Music Informatics

Alan Smaill

Feb 4th 2014

Today

- Beat tracking
- real time vs offline
- input: midi vs audio
- real-time audio metrical analysis

Beat tracking

We have seen a a simple set of rules can make a good job of analysing the metrical structure of WTM, in a special particular case, starting from idealised durations taken from a score (rather than eg from midi file of real performance).

The problem of beat tracking is to work out in general in an automatic way where the beat falls in a given piece.

It probably doesn't make sense to try to develop something that will work for arbitrary sorts of music from arbitrary traditions, since very different temporal organisations can be found in different styles and musical cultures.

Some versions of the problem

The basic task involves working out where the underlying pulse is, ie where the listener will tap along – the tactus. (This makes it an empirical question without a definitive ground truth.)
There are variations depending on how what from the input takes, and whether processing is done in real time.

There is a related but distinct problem called score following where along with the musical input, the score that is being performed is also given to the analysis system – such a system needs to be robust when the performer does not follow the score exactly.
We will look at this problem later.

Variants

- Input: midi-like **or** sound-wave.
  midi will give precise timings for note onsets, so this is the easier version
- Processing: real time, or off-line; real-time is more challenging, but that is the more interesting task for most musical systems.

A beat-tracking component is needed for systems that eg

- score extraction from midi data
- enable musically aware actions in machines (dancing robot)
- generate accompaniment to a soloist automatically, where prediction of the timing of the next note is important
- for virtual musical performers improvising with human performers; again want some ability to play "in time".

Real time beat tracking

Although people without a musical background pick up an underlying beat without too much trouble, this is a difficult task for a machine at the current time. Music with a pronounced regular beat is of course easier to work with than music with syncopation, or music with tempo fluctuations.

Likewise, if a system is specialised for a particular style, it is also easier to get good results.

There have been some competitions for beat-tracking systems (not necessarily real-time) so there are available sound files, together with annotations where listeners have indicated where they hear the beat, form the MIREX conference:

`http://www.music-ir.org/mirex2006/index.php/Audio_Beat_Tracking`

Evaluation

Care is needed to get a good evaluation of performance in this task.

- Human listeners disagree
  but in systematic ways, usually (eg twice the number of beats)
- It takes time to work out the beat at the start;
  so ignore behaviours for a short time at the start
- Whether following tempo fluctuations is more important than getting local timing of beat accurately;
  if the latter is the aim, it makes sense to work with a window (25 sec in the MIREX case), and look for correlations between machine output and that of all the listeners.

We'll look at Simon Dixon's BeatRoot system;
sources and documentation, including papers on versions of the algorithm, and comparisons with other systems, are available at:

> *http://www.elec.qmul.ac.uk/people/simond/beatroot/*

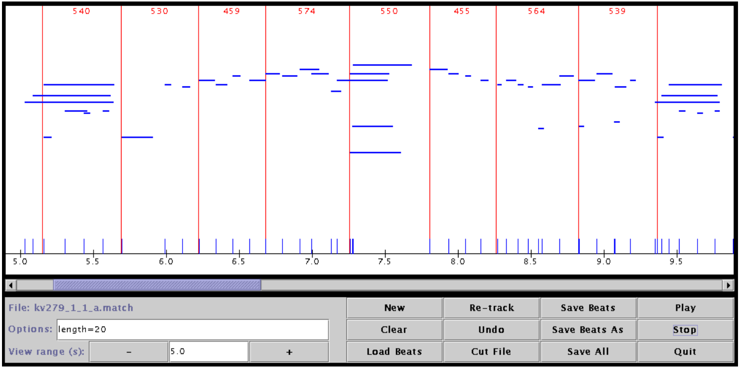
This work has looked at various algorithms, and tuning of the parameters involved. There is an interactive system that allows users to annotate music (as midi or sound-files) with information on where the beat is, and to override mistaken beat indications, via a graphical interface.

Acknowledgements to Simon Dixon's web page for the information in the next slides.

The components of the system:

- onset detection (start of notes)
- tempo induction (using onset information):
  * look at inter-onset intervals, and clusters of these
  * look for relations between clusters (integer multiples of duration)
- beat tracking (using both onset and tempo information):
  * beat tracking agents keep different hypotheses of how notes fit to beat
  * select most successful hypothesis

Onset detection: midi for free



We have immediate access to note onset times.

Onset detection: sound-files

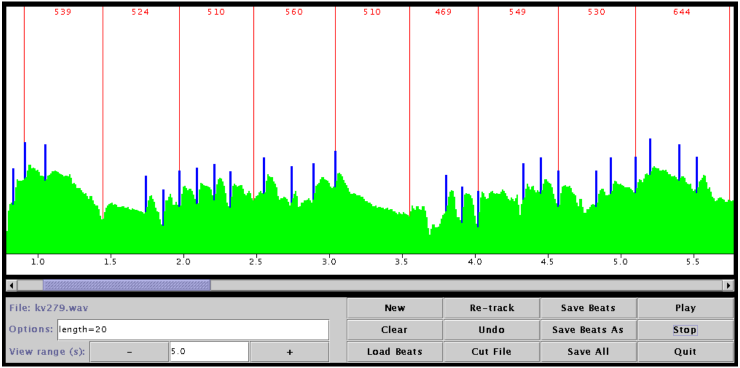
Different algorithms for this are known.
The spectral flux onset algorithm divides the sound signal into several bins for different frequency ranges. Now look for places where the change in magnitude of the signal in the frequency range.
The onset detection function adds the size of the change from all ranges where the change is positive. Where there are different notes, and eg percussion, change may be negative at some frequency bands, and positive in the bands associated with new sounds.
Look for local maxima in this detection function:

- maximum in a small interval (in past and future) around the time
- larger by a threshold than the average value in the interval

Local interval

The local interval around which a maximum is looked for is small (eg 0.06 sec).
The question of the perceptual accuracy of this algorithm as a model of human note onset judgements would need to look closely at how this is done. We believe that the perceptual "now" involved in this sort of judgement allows this sort of processing as a judgement over short time periods.

Onset detection: from audio



where are the “real” note onsets?

Now look at the inter-onset intervals (IOIs):
ie, duration between all inter-onset times (not just consecutive onsets)
We expect a pattern where some of these intervals will be multiples of some an underlying pulse.
Look for clusters of IOI

assign duration to cluster if it differs from the mean of the cluster by less than a fixed small duration.

Finally look for approximate integer relations between clusters. Information about the clusters, their size, and how well they relate to other clusters, is passed to the beat-tracking component.

Beat tracking agents

This supports a number of competing hypotheses about the beat using a version of multiple agents system, one for each hypothesis. A number of agents process the music sequentially, and the agents make predictions about where the next beat will be according to their hypothesis.

Agents are initialised with a tempo hypothesis and a first onset time taken from the early onset times detected in the music.

Where the agent predicts a beat:

- if an onset occurs in a tight window round the prediction, that is taken as a beat by that agent
- if an onset occurs in a looser window round the prediction, allow for two possibilities (on the beat, or not on the beat) by creating new agent.

Beat tracking agents ctd

Agents can be thrown away if their predictions match suitably with those of another agent, or if they keep failing to predict any onsets (they have lost track of the beat).

Agents are rated by an evaluation function that takes into account

- how evenly beat times are spaced
- how many predicted events correspond to actual onsets
- salience of the onsets ("weight" of spectral flux change)

The output is the analysis of the agent with the best overall evaluation.

Comments

This approach works as well as anything at the moment, but struggles with certain music that people find relatively easy.
It is reminiscent of Longuet-Higgins' approach, in that

- processing is incremental
- integer multiples between durations are important
- the aim is to model a human musical capability
- the system outputs a unique answer

However, Beatroot's support of competing hypotheses allow it in principle to model garden path experience in beat analysis by looking dynamically at the current best agent as modelling the current understanding of where the beat is.

A more recent project working on beat-tracking is the B-Keeper project:
http://www.b-keeper.org/
and demo at:
http://www.elec.qmul.ac.uk/digitalmusic/downloads/b-keeper/

Summary

- beat tracking problem
- versions depending on input and real-time
- BeatRoot approach as an example of offline beat-tracking