

# Tutorial: Music Signal Processing

Mark Plumbley and Simon Dixon

`{mark.plumbley, simon.dixon}@eecs.qmul.ac.uk`

`www.elec.qmul.ac.uk/digitalmusic`

Centre for Digital Music  
Queen Mary University of London

IMA Conference Mathematics in Signal Processing

17 December 2012

# Overview

- Introduction and Music fundamentals
- Pitch estimation and Music Transcription
- Temporal analysis: Onset Detection and Beat Tracking
- Conclusions

## Acknowledgements:

This includes the work of many others, including Samer Abdallah, Juan Bello, Matthew Davies, Anssi Klapuri, Matthias Mauch, Andrew Robertson, ...

Plumbley is supported by an EPSRC Leadership Fellowship

# Introduction: Music Fundamentals

# Pitch and Melody

- Pitch: the perceived (fundamental) frequency  $f_0$  of a musical note
  - related to the frequency spacing of a harmonic series in the frequency-domain representation of the signal
  - perceived logarithmically
  - one octave corresponds to a doubling of frequency
  - octaves are divided into 12 semitones
  - semitones are divided into 100 cents
- Melody: a sequence of pitches, usually the "tune" of a piece of music
  - when notes are structured in succession so as to make a unified and coherent whole
  - melody is perceived without knowing the actual notes involved, using the *intervals* between successive notes
  - melody is translation (transposition) invariant (in log domain)

# Harmony

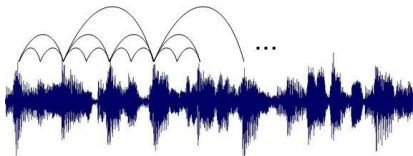
- Harmony: refers to relationships between simultaneous pitches (chords) and sequences of chords
- Harmony is also perceived relatively (i.e. as intervals)
- Chord: two or more notes played simultaneously
- Common intervals in western music:
  - octave (12 semitones,  $f_0$  ratio of 2)
  - perfect fifth (7 semitones,  $f_0$  ratio approximately  $\frac{3}{2}$ )
  - major third (4 semitones,  $f_0$  ratio approximately  $\frac{5}{4}$ )
  - minor third (3 semitones,  $f_0$  ratio approximately  $\frac{6}{5}$ )
- Consonance: fundamental frequency ratio  $\frac{f_A}{f_B} = \frac{m}{n}$ , where  $m$  and  $n$  are small positive integers:
  - Every  $n$ th partial of sound A overlaps every  $m$ th partial of sound B

# Timbre / Texture

- Timbre: the properties distinguishing two notes of the same pitch, duration and intensity (e.g. on different instruments)
- “Colour” or tonal quality of a sound
- Determined by the following factors:
  - instrument
  - register (pitch)
  - dynamic level
  - articulation / playing technique
  - room acoustics, recording conditions and postprocessing
- In signal processing terms:
  - distribution of amplitudes of the composing sinusoids, and their changes over time
  - i.e. the time-varying spectral envelope (independent of pitch)

# Rhythm: Meter and Metrical Structure

- A pulse is a regularly spaced sequence of accents (beats)
- Metrical structure: hierarchical set of pulses
- Each pulse defines a metrical level



- Time signature: indicates relationships between metrical levels
  - the number of beats per measure
  - sometimes also an intermediate level (grouping of beats)
- Performed music only fits this structure approximately
- *Beat tracking* is concerned with finding this metrical structure

# Expression

- Music is performed expressively by employing small variations in one or more attributes of the music, relative to an expressed or implied basic form (e.g. the score)
- Rhythm: tempo changes, timing changes, articulation, embellishment
- Melody: ornaments, embellishment, vibrato
- Harmony: chord extensions, substitutions
- Timbre: special playing styles (e.g. sul ponto, pizzicato)
- Dynamics: crescendo, sforzando, tremolo
- Audio effects: distortion, delays, reverberation
- Production: compression, equalisation
- ... mostly beyond the scope of current automatic signal analysis



# High-level (Musical) Knowledge

- Human perception of music is strongly influenced by knowledge and experience of the musical piece, style and instruments, and of music in general
- Likewise the complexity of a musical task is related to the level of knowledge and experience, e.g.:
  - Beat following: we can all tap to the beat ...
  - Melody recognition: ... and recognise a tune ...
  - Genre classification: ... or jazz, rock, or country ...
  - Instrument recognition: ... or a trumpet, piano or violin ...
  - Music transcription: for expert musicians — often cited as the "holy grail" of music signal analysis
- Signal processing systems also benefit from encoded musical knowledge

# Pitch Estimation and Automatic Music Transcription

# Music Transcription

- Aim: describe music signals at the note level, e.g.
  - Find what notes were played in terms of discrete pitch, onset time and duration (*wav-to-midi*)
  - Cluster the notes into instrumental sources (*streaming*)
  - Describe each note with precise parameters so that it can be resynthesised (*object coding*)
- The difficulty of music transcription depends mainly on the number of simultaneous notes
  - *monophonic* (one instrument playing one note at a time)
  - *polyphonic* (one or several instruments playing multiple simultaneous notes)
- Here we limit transcription to multiple pitch detection
- A full transcription system would also include:
  - recognition of instruments
  - rhythmic parsing
  - key estimation and pitch spelling
  - layout of notation

# Pitch and Harmonicity

- Pitch is usually expressed on the *semitone* scale, where the range of a standard piano is from A0 (27.5 Hz, MIDI note 21) to C8 (4186 Hz, MIDI note 108)
- Non-percussive instruments usually produce notes with *harmonic* sinusoidal partials, i.e. with frequencies:

$$f_k = kf_0$$

where  $k \geq 1$  and  $f_0$  is the *fundamental frequency*

- Partial produced by struck or plucked string instruments are slightly *inharmonic*:

$$f_k = kf_0 \sqrt{1 + Bk^2} \text{ with } B = \frac{\pi^3 Ed^4}{64TL^2}$$

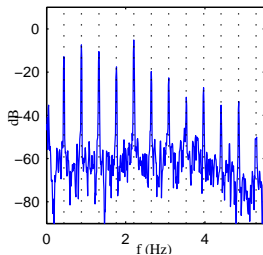
for a string with Young's modulus  $E$  (inverse elasticity), diameter  $d$ , tension  $T$  and length  $L$

# Harmonicity

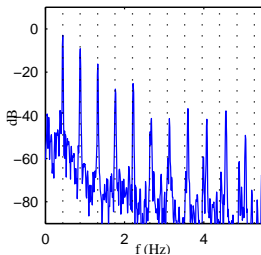
- Magnitude spectra for 3 acoustic instruments playing the note A4 ( $f_0 = 440$  Hz)



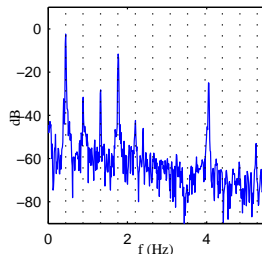
violin



piano



vibraphone



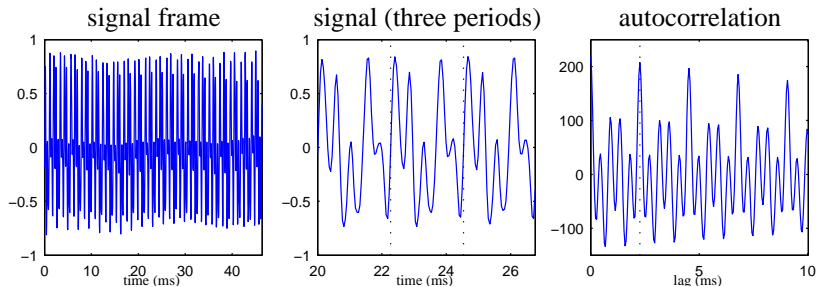
- Note: the frequency axis should be in kHz

# Autocorrelation-Based Pitch Estimation

# Autocorrelation

The *Auto-Correlation Function* (ACF) of a signal frame  $x(t)$  is

$$r(\tau) = \frac{1}{T} \sum_{t=0}^{T-\tau-1} x(t)x(t+\tau)$$



# Autocorrelation

- Generally, for a monophonic signal, the highest peak of the ACF for positive lags  $\tau$  corresponds to the fundamental period  $\tau_0 = \frac{1}{f_0}$
- However other peaks always appear:
  - peaks of similar amplitude at integer multiples of the fundamental period
  - peaks of lower amplitude at simple rational multiples of the fundamental period



# YIN Pitch Estimator

- The ACF decreases for large values of  $\tau$ , leading to inverse octave errors when the target period  $\tau_0$  is not much smaller than frame length  $T$
- An alternative approach called YIN is to consider the difference function:

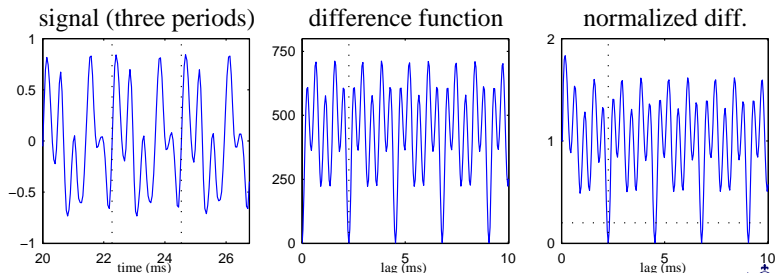
$$d(\tau) = \sum_{t=0}^{T-\tau-1} (x(t) - x(t + \tau))^2$$

which measures the amount of energy in the signal which cannot be explained by a periodic signal of period  $\tau$  (de Cheveigné & Kawahara, JASA 2002)

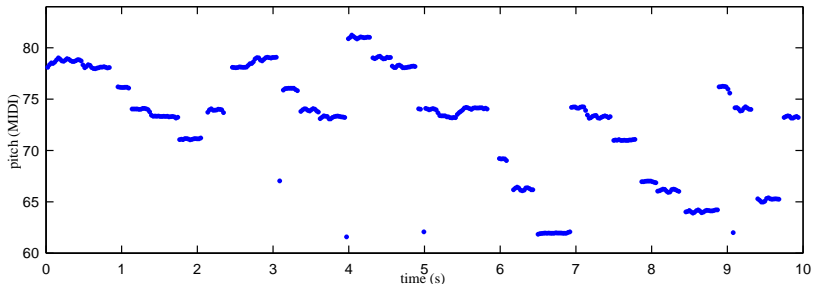
- The normalised difference function is then derived as

$$d'(\tau) = \frac{d(\tau)}{\frac{1}{\tau} \sum_{t=1}^{\tau} d(t)}$$

- The first minimum of  $d'$  below a fixed non-periodicity threshold corresponds to  $\tau_0 = \frac{1}{f_0}$
- $\tau_0$  is estimated precisely by parabolic interpolation
- The value  $d'(\tau_0)$  gives a measure of how periodic the signal is:  $d'(\tau_0) = 0$  if the signal is periodic with period  $\tau_0$



# YIN: Example

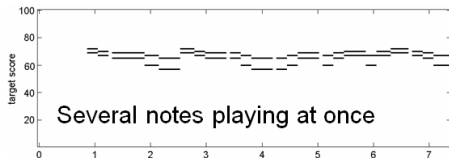


- YIN performs well on monophonic signals and runs in real-time 🎵
- Post-processing is needed to segment the output into discrete note events and remove erroneous pitches (mostly at note transitions)


# Polyphonic Pitch Estimation


# Polyphonic Pitch Estimation: Problem

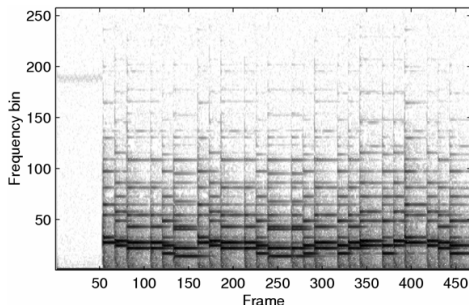
Notes  MIDI  
("Piano Roll")



(Liszt: Etude No. 5 aus Grandes Etudes de Paganini. MIDI from Classical Piano Midi Page <http://www.piano-midi.de>, copyright Bernd Krueger)

 Synth/sample/FT

Task: Extract notes  
from e.g. this 



# Nonnegative Matrix Factorisation (NMF)

- NMF popularized by Lee & Seung (2001)
- NMF models the observed short-term power spectrum  $X_{n,f}$  as a sum of components with a fixed *basis spectrum*  $U_{c,f}$  and a time-varying gain  $A_{c,n}$  plus a residual or error term  $E_{n,f}$  (Smaragdis 2003)

$$X_{n,f} = \sum_{c=1}^C A_{c,n} U_{c,f} + E_{n,f},$$

or in matrix notation  $X = UA + E$

- The only constraints on the basis spectra and gains are (respectively) statistical independence and positivity
- Residual assumed e.g. Gaussian (Euclidean distance)

## NMF

- The independence assumption tends to group parts of the input spectrum showing similar amplitude variations
- The aim is to find the basis spectra and the associated gains according to the *Maximum A Posteriori* (MAP) criterion

$$(\hat{U}, \hat{A}) = \arg \max_{U, A} P(U, A | X)$$

- The solution is found iteratively using the multiplicative update rules

$$A_{c,n} := A_{c,n} \frac{(U^t X)_{c,n}}{(U^t U A)_{c,n}}$$

$$U_{c,f} := U_{c,f} \frac{(X A^t)_{c,f}}{(U A A^t)_{c,f}}$$

- Update rules ensure convergence to a local, not necessarily global, minimum

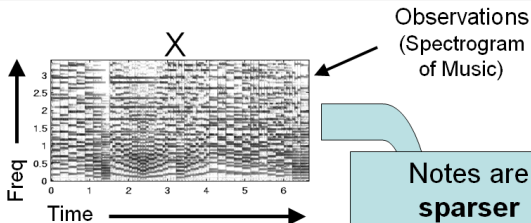
# NMF

- The basis spectra are not constrained to be harmonic, nor to have a particular spectral envelope
- This approach is valid for any instruments, provided the note frequencies are fixed
- However the components are not even constrained to represent notes: some components may represent chords or background noise
- Basis spectra must be processed to infer pitch — one pitch might be represented by a combination of several basis spectra
- Variants of NMF add more prior information, e.g. e.g. sparsity, temporal continuity, or initial harmonic spectra, alternative distortion measures, e.g. Itakura-Saito NMF (Fevotte et al, 2009)



# NMF + Sparsity: Nonnegative Sparse Decomp

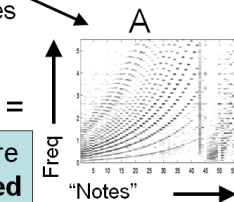
Harpichord music:  
Bach Partita  
in A Minor BWV827



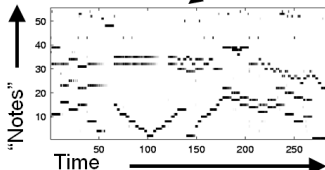
Notes are  
**sparser**  
than spectrogram

Sparse  
Decomposition  $X = A \times S$

Note  
frequencies



Notes



Abdallah & P. (2001). Original:

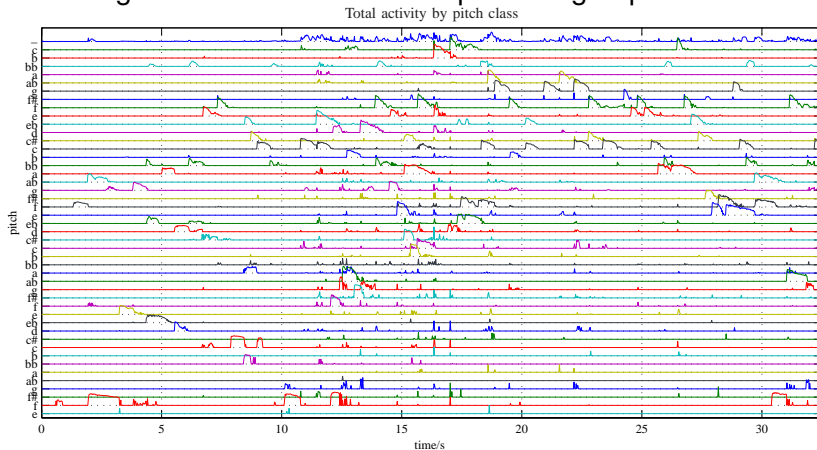


Resynth:



# Groups instead of individual spectra

## Modelling real instruments needs spectrum groups



# Probabilistic Latent Component Analysis (PLCA)

- PLCA: probabilistic variant of NMF (Smaragdis et al, 2006)
- Using constant-Q (log-frequency) spectra, it is possible to share templates across multiple pitches by a simple shift in frequency
- Pitch templates can be pre-learnt from recordings of single notes
- e.g. (Benetos & Dixon, SMC 2011)

$$P(\omega, t) = P(t) \sum_{p,s} P(\omega|s, p) *_\omega P(f|p, t) P(s|p, t) P(p|t)$$

$P(\omega, t)$  is the input log-frequency spectrogram,

$P(t)$  the signal energy,

$P(\omega|s, p)$  spectral templates for instrument  $s$  and pitch  $p$ ,

$P(f|p, t)$  the pitch impulse distribution,

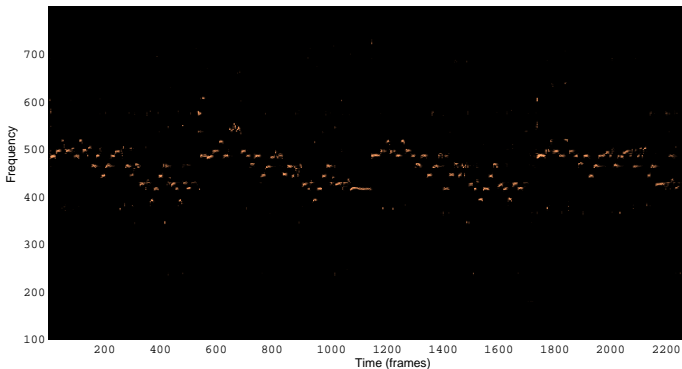
$P(s|p, t)$  the instrument contribution for each pitch, and

$P(p|t)$  the piano-roll transcription.

# Example: PLCA-based Transcription

Transcription of a Cretan lyra excerpt

Original:  Transcription: 



# Chord Transcription

# A Probabilistic Model for Chord Transcription

- Motivation: intelligent chord transcription
  - Modern popular music
- Front end (low-level) processing
  - Approximate transcription (Mauch & Dixon ISMIR 2010)
- Dynamic Bayesian network (IEEE TSALP 2010)
  - Integrates musical context (key, metrical position) into estimation
- Utilising musical structure (ISMIR 2009)
  - Clues from repetition
- Full details in Matthias Mauch's PhD thesis (2010):  
*Automatic Chord Transcription from Audio Using Computational Models of Musical Context*

# The Problem: Chord Transcription

- Different to polyphonic note transcription
- Abstractions
  - Notes are integrated across time
  - Non-harmony notes are disregarded
  - Pitch height is disregarded (except for bass notes)
- Aim: output suitable for musicians

0:10  
G

D/F#

Em

Bm7

G

An-oth-er red let-ter day, so the pound has dropped and the child-ren are cre-at-ing...

## 15 Friends Will Be Friends

G B<sup>7</sup> Em G<sup>7</sup> C F C G D/F# Em Bm<sup>7</sup> G

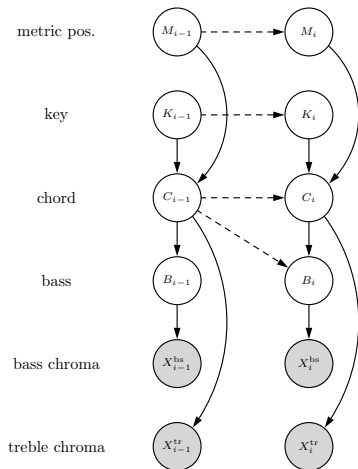
# Signal Processing Front End

- Preprocessing steps
  - Map spectrum to log frequency scale
  - Find reference tuning pitch
  - Perform noise reduction and normalisation
  - Beat tracking for beat-synchronous features
- Usual approach: chromagram
  - Frequency bins of STFT mapped onto musical pitch classes (A, B $\flat$ , B, C, C $\sharp$ , etc)
  - One 12-dimensional feature per time frame
  - Advantage: data reduction
  - Disadvantage: frequency  $\neq$  pitch
- Approximate transcription using non-negative least squares
  - Consider spectrum  $X$  as a weighted sum of note profiles
  - Dictionary  $T$ : fixed spectral shape for all notes
  - $X \approx Tz$
  - Solve for note activation pattern  $z$  subject to constraints
  - NNLS: minimise  $\|X - Tz\|$  for  $z \geq 0$



# Musical Context in a Dynamic Bayesian Network

- Key, chord, metrical position and bass note are estimated simultaneously
  - Chords are estimated in context
  - Useful details for *lead sheets*
- Graphical model with two temporal slices: initial and recursive slice
  - Nodes represent random variables
  - Directed edges represent dependencies
  - Observed nodes are shaded



# Evaluation Results

## MIREX-style evaluation results

Model	RCO
Plain	65.5
Add metric position	65.9
Best MIREX'09 (pretrained)	71.0
Add bass note	72.0
Add key	73.0
Best MIREX'09 (test-train)	74.2
Add structure	75.2
Use NNLS front end	80.7

## Conclusions

- Modelling musical context and structure **is** beneficial
- Further work: separation of high-level (note-given-chord) and low-level (features-given-notes) models

# Onset Detection and Beat Tracking

# Time Domain Onset Detection

- The occurrence of an onset is usually accompanied by an amplitude increase
- Thus using a simple envelope follower (rectifying + smoothing) is an obvious choice:

$$E_0(n) = \frac{1}{N+1} \sum_{m=-N/2}^{N/2} |x(n+m)| w(m)$$

where  $w(m)$  is a smoothing window and  $x(n)$  is the signal

- Alternatively we can square the signal rather than rectify it to obtain the local energy:

$$E(n) = \frac{1}{N+1} \sum_{m=-N/2}^{N/2} (x(n+m))^2 w(m)$$

# Time Domain Onset Detection

- A further refinement is to use the time derivative of energy, so that sudden rises in energy appear as narrow peaks in the derivative
- Research in psychoacoustics indicates that loudness is perceived logarithmically, and that the smallest detectable change in loudness is approximately proportional to the overall loudness of the signal, thus:

$$\frac{\partial E / \partial t}{E} = \frac{\partial(\log E)}{\partial t}$$

- Calculating the first time difference of  $\log(E(n))$  simulates the ear's perception of changes in loudness, and thus is a psychoacoustically-motivated approach to onset detection

# Frequency Domain Onset Detection

- If  $X(n, k)$  is the STFT of the signal  $x(t)$  for  $t = nR_a$ , then the local energy in the frequency domain is defined as:

$$E(n) = \frac{1}{N} \sum_{k=-N/2}^{N/2} |X(n, k)|^2$$

- In the spectral domain, energy increases related to transients tend to appear as wide-band noise, which is more noticeable at high frequencies
- The high frequency content (HFC) of a signal is computed by applying a linear weighting to the local energy:

$$\text{HFC}(n) = \frac{1}{N} \sum_{k=-N/2}^{N/2} |k| \cdot |X(n, k)|^2$$

# Frequency Domain Onset Detection

- Changes in the spectrum are better indicators of onsets than instantaneous measures such as HFC
- For example, the spectral flux (SF) onset detection function is given by:

$$\text{SF}(n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} H(|X(n, k)| - |X(n-1, k)|)$$

where  $H(x)$  is the half-wave rectifier:

$$H(x) = \frac{x + |x|}{2}$$

so that only the increases in energy are taken into account

- An alternative version squares the summands

# Phase-Based Onset Detection

- An alternative is to use phase information
- If  $X(n, k) = |X(n, k)| e^{j\phi(n, k)}$ , then the phase deviation onset detection function PD is given by the mean absolute phase deviation:

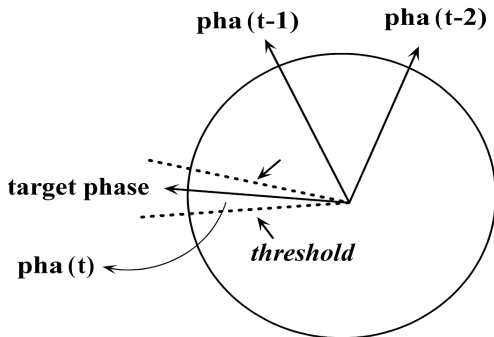
$$PD(n) = \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} |\text{princarg}(\phi''(n, k))|$$

$$PD(n) = \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} |\text{princarg}(\phi(n, k) - 2\phi(n-1, k) + \phi(n-2, k))|$$

- The PD function is sensitive to noise: frequency bins containing low energy are weighted equally with bins containing high energy, but bins containing low-level noise have random phase



# Phase-Based Onset Detection



# Complex Domain Onset Detection

- Another alternative approach is to consider the STFT bin values as vectors in the complex domain
- In the steady-state, the magnitude of bin  $k$  at time  $n$  is equal to its magnitude at time  $(n - 1)$
- Also, the phase is the sum of the phase at  $(n - 1)$  and the rate of phase change  $\phi'$  at  $(n - 1)$
- Thus the target value is:

$$X_T(n, k) = |X(n - 1, k)| e^{j(\phi(n-1,k) + \phi'(n-1,k))}$$

# Complex Domain Onset Detection

- Sum of absolute deviations of observed values from the target values:

$$CD(n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} |X(n, k) - X_T(n, k)|$$

- To distinguish between onsets and offsets, the sum can be restricted to bins with increasing magnitude:

$$RCD(n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \begin{cases} |X(n, k) - X_T(n, k)|, & \text{if } |X(n, k)| \geq |X(n-1, k)| \\ 0, & \text{otherwise} \end{cases}$$

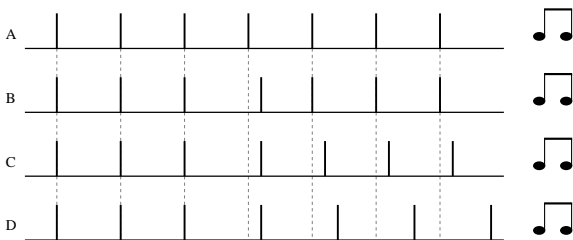
- Onset Detection Tutorial:  
Bello et al (IEEE Trans SAP, 2005)

# Tempo

- Tempo is the rate of a pulse (e.g. the nominal beat level)
- Usually expressed in beats per minute (BPM)
- Problems with measuring tempo:
  - Variations in tempo: people do not play at a constant rate, so tempo must be expressed as an average over some time window
  - Not all deviations from metrical timing are tempo changes
  - Choice of metrical level: people tap to music at different rates; the “beat level” is ambiguous (problem for development and evaluation)
  - Strictly speaking, tempo is a perceptual value, so it should be determined empirically

# Timing

- Not all deviations from metrical timing are tempo changes





- Nominally on-the-beat notes don't occur *on* the beat
  - difference between notation and perception
  - “groove”, “on top of the beat”, “behind the beat”, etc.
  - systematic deviations (e.g. swing)
  - expressive timing
  - see (Dixon et al., *Music Perception*, 2006)

# Tempo Induction and Beat Tracking

- *Tempo induction* is finding the tempo of a musical excerpt at some (usually unspecified) metrical level
  - Assumes tempo is constant over the excerpt
- *Beat tracking* is finding the times of each beat at some metrical level
  - Usually does not assume constant tempo
- Many approaches have been proposed
  - e.g. Goto 97, Scheirer 98, Dixon 01, Klapuri 03, Davies & P. 05
  - reviewed by Gouyon and Dixon (CMJ 2005)
  - see also MIREX evaluations (Gouyon et al., IEEE TSAP 2006; McKinney et al., JNMR 2007)

# Tempo Induction

- The basic idea is to find periodicities in the audio data
- Usually this is reduced to finding periodicities in some feature(s) derived from the audio data
- Features can be calculated on events:
  - E.g. onset time, duration, amplitude, pitch, chords, percussive instrument class
  - To use all of these features would require reliable onset detection, offset detection, polyphonic transcription, instrument recognition, etc
  - Not all information is necessary:  
 Original   $\Rightarrow$  Onsets 
- Features can be calculated on frames (5–20ms):
  - Lower abstraction level models perception better
  - E.g. energy, energy in various frequency bands, energy variations, onset detection features, spectral features

# Periodicity Functions

- A *periodicity function* is a continuous function representing the strength of each periodicity (or tempo)
- Calculated from feature list(s)
- Many methods exist, such as autocorrelation, comb filterbanks, IOI histograms, Fourier transform, periodicity transform, tempogram, beat histogram, fluctuation patterns
- Assumes tempo is constant
- Diverse pre- and post-processing:
  - scaling with tempo preference distribution
  - using aspects of metrical hierarchy (e.g. favouring rationally-related periodicities)
  - emphasising most recent samples (e.g. sliding window) for on-line analysis



# Example 1: Autocorrelation

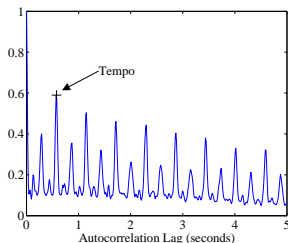
- Most commonly used
- Measures feature list  $x(n)$  self-similarity vs time lag  $\tau$ :

$$r(\tau) = \sum_{n=0}^{N-\tau-1} x(n)x(n+\tau) \quad \forall \tau \in \{0 \dots U\}$$

where  $N$  is the number of samples,  $U$  the upper limit of lag, and  $N - \tau$  is the integration time

# Autocorrelation

- ACF using normalised variation in low frequency energy as the feature:



- Variants of the ACF:
  - Narrowed ACF (Brown 1989)
  - “Phase-Preserving” Narrowed ACF (Vercoe 1997)
  - Sum or correlation over similarity matrix (Foote 2001)
  - Autocorrelation Phase Matrix (Eck 2006)

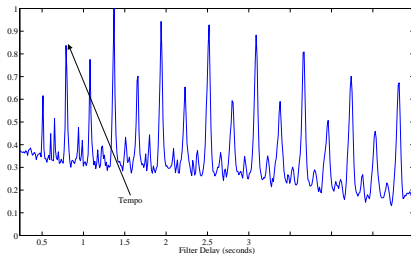
# Example 2: Comb Filterbank

- Bank of resonators, each tuned to one tempo
- Output of a comb filter with delay  $\tau$ :



$$y_{\tau}(t) = \alpha_{\tau} y_{\tau}(t - \tau) + (1 - \alpha_{\tau}) x(t)$$

where  $\alpha_{\tau}$  is the gain,  $\alpha_{\tau} = 0.5^{\tau/t_0}$ , and  $t_0$  is the half-time

- Strength of periodicity is given by the instantaneous energy in each comb filter, normalised and integrated over time



# Beat Tracking

- Complementary process to tempo induction 
- Fit a grid to the events (respectively features)
  - basic assumption: co-occurrence of events and beats
  - e.g. by correlation with a pulse train
- Constant tempo and metrical timing are not assumed 
  - the “grid” must be flexible
  - short term deviations from periodicity
  - moderate changes in tempo
- Reconciliation of predictions and observations
- Balance:
  - reactivity (responsiveness to change)
  - inertia (stability, importance attached to past context)

# Beat Tracking Approaches

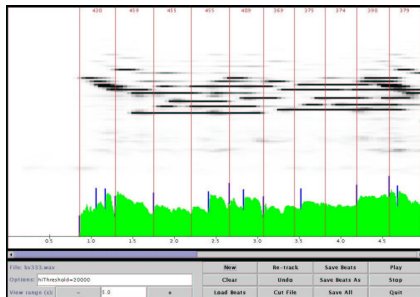
- Top down and bottom up approaches
- On-line and off-line approaches
- High-level (style-specific) knowledge vs generality
- Rule-based methods
- Oscillators
- Multiple hypotheses / agents
- Filter-bank
- Repeated induction
- Dynamical systems
- Bayesian statistics
- Particle filtering

# Example: Comb Filterbank

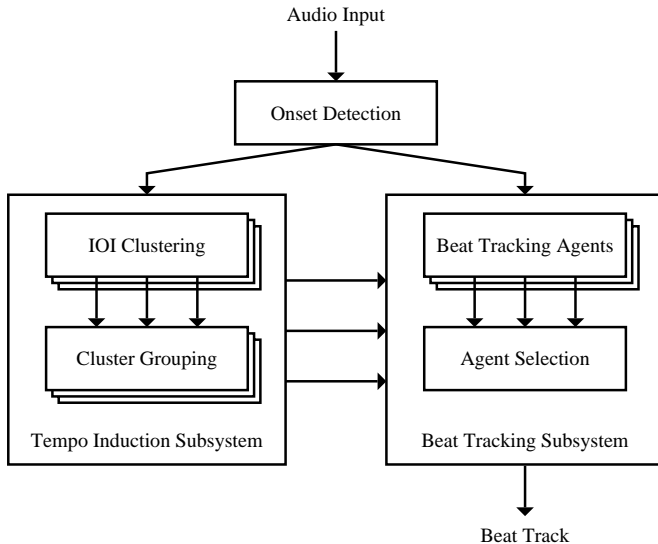
- Schierer 1998
- Causal analysis
- Audio is split into 6 octave-wide frequency bands, low-pass filtered, differentiated and half-wave rectified
- Each band is passed through a comb filterbank (150 filters from 60–180 BPM)
- Filter outputs are summed across bands
- Filter with maximum output corresponds to tempo
- Filter states are examined to determine phase (beat times)
- Tempo evolution determined by change of maximal filter
- Problem with continuity when tempo changes

# Example: BeatRoot

- Dixon, JNMR 2001, 2007
- Analysis of expression in musical performance
- Automate processing of large-scale data sets
- Tempo and beat times are estimated automatically
- Annotation of audio data with beat times at various metrical levels
- Interactive correction of errors with graphical user interface



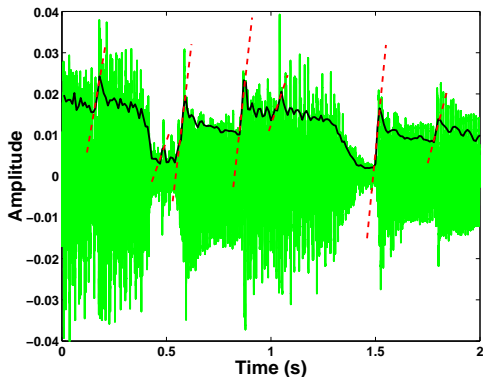
# BeatRoot Architecture





# Onset Detection

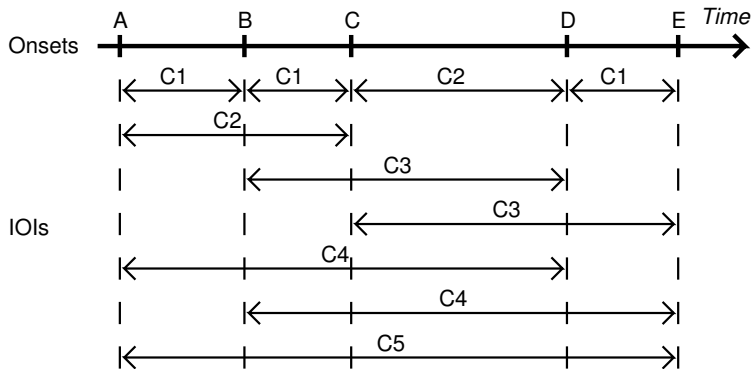
- Fast time domain onset detection (2001)
  - Surfboard method (Schloss '85)
  - Peaks in slope of amplitude envelope



- Onset detection with spectral flux (2006)

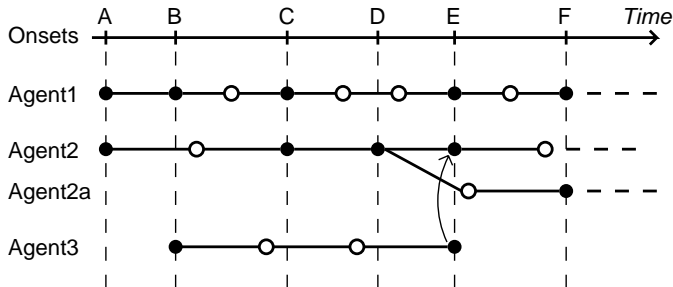
# Tempo Induction

- Clustering of inter-onset intervals
- Reinforcement and competition between clusters


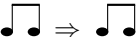



# Beat Tracking: Agent Architecture

- Estimate beat times (phase) based on tempo (rate) hypotheses
- State: current beat rate and time
- History: previous beat times
- Evaluation: regularity, continuity & salience of on-beat events

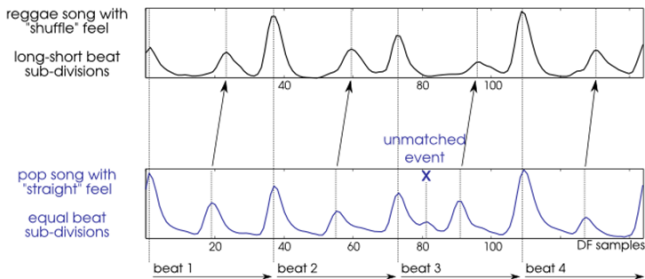


# Results

- Tested on pop, soul, country, jazz, ... 
- Only using onsets:   $\Rightarrow$  
- Results: ranged from 77% to 100%
- Tested on classical piano (Mozart sonatas, MIDI data)
  - Agents guided by event salience calculated from duration, dynamics and pitch
  - Results: 75% without salience; 91% with salience

# Rhythm Transformation

- Extend Beat Tracking to Bar level: Rhythm Tracking
- Rhythm Tracking on model (top) and original (bottom)
- Time-scale segments of original to rhythm of model

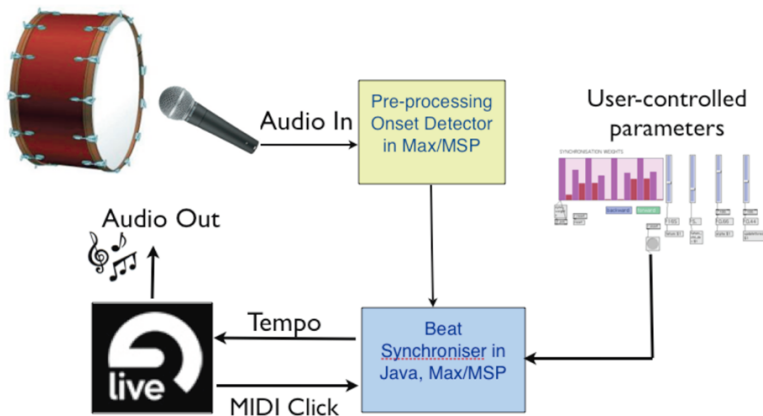


Original:  Model:  Result: 

# Live Beat Tracking

# Live Beat Tracking System: B-Keeper

Robertson & P. (2008, 2012)



[Video: <http://www.youtube.com/watch?v=iyU61cG-j0Y>]

# Conclusions



# Conclusions

- Introduction and Music fundamentals
- Pitch estimation and Music Transcription
  - Pitch Tracking: Autocorrelation
  - Nonnegative Matrix Factorization (NMF)
  - Chord Analysis
- Temporal analysis
  - Onset Detection
  - Beat Tracking
  - Rhythm Analysis
- Many other tasks & methods not covered here:
  - Music audio coding, Phase vocoder, Sound synthesis, ...

## Further Reading ...

- *Sound to Sense – Sense to Sound: A state of the art in Sound and Music Computing*, ed. P Polotti, D Rocchesso (Logos, 2008)  
Available at <http://smcnetwork.org/node/884> (PDF)
- *DAFX - Digital Audio Effects*, ed. U Zölzer (Wiley, 2002)
- *The Computer Music Tutorial*, C Roads (MIT Press, 1996)
- *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing and Programming*, ed. R Boulanger
- *Signal Processing Methods for Music Transcription*, ed. A Klapuri and M Davy (Springer 2006)
- *Musical Signal Processing*, ed. C Roads, S Pope, A Piccialli and G de Poli (Swets and Zeitlinger 1997)
- *Elements of Computer Music*, F R Moore (Prentice Hall 1990)
- *The Science of Musical Sounds*, J Sundberg (Academic Press 1991)