ECS7006 Music Informatics

Week 4 – Beat Tracking

School of Electronic Engineering and Computer Science
Queen Mary University of London

prepared by Simon Dixon
using material by Juan Bello, Emmanuel Vincent and Meinard Müller

s.e.dixon@qmul.ac.uk

2023

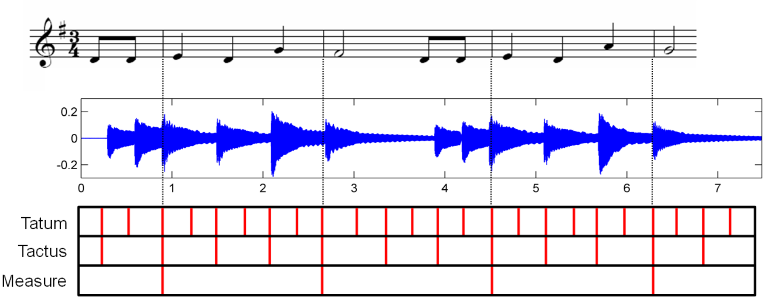
Introduction: Rhythm and Metre

Rhythm: Overview

- *Rhythm* refers to the temporal characteristics of music
  - at the event level or higher
  - i.e. not including the temporal evolution of individual notes
- Rhythm has the following components:
  - **Pulse**: an equally spaced *sequence of perceived accents* in time
  - **Metre**: the *structure* that best fits the event occurrences
  - **Tempo**: the *rate* of the primary pulse (metrical level)
  - **Timing**: *when* events occur (relative to the metre); includes discontinuities in the pulse
  - **Grouping**: phrase structure and form (sometimes excluded from rhythm)

Pulse and Beat

- **Pulse**: a regularly spaced sequence of accents (*beats*)
  - Not unique: multiple pulses exist simultaneously
  - The primary pulse is called the *beat* or *tactus* (the rate at which one claps or taps along with music)
  - Bilmes (1993) called the fastest pulse the *tatum*



Metrical Structure

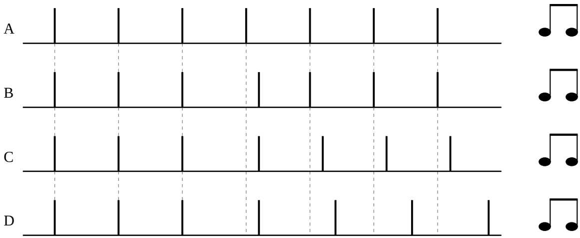
- **Metrical structure**: (usually) hierarchical set of pulses
  - Each pulse defines a *metrical level*
  - Higher metrical levels correspond to longer time divisions
- Time signature: indicates relationships between metrical levels
  - e.g. $\frac{3}{4}$, $\frac{4}{4}$, $\frac{6}{8}$, $\frac{7}{8}$
  - the number (top) and type (bottom) of beats per bar (US: *measure*)
  - compound time signatures (e.g. $\frac{6}{8}$) also define an intermediate level (subdivision of beats)
- All durations are simple rational multiples of a reference unit
  - This “grid” structure doesn’t model performed music 
- Metrical structure is explicit in the score (time signature, bar lines, note durations), but only implicit in audio 

Tempo

- **Tempo** is the *rate* of the primary 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 (this is a problem for development and evaluation)
  - Strictly speaking, tempo is a perceptual value, so it should be determined empirically (e.g. listening tests)

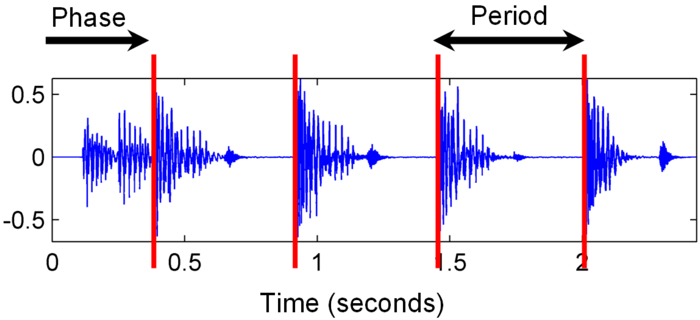
Timing

- Not all deviations from metrical timing are tempo changes



- Nominally on-the-beat notes don't occur *on* the beat
  - differences between notation, performance 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**: finding the tempo of a musical excerpt at some (usually unspecified) metrical level
  - Assumes tempo is constant over the excerpt
- **Beat tracking**: finding the times of each beat at some metrical level
  - Usually does not assume constant tempo



Tempo Induction

Tempo Induction

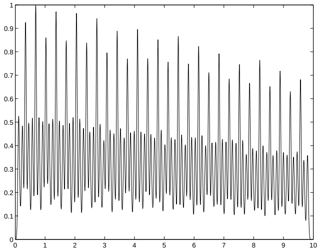
- The basic idea is to find *periodicities* in audio features, assuming:
  - strong metrical positions often coincide with note onsets
  - for a given metrical level, these accents are regularly spaced
- Features can be calculated on *events*:
  - e.g. onset time, duration, amplitude, pitch, chords, 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: Onsets  (Original )
- Features can be calculated on *frames* (5–20ms):
  - e.g. energy, energy in various frequency bands, energy variations, onset detection features, spectral features
- An onset detection function is an obvious candidate feature

Periodicity Functions

- A *periodicity function* is a continuous function representing the strength of each periodicity (or tempo)
- Calculated from features extracted from the audio
- Constant tempo is assumed
- Many methods: e.g. autocorrelation, comb filterbanks, IOI histograms, Fourier transform, periodicity transform, tempogram, beat histogram, fluctuation patterns
- Diverse pre- and post-processing techniques are used to emphasise and select the peak that corresponds to the desired metrical level

Periodicity Function: Time Interval Histogram

- Compute onset times and inter-onset intervals (IOIs)
- Create a histogram by assigning IOIs to discrete periodicity bins
- Smooth with e.g. Gaussian window
- IOI clustering scheme (Dixon 2001) allocates bins dynamically, with bin width proportional to the period



Time (seconds)

Periodicity Function: Autocorrelation

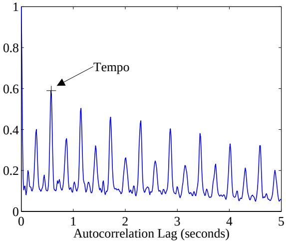
- Measures self-similarity of feature $x(n)$ vs time lag τ:

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

  where N is the number of samples, U the upper limit of lag, and $N - \tau$ is the integration time
- Simple and efficient to implement via FFT

Autocorrelation

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



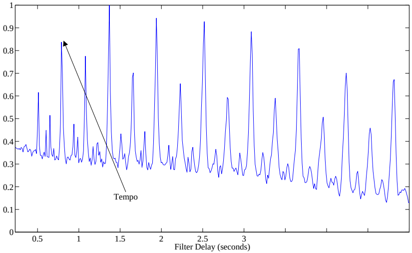
- Variants of the ACF:
  - Narrowed ACF (Brown 1989)
  - "Phase-Preserving" Narrowed ACF (Vercoe 1997)
  - Sum or correlation over similarity matrix (Foote 2001)
  - Autocorrelation Phase Matrix (Eck 2006)

Periodicity Function: Comb Filterbank

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

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

  where α_τ 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

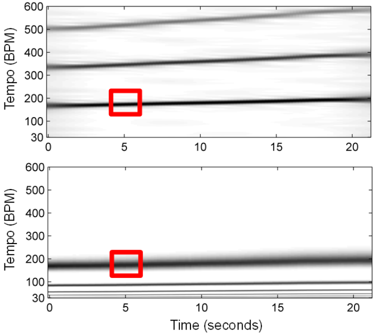


Interpreting the Periodicity Function

- A scalar value for the main tempo has to be extracted from the peaks of the periodicity function
- The simplest approach is to take the position of the maximum
- Other approaches weight the PF with a prior (tempo preference), or restrict it to a specific range, e.g. 61-120 BPM
- Consider constraints imposed by the metrical hierarchy
  - perform beat tracking and return the tempo of the most regular beat sequence (Dixon 2001)
  - consider only periodic peaks (Gouyon and Herrera, 2003)
  - score all beat and bar level hypotheses, favouring rationally-related periodicities (Dixon et al., 2003)
  - probabilistic framework (Klapuri et al., 2006)

Tempogram: Tracking Periodicity over Time

- A **tempogram** is a time-frequency representation similar to a spectrogram, where the frequencies involved correspond to rhythmic periodicities (30–300 BPM $\equiv$ 0.5–5 Hz)
- Can be calculated from a periodicity function using a sliding window
- Fourier-based tempograms tend to have peaks at the tempo and integer multiples of the tempo
- Autocorrelation-based tempograms tend to have peaks at the tempo and integer quotients of the tempo
- Can combine both methods (Peeters 2007)



Beat Tracking

Beat Tracking

- Complementary process to tempo induction
- Fit a “grid” to the events (or features) 
  - basic assumption: co-occurence 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
- Reconcile predictions with observations, balancing:
  - reactiveness (responsiveness to change)
  - inertia (stability, importance attached to past context)

Beat Tracking Approaches

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

Beat Tracking as State Space Search

- Set of state variables
- Initial state
- Observations (data)
- Goal state (an explanation of the observations)
- Set of actions (causing state transitions)
- Utility function (on states, to evaluate actions)

State Model: Variables

- Pulse period (tempo)
- Pulse phase (beat times)
  - expressed as time of first beat (constant tempo) or current beat (variable tempo)
- Current metrical position (models of complete metrical structure)
- Confidence measure (multiple hypothesis models)
- History (memory)

State Model: Observations (Features)

- All events or those in a local time window
- Onset times, durations, inter-onset intervals (IOIs)
  - equivalent only for monophonic data without rests
  - longer notes are more indicative of beats than shorter notes
- Dynamics
  - louder notes are more indicative of beats than quieter notes
  - note-based dynamics are difficult to measure
- Pitch and other features
  - lower notes are more indicative of beats than higher notes
  - particular instruments are good indicators of beats (e.g. snare drum)
  - harmonic change can indicate a high level metrical boundary

State Model: Actions and Evaluation

A simple beat tracker with state (b_n, T_n), where b_n is most recent beat time, T_n is inter-beat interval (period), input is set of onsets $\{O_1, O_2, ...\}$:

- Predict the next beat location based on current beat and beat period:

$$\hat{b}_{n+1} = b_n + T_n$$

- Choose closest event and update state variables accordingly:

$$\begin{aligned} b_{n+1} &= O_k, \quad \text{where } k = \arg\min_m |\hat{b}_{n+1} - O_m| \\ T_{n+1} &= \alpha(b_{n+1} - b_n) + (1-\alpha)T_n \end{aligned}$$

  α is a constant controlling balance of reactiveness vs inertia

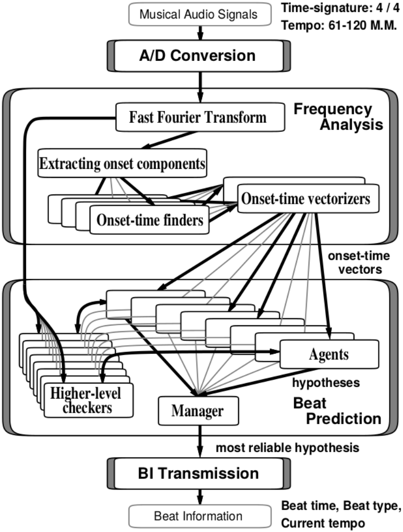
- Evaluate actions on the basis of agreement with prediction
  - Sometimes onset O_k is not close enough and should be ignored

Example 1: Metrical Parsing

- Dannenberg and Mont-Reynaud, ICMC 1987
- On-line algorithm processing MIDI data
- All incoming events are assigned to a metrical position
- Deviations serve to update period
- Position in metrical structure determines weight of update
- Reactiveness/inertia adjusted with *decay* parameter
- Extended to track multiple hypotheses (Allen et al. 1990)
  - delay commitment to a particular metrical interpretation
  - less reactive, but more robust
  - evaluate hypotheses based on musical knowledge
  - prune hypotheses with dynamic programming

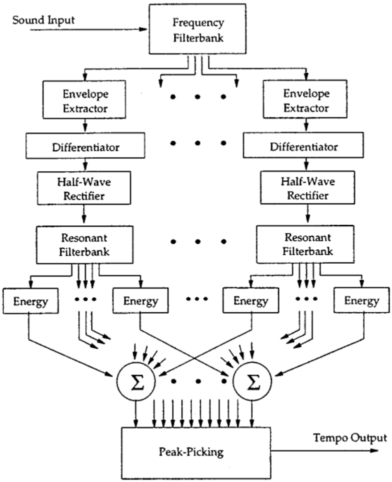
Example 2: Multiple Agents

- Goto and Muraoka, ICMC 1995
- Real-time audio beat tracking
- Detects onsets, specifically labelling bass and snare drums
- Finds beats ($\frac{1}{4}$ and $\frac{1}{2}$ note level)
- Matches drum patterns with templates to avoid doubling errors and phase errors
- Beat times predicted with auto-correlation (tempo) and cross-correlation (phase)
- The 14 pairs of agents evaluate their own reliability based on fulfilment of predictions



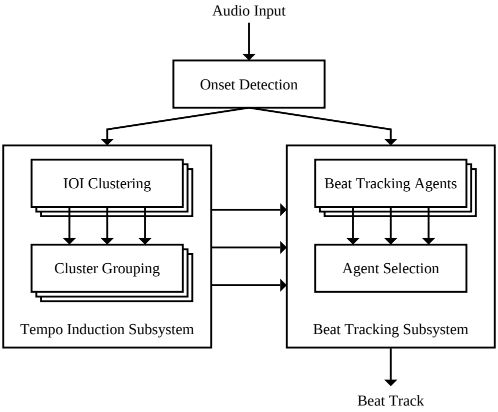
Example 3: Comb Filterbank

- Scheirer, JASA 1998
- 6 octave-wide filters, LPF, differentiation, rectification
- Comb filterbank (150 filters, 60–180 BPM)
- Sum outputs across bands
- Maximum output corresponds to tempo
- Filter states determine phase (beat times)
- Problem with continuity when tempo changes



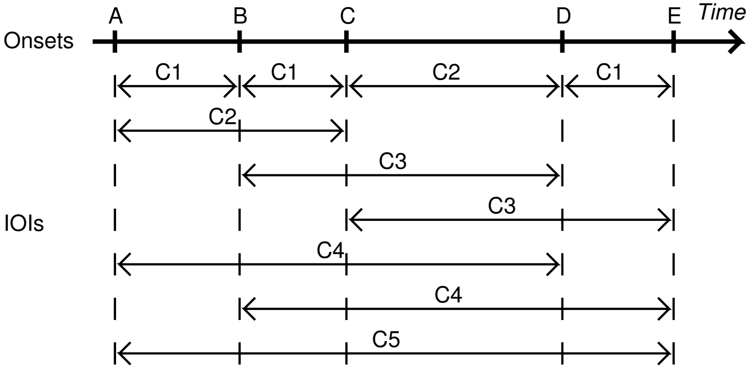
Example 4: BeatRoot

- Dixon, JNMR 2001, 2007
- Large-scale analysis of expression in musical performance
- Annotation of audio data with beat times at various metrical levels
- Interactive correction of errors with graphical user interface



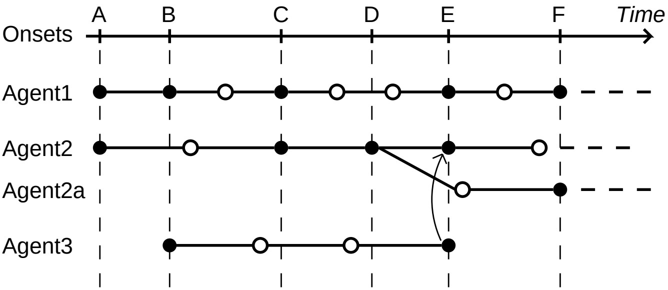
BeatRoot: Tempo Induction

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



BeatRoot: Beat Tracking Agents

- 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



BeatRoot: Results

- Tested on pop, soul, country, jazz, ... 
  - Only using onsets:  ⇒ 
  - 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

Beat Tracking: State of the Art

- Krebs et al. (2016) use recurrent neural networks to model rhythmic and harmonic content of the signal, with the output combined and fed into a dynamic Bayesian network which acts as a rhythmic language model
- An alternative model using conditional random fields (Böck and Korzeniowski) shows superior performance on a dataset of Chopin mazurkas
- Reference implementations are available as part of the Madmom library

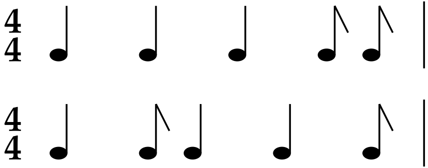
Beat Tracking: Resources

- Example approaches: (Goto 1997; Scheirer 1998; Dixon 2001; Davies 2005; Klapuri 2006; Ellis 2007; Grosche and Müller 2011; Böck & Schedl 2012; Krebs et al. 2015; Durand et al. 2015)
- Literature reviews: Gouyon and Dixon (CMJ 2005); Grosche (PhD, 2012); Holzapfel et al. (IEEE TSALP 2012)
- Evaluations: Gouyon et al. (IEEE TSAP 2006); McKinney et al. (JNMR 2007); MIREX: `https://www.music-ir.org/mirex/wiki/2019:Main_Page`

Rhythm Patterns

Characterisation & Classification by Rhythm Patterns

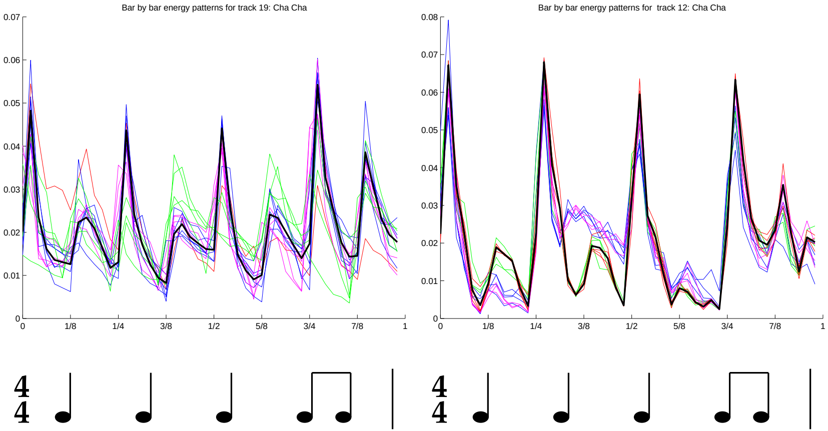
- Many analysis methods compute a distribution of time intervals ignoring their order (e.g. beat histogram, modulation energy, periodicity distribution)
- Temporal order defines patterns (musically important!)
- Classification of ballroom dance music by rhythm patterns (Dixon et al., ISMIR 2004)
- Patterns: one-dimensional vector of energy in bar-length segments
- Temporal order (within each bar) is preserved
- Musically meaningful (high level) interpretation of patterns



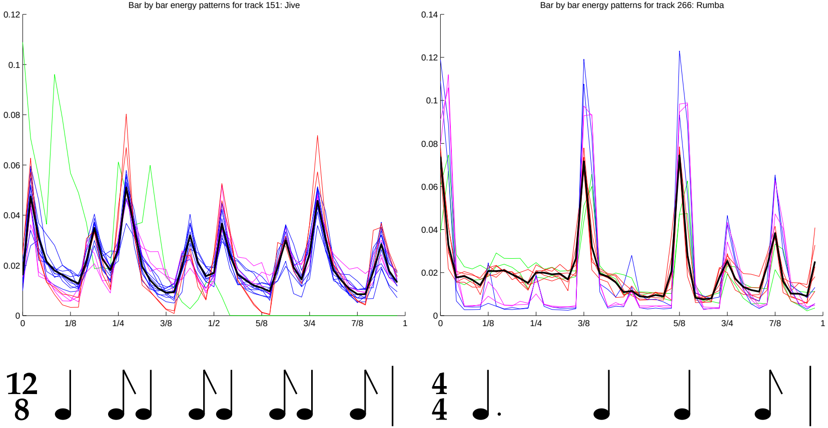
Pattern Extraction

- The boundaries of the first bar were found using BeatRoot, and any errors were corrected manually
- The subsequent bar boundaries were found by correlation, based on the assumption that bars contain similar rhythmic patterns
- The amplitude envelope was computed by LP-filtering and downsampling to produce a vector of 72 samples per bar
- These vectors were clustered using the k-means algorithm (k=4) and the Euclidean distance for comparing vectors
- The largest cluster was selected as containing the most significant rhythm pattern, which is represented by the average of its constituent vectors

Rhythm Pattern Examples: Cha Cha



More Rhythm Pattern Examples: Jive and Rumba



Classification

- Standard machine learning software: Weka
  - k-NN, J48, AdaBoost, Classification via Regression
- Feature vectors:
  - Rhythm pattern
  - Derived features
  - Periodicity histogram
  - IOI histogram / “MFCC”-like features
  - Tempo

Classification Results

Feature sets	Without RP	With RP (72)
None (0)	15.9%	50.1%
Periodicity histograms (11)	59.9%	68.1%
IOI histograms (64)	80.8%	83.4%
Periodicity & IOI hist. (75)	82.2%	85.7%
Tempo attributes (3)	84.4%	87.1%
All (plus bar length) (79)	95.1%	96.0%

Rhythm Patterns: Summary

- Only rhythm
  - No timbre (instrumentation), harmony, melody, lyrics
- One pattern
  - Sometimes trivial
- Short pieces (30 sec), low quality sound
- Up to 96% classification — surprising result