Tempo extraction status update

Simon de Leon

MUMT611

4/6/06

Agenda

- Introduction
- Review of technique
- Accomplished work
- Future work
- Conclusion

Introduction

- Implementation (and possible improvement) of winning MIREX 2005 tempo extraction algorithm [1]
- Goal of final project divided into three parts
  - Onset algorithm (spectral flux)
  - Periodicity detection
  - Impose metronome tick track
- All goals “semi” complete
  - New problems and resulting sub-goals not fully described in paper
  - Need to evaluate implementation against groundtruth data

Introduction

- Never-ending improvement and difficulty with particular genres – maybe it will never be complete
- Straight-ahead music (most hip hop, R&B, pop, rock) works well
- Difficulties occur when there are
  - Silent parts
  - Rapid tempo changes or free flowing tempo
  - Genres that shuffle the rhythm and accent the off-beats (samba, broken beat)

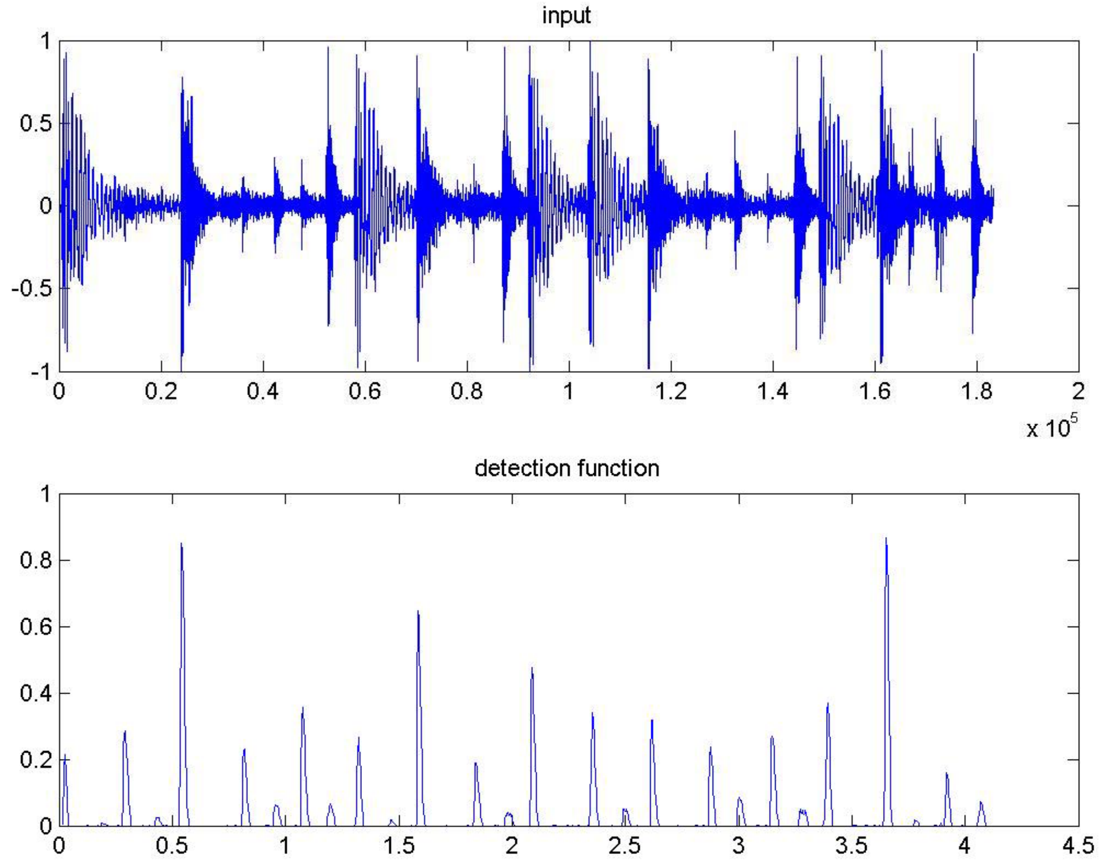
Review of technique

- See [1] for fine details
- Basic idea is to extract spectral flux and determine its periodicity via auto-correlation
- Low-level method that assumes salient features correspond to fastest change in spectrum
  - Inherent limitation: this was addressed in beat spectrum method, and can possibly be addressed with classifiers

Spectral flux

- Step 1: Take STFT of signal
- Step 2: Take time derivative of frequency components (spectral energy flux)
  - a) Low-pass filter STFT magnitude
  - b) Apply logarithmic compression [2]
  - c) Pass through FIR filter differentiator [3]
- Step 3: Use dynamic threshold and remove the smallest onset spectral energy flux “spikes” from previous step

Spectral flux



- Detection function extracted from one measure of The Incredible Bongo Band's "Apache"

Periodicity

- Perform autocorrelation in the lag range corresponding to 200 BPM to 60 BPM
- The lag of the 3 largest peaks are examined for a multiplicity relationship
- If no multiplicity relationship found, take lag of largest peak as the period
- This seems to be a source of problems with broken beat, samba, etc. where the highest detection function peaks do not correspond to beat onsets
  - Result: incorrect tempo extracted, and metronome overlay constantly struggles to re-align phase

"I gotta have more cowbell, baby!"

- Overlay cowbells at extracted tempo
- Continually re-aligning phase as tempo changes is challenging
- Multiplicity of the cowbell period is a bit of a guessing game
- Search local area of potential cowbell for peak cross-correlation with corresponding section of detection function

Audio examples

- Straight-ahead rhythmic music fairly easy since tempo is steady and strongly accented at perceived beats
- There are some features that have not yet been implemented that may help, described next section

Future work

- Initial phase-alignment is difficult
  - Often we can get the correct tempo but the cowbells fall on the off-beats
- For longer clips with changing tempos, need to constantly perform correlation on subsections of the detection function
- Need to test against groundtruth data
- How do we intelligently deal with problematic areas
  - Drum fills, silent breaks, difficult genres (broken beat, samba, classical, etc.)
- Would be fun to do an auto-DJ MATLAB script for an easy genre

Conclusion

- From preliminary implementation, there are a number of unavoidable “ad-hoc” parameters
  - Fine-tune them for each genre
- Long term
  - This technique will never overcome difficulty of drum fills, silent breaks, and music that is rhythmically “shuffled” i.e. broken beat
  - Maybe use something more intelligent (classifiers, periodicity of FFT similarity, etc.)

References

[1] Alonso, M., B. David, and G. Richard. 2004. Tempo and beat estimation of musical signals. *Proceedings of the 5th International Conference on Music Information Retrieval*.

[2] Klapuri, A. 1999. Sound onset detection by applying psychoacoustic knowledge. *Proceedings of the IEEE International Conference of Acoustics, Speech and Signal Processing*: 3089-3092.

[3] Proakis, J., and D. Manolakis. 1995. *Digital signal processing: Principles, algorithms and applications*. 3rd Ed. New York: Prentice Hall.