Key Constraint

Chris Martin

Kelsey Francis

https://github.com/kelseyfrancis/keyconstraint/

Overview

- Identify key of piece of music (Kelsey)
  - from audio signal
  - offline (i.e., not real-time)

- Instrument that synthesizes notes (Chris)
  - in detected key
  - on-screen MIDI keyboard

Demo

Key identification

- Pitch class profiles (Fujishima, 1999)
- Machine learning

Key identification (pitch class)

- Pitch class - note in all octaves
  - e.g., C, G, F#
  - one for each note/semitone

- $p = 69 + 12 \log_2 (f / 440)$
  - real, linear *pitch space*
  - 60 = middle C (aka C4)

- pitch class = p mod 12
  - 0 = C, 11 = B

Reference: en.wikipedia.org/wiki/Pitch_class

Key identification (PCP)

1. Spectrum analysis (FFT)
2. For each bin, find pitch class
   - only consider if pitch class within 0.1 of integer
3. Sum magnitudes in each pitch class
4. Normalize

Key identification (machine learning)

1. Extract feature vector from audio signal
2. Label feature vector with key
3. Train classifier
4. Predict label for unlabeled feature vector

Key identification (libraries)

Signal processing

- JTranforms (FFT)

- Machine Learning
  - Weka

- General stuff that Java should already have
  - Guava

Key identification (misc)

- WAVE file I/O
- Cue Sheet I/O
  - split WAV into tracks
- Window function
  - Hamming, Hann, Vorbis

Key identification (experimentation)

- Classifier
  - NNGe, Multilayer Perceptron, Random Trees, Naive Bayes
- Window function
  - Hamming, Hann, Vorbis
- Deviation from pitch class heuristic
  - 0.1, 0.05, 0.2, etc.
- Differences in training and testing music
  - piano vs. orchestra

Key identification (strengths/weaknesses)

- Decent accuracy with limited training data
  - seems to be sensitive to type/number of instruments
- Many wrong results are partly right
  - Circle of fifths, relative minor
- Slow
- Tedious to label songs

Key identification (future work)

- Band pass filter input

- Constant Q transform
- STFT
- Window size/overlap

- Find best classifier (or combination thereof)

- Large, labeled dataset

Reference: en.wikipedia.org/wiki/Constant_Q_transform

Synthesis

- Mozilla Audio Data API
  - http://chris-martin.github.com/webaudio-sandbox/
  - Latency too high

- MIDI input -> Python -> aplay
  - Some performance difficulty, but not insurmountable

Synthesis

- Features
  - Oscillator (comes in various waveforms)
  - Addition
  - Amplitude modulation/enveloping (ADSR)
  - Frequency modulation
  - Memoization into wave tables

- Future features
  - Filters
  - Serialize wave tables to filesystem

Synthesis

- next(t, n) : samples
  - t : time increment (float or numpy.array of floats)
  - n : number of samples requested
  - samples : numpy.array of floats

- liveness() : ['live', 'sleep', 'dead']