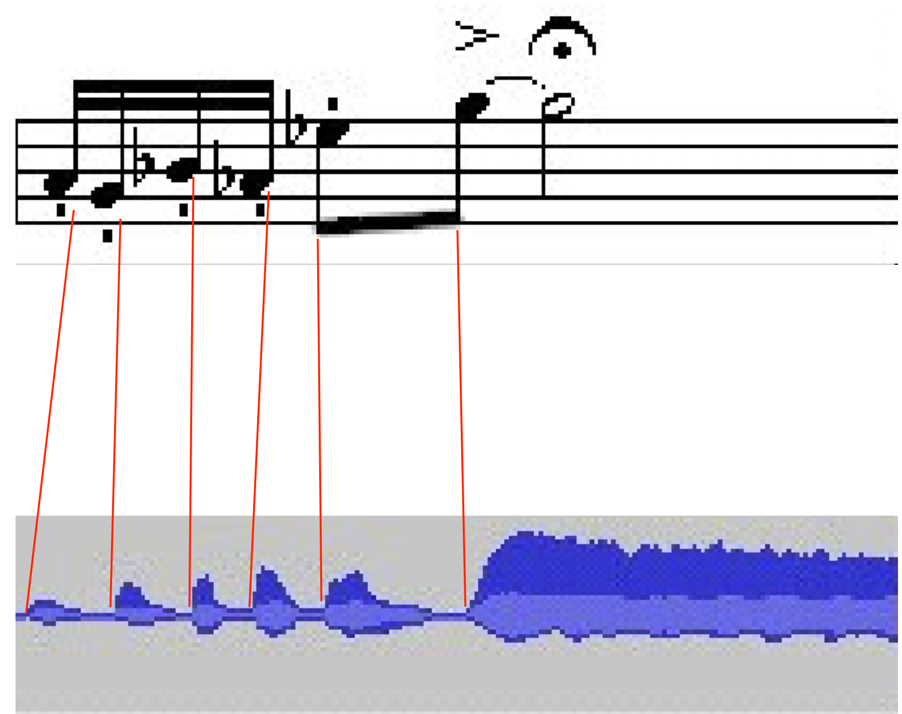
Align

Symbolic Representation

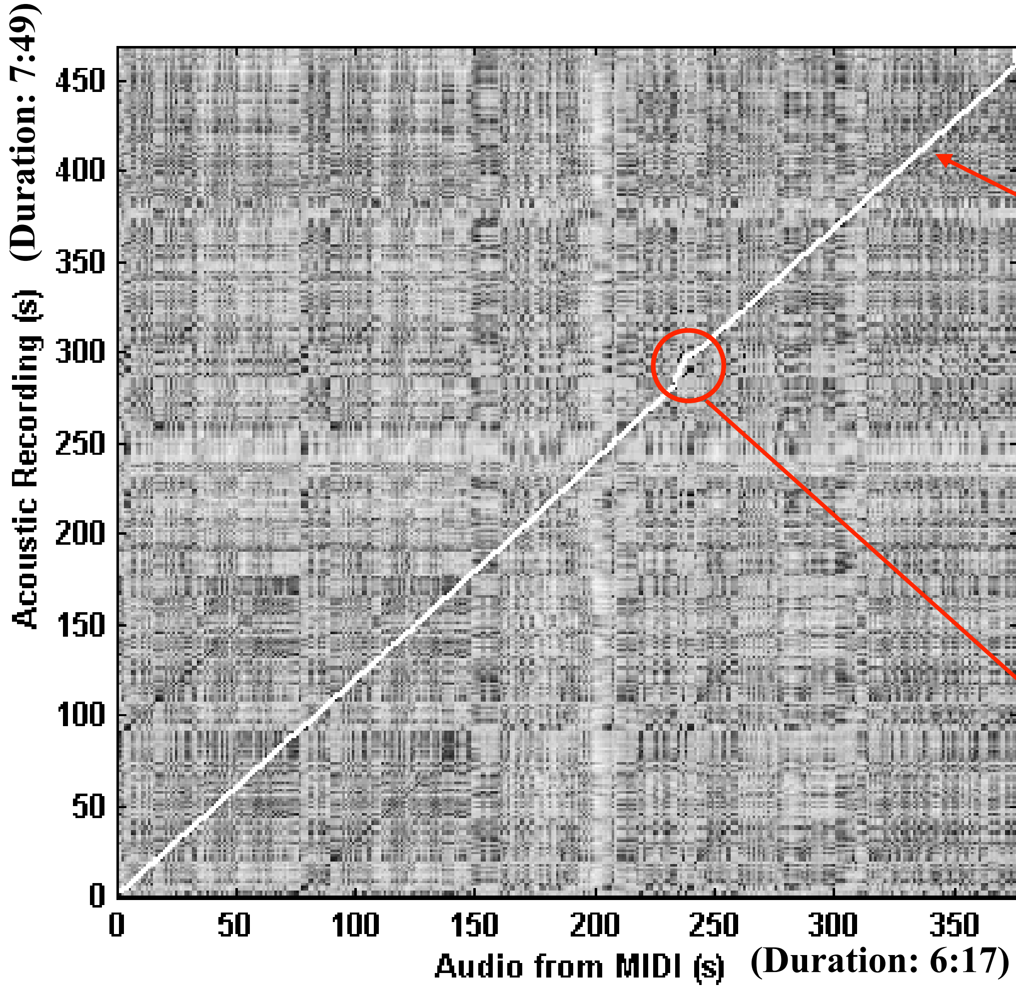
- easy to manipulate
- "flat" performance

Audio Representation

- expressive performance
- opaque & unstructured



POLYPHONIC AUDIO AND MIDI ALIGNMENT



Optimal Alignment Path

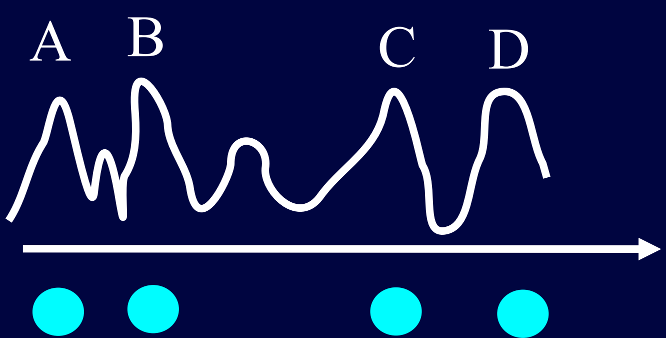
Oboe solo:

- Acoustic Recording
- Audio from MIDI

Similarity Matrix for Beethoven's 5th Symphony, first movement

Performance matching

Yang, WASPAA 99



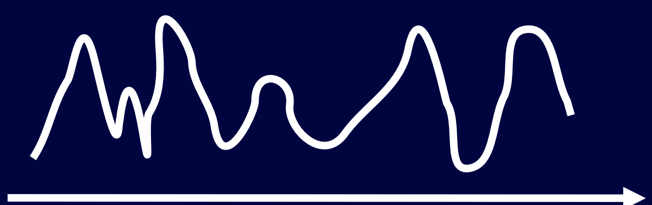
Power plot

24d "pitch"
vectors from FFT

Nearest neighbor with
Locality-Sensitive Hashing
Identical, different copy,
different vocals,
different performance (80%)

Characteristic sequence

Foote, ISMIR 00

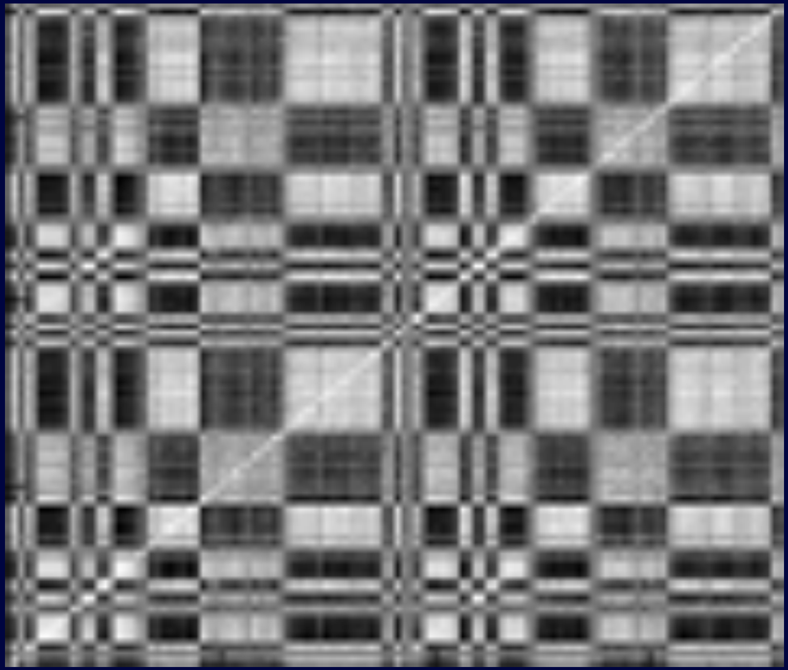


Dynamic programming
Symphonies

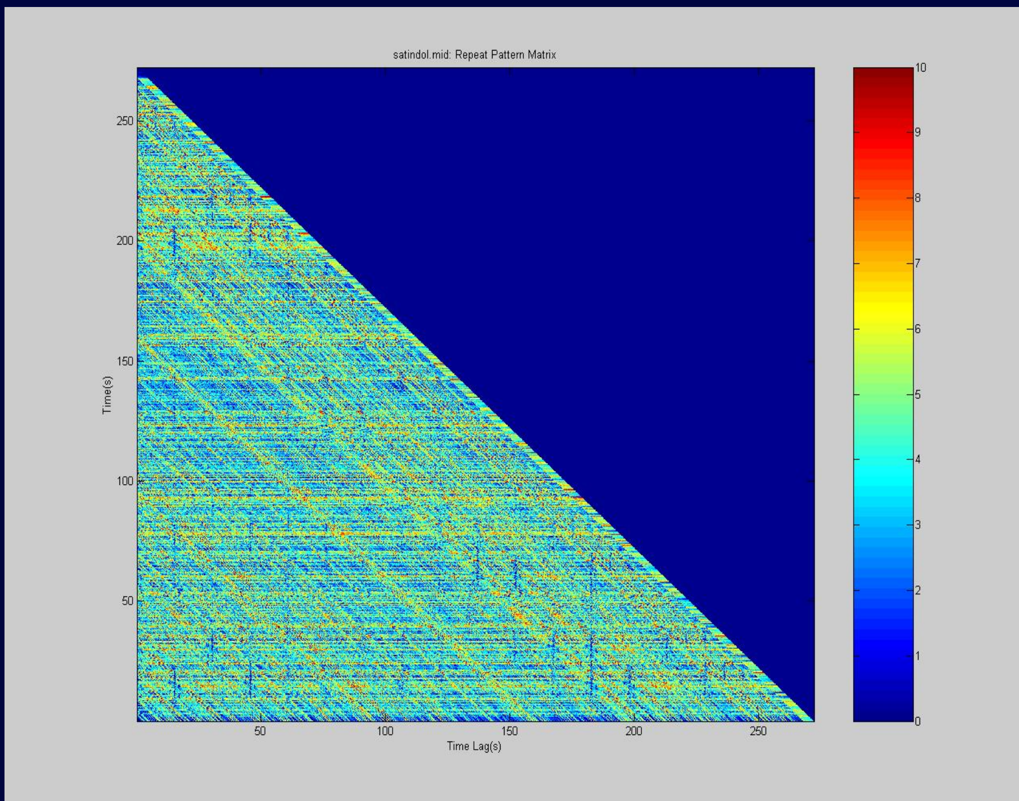
Structural Analysis

Dannenberg & Hu, ISMIR 2002
Tzanetakis, Dannenberg & Hu, WIAMIS 03

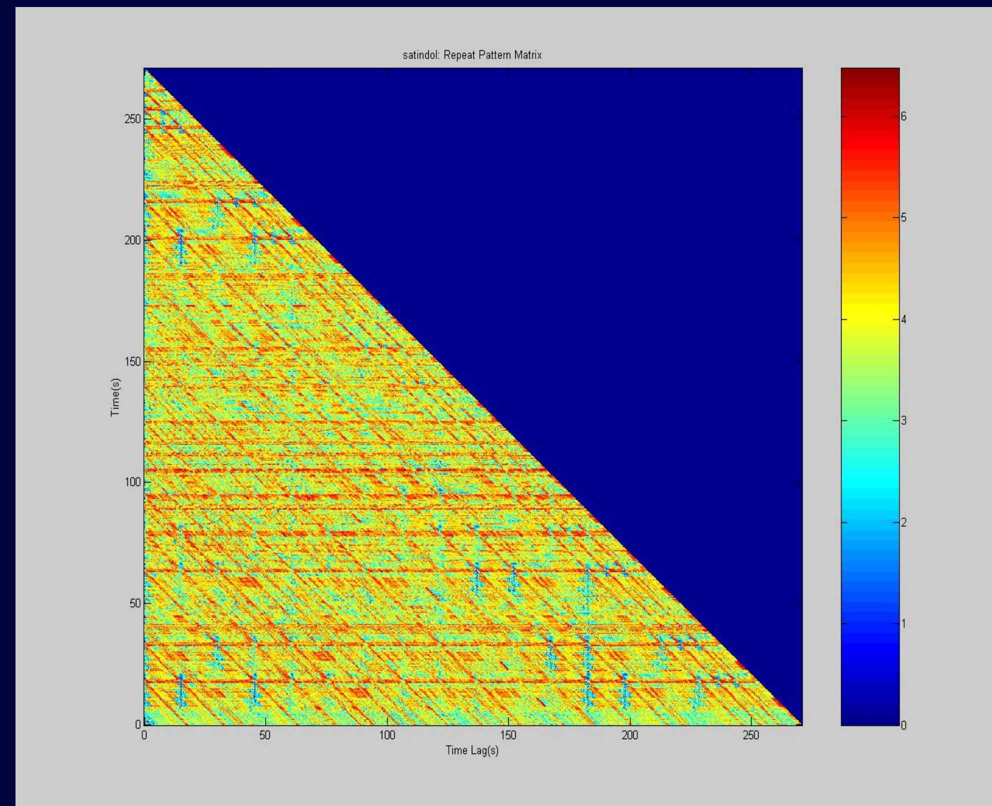
- Similarity matrix
- Representations
  - Notes
  - Chords
  - Chroma
- Greedy hill-climbing algorithm
  - Recognize repeated patterns
- Result = AABA (explanation)



Similarity Matrices

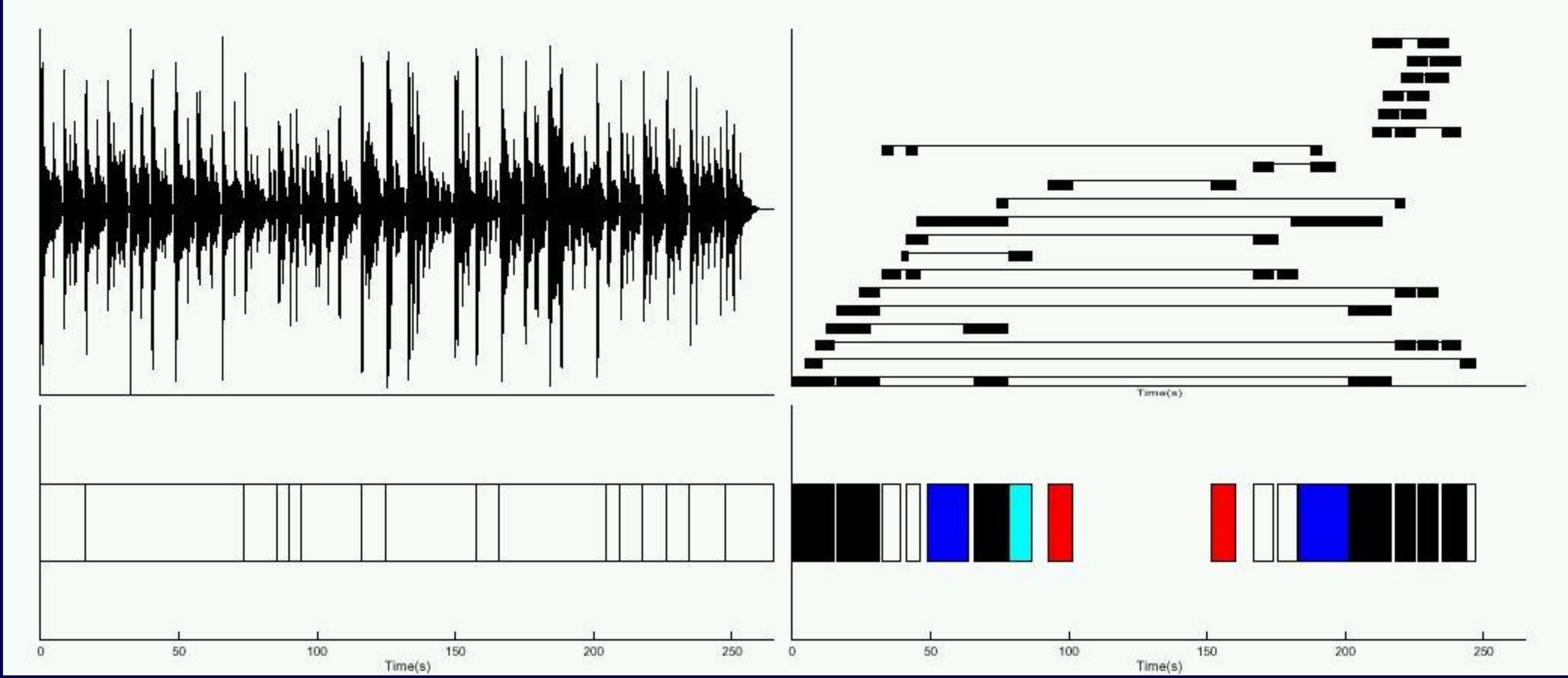


Satin Doll - MIDI



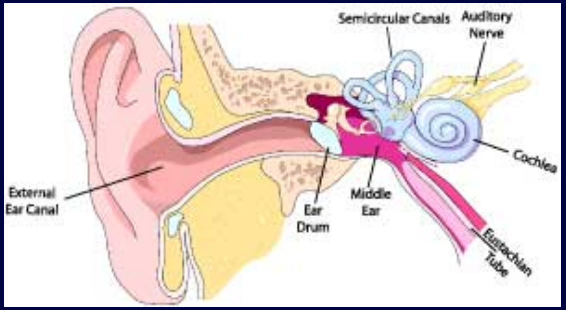
Satin Doll – Audio-from-midi

An example – Naima



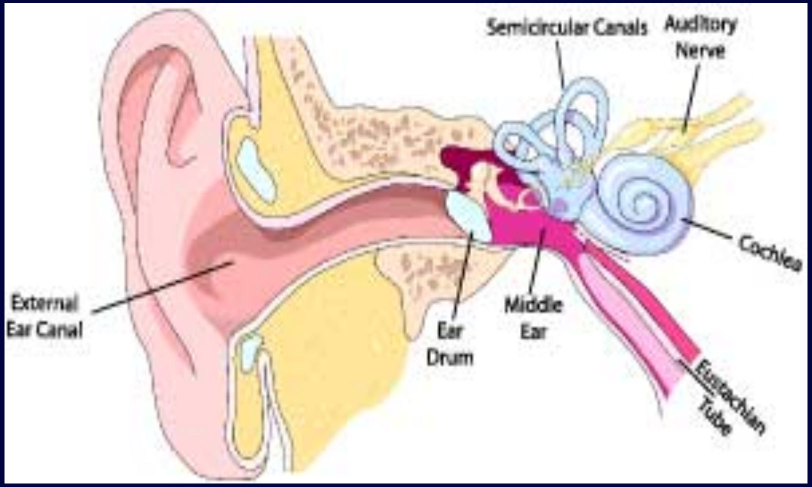
Perception-based approaches

- Pitch perception
- Loudness percetion
- Critical Bands

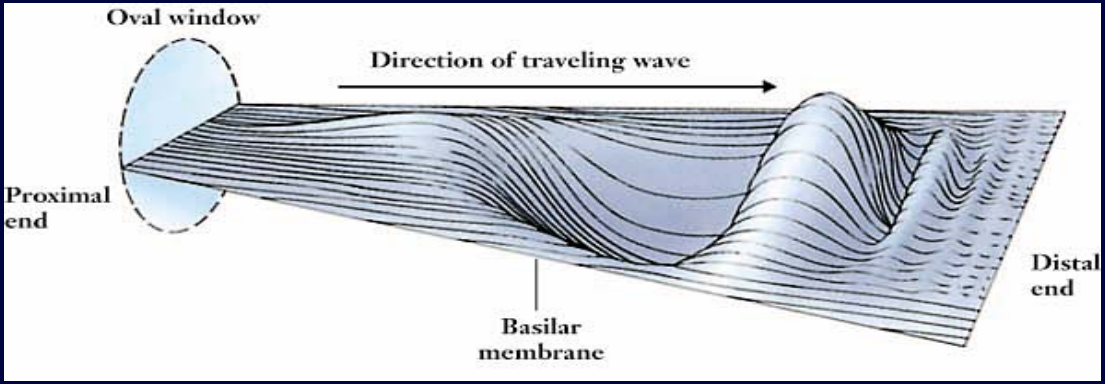


- Mel-Frequency Cepstral Coefficients
- Masking
- Perceptual Audio Compression (MPEG)

The Human Ear



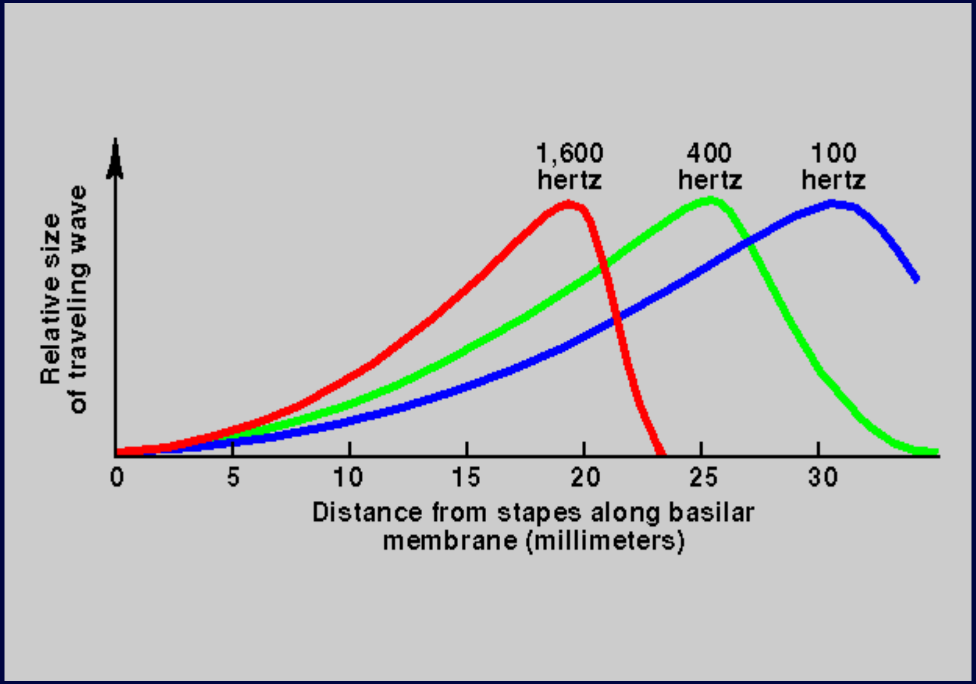
Pinna
Auditory canal
Ear Drum
Stapes-Malleus-Incus (gain control)
Cochlea (freq. analysis)
Auditory Nerve ?



Wave travels to cutoff slowing down increasing in amplitude power is absorbed

Each frequency has a position of maximum displacement

Masking



Two frequencies -> beats
-> harsh
-> seperate

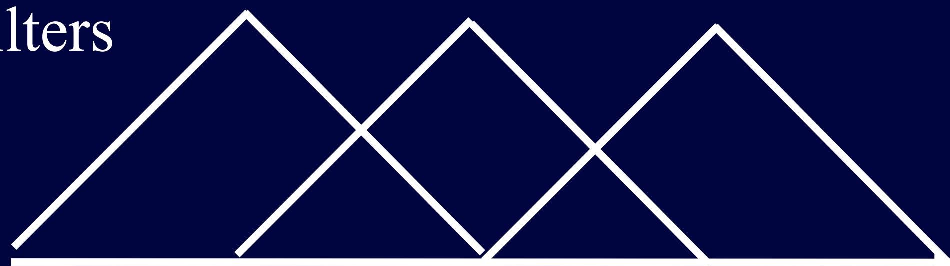
Inner Hair Cell excitation

Frequency Masking
Temporal Masking

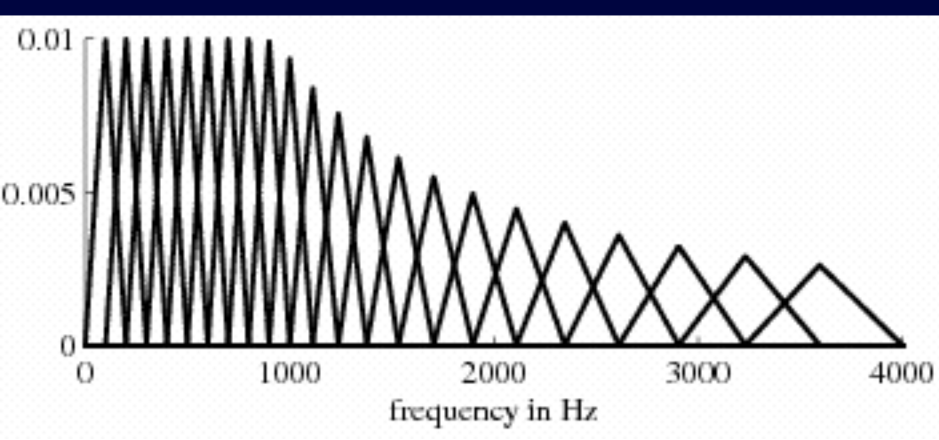
Pairs of sine waves (one softer) – how much weaker in order to be masked ? (masking curves) wave of high frequency can not mask a wave of lower frequency

Mel Frequency Cepstral Coefficients

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

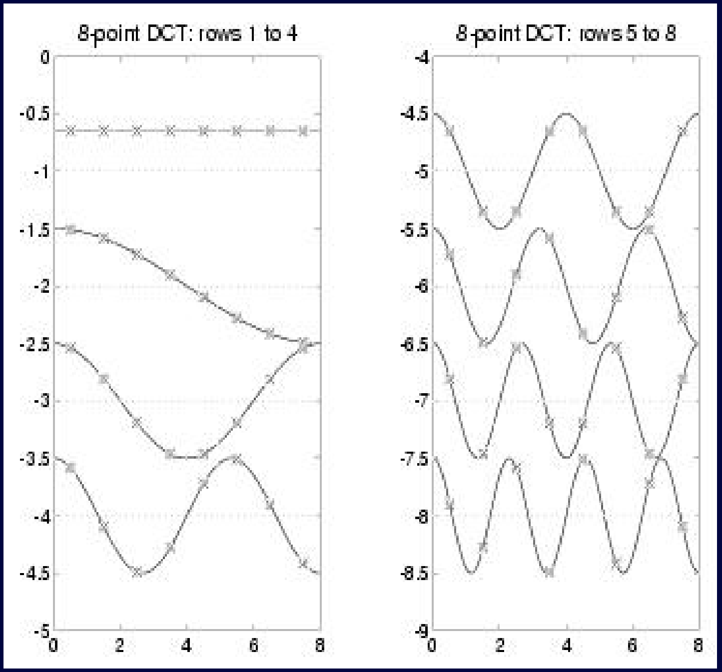


Mel-filtering → Log → DCT → MFCCs



Discrete Cosine Transform

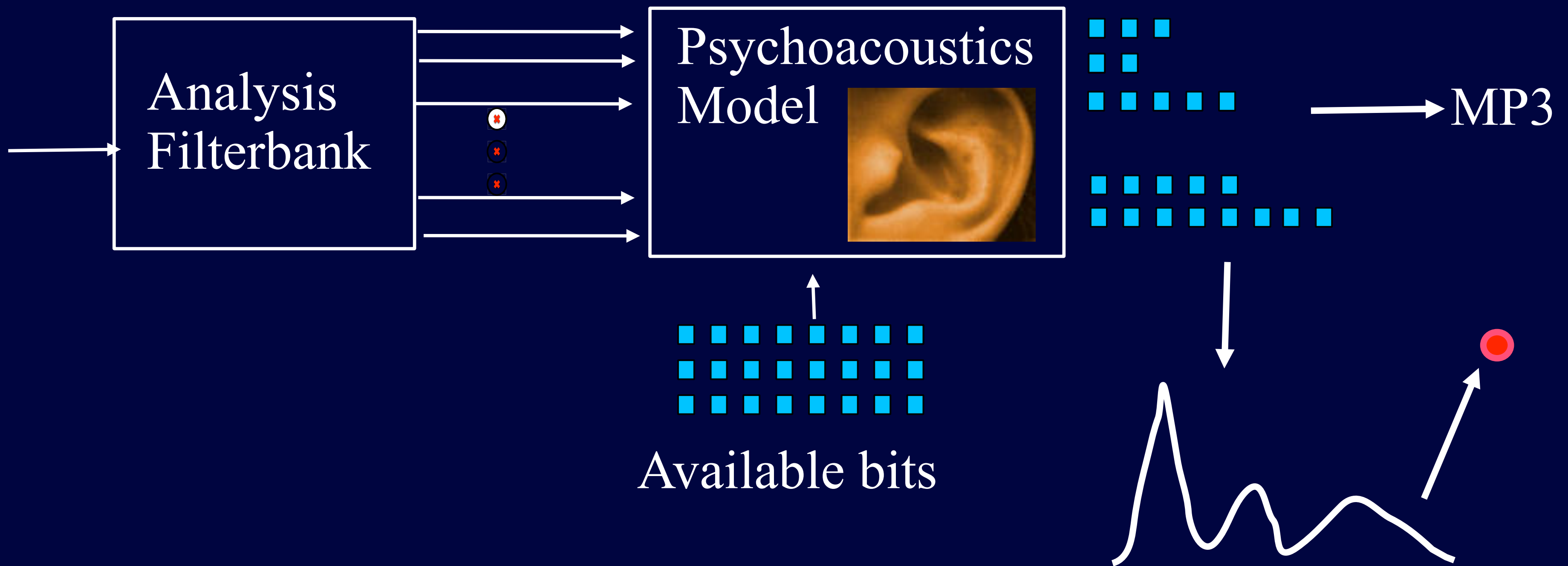
- Strong energy compaction
- For certain types of signals approximates KL transform (optimal)
- Low coefficients represent most the signal - can throw high
- MFCCs keep first 13-20



- MDCT (overlap-based) used in MP3, AAC, Vorbis audio compression

MPEG Audio Feature Extraction

Pye ICASSP 00
Tzanetakis & Cook ICASSP 00



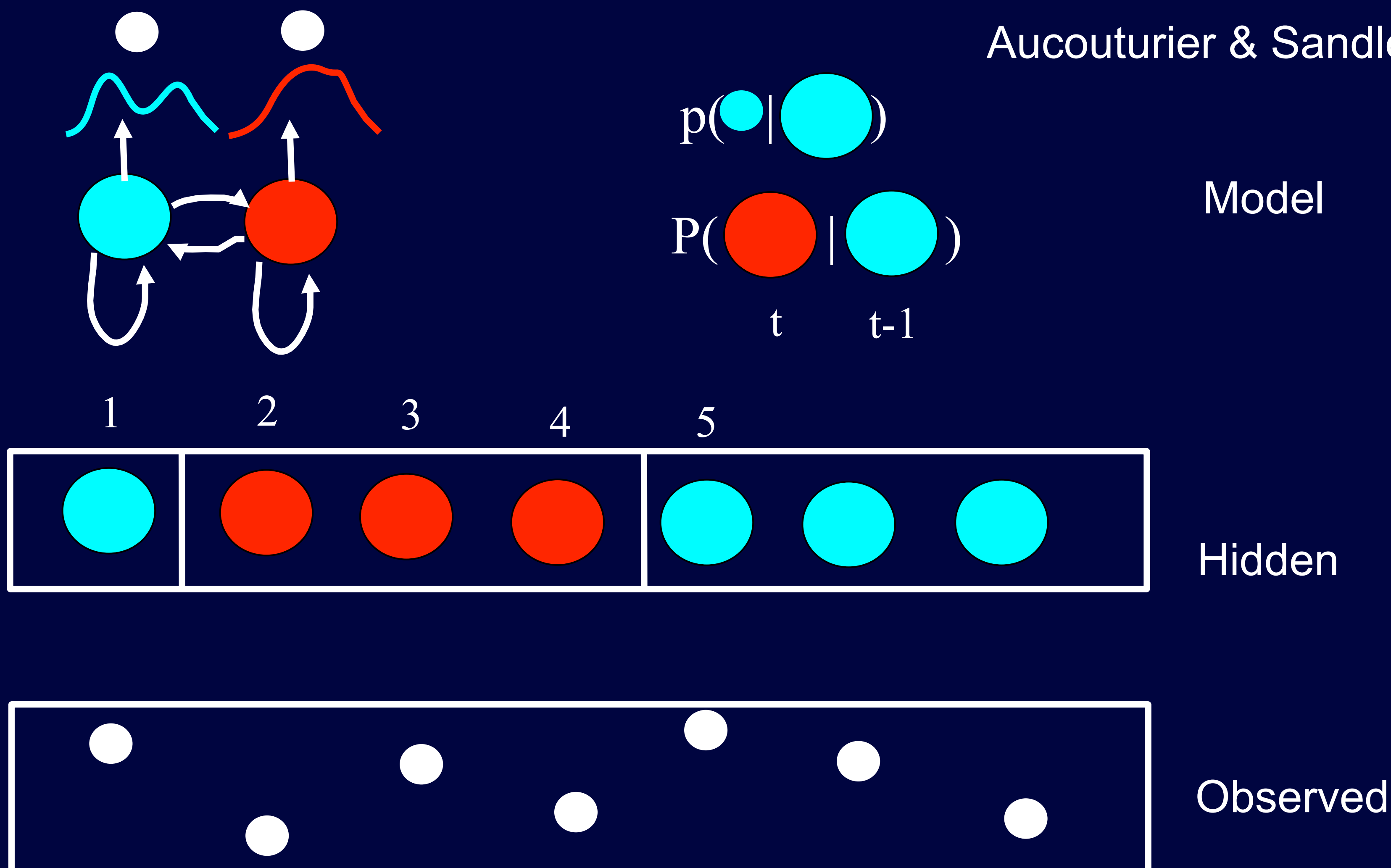
Perceptual Audio Coding (slow encoding, fast decoding)

Psychoacoustic Model

- Each band is quantized
- Quantization introduces noise
- Adapt the quantization so that it is inaudible
- Take advantage of masking
  - Hide quantization noise where it is masked
- MPEG standarizes how the quantized bits are transmitted not the psychoacoustic model – (only recommended)

HMM segmentation

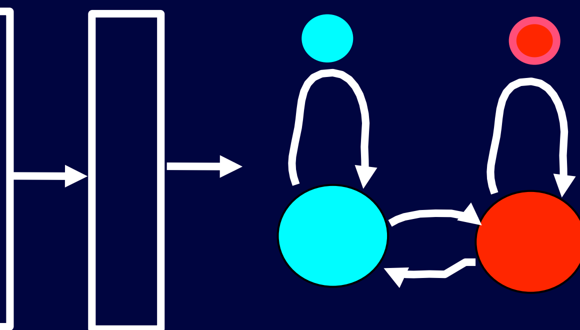
Aucouturier & Sandler, AES 01



Locating singing voice segments

Berenzweig & Ellis, WASPAA 99

Multi-layer perceptron
2000 hidden units
54 phone classes



16 msec
p(phone class)

80% accuracy

“Classic” multi-stage approach

Time-Frequency representation

↓

Grouping Cue 1

↓

Grouping Cue 2

Short Time Fourier Transform
Discrete basis: windowed sine waves

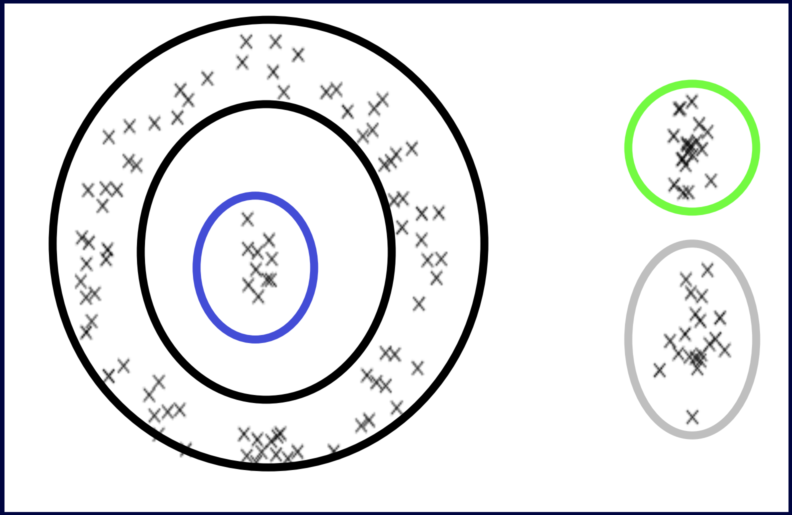
Partial Tracking (McAuley & Quatieri)

Sound source formation:
grouping of partials based on harmonicity

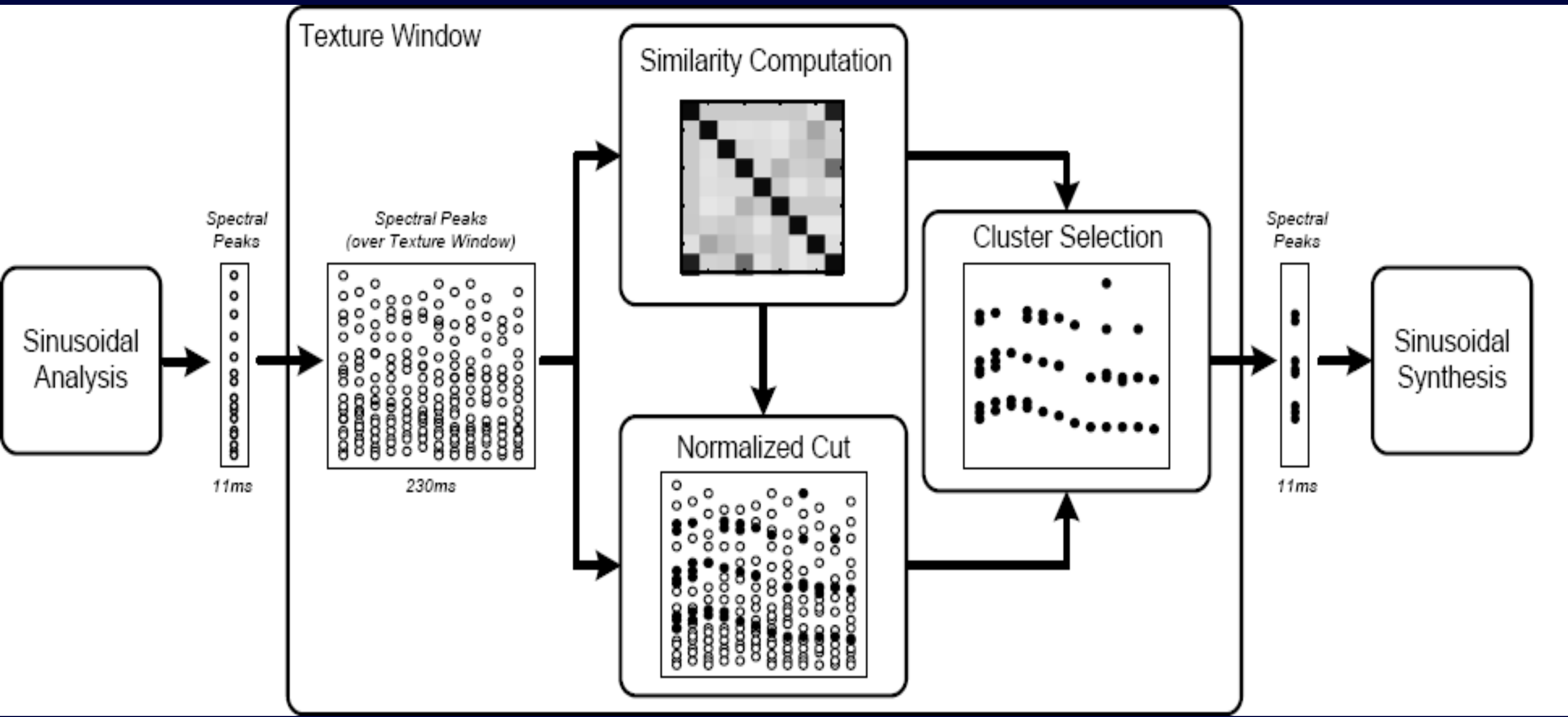
PROBLEMS: Difficult to decide ordering, brittle

Spectral Clustering

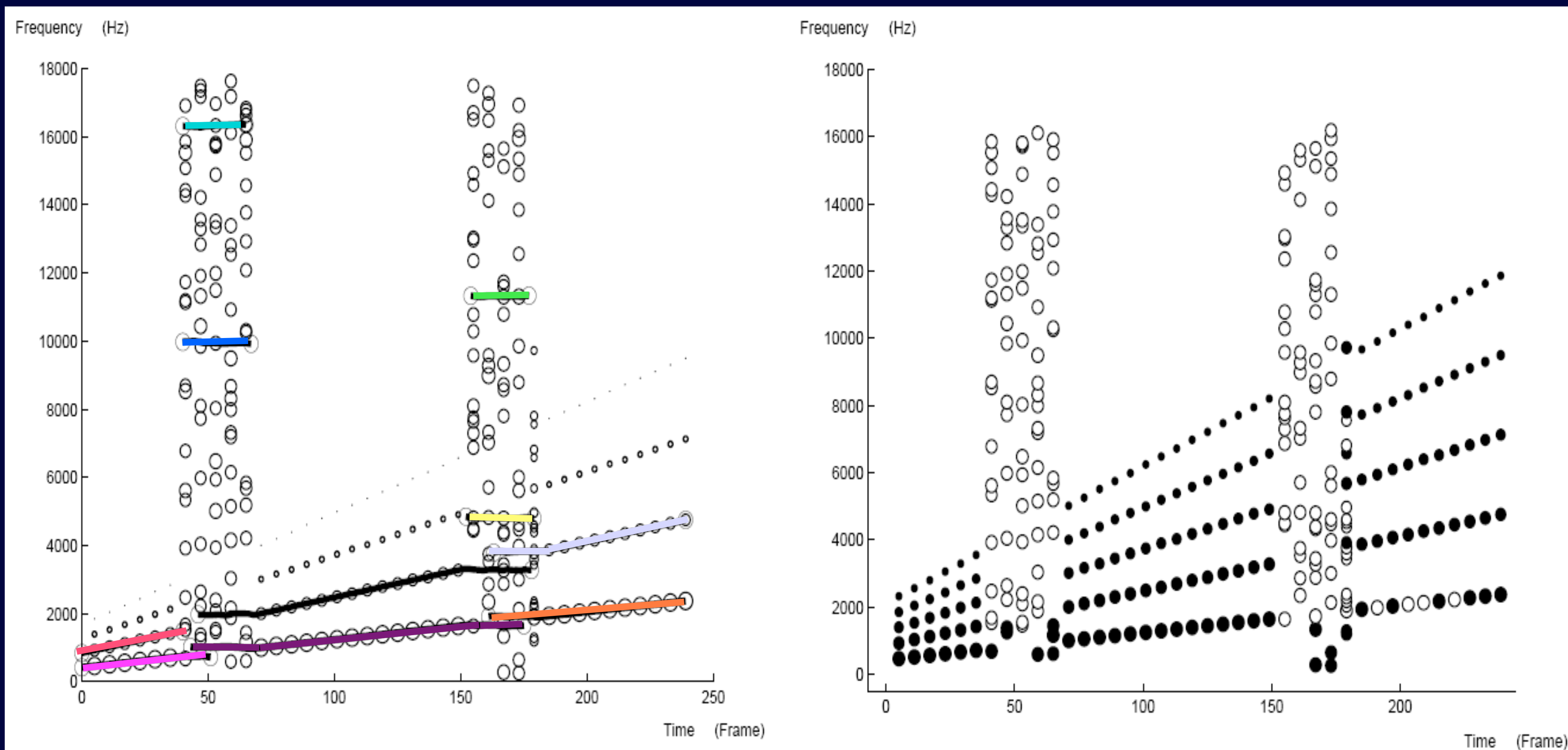
- Alternative to traditional point-based algorthms (k-means)
- Doesn't assume convex shapes
- Doesn't assume Gaussians
- Avoid multiple restarts
- Eigenstructure of



Sound Source Separation using Spectral Clustering



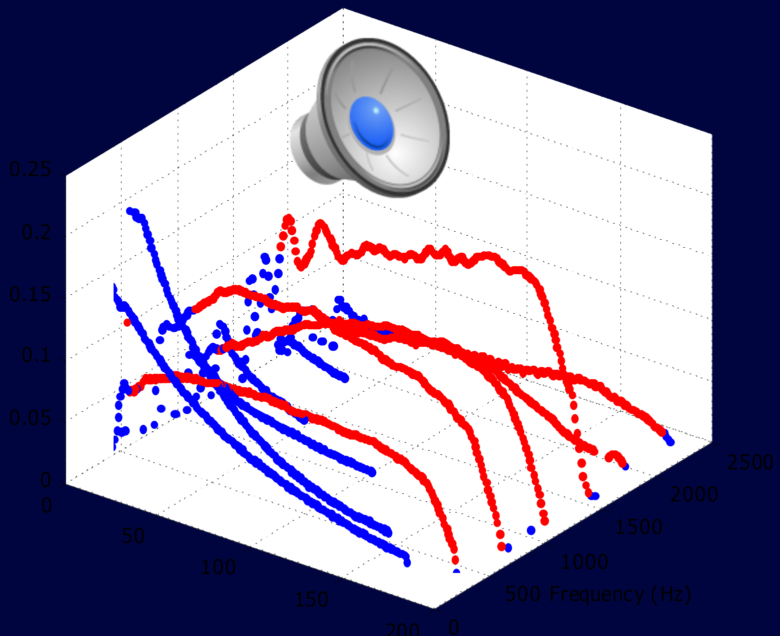
Comparison with partial tracking

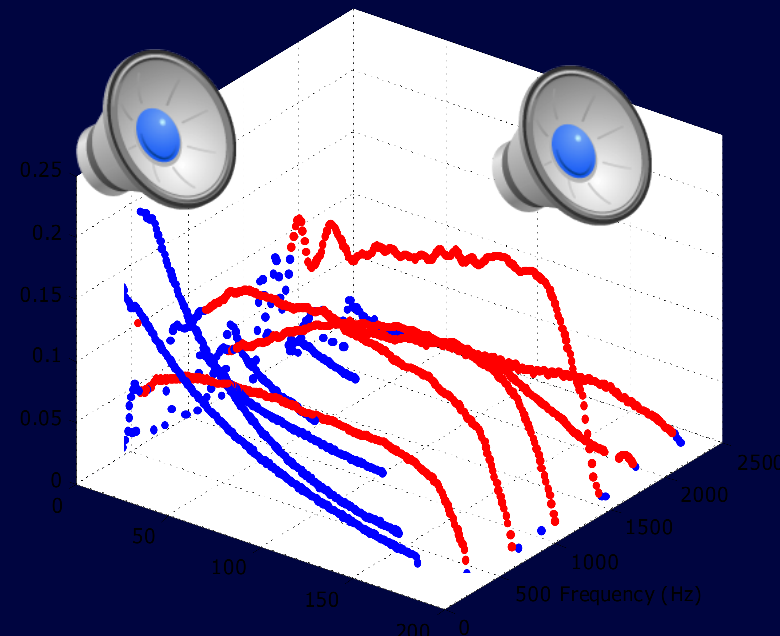


MacAuly and Quatieri Tracking of Partials

Proposed Approach

Synthetic Mixtures of Instruments





4-note mixture

Instrumentation detection based on timbral models

Martins, et al, ISMIR07

“Real world” separation results

- Mirex database
- Live U2

More examples: http://opihi.cs.uvic.ca/NormCutAudio
http://opihi.cs.uvic.ca/Dafx2007

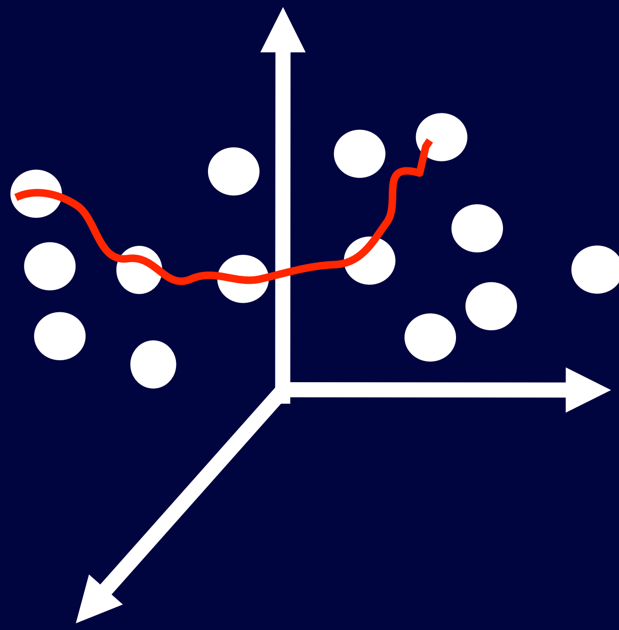
Playlist generation

Tewfik, ICASSP 99
Pachet, IEEE Multimedia00

(s1,s2,s3, ... , sn) 20% slow songs, 80% fast, female jazz singers

Constraint-satisfaction problem
Smooth transitions

Technical attributes (artist, album, name)
Content attributes (jazz singer, brass)



Audio Thumbnailing

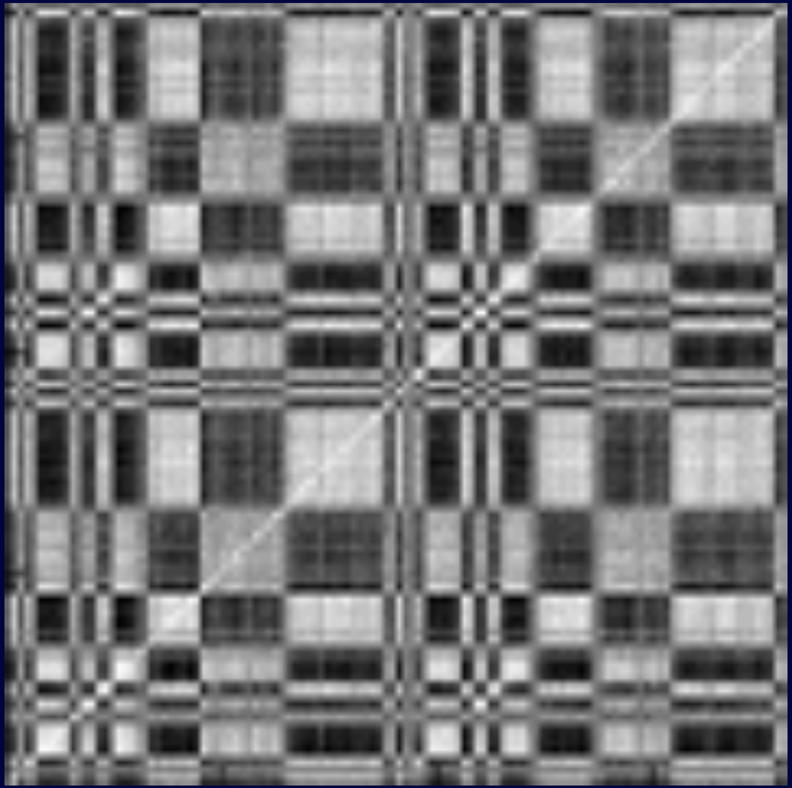
- Representative short summary of piece
  - Segmentation-based
  - Repetition-based
- Hard to evaluate

Segmentation-based Thumbnailing

- Begin and end times of a 2 second thumbnail that best represents the segment
  - 62% first two seconds of the segment
  - 92% two seconds within the first five seconds of the segment
- Automatic thumbnailing
  - first 5 sec + best effort about 80% "correct"

Repetition-based thumbnailing

Logan, B., ICASSP 00
Bartch and Wakefield,WASPAA99

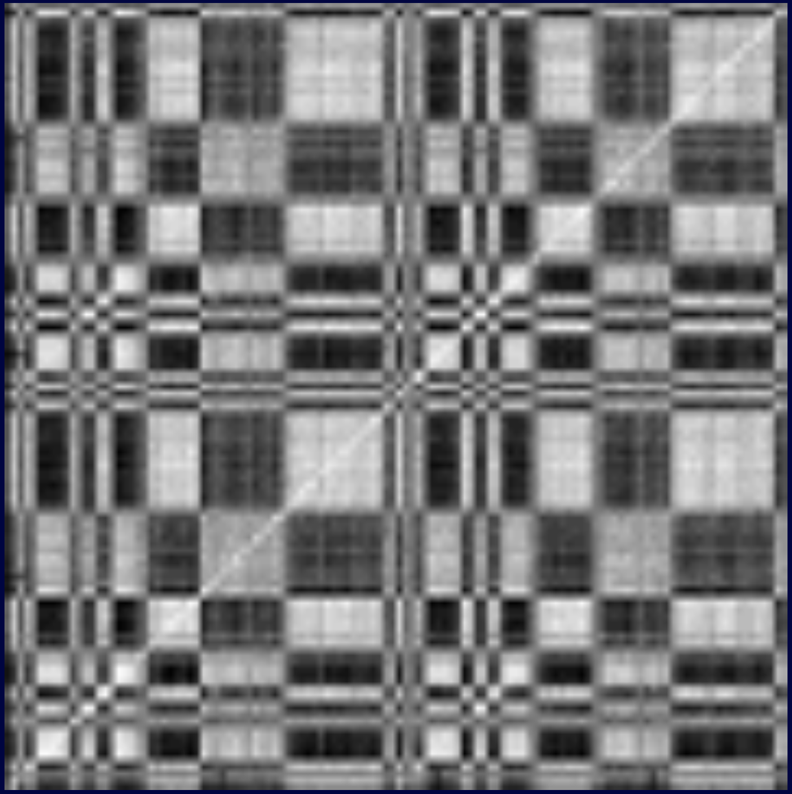


Thumbnail = maximum repeated segment

Alternatives: Clustering, HMM

Structure from similarity

Foote et al, ISMIR 02
Dannenberg et al, ICMC 02



Feature vector trajectory
Correlation at various time lags

ABAA'

Query-by-humming

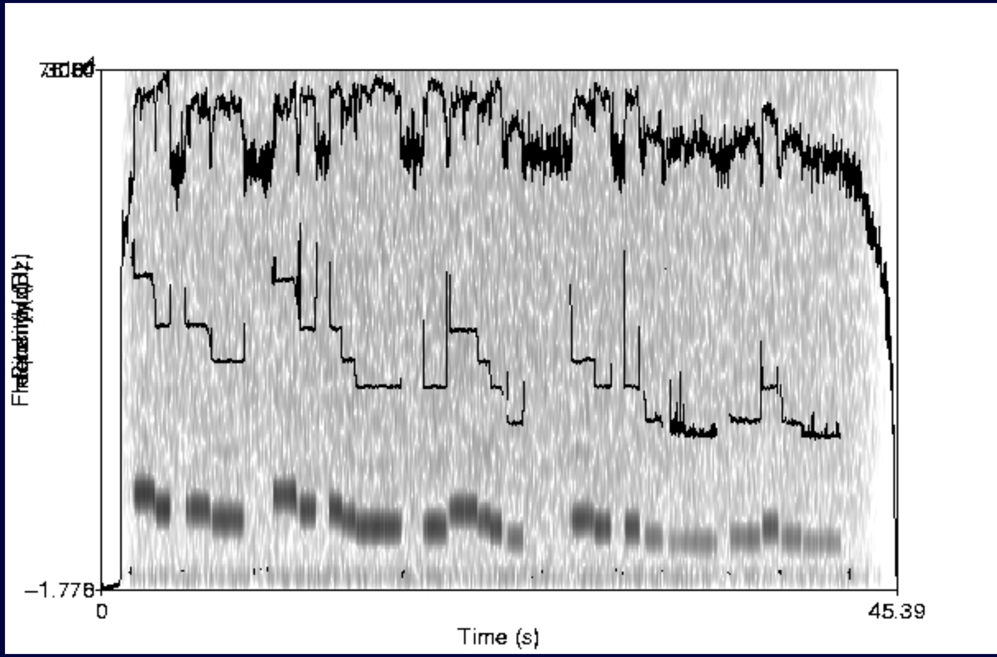
- User sings a melody
- Computer searches database for song containing the melody
- Probably less useful than it sounds but interesting problem
- The challenge of difficult queries

The MUSART system

- Query preprocessing
  - Pitch contour extraction (audio)
  - Note segmentation (symbolic)
- Target preprocessing (symbolic)
  - Theme extraction
  - Model-forming, representation
- Search to find approximate match
  - Dynamic Programming, HMMs

Representations

- Pitch and tempo invariance
  - Quantized pitch intervals
  - Quantized IOI ratio
- Approximate matching
  - HMM
  - Dynamic programming
  - Time Series

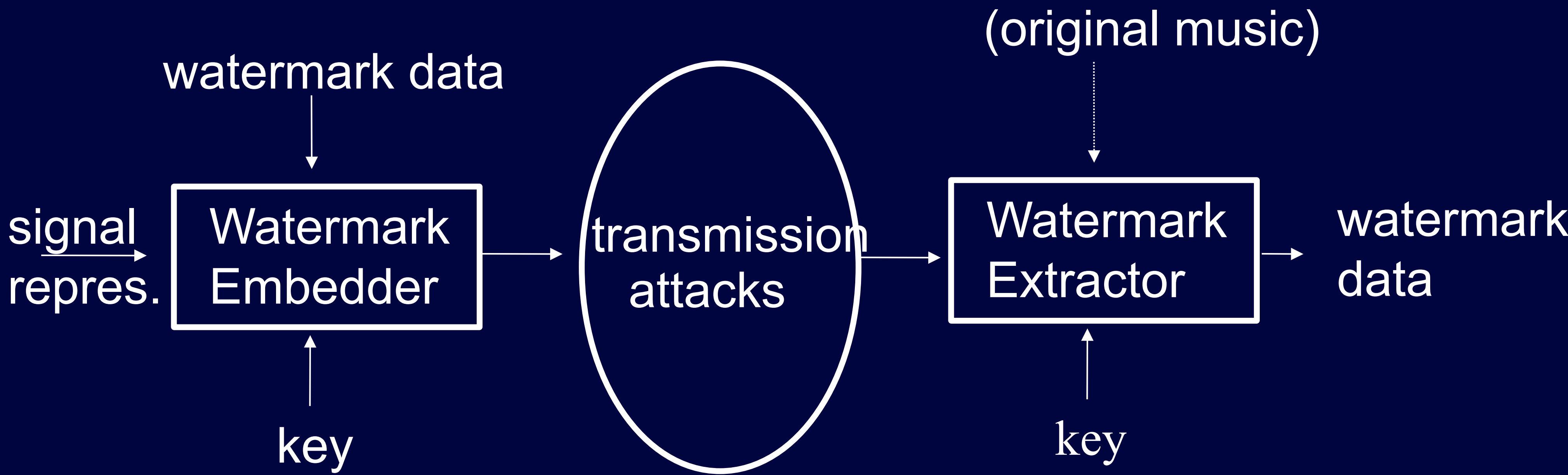


Audio Fingerprinting and Watermarking

- Watermarking
  - Copyright protection
    - Proof of ownership
    - Usage policies
  - Metadata hiding
- Fingerprinting
  - Tracking
  - Copyright protection
  - Metadata linking

Watermarking

- Steganography (hiding information in messages – invisible ink)



Desired Properties

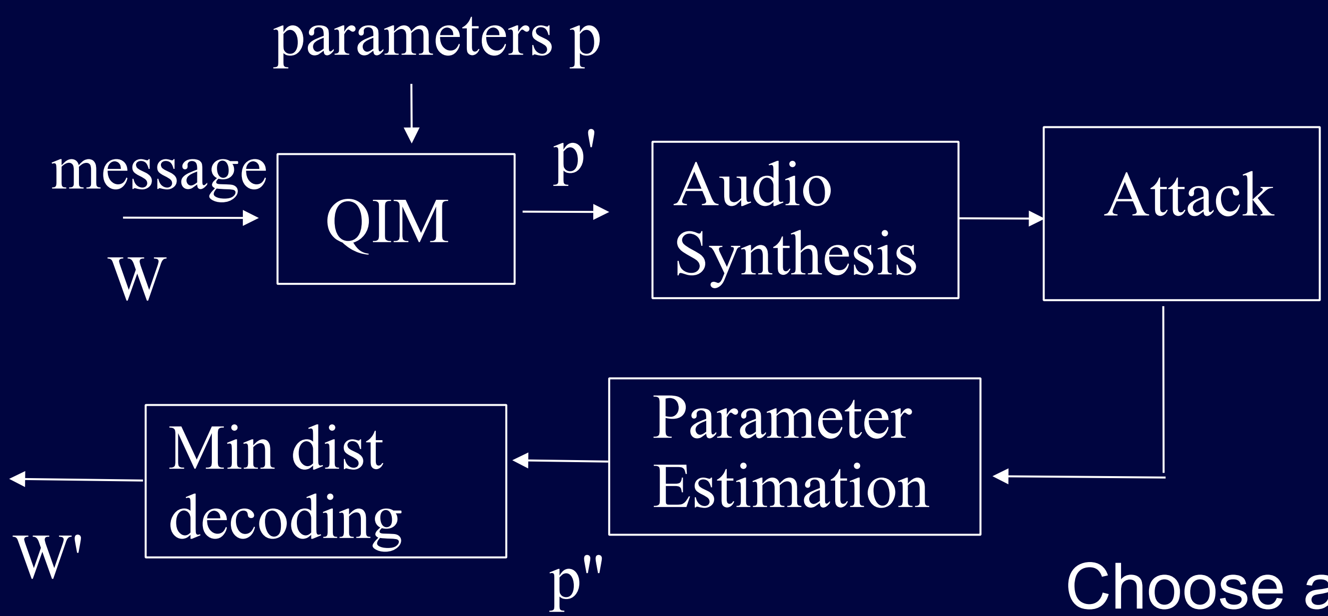
- Perceptually hidden (inaudible)
- Statistically invisible
- Robust against signal processing
- Tamper resistant
- Spread in the music, not in header
- key dependent

Representations for Watermarking

- Basic Principles
  - Psychoacoustics
  - Spread Spectrum
    - redundant spread of information in TF plane
- Representations
  - Linear PCM
  - Compressed bitstreams
  - Phase, stereo
  - Parametric representations

Watermarking on parametric representations

Yi-Wen Liu
J. Smith 2004



Choose attack tolerance quantize so perceptual distortion < t and lattice finding possible

Problems with watermarking

- The security of the entire system depends on devices available to attackers
  - Breaks Kerckhoff's Criterion: A security system must work even if reverse-engineered
- Mismatch attacks
  - Time stretch audio – stretch it back (invertible)
- Oracle attacks
  - Poll watermark detector

Audio Fingerprinting

- Each song is represent as a fingerprint (small robust representation)
- Search database based on fingerprint
- Main challenges
  - highly robust fingerprint extraction
  - efficient fingerprint search strategy
- Information is summarized from the whole song – attacks degrade unlike watermarking

Hash functions

- H(X) -> maps large X to small hash value
- compare by comparing hash value
- Perceptual hash function ?
  - impossible to get exact matching
- Perceptually similar objects result in similar fingerprints
- Detection/false alarm tradeoff

Properties

- Robustness
- Reliability
- Fingerprint size
- Granularity
- Search speed and scalability

Fraunhofer

Allamanche Ismir 2001

- LLD Mpeg-7 framework (SFM)
- Vector quantization (k-means)
  - Codebook of representative vectors
- Database target signature is the codebook
- Query -> sequence of feature vectors
- Matching by finding “best” codebook
- Robust not very scalable (O(n) search))

Philips Research

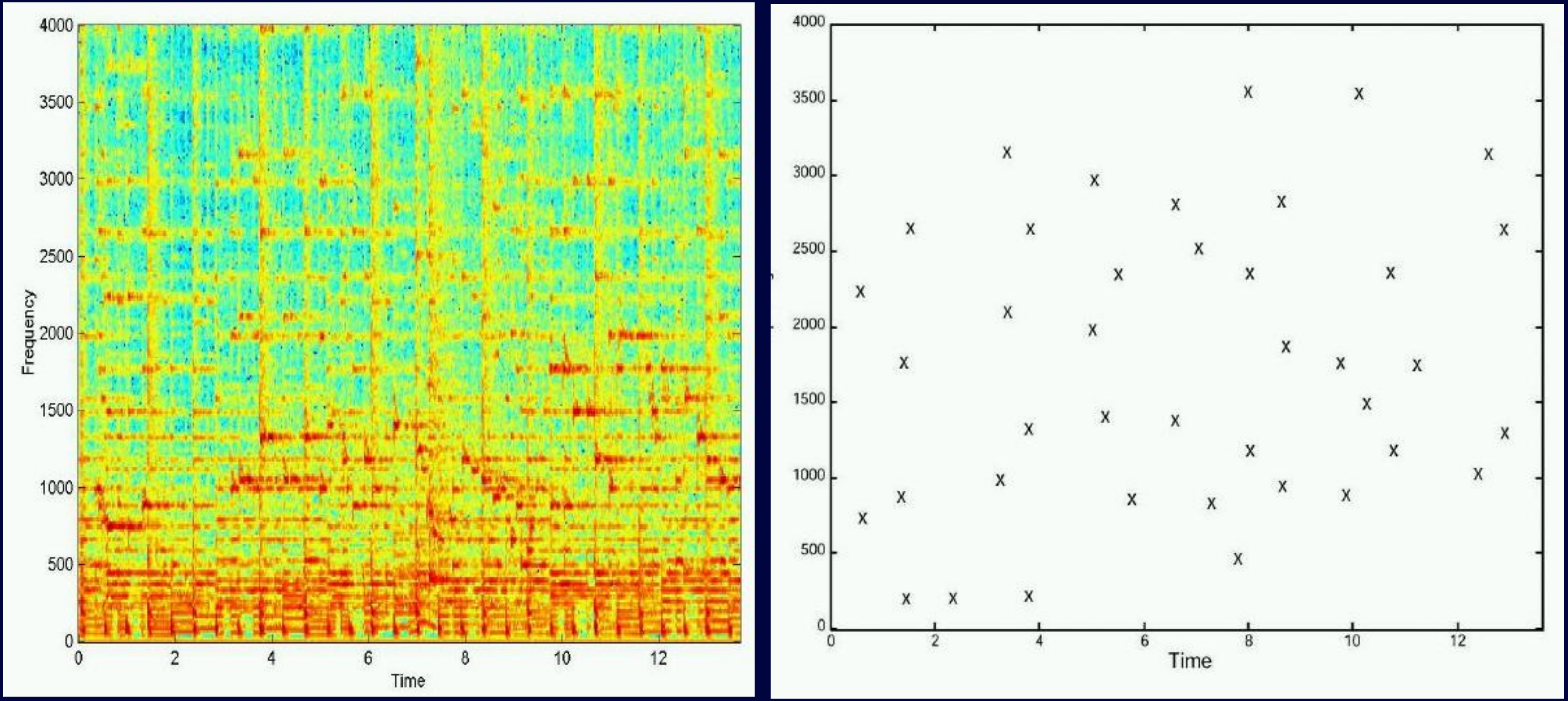
Haitsa & Kalker
Ismir 2002

- 32-bit subfingerprints for every 11.6 msec
- overlapping frames of 0.37 seconds (31/32 overlap)
- PSD -> logarithmic band spacing (bark)
- bits 0-1 sign of energy
- looks like a fingerprint
- assume one fingerprint perfect – hierarchical database layout (works ok)

Shazam Entertainment

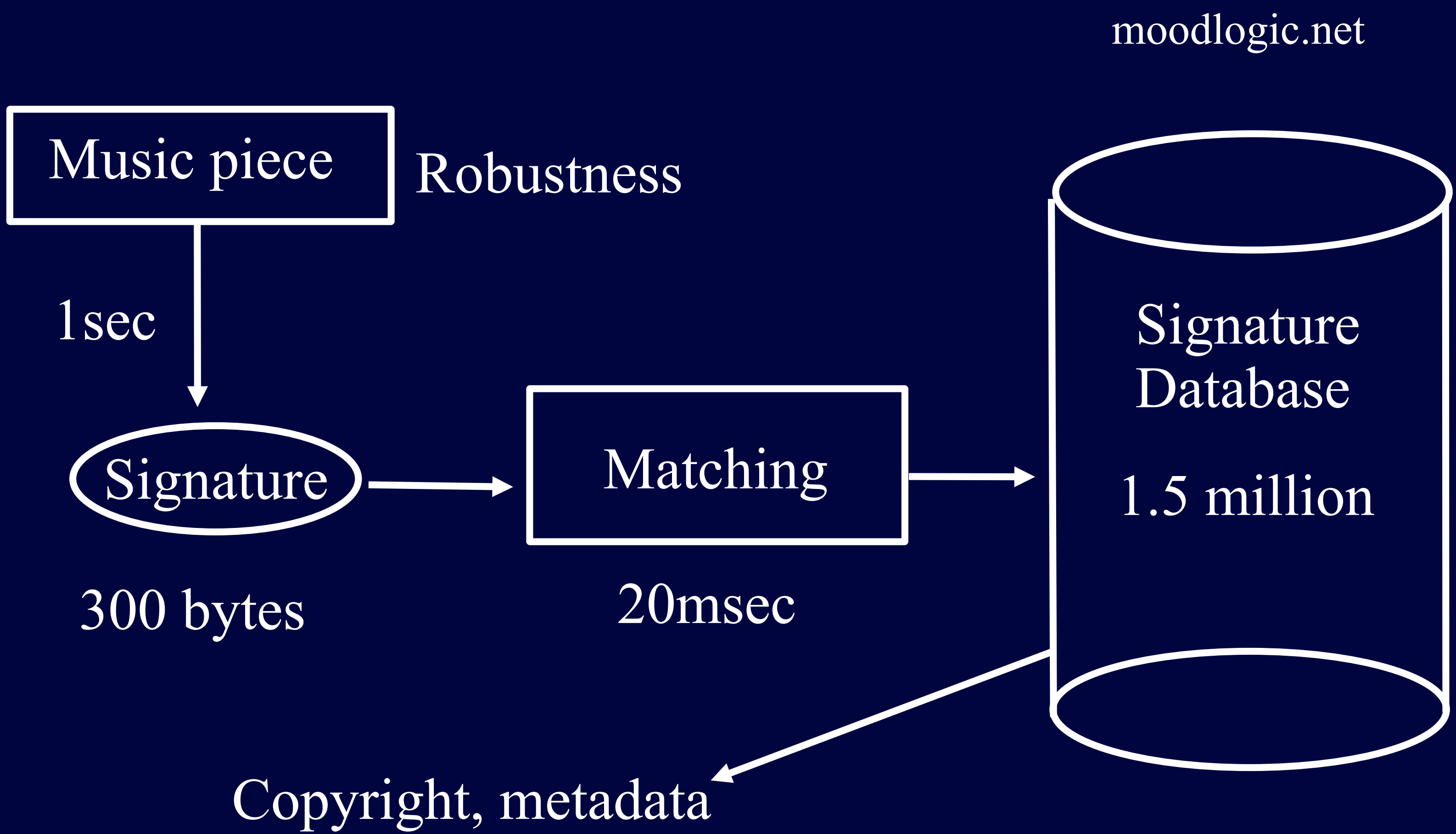
- Pick landmarks on audio – calculate fingerprint
- histogram of relative time differences for filtering
- Spectrogram peaks (time, frequency)

Spectrogram Peaks



Very robust – even over noisy cell phones

Audio Fingerprinting



University of Victoria

STEREO PANNING FEATURES FOR CLASSIFYING RECORDING PRODUCTION STYLE

ISMIR 2007

8th International Conference on Music Information Retrieval

23rd – 27th September 2007

Vienna, Austria

George Tzanetakis gtzan@cs.uvic.ca

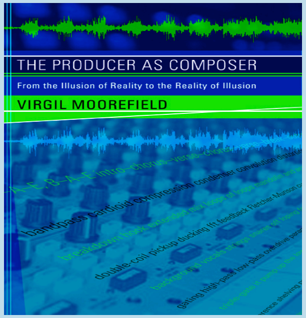
Randy Jones

Kirk McNally

Motivation

- The “classic” audio MIR system

- Mixing and production are critical in modern recordings
  - The Producer as a Composer - Virgil Moorefield, MIT Press, 2005
  - “Famous” record producers
    - Phil Spector, George Martin, Quincy Jones, Brian Eno



- Can recording production information and more specifically stereo mixing information assist the automatic extraction of information from audio signals ?

Related Work

- The "album" effect
  - Artist identification performance degrades when different albums are used for training & evaluation (Whitman, IEEE NNSP, 2003)
  - Compensate mastering equalization curves (Kim, ISMIR 2006)
- "Glass" ceiling of timbral features
  - (Aucouturier, Journal of Negative Results in Speech and Audio 2004)
- Stereo-based source separation
  - For each FFT bin calculate panning coefficient
  - Group together bins that have similar panning coefficients as belonging to the same sound source
  - (Avendano, WASPAA 2003) (Woodruff, ISMIR 2003)

Stereo Panning Spectrum

$$\psi(k) = 2 * \frac{|X_l(k)X_r^*(k)|}{|X_l(k)|^2 + |X_r(k)|^2}$$

$$\psi_l = \frac{|X_l(k)X_r^*(k)|}{|X_l(k)|^2}, \psi_r = \frac{|X_r(k)X_l^*(k)|}{|X_r(k)|^2}$$

$$\Delta(k) = \psi_l - \psi_r$$

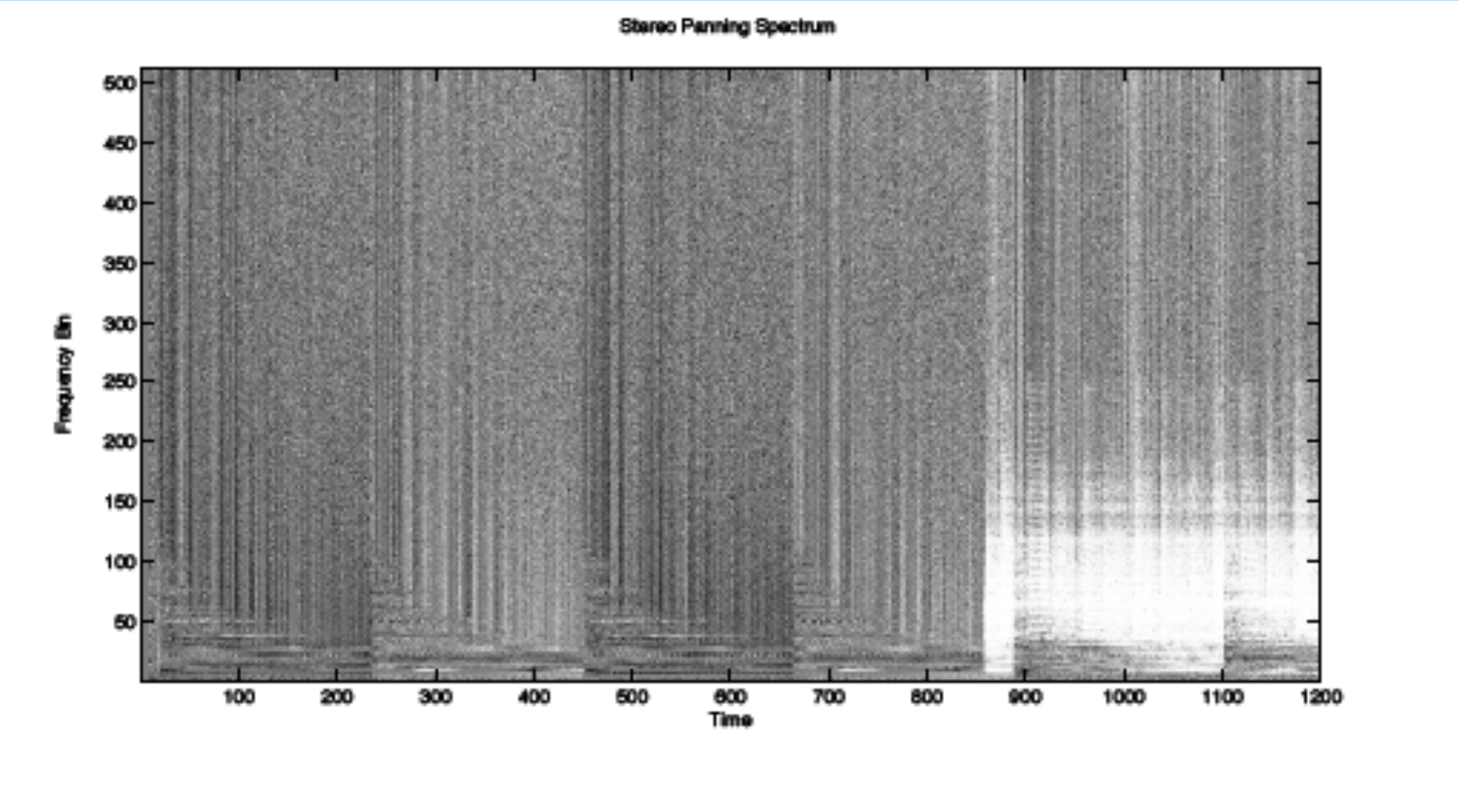
$$\hat{\Delta}(k) = \begin{cases} +1, & \text{if } \Delta(k) > 0 \\ 0, & \text{if } \Delta(k) = 0 \\ -1, & \text{if } \Delta(k) < 0 \end{cases}$$

$$SPS(k) = [1 - \psi(k)] * \hat{\Delta}(k)$$

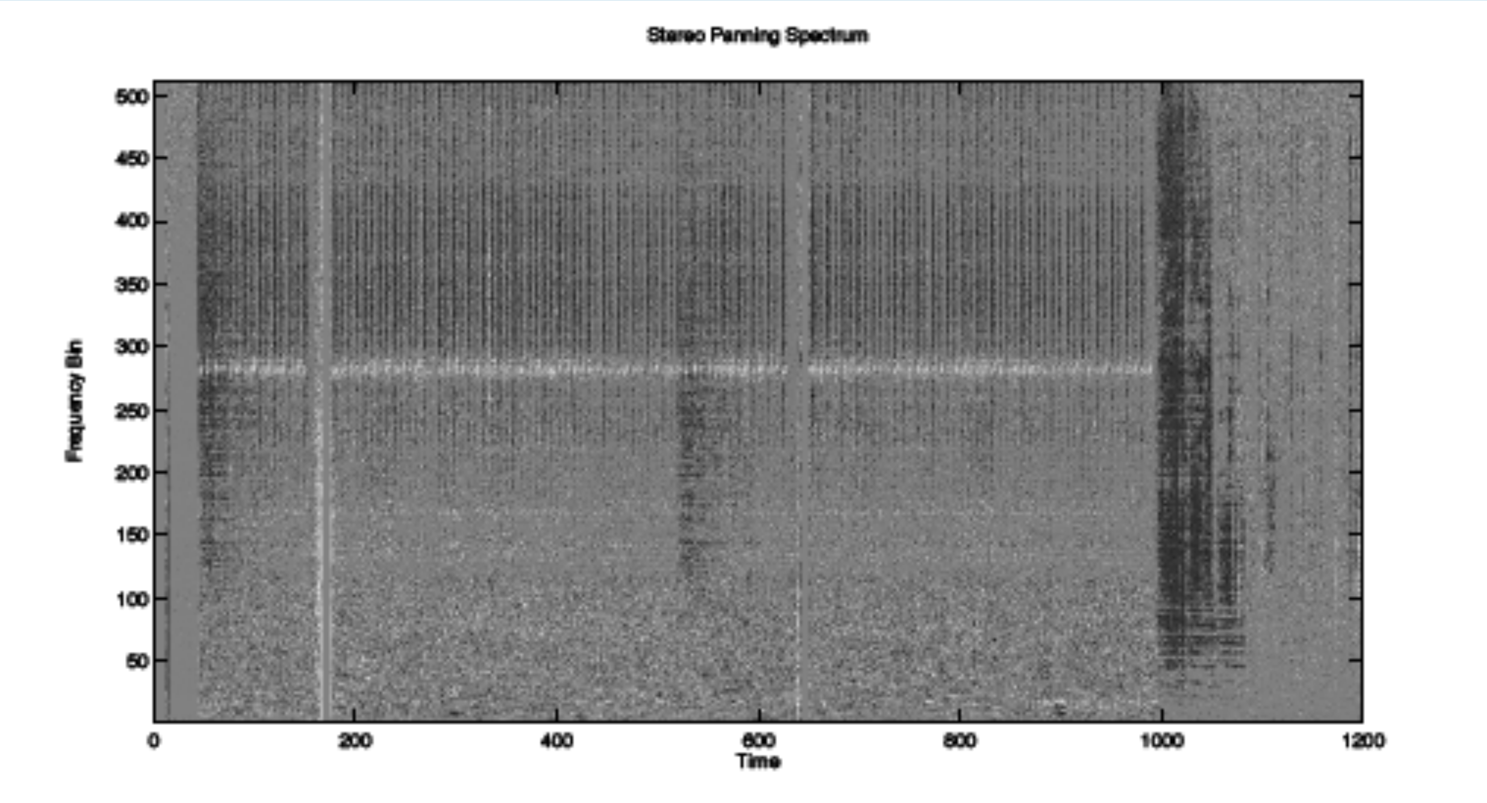
Based on
(Avendano, WASPAA 2003)

For every FFT bin a
panning coefficient
between -1 (left) and +1 (right)
with the center at 0

Example I



Hell's Bells - ACDC (black left, white right)
Important note: Just panning information
(invariant to frequency content/dynamics)



Supervixen by Garbage (black left, white right)

Stereo Panning Spectrum Features

$$P_{l,h} = \sqrt{\frac{1}{h-l}\sum_{k=l}^{h}[SPS(k)]^2}$$

Panning RMS for subbands
Low (0-250 Hz)
Medium (250-2500 Hz)
High (2500-22050 Hz)

$$\Phi(t) = [P_{total}(t), P_{low}(t), P_{medium}(t), P_{high}(t)]$$

$$m\Phi(t) = mean[\Phi(t-M+1), .., \Phi(t)]$$
$$s\Phi(t) = std[\Phi(t-M+1), .., \Phi(t)]$$

Texture-window features
For dynamics

Final features are the means and standard deviations of the entire audio clip resulting in a 4*2*2 = 16 dimensional vector / audio clip

Experiments

- Collections
  - 1960s “garage” (The Byrds, The Kinks, Buddy Holly)
  - 1980s “grunge” (Nirvana, PearlJam, RadioHead)
  - Acoustic Jazz (Miles Davis, John Coltrange, Wynton Marsalis)
  - Electric Jazz (Return to Forever, Weather Report, Mahavishnu Orchestra)
- Configurations
  - STEREO Stereo Panning Spectrum Features (SPSF)
  - STEREO MFCC (concatenate MFCCs for left and right)
  - STEREO (SPSF+MFCC)

Results (10-fold cross-validation)

Garage/Grunge	ZeroR	NBC	SMO	J48
SPSF	56.4	77.2	81	84.2
SMFCC	56.4	74. 6	76.7	71.6
SPSF+SMFCC	56.4	82.7	83.7	83.2

Acoustic/Electric	ZeroR	NBC	SMO	J48
SPSF	51.3	99.4	99.7	99.1
SMFCC	51.3	71.8	79.4	68.4
SPSF+SMFCC	51.3	98.5	99.1	99.1

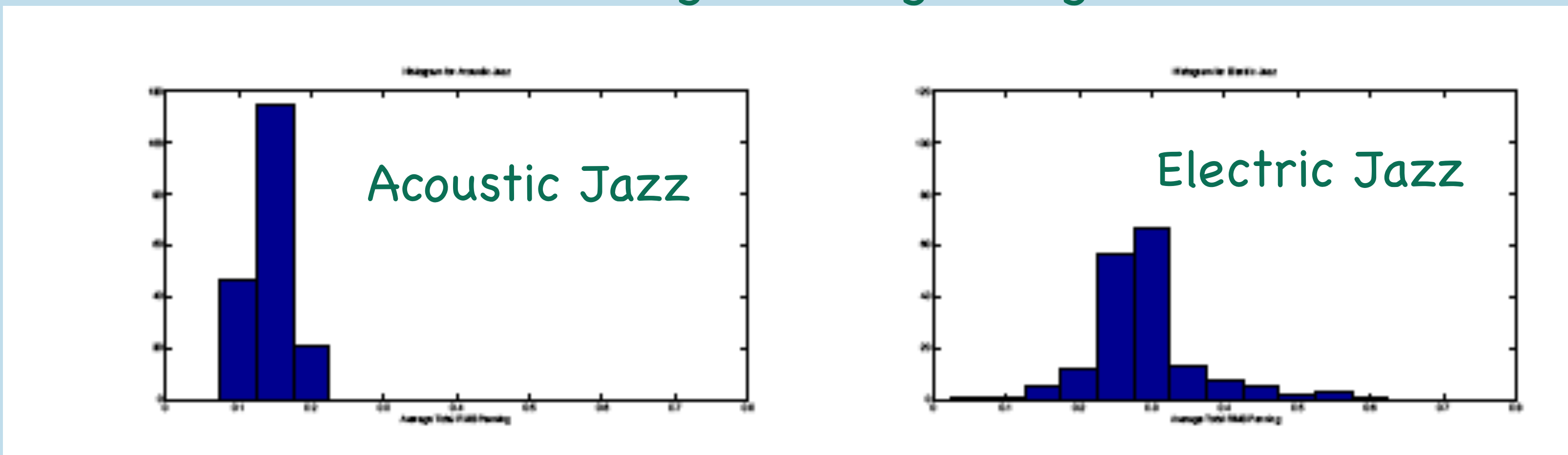
Gr/Ga/Aj/Ej	ZeroR	NBC	SMO	J48
SPSF	29.8	73.6	81	76.5
SMFCC	29.8	56.4	65.9	52.3
SPSF+SMFCC	29.8	75.2	87.4	79.9

WEKA
The University of Waikato

Naïve Bayes Classifier
Linear SVM trained
Using SMO
Decision Tree

Some more results

Mean Average Panning Histograms



Artist20Id	ZeroR	SPSF	SMFCC	SPSF+SMFCC
fold0	6.7	19.4	32.2	31.9
fold1	5.9	16.9	31.05	28.7
fold2	4.6	20	35.1	28.7
fold3	5.2	13.9	37.5	36.6
fold4	5.1	20.4	35.4	36.6
fold5	5.0	12.5	17.6	18.8
avg	**5.4**	**17.1**	**31.4**	**32.6**

Artist20 Id
with album folds
(not in paper)

MIREX 2007 results

- Music Information Retrieval Evaluation Exchange (MIREX)
  - Annual forum for comparing algorithms of different MIR tasks
  - 2007 “classification tasks”
    - Audio Artist Identification, Classical Composer Identification
    - Audio Genre Classification, Audio Mood Classification

	artist	genre	composer
SPSF	14.22	41.41	26.19
SMFCC	36.50	61.79	45.13
SPSF/SMFCC	45.16	64.13	49.89
Best MIREX07	48.14	68.29	53.72

Implementation

MARSYAS

MUSIC ANALYSIS, RETRIEVAL AND SYNTHESIS FOR AUDIO SIGNALS

http://marsyas.sourceforge.net

```
bextract -e STEREOSPS /pathto/mirexcollection.txt -w fmatrix.arff
bextract -e STEREOMFCC /pathto/mirexcollection.txt -w fmatrix.arff
bextract -e STEREOSPSMFCC /pathto/mirexcollection.txt -w fmatrix.arff
```

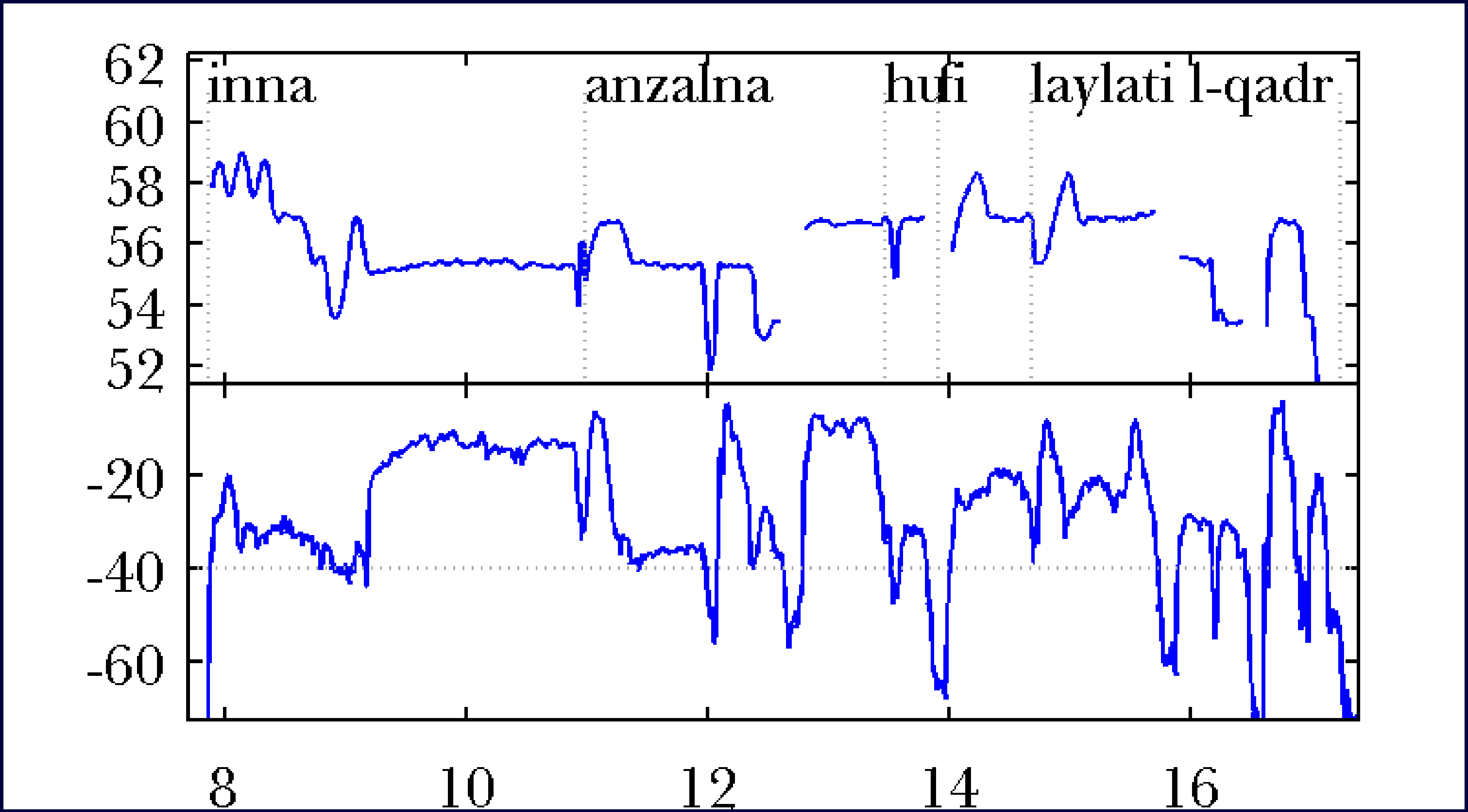
Future Work & Acknowledgements

- Characterize more generally recording production style
- Audio-based record producer identification
  - Maybe a future MIREX task ?
- Dave Pensado describing one of his mixes (Mix Magazine)
  - "Massive club bottom, hiphop sensibility in the bottom, and this real smoothed-out, classy Quincy Jones types top"
  - Can we move this type of description into audio MIR ?
- Many thanks to:
  - NSERC, SSHRC Canada
  - Carlos Avendano for clearly describing his algorithm
  - Dan Ellis for providing stereo files for the ArtistID 20
  - Perry Cook for suggesting using stereo information a long time ago

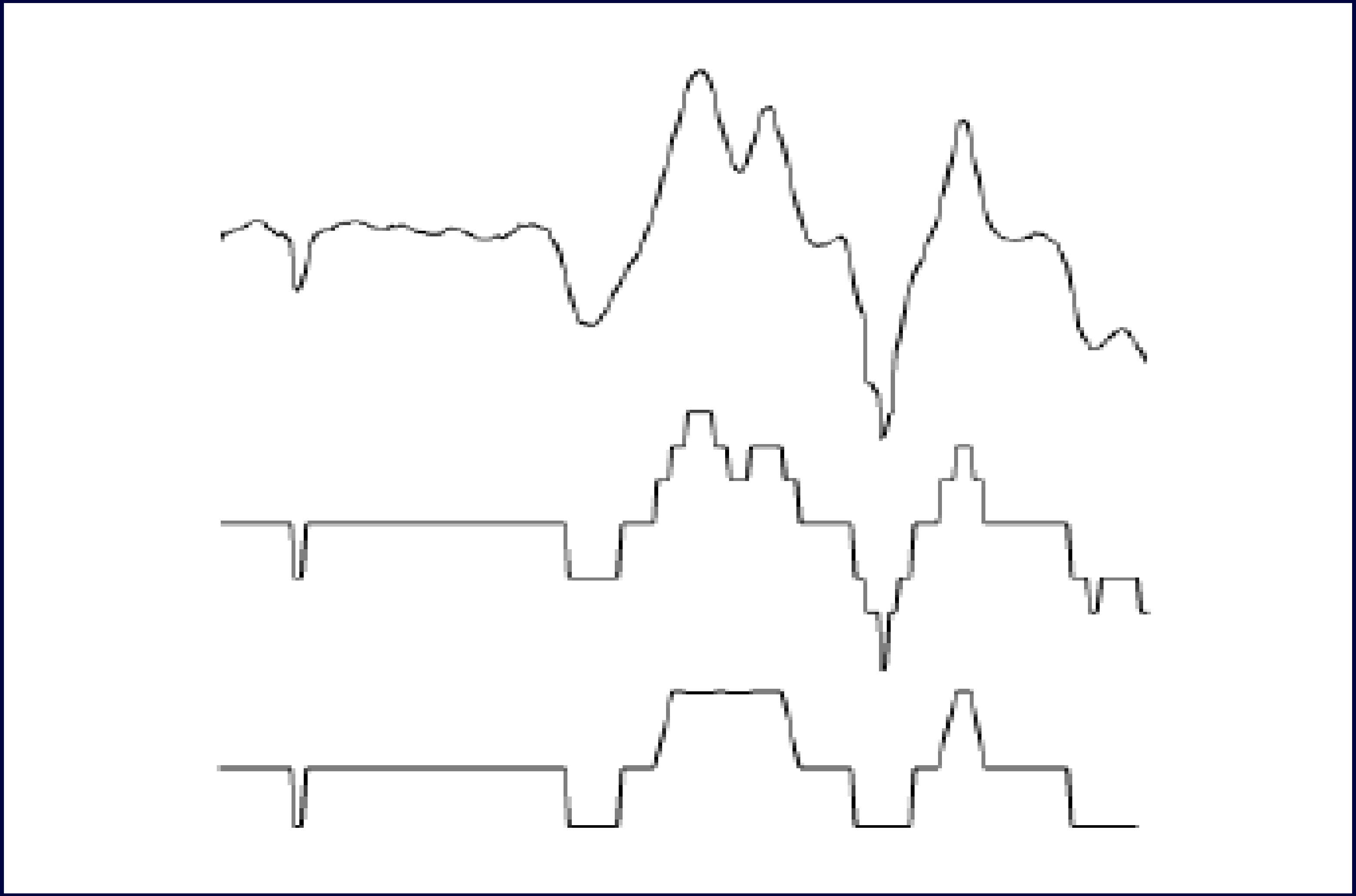
Computational Musicology - recitation research

- Transition from oral to written transmission
- Study how diverse recitation traditions having their origin in primarily non-notated melodies, later became codified
- Hungarian siratok, torah cantillation, koran recitation, 10th century St. Gallen plainchant

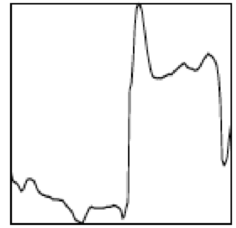
Pitch Contour Extraction



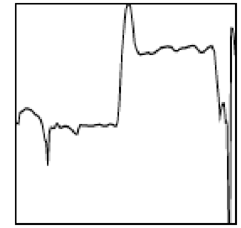
Histogram-based contour abstraction



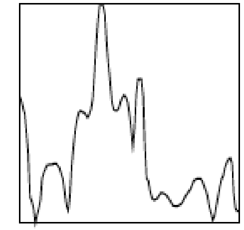
Dynamic-Time Warping



(a) F0 Contour of 11 Pashta

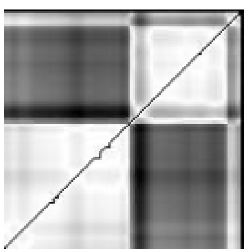


(b) F0 Contour of 42 Pashta

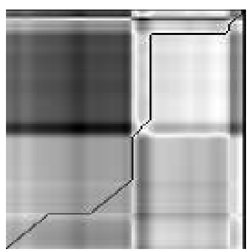


(c) F0 Contour of 18 Sof Pasuq

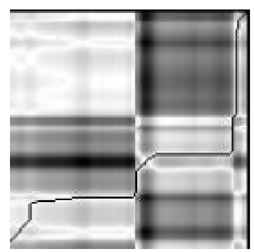
(d) F0 Contour of 11 Pashta Doubled



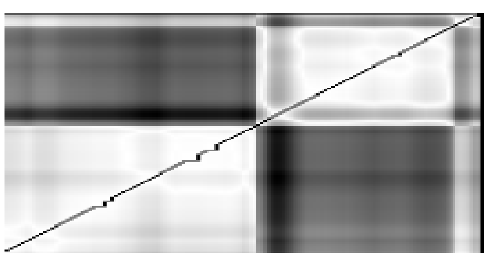
(a) DTW of 11 Pashta vs 11 Pashta



(b) DTW of 11 Pashta vs 42 Pashta



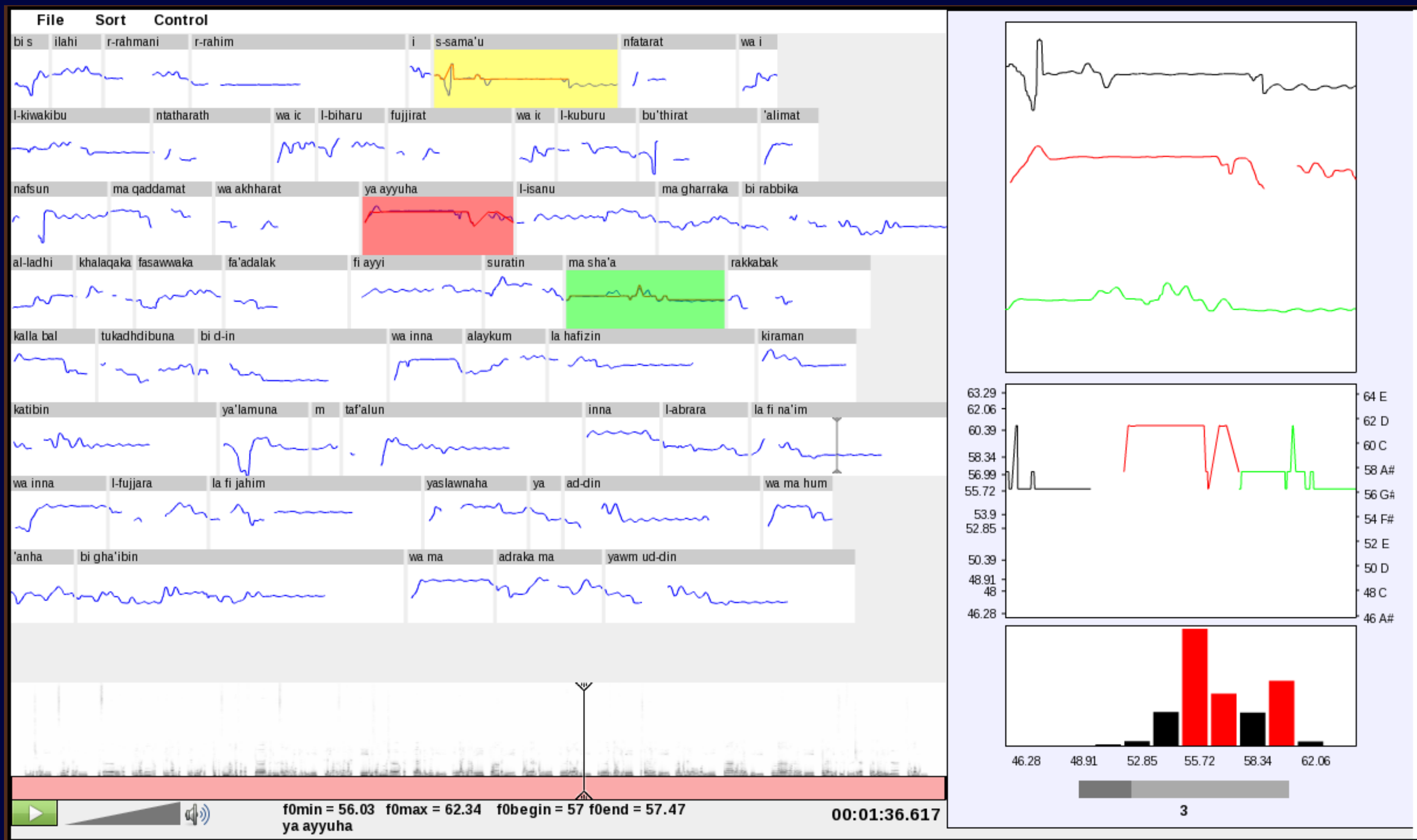
(c) DTW of 11 Pashta vs 18 Sof Pasuq



(d) DTW of 11 Pashta vs 11 Pashta Doubled

Cantillion

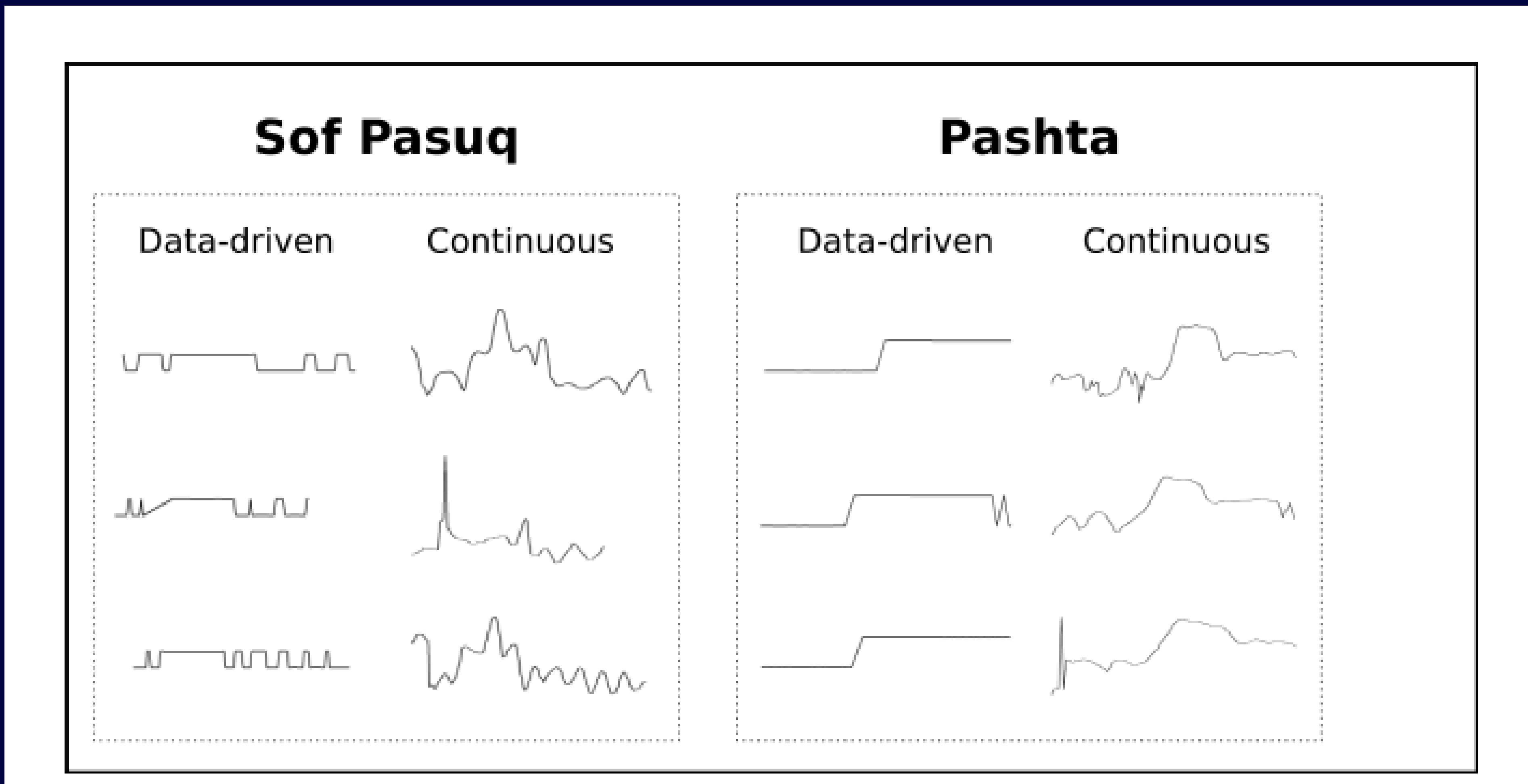
http://cantillion.sness.net



Retrieval

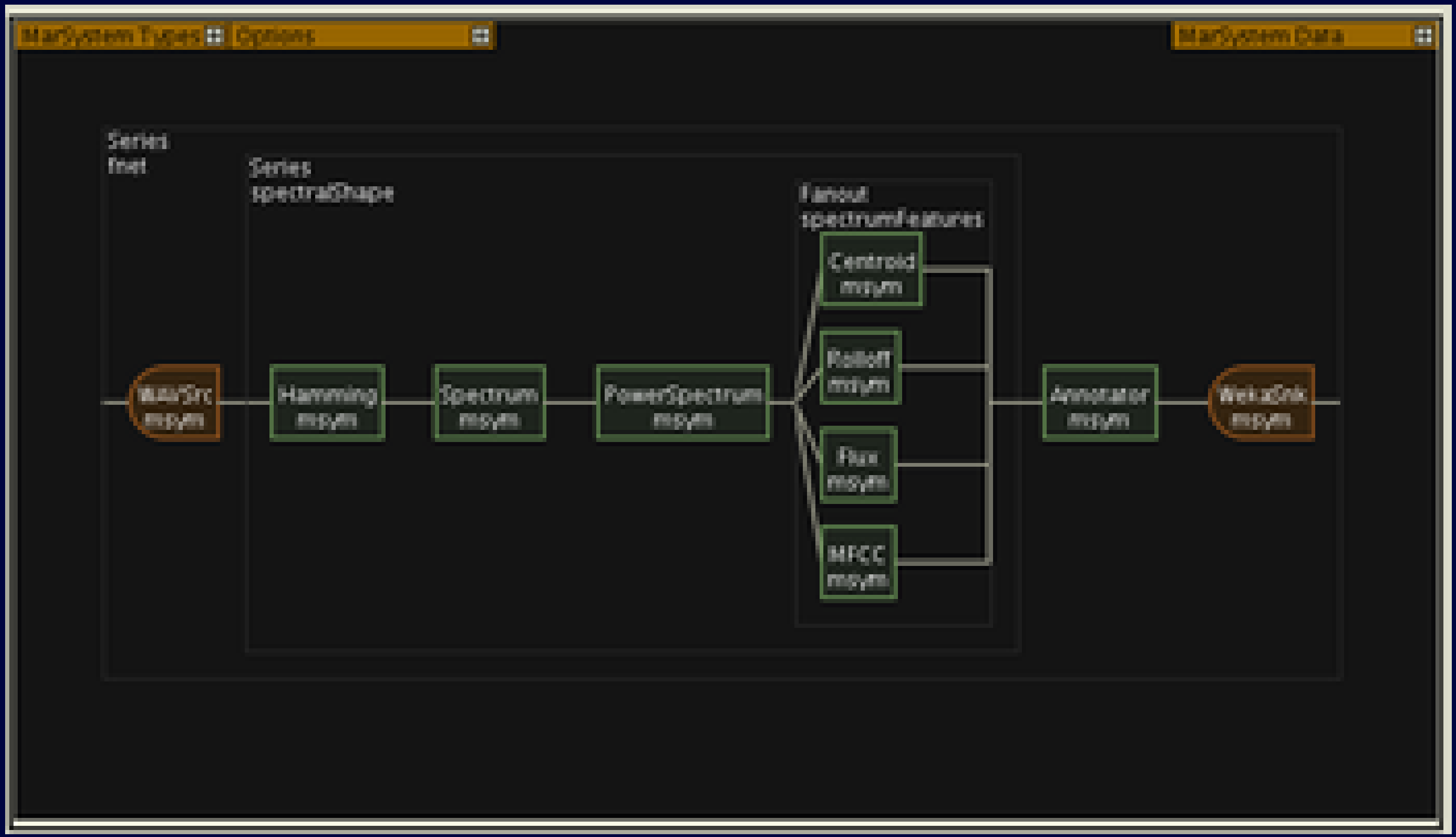
Gesture (Hungary)	Average Precision (Hungary)	Gesture (Morocco)	Average Precision (Morocco)
tipha	0.662	katon	0.453
pashta	0.647	mapah	0.347
mapah	0.641	tipha	0.303
katon	0.604	sofpasuq	0.285
etnachta	0.601	pashta	0.242
sofpasuq	0.591	merha	0.251
merha	0.537	etnachta	0.150
revia	0.372	zakef	0.125
zakef	0.201	revia	0.091
kadma	0.200	kadma	0.043

Retrieval at different levels of abstraction



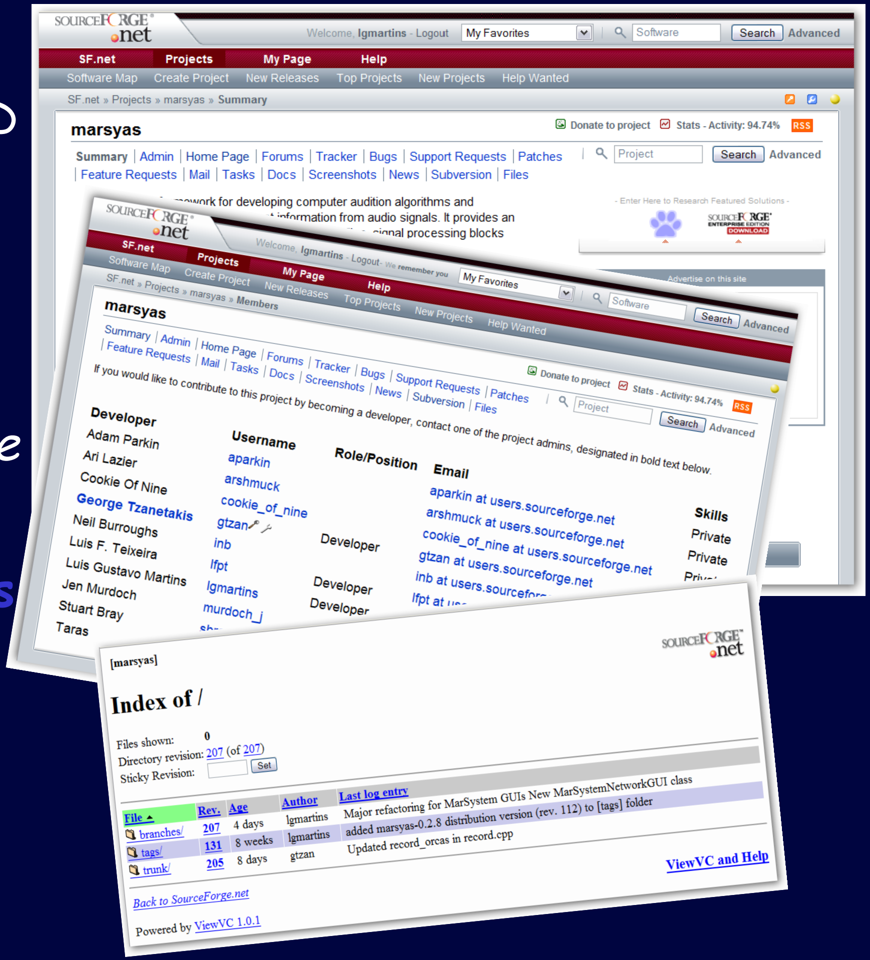
Marsyas Overview

- Software framework for audio analysis, synthesis and retrieval
  - Efficient and extensible framework design
    - specific emphasis on **Music Information Retrieval (MIR)**
    - **C++**, *OOP*
    - **Multiplatform** (Linux, MS Windows®, MacOSX®, …)
  - Provides a variety of building blocks for performing common audio tasks:
    - soundfile IO, audio IO, **signal processing** and **machine learning** modules
    - blocks can be combined into **data flow networks** that can be modified and controlled dynamically while they process data in soft **real-time**.


MarSystem Types
Options
MarSystem Data
Series
fnet
Series
spectralShape
Fanout
spectrumFeatures
Centroid
msym
Rolloff
msym
Flux
msym
MFCC
msym
WAVSrc
msym
Hamming
msym
Spectrum
msym
PowerSpectrum
msym
Annotator
msym
WekaSnk
msym

Marsyas Overview

- Marsyas Brief History
  - 1998 ~2000
    - Created by **George Tzanetakis** during his PhD
  - 2000 ~2002
    - Marsyas 0.1
      - First *stable* revisions of the toolkit
      - Distributions hosted at *SourceForge*
      - Creation of a developer community
        - User and Developer **Mailing lists**
  - 2002 ~ …
    - **Marsyas 0.2**
      - Major framework revision
      - SourceForge **SubVersion**



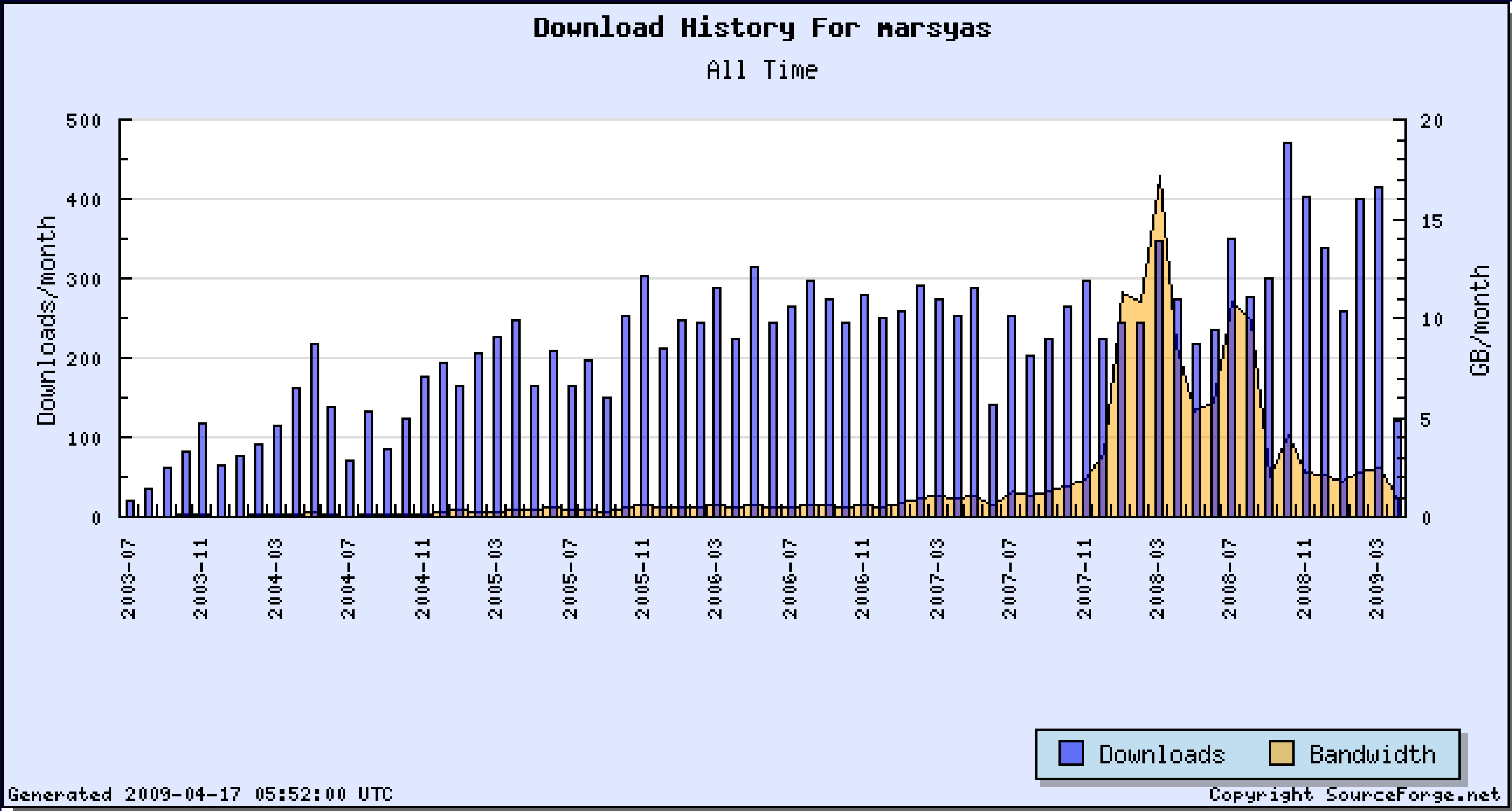
Related Work Context

- Open Source frameworks
  - CLAM (http://clam.iua.upf.edu/)
  - STK (http://ccrma.stanford.edu/software/stk/)
  - Chuck (http://chuck.cs.princeton.edu/)
  - PureData (Pd) (http://crca.ucsd.edu/~msp/software.html)
  - Open Sound Control (OSC) (http://cnmat.berkeley.edu/OpenSoundControl/)

Commercial toolkits

- MAX/MSP® (http://www.cycling74.com/)

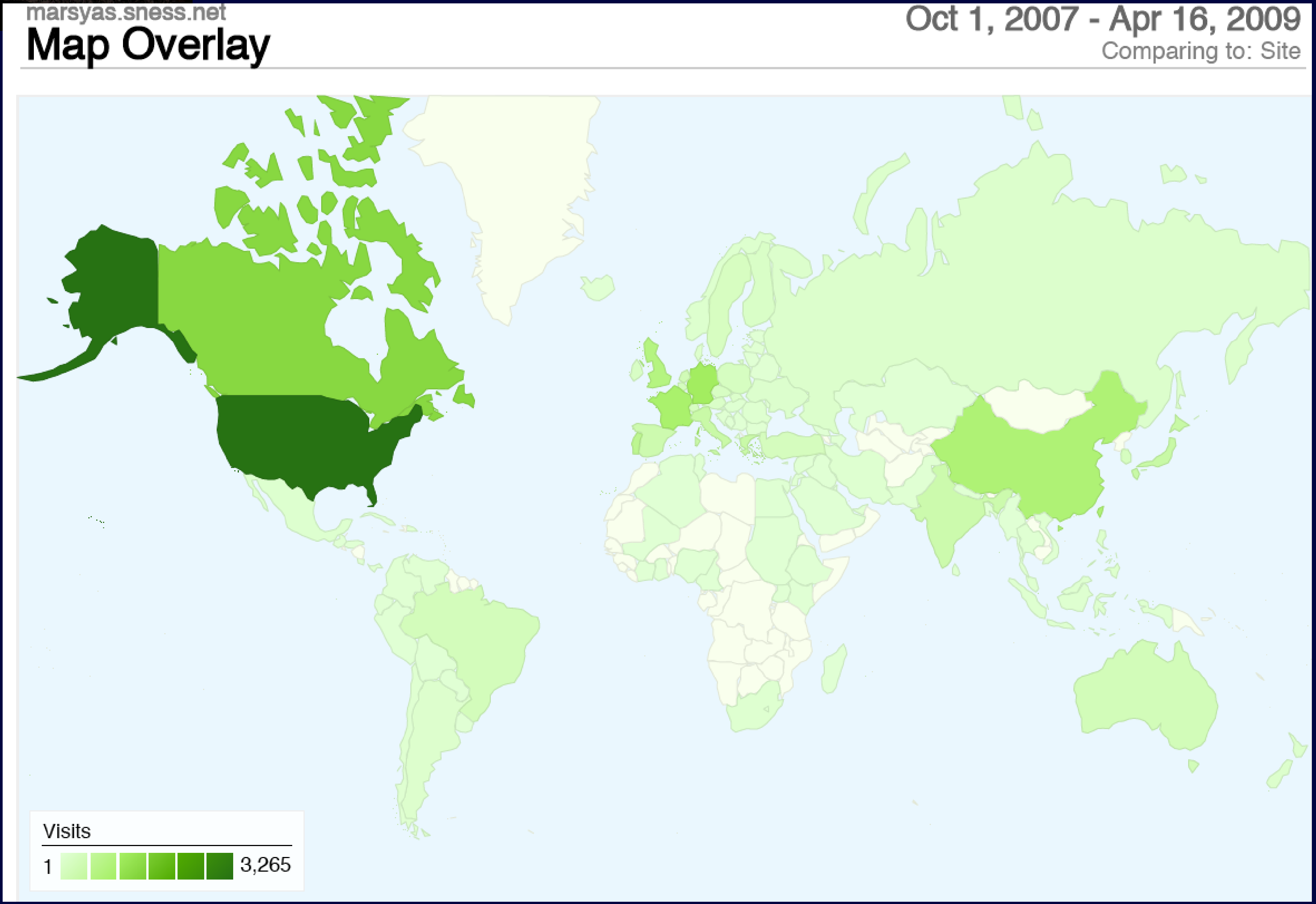
Statistics


Download History For marsyas
All Time
Downloads/month
GB/month
500
400
300
200
100
0
20
15
10
5
0
2003-07
2003-11
2004-03
2004-07
2004-11
2005-03
2005-07
2005-11
2006-03
2006-07
2006-11
2007-03
2007-07
2007-11
2008-03
2008-07
2008-11
2009-03
Downloads
Bandwidth
Generated 2009-04-17 05:52:00 UTC
Copyright SourceForge.net

Statistics


marsyas.sness.net
Visitors Overview
Oct 16, 2007 - Apr 15, 2009
Comparing to: Site
Visitors
800
400
800
400
Oct 16, 2007 - Oct 31, 2007
Jan 1, 2008 - Jan 31, 2008
Apr 1, 2008 - Apr 30, 2008
Jul 1, 2008 - Jul 31, 2008
Oct 1, 2008 - Oct 31, 2008
10,511 people visited this site
16,440 Visits
10,511 Absolute Unique Visitors
52,071 Pageviews
3.17 Average Pageviews
00:02:53 Time on Site

Statistics



marsyas.sness.net
Map Overlay
Oct 1, 2007 - Apr 16, 2009
Comparing to: Site
Visits
1
3,265

Statistics

16,445 visits came from 112 countries/territories

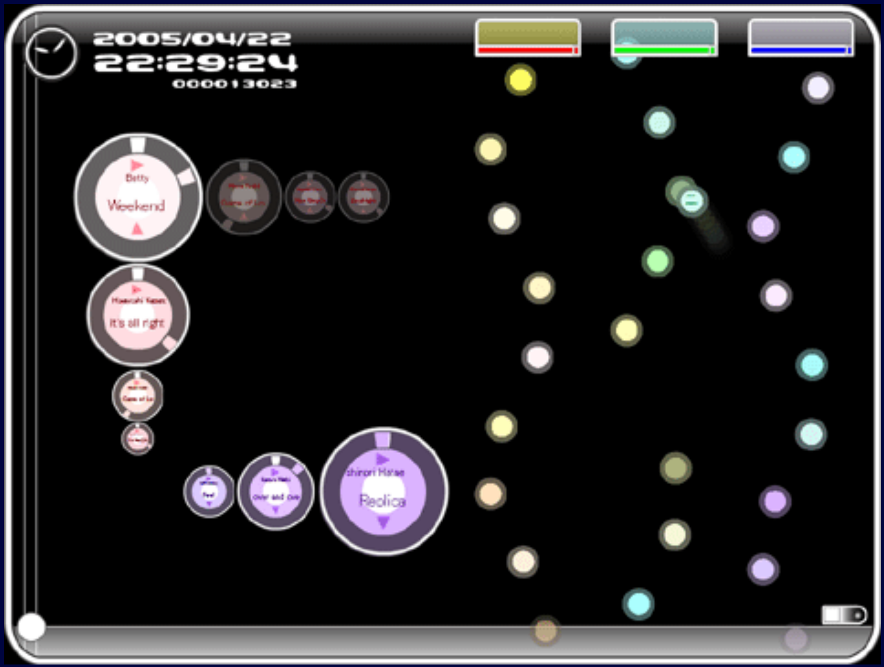
Site Usage

Visits	Pages/Visit	Avg. Time on Site	% New Visits	Bounce Rate
16,445 % of Site Total: **100.00%**	**3.17** Site Avg: **3.17** (0.00%)	**00:02:53** Site Avg: **00:02:53** (0.00%)	**63.98%** Site Avg: **63.92%** (0.10%)	**38.74%** Site Avg: **38.74%** (0.00%)

Country/Territory	Visits	Pages/Visit	Avg. Time on Site	% New Visits	Bounce Rate
United States	3,265	2.99	00:02:18	78.87%	44.90%
Canada	1,467	3.30	00:02:55	49.76%	31.49%
Germany	1,123	3.55	00:03:17	63.05%	34.91%
France	1,007	3.10	00:02:33	49.85%	36.25%
China	919	3.11	00:04:17	51.36%	31.56%
United Kingdom	805	3.42	00:02:57	71.55%	37.76%
Taiwan	698	3.29	00:03:31	40.69%	39.11%
Portugal	649	3.05	00:03:26	43.14%	44.99%
Japan	451	3.00	00:02:57	61.42%	47.67%
Italy	442	3.89	00:03:10	72.40%	30.09%

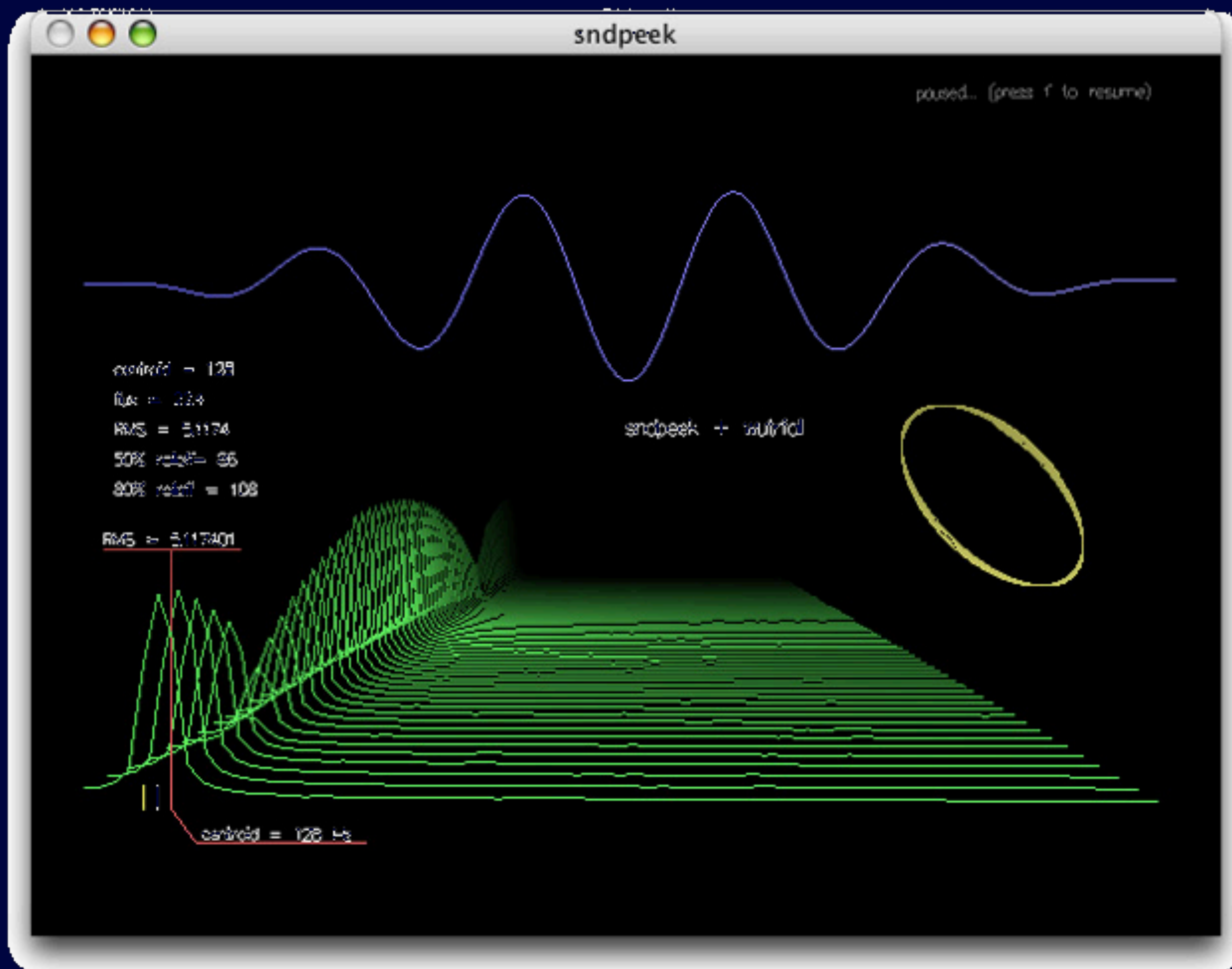
Users & Applications

- Musicream (Masataka Goto)
  - Music playback system with similarity capabilities
    - Uses Marsyas as its music similarity engine



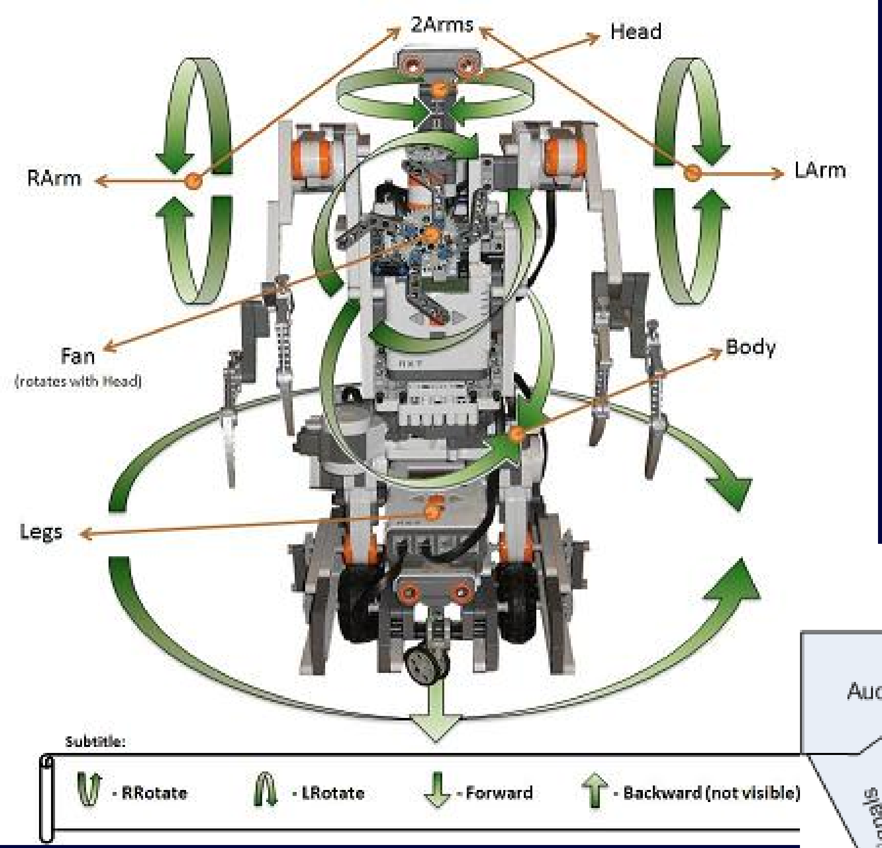
http://staff.aist.go.jp/m.goto/Musicream/

Users & Applications

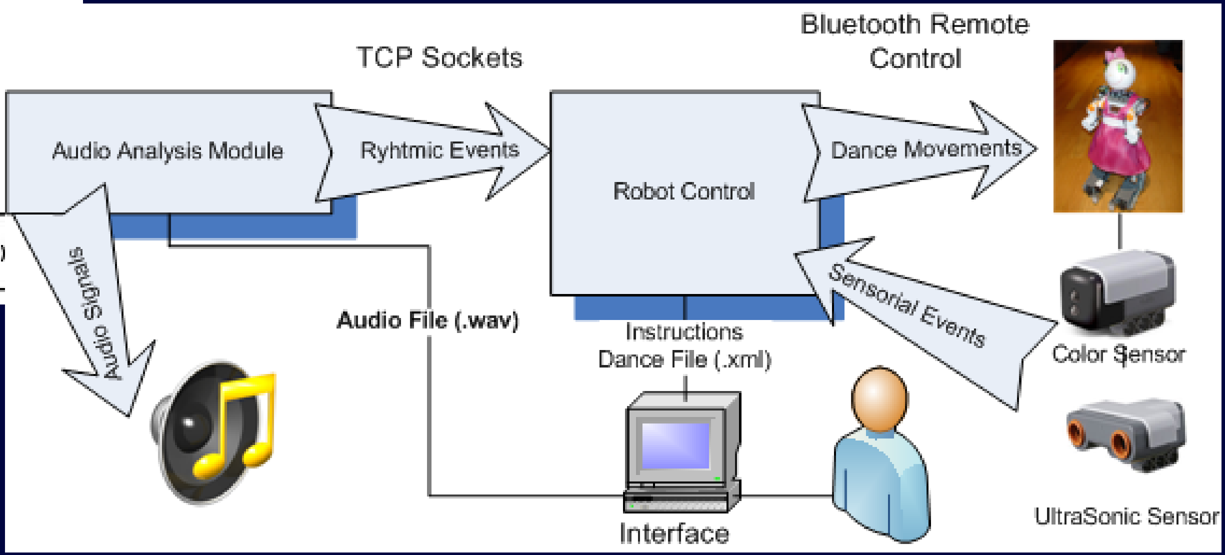


http://www.cs.princeton.edu/sound/software/sndpeek/

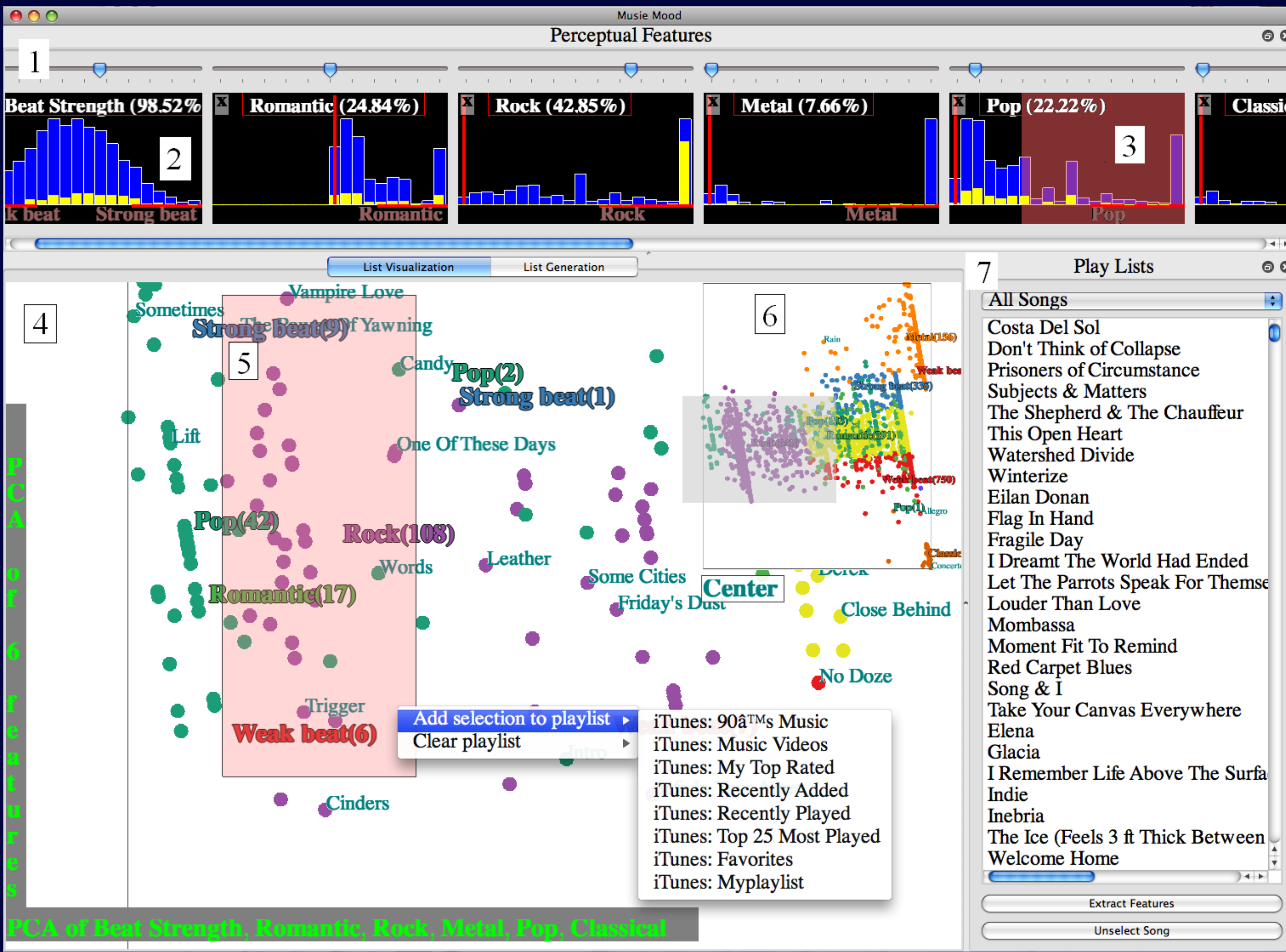
Users and Applications



Joao Oliveira, Fabien Gouyon, and Luis Paulo Reis
INESC Porto, Portugal



Users and Applications

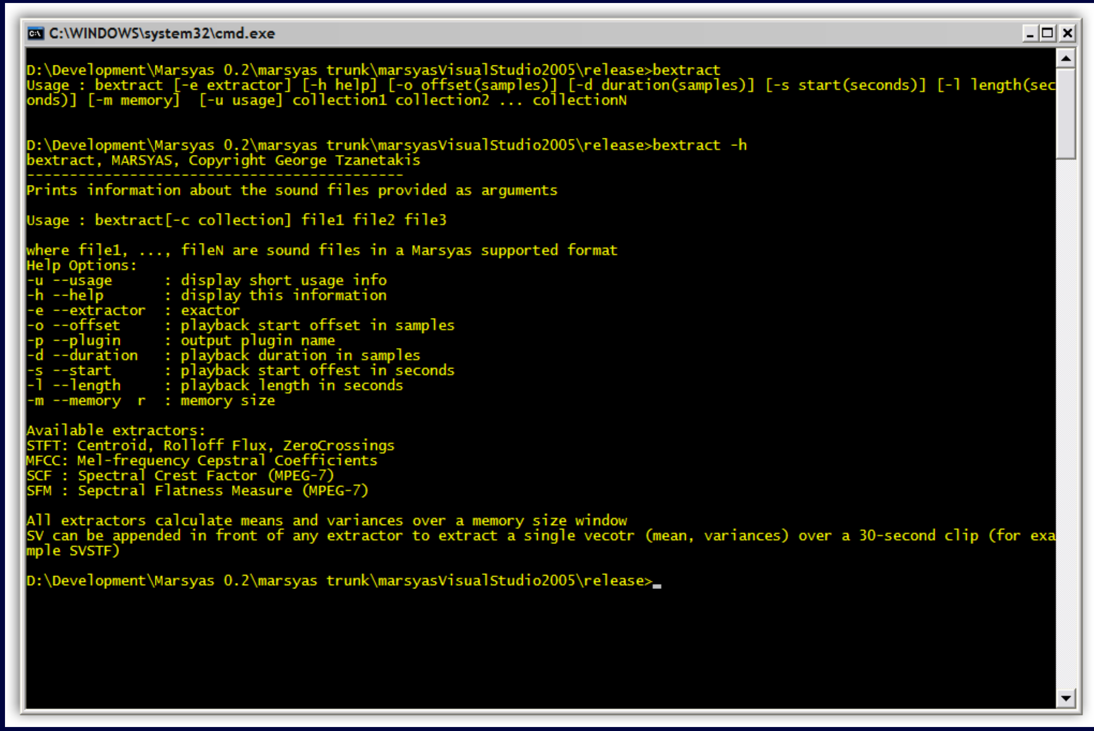


MusieMood
Vladimir Kim
Steven Bergner
Torsten Moller

Simon Fraser Univ
Vancouver, Canada

Usage Scenarios

- Marsyas command line tools
  - Efficient
  - Execute in real-time (when applied)
  - No library dependencies
  - Tools and examples:
    - **`sfplay`**
    - **`bextract`**
    - **`phasevocoder`**
    - **`sfplugin`**
    - …



```
C:\WINDOWS\system32\cmd.exe
D:\Development\Marsyas 0.2\marsyas trunk\marsyasVisualStudio2005\release>bextract
Usage : bextract [-e extractor] [-h help] [-o offset(samples)] [-d duration(samples)] [-s start(seconds)] [-l length(sec
onds)] [-m memory]  [-u usage] collection1 collection2 ... collectionN


D:\Development\Marsyas 0.2\marsyas trunk\marsyasVisualStudio2005\release>bextract -h
bextract, MARSYAS, Copyright George Tzanetakis
--------------------------------------------
Prints information about the sound files provided as arguments

Usage : bextract[-c collection] file1 file2 file3

where file1, ..., fileN are sound files in a Marsyas supported format
Help Options:
-u --usage      : display short usage info
-h --help       : display this information
-e --extractor  : exactor
-o --offset     : playback start offset in samples
-p --plugin     : output plugin name
-d --duration   : playback duration in samples
-s --start      : playback start offest in seconds
-l --length     : playback length in seconds
-m --memory  r  : memory size

Available extractors:
STFT: Centroid, Rolloff Flux, ZeroCrossings
MFCC: Mel-frequency Cepstral Coefficients
SCF : Spectral Crest Factor (MPEG-7)
SFM : Sepctral Flatness Measure (MPEG-7)

All extractors calculate means and variances over a memory size window
SV can be appended in front of any extractor to extract a single vecotr (mean, variances) over a 30-second clip (for exa
mple SVSTF)

D:\Development\Marsyas 0.2\marsyas trunk\marsyasVisualStudio2005\release>
```

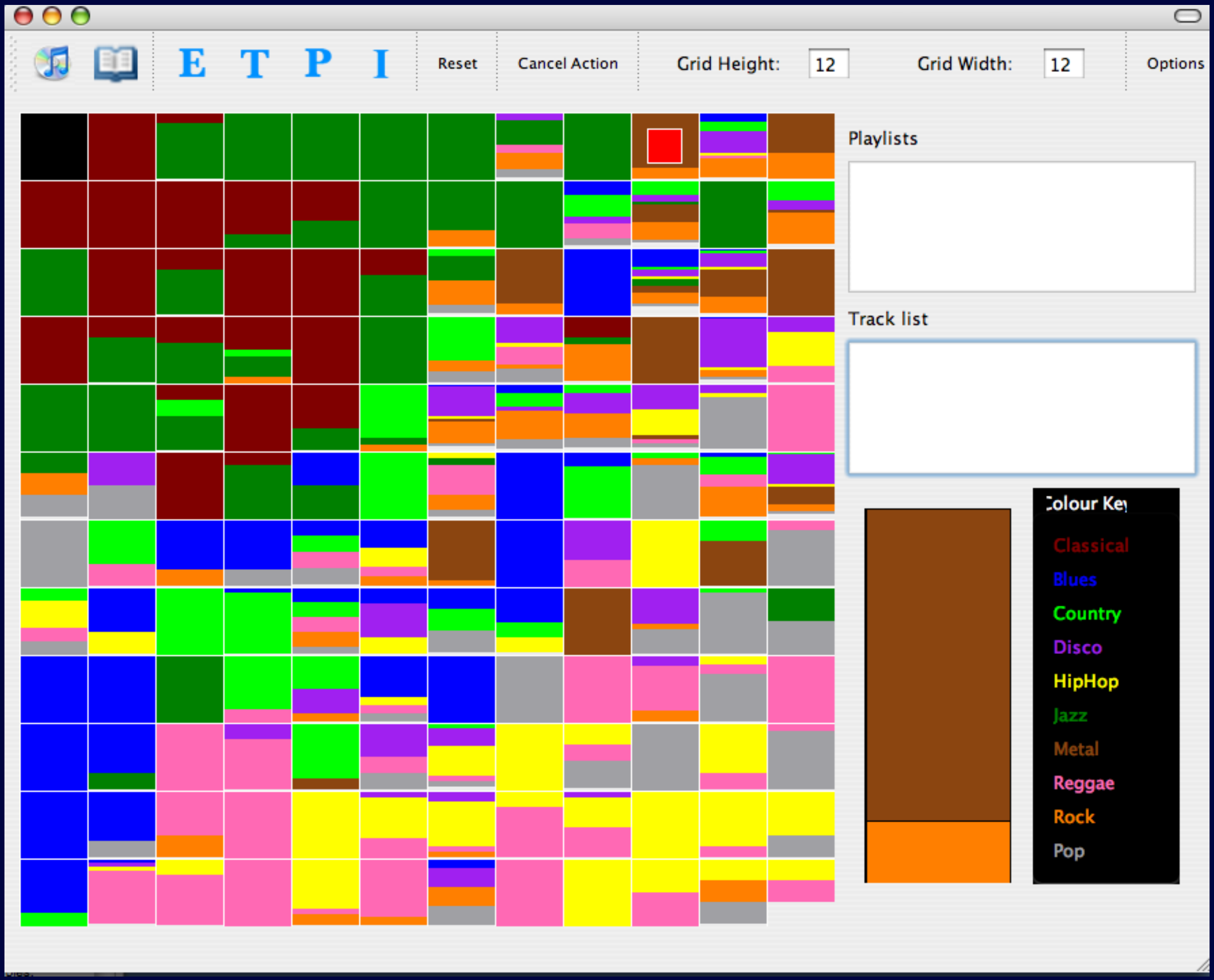
Usage Scenarios

- Some examples:

```
> bextract -e STFTMFCC music.mf speech.mf -p ms.mpl -w myweka.arff
```

```
> sfplugin -p ms-mpl unknownAudioSignal.wav
```

Usage Scenarios

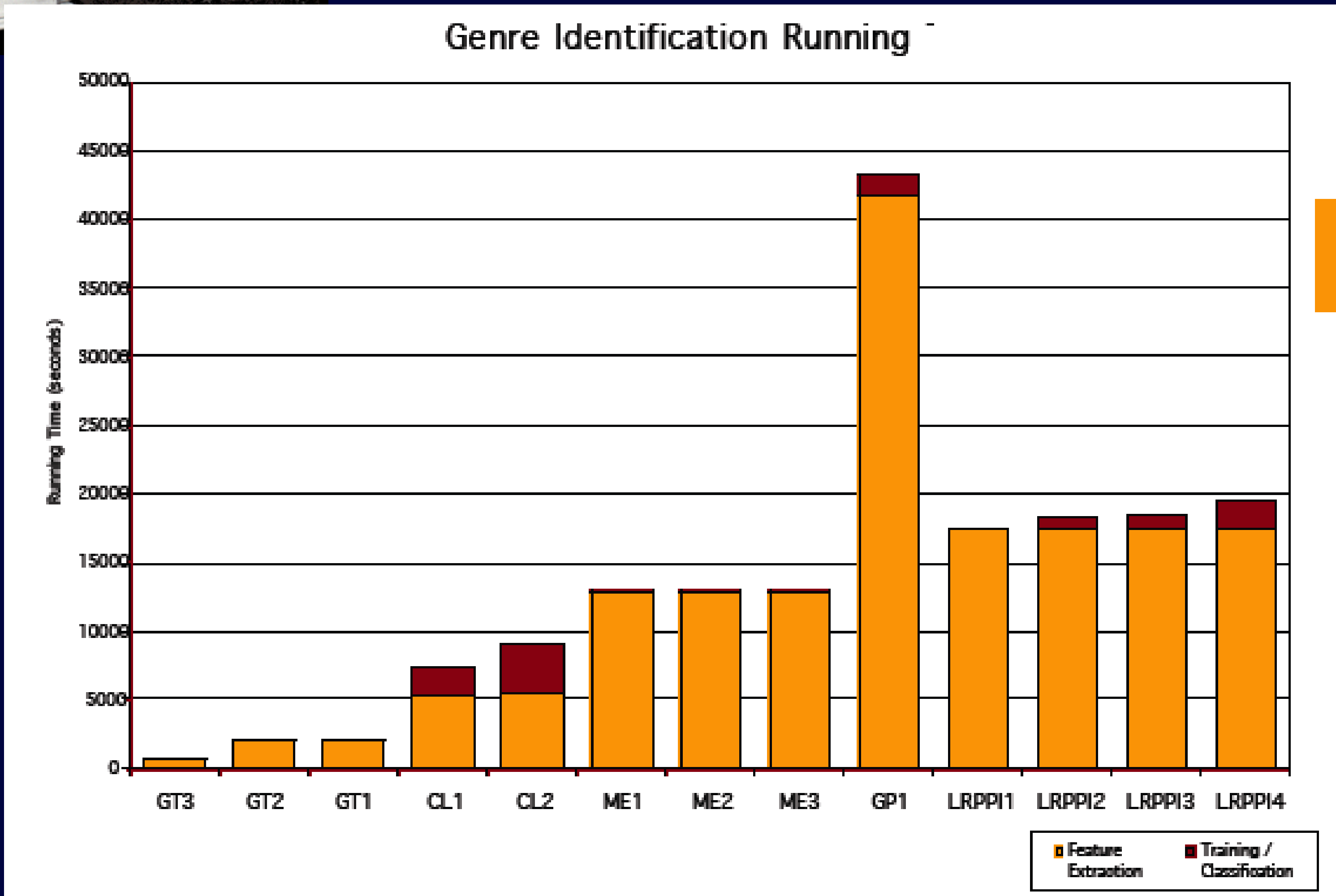


MarGrid2
MarPlayer
MarPhasevocoder
MarNetworkWidget
…

MIREX 2008 results

Artist Identification	47.65	47.25	47.16	43.47	35.42	35.27	33.68	33.2	32.52	29.87	1.11		
	ME1	ME3	ME2	GT2	LRPPI1	GT3	GT1	LRPPI2	LRPPI4	LRPPI3	GP2		
Composer Identification	53.25	53.1	52.89	48.99	45.82	43.81	39.54	39.47	39.43	37.48	34.13		
	ME1	ME2	ME3	GP1	GT2	GT3	LRPPI4	GT1	LRPPI2	LRPPI3	LRPPI1		
Genre Classification	66.41	65.62	65.41	65.3	65.2	65.06	64.71	63.9	63.39	62.26	62.04	60.84	60.46
	GT2	GT3	ME1	ME2	ME3	LRPPI1	GT1	GP1	CL2	LRPPI2	CL1	LRPPI3	LRPPI4
Latin Genre Classification	65.17	64.04	62.72	62.23	59.55	59	58.64	54.99	54.7	54.15	53.79	53.67	53.65
	CL1	CL2	GP1	LRPPI2	LRPPI3	LRPPI4	LRPPI1	ME3	ME2	ME1	GT2	GT3	GT1
Music Mood Classification	63.67	58.2	56	55.5	55.5	55	54.5	52.5	50.33	50	49.83	49.67	30.33
	GP1	GT3	LRPPI1	LRPPI4	LRPPI2	GT1	LRPPI3	GT2	ME1	ME2	KL	ME3	HW

MIREX 2008 Results



Genre

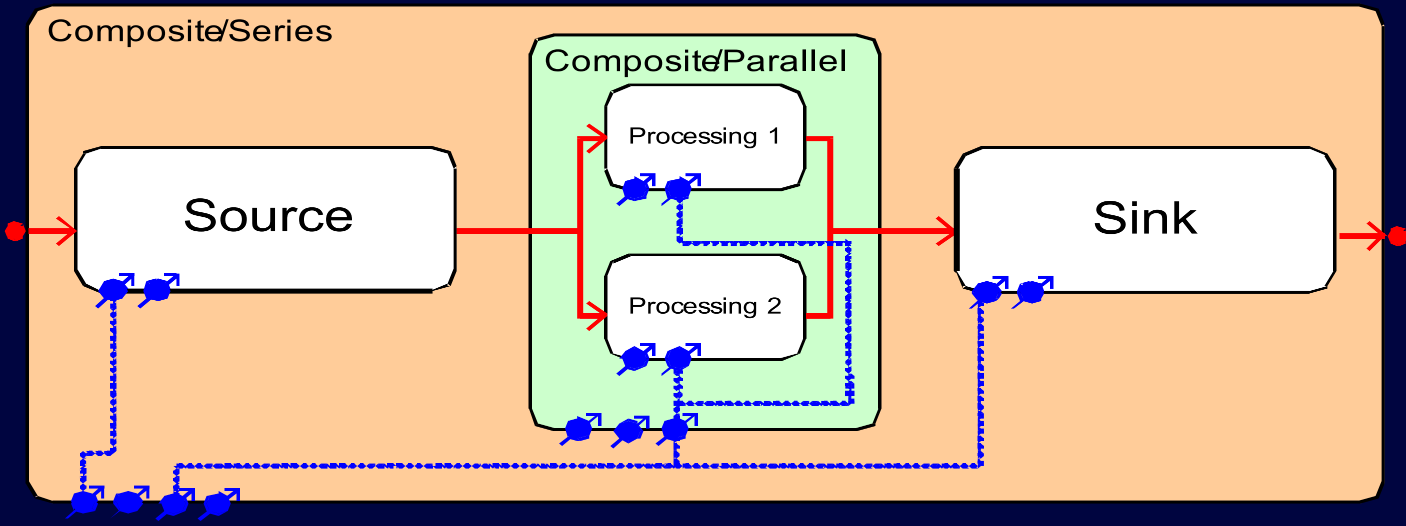
Group	Feature Extraction	Training / Classification
GT3	0:12	0:01
GT2	0:35	–
GT1	0:36	–
CL1	1:29	0:33
CL2	1:31	1:01
ME1	3:35	0:02
ME2	3:35	0:02
ME3	3:35	0:02
GP1	11:37	0:25
LRPPI1	28:50:00	0:02
LRPPI2	28:50:00	0:17
LRPPI3	28:50:00	0:20
LRPPI4	28:50:00	0:35

Architecture

- Biggest challenge: expressivity without sacrificing efficiency
- Compile-time
  - Definition of processing blocks
- Run-time
  - Assembling network
  - Passing data through it
  - Changing behavior through controls

Architecture

- Marsyas 0.2
  - New dataflow model of audio computation

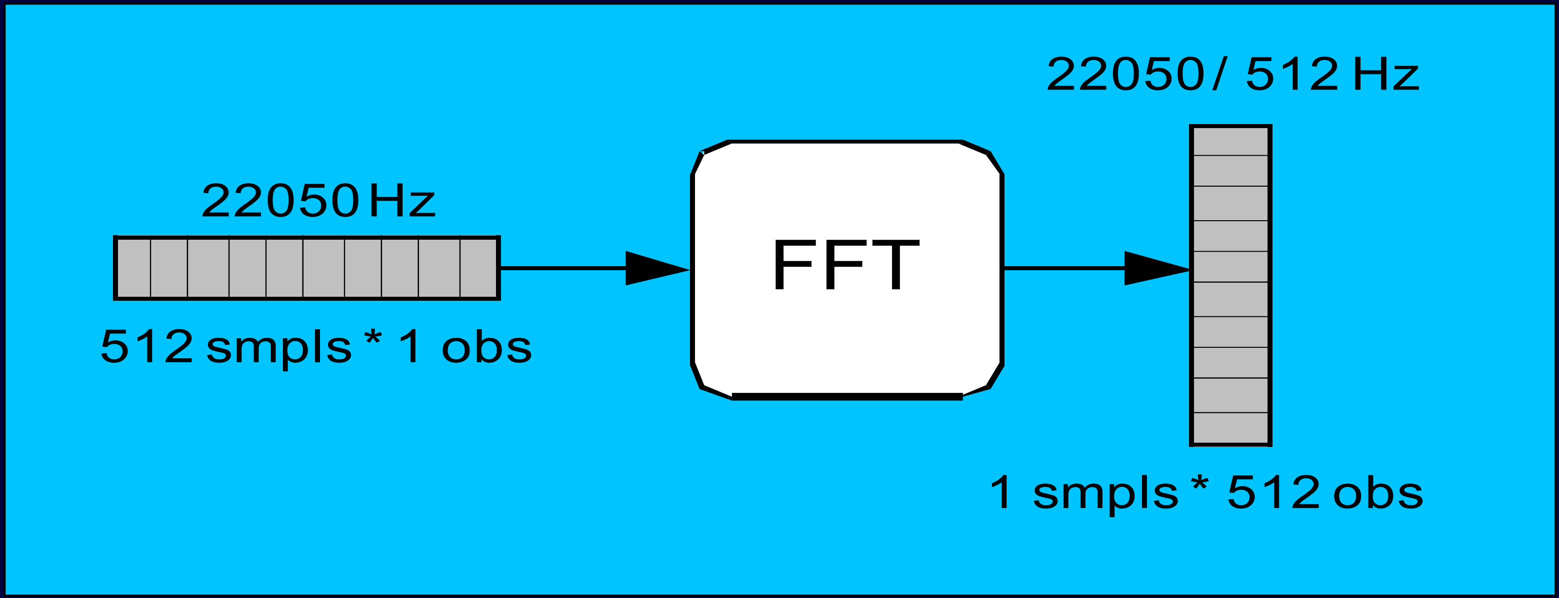


hierarchical messaging system used to control the dataflow network (inspired on *Open Sound Control (OSC)*)

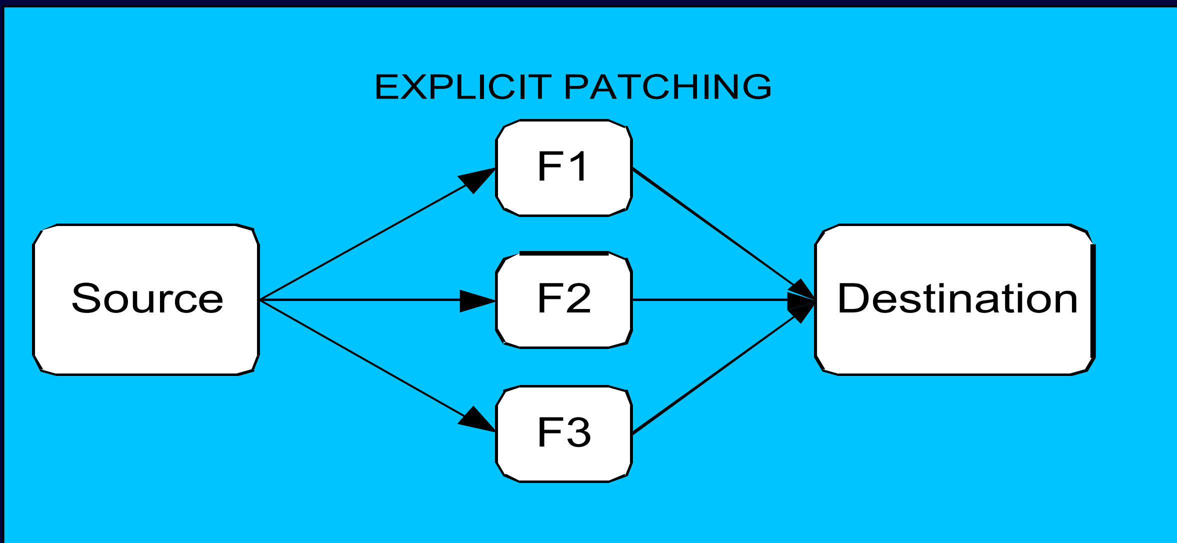
  - general matrices instead of 1-D arrays as data

Architecture

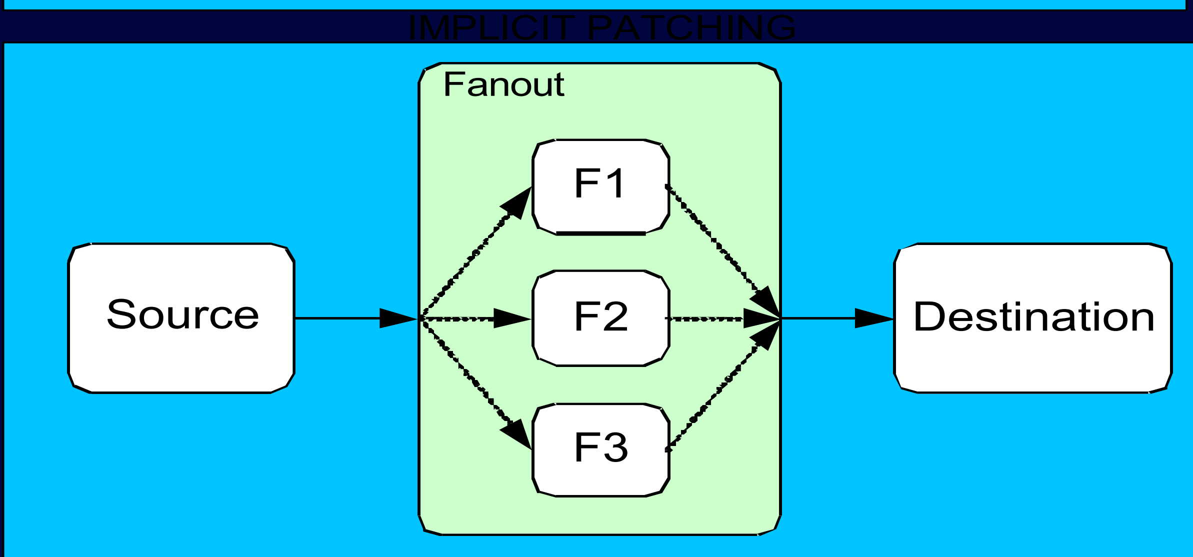
- MarSystem Slices



Architecture

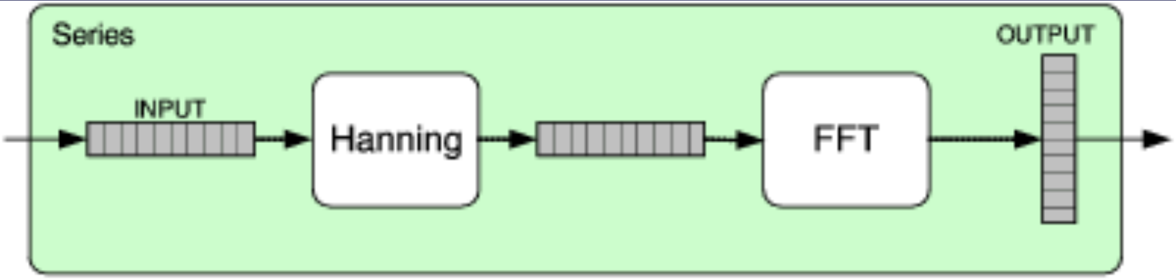


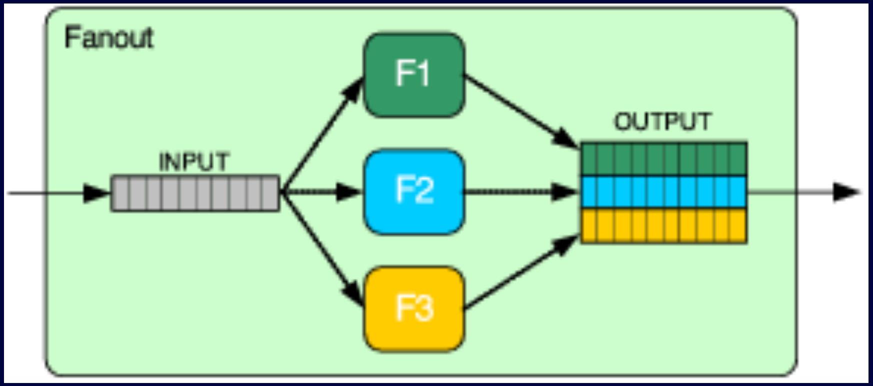

```
# EXPLICIT PATCHING:
    source, F1, F2, F3, destination;
# Connect the in/out ports
    connect(source, F1);
    connect(source, F2);
    connect(source, F3);
    connect(F1, destination);
    connect(F2, destination);
    connect(F3, destination);
```

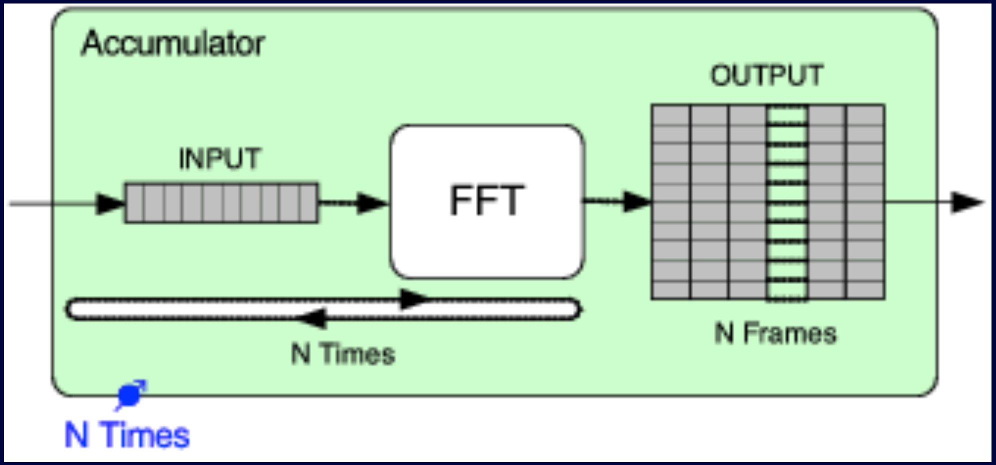



```
# IMPLICIT PATCHING
    source, F1, F2, F3, destination;
    Fanout mix;
    mix.add([F1, F2, F3]);
    Series net;
    net.add(source,mix,destination);
```

Architecture



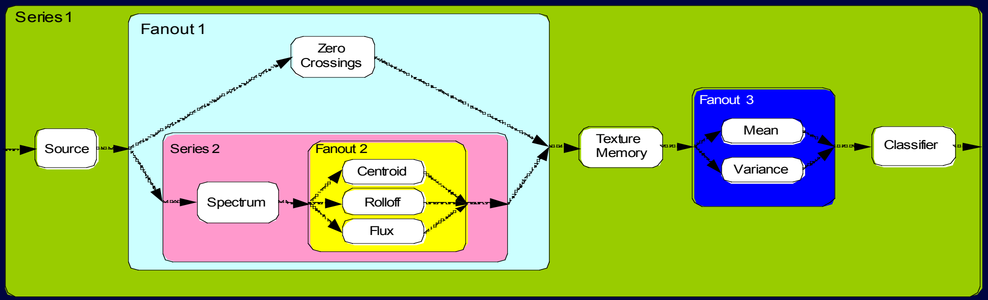




Architecture

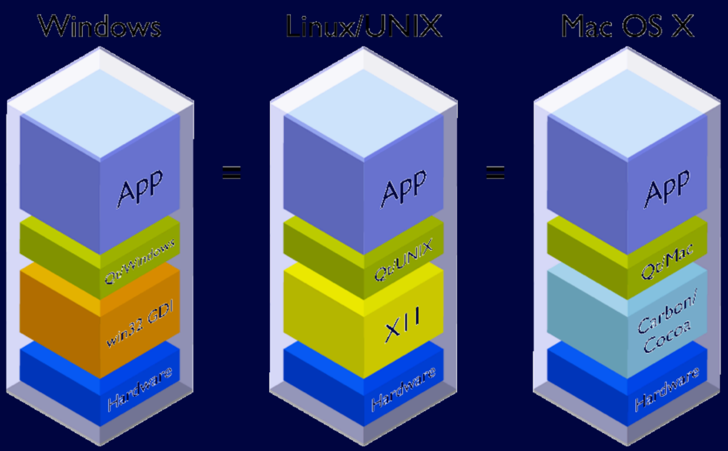
➢ Feature Extraction using Implicit Patching

```
MarSystemManager mng;

MarSystem* Series1 = mng.create("Series", “Series1");
MarSystem* Fanout1 = mng.create(“Fanout", “Fanout1");
MarSystem* Series2 = mng.create("Series", “Series2");
MarSystem* Fanout2 = mng.create(“Fanout”, “Fanout2”);
MarSystem* Fanout3 = mng.create(“Fanout”, “Fanout3”);

Fanout3->addMarSystem(mng.create(“Mean”, “Mean”));
Fanout3->addMarSystem(mng.create(“Variance”, “Variance”));

Fanout2->addMarSystem(mng.create(“Centroid”, “Centroid”));
Fanout2->addMarSystem(mng.create(“RollOff”, “Rolloff”));
Fanout2->addMarSystem(mng.create(“Flux”, “Flux”);

Series2->addMarSystem(mng.create(“Spectrum”, “Spectrum”);
Series2->addMarSystem(Fanout2);

Fanout1->addMarSystem(mng.create(“ZeroCrossings”, “ZeroCrossings”);
Fanout1->addMarSystem(Series2);

Series1->addMarSystem(mng.create("SoundFileSource",“Source"));
Series1->addMarSystem(Fanout1);
Series1->addMarSystem(mng.create(“Memory”, “TextureMemory”));
Series1->addMarSystem(Fanout3);
Series1->addMarSystem(mng.create(“classifier”, “Classifier”));
```

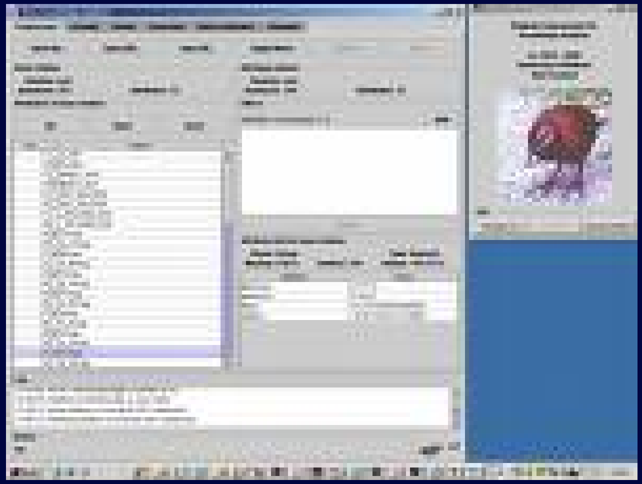


Interoperability

- Python (Ruby, Java) bindings
- Qt 4.x toolkit
- MATLAB
- Weka
- MIDI
- Open Sound Control (OSC)
- Max/MSP external
- VAMP plugin



Qt4® is available as GPL open source code for all platforms



Industrial Collaborations

- Marsyas is licensed under the GNU Public License
  - Anyone can download and modify the source code for free
  - Software that uses Marsyas source code must also be GPL
  - Copyright remains with the author

Modes of collaboration

- Modes of collaboration with industry
  - Consulting
  - Prototype and rewrite
  - Internal tool/batch processing
  - Based on proprietary data
  - Commercial Licensing

How can you help ?

- Encourage your students to use Marsyas
- Encourage your students to work on open source projects
- Donate money to open source projects - even small amounts can make a huge difference
- Do not hesitate to get involved in the community of a project
- Hire open source developers for particular tasks

Summary

- Marsyas is a free software framework for research in audio processing
- It has been in development for 10 years and has steadily been growing
- Several academic and commercial projects have used Marsyas
- Open source development in academic environments is challenging but has its rewards
- Any questions ?